DYNAMIC MULTI-OBJECTIVE OPTIMIZATION USING EVOLUTIONARY ALGORITHMS

By
DANIEL HERRING
ORCID: 0000-0003-3825-0411

A thesis submitted to the University of Birmingham
and the University of Melbourne for the degree of
Doctor of Philosophy

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham

School of Computing and Information Sciences
Faculty of Engineering and Information Technology
The University of Melbourne

November 2022

ABSTRACT

Dynamic Multi-objective Optimization Problems (DMOPs) offer an opportunity to examine and solve challenging real world scenarios where trade-off solutions between conflicting objectives change over time. Definition of benchmark problems allows modelling of industry scenarios across transport, power and communications networks, manufacturing and logistics. Recently, significant progress has been made in the variety and complexity of DMOP benchmarks and the incorporation of realistic dynamic characteristics. However, significant gaps still exist in standardised methodology for DMOPs, specific problem domain examples and in the understanding of the impacts and explanations of dynamic characteristics. This thesis provides major contributions on these three topics within evolutionary dynamic multi-objective optimization.

Firstly, experimental protocols for DMOPs are varied. This limits the applicability and relevance of results produced and conclusions made in the field. A major source of the inconsistency lies in the parameters used to define specific problem instances being examined. The uninformed selection of these has historically held back understanding of their impacts and standardisation in experimental approach to these parameters in the multi-objective problem domain. Using the frequency and severity (or magnitude) of change events, a more informed approach to DMOP experimentation is conceptualized, implemented and evaluated. Establishment of a baseline performance expectation across a comprehensive range of dynamic instances for well-studied DMOP benchmarks is analyzed. To maximize relevance, these profiles are composed from the performance of evolutionary algorithms commonly used for baseline comparisons and those with simple dynamic responses. Comparison and contrast with the coverage of parameter combinations in the sampled literature highlights the importance of these contributions.

Secondly, the provision of useful and realistic DMOPs in the combinatorial domain is limited in previous literature. A novel dynamic benchmark problem is presented by the extension of the Travelling Thief Problem (TTP) to include a variety of realistic and contextually justified dynamic changes. Investigation of problem information exploitation and it's potential application as a dynamic response is a key output of these results and context is provided through comparison to results obtained by adapting existing TTP heuristics. Observation driven iterative development prompted the investigation of multi-population island model strategies, together with improvements in the approaches to accurately describe and compare the performance of algorithm models for DMOPs, a contribution which is applicable beyond the dynamic TTP.

Thirdly, the purpose of DMOPs is to reconstruct realistic scenarios, or features from them, to allow for experimentation and development of better optimization algorithms. However, numerous important characteristics from real systems still require implementation and will drive research and development of algorithms and mechanisms to handle these industrially relevant problem classes. The novel challenges associated with these implementations are significant and diverse, even for a simple development such as consideration of DMOPs with multiple time dependencies. Real world systems with dynamics are likely to contain multiple temporally changing aspects, particularly in energy and transport domains. Problems with more than one dynamic problem component allow for asynchronous changes and a differing severity between components that leads to an explosion in the size of the possible dynamic instance space. Both continuous and combinatorial problem domains require structured investigation into the best practices for experimental design, algorithm application and performance measurement, comparison and visualization. Highlighting the challenges, the key requirements for effective progress and recommendations on experimentation are explored here.

DECLARATION OF AUTHORSHIP

I, DANIEL HERRING, declare that this thesis entitled, DYNAMIC MULTI-OBJECTIVE OPTIMIZATION USING EVOLUTIONARY ALGORITHMS is my own work. I can confirm:

- This thesis comprises only my original work towards the degree of DOCTOR OF PHILOSOPHY;
- due acknowledgement has been made in the text to all other material used;
- the thesis is fewer than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices.

Signed: Daniel Herring

Date: 24th November 2022

PREFACE

- This work was supported by the Priestly Scholarship Scheme for joint awards and study at the University of Birmingham, UK and the University of Melbourne Australia.
- The work towards this thesis was completed under the guidance of supervisors Prof. Michael Kirley from the University of Melbourne and Prof. Xin Yao and Prof. Per Kristian Lehre from the University of Birmingham.
- Publications arising from the work presented in this thesis are co-authored by supervisors only.
- The publication status of content within each chapter is detailed below:
  - Chapter 3 contains work published at the *2022 Genetic and Evolutionary Computation Conference (GECCO)* in July 2022, in addition to unpublished material.
  - Chapter 4 contains work submitted to *Swarm and Evolutionary Computation* on 12th August 2022 and is available on arXiv. It also contains work published at the *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* in December 2020.
  - Chapter 5 contains primarily unpublished material not yet submitted for publication in addition to work published in *2019 IEEE Congress on Evolutionary Computation (CEC)* in June 2019.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisors; to Prof. Michael Kirley for his endless support, encouragement, and inspiration; to Prof. Xin Yao for his tremendous knowledge and insights. Both of them have helped me refine every paper and chapter with their wisdom, they have helped me grow as a researcher and have made this entire PhD process more enjoyable. I greatly appreciate all that they have taught me.

I'd also like to thank Prof. Per Kristian Lehre for the regular discussions and check-ins that have been a reliable constant throughout the entire process of completing this PhD. He has been an inspirational presenter and a friendly face at many of the conferences I've been fortunate enough to have attended.

Similarly, my thanks my research committees; to John Bullinaria and Ata Kaban in Birmingham and Alistair Moffat and Shanika Karunasekera in Melbourne. Your insights and feedback helped me greatly.

None of this would have been possible without the Priestley Scholarship scheme and I am very grateful to have been a part of it. It has enabled me to learn from inspirational researchers on opposite sides of the planet and introduced me to lifelong friends in the process. I hope that many other future students receive this fantastic opportunity. I'd also like to thank the financial support provided by both the University of Birmingham's Computer Science Department and the University of Melbourne's Computing and Information Systems department which allowed me to attend international conferences and present my work.

My thanks also to the administrative teams and Birmingham and Melbourne, who handled my many, many questions with great patience. Specifically, to Sarah Brookes and Kate Sterne; they assisted me through some unpredictable challenges with compassion and understanding, Birmingham is lucky to have them.

To my friends in Melbourne, especially to Esty and Steph, and the fantastic community that the CIS department has cultivated, you all made being far from home a lot easier. To my friends in Birmingham, otherwise known as 'Group Four' – the greatest group of friends I could hope for. Also, Bradley and David for their discussions both relevant and not, for proof reading my writing and for making the pandemic lockdowns less isolating.

Finally, many thanks to my family. To my parents and my brother Joe, for their constant support and unwavering belief in me. To Carlos, for his endless patience and encouragement, willingness to listen to my ramblings about algorithms and papers, the constant supply of baked goods and reminding me to take a break - I couldn't have done this without you.

PUBLICATIONS

2019 – Daniel Herring, Michael Kirley, and Xin Yao. (2019) Investigation of Asynchrony in Dynamic Multi-Objective Optimization. *In 2019 IEEE Congress on Evolutionary Computation (CEC). IEEE Press,* 3165–3172. doi:10.1109/CEC.2019.8790270

2020 – Daniel Herring, Michael Kirley and Xin Yao. (2020) Responsive Multi-population Models for the Dynamic Travelling Thief Problem, *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 297-304, doi: 10.1109/SSCI47803.2020.9308388.

2020 – Daniel Herring, Michael Kirley, Xin Yao. (2020) Dynamic Multi-Objective Optimization of the Travelling Thief Problem. *arXiv preprint*, doi: 10.48550/ARXIV.2002.02636. https://arxiv.org/abs/2002.02636

2022 – (Under Review) Daniel Herring, Michael Kirley, Xin Yao. (2022) A Comparative Study of Evolutionary Approaches to the Bi-objective Dynamic Travelling Thief Problem. *Swarm and Evolutionary Computing.* Paper available on request.

2022 – Daniel Herring, Michael Kirley, and Xin Yao. (2022) Reproducibility and baseline reporting for dynamic multi-objective benchmark problems. *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '22). Association for Computing Machinery, New York, NY, USA,* 529–537. doi:10.1145/3512290.3528791

Code repositories:

The code used to produce the results in this work are included in the following locations. They are also available upon request via email.

Chapter 3:
https://github.com/Herring1/Dynamic-Parameter-Testing-for-DMOPs-Ch3

Standalone DPTP platform:
https://github.com/Herring1/Dynamic-Parameter-Testing-Platform-Ch3

Chapter 4:
https://github.com/Herring1/Dynamic-Travelling-Thief-Problem-Ch4

Chapter 5:
https://github.com/Herring1/Multi-Dynamic-DMOP-Test-Functions-Ch5

Contents

List of Figures

List of Tables

List of Acronyms

- **DMO** - Dynamic Multi-Objective Optimization
- **DMOP** - Dynamic Multi-Objective Optimization Problem
- **DPTP** - Dynamic Parameter Testing Platform
- **TSP** - Travelling Salesperson Problem
- **KP** - Knapsack Problem
- **EA** - Evolutionary Algorithm
- **MOEA** - Multi-Objective Evolutionary Algorithm
- **GD** - Generational Distance
- **HV** - Hypervolume
- **HVD** - Hypervolume Difference

Chapter 1

Introduction

1.1 Dynamic Multi-Objective Optimization

Dynamic multi-objective optimization integrates two research areas together to formulate problem scenarios with realistic characteristics, multiple competing objectives and some dependency on, or variation over time. Multi-objective optimization has received much attention over the past 50 years and likewise dynamic optimization problems have been increasingly studied over the past two decades.

In its simplest form a dynamic multi-objective optimization problem can be defined as follows. At least two objective functions with opposing goals that are fulfilled by the values of a decision vector which represents a potential solution to the problem. In order to be ‘dynamic’ some aspect of the problem changes over time, most commonly within the objective functions, such that the goal varies over time or the suitability of particular solutions is subject to change. An illustrative example can be found in the optimization of a financial stock portfolio with objectives for stock value and portfolio diversity (as in [1]). As the value of particular assets change over time, the performance of a portfolio solution will also change over time, meaning that a better set of solutions may be found for each dynamic interval of the problem.

Various definitions for dynamic problems can be found throughout the literature, however a major distinction exists between problems that can be solved ‘offline’ to those that require an ‘on-line’ optimization with methods such as evolutionary algorithms. For problems that can be solved offline, the pattern of changes can be incorporated into a static (unchanging) objective function. For example in dynamic energy pricing [2] or the portfolio optimization example as before, a window, error bound or margin can be used to encompass the range explored by the dynamic states, similar to how uncertainty can be managed [3]. Alternatively, if the dynamic change events can be assumed to be known *a priori* the problem can be decomposed from successive dynamic intervals into a series of independent static optimization problems. The scope of dynamic problems covered in this work does not include these offline problems. Similarly, methods that aim to find robust solutions are beyond the scope of this work; these are algorithms that ignore the distinct differences in problem states in favour of finding solutions that adequately achieve across the range of dynamic intervals.

Dynamic multi-objective optimization (DMO) encompasses numerous subtopics each of which has a rapidly developing corpus of research. A key part of this research comes from the scale of the potential solution space and the properties of DMO problems (DMOPs) which makes the application of evolutionary computation techniques particularly suitable. Generally, however, these topics are motivated by the unique challenges that DMO problems pose, but are grounded in their 'static' (non-dynamic) foundations. These include the formulation of novel benchmark problems, the design of evolutionary algorithms to find solutions for DMO problems, including the development of effective response mechanisms to dynamic events and the effective and informative measurement and evaluation of their performance. The additional complexity introduced by considering variations in the problem state over time creates a multitude of additional research directions, including change detection techniques, parameterization of the dynamics and novel visualization techniques. A common goal across the majority of DMO research is the effective tracking of an optimal set of solutions representing the trade-off surface between the multiple objectives throughout the dynamic intervals (problem states) of a problem instance. This is a sufficiently challenging task in itself and as such considering the selection of any specifically found solutions for deployment is beyond the scope of this work. Mechanisms for the selection of single or multiple solutions from the optimal set is an active research topic that remains an open question, the outcomes of which are also pertinent to static multi-objective optimization.

A major driving force for optimization research generally is to provide useful solutions to real problems, or simplified formulations of them, in order to solve directly or provide a mechanism with which to solve tasks. Additionally, developing understanding at a fundamental level is a key motivator in DMO, particularly in terms of: the components of problems and their impacts; the interaction between problem features and algorithms, and the efficacy, efficiency and complexity of developed mechanisms.

However, there are significant gaps in practices that thus far have hindered the cohesive progress towards understanding the impacts of dynamics in DMO, as well as significant opportunity to improve the realism and the range of dynamic characteristics that exist within benchmark functions. These are the key focus of this thesis, the motivations for which have been formulated into a number of specific research questions.

1.2 Statement of Research Questions

Firstly, the existing approaches to evaluating performance are hindered by the lack of consideration of the scale of the possible dynamic instance space generated by the parameters controlling the nature of changes. Reproduciblity of results and comparability of benchmark instance sets must be reconciled through the establishing of a comprehensive testing framework for DMOPs. Therefore we can frame the research questions surrounding this topic as follows:

> **RQ1:** *How can we characterise the impacts of dynamic parameters in DMOPs for more meaningful benchmark evaluation?*

The following key areas are addressed to answer this question:

- Investigation of the impacts of the frequency and severity of change parameters allowing for the establishing of a baseline of expectable performance, integral for cohesive progress in DMO.
- Definition of the possible dynamic instance space for benchmarks using a comprehensive set of frequency and severity combinations, allowing for a contrast to the scope of the previous literature.
- Demonstration of the importance of informed selection of dynamic instances of DMOP benchmarks, as well as suitable baseline comparison algorithms, whether they are static MOEAs or simple dynamic responses.

Beyond existing DMO practices, there are also significant limitations in the representation of realistic dynamics and multi-objectivity in the combinatorial problem domain. The development from realistic static problems to incorporate a variety of dynamic events provides challenging scenarios for which we must develop novel ways to find solutions and response to the dynamic events. Given the lack of existing DMOP benchmarks in the combinatorial domain that contain strong justifications for their dynamic components, we can develop existing static problems to provide more realistic test instances. The following question can be formulated:

> **RQ2:** *How can we incorporate dynamics into existing static problems to generate and find solutions for realistic combinatorial test instances that better reflect characteristics observable in real-world systems?*

Exploitation of problem knowledge can be employed in combinatorial problem classes; solvers are employed for problem components where robust methods have been established [4]–[6]. The extension of this principle to the newly defined dynamic instances of a realistic, bi-objective combinatorial problem, the Traveling Thief Problem, provides an additional research question:

For this question the following key points are addressed:

- Justified development of a variety of novel dynamic problems within an established static problem framework allowing for the leverage of existing knowledge to formulate a well-supported contribution.
- Demonstration of methods to find solutions in the proposed dynamic instances based on performance observations using combinations of exploited problem information for the development of a dynamic response method.
- Highlighting of the suitability of agile approaches to finding solutions through the comparison of performance and efficiency with adapted heuristics known to provide good performance in static cases.
- Further improvement in performance through informed development of a multi-population algorithm utilizing previously observed algorithm behaviours and exploiting search landscape localization characteristics.

The realistic characteristics present within existing problems do not consider features that are present within many real world systems. The most obvious of these is multiple dynamic components within a problem. This adds additional challenge to an already complex problem class and therefore an investigation into the key considerations, observations and recommendations for navigating this novel problem class is required. Therefore, the research question to be addressed is as follows:

> **RQ3:** *How can we apply similar methodologies for both continuous and combinatorial domain problems to define the key features, challenges and potential approaches to multi-dynamic multi-objective optimisation problems?*

The following points address this research question:

- Provision of relevant examples of multi-dynamic problems by adapting existing continuous and combinatorial benchmarks and problems.
- Evaluation of the impacts of frequency and severity parameters in combinatorial instances and a distinction between coincidental and non-coincidental changes.
- Identification of performance motifs and characteristics in baseline performance across the possible dynamic instance space for continuous instances, providing insights generally for the problem class.
- Documentation of the key challenges present for multi-dynamic problems and the formulation of recommendations and guidance for future experimentation.

These research questions highlight some of the key challenges within the field of DMO currently. They are relevant to research across the field and the outcomes of undertaken work to answer these questions, provides meaningful insights to the wider field.

1.3 Novel Contributions

Together, the stated research questions showcase the identified fundamental gaps within practice and understanding that, once addressed, allow for meaningful, cohesive and efficient progress to be made within DMO. Whilst seemingly disparate in their targets, the overall contributions of this thesis suggest improvements to the currently available roster of realistic test problems; a reframing of previously incongruent practices for continuous instances; and pioneering approaches for investigating novel DMO problem classes with additional realistic characteristics. The contributions in regards to these questions have significant specific impact but share widely applicable insights and relevant outcomes for other aspects of DMO. This relevance is addressed in more detail within each of the chapters.

The limitations surrounding experimental protocols for DMO benchmarks, as highlighted in RQ1, are addressed in Chapter 3. Identifying specific limitations of existing methodology and the general challenges accompanying dynamic problems, allows for the establishment of a comprehensive investigative procedure for DMO problems going forward. An approach is presented that generates relevant results that both explore the impacts of dynamics, determine the effectiveness of simple responses and enable informed selection of dynamic instances for future experimentation. A comprehensive investigation using the parameters which define the nature of dynamics in well-known DMO benchmark problems highlights the scale of the possible instance space for DMO problems and the challenges presented by inconsistent parameter usage. Illustration of performance of well known multi-objective evolutionary algorithms generates a baseline of expectable performance where they succeed and can inform on future design of experiments where they do not.

Chapter 4 addresses the identified limitations within combinatorial DMO given in RQ2. Justifications are provided for the variety of dynamic change events that are proposed for the Travelling Thief Problem. The resulting array of dynamic instances allows for investigation of a diverse range of realistic scenarios that can more closely replicate real world systems. An iterative design approach based mechanisms to find solutions to the novel problem instances is adopted due to their complexity and the specificity of existing heuristics for the static problem. This also limits performance comparisons to adapted versions of these static heuristics. As a result of representative testing of multiple patterns of dynamics, the large quantity of data collected and comparisons required demands novel visualization techniques and meaningful statistical testing.

The culmination of the previous two chapters is a forward-looking approach in Chapter 5, with the first steps towards understanding increasing complexity in dynamic systems, as stated in RQ3. Given the development of tools to comprehensively investigate the dynamic instance space in continuous problems, the application to a previously overlooked, but highly relevant problem class is undertaken. The seemingly minor conjunction of multiple dynamic components within a problem is characteristic of real world systems yet has been ignored in previous research. There are obvious and significant challenges associated with this development, however the work presented here draws on the methodologies established in the preceding chapters to clarify the challenges specific to these problems and their impact on established methods. A natural opportunity to extend investigations to explore the proposed dynamic travelling thief instances together with continuous problem examples, provides a conclusive foundation for future investigation of these problems.

1.4 Chapter Organization

The remainder of this thesis is organized as follows. A comprehensive review of DMO literature is presented in Chapter 2: the fundamentals of DMO and the varied and overlapping definitions for problem classification; an original summary of available continuous DMOP benchmarks is provided together with an in-depth review of combinatorial benchmarks, dynamic algorithm mechanisms and performance measurements. Both recent and fundamental works are considered to provide the motivations for the research questions stated above, highlight the gaps in current knowledge and provide the groundwork for the contributions provided in the following chapters.

As above, Chapter 3 focuses on the establishment of a practices for reproducible and meaningful experimentation on DMOP benchmarks by considering the scale of the dynamic instance space generated by the frequency and severity of change parameters. Chapter 4 addresses the paucity of realistic dynamic and multi-objective optimization benchmarks in the combinatorial domain through the definition of a number of novel dynamic formulations of the Travelling Thief Problem, together with development and evaluations of suitable responsive algorithms. Chapter 5 provides preliminary and exploratory investigation into multi-dynamic problems, providing observation-based insights and establishing the key challenges and recommendations for these as a logical next step for the development of realistic problem classes within DMO. Finally, a summary of the contributions and conclusions, together with an evaluation of the limitations and considerations for potential future work are given in Chapter 6.

Chapter 2

Literature Review

This chapter gives a general overview of literature relevant to Dynamic Multi-objective Optimization (DMO) in terms of problems, algorithms and performance measurements and their relevance to the presented contributions. A more detailed review of the specifically relevant literature is presented within each of the following chapters.

2.1 Introduction and Motivations

The diversity of topics within DMO requires the consideration of all aspects of problems, performance and algorithms. However, three opportunities present clearly from the current state of DMO research. Fundamentally, a previous lack of consistency in DMOP practices prompts the establishing of baseline practices that allow for coherent, cohesive and meaningful progress. Further opportunity exists to fill the gaps in combinatorial DMOPs for the definition of a useful benchmark framework that allows complex and realistic scenarios to be examined that are similar to real world application domains. Beyond this, consideration of the currently undocumented characteristics of real world systems as features of optimization problems, provides scope for exploration both into the impacts and challenges they present, but also into potential practices for explanation and understanding.

To expand on the first of these points, established codes of practice and consensus on protocols exists in many fields of research. The purpose of these is to ensure comparability of results and consistent reporting and to facilitate and expedite research so that progress happens more rapidly. Research on DMOPs has some consistency but lacks a cohesive approach to experimentation on dynamic instances generally. This makes comparisons of results difficult, or if comparable the scope of relevance is narrow. There are strengths and weaknesses of different practices of sampling, however consideration of parameter impacts contributes appropriate instance sampling. Generally, a more informed approach towards experimental procedures for DMOPs is required.

Secondly, significant opportunity exists for a middle-ground between simplistic benchmark problems and specific real world scenarios. Specifically in the multi-objective and combinatorial problem domain, there is a lack of dynamic problems with realistic or challenging aspects incorporated. The numerous potential applications to transport, logistics and fleet management problems, as well as mobile and network communications, provide the motivation for a realistic DMOP framework in the combinatorial space.

Finally, the variety, complexity and scale of problems examined within existing DMOP literature is vast. However, the continued incorporation of realistic characteristics of real world systems will help to drive progress towards better algorithms, better solutions and new application opportunities. Highlighting the key challenges and improving the understanding of algorithm behaviours, search landscapes and problem difficulty are also significant outcomes of establishing and attempting to solve novel classes of problems. Augmenting problems in both combinatorial and continuous domains, with features and characteristics to more faithfully portray realistic systems, allows for informed algorithm development as better test instances are available for evaluation.

2.1.1 Organization of Review

A consideration of the relevant literature that provides assistance in conceptualization, motivation and understanding is provided in the following sections. Beginning with a brief consideration of the foundations and historically pertinent contributions, the directly relevant emergent works provide the context for the specialized literature on dynamic multi-objective optimization that follow. A description of the common practices, including classification of dynamics, the consensus on protocols and measurement of time, definitions of frequency and severity are included. These highlight the requirement for a cohesive approach in other aspects of DMOP research. A collection of continuous and combinatorial benchmarks as well as relevant work on real world problems is documented to provide scale and context for the contributions. The diverse range of algorithms used to find solutions to DMOPs and benchmarks is given here to demonstrate an awareness of the variety of powerful alternatives to find solutions to these problems. Describing the quality of found solutions and the performance of these algorithms is a challenging task and there are variety of approaches, both novel measurements specifically defined for DMOPs and those adapted from the non-dynamic (static) metrics. Finally, the key motivations are revisited in the context of the surveyed literature.

2.2 Foundations of Dynamic Evolutionary Optimization

Evolutionary Algorithms (EAs) comprise a variety of methods that use evolutionary operators and concepts of natural selection and survival of the fittest in order to optimize a given problem. The foundation of the field comes from the Genetic Algorithms (GAs) popularised by John Holland [7], [8] and developed by David Goldberg [9]–[11] soon after. Klockgether & Schwefel applied evolutionary methods for the design of nozzles to optimize air flow [12]. Grefenstette also used a GA to tune the parameters of a GA [13]. Together these works provided the foundations for the extensive and diverse research corpus that has developed over the past decades. Numerous other Evolutionary Computation (EC) methods have been defined and reviewed since including Differential Evolution [14], Genetic Programming [15], [16], Ant Colony Optimization [17], Particle Swarm Optimization [18].

Early Approaches for Dynamics

Early approaches for dynamic problems and problems with time varying components set the groundwork for the contemporary taxonomy of algorithms and problems used. For example, Wierzbicki investigated the dynamic properties of multi-objective optimization [19] after which, Cobb and of Grefenstette proposed hypermutation to boost diversity in the population in response to dynamic changes [20], [21]. Solution 'age' was also used as an early diversity maintenance technique in single objective DOPs [22]. Memory based methods for DMOPs were primarily introduced by Branke et al. [23]. Early multi-population methods were applied to single objective DOPs and proposed restrictive crossover to preserve diversity and avoid stagnation [24], [25]. The Moving Peaks Benchmark [23], Generalized Moving Peaks Benchmark [26] and the dynamic XOR [27] remain some of the fundamental contributions to the generation of dynamic test problems. Since these early works, there has been an explosion in the variety, complexity and ingenuity of proposed algorithms, benchmarks, measurements and applications for DMO research.

The basic definition of a DMOP with a scalable number of objectives is given below in Equation 2.1. This is adapted from a default definition for a continuous multi-objective benchmark function; the dynamic aspect is included here within the objective functions as a dependence on the variable t.

$$\begin{aligned} \vec{\mathbf{x}} &= [x_1, x_2, \dots x_n] \\ \vec{\mathbf{F}}(\vec{\mathbf{x}}, t) &= [f_1(\vec{\mathbf{x}}, t), f_2(\vec{\mathbf{x}}, t), \dots f_M(\vec{\mathbf{x}}, t)] \\ h_1(\vec{\mathbf{x}}) &\leq 0, h_2(\vec{\mathbf{x}}) = 0 \end{aligned} \tag{2.1}$$

Types of Dynamic Change

There are a number of ways that dynamic change events have been incorporated into multi-objective optimization problems, some of which have received more attention than others. Table 2.1 highlights the major types of changes and the example modifications to the previous default DMO formulation in Equation 2.1.

Table 2.1: Types of dynamic change event and their example representations from the dynamic optimization literature.

Description of change	Modification to Equations (t)	Example
Objective functions	$\vec{\mathbf{F}}(\vec{\mathbf{x}},t) = [f_1(\vec{\mathbf{x}},t), f_2(\vec{\mathbf{x}},t), \dots f_M(\vec{\mathbf{x}},t)]$	[28], [29]
Decision variables	$\vec{\mathbf{x,t}} = [x_{1,t}, x_{2,t}, \dots x_{n,t}]$	[30]
Number of objective functions	$\vec{\mathbf{F}}(\vec{\mathbf{x}},t) = [f_1(\vec{\mathbf{x}}), f_2(\vec{\mathbf{x}}), \dots f_{M(t)}(\vec{\mathbf{x}})]$	[31]–[33]
Number of decision variables	$\vec{\mathbf{x}} = [x_1, x_2, \dots x_{n(t)}]$	[32], [33]
Range of decision variables	$\vec{x} \in \Omega(t)$	[34]
Constraints	$h_1(\vec{\mathbf{x}},t) \leq 0, h_2(\vec{\mathbf{x}},t) = 0$	[35]

2.3 Classifications of Dynamic Changes

Early attempts to classify changes in non-stationary or dynamic problems were provided by Trojanowski based on predictability [36]. Three categories are defined:

- Random Changes: successive problem states are independent of one another.
- Non-random and non-predictable changes: problem states are not independent but the relation is so complex so to be unpredictable.
- Non-random and predictable changes: dynamic changes are deterministic, whether cyclical or non-cyclical, the sequence or perturbations can be predicted and exploited to improve the search process.

Categorization by these groupings omits information about the impacts of changes, but is useful when evaluating the performance of algorithmic responses based on predictability of changes.

2.3.1 Classification based on Pareto Set & Pareto Front changes

Farina, Deb & Amato [28] proposed a classification for dynamics based on their impacts on the Pareto-Optimal Set (POS) and Pareto-Optimal Front (POF), (these being the best performing selection of non-dominated solutions in the decision and objective space respectively). The 'Type' system consists four categories based on the impacts of the dynamics: only the POS changes (Type I), both the POS and POF change (Type II), only the POF changes (Type III) and dynamic changes do not impact either the POS or POF (Type IV). This can also be called effect-based classification [37].

The *Type* classifications are widely used in DMO literature, being the primary system mentioned in major surveys of the area [29], [38]–[41] and commonly reported alongside problems and benchmark suites defined since.

This method of classification by impacts is both intuitive and useful, offering a simple grouping strategy that assists in informing the construction of diverse test sets to cover a variety of problem types. However, there are some limitations in the grouping of problems that results from this. For example, ensuring a range of *Types* are included in a problem test set may not provide the breadth of experimentation required to make meaningful con-

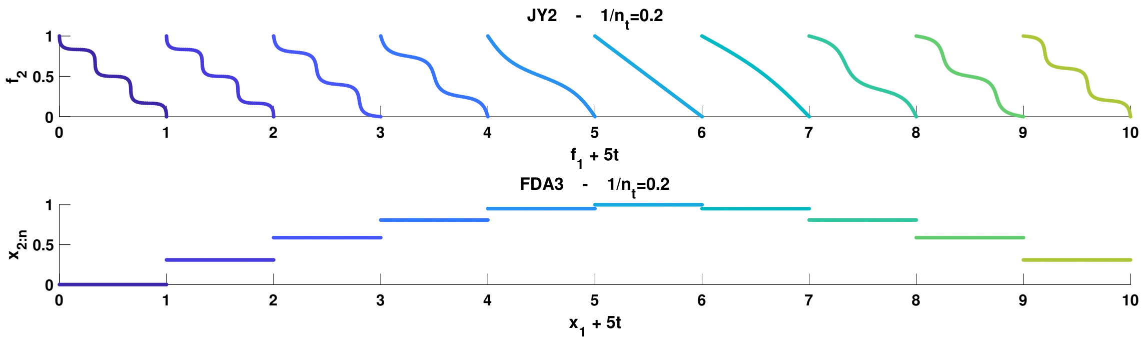


Figure 2.1: Example dynamic change impacts using (upper) POF changes in the Type II JY2 problem [29] (lower) POS changes in the Type I FDA3 problem [28]. Consecutive dynamic interval states are shifted along the x-axis for comparison.

clusions on algorithm performance. For example, broadly speaking, if an algorithm that can effectively navigate the decision space of a problem can also cope with (or mitigate the impacts of) the dynamic changes in a Type I problem, then, by natural extension, it should also be able to handle Type II & Type III problems (unless the change is in the density of solutions in the objective space, for instance). Therefore the different Types may not represent distinct challenges. Furthermore, the possible scope of dynamic change events which can constitute different *Types* results in a variety within each class that makes classification trivial in terms of conclusions. For instance, if the dynamic change event was in the *number of* decision variables, most basically this would represent a Type I problem as much as a change in the objective function equations may. Therefore, designing an algorithm targeted at solving 'Type I' problems may not handle the range of possible dynamic change events that comprise this category. Whilst useful and informative, the Type classification should be used in conjunction with other information about the nature of the dynamics in DMO problems. Figure 2.1 illustrates some examples of POF and POS changes visible in DMO benchmarks. There exists a variety of other classifications for dynamic and DMO problems that have been proposed since, many of these are complimentary and can be used in conjunction with a *Type* label to provide a more complete picture of dynamic features.

2.3.2 Classification based on Frequency, Severity & Recurrence

Grouping by frequency of changes, the severity of change or the recurrence of dynamic intervals (problem states induced by a dynamic change event) has been proposed as an alternative classification [42] (recast in [43]). These methods of classification result in grouping of problems that may suit particular algorithmic response mechanisms. For example, grouping similar problems with rapid frequency of changes will suit algorithms that ensure diversity of solutions in the response mechanism. This approach to problem classification allows for composition of test sets with a variety of characteristics to provide challenges that may not be accurately described under the Type system. However,

Table 2.2: Order based classification and the assignment based on problem state relationship. Reproduced from [46].

Dynamic Component	Independent System States	Correlated System States
Parameters	1^{st} order	3^{rd} order
Functions	2^{nd} order	3^{rd} order
Environment	4^{th} order	4^{th} order

simply classifying problems by the severity of their change or by how often the same dynamic intervals occur may not effectively summarise a set of problems. Combination of information, to provide a comprehensive description of DMOP characteristics removes ambiguity, clarifies justifications for including particular problems and allows for accurate conditioning of conclusions.

2.3.3 Classification by Spatial and Temporal Features

Goh expands on these previous classifications specifically for MO problems by separating the spatial and temporal components to dynamic change events [44]. This system combines aspects of the systems proposed before, including the the impacts of change [28], the pattern and predictability [45] and the nature of changes [42] (excluding the severity of change). Spatial components are separated into physical (changes in the POS, POF, the fitness landscape or a combination of these) and non-physical relating to the nature of the change: ‘random’ changes; ‘trend’ change where a pattern is present between dynamic intervals; and ‘periodic’ changes where a set of dynamic intervals are revisited in sequence. A problem may have both ‘trend’ and ‘periodic’ changes or just one of them. The temporal features relate to the scheduling of change events: None (static problem), random (non-fixed frequency), Fixed (constant frequency), Scheduled (can be random or fixed, but is predetermined), Conditional (changes triggered by satisfying a predefined condition). These properties are compositional in the description of a DMOP and provide insight into the affectation of the dynamics as well as their impacts on the problem state, the latter of which is only covered in the *Type* system.

2.3.4 Cause-based Classification

An alternative, similarly compositional, component-based classification is given in [46]–[48]. Four *orders* are proposed: 1^{st} order – Dynamic Parameter Evolution; 2^{nd} order – Dynamic Function Evolution; 3^{rd} order – Dynamic State-dependency Evolution; 4^{th} order – Online dynamic evolution with environmental changes. With the clarification of order assignment provided in Table 2.2 (reproduced from [46]):

The schema does not explicitly consider problems with dynamic constraints and there is significant information missing about the impacts and nature of the dynamics. An example is given for a 3^{rd} order problems as the multi-objective moving peaks benchmark and for 4^{th} order, modification of the MNK-landscape problem is suggested [49].

Additional classifications also exist within the literature, for example Jin and Sendhoff proposed a simple classification based on the movement of an optimum through parameter space [50].

Generally, all of these different classifications for DMO problems are valid, however different information is used to label and categorize instances of problems. Logically, the motive for employing difference scheme suits the investigation and design of experiments of examining particular hypotheses. For example, a classification that separates problems based on the predictability of dynamic changes may be most useful to evaluate the limitations of algorithms that use prediction in their dynamic responses. Conversely this might highlight the limitations of such a method, where generally good performance of novel algorithms is most often the contribution of research articles. The choice of classification schema can therefore be driven by experimental design principles and goals, however the familiarity and popularity of opposing schema may also play a role in adoption.

2.3.5 Common Practices: Measuring Time, Onset, Change Detection, & Known vs. Unknown events

Across the DMO literature, most commonly time is measured in generations, however there are a number of alternatives including the number of function evaluations [35], [51]–[55], or in real time (e.g. seconds or minutes) [56]–[60]. The latter of this is most commonly used in realistic problem scenarios or where real data is used in simulations. The value of t, shown previously in Equation 2.1 controls the current problem state at a given time. It is updated according to Equation 2.2 and depends on the frequency and severity of change. Commonly t is a decimal value, and the increment in its value controls the change to a new problem state and signals a new dynamic interval. The size of the change in its value denotes the magnitude of the change event ($\frac{1}{n_t}$), and the incidence of the changes themselves are controlled by the frequency of change parameter, τ_t. This formulation first appeared in the FDA problem definitions from Farina et al. [28] and has been widely adopted since.

$$t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor \tag{2.2}$$

where n_t is the severity of the change, τ_t is the frequency of change and τ is the current iteration (generation).

There are two commonly adopted practices when it comes to the the appearance of the first change event. Given a frequency of τ_t generations between change events, many works opt for a delayed onset of a fixed number of generations. If the length of the onset delay is at least greater than τ_t, this provides an algorithm additional time to approximate the Pareto Front in the first interval. Commonly a value of 50 generations is used [32],

[61], [62], however other studies have used values of 100 [63]. The alternative is for no delay to occur and the first change occurs after exactly τ_t generations.

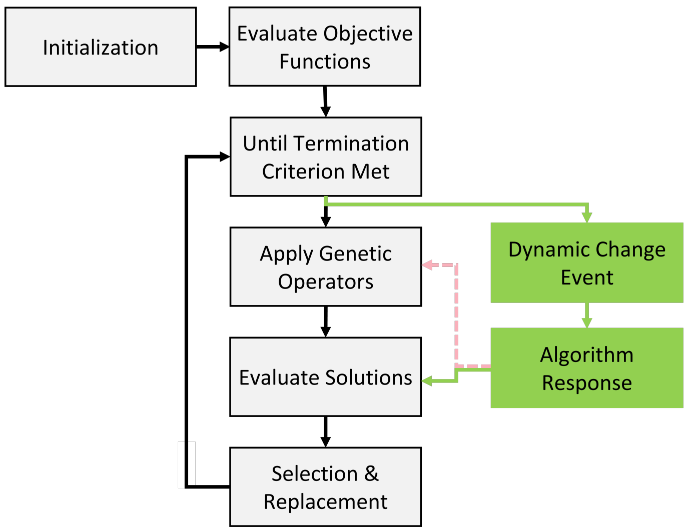


Figure 2.2: Introduction of dynamic problem changes in algorithm process. The dashed arrow indicates that some responses may skip the application of genetic operators in the generation of change.

A major topic of research within DMO concerns strategies to detect changes in the problem state. In the example algorithm process shown in Figure 2.2, the change events occur after a round of genetic operators, or more generally at the start of a generation/iteration. Widespread usage of re-evaluation based detection methods can be attributed to their effectiveness. Under this scheme, a proportion of the population is re-evaluated and if the objective values are different to the previous evaluation (despite no additional operations having been performed) then a dynamic change must have occurred in the problem. Some works use a distributed slice of the population [64], some use additional solutions external to the optimizing population with fixed decision variable values, known as sentinel solutions [65], [66]. Investigations have specifically addressed the selection of sentinel solutions for detecting changes [65], [67]. Alternatively, the changes can be detected based on algorithm behaviour and performance [68] or the average population fitness [69]. Boulesnane & Meshoul [70] provide a recent commentary and survey on the utility of change detection mechanisms.

Generally, however, as algorithmic responses for DMOPs are the focus of many recent papers, the changes are usually prescribed and the response is automatically triggered after a change occurs. There is limited work looking at non-fixed frequency [71] and stochastic sequences of changes in continuous problems [29], [72].

The variety in settings and practices for DMO experiments described above add to the difficulty of ensuring the reproducibility of results. Clear and concise reporting of the problem parameters, algorithm settings and other settings has improved in recent literature, however the accurate conditioning of results and scope requires further attention. The key challenges facing reproducibility are different for continuous and combinatorial problems, however there are some consistent obstacles. For example, in combinatorial problems, such as dynamic instances of the Travelling Salesman Problem (TSP), Knapsack (KP) and Dynamic Vehicle Routing Problems (DVRPs), a sample of stochastic instances can be used due to the scale of the possible dynamic instance space. These challenges are addressed generally in Section 2.6 and in each of the contribution chapters of this thesis.

2.4 Dynamic Multi-Objective Optimization Problems (DMOPs)

2.4.1 Continuous Dynamic Multi-Objective Optimization Benchmark Problems

A large variety of benchmark functions have been proposed for DMO experiments. The proposition of novel benchmarks has usually been based on providing a testing environment or a problem feature that does not appear in existing suites before it. However, a key notion of balanced, diverse and challenging problem suites is often hard to achieve. Table 2.3 summarises the key benchmark suites in DMO, together with the number of problems, the dimension and the key features that they provide compared with other test functions.

Problem Generators: Features, Flexibility & Usage

A number of problem generators have also been proposed for DMO. These enable the construction of novel problem scenarios by setting a number of parameters or combining distinct function components. Jin and Sendhoff [50] proposed their framework for generating DMOPs based on the dynamic weighting of several objective functions to provide a single fitness value and there has been some limited usage [87], [88] since its inception. Other generators include the DSW and DTF methods proposed by Mehnen et al. [74]. These are capable of generating a variety of DMOP problems through the selection of 6 parameters that control aspects of the Pareto Front geometry. Tang et al. [89] proposed a generator for problems with a scalable number of objective functions, and Gee et al. [81] provides a framework of compositional elements used to generate different test functions – similar to the construction of the WFG suite [90] in static MO research. Recently, together with the suite of proposed DMOP benchmarks, Jiang et al. [29] proposed a general methodology for the construction of novel problems for DMO research. Throughout the literature, there is limited further use of problem generators beyond their defining papers. Often, there is a preference for established problems and suites both for ease and comparability. However, the motivations for benchmark selection; the examination of algorithm performance against problems with specific characteristics, may be better suited to the use of problem generators. The issue of reproducibility and comparability, particularly in terms of considering the possible dynamic instance space for a problem, may be compounded by more widespread usage of these generators.

Table 2.3: Continuous DMOP Benchmark Suites based on comprehensive sample of literature (papers that cite [28]), M indicates any number of objective functions can be specified. Usage figures are indicative rather than exhaustive.

Suite	# Problems	# Objectives	–	Remarks	Usage
FDA (2003) [28]	5	2(4), 3/M(1)		POF geometry, POF/POS shift, changes in distribution/density of solutions,	84
FDA2$_{mod}$/3$_{mod}$ (2006/2009)[73], [74]	2	2,3		Modification clarify changes in convexity/concavity	5
FDA5$_{iso}$/$_{dec}$ (2008) [75]	2	3		Isolated and Deceptive POFs	4
dMOP (2009) [76]	3	2(2), 3(1)		POF solution density changes, combined POS/POF solution density changes	44
HE (2013) [41]	10	2		Various POS/POF change types, complex POF geometries	8
DMZDT (2010) [55]	4	2		Adapted from ZDT functions	5
WYL (2010) [55]	1	2		Serial composition of multiple objective functions	3
F (2014) [77]	4	2(3), 3(1)		Complex POS geometries	15
DIMP (2010) [78]	2	2		POS changes with unique decision variable impacts	7
ZJZ (2007) [79]	1	2		Non-linear linkages between decision variables	9
DSW (2006) [74]	3	2		Proposed problem generation using parameter-controlled function components	2
UDF (2014) [80]	9	2(8), 3(1)		Variety in POS/POF geometry changes, disconnected POFs	9
DF (2018) [34]	14	2(9), 3(5)		Competition set composed of novel and borrowed problems: mixed convexity/concavity, variable linkage, range and bounds of POF and POS, solution density, disconnected/holes/degenerate POF regions	12
GTA (2017) [81]	24	2		Composable problem framework with additive and multiplicative components allowing generation of instances with a variety of characteristics	2
T (2011) [33]	4	2(3), M(1)		Time linkage between variables	1
JY (2016) [29]	10	2		Justification for dynamic change types from real world examples, POF geometry/disconnected, POS shift, solution density, mixed concavity/convexity, stochastic dynamic interval changes	10
SDP (2019) [32]	15	2/M		Justification for dynamic change types from real world examples, Variable number of decision variables, objectives,	1
RDP (2021) [72]	8	2(4), M(4)		Randomised (stochastic) sequence of dynamic intervals, scalable number of decision variables, Deceptive POF, mixed concavity/convexity in POF	1
CLY (2018) [31]	6	2, M		Changing number of objectives, (increments of 1)	1
FUN (2018) [82]	3	2		Rotational changes of POS	3
LF (2022) [83]	6	2		Variable change types within an instance	1
DCTP (2015) [35]	4	2		Constrained problems, tunable PF difficulty with infeasible zoning and definition of disconnected regions	4
TP (2008) [75]	2	2		Discontinuous and convex POF	1
WJL (2014) [84]	2	2		Non-linear correlation between decision variables	1
GCM (2005) [85]	3	2(1), 4(2)		Similar to WYL, change in the objective function definition rather than time depedency	1
LDE (2018) [86]	4	2		‘Less-detectable’ environmental changes, partially time-variable POS/POF	1

Key Suites: Origins, Limitations & Justifications

Since its inception almost two decades ago, the FDA suite [28] has been the most widely used in DMO literature to evaluate the performance of novel algorithms. The 5 continuous benchmarks proposed covered the first three of the major classifications under the *Type* system, with POS shifts, POF shifts and geometry changes, solution density changes in two and three objectives. These problems were adapted from static multi-objective benchmarks in the ZDT suite [91] and the DTLZ suite [92]. Farina et al. also provided a number of combinatorial examples of DMO problems. A basic multi-knapsack problem with potentially dynamic profits, weights and capacity was proposed without experiments. A multi-objective dynamic TSP problem was proposed: each objective has its own weight function and the cities swap across a circular arrangement. Both of these definitions provide a simplistic example of the possible formulation of combinatorial or discrete DMO benchmarks, however they have not received much direct usage in the literature compared with the proposed continuous benchmarks. Finally, an optimal control problem is formulated to approximate more closely a real world scenario. In the context of more recently proposed benchmark functions, the FDA problems are simplistic [29], [32]. However, they provided the first examples of DMO problems upon which dynamic response mechanisms could be evaluated and compared. This paper serves as the foundation for much of the DMO literature since and sparked the development of hundreds of algorithms and dozens of further benchmark functions.

As shown in Table 2.3, new benchmark suites and modifications to existing problems have been defined to incorporate additional characteristics for algorithms to handle. These include mixed concavity to convexity in the Pareto Front or a change between the two, deceptive POS/POF geometry, POFs with disconnected or isolated regions and degeneracy.

More recently proposed suites, such as the SDP [32] and JY [29] sets, contain benchmarks with dynamic natures inspired by real world problems or application studies with specific characteristics. Coherent, meaningful design of problems is as important as the informed selection of the range of dynamic instances of them used in experiments.

Composition of experimental sets of benchmark functions varies wildly in the previous literature. The usage statistics in Table 2.3 highlight where at least one of the suite's problems has been used in experiments, however, there are few that use all problems in a proposed suite. Common practice uses a sample from multiple suites, which can be for a number of reasons, including:

- Availability of source code and difficulties in replication or implementation from paper descriptions.

- Deliberate selection of benchmarks with easily handled characteristics, or composition of range of characteristics.
- Limitations of visibility of alternative benchmark functions (newer suites may be less well known).

All of these contribute to the experimental inconsistency visible in previous literature. Several works specifically address this challenge and purport the importance of diversity in experimental test sets [37], [41]. Other areas with EC have proposed standardised test sets, such as COCO for (static) multi-objective optimization [93], which comprises 55 bi-objective continuous problems and has been widely adopted.

Discussion on Selection of Problem Instances for Experiments

Helbig & Englebrecht [41] also highlight the importance of testing a diverse range of frequency and severity of change parameters with DMO experiments. Despite the clear requirement of strong conclusions to be based on a comprehensive set experiments, the range of dynamic instances examined is very limited as standard. As described previously, a dynamic instance refers to a specific instance of a problem generated by using a particular set of parameters that describe the dynamic changes that occur within it. Most simply the dynamic instances can be controlled using the frequency (τ_t) and severity (n_t) parameters, but the starting value of t, the number of decision variables and the selection and transformation of performance measurements can all results in incomparable instance testing. Historically, a single combination of frequency and severity is commonly employed in experiments, usually influenced by the combination used in the defining papers for each benchmark function. Recent works examine up to eight combinations [62], [94], [95], but the size of the potential dynamic instance space is much larger than this. The composition of a diverse set of benchmark functions is a valid pursuit, however consistent practice of dynamic instance selection is equally important in terms of providing relevant and comparable results. This topic, and solutions to this question are explored further in Chapter 3. Generally, the current state of the DMO literature highlights the requirement for a coherent and cohesive approach to experimentation. Lacking so far, the consistency of reporting results and problem parameters is crucial to the relevance and applicability of conclusions. A generalized and logical approach that provides comprehensive evaluation of algorithm performance whilst highlighting the problems or instances that should be targeted by further research, is one of the key, currently unfulfilled, targets for DMO research.

2.4.2 Combinatorial Dynamic Multi-Objective Optimization Benchmark Problems

As with continuous benchmark problems, there is a variety of combinatorial and discrete optimization problems that EC techniques have been applied to. Many of these are derived as standard scenarios from real world applications or to provide a simplified formulation of a complex system. In terms of DMO, the Travelling Salesperson Problem (TSP) has been adapted to consider dynamic and multi-objective characteristics [69], [96], [97]. The established library of static, single-objective TSP problems contained in TSPLIB [98], serves as the basis or inspiration for many of these considered instances. Similarly, the knapsack problems (KP), Bin-Packing Problems (BPP) and Scheduling Problems (including Job-Shop, Fleet Management and Workforce Management) have all seen extended to consider DMO formulations. Whilst continuous benchmarks are often designed around the incorporation of characteristics into an abstract mathematical framework, combinatorial problems have evolved from and are grounded in the direct applicability to a recognisable environment.

However, there are two key limitations of all of these examples: the realistic nature of the dynamics they propose to incorporate and the reproducibility of test instances. For example, DMO-TSP's draw backs are in the multi-objective component and if simulating traffic, there seems to be no consensus method of replicating specific dynamic instances.

The following sections detail more examples of combinatorial benchmark problem types and whether significant works towards DMO formulations have been proposed.

Navigation Benchmarks: Vehicle Routing, Arc Routing

Optimisation of navigation-based problems is routed in the utility of the solutions from the perspective of time, money, fuel and efficiency. The construction of test functions from real scenarios is commonplace, including vehicle routing [98], circuit board manufacture [99] and more recent for aerospace applications [100]–[102]. There are numerous further applications, especially in mobile and network communications scenarios or dynamic community detection problems [103], where agents or nodes may be non-static.

It proceeds that in order to be realistic, formulations should accurately consider the dynamics present in some of these systems. Many examples for single-objective optimisation have been proposed, include the Dynamic Vehicle Routing Problems of Psaraftis [104] and the Dynamic Capacitated Arc Routing Problems examined by Handa et al. [105] and Tagamouti et al. [106] . Within many of these problems the dynamics are formulated around modelling the traffic, services or demands within these systems by altering dynamically, the weights within the distance matrix. This is intuitive and an effective formulation of dynamics that results in challenging instances [107]. The Dynamic Pickup and Delivery Problem (DPDP) is another example of a combinatorial problem that has grounding in recognisable scenarios such as grocery delivery and delivery from online retailers. Examples are mostly limited to a single objective function and dynamics are delivery window slots or traffic [108]–[110]. Comprehensive surveys exist for vehicle routing problems [111], [112] and real world formulations have also been considered for other problem classes [110], [113], [114].

Dynamic TSP formulations offer little in the way of justifications beyond these types of simple connection weight changes, an early example of which was proposed for single objective TSP instances [115]. A more developed formulation was proposed in the CHN144 benchmark problems by [96], [116] which describe a concatenation of a regular tsp problem with multiple geostationary and earth-orbiting satellites. The range of dynamic aspects of the instances is simplistic and limited, however it provides an example of situations where the dynamics of a system may be a relatively minor component of the problem.

Several DMO-TSP definitions exist in the literature, however there are significant limitations to their realism and the justifications in the dynamics or the objective functions beyond the first. These examples are covered in more detail within Chapter 4 where the above limitations motivate the development of the Dynamic Travelling Thief Problem.

Logistics & Selection Combinatorial Problems

Dynamic combinatorial optimization problems can also be used to model a variety of logistics and management scenarios, including for fleets of vehicles [117], [118], hospital resources [119] and military mission planning [120], [121]. The aforementioned navigation and routing problems, particularly instances of the DPDP can be considered in this category with the introduction of multiple vehicles [109], [122]. Whilst these classes of problems are not directly addressed in the contributions of this thesis, the principles behind their study is relevant here; the examination of a realistically formulated dynamic scenario and attempts to provide good solutions for them.

Similarly, selection-based combinatorial problems have a variety of methods and instances that have been developed to model specific scenarios in real systems. Many of these are not simultaneously dynamic and multi-objective with recent works, for example, still addressing the single-objective dynamic knapsack problem (DKP) [123]. Early DKP definitions can be found in [124] whilst an example of the (static) multi-objective knapsack problem can be found in [125]. There remain a few examples of DMO combinatorial selection problems; Lafetá and Oliveira [126] and Roostapour et al. [127] both provide examples of evolutionary algorithms applied to the dynamic multi-objective Knapsack Problem, where the competing objectives are the minimization of selected item weights and the maximization of their cumulative profits. The manufacturing resources allocation problem formulated by Perry & Hartman [128], [129] uses a system with multiple knapsacks.

Stacking operations and warehouse management are also examples of combinatorial dynamic optimization problems [130], [131] where the actions of cranes or other machinery, the location and organization of items and the schedule of processes can be optimized.

Recently a benchmark framework was proposed for these problems [1] and competitions have been held using these types of problems [2]. Rule-based and policy systems have also been proposed for dynamic cargo container stacking and organisation problems [132], [133].

A single objective example with the potential for multi-objective formulation, Epstein & Levy [134] consider an extension of the dynamic bin packing problem [135] to include multiple dimensions (the area or volume of the items must be considered). Another more abstracted example of dynamic multi-dimensional resource consumption is given in [136]. These examples therefore integrate a component from the stacking problems to improve the realistic features of the optimization task. The Travelling Thief Problem (TTP) [137] more explicitly joins together a knapsack and TSP problem components and the novel dynamic formulations of this are presented in Chapter 4. The Packing While Travelling problem devised by Polyakovskiy et al. [138] is a static example of the TTP, where the tour is fixed and the knapsack component of the problem must be solved.

All of these examples are very clearly formulated around their relevant application domain which therefore limits the applicability of optimization algorithms designed to solve these specific problems. The limited generalization into a flexible-yet-realistic benchmark framework that simultaneously considers dynamic and multi-objective problem aspects is current limitation of the combinatorial optimization literature.

Therefore an opportunity for a more realistic benchmark problem or framework that is more closely related to a real world scenario is required in the multi-objective combinatorial domain.

Motivations from real world systems

Generally, the examples above demonstrate that both combinatorial and continuous problems are formulated to capture some aspect of real world systems that requires optimization. The end-goal, whether explicitly stated or not, for much of algorithm design literature is the deployment on a real world system or the application to problems beyond benchmarks. Table 2.4 below highlights some works that have addressed real world problems in different domains, together with a brief description of the dynamic components.

Potential Future Applications for DMO

There are additional examples spanning other domains, and summaries of these can be found in [41] and [37]. Each of these sectors of industrial application have additional problem scenarios that can be adapted from existing works on static version of the problems. For example, the work of Mytilinou & Kolios [150] investigates EA approaches

[1]Dynamic Stacking Benchmarks: https://github.com/dynstack/dynstack

[2]Dynamic Stacking Competition: https://www.spotseven.de/gecco-2021-competition-on-dynamic-stacking-optimization-in-uncertain-environments/

Table 2.4: Real world problems with dynamic optimization formulations; many are DMO formulations.

Domain	Problem	Objectives	Dynamics	
Navigation	Public Transport	(2) Min. passenger waiting time, min. penalties due to control actions	Real-time optimization where bus arrivals trigger transitions between problem states.	[139]
Navigation	Salting Routes (DCARP)	(1) Min. total distance	Demand (salt amount required according to real temperature predictions) on costs changes	[140]
Navigation	Routing with Traffic (Shenzhen Data - DPDP)	(3) Min. workload, min. tour length, min. response time	Some requests in the network are dynamically changing.	[110]
Navigation	Routing with Traffic (Tokyo Data)	(3) Min. travel time, min. tour length, min. of road penalties	Traffic on roads updated by sensors changes travel times of specific edges.	[113]
Energy	Combustion Optimization	(2) Min. NO_x emissions, min. Boiler efficiency	Dynamic states in the combustion process, e.g. boiler temperature	[60]
Energy	Dynamic Energy Pricing	(2) Max. profits, min. costs due to cable overheating	Unit pricing for energy is dynamic	[2]
Energy	Hydrothermal Power Dispactch	(2) Min. cost, min. emission	Changes of in energy demand over a fixed schedule	[141]
Energy	Gas Turbine Operation	(2) Max. output power, min. pollutant emissions	Goal is robustness under perturbation: changes in external temperature/pressure affecting objective functions calculation.	[142]
Energy	Allocation of Renewable Energy (Stochastic Knapsack)	(1) Max. usage of available resources, constrained by energy consumption demand of users (values and weights resp.)	Energy availability and harvesting are split into intervals and vary across them.	[143]
Energy	Design of Renewable Energy Systems	(3) Min. net present cost, minimization of CO_2 emissions, min. loss of load probability	Output of the connected renewable components varies over time.	[144]
Energy	Power Dispatch with Interactive Wind Contribution	(2): Min. operational cost, min. system power loss	Varying wind power and load forecasts impact solution fitness.	[145]
Manufact.	Steel Manufacturing	(3) Min. cost, minimization of continuity violations, min. cost from flexibility deviations.	Tasks for scheduling can have dynamic duration and the arrival time of tasks is also subject to change.	[146]
Chemical	Water Pollution	(2) Min. pH deviaton, min. dissolved oxygen deviation	pH, dissolved oxygen, permanganate demand and ammonia nitrogen levels change over monthly readings.	[147]
Chemical	Greenhouse Control	(3) Max. crop profit, minimize heat cost, minimize CO_{2} cost	Real time is used, changing environments every 14.4 minutes; varying external temperature and CO_2 density, solar intensity.	[148]
Other	Parameters for Digital Watermarking	(1) Min. computational burden from re-optimization	Sequence of images represent the passage of time, the parameters corresponding to watermarking depend on the document.	[149]

to a multi-objective wind farm location optimization, and could be extended to consider mobile platforms, blade orientation, dynamic elevation or a calculation of fitness using dynamic wind speed profiles. Wind farm optimization is a popular topic [151]–[153] and some preliminary works have considered dynamic aspects within these systems [154]–[156]. Protein classification has been optimized using a dynamically adapting EA [157], however the impacts of temperature and pH in a dynamic instance could be modelled. The vegetable crop yield example presented by Zeng et al. [158], could be developed into testable instances and emulsion (Fonteix et al. [159]) and styrene polymerization [160] problems existing in the literature could be modelled in the dynamic case as the composition of the substrate mixture changes during the reaction.

Space navigation is an emerging field for DMO; a recent competition at GECCO2022 involved three different optimization problems [161], including close similarity to the Travelling Thief Problem and future problem could consider the dynamics of these systems that the competitions are proposed in. The acquisition of rare and valuable materials from asteroids [102], [161] provides a rich environment for problem formulations in which algorithms must be capable of handling the complexity of the systems dynamic aspects. Some recent works have looked at space debris clearance [100], [101] and interplanetary trajectory optimization [162] as examples of problems where EC can generate useful solutions.

The investigation of these diverse topics and the innovative application to new domains relies on the three key motivators linked to the contributions presented in the following chapters. First of these is the establishment of consistent DMO practices and an improved understanding of dynamics parameter impacts, together with tools to determine baseline expectable performance and facilitate reproducibility of results. Secondly, the bridging of simplistic mathematical benchmarks and the complex, end-goal deployment scenarios through the use of more realistic DMO benchmarks (particularly for combinatorial DMO) with justifiable dynamic changes. Thirdly, the investigation of novel properties of more realistic problems within DMO, i.e. multiple independently changing dynamic components; achieving a fundamental understanding of the challenges this presents, the suitability of existing approaches and the developments required for effective experimentation using this novel problem class.

2.5 Dynamic Multi-Objective Optimization Algorithms

A significant proportion of the DMO literature is dedicated to determining appropriate methods for solving both benchmarks and real world problems. There are several recently proposed taxonomies of such methods, each with valid groupings and hierarchies based on response type and including detection mechanisms and problem types [37], [39]. Comprehensive reviews exist on the topic, however due to the rapid turnover of new publications in the area, these are never perfectly up to date. For example, the survey of Cruz et al. [163] covers research until 2011, Nguyen et al. [38] provides a good summary of literature up to 2012, then updated in Helbig & Englebrecht [164] in 2014 and again by Azzouz et al.'s [43] survey up to 2016. Very recent surveys exist for dynamic optimisation [39] (mostly in the context of single-objective problems) and for dynamic multi-objective optimization [37] and swarm intelligence methods for DMO [165]. Whilst these works comprise an extraordinary resource in summarising recent contributions to DMO, critical evaluation is often not the focus of these surveys. The remainder of this section will provide examples of algorithms in different key 'families' of DMO responses whilst offering a brief note on the strengths and weaknesses and the relevance to experimental protocol design, instance selection and the conditioning of conclusions.

Core Principles: Basic Type Breakdown

The general classes of algorithmic responses can be grouped in any number of ways based on subjectivity or on grouping by mechanism components or outcomes. criteria. Therefore, the main classes presented here are: Diversity-based and diversity manipulation mechanisms; History-based and Fixed-information methods; Hybrid, Adaptive and Transfer-based Algorithms; Other EC and related methods.

Diversity, Multi-population, Co-evolutionary Algorithms for DMOPs

Diversity is the cornerstone of many evolutionary techniques as it allows the exploration of the search space, prevents premature convergence and can be used to avoid local optima. Diversity is also employed as a reactionary mechanism for dynamic changes or codified within the algorithm structure to ensure it is maintained.

Diversity introduction for population based algorithms is one of the simplest response mechanisms and has been used since dynamic optimization problems were first formulated. Cobb proposed using hypermutation to improve performance in dynamically changing problems. Hypermutation as an adaptive operator increases in the probability of mutation events when producing offspring solutions and is triggered by poor algorithm performance (such as immediately after a change event). A wealth of strategies have been proposed since, the most popular of these involve 'immigrant' solutions. Extra solutions are added to the pool of solutions in response to dynamic changes in order to boost the population diversity. These solutions, if randomly generated are sometimes referred to as 'random immigrants' [21], although solutions can be generated via mutation or other recombination mechanisms. Deb et al. [141] extended the well known NSGA-II algorithm to handle dynamic instances using diversity introduction techniques. An *A* variant, adding a parameter-controlled percentage of random solutions and a *B* variant that provides mutated solutions are proposed. Both variants are widely used as baseline comparison algorithms for novel methods or similar methods have been devised [166]. Additionally random re-initialization is a common baseline comparison response strategy and is sometimes also called ‘random restart’ [29], [53], [63], [72], [85], [167]. It represents the most severe and destructive form of diversity introduction, where the entire population is discarded and a new set of solutions is randomly initialized. Generated solutions can also be guided by the centroid of previous solutions [145], mutating previously found good solutions [168] or by using local search to improve the quality of solutions [169].

Maintaining diversity can also mitigate the impact of the changes. For example, Bui et al. suggested defining an additional objective to be optimized that focuses on the diversity [170]. Fitness for this objective is calculated in a number of ways, including using the age of solutions or calculated by distance to the closest neighbour or all other solutions. Similarly to hypermutation discussed previously, dynamic adjustment of mutation, crossover and objective weighting parameters has been used to improve diversity and promote exploration [121]. Thermodynamic Genetic Algorithms (TDGAs) are also proposed [171], [172] that use concepts of temperature and entropy as in Simulated Annealing [173] to control mutation explicitly. Forcible partitioning of solutions within the decision or objective space is another way to guarantee diversity is maintained within the optimization run. The method proposed by Zeng et al. [158] maintains multiple clusters of solutions that contribute to generating a new population after each dynamic change. A similar clustering-based method has been applied in a real world DMO hospital resource management problem scenario [119].

Multi-population methods are a more strict way for separating groups of solutions or partitioning the search space. Branke et al. [174] and Oppacher [175] provided early examples of this, where a core population is used in addition to smaller, ‘colony’ populations that are forced to explore distinct regions of the search space. Das et al. [176] merges diversity maintenance with multiple populations by using additional concepts such as adaptive quantum individuals and solutions with trajectories based on Brownian motion within each sub-population. Multi-population methods for DMO have recently been applied to Particle Swarm Optimization (PSO) methods where a separate swarm population is optimized for each objective of the problem [177] or on decomposed components of the problem [178].

Parallel processing techniques, such as those in Camara et al. [179], can also be used for DMOPs by using both data decomposition (sequential solving of problem states) and functional decomposition (separation of tasks to be run concurrently). The principles and motivations of parallel algorithms are similar to those of multi-population techniques and mechanisms such as migration can be employed in both. The speed of computation can also be improved by using parallel methods [73], [180] and when deploying these methods using high performance computing [181]. Parallel methods have also been applied to DMO instances of TSP problems [182], where a 16-computer architecture is used to explore the locally optimal solutions in a discretized objective space.

Co-evolutionary algorithms provide yet further development of diversity introduction, maintenance and the distributed exploration of the search space. A number of paradigms are extended from existing methods for static problems, including cooperative co-evolution [183]–[186] and competitive-cooperative co-evolution [76], [187]. Cooperative co-evolutionary algorithms used information sharing (parts of, or full solutions are shared between a number of discrete populations) to drive convergence but also maintain and introduce diversity. The recently proposed algorithm by Xie et al. [188] explicitly balances the two goals of convergence and diversity within a cooperative co-evolutionary framework.

Whilst these different methods vary in complexity and mechanism, there are some unifying strengths and weaknesses that they share. For example, each of a range of showcased algorithms has its own mechanism of incorporating diversity as a key aspect of the search for good solutions to a problem, each of which is effective in different circumstances. Diversity is also key to handling a number of key problem features that can confound more simplistic solvers (e.g. basic hill climbing strategies), including problems with deceptive POS/POFs or those with disconnected or isolated regions. However, these types of landscapes can also pose significant challenges to diversity focused methods, the extreme cases, for which other response mechanisms may struggle too, (e.g. highly multi-modal, many local optima or highly disconnected problems) may hinder the ability of these diversity-focused methods to provide good solutions. Other challenging aspects may include problems with low frequency changes (rapid changes, short time between intervals) as the introduction and maintenance of diversity after change events is followed by the utilization of this solution information to drive the algorithm towards good solutions in the new problem state. If there is insufficient time for this exploitation, the algorithms may again, struggle to provide good solutions. Finally, each of the aforementioned classes of diversity-focused methods relies on the selection or setting of particular parameters; the proportion of the population to be replaced with random immigrants; the initial weighting of a diversity objective, the number of populations and their size; the communication frequency between isolated population of solutions; or the maximum mutation rate and the 'cooling period' for hypermutation responses. Whilst (almost) every optimization is dependent on parameters for its correct function, these methods are reliant on an appropriate setting or design based on the problem application or the characteristics of the dynamics in the examined suite of benchmarks. For example, a dynamic instance with very low severity changes may be tackled with a 'smaller' response (e.g. fewer random immigrant solutions). Adaptive methods and hybrid mechanism are described later in this section to overcome some of these limitations.

Memory, Prediction & Preference Algorithms for DMOPs

Memory and the exploitation of historical information is an intuitive approach to dynamic, non-stationary and time-varying problems. There are a variety of ways that this information has been exploited as an algorithmic response to dynamic change events.

Implicit memory methods incorporate redundant code through diploid genomes. This means that for each decision variable, there is a secondary 'allele' which contains a different value. The visibility of each allele is determined by a dominance table which can change adaptively depending on the detected changes. There are many examples of dilpoidy being employed for dynamic problems with a single objective [11], [189]–[192], however fewer examples exist for multi-objective problems [159], [160]. The concept has also been extended to consider multiploidy for dynamic problems [189], [193], [194]. Nguyen and Yao [38] split memory-based methods into implicit and explicit memory techniques. Further categorization into direct and associative separates them based on the type of information stored. Direct memory methods store previously found good solutions, either in a dedicated archive or separately from the optimizing population and there are examples of this mechanism in the DMO literature [23], [195]–[197]. Less commonly employed, associative memory techniques store, for example, information about the previous problem states or about the nature of the changes [69], [192].

Explicit memory is any method which employs an archive of solutions which can be re-inserted into the population, or used in other ways in response to changes [23]. The maintenance of an archive of previous best solutions is commonplace across DMO algorithms. The training of generalized models or distributions using historical solution information and recall from these can also be considered as explicit memory [149]. Short term prediction methods, that use information from the previous environment also exist. Wu et al. [198] formulates a re-initialization strategy to boost diversity that combines predicted shift of the POS using the centroid of solutions in the previous environment, with a local search strategy.

Prediction strategies exploit the problem history for different response mechanisms given the serial nature of successive states in dynamic instances. Whilst most prediction methods focus on forecasting information about the next problem state, the changes can also be anticipated [199]. Forecasting methods are much more prevalent in the literature and a variety of temporal model types have been constructed including, Auto-Regressive Moving Average (ARMA) [77], [79], Kalman-Filter [53], [200], Polynomial Regression [201], Fractional Order Displacement [202]. The predicted information is used in a variety of ways, most commonly to guide partial or complete reintialization of the population around a predicted decision space centroid or region of the search space [40], [77]–[79], [84], [203]–[206]; Biswas et al. [80] proposed a 'controlled extrapolation' re-initialization method using a model similar to [79]. Many recent methods proposed a composition of multiple models or a partitioning of the decision or objective space for prediction [82], [207]–[210]. Very recent innovations have extrapolated additional information from the optimization to inform prediction; Li et al. [211] anticipates the trajectory of solutions, whilst Zhao et al. [212] extracts knowledge from the search space in a method inspired by signal restoration. Public transport [139] and combustion outputs [60] are examples of real world problems that prediction strategies have been employed for.

Memory strategies have been shown to be useful when the problem states in the dynamic instance are revisited [23] and the benefits are intuitive. Assuming the automatic detection of changes, this makes memory-based mechanisms useful when the dynamics occur in a cyclic or periodic nature. The same problem characteristics make prediction methods particularly useful as the utility of the constructed models is stronger if the pattern of changes is clearer. However, the limitations for both memory and prediction-based responses present when the dynamics are stochastic, non-cyclical or the sequence of changes is irregular or unpredictable. In the early stages of the optimization process, both types of strategies cannot provide much useful information. Memory-based methods provide good solutions of information from previously visited problem states, if a new state presents, there may be no relevant information that the response can provide. Prediction strategies rely on the construction of the model and the use of training data in order to provide a useful prediction, this cannot occur if sufficient historical data has not be accumulated. To counter these limitations, memory and prediction methods are often combined with diversity-based methods to form hybrid strategies.

Preference incorporation is a growing topic in EC generally, but has been applied to DMO in the form of reference point guidance [94] and hyper-heuristics with plane separation [213]. Both of these are established methods in preference-based and interactive optimization, however a major limitation lies in the informed selection of reference points and the regions of interest, particularly in the different dynamic intervals of DMOPs.

Hybrid, Ensemble, Adaptive, & Transfer Learning Algorithms for DMOPs

Hybrid dynamic response strategies are methods that combine multiple response mechanisms. Generally, many of the hybrid methods are more recent, as the combination of established and refined response mechanisms to mitigate their individual limitations becomes possible. For example, prediction methods are combined with memory based methods [214], [215] to further exploit cyclical dynamics, or multiple prediction models are combined [216], [217] to boost prediction performance. Prediction-based strategies are also commonly combined with diversity introduction mechanisms [83], [218]–[220] and diversity maintenance schemes [221] in order to reduce the impacts of prediction error after changes. This can also benefit the early stages of the optimization where the prediction model is still being constructed. Yong et al. [222] have recently suggested a transfer learning-based scheme to avoid the early-optimization issues of a prediction method. Prediction strategies are also used in novel ways, such as estimating the problem state for the switching of optimization tasks with a hybrid algorithm from tracking optima to finding robust solutions [71]. Memory mechanism are a popular hybridisation candidate, having been combined with diversity introduction [113], local search mechanism [54] and adaptive operators [55]. An ensemble of mechanisms is proposed for adaptive population management using memory, local search and diversity introduction methods [223] that can be adapted based on the change severity [53]. More innovative combinations compound several types of local search mechanism [110] or local and global search operators [224] and even in the composition of immune response and co-evolutionary mechanisms [225]. The range of possible hybrid strategies is limited only by the creativity of the designer and the utility they have across the DMOP space. The combination of mechanisms is an alternative to adaptive methods for alleviating the limitations of mechanism that present when used on their own. Because of this, it appears recent publications rarely use strictly a single class of response as their dynamic response mechanism. Many of the previously mentioned methods within the other categories employ aspects of other mechanism to find a balance between convergence and diversity. For example the cooperative co-evolutionary mechanism by Xie et al. [188] uses a hybrid mechanism of prediction and diversity introduction and maintenance in response to quickly react to change events.

The strengths and weaknesses of hybrid strategies are dependent on the combination of response mechanisms used, however generally hybridisation is effective for mitigating many of the limitations of individual mechanisms. Only the combination of computationally expensive mechanism may slow the overall optimization process.

Adaptive mechanisms alter the strength of response or some aspect of the optimization in order to improve the performance based on a stimulus. For example, the strength of response can be based on the severity of change [136], [146] and can be based on the adaptation of genetic operators (and their parameters) [146], [226]; the priority of objective functions [121]; or in the proportion of immigrant solutions generated randomly [136], [227], [228] or by mutation [64]. Alternatively, the number or size of sub-populations can be adaptively altered in response to dynamic changes [229], [230] or the probability of using different response strategies in an ensemble is adaptively updated based on previous performance [231], [232]. Adapting aspects of the response mechanisms increases the flexibility and applicability to unseen or unknown dynamic instances, however the tuning of the information that feeds the adaptive process requires careful design. For example, the process of estimating the severity of change must be accurate, otherwise the enacted response will be less relevant to the new problem state.

Transfer learning has also been used as a mechanism for solving DMOPs. The principle of this mechanism is to build a model based on previous optimization runs that assists the optimization of future runs. Jiang et al. construct a latent space [95] and manifold memory [233] to improve performance. Li et al. [234] uses a clustering-based model for application to planning for unmanned vehicles. A limitation of transfer learning methods is the complexity of the constructed model may slow the overall optimization process. Recent work by Chen et al. [235] attempts to simplify transfer models to speed up computation. Despite this, transfer learning is a powerful tool that can increase the generalizable capability of developed algorithms for DMO.

Other EC and Related Methods for DMOPs

Other evolutionary computation and related methods have been applied to dynamic and DMO problems and there are recent and comprehensive surveys documenting some of them. Some swarm and multi-swarm based methods (e.g. Particle Swarm Optimization) have been included in the previous sections due to their relevant response mechanisms, however there are many further examples [40], [60], [75], [165], [196], [236]–[241] . Ant Colony Optimization (ACO) is also a popular choice for both dynamic scheduling and routing problems among others [56], [242]–[247]

Fewer works have also developed Artificial Immune Systems (AIS) or clonal algorithms to handle dynamic problems [115], [147], [248]–[252]. A subset of the literature considers innovative approaches to handling dynamic problems using methods such as Membrane Computing [33], Deep Reinforcement Learning [59], A-Life models [253] or, recently considering additional biological mechanisms such as epigenetic blocking [254].

2.6 Performance Measurement in Dynamic Multi-Objective Optimization

As with benchmark problems and algorithm mechanisms, the performance measurements employed within DMO have been subject to survey. Helbig & Engelbrecht [255] provides a summary of many performance measurement types and gives their strengths, weakness and compatibility. Sections in several other reviews provide summaries of measurements used throughout the dynamic optimization domain [38], [43], [163], [256], [257] and the challenges [43], [255]–[257] of reporting performance in DMO.

Adapted vs. Purpose-Built Measurements for DMO

Many of the measurements used in DMO have been extended from static optimization and multi-objective problems, whilst some have been designed to specifically capture the algorithmic behaviours over time. Comprehensive surveys have been carried out for dynamic optimization [180] and for DMO [255]. The simplest forms of performance measurement are time-averaged averaged versions of static measurements; commonly used measurements such as Hypervolume, Generational Distance [74], [79], Inverted Generational Distance, Maximum Spread and Set Coverage [85] have all been adapted in this way by dividing the sum of the performance measurements taken in each iteration (generation) and dividing by the number of generations. Again, Helbig and Engelbrecht [255] provide a comprehensive summary of these adaptations and their usage.

This practice of adapting static measurements for DMO has multiple drawbacks for the effective reporting of algorithm performance. Firstly, the condensing of the dynamic performance to a single value to represent achievement on a particular instance results in an unavoidable loss of information. The temporal behaviour is key to interpreting algorithm performance and can not be effectively captured in a simple average-over-time measurement. For example, in a typical problem instance consisting of 10 dynamic intervals, one algorithm may achieve middling performance across all intervals, whilst another algorithm may achieve a similar performance score from poor performance in the first five intervals and good performance in the final five intervals. Whilst succinct, time-averaged measurements provide little insights into the dynamic behaviour of algorithms when precaution is not taken in reporting.

Several performance measurements specifically designed for dynamic optimization attempt to more accurately represent algorithm performance from a robustness-to-changes perspective. The *accuracy*, *stability*, and *reactivity* [179] metrics are designed to capture specific aspects of algorithm performance on DMO instances.

The *accuracy* is a general measurement of performance at a given generation, measured as the difference between the maximum possible hypervolume for the Pareto Front and the maximum hypervolume achieved so far. The *stability* metric provides a measurement of the impact of the change on the performance (using accuracy) relative to the previous iteration; *reactivity* measures the recovery rate of an algorithm after a change event has occurred (again derived from an accuracy measurement).

Hypervolume is one of the most commonly taken measurements across static and dynamic multi-objective optimization as it provides a summary of information provided separately in extent, spread and density measurements (this summarization can also be a limitation to its informativeness). Adaptations in the form of Hypervolume Difference (HVD) and Hypervolume Ratio (HVR) are employed in DMO (e.g. [79], [141], [180] & [87], [180], [197], [258], resp.) however like the aforementioned measurements, these are often presented as a single value of performance, condensing varying algorithm performance into an uninformative number.

Limitations & Challenges: Known vs. Unknown POFs

Calculation of many measurements for DMO rely on knowledge of the POF, for example to place the reference points for GD and IGD measurements. For benchmarks, this knowledge comes with it's definition but in terms of complex, real world and many combinatorial problems, the true POF is not always known for all or some of the dynamic intervals. Again, the hypervolume prevails as POF-free measurement, however the selection of the nadir point (reference vector) requires careful selection and clear reporting, otherwise results are incomparable. Cámara et al. [73] proposed a method for approximating an unknown POF in order to calculate a hypervolume to be used in the calculation of *accuracy* measurements. Similarly, Helbig & Engelbrecht specifically detail the challenges that present when the POF is not known [255].

Whilst the methods of reporting performance are intended to showcase an algorithm's performance in the context of similar or distinct algorithms, several considerations can be made when taking measurements to ensure purposeful reporting.

If the POF changes location or geometry in the objective space with dynamic changes then the hypervolume is not comparable between intervals, meaning an average should not be taken over them, unless it is abstracted to an HVD or HVR value.

Reporting performance separately in each interval of dynamic instances has been used infrequently as it requires more space to present tables of results and makes succinct statements on performance more difficult.

End-of-interval measurements provide useful information from minimal data. Recording the hypervolume (or other measurement) in the generation immediately before a change (after detection if necessary) gives a reading of an (elitist) algorithm's best achievement for the entire time it was optimizing the problem state in the given dynamic interval. The alternative is to take measurements for every generation, which can include the relatively poor performance of the recovery period after each change.

Reporting performance over multiple patterns of changes is challenging as these instances are not easily comparable. Abstraction to HVD or HVR measurements before averaging provides the simplest method for allowing comparability, however these require that the POF is known which is often not the case for many combinatorial problems where this issue is most prevalent.

An alternative to absolute measurement reporting, comparative results can also be presented for algorithm performance. This type of performance reporting involves (usually) pair-wise comparisons between methods to establish the difference in their achieved performance measurements. This naturally allows for informed application of statistical testing using samples of measurements throughout the dynamic intervals, to for example, provide a measurement of the proportion of statistically significant end-of-interval outperformance over another method. It is important to select carefully the comparison methods to ensure a relevant and applicable baseline is established, for example, by using a well-known algorithm from static problems such as the NSGA-II [259]. These principles also informed the approach adopted in the dynamic multi-objective combinatorial instances of the Travelling Thief Problem addressed in Chapter 4, since this problem has an unknown POF.

Selection of Algorithms for Performance Comparison

Another important topic in the reporting of algorithm performance is the set of comparison methods selected. These provide the context for the performance of any novel method, however the inconsistency of selected baseline algorithms can hinder the relevance and applicability of presented results. It is intuitive to compose results from a set of algorithms believed to be the 'state-of-the-art', a classification that is subject to change, to provide context to the results of a novel algorithm. However, the selection of comparison algorithms is unconstrained by any particular established practice. This combined with the incoherence in approach to experimental parameters for DMOPs, leads to a generalization of performance and a fuzzy acceptance of what constitutes the current 'state-of-the-art'. Whilst no obvious solution presents itself for this difficulty, addressing the scope of experimental parameters and establishing a baseline of expectable performance across the dynamic instance space can provide some structure to the presentation of results going forward. This topic, together with the previous comments on establishing and reporting baseline performance are developed within Chapter 3.

Performance Visualization: Approaches and Limitations

Presentation of averaged measurements or tables of statistical testing outcomes, measurement profiles throughout the optimization and obtained Pareto Fronts are exmaples of commonly employed methods for visualizing performance in DMO. There are benefits and limitations to each of these which are only briefly touched on here; development and improvement of visualization techniques is an ongoing field of research across all aspects of evolutionary computation.

Single values for performance presented in tables is very well established practice for the succinct and efficient documentation of achievement and direct comparison of performance. Generally, good scientific practice requires that these values are the average or aggregate of a number of repeats, meaning that a sample of measurements exists and therefore statistical tests are often applied. The efficiency of this mode of presentation is desirable, however, as mentioned previously, a large amount of information is lost that could provide meaningful insight.

Profiles of measurements, meaning the average of the measurement recorded in each generation of the optimization procedure can highlight aspects of algorithm performance that cannot be communicated in a single value result. Aside from the previously mentioned considerations surrounding incomparability of dynamic intervals, the behaviour of the algorithm may be different throughout the optimization such that insights about specific problem states that are more or less challenging can be obtained from this kind of visualization (an illustrative example of this is shown in Figure 2.3). Moreover, comparison between methods using measurement profiles can provide additional information in cases

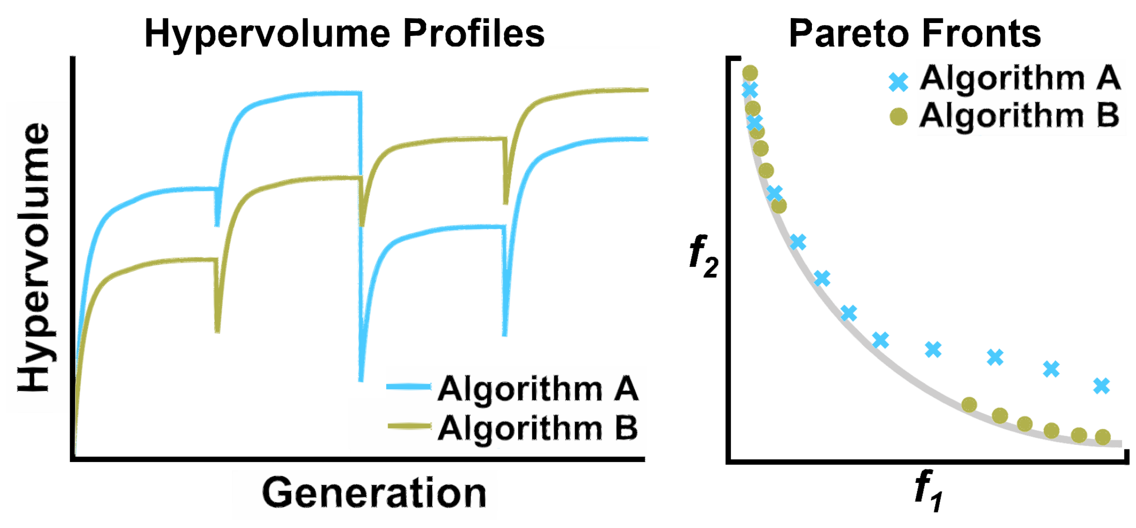


Figure 2.3: Illustrative diagrams showing additional observations that made through (left) comparison of measurement profiles and (right) comparing approximated Pareto Fronts.

where different methods succeed in different intervals; a single value cannot provide this. These comparisons and the interpretation of performance in these cases generally qualitative unless further procedures are taken to distill this information into a format that can be used to give concrete conclusions. The approach to comparing performance in Chapter 4 for the Dynamic Travelling Thief Problem instances addresses this transformation in a measurement that respects the proportion of generations with significant improvement.

Pareto Fronts obtained by algorithms are a coarse grained and mostly illustrative method for visualising performance. The benefits of plotting obtained Pareto Fronts provides an accessible interface to compare aspects of algorithm performance that may not be captured using measurement profiles or single values for performance. For example, there may be regions of the objective space where a low density of solutions is achieved, despite a good or similar hypervolume (or other) measurement to another algorithm. An illustrative example of this is shown in Figure 2.3 Visualization of Pareto Fronts however can become difficult and space-intensive in instances where there are many dynamic intervals, the geometry of PF changes is complex or there are many algorithms to compare. Problems with more than 3 objectives also become difficult to illustrate.

Some novel methods bridge these levels of abstraction, for example the Empirical Attainment Function (EAF) proposed by [260], [261] and utilized for DMO by [103] builds a probabilistic model of attainment across the Pareto Front. The challenges here are similar to presenting the Pareto Fronts themselves; many changes and complex geometries make presenting EAF results difficult.

Each of these methods provides different and useful information that is not captured within the others. Whilst not always feasible a combination of performance visualization can strengthen conclusions through additional insights, often the problems or motivations for the experiments define the level of abstraction or explicitness required in the results. Generally the importance of preserving temporal information and accurate reporting of performance is the primary goal in DMO, where additional credence should be given to comparability and reproducibility.

Additionally, important aspects of DMO experiments go beyond just choosing the correct visualization method for the results. The selection of appropriate problem suites based on experimental goals; the informed experimentation across the possible dynamic instance space for these problems; the comparison with relevant baseline algorithms and robust experimental methodology are all key requirements of generating meaningful contributions in DMO.

Accessibility, Reproducibility, & Explainability in DMO

The motivations for the presented work follow key themes of reproducibility, accessibility and explainability. These qualities are essential for the maintenance of good research practice and the communication of results and therefore progress in any particular field. Accessibility in this instance refers to both the clarity of presented motivations, including descriptions of methodology that avoid vagueness and any omission of crucial details. Ensuring clarity makes the arguments for the work easily accessible to the audience. It is important to note that a clear description does not need to sacrifice details, complex concepts or specific terminology. Specifically for DMO, whilst the concepts of dynamics are relatively simple, the specific implementation and the associated motivations must be accessible to the reader.

A very closely linked concept is reproducibility. Ensuring the presented results can be replicated without extreme difficulty should be the standard across research. There are levels to the transparency necessary to afford this however; proprietary restrictions, continued and related research and sensitive data sets can all be obstacles to the explicit sharing of data or code. As with making academic outputs accessible, effective and clear descriptions of methods, performance measurement calculations, data manipulations and statistical procedures are required to facilitate the reproduction of results and therefore verify findings. Generally, to cohesively make meaningful progress in any field, reproducibility should be a standard requirement for published works. The ACM Digital Library has recently proposed a supplemental award system for reproducibility based on the provisions made by authors. Reproducibility is becoming a key interest in the general computer science domain, for example the Supercomputing conference (https://supercomputing.org/), operating system design conferences (EuroSys[3], OSDI[4] & ACM-SOPS[5]) and the recently founded *Transactions on Evolutionary Learning and Op-*

[3]EuroSys: https://dl.acm.org/conference/eurosys/
[4]https://dl.acm.org/conference/osdi
[5]https://dl.acm.org/conference/sosp

timization journal [262] have all adopted this system. Recently, in the context of software engineering, reproducibility is posited as a key skill to be included in undergraduate curricula [263]. For DMO, reproducibility is of particular importance as the sequence of problems states represents a specific instance of a dynamic problem. For combinatorial DMO problems, due to the number of different possible dynamic instances, the pattern of changes or the process to generate them must be carefully reported to allow for reproduction of the results. For example, [182] is an example of a combinatorial DMO problem where the process of instance creation is explained, however the exact instances cannot be recreated. The 'random' changes implemented to generate the different dynamic intervals in the problems are not explicitly reported or the seed and random number generation methods not clarified.

Explainability of algorithm behaviours and the results they produce is crucial to the effective communication of research. In the presentation of data and the accompanying analysis, there are significant challenges that DMO research provides in terms of explainability. Whilst the formulation of dynamic problems from real world scenarios is often an intuitive process, justifying the measurement protocols and explaining the process of arriving at results can be difficult. In DMO problems, tracking the best set of solutions is a common goal, however measurements associated with this set can be taken in a variety of ways. The measurements themselves distill different information about the algorithm behaviour, resulting in different levels of information loss when presenting performance as single values. Beyond numerical results, a concise and purposeful accompanying explanation is key to the relevance of the experimental outcomes and their support for any conclusions made.

Additionally, most published articles highlight some further aspect of the experiments to be investigated or an application to additional problems and extension of the algorithm response. The work [264] & [265] both provide recent and expert insights into the remaining challenges facing evolutionary dynamic multi-objective optimization going forward.

2.7 Conclusion

The presented review of the literature highlights the comprehensive research attributed to some topics within DMO and the relative scarcity of attention that other areas have received. Numerous benchmarks suites and test problems have been defined, some including directed design based on realistic dynamic characteristics. However, no cohesive approach is provided for experimental methodology and the selection of dynamic instances for robust conclusions. The extension of these realistic dynamic characteristics is also largely missing from the combinatorial domain of DMO problems. Definition of a realistic DMO benchmark that contains justifiable dynamic change events is of significant benefit to the field. Allowing the flexibility of this problem to examine a variety of scenarios that could present in real world problems in a variety of application domains expands the relevance of this contribution further. Finally, the implementation of additional characteristics that intuitively exist in real problems is a logical progression for the field of DMO. A structured, justified and informed approach to investigating novel dynamic problem classes will allow for the identification of the key challenges and obstacles and avoid the non-cohesive approach taken to DMO experimentation in the past. The work presented in the remaining chapters will address the major limitations identified within the literature. Chapter 3 focuses on the DMO understanding and the establishment of baseline practices using the frequency and severity parameters. Chapter 4 provides the definitions of the dynamic Travelling Thief Problem and the development of methods to find solutions for them. Finally, Chapter 5 presents the observations and challenges involved with the development of DMO problems to include multiple dynamically changing components.

Chapter 3

Establishing Baseline Performance and Reproducible Experimental Protocols for DMOP Benchmarks

3.1 Introduction

Dynamic Multi-objective Optimization Problems (DMOPs) often have complex objective functions, constraints or definitions regarding decision variable correlations or distributions. All of these components have a wealth of research and attention, compared with the parameters controlling the nature of the changes which are less well studied. There are a number of different parameters and settings that impact the complexity of the dynamic changes. For example, as discussed in Section 2.3, instances with periodic and cyclical environments, specified limits on the value of t, the delay in onset of changes, and those with non-fixed severity and frequency each provide additional challenges to algorithms. However, the key parameters governing dynamic change events in DMOPs; the frequency and severity of changes (sometimes referred to as the magnitude) describe the core characteristics of DMOP instances. These respectively determine how often the changes occur and the scale of the difference between successive dynamic problem environments.

Standard practice has originated from fundamental papers in DMOP research; the work of Farina et al. [28] proposed a simple method of updating the value of a variable t according to a severity, $\frac{1}{n_t}$, at intervals determined by a frequency parameter, τ_t, using iterations as units. Equation 3.1 shows this calculation. This paradigm is used almost universally across DMOP and DMOEA literature. Some authors utilize function evaluations [35], [51]–[53], [203], or real time [56]–[58] in place of generations/iterations as the frequency parameter unit. Exceptions include works where dynamic change events follow scheduled sequences of events and an update calculation is not required [59], [266]–[268].

The majority of works give little justification for the selection of frequency and parameter selection. There are a number of works that justify usage as being the same as in Farina's work [28] when using the FDA problems. Others [29], [38], [41], [257], [269], understand the importance of the frequency and severity parameters and use several frequency-severity combinations in an attempt to diversify the coverage of problem instances.

There are many real world problems scenarios that may contain dynamics that change at irregular frequencies, meaning that there is a varying amount of time or iterations between change events. Experiments in the literature focus on constant frequencies for the most part.

Some attention has been paid to problems where there is a changing or unknown severity of change, an intuitive feature of realistic scenarios. Several works exist proposing algorithms that adapt their dynamic response based on the measured severity of change [53], [61], [270]. These methods also rely on reliable detection of changes, an additional topic that has received significant attention [39], [65], [70].

Despite DMOP literature focusing on using benchmarks to determine novel algorithm performance and the majority of experimental methodologies considering constant frequencies and severity values, there remains significant gaps in understanding the impacts of these parameters.

Particularly, the inconsistency of frequency and severity parameter usage in the literature adds to the limitations of previous conclusions. Despite using the same or similar sets of benchmark problems with consistent numbers of decision variables, results become incomparable when the frequency and severity parameters are different. Therefore a clearer picture of the impacts of frequency and severity parameter combinations is required.

Furthermore, establishing a baseline of achievable 'good performance' must be done for each problem separately - conclusions based on similarity of definition/character are less concrete than explicitly determining baseline performance.

Little understanding is explicitly present in the literature for the comparative impact of frequency and severity between different problems. We can therefore illustrate via experimentation the difference in the combination space between benchmark problems.

Typically, methodologies use multiple benchmarks to demonstrate or compare algorithm performance across problems with a range of characteristics. For example, including problems with changes of different types (Farina [28] classification), PF geometries and connectivity, and objectives improves the strength of conclusions for algorithm performance. However, if we consider the limited range of frequency and severity combinations that are used within the literature, these represent a small portion of the possible instance space and therefore the relevancy of the conclusions is limited.

Testing many combinations across the frequency and severity domain enables an understanding of the limitations of an algorithm or method beyond its performance on a single or narrow range of instances.

Comprehensive examination of combinations across commonly used and recent DMOP benchmarks via the application of popular non-dynamic MOEAs provides numerous significant contributions. Firstly, it gives an insight into the comparative utility of different comparison algorithms for determining novel algorithm performance in context. Secondly, determination of the relative difficulty of different DMOP benchmarks and its relationship to frequency and severity parameter values. Thirdly, the opportunity to ascertain explicitly the utility of the commonly used 'random restart' method across the frequency-severity combination space. This response is often used as a baseline comparison, with the best use-case quoted as high severity changes where previous solutions may not be relevant to the new problem environment. Finally, these results set a precedent for reproducibility through the provision of source code and design of experimental procedures regarding selection of instance with different frequency and severity of changes.

3.2 Background

Some of the principal benchmark dynamic multi-objective optimization problems (DMOPs) were defined nearly two decades ago by Farina et al. [28]. Since then, a plethora of benchmark problems have been proposed [26], [29], [31]–[33], [41], [72], [81], [89], [271], allowing for the testing of different characteristics of real world systems in a controllable environment. Dynamic Optimization Problem (DOP) generators such as Moving Peaks [23], [26] and the Dynamic XOR [27] and others are well-known single-objective environments in which the difficulty of a problem instance can be controlled by increasing the number of peaks or bits respectively. Comprehensive surveys of methods, measurements and problems are given in [38], [41], [43], [255], [256], [272], [273].

Frequency

The frequency of change in dynamic problems is an intrinsic component of this class of problems and is represented by the symbol τ_t (sometimes τ_T). The selection of a frequency parameter value requires a unit of time for its context. In the literature, the most common units are generations or iterations, meaning for example, one full evaluation of the population in a population-based algorithm. The number of function evaluations can also be used, however in works that do so [52], [53], [203], given a population size and a complete algorithm description or pseudocode, this can be converted into generations. Many works using real world problems or simulations select their dynamic frequency based on realistic events and therefore use real time [56], [274]; seconds [275] or hours [143] or days & weeks [59].

Intuitively the schedule of events in a real world problem may be unknown, at non-fixed frequency and may be best described in real time. However, the widely adopted use of generations (and function evaluations) allows for comparability of results where a difference in computation resources/hardware would otherwise complicate relevancy.

Severity

As with frequency, the severity of changes, also sometimes called the magnitude of a change is a key component of dynamic problems. The parameter is a measurement of the difference between successive problem environments. Typically, severity is denoted by n_t, but s [276], [277] is also used. The value of n_t effectively represents the number of problem states that the dynamic problems go through upon changes; the time variable t within DMOP equations is calculated at any given time step using Equation 3.1.

$$t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \tag{3.1}$$

where n_t, τ_t and τ are the severity and frequency and current time respectively.

Weicker [278] analyses the impact of different severity values in simple single objective dynamic framework and the relationship with population size.

Parameter Selection

Various recommendations are made in the literature as to the values of n_t and τ_t that should be investigated. For example, Helbig & Engelbrecht [41] suggest values of $n_t = \{1, 10, 20\}$ and values of $\tau_t = \{5, 10, 25, 50, 100\}$ to be used in various combinations. The parameters combinations in these ranges enable the investigation of algorithms in environments with fast-changing and high-magnitude changes or in those with slower changes and smaller magnitude changes, where effective change detection may be a motivator. In contrast, Farina et al [28] when defining the fundamental FDA suite suggest a single combination of values with $n_t = 10, \tau_t = 5$ and the number of decision variables as $n = 20$.

Where authors have selected a variety of combinations for their experiments, some understanding of the importance of the frequency and severity parameters is apparent [61], [79], [94], [95]. Often these works include a statement justifying selections ‘to examine the impacts of frequency and/or severity’, only to select a handful of frequency and severity combinations to examine. A more comprehensive picture is required to make conclusions about algorithm performance on DMOPs across the frequency and severity combination space.

To illustrate the usage of frequency and severity parameters in the literature, values used in experiments were extracted from work in the reference list of a recent survey paper entitled ”*Evolutionary Dynamic Multi-Objective Optimisation: A Survey*” by authors

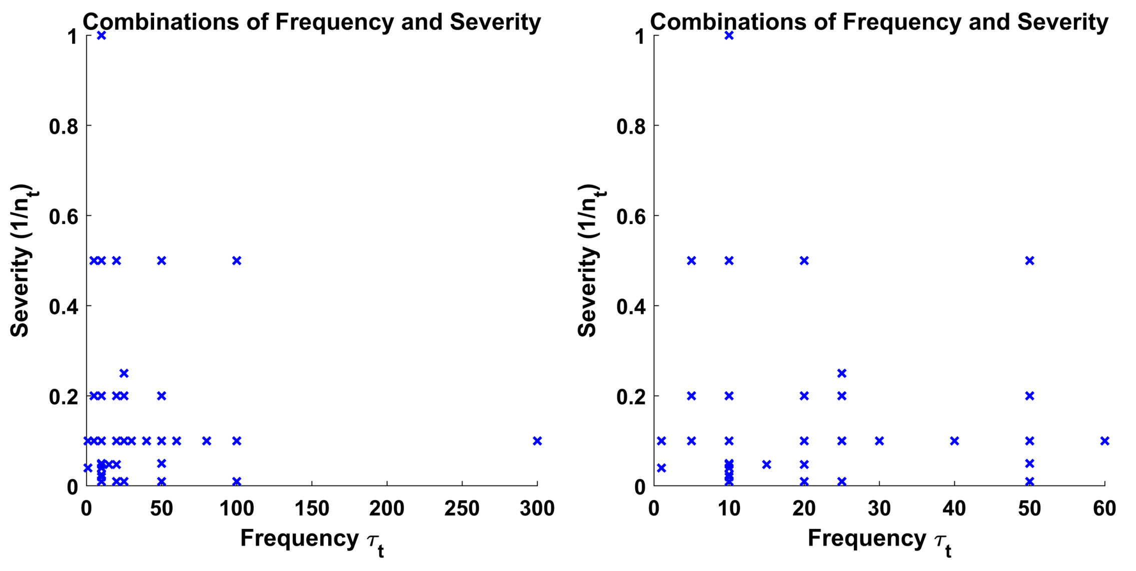


Figure 3.1: Visualisation of DMO literature usage of frequency and severity of change parameter values. (left) The spread of frequency and severity parameter combinations used in experiments in the DMOP literature. (right) Zoomed-in section of left panel to better show lower frequencies.

Jiang et al. published in *ACM Computing Surveys 2021* [37]. The results are presented in the following figures.

Figure 3.1 illustrates the combinations of frequency and severity values investigated in the DMO literature. As mentioned previously, there are works that consider real-time frequency or atypical severity [31], [141] and as such cannot be included here. Figure 3.1 allows for several interpretations to be made. Firstly, in terms of frequency values, there are many different values that have been combined with a severity $n_t = 10$. Whilst this may seem like a positive, these different combinations are spread amongst different works in the literature, with examination of multiple combinations remaining rarer. Secondly, the use of severity values is generally in $n_t \geq 5$ with much fewer instances considering changes with severity $1 < n_t < 5$. Ultimately, there is little consistency in terms of the values of frequency and severity values, and their combinations, used in DMO experiments.

Notably, some values of severity and frequency are used more often than others, however the prevalence of different parameter values is best illustrated in Figure 3.2.

A severity of $n_t = 10$ and a frequency of $\tau_t = 10$ are the most commonly used parameter values. Other severity parameter values have a low usage within the literature and some values e.g. $n_t = 21$ are used rarely [279]. A wider variety of frequencies are investigated with 12 different values occurring in the sampled literature. Some of the values for frequency and severity, for example $\tau_t = 300$ and $n_t = 21$ are used in multiple papers by the same or similar authors but appear in no other works. This limits the possible comparison of any results based on the problem instances examined. More

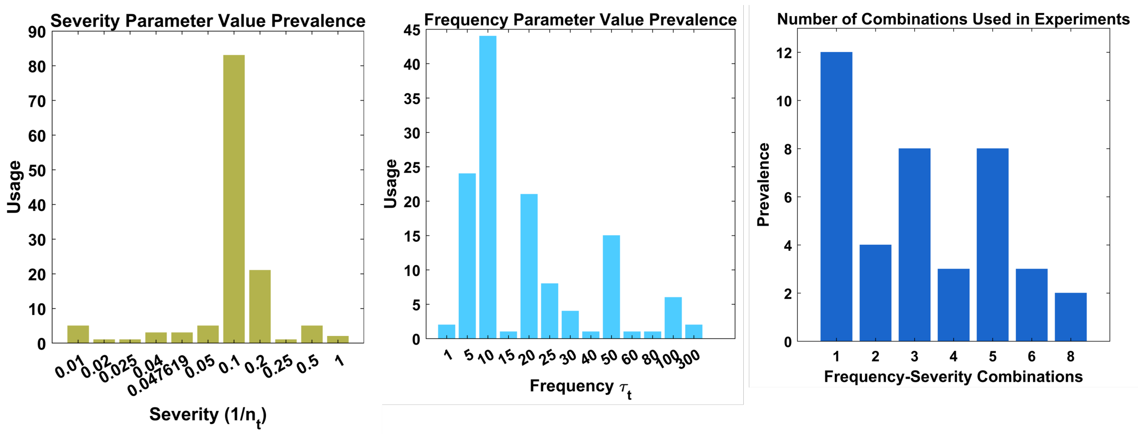


Figure 3.2: (left) The range of frequency values used within DMOPs from the literature. (centre) The range of severity values used within DMOPs from the literature. (right) The number of frequency and severity parameter value combinations used for DMOP experiments in the literature. Values extracted from references list in [37]

generally, the diversity of frequency and severity parameters that exist in the literature illustrate the inconsistency, ill-informed experimentation and a lack of standard practice with regards to frequency and severity parameters.

The number of frequency and severity parameters combinations that are considered is also detrimental to the strength of conclusions made using DMOP benchmarks. Figure 3.2 shows the number of combinations examined in the sampled literature. Each combination of parameters represents a different instance in the frequency-severity combination space for a particular benchmark problem. The most common practice is to use a single combination of values, despite recommendations [41] to use a variety. Within the surveyed literature, the largest number of frequency and severity pairings used in experiments was 8 [53], [61]. Even with this many combinations, unless informed selection of the frequency and severity values occurs, there may be severe limitations to the validity of conclusions made using the results of such experiments for a number of reasons. Foremost, the same frequency and severity values can have different impacts on algorithm performance across different benchmark problems. Secondly, selection of a minimal set of combinations excludes a large proportion of the possible frequency–severity combination instance space. This means that a conclusion of good performance of a novel algorithm on a particular benchmark problem using 8 (or fewer combinations) may not accurately describe the algorithms performance on that problem generally (across the possible frequency–severity combination instance space).

Whilst feasibility of testing a larger number of frequency and severity combinations may be an argument against such criticisms, presented in this thesis is an open-source platform for which comprehensive examination of basic MOEA performance across 630 frequency-severity combinations can be reproduced. Results are presented for commonly employed DMOP benchmarks and a number of well-known MOEAs to provide insights based on the maximum capability of methods not specifically designed for dynamic problems. Recommendations are needed for baseline comparison algorithms, individual benchmarks and for basic dynamic responses with regard to selecting frequency and severity parameters for experiments.

DMOEAs with Change Severity Dependent Responses

A number of algorithms have been proposed which attempt to optimise DMOPs with different change severity values by adapting the response or employing a variety of techniques based on commonly shared assumptions of suitability. The work of Azzouz et al [53] investigates the utility of different response strategies including, local search, memory-based and randomization across high and low change severity problem instances. They also propose a hybrid and an adaptive-hybrid method which combines solutions from the different strategies in a fixed ratio, for the hybrid algorithm, or in proportions dependent on the magnitude of change for the adaptive hybrid algorithm. The severity remains constant throughout each instance and the performance of each algorithm is dependent on the reliable detection of changes. This approach is similar to hyper-heuristic and ensemble methods, or algorithm selection methods that have been proposed for DMOPs [53], [188], [223].

Generally, the concept of algorithm that can adaptively handle non-fixed severity and frequency of changes aligns with the motivation to design algorithms for real world scenarios. It is intuitive that some scenarios involving natural phenomena such as wind or weather, or dynamic routing and the dynamic travelling thief problem (DTTP - See Chapter 4) may have change events with varying magnitudes according to realistic circumstances. Adapting the response of the algorithm in these cases is likely to improve performance. For example, in DTTP instance where many city locations change (high severity), it may be more effective to include random or highly mutated solutions rather than utilising recently found solutions that may be more useful for a low severity change.

Use of MOEAs for DMOPs

There are a wealth of algorithms that have been proposed for solving DMOPs with a variety of complex and innovative methods (See Chapter 2 for more detail). Many of these works use the performance of well-known MOEAs (and simple modifications of them) as baselines for comparison. These ‘static’ or generic MOEAs have not been designed specifically for DMOPs. The NSGA-II [92] has seen comparisons [31], [56], [74], [85], [167], [280], [281] and simple modifications to include random or mutated solutions in response to changes (DNSGA-II algorithm types A and B respectively [141]) are also used [29], [31], [32], [35], [41], [52], [80], [167], [271], [280], [282]–[284]. The NSGA-III algorithm is also used and modified for performance comparisons [216], [271], as is the SPEA2 [285] algorithm [29], [63], [74], [216], [280]. The non-dynamic MOEA/D algorithm [286] is also compared [29], [31], [32], [167], [280], [281] and augmented with Kalman Filter prediction [31], [81], [200], reinforcement learning [287], [288], intensity of environmental change handling [72], [287] & a first order difference model [271].

Widespread use of non-dynamic MOEAs in DMOP experiments is intuitive; these are the algorithms that provide the best performance on static problems and this may extend to dynamic problems too. However it is common that MOEAs are given a ‘random restart’ response where they are forced to reinitialize in response to dynamic changes [29], [53], [63], [72], [167]. In the results presented in the following sections, we compare the simple modification to add random and mutated solutions in response to a change, a ‘do nothing’ approach and contrast it to the poor performance when restarting generic MOEAs for DMOPs.

We focus on the use of non-dynamic MOEAs here rather than the plethora of detailed and complex dynamic methods (detailed in Chapter 2). As discussed, there are many purpose-built methods for DMOP that include prediction mechanisms [77]–[79], [203], [289], multiple archive based methods [31] and ensemble methods [53], [270].

3.3 Multi-Objective Evolutionary Algorithms (MOEAs)

3.3.1 NSGA-II

The non-dominated sorting algorithm was introduced by Deb et al. [92], which has served as one of the most popular MOEAs in the past two decades. As an elitist algorithm, it uses tournament selection for offspring solution generation by crossover with mutation subsequently applied. Pseudocode of the algorithm is included in Algorithm 1. The algorithm uses Pareto-dominance ranking based selection with crowding distance used to break ties when forming the population for the next generation from the combined pool of current and offspring solutions.

3.3.2 NSGA-III

The successor to NSGA-II, the Non-Dominated Sorting Algorithm III [290], as shown in Algorithm 2, makes use of reference points to promote better coverage of the Pareto Set. It is commonly used for problem with many objectives (four or more objectives) as the reference points allow for more guided optimization in higher dimensions. The inclusion here allows for an evaluation of its suitability for ‘simpler’ problems with fewer objectives, but that have dynamic aspects.

Algorithm 1 The Non-dominated Sorting Algorithm II, (NSGA-II)

```
1: Require: $popsize, maxiter$;
2: $P \leftarrow InitializePopulation(popsize)$;
3: $EvaluateObjectiveValues(P)$;
4: $AssignRanks(P)$; \\ based on domination level
5: $iter \leftarrow 1$;
6: while $iter \leq maxiter$ do
7:     $Q \leftarrow GeneticOperators(P)$;
8:             \\ Generate Child Population
9:             \\ Tournament selection,
10:            \\ Crossover & Mutation
11:    $EvaluateObjectivesValues(Q)$;
12:    $P' \leftarrow P \cup Q$;
13:    $RankAndNonDominatedSort(P')$;
14:    $CalculateCrowdingDistance(P')$;
15:    $P \leftarrow SelectNextGeneration(P')$;
16:            \\ Crop $P'$ to $popsize$
17:            \\ with selection based on
18:            \\ minimum Rank and Crowding
19:            \\ Distance;
20:    $iter++$
21: end while
22: return $P$
```

Algorithm 2 The Non-dominated Sorting Algorithm III, (NSGA-III) procedure at generation t, [290]

```
1: Require: $H$ structured reference points $Z^s$,
2:             parent population $P_t, popsize$;
3: $S_t \leftarrow \emptyset, i \leftarrow 1$;
4: $Q_t \leftarrow GeneticOperators(P_t)$;
5: $R_t \leftarrow Q_t \cup P_t$;
6: $(F_1, F_2, \ldots) \leftarrow NonDominatedSort(R)$;
7: while $|S_t| < popsize$; do
8:     $S_t \leftarrow S_t \cup F_i$;
9:     $i \leftarrow i + 1$;
10: end while
11: $F_L \leftarrow F_i$; \\ Last included front
12: if $|S_t| = popsize$; then
13:     $P_{t+1} \leftarrow S_t$
14: else
15:     $P_{t+1} \leftarrow \bigcup_{j=1}^{L-1} F_j$;
16:         \\ Fill population with solutions
17:         \\ up until last included front
18:     $K \leftarrow popsize - |P_{t+1}|$;
19:         \\ Remaining number of solutions
20:         \\ to select
21:     $Normalize(F^M; S_t, Z^r, Z^s)$;
22:         \\ Normalize Objective Functions
23:         \\ and create reference set $Z^r$
24:     $[\pi(s), d(s)] \leftarrow Associate(S_t, Z_r)$;
25:         \\ Associate each member $s \in S_t$
26:         \\ $\pi(s)$ : closest reference point
27:         \\ $d(s)$ : distance from $s$ to $\pi(s)$
28:     $\rho_j \leftarrow \sum_{s \in S_t/F_L}((\pi(s) = j) \quad ? \quad 0 : 1)$;
29:         \\ Compute niche count of reference
30:         \\ point $j \in Z^r$
31:     $Niching(K, \rho_j, \pi(s), d(s), Z^r, F_L, P_{t+1})$
32:         \\ Choose $K$ members, one at a time
33:         \\ from $F_L$ to complete $P_{t+1}$
34: end ifreturn $P_{t+1}$
```

3.3.3 MOEA/D

The Multi-Objective Evolutionary Algorithm based on Decomposition was initially proposed by Zhang et al. [286] and is an alternative approach to the dominance-based NSGA-II, NSGA-III and SPEA2 algorithms discussed here. Pseudocode is given in Algorithm 3, where a series of reference vectors are defined based on a decomposition of the problem to be specified by the user. Solutions compete based on proximity to them whilst focusing on the optimization of the objective functions. Mutation and crossover are employed to produce a single offspring at a time, which is evaluated and replaces all current solutions with worse objective values. Differential Evolution (DE) methods can also form the core process of the optimizer [80], [216].

Algorithm 3 Multi-Objective Evolutionary Algorithm based on Decomposition, (MOEA/D) [286]

```
1: Require: $popsize, (T)NeighborhoodSize,$
2:          $MaxReplacement, (\delta)Replacement$
3:          $Probability, MaxIter;$
4: $P, z^* \leftarrow InitializeSubproblemsAndRefPoint(popsize);$
5: $\lambda \leftarrow InitializeWeightVectors(popsize);$
6: $ParetoApproxSet \leftarrow \emptyset;$
7: **while** $iter \leq maxiter$ **do**
8:     **for** i = 1 to $popsize$ **do**
9:         $Ind \leftarrow SelectNeighborhood(P_i, T, popsize, \delta);$
10:        $y \leftarrow GenerateCandidate(P, Ind);$
11:            \\ Using genetic operators
12:        $z^* \leftarrow UpdateReferencePoint(y, z^*);$
13:        $c \leftarrow 0$
14:        **while** $c \neq MaxReplacement \&\& Ind \neq 0$ **do**
15:            $j \leftarrow GetIndexValue(Ind);$
16:            **if** $g_j(y|\lambda^j, z^*)$ is better than $g_j(P_j|\lambda^j, z^*)$ **then**
17:                $P_j \leftarrow y;$
18:                $c \leftarrow c + 1;$
19:            **end if**
20:            $Ind \leftarrow RemoveIndexValue(j, Ind);$
21:        **end while**
22:    **end for**
23:    $ParetoApproxSet \leftarrow UpdateParetoApproxSet(P, PS);$
24: **end whilereturn** $PS$
```

Algorithm 4 Strength Pareto Evolutionary Algorithm 2, (SPEA2) [285]

```
1: Require: $popsize, ArchiveCapacity, maxiter$ ;
2: $P \leftarrow InitializePopulation(popsize);$
3: $ExtArch \leftarrow \emptyset;$
4: **while** $iter \leq maxiter$ **do**
5:     $EvaluateObjectivesValues(P);$
6:     $EvaluateObjectivesValues(ExtArch);$
7:     $UpdateExtArch;$
8:         \\ Add non-dominated solutions
9:         \\ from $P$ to $ExtArch$
10:    **if** $|ExtArch| > ArchiveCapacity$ **then**
11:        $TruncationOperator(ExtArch);$
12:    **end if**
13:    $P' \leftarrow GeneticOperators(P);$
14:        \\ Apply mutation and crossover
15:        \\ operators to generate offspring
16:    $P \leftarrow P';$
17: **end whilereturn** $P, ExtArch$
```

As a very popular algorithm, MOEA/D has been the foundation of many novel DMOEA algorithms. The algorithm has been modified to include Kalman-Filter Predictions [31], [81], [200], Surrogate Assisted Transfer Learning Models [283], Difference Models [282], Controlled Extrapolation [80] & Inverse Modelling Approaches [281].

Table 3.1: Experimental parameters for algorithms and those common across all problems. See PlatEMO or the source code for the DPTP implementation for more information.

Algorithm Parameters		
All	Population Size (N)	100
NSGA-II, NSGA-III, SPEA2	Polynomial Mutation Probability	1/D
	Mutation distribution index	20
	Simulated Binary Crossover probability	1
	Crossover distribution index	20
NSGA-III	Reference Vector Set Size	100 (N)
MOEA/D	Aggregation Function Index	2

Problem Parameters	
Decision Variables (D)	20
Initial t value	0
Maximum t value	15
Dynamic Onset Delay	50 (generations)
No. of repeats per $n_t - \tau_t$ combination	5

3.3.4 SPEA2

The Strength Pareto Evolutionary Algorithm was first introduced by Zitzler & Thiele [291] and subsequently improved as a second version by Zitzler et al. [285]. Example pseudocode is given in Algorithm 4. SPEA2 implementations usually maintain an external archive of solutions together with an exploratory population, however the PlatEMO implementation uses a truncation of the population at each selection stage based on a distance based dominance relation.

These algorithms are implemented as in the PlatEMO platform and the relevant parameters are listed in Table 3.1. The Hypervolume Difference measurement is selected for its relevance and ease of understanding. It is simply calculated using the following equation:

$$HVD_n = (OptimalHypervolume)_N - (AchievedHypervolume)_N \tag{3.2}$$

where n is the number of solutions used in the optimal calculation and as the population size for the algorithms. The optimal value for this is 0, meaning that the solutions found by the algorithm perfectly match the optimally distributed Pareto Front sample. The magnitude of any deviation from true hypervolume is specific to the problem and so we forego normalization in this instance.

Table 3.2: DMOP benchmark problems. The number of objectives, M, is limited to 2 and the number of decision variables, n, is fixed at 20 based on literature usage. All objective functions are minimizations.

Name	Objective Functions	Decision Variables	Type
FDA1 [28]	$f_1(\mathbf{x}) = x_1 \quad f_2(\mathbf{x}) = g * h$ $g(\mathbf{x_{II}}) = 1 + \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2; \quad h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}};$ $G(t) = \sin(0.5\pi t);$	$x_I \in [0,1]; \quad x_{II} \in [-1,1];$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n);$	I
FDA3 [28]	$f_1(\mathbf{x}) = \frac{1}{\lvert\mathbf{x_I}\rvert} \sum_{x_i \in \mathbf{x_I}} (x_i^{F(t)}) \quad f_2(\mathbf{x}) = gh$ $g(\mathbf{x_{II}}) = 1 + G(t) + \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2; \quad h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}};$ $G(t) = \lvert\sin(0.5\pi t)\rvert; \quad F(t) = 10^{2sin(0.5\pi t)};$	$x_I \in [0,1]; \quad x_{II} \in [-1,1];$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n);$	II
dMOP1 [76]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + 9 \sum_{x_i \in \mathbf{x_{II}}} (x_i)^2$ $h = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 0.75sin(0.5t) + 1.25$	$x_i \in [0,1]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	III
dMOP2 [76]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + 9 \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2$ $h = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 0.75sin(0.5t) + 1.25; \quad G(t) = sin(0.5\pi t)$	$x_i \in [0,1]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	II
ZJZ [79]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t) - x_1^{H(t)})^2; \quad h = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 1.5 + G(t); \quad G(t) = sin(0.5\pi t)$	$x_I \in [0,1]; \quad x_{II} \in [-1,2]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	II
DIMP1 [78]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + 9 \sum_{x_i \in \mathbf{x_{II}}} (x_i - G_i(t))^2; \quad h = 1 - \left(\frac{f_1}{g}\right)^2$ $G_i(t) = sin(0.5\pi t + 2\pi \frac{i}{n+1})^2$	$x_I \in [0,1]; \quad x_{II} \in [-1,1]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	I
DIMP2 [78]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + 2(n-1) + \sum_{x_i \in \mathbf{x_{II}}} [(x_i - G(t))^2 - 2cos(3\pi(x_i - G_i(t)))]$ $h = 1 - \sqrt{\frac{f_1}{g}}; \quad G_i(t) = sin(0.5\pi t + 2\pi \frac{i}{n+1})^2$	$x_I \in [0,1]; \quad x_{II} \in [-2,2]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	I
HE1 [41]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + \frac{9}{n-1} \sum_{x_i \in \mathbf{x_{II}}} x_i;$ $h = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} sin(10\pi t f_1)$	$x_i \in [0,1]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	III
HE2 [41]	$f_1(\mathbf{x_I}) = x_1; \quad f_2 = gh$ $g(\mathbf{x_{II}}) = 1 + \frac{9}{n-1} \sum_{x_i \in \mathbf{x_{II}}} x_i; \quad h = 1 - \left(\sqrt{\frac{f_1}{g}}\right)^{H(t)} - \left(\frac{f_1}{g}\right)^{H(t)} sin(10\pi t f_1)$ $H(t) = 0.75sin(0.5\pi t) + 1.25;$	$x_i \in [0,1]$ $\mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \dots, x_n)$	III
JY1 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I);$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I);$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2;$ $G(t) = \sin(0.5\pi t); \quad A_t = 0.05; \quad W_t = 6;$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	I
JY2 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I);$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I);$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2;$ $G(t) = \sin(0.5\pi t); \quad A_t = 0.05; \quad W_t = \lfloor 6\sin(0.5\pi(t-1)) \rfloor;$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	II
JY3 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi y_1);$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - y_1 + A_t \sin(W_t \pi y_1);$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} (y_i^2 - y_{i-1})^2;$ $y_1(t) = \lvert x_1 \sin((2\alpha + 0.5)\pi x_1) \rvert; \quad y_i = x_i, i = 2, \dots, n;$ $\alpha = \lfloor 100\sin^2(0.5\pi t) \rfloor; A_t = 0.05; \quad W_t = \lfloor 6\sin(0.5\pi(t-1)) \rfloor;$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	II
JY5 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I);$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I);$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} x_i^2; \quad G(t) = \sin(0.5\pi t);$ $A_t = 0.3\sin(0.5\pi(t-1)); \quad W_t = 1;$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	III
JY6 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I);$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I);$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} (4y_i.^2 - \cos(K_t \pi y_i) + 1));$ $G(t) = \sin(0.5\pi t); \quad A_t = 0.1; \quad W_t = 3;$ $K_t = 2 * \lfloor 10 * \lvert G(t) \rvert \rfloor; \quad y_i = x_i - G(t)$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	I
JY8 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I)^{\alpha_t};$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I)^{\beta_t};$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} x_i^2; \quad G(t) = \sin(0.5\pi t);$ $A_t = 0.05; \quad W_t = 6;$ $\alpha_t = \frac{2}{\beta_t}; \quad \beta_t = 10 - (9.8 * \lvert G(t) \rvert)$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	III
JY9 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I);$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I);$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i + \sigma - G(t)).^2;$ $G(t) = \lvert\sin(0.5\pi t)\rvert; \quad \sigma \equiv \left\lfloor \frac{\tau}{\tau_t \rho_t} \right\rfloor \pmod 3$ $A_t = 0.05; \quad W_t = \lfloor 6\sin^{\sigma}(0.5\pi(t-1)) \rfloor;$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	(I,II,III)
JY10 [86]	$f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(x_I + A_t \sin(W_t \pi x_I)^{\alpha_t};$ $f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t \sin(W_t \pi x_I)^{\beta_t};$ $g(\mathbf{x_{II}}, t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i + \sigma - G(t)).^2;$ $G(t) = \lvert\sin(0.5\pi t)\rvert; \quad \sigma \equiv \left(\left\lfloor \frac{\tau}{\tau_t \rho_t} \right\rfloor + R\right) \pmod 3$ $A_t = 0.05; \quad W_t = 6; \quad \alpha_t = \beta_t = 1 + \sigma G(t)$	$\mathbf{x_I} = (x_1);$ $x_I \in [0,1];$ $\mathbf{x_{II}} = (x_2, \dots, x_n)$ $x_{II} \in [-1,1];$	(I,II,III)

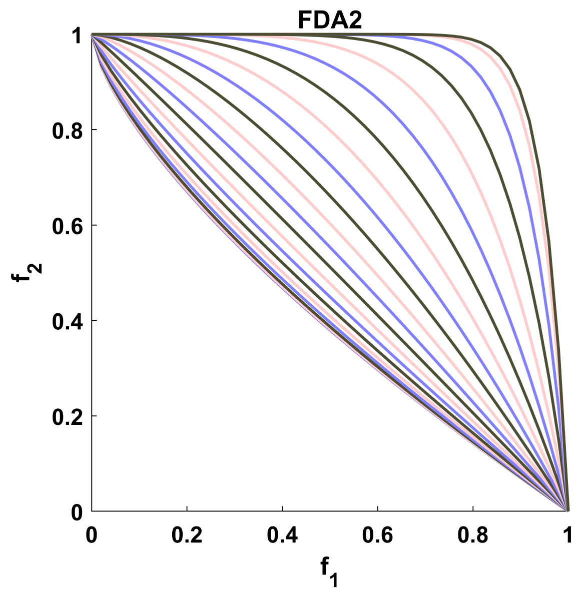


Figure 3.3: Obtainable POFs for the FDA2 problem according to the problem definition in [28]

3.4 DMOP Benchmarks

The FDA2 problem

The FDA2 problem, originally defined by Farina et al [28], has been quoted with slight differences in its definition across the literature [41], [225]. Some works, including some recently published, use other problems from the FDA set but avoid FDA2 in particular [75], [283], [292], whilst others that use the FDA2 problem conspicuously omit any figure shown the attainment of the POF for FDA2 [282], [293]. The work by Chen et al. [294] uses the FDA2 problem to test an algorithm for finding robust solutions over time. The obtainable set of POFs for FDA2 is illustrated within this work and resembles those shown in Figure 3.3, which does not match the presented figures in the original paper [28] and in [41]. Due to these issues, the FDA2 problem is not considered within this work.

A modified version, named $FDA2mod$ was defined in [141]. The problem has significant changes to the objective functions but is still a Type-II problem with a PF that changes from convex to concave as the value of t changes. The severity of change is dependent on the specified frequency of change, making an examination of frequency and severity combinations difficult. $FDA2mod$ is not considered here.

3.5 DPTP Platform

To effectively examine the severity and frequency parameters whilst ensure reproducibility and transparency of these results, we provide the Dynamic Parameter Testing Platform (DPTP). This is a MATLAB program with a GUI with in-built plotting, real-time optimization visualization and the ability to graph results for pairwise combinations of different parameters. The version used to generate the results in this Thesis can be found at https://github.com/DPTP2022/DPTP.

3.5.1 Motivation and Origin

Of the surveyed literature, only the works proposing the GTA [81] and SDP [32] suites provided code to replicate the problem implementations. Given that the equations defining the FDA problems and others can be inconsistent in the literature [28], [41] provision of the code used to generate the results is a simple but important way to ensure the reproducibility of experiments and results by others.

We therefore construct a framework to demonstrate MOEA performance on DMOPs. Inspired by the existing PlatEMO MATLAB implementation [295], our platform is adapted to include DMOPs and allow for experimentation on dynamic frequency and severity. The class structures and main running functions of the framework used here are distinct from those used in PlatEMO and this platform is named **D**MOP **P**arameter **T**esting **P**latform (DPTP).

3.5.2 Basic Operation and Outputs

Through the GUI, experimentation on any of the problem parameters is possible, including the number of decision variables and their ranges, the number of objectives. The selected algorithm can be augmented with any of the simple response mechanisms and additional parameters controlling the dynamics including the range of t, cycling behaviour and delay onset can be controlled. DPTP allows for fine-grained examination for ranges for two parameters to produce heatmaps like those presented in this paper. A variety of metrics can be recorded as in PlatEMO, including GD, IGD, HV and others. The well-designed and comprehensive RDP [72], GTA [81] and SDP [32] benchmark sets will be added in future in addition to a number of dynamic algorithms.

3.5.3 Provision of Code, Reproducibility and Further Additions

As the volume of research increases continuously, additional attention is needed towards the reproducibility of results and inclusion of data sets; the provision of source code to accompany papers; and the transparency and thoroughness in explanations of experimental methodologies. Some progress is being made concerning this, for example, the introduction of the reproducibility ‘medals’ to accompany publications in some journals, including the ACM Transactions on Evolutionary Learning & Optimization. Whilst this is crucial for efficient scientific progress going forward, previous publications are not held to the same standards and retrospective application of these standards is unlikely.

The results provided within this work highlight the need for standardisation of baseline performance practices, particularly in the dynamic multi-objective research domain.

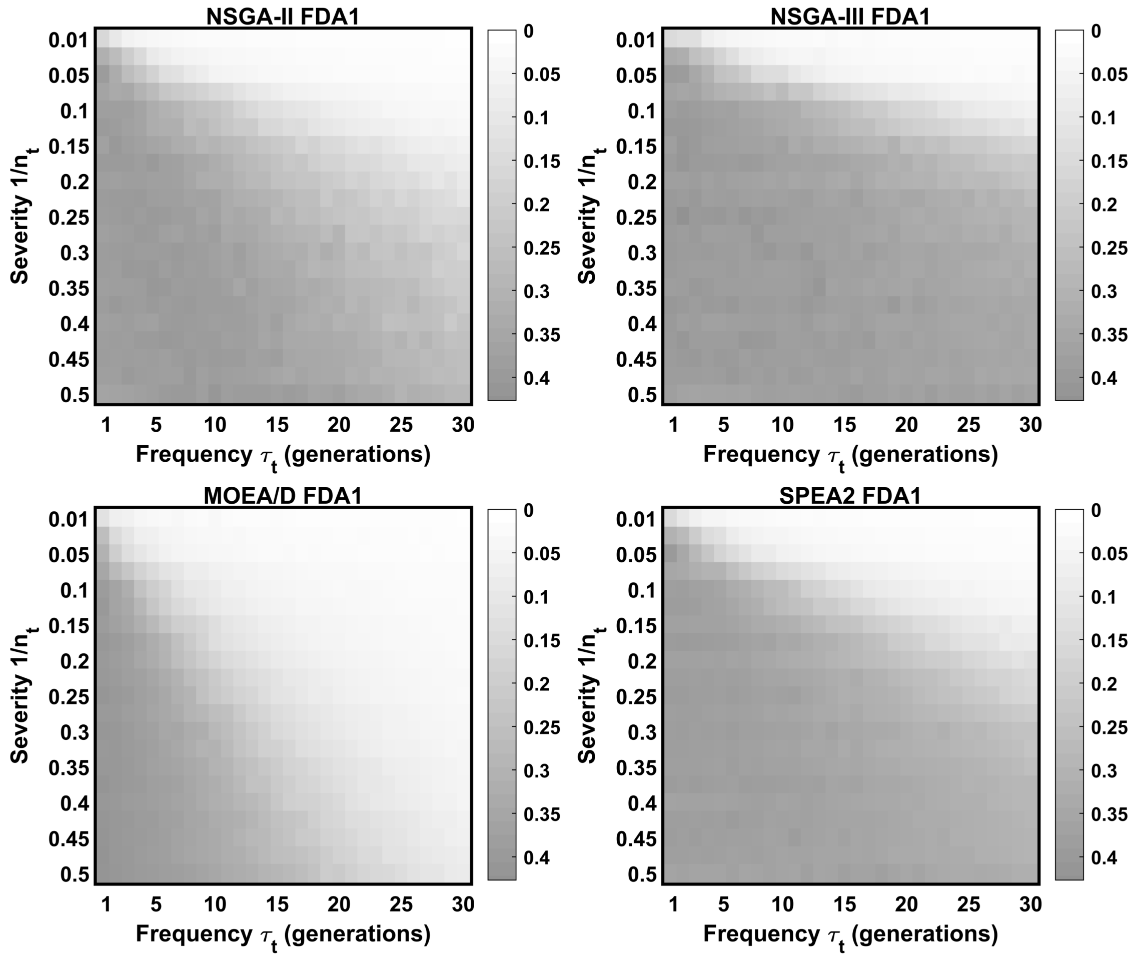


Figure 3.4: MOEA hypervolume difference (HVD) performance heatmaps across 630 severity and frequency combinations on the FDA1 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. Each subplot illustrates the performance across these instances of a difference MOEA from NSGA-II, NSGA-III, MOEA/D and SPEA2.

Clear methodology, evidenced justifications and transparency and conditioning of conclusions allow for cohesive progression in this research area.

3.6 Comprehensive Testing of Frequency-Severity Combinations

A detailed explanation of the results is presented for the FDA1 problem, with the key features used in the subsequent of other problems and comparisons in this section.

Figure 3.4 shows the heatmaps of Hypervolume Difference (HVD) across the frequency and severity parameters for the different MOEAs require to establish a baseline achievable performance. Each cell in a heatmap corresponds to an an algorithms performance across the dynamic intervals of the problem instance with a specific combinations of frequency and severity parameter values. The value for each cell is calculated as the mean HVD at the end of each dynamic interval (just before the next change occurs) for 5 repetitions. As the number of dynamic intervals is fixed at 30, this means each cell is the mean of 150 data measurements. Since the HVD measures the difference to the optimum achievable hypervolume (as in Eq. 3.2), normalisation is not compulsory for comparison on the same problem. Normalisation becomes difficult for the heatmaps as the measurements are absolute differences to the optimum hypervolume in each dynamic interval. In Type II and III problems the POF changes and so this optimum HV value may differ, however the HVD remains a robust measure for determining good performance. Comparison between problems will require normalisation within each dynamic interval of an instance to a percentage difference instead. Other popular measurements such as (Inverted) Generational Distance could also be used in place of hypervolume here.

The values represented in each cell and the corresponding shade in the colour range are based on the algorithm's achieved mean HVD value across th intervals of the problem. The colour range is bounded by zero for a perfect HV attainment in every interval and by the worst achieved mean HVD across all algorithms in a considered comparison (series of subplots). Therefore the colour range may be different for different problems as the maximum hypervolume (not achieved - darkest value) varies between problems due to POF geometry. Regarding normalization procedures, there are two possibilities that are discounted here. Firstly normalization within a comparison; normalizing the colour range to the worst achieved values of any algorithm in the comparison may improve readability, however in an event where good performance is achieved by all, the threshold for what is considered 'poor performance' is higher and the wider context of the results is lost. Secondly, normalization within each heatmap to the worst achieved value; this further removes the ability to compare problems as the colour range becomes based on the range of performance for a specific algorithm. By instead choosing to use absolute values, we show the raw performance capability of algorithms and facilitate easier comparison of these results for future works.

Each subplot in Figure 3.4 illustrates the performance over the frequency-severity combination space for a particular non-dynamic MOEA algorithm. Basic interpretation allows for significant insights to be gained. Intuitively there are differences in the patterns visible for each algorithm. Where the a cell is white in the heatmap, this corresponds to a HVD of zero, meaning a perfect approximation to the POF is achieved. The darker the shade of grey, the greater the difference to the optimal set of solutions is in terms of achieved hypervolume. On all of the heatmaps there is a clear triangular motif for the white region with differing geometry for each of the algorithms. The lowest frequency changes (largest τ_t values) and the lowest severity values (smallest $1/n_t$ values) represent the easiest instances in the combination space and are white cells in the top right of each heatmap. The MOEAs can easily handle the FDA1 problem with these parameter values. Conversely, the most difficult instance arise with combinations of high severity and high frequency values (rapid, high-magnitude change events). These are uniformly not well handled by the MOEAs, visible in the grey cells in the bottom-left of each heatmap.

The visible boundary between good and bad performance, lighter grey regions at the edge of the white cells, is different for each of the MOEAs. At lower frequencies, optimal hypervolume attainment can be achieved with higher severity changes. At higher frequencies (lower τ_t values) the tolerance for higher severity changes is much lower, with a steep decline in the performance as severity values increase. There is a trade-off between frequency and severity that results in the triangular shape of good performance, with the extent tolerance varying between the algorithms.

The MOEA/D algorithm has the largest proportion of white cells in the heatmaps compared to the other algorithms. This illustrates that MOEA/D has better coverage across the problem instances generated by different frequency and severity parameter combinations. The NSGA-II, NSGA-III & SPEA2 algorithms all have similar coverage of the combination space, with subtle differences visible in the heatmaps. Firstly, SPEA2 has uniform poor hypervolume attainment as the values of the frequency and severity parameters respectively decrease and increase. For NSGA-II this poor performance is less uniform, with some better performance at low frequency, high severity changes and occasionally (but unreliable) better performance towards the most difficult combinations. The NSGA-III algorithm has less coverage (less white area) than the NSGA-II and SPEA2 algorithms, indicating that this algorithm is more sensitive to different frequency and severity parameters in terms of achieving good performance.

Generally, the results in Figure 3.4 indicate that MOEAs can provide competitive performance on DMOPs without specialised response mechanisms. This good performance is conditioned on the selection of appropriate frequency and severity parameter values, however the general coverage achieved covers many of the combinations previously used in the literature. This highlights the importance of careful selection of the parameters and the necessity of the work presented here. Previous conclusions on novel algorithm performance made using instances with frequency and severity combinations that can effectively be handled by non-specialized MOEAs require further investigation. More generally, as a non-standard selection of frequency and severity parameter combinations has been previously explored in the literature, the wider relevancy of conclusions on novel algorithm performance is limited to works that use the same instances and should more clearly be conditioned on the parameters used.

Since the performance of these MOEAs differs on the same problem (FDA1 here), there is potential for exploitation of this knowledge to improve the appearance of novel algorithm results. Specifically, if a novel DMOEA is examined against NSGA-II with $\tau_t = 10, n_t = 10$ the results are likely to better for the new method. However, if MOEA/D were to be used as the comparison algorithm, since this combinations on FDA1 can effectively be handled by MOEA/D, the conclusions may be different. This highlights the need for a standard experimental practice for DMOP benchmarks in terms of frequency and severity. The contribution of this work is not necessarily to recommend specific parameter combinations to use, although recommendations are given, but more so that greater awareness is required when selecting test instances for problems. Variety has commonly been called for in previous works, both in terms of benchmark problems, and comparison with existing algorithms [32], [37] but also in frequency and severity parameters [41]. Whilst the first two are mostly respected, the scope of examined dynamics parameters has been limited thus far.

The issues highlighted can be mitigated in two ways. Firstly, through a greater understanding of frequency and severity impacts across a variety of benchmarks via comprehensive testing of combinations. Secondly, through recommendations for regions of the combination space in each problem to be investigated or avoided based on the performance achievable with unmodified MOEAs. Collectively, the examined MOEAs provide a baseline insight into the combinations of frequency and severity that pose little difficulty and should be used only to verify baseline performance of new methods. Furthermore, for each problem examined here, they provide insights into the regions of the combination space that require further investigation and intelligent design of algorithms to effectively handle. This informed approach to frequency and severity parameter selection should be a driving force in the design of experiments going forward.

The remainder of this section examines the results of this methodology on a number of popular and recent DMOP benchmarks. It provides a more complete picture of the variety of impacts and combination space topologies that present across well known problems.

3.6.1 Comparison of Static MOEAs

There are a large number of DMOP benchmarks in the literature, however the prevalence of their usage varies based on recency, relevance and previous usage. Here, results for frequency and severity combinations are presented for a sample of the popular and some more recent benchmarks.

The performance heatmaps for the four MOEAs on the FDA3 problem are illustrated in Figure 3.5. This demonstrates the difference in performance in the frequency-severity combination space between different benchmark problems. The differences to the FDA1 heatmaps in Figure 3.4 are relatively minor but nevertheless showcase the importance of informed frequency-severity parameter selection. FDA3 contians dynamic components that alter the density of solutions on the POF as time goes on.

Generally the patterns present in the heatmaps are similar to those in FDA; with the better performance region similar in geometry and with MOEA/D having the largest good performance area. Notably, the attainment of perfect hypervolume is limited to the lowest severity changes for NSGA-II, NSGA-III & SPEA2 with a zero-value HVD absent. MOEA/D however achieves optimal HVD values and good performance up to severity

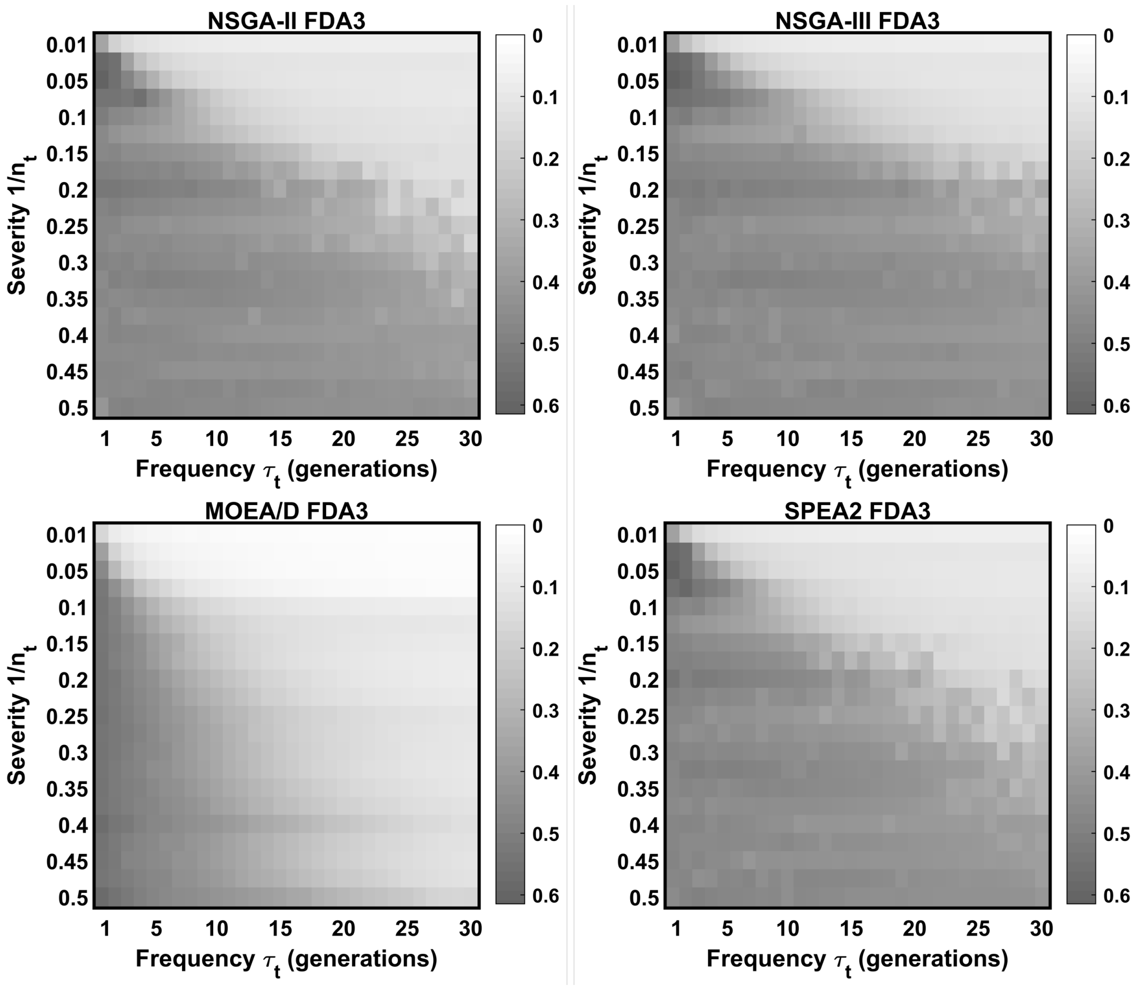


Figure 3.5: MOEA hypervolume difference (HVD) performance heatmaps across 630 severity and frequency combinations on the FDA3 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. Each subplot illustrates the performance of a different MOEA; NSGA-II, NSGA-III, MOEA/D and SPEA2 are shown.

values of $\frac{1}{n_t} = 0.05$ for frequencies of $\tau_t = 10$ or greater. MOEA/D also has uniformly changing performance as the frequency decreases (τ_t becomes larger), however, the other algorithms have variable performance at frequencies above $\tau_t = 15$ and severity values greater than $\frac{1}{n_t} = 0.15$. There is also a region of the combination space which all algorithms appear to struggle with; high-frequency changes at severity values between $0.025 \leq \frac{1}{n_t} < 0.1$ with a correlated increase in performance as severity increases and frequency decreases. NSGA-II & SPEA2 have similar performance coverage and NSGA-III is suitable in fewer combinations of frequency and severity.

The FDA3 problem highlights the complex diversity of performance across the combinations of frequency and severity and suggests the suitability of algorithms can be unpredictable for some regions of the combinations space. This reaffirms the necessity of standardisation of experimental practice, awareness of parameter impacts and conditioning of conclusions on the frequency and seveirty combinations used.

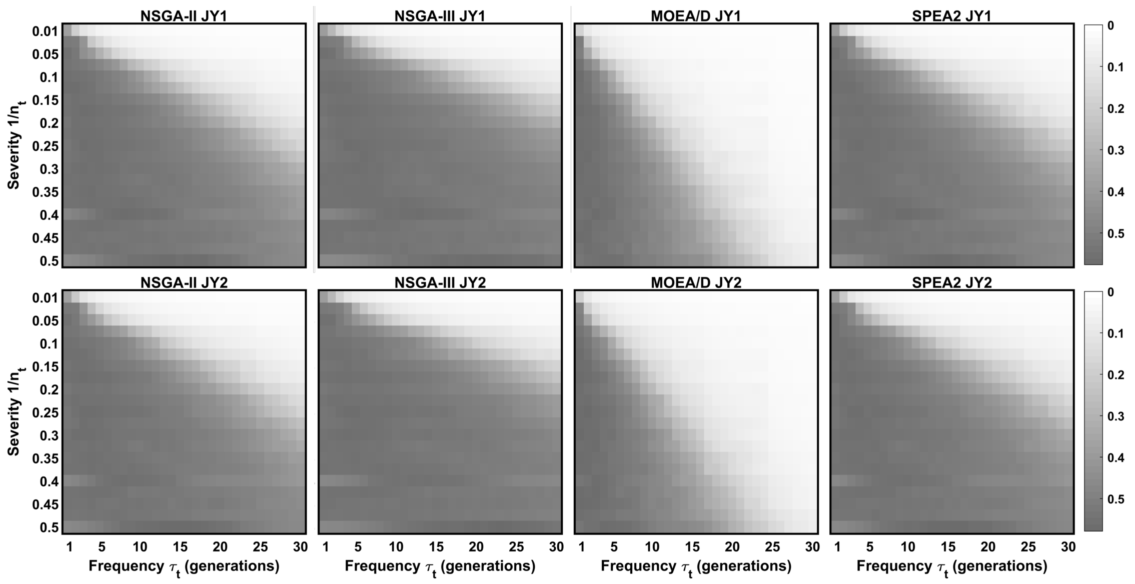


Figure 3.6: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for (top row) the JY1 and (bottom-row) the JY2 problems. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. The colour bar on the right applies to all subplots in a particular row: the gradient is equated for comparison.

The JY problem set

The problems in the JY set were defined by Jiang et al. [86]. As a more recent benchmark suite, some of the problems are constructed based on observable features in real world DMOPs. So far, few works have used these problems [282], but more recently defined suites (RDP [72], SDP [37]) have yet to receive much attention. Source code is not provided in the original paper, so all problems are recreated based on the descriptions alone. The JY4 & JY7 problems are not included in these results. The POF of the JY4 contains many small disconnected regions for certain values of t. This makes measurement of the optimal HV difficult when using a limited number of solutions and as such the JY4 problem needs an alternative measurement system. The JY7 problem was omitted due to comparability issues; the $n = 20$ value used for all problems featured gave inconsistent and inconclusive results. This is a challenging problem and further work will address application of the frequency-severity combination methodology for this problem.

Establishing Limitations of MOEA Performance - JY1 & JY2

The heatmaps of MOEA performance on the JY1 & JY2 problem across the frequency-severity combination space, shown in Figure 3.6, mirror closely the results seen for the FDA problems. Both JY1 & JY2 problems have more definite boundaries between the good and bad performance regions of the space. The NSGA-II and SPEA2 algorithms have very similar patterns of performance, achieving good performance up to larger severity changes at lower frequency changes; as the frequency increases the severity of change that these algorithms can handles decreases. The NSGA-III algorithm has a smaller re-

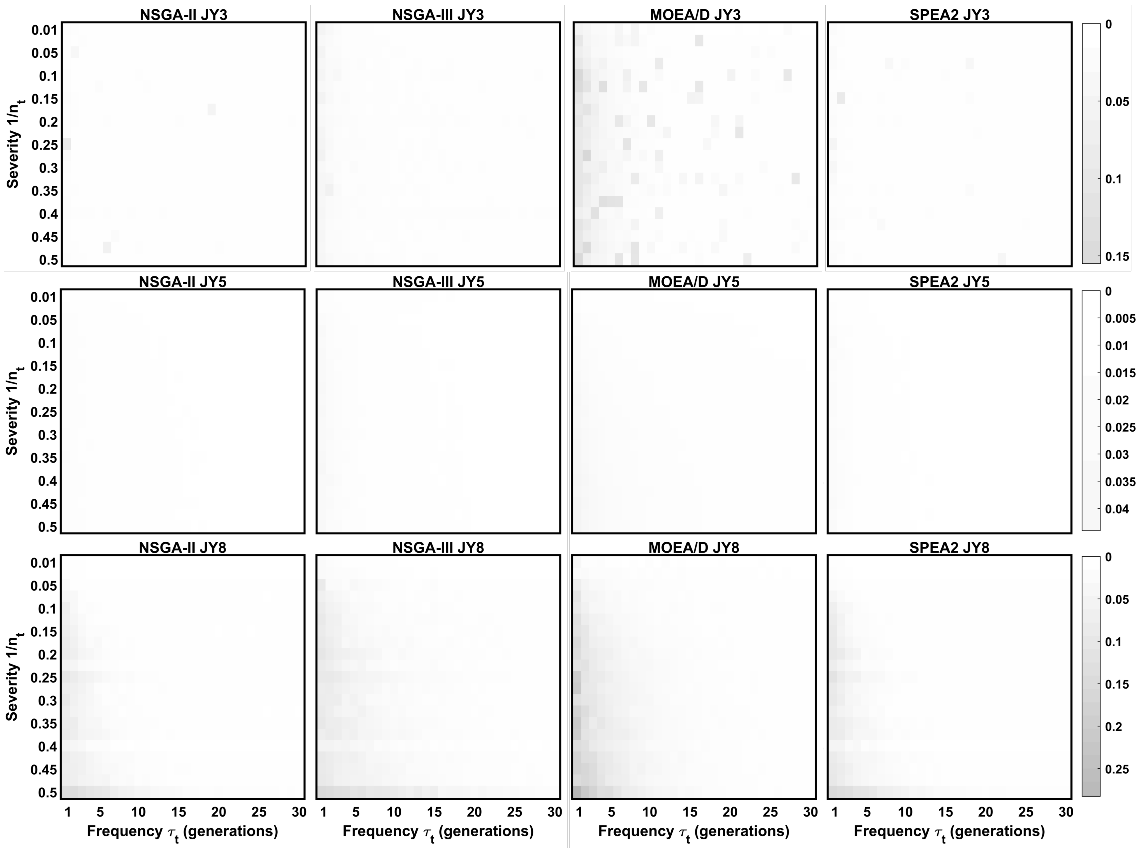


Figure 3.7: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for (top row) the JY3, (middle row) JY5 and (bottom-row) JY8 problems. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance.

gion of good performance – fewer combinations of frequency and severity can effectively be handled when using this algorithm. As with the FDA1 & FDA3 problems, the MOEA/D algorithm achieves the best coverage of frequency-severity combinations in terms of achieving a minimal HVD. the darker shade of grey between these heatmaps and those for the FDA problems is due to the difference in POF geometry resulting in different maximum hypervolumes and therefore different maximum HVD values for the worst performance.

MOEA Capability - JY3, JY5 & JY8

The JY3 problem required modification from the paper definition for proper execution. The problem contains variable interdependence at adjacent variable indexes. However, the second index's ($\mathbf{x_{II}}(1) \equiv \vec{x}(2)$) dependency was changed to the final index. This means the dependency 'wraps' around the decision vector. Without this alteration, the stated POF in the original definition is not achievable.

The frequency-severity combinations space performance results for the JY3, 5 & 8 problems are shown in Figure 3.7. The heatmaps appear to have almost exclusively zero

HVD, meaning optimal tracking of the POF, or near-zero values in the majority of the combinations space. Notably, the most difficult combinations, the high frequency and high severity instances (bottom left of each heatmap), are where the algorithms perform slightly worse. MOEA/D has sporadic poorer performance at seemingly random combinations of frequency and severity on JY3, with NSGA-II, NSGA-III & SPEA2 having much fewer but still random occurrences with better performance. All of the MOEAs effectively handle the JY5 problem across all combinations of frequency and severity. For JY8, MOEA/D appears to have the worst performance in the high-severity high frequency region and more generally at the highest frequency of dynamics. The decomposition mechanism sets MOEA/D apart from the other algorithms and may explain the relatively worse performance as variable interdependency may interfere with its operation in some cases. Generally, the MOEA algorithm performance on these problems indicates that some proposed DMOP benchmarks are not challenging even with most rapid and severe dynamic changes. As all problems presented here use $n = 20$ decision variables, (as this a the consistently used value in the literature), intuitively, increasing n may provide a more challenging DMOP.

MOEA Difficulty - JY6, JY9 & JY10

Performance heatmaps for the JY6, JY9 & JY10 problems are shown in Figure 3.8; all the algorithms struggled more than on the previously presented problems.

For JY6, the only good performance is seen at the lowest severity value and gets better for all algorithms as the frequency decreases (τ_t value increases). The remainder of the frequency and severity combinations cannot be effectively handled by the static MOEAs. There are a selection of severity values ($\frac{1}{n_t} = 0.25, 0.4, 0.5$) for which performance is slightly better at the highest frequencies for NSGA-II, NSGA-III and SPEA. JY6 is a Type I problem meaning that the POS changes as a result of the dynamics; it may be that these severity values result in subsequent dynamic intervals with sufficiently similar problem states such that the previous population is relevant. However, given this observation the performance on these combinations is still poor with large HVD values recorded. The MOEA/D algorithm again has an additional region of slightly better performance than the other algorithms examined. From the lowest frequency changes with increasing severity there are slightly lower HVD values recorded; as the frequency increases, the maximum severity value that this performance applies to decreases.

Both JY9 and JY10 have less uniform patterning than the previously examined problems. These problems have complex and rich dynamic behaviours, changing through Types I, II & III in sequence or randomly for JY9 & JY10 respectively. The performance heatmaps for JY9 contain a distinct horizontal banding across the combinations. This means there is similar HVD values achieved for a given severity value regardless of the frequency of changes. Generally, there is little to distinguish the performance of NSGA-

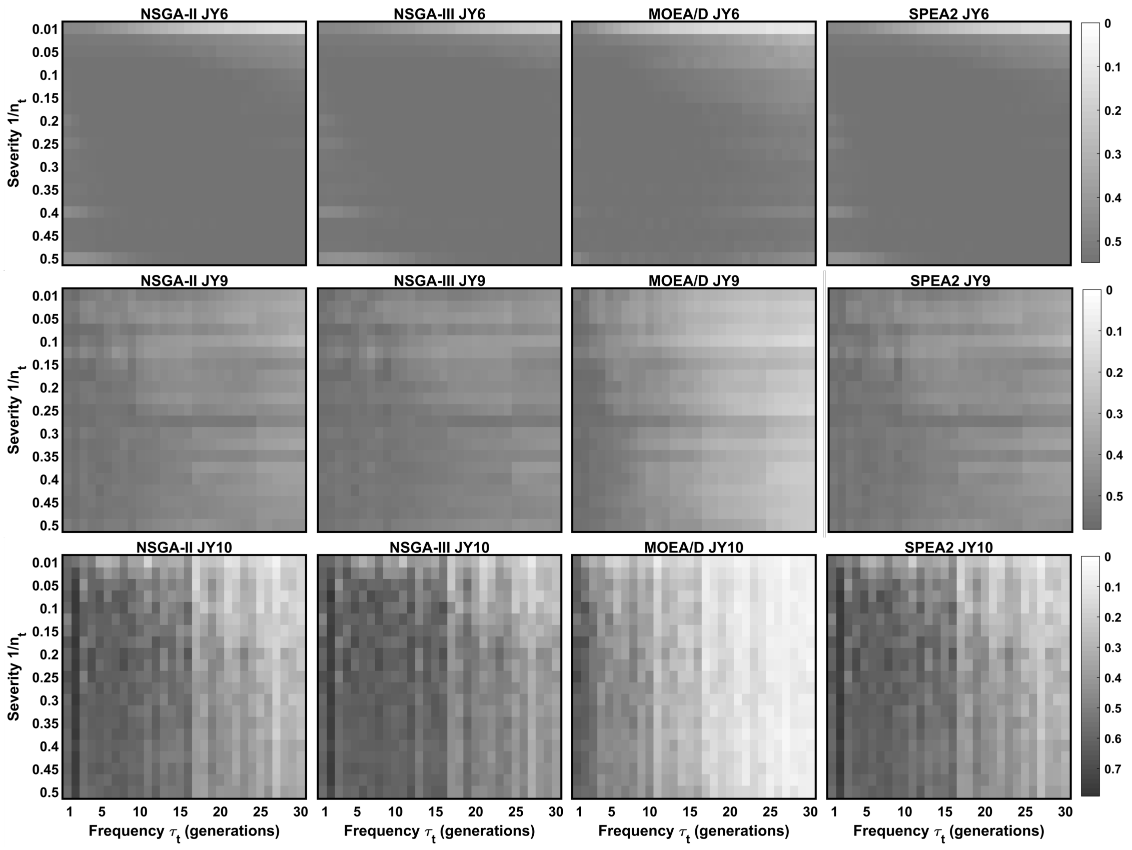


Figure 3.8: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for (top row) the JY6, (middle-row) JY9 and (bottom-row) JY10 problems. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance.

II, NSGA-III and SPEA2 across the combination space. Notably, MOEA/D still retains the improved capability across more of the combinations than the other algorithms, this is especially visible for JY10. The patterning for JY10, in contrast to JY9, shows vertical banding. This implies that similar HVD values are achieved for specific frequency values, regardless of the severity of change. A frequency of $\tau_t = 2$ appears challenging for the algorithms for all but the lowest severity of change.

These three problems highlight that there are DMOP benchmarks that cannot effectively be handled by MOEAs for many or most of the investigated combinations of frequency and severity. Furthermore, the patterning in the heatmaps of JY9 and JY10 highlights that the frequency and severity parameters have significant impact on the problem in terms of the algorithm's ability to handle the dynamics. More specifically, the frequency and severity parameters have complex effects that are specific to the problem, highlighting the necessity of informed selection of parameters. Adoption of a single set of frequency and severity parameters for a variety of the benchmarks shown so far will result in a set of instances with variable challenge and unclear motive for selection. Informed choice

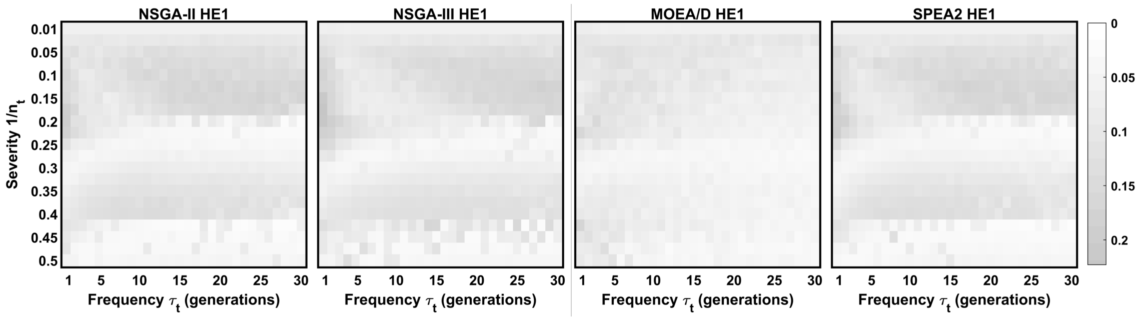


Figure 3.9: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for the HE1 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance.

and improved understanding of frequency and severity impacts for each problem can help with the design of experiments.

Complex Performance Patterning - HE problems

The HE problems were defined by Helbig and Engelbrecht in [255]. The HE1 & HE2 problems both have a disconnected POF with an increasing number of disconnected regions as the value of t increases and the geometry becomes more complex. There are 10 defined DMOP benchmarks with the HE prefix, however widespread use is not apparent in the literature ([52] used HE3, HE4 & HE5).

The performance heatmaps for HE1 across the frequency and severity combination space are shown in Figure 3.9. Each of the heatmaps shows complex patterning across the combinations. MOEA/D has the best coverage of low HVD scores with good performance at the lowest frequency of change and around the middle of the examined severity range. The performance of NSGA-II, NSGA-III and SPEA2 are all very similar, including non-uniform good performance at the highest severity changes across all frequencies.

The distinct patterning builds upon the previous observations that frequency and severity have complex impacts on DMOP benchmarks. Selection of parameters for the HE problems should be done carefully in order to best test algorithms; some regions of the frequency-severity combinations space do not provide challenging test environments as they can be easily handled by the MOEAs.

The HE2 problem, illustrated in 3.10 has somewhat similar patterning for all the algorithms as in the HE1 problem. For the NSGA-II, NSGA-III & SPEA2 algorithms, the lowest severity instances are trivial at all frequencies. As the severity of changes increases, there are two pairs of alternating good and bad performance that persist across all frequency values. Between $0.025 \leq \frac{1}{n_t} \leq 0.175$ and $0.325 \leq \frac{1}{n_t} \leq 0.4$ the algorithms perform worse than outside of these ranges where a near-zero HVD is achieved for most frequency values. The MOEA/D algorithm has better HVD values across the entire range of frequency and severity combinations, with the patterning described for

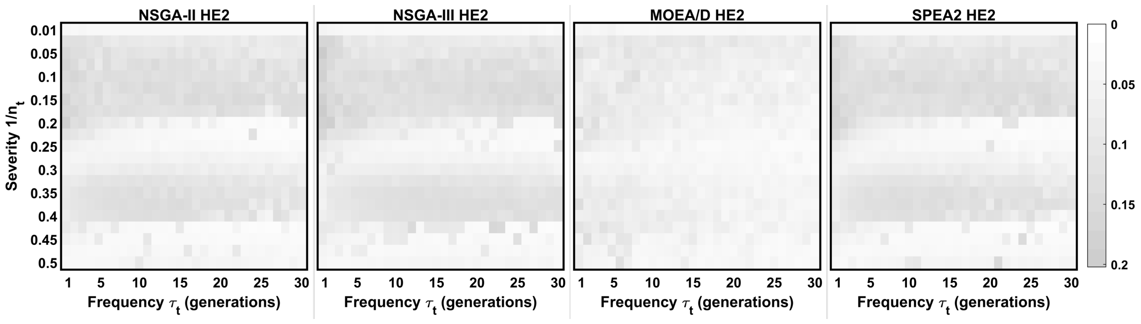


Figure 3.10: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for the HE2 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance.

the other problem still faintly visible, meaning that this algorithm can better handle these combinations too. There is a notable absence of the typical triangular motif seen in the other problems where the lowest frequency and lowest severity instances are more easily handled. Instead, the alternative patterning in the HE1 and HE2 problems implies both that algorithms differently handle changes in the frequency and severity parameters for these problems and that the problems themselves change unexpectedly with the typical increases in frequency and severity. For HE2, similar to the observations for JY9 in Figure 3.8, the performance is similar for most frequency values, meaning that the severity is the dominant parameter controlling algorithm performance on these problems.

Additional Examples - dMOP1, dMOP2 & ZJZ problems

The dMOP1, dMOP2 [76] and ZJZ [79] problems' performance heatmaps are given in Figure 3.11. The results for dMOP1 appear similar to those for JY3, JY5 & JY8 given in Figure 3.7. The majority of the combinations space contains zero or near-zero HVD values for all algorithms, meaning that most instance of this problem can effectively be handled by the MOEAs. Only at the highest frequency of changes is there a degradation of performance; NSGA-III tolerates the faster changes least well, with NSGA-II & SPEA2 still achieving lower HVD values compared to NSGA-III. The MOEA/D algorithm performs best, with all combinations of frequency and severity posing little challenge.

On dMOP2, there is the characteristic triangular patterning seen for previous problems (FDA1, FDA3, JY1 & JY2) albeit with less capability at the lower frequencies. As observed for many of the previous problems, NSGA-III has the smallest area of zero-HVD coverage, NSGA-II and SPEA2 have very simlar profiles and MOEA/D has some additional better observed performance at lower frequencies for higher severity changes. Many of the instances generated by combining different frequency and severity values for this problem are challenging for the MOEAs examined.

The ZJZ problem poses a greater challenge to the algorithms, with generally poor hypervolume attainment (and therefore large HVD values) across the combination space.

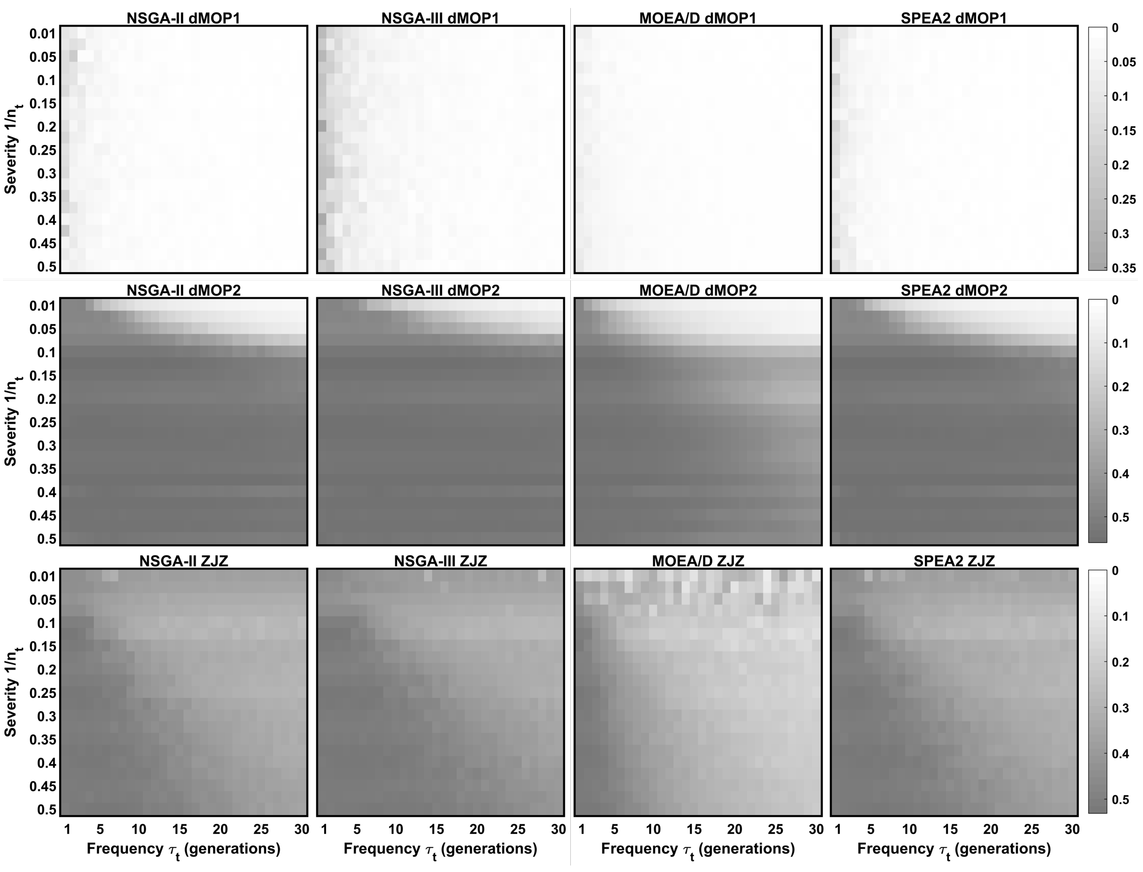


Figure 3.11: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for the (top-row) dMOP1, (middle-row) dMOP2 and (bottom-row) ZJZ problems. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance.

The triangular pattern is visible, however the two distinct regions are poor performance and worse performance rather than a zero-HVD region. The NSGA-II, NSGA-III and SPEA2 algorithms achieve very similar results across the frequency and severity ranges. MOEA/D has some non-uniform better performance for the lower severity changes across different frequencies as well as minor improvements in HVD values across the entire range of both parameters. The ZJZ problem, like the JY6 & JY9 problems, represents a challenging instance space that cannot be effectively handled by unmodified MOEAs and therefore requires the design of dynamic response mechanisms in achieve good performance.

Highlighting Differential Capability of MOEAs - DIMP1 & DIMP2 problems

The DIMP1 & DIMP2 [78] problem performance heatmaps are shown in Figure 3.12. There is similar patterning for DIMP1 across the algorithms as for FDA1 & FDA3; the triangular region of zero-HVD values extending from the lowest frequency, lowest severity combination. Furthermore the observed algorithm differences are reflected again in the results for this problem; MOEA/D has increased good performance coverage of the

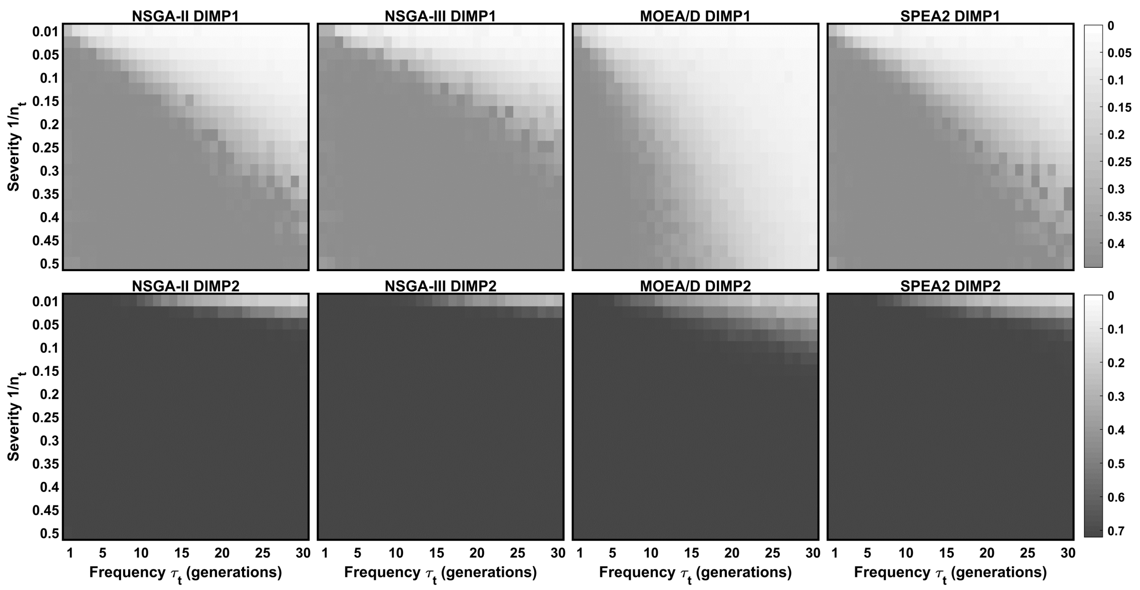


Figure 3.12: Heatmaps of hypervolume difference (HVD) for NSGA-II, NSGA-III, MOEA/D and SPEA2 across severity and frequency combinations for the (top-row) DIMP1 and (bottom-row) DIMP2 problems. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance.

combination space, NSGA-II & SPEA2 perform similarly to each other across the frequency and severity combinations, both with visibly better coverage of the space than the NSGA-III algorithm. There is sporadic poor performance on the diagonal boundary for the dominance-based algorithms, interrupting the smooth gradient of performance improvement visible in the JY1 & JY2 problems. These algorithm differences in the frequency and severity combination space are persistent across many of the problems examined so far, strengthening the necessity of careful selection of these parameters depending on the suite of problems.

The DIMP2 problem is another example of a challenging frequency-severity combination space for the MOEAs. The HVD values are high across the majority combinations for all the algorithms. Only on the lowest severity changes and at low frequencies do NSGA-II & SPEA2 achieve any level of hypervolume attainment. NSGA-III has an even smaller range of capability, with no zero-HVD values achieved for any combination. MOEA/D, as in previous problems, achieves minor improvements in performance at lower frequencies for higher severity of changes.

3.6.2 Comparison of Problems Using Sample of Literature Parameter Values

Showcasing the performance of the MOEAs across the combination space on a problem-by-problem basis provides an overview of the challenging and easy-handled instances and the importance of careful selection of parameters. However, examining a sample of problems with respect to specific values of frequency an severity used in the literature provides a contextually grounded example to support the conclusions drawn from the results presented.

For a selection of the presented problems, a comparison of the combination spaces, highlighted with specific combinations of frequency and severity are examined to illustrate the potential weakness in conclusions based on uninformed selection of these parameters.

Figure 3.13 shows all the previously analysed DMOP benchmarks for the NSGA-II algorithm with commonly used frequency and severity parameter combinations from the literature marked on each heatmap. The crossed correspond to combinations of $n_t - -\tau_t$ of 10-10, 10-20, 20-5 & 5-10. The previous sections drew attention to the necessity of careful selection of frequency and severity parameters due to the potential for misleading conclusions to be drawn. Here to clarify further, a direct comparison across the DMOP benchmarks with marked combinations illustrates this key point. Additionally, these conclusions apply specifically to the problems examined, but can apply more generally to the wider domain.

For the JY3, JY5, JY8 & DMOP1 problems, since the examined range of frequency-severity combinations poses little challenge for the basic NSGA-II algorithm, using the marked frequency and severity parameter values will provide little insight into the benefit of any novel strategy. It is expected that significant improvements from a novel algorithm will be limited.

The HE1 & HE2 problems illustrate that the impacts of frequency and severity can be complex and non-intuitive. Selection of frequency and severity parameter combinations in experiments without understanding their potential impacts on the problem can lead to less well supported conclusions. Since the NSGA-II algorithm can effectively cope with some regions of the instance space created by the frequency and severity combinations, careful selection of these parameters or an informed experimental strategy is required.

The FDA1, FDA3, JY1, JY2 & DIMP1 problems illustrate cases where some of the commonly used literature values for severity and frequency result in instances that can easily be handled by simple MOEAs and others create instances that cannot. Conclusions should not be made in comparison to the NSGA-II algorithm on an easily-handled instance (the blue marker); the novel algorithm does not improve the overall coverage of the instance space created by frequency-severity combinations. Instead this combination

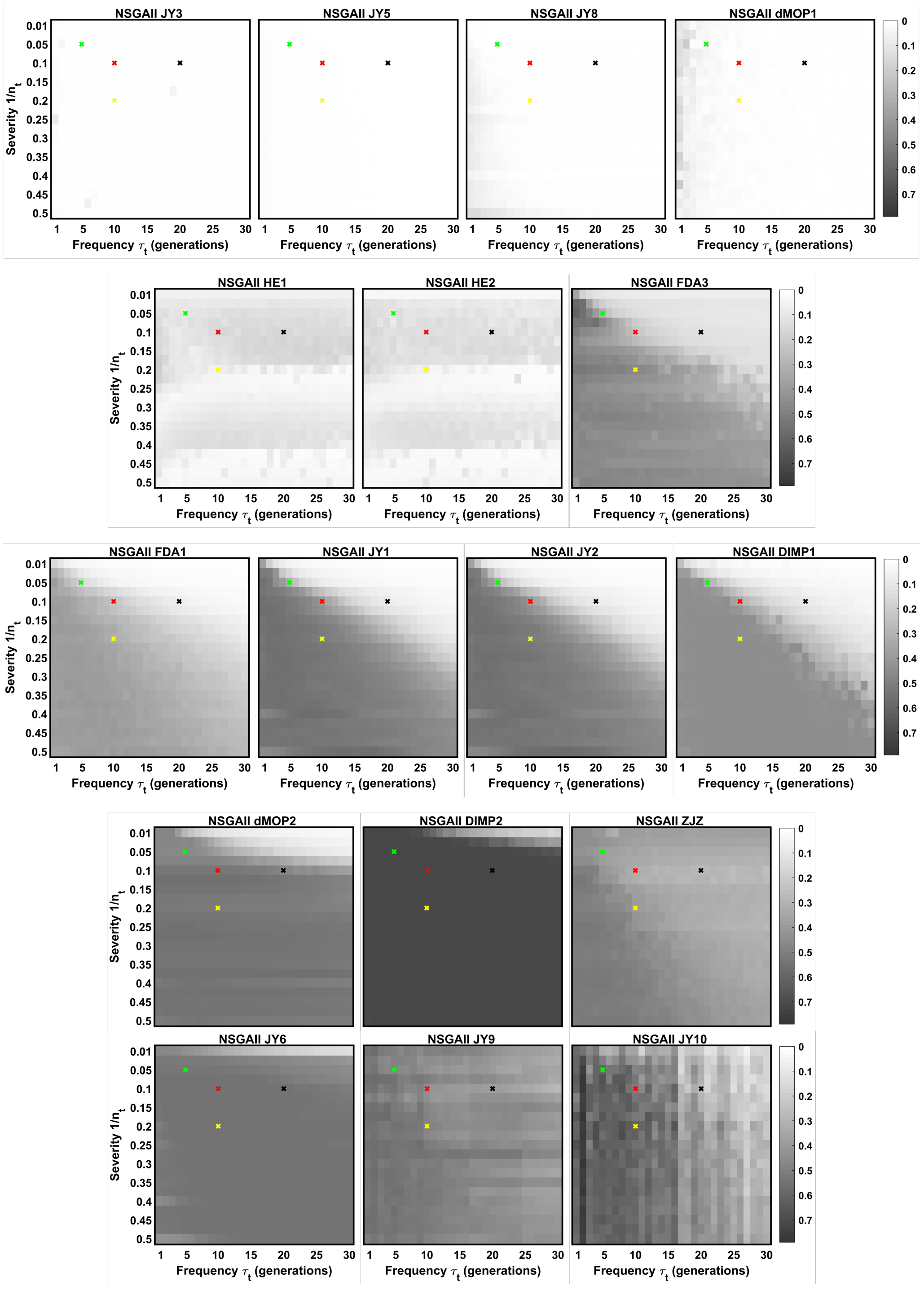


Figure 3.13: Frequency-severity combination HVD performance heatmaps for the NSGA-II algorithm on 17 DMOP benchmarks. The crosses on each subplot correspond to the example n_t–τ_t combinations discussed in the text: 10-10 (red), 10-20 (blue), 20-5 (green) & 5-10 (yellow).

can serve as a verification of basic capability of a new method, whilst experiments can focus on the regions of the combination instance space that cannot effectively be solved using unmodified MOEAs (the yellow marker). The instances that lie just beyond the boundary of MOEA capability (green and red markers) may provide the easiest targets for improvement using engineered response mechanisms.

When using the DMOP2, DIMP2, ZJZ, JY6, JY9 & JY10 problems the extent of the NSGA-II algorithm's good performance, as an indicator of capability (and more generally, that of the other examined MOEAs), is limited. For any of these problems, the indicated combinations are not effectively handled and therefore provide a clear baseline for comparison where any improvement over the HVD values achieved by NSGA-II is useful. The majority of the examined instance space generated by different frequency-severity combinations for these problems cannot generally be handled by static MOEAs with no special mechanisms. It is these problems, together with the instances from other DMOP benchmarks that are not effectively solved by unmodified MOEAs that should be the focus of algorithm development for improving performance. The problems and problem instances that can effectively be handled by generic MOEAs should be trivial for a purpose-built DMOEA and are useful only in verifying this assumption.

The heatmaps illustrated in Figure 3.13 also serve to illustrate further the complex impacts that frequency and severity parameters have on the DMOP benchmarks employed to enable quantification of algorithm performance. This can be explained by observing a particular combination of frequency and severity, for example $n_t = 10$, $\tau_t = 20$, (the combination first proposed by Farina et al. [28]) shown by the blue marker in the heatmaps. Across the examined DMOP benchmarks, this single combination provides a simple MOEA baseline comparison algorithm with both easily handled and difficult problem instances and comparison with a novel strategy may yield improvements for the difficult instances and similar performance for the instances the MOEA finds trivial. This describes the standard practice of many experimental protocols in DMO studies, however comparison using a single frequency-severity instance, or a limited range of instances, in a space shown to be richly dependent on the frequency and severity parameters limits the applicability of any conclusions that follow such narrow methodology. Naturally exhaustive experimentation is infeasible in many cases, however careful and informed selection of a range of combinations, specific to each problem, for which the performance of a baseline comparison algorithm is known can help to clarify and generalise the conclusions made for the performance of a novel algorithm.

3.6.3 Comparison of Dynamic Responses

The choice of the baseline algorithm impacts the outcomes and conclusions. As previously mentioned, many works use static MOEA algorithms in comparison to novel proposed algorithms to provide context to achieved performance measurements. Further to this, the augmentation of these unmodified MOEAs with a random restart method serves as a common baseline for comparison.

Augmenting MOEAs with simple responses to dynamic changes is also commonly employed [29], [53], [63], [72], [167] and can serve to provide context for the performance of more complex novel mechanisms. Here we compare the basic MOEAs with

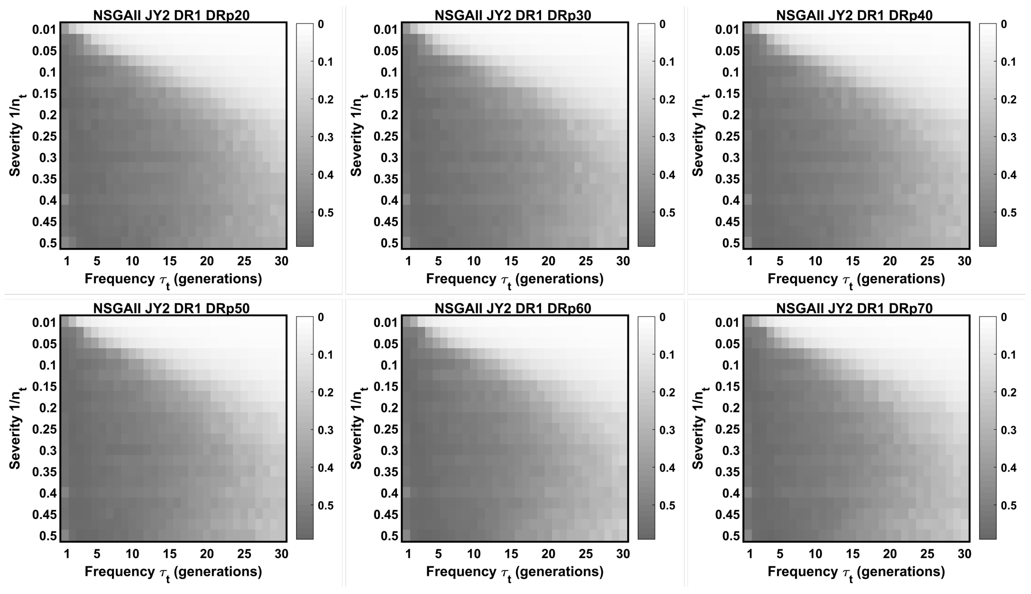


Figure 3.14: Frequency-severity combination HVD performance heatmaps illustrating the impact on hypervolume attainment of NSGA-II on the JY2 problem when using different percentages of solution replacement by the DR1 response. Subplot titles indicate this percentage ranging from 20% to 70%.

versions that replace a proportion of the offspring population with mutated solutions (Dynamic Response 1 - DR1); or with randomly generated solutions (DR2); or replace the entire set of solutions with a randomly generated populations (DR3) also known as a random-restart or random reinitialization method. The DR3 mechanism is commonly used as a baseline comparison method, however the results presented here show the ineffectiveness of this compared with using no triggered response mechanism at all. We posit here that it should not be used as part of a baseline comparison method unless there is a specific justification or information that suggests it provides relevant contrast.

The percentage of the population replaced with random or mutated solutions in DR1 and DR2 appears to not have a significant impact on the performance of these methods. Figure 3.14 shows no visible difference in coverage of the dynamic instance space between percentages of 20 and 70.

Figure 3.15 illustrates the impacts the basic dynamic response mechanisms have on performance across the frequency-severity combination space, compared with the unmodified MOEA algorithms (DR0). Each row of subplots corresponds to one of the MOEAs whilst each column corresponds to a different dynamic response mechanism. Between the DR0, DR1 & DR2 mechanisms there are very minimal differences for all of the algorithms. However, when compared to the random restart method (right-most column of heatmaps in Figure 3.15) there is a clear difference in the performance patterning of HVD measurements across the frequency and severity combination space. Poor performance is seen at high frequencies regardless of the severity of change.

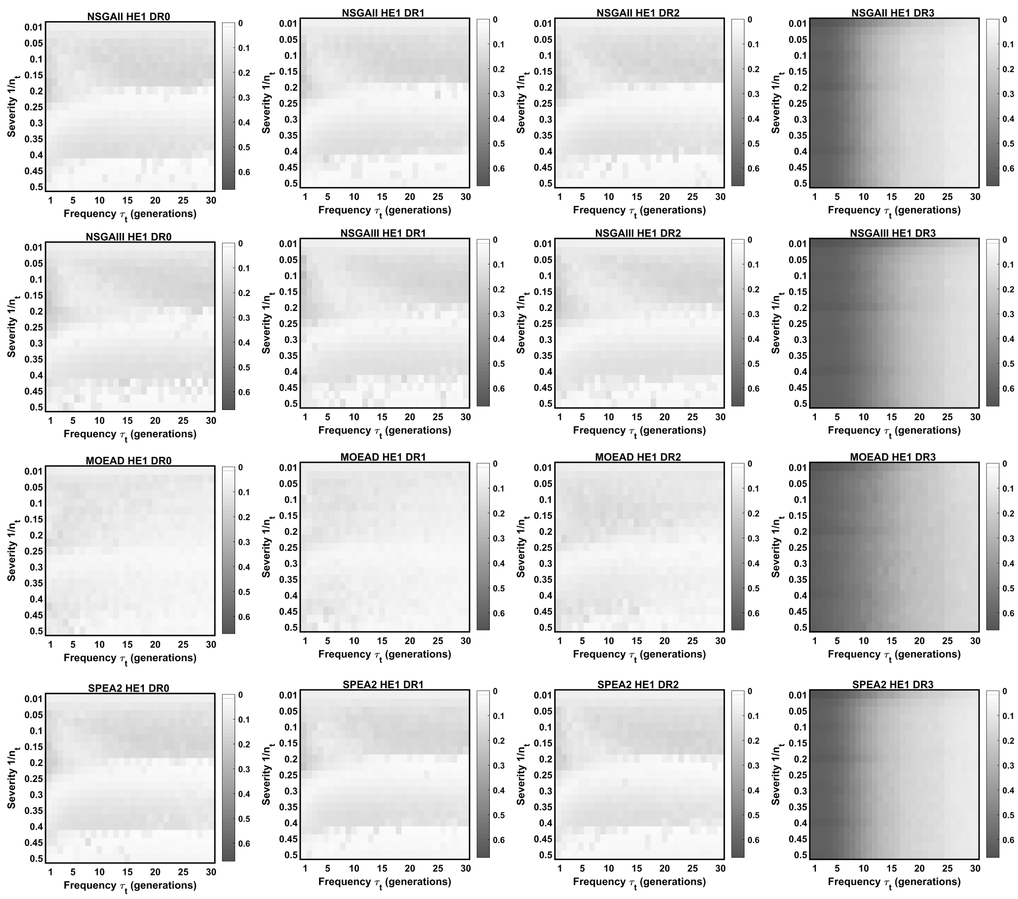


Figure 3.15: Heatmaps of hypervolume difference (HVD) for the four MOEAs (rows - NSGA-II, NSGA-III, MOEA/D & SPEA2) with different simple dynamic response mechanisms across the severity and frequency combinations for the HE1 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. (left column) **DR0** - no mechanism, default algorithm operation; (centre-left column) **DR1** - replace 20% of offspring in generation of change with randomly generated solutions (as in DNSGA-II-A [141]); (centre-right column) **DR2** - as in DR1, except solutions are generated through mutation (as in DNSGA-II-B [141]); (right column) **DR3** - random restart method, entire population is reinitialized with randomly generated solutions after a change event.

Interestingly, the performance of the NSGA-II and SPEA2 algorithms is improved by the random restart method in some cases. In the low frequency regions between the poor performance severity windows for the unmodified algorithms, the performance when using the random-restart method is better (lower HVD values are achieved). This may highlight that these algorithms stagnate on the problem for certain severity-frequency combinations. For the other algorithms, the unmodified algorithms or the solution addition responses provide better performance than random re-initialization after a change.

Generally HVD values when using the random-restart method on any algorithm results in a frequency-dependent performance patterning, independent of severity of change. When observing the results of each algorithm and response mechanism on a different problem, such as the JY1 benchmark in Figure 3.16 further insights can be gleaned. In the low frequency, high severity region of the combination space the random restart method can provide minor improvements to algorithm performance, but the benefit is limited by

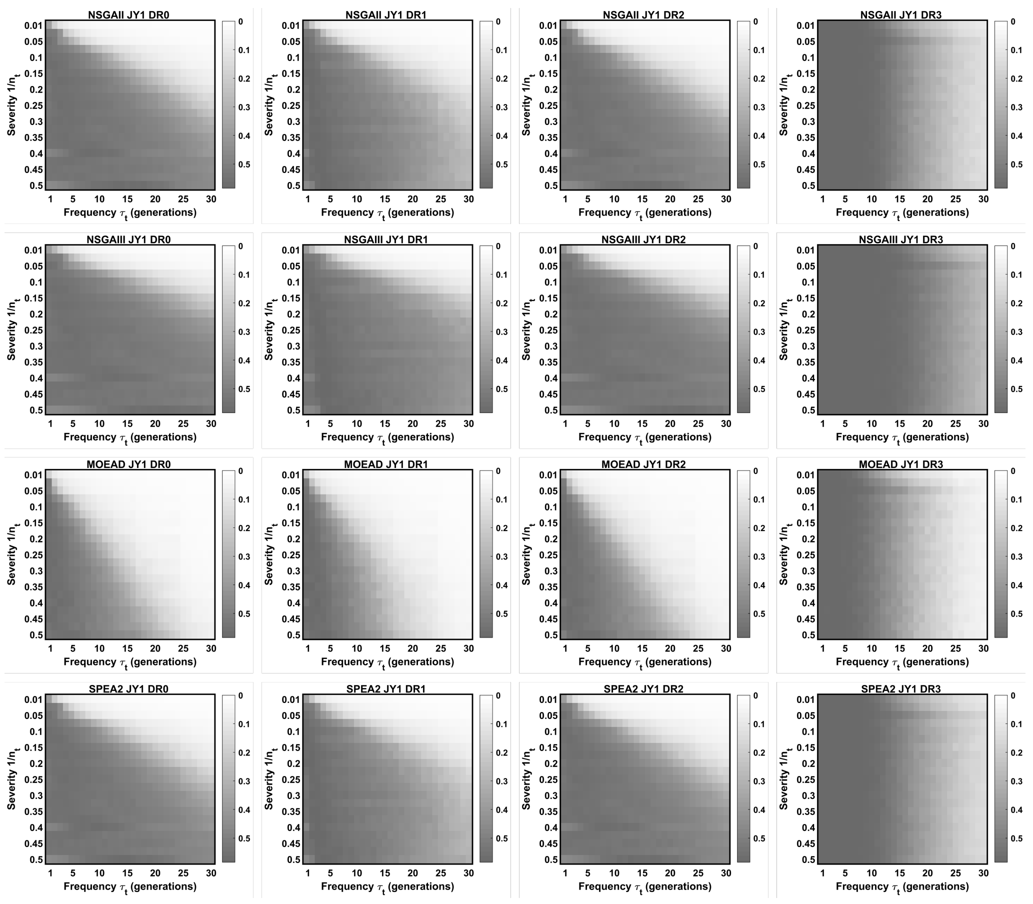


Figure 3.16: Heatmaps of hypervolume difference (HVD) for the four MOEAs (rows - NSGA-II, NSGA-III, MOEA/D & SPEA2) with different simple dynamic response mechanisms across the severity and frequency combinations for the JY1 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. (left column) **DR0** - no mechanism, default algorithm operation; (centre-left column) **DR1** - replace 20% of offspring in generation of change with randomly generated solutions (as in DNSGA-II-A [141]); (centre-right column) **DR2** - as in DR1, except solutions are generated through mutation (as in DNSGA-II-B [141]); (right column) **DR3** - random restart method, entire population is reinitialized with randomly generated solutions after a change event.

the increase in the frequency of changes. This improvement is likely due to the converged population of solutions lacking the diversity to effectively succeed in the post-change problem environment after a high-severity change. This means that the diversity introduced by the addition of random solutions can improve performance in the high-severity, low-frequency region of the instance combination space. The DR1 patterning (middle-left column of heatmaps) features some of this improvement compared to the DR0 and DR2 responses for the NSGA-II, NSGA-III & SPEA2 algorithms. For MOEA/D the benefit of DR3 is less clear compared to the unmodified algorithm, however DR1, the addition of randomly generated solutions shows a clear improvement in good performance coverage.

Building on these observations for the random restart method and noting that the MOEA/D algorithm receives minimal benefits in terms of achieving better performance in the frequency-severity combination instance space, Figures 3.17 & 3.18 illustrate the benefits of DR0 and DR3 for all of the DMOP benchmarks previously examined.

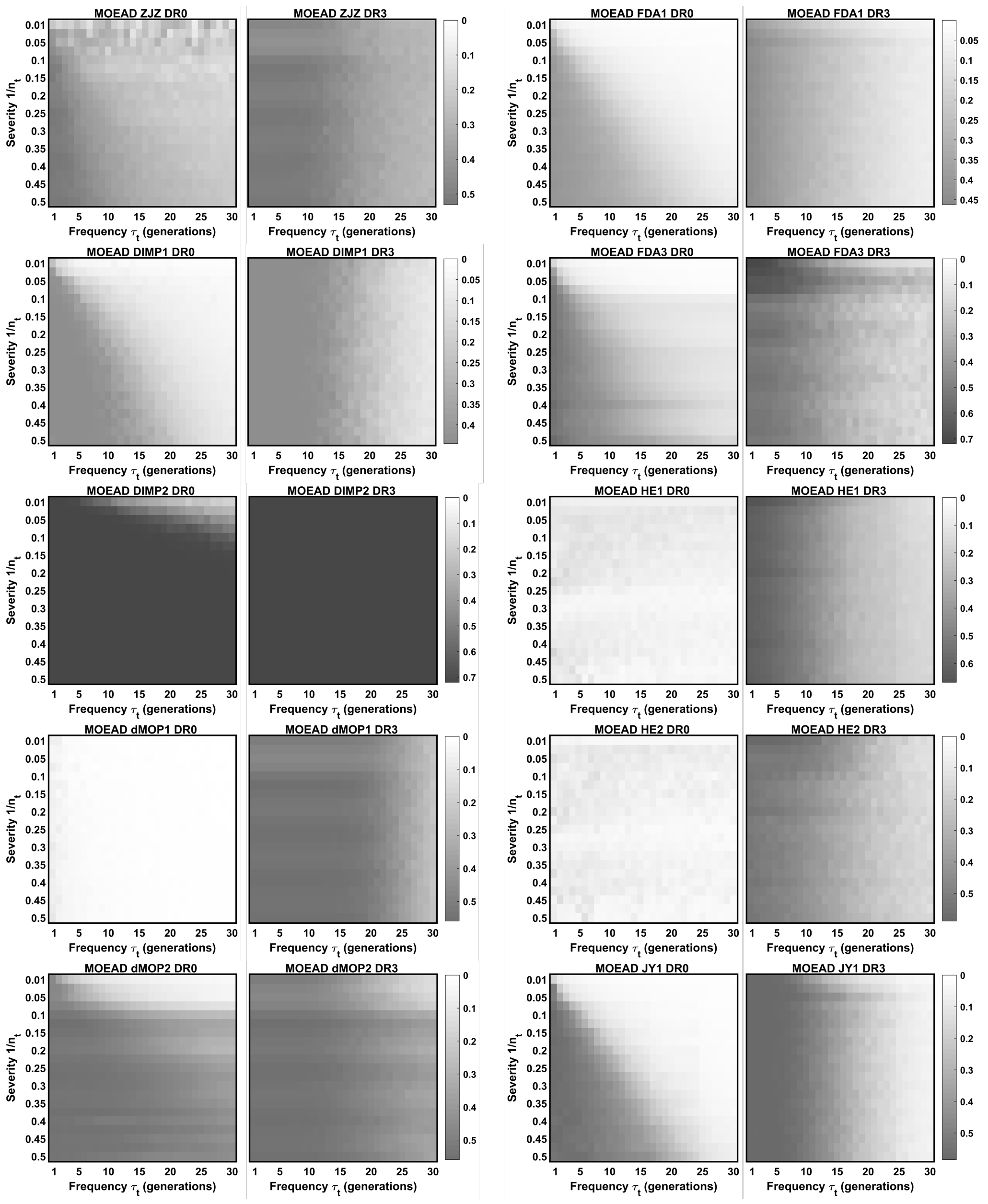


Figure 3.17: Heatmaps of hypervolume difference (HVD) for the MOEA/D algorithm with two different dynamic response mechanisms across 10 of the DMOP benchmarks. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. Each pair of heatmaps comprises a comparison of the performance of the unmodified MOEA/D algorithm (DR0) with the MOEA/D plus random restart (DR3).

Across every pairwise comparison in Figures 3.17 & 3.18, the unmodified algorithm, DR0, shows a greater coverage of the space compared with the algorithm modified with the random restart response. The improvement in the high-severity, low-frequency region does not out-compete the default algorithm's HVD achievement.

Figure 3.19 shows the heatmaps NSGA-II with DR0 & DR3 for the JY1 problem with the commonly used literature values overlaid. We see that even in the region where the minimal improvements can be gained by randomly restarting (bottom-right sector,

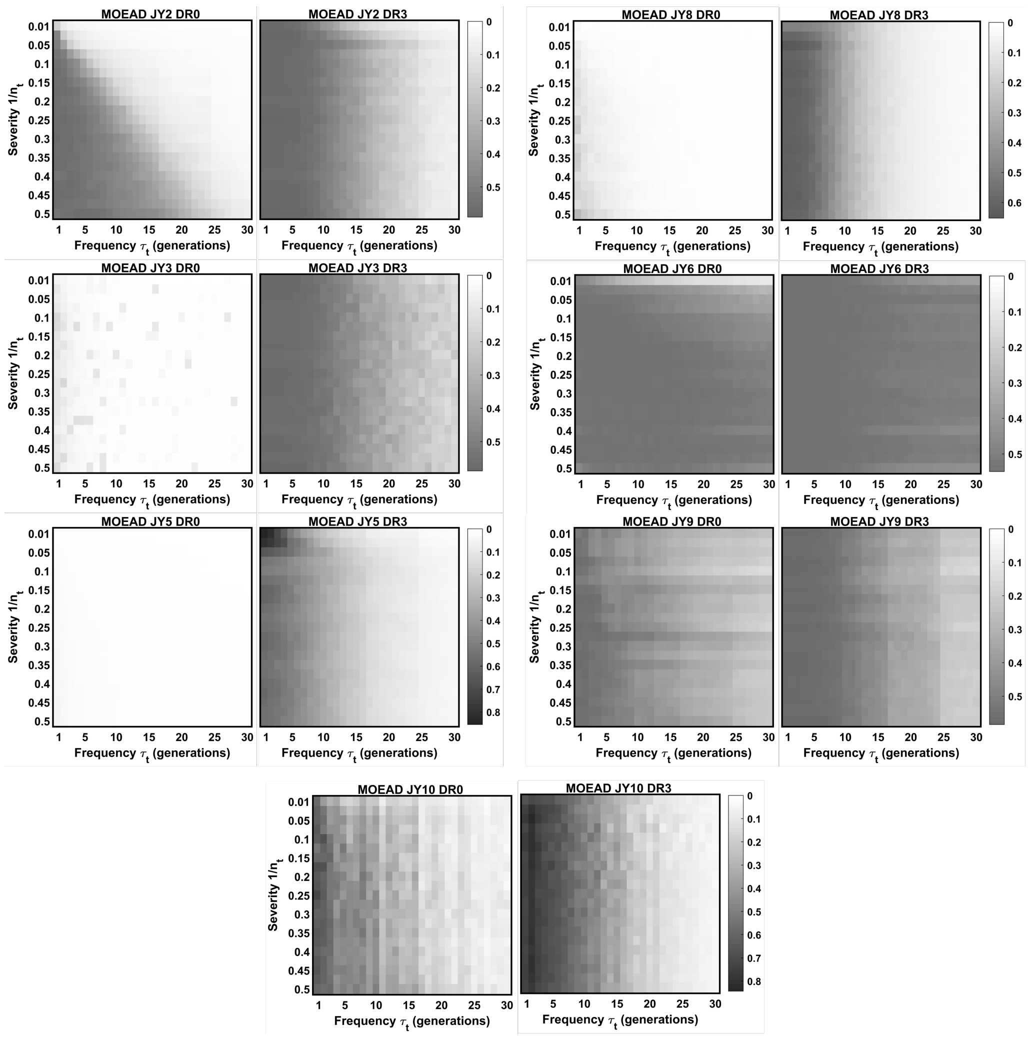


Figure 3.18: Heatmaps of hypervolume difference (HVD) for the MOEA/D algorithm with two different dynamic response mechanisms across the remaining seven DMOP benchmarks. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. Each pair of heatmaps comprises a comparison of the performance of the unmodified MOEA/D algorithm (DR0) with the MOEA/D plus random restart (DR3).

low frequency, high severity changes), the usage of these parameter combinations is not prevalent in the literature. Therefore the previous usage in the literature as a baseline comparison method is unjustified and should have been substituted for the unmodified algorithm. It could be argued that the random-restart method represents a simple response strategy with expected poor performance in many cases, however using the unmodified algorithm is similarly intuitive. However, as this provides good results across many of the commonly used frequency and severity parameter combinations (as shown in the DR0 heatmap of Figure 3.19) the random restart method would provide a novel algorithm with a better performance profile, depending on the benchmark problems. Deliberate usage of a poor performance comparison algorithm, whilst not incorrect, may be misleading in terms of the magnitude of performance gains and the true benefits of a proposed novel method.

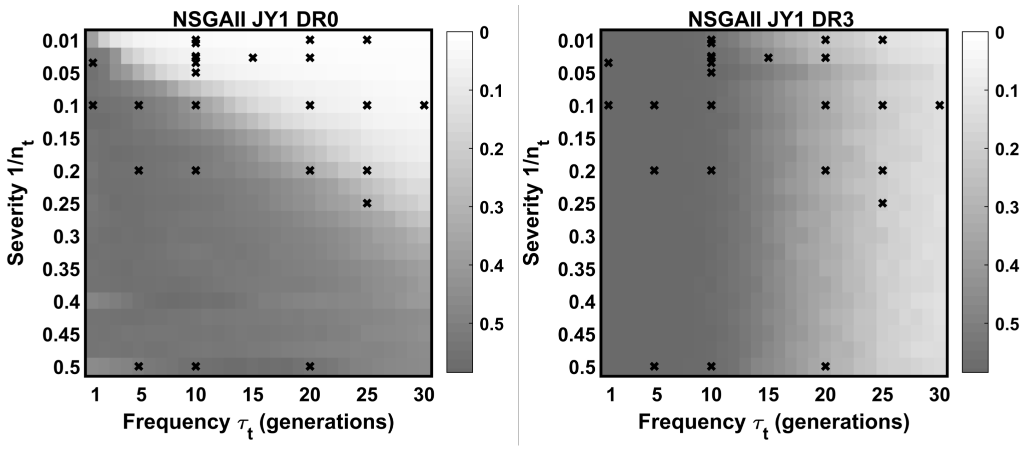


Figure 3.19: Heatmaps of hypervolume difference (HVD) for the NSGA-II algorithm with two different dynamic response mechanisms (DR0 & DR3) on the JY1 problem. Zero values (white cells) indicate perfect hypervolume attainment, darker values indicate worse performance. The commonly used combinations of frequency and severity parameters from the literature, as displayed in Figure 3.1, are overlaid on the heatmaps to highlight the inconsistency of the instance selection in previous experiments as well as the relative unsuitability of the random-restart response compared with the unmodified algorithm.

In summary, the random restart response has very limited benefits in a specific region of the frequency-severity combination space and is dependent on the base optimization algorithm. Generally, the usage of the random restart mechanism is not recommended unless it can be justified empirically or evidenced through preliminary experimentation. Based on the previously employed frequency and severity parameter combinations from the literature, since unmodified MOEAs provide competitive performance in many cases, their 'footprint' in the combination space provides a more useful baseline comparison than a random restart method. Basic capability testing of any novel method can be compared on problems and instances with frequency and severity parameters that unmodified MOEAs can easily handle. Comparison on multiple instances where MOEAs struggle provides a more complete picture of an algorithm's ability to handle a variety of instances of dynamic benchmarks, beyond just a single combination of the frequency and severity parameters.

3.6.4 Recommendations

The results presented highlight that benchmark DMOPs can have the same frequency and severity parameter values but provide different challenges to the MOEAs. Therefore recommendations are provided in Table 3.3 for groups of problems distilled from similar observed performance of the MOEAs across the entire examined frequency-severity combination space. In hopes of posterity, these recommendations are provided for researchers that may use the benchmarks examined in this work, such that a complete picture of performance across the range of possible frequency-severity instances may be established. Such practice provides greater insight and more meaningful contributions than testing on a single or limited range of parameter combinations.

Table 3.3: Suggested severity and frequency combinations based on the summary of comprehensive combination testing. Combinations marked with an asterisk (*) are not challenging for the MOEA/D algorithm, but are for the NSGA-II, NSGA-III & SPEA2 algorithms. Combinations marked with a dagger (†) are only easily handled by the MOEA/D algorithm. These conditions should be observed depending on the selected baseline comparison algorithm.

Problem Group	Baseline Combinations ($n_t - \tau_t$)	Challenging Combinations ($n_t - \tau_t$)
JY3, JY5, JY8, DMOP1	10–10, 10–20, 4–10, 4–20, 2–10, 2–20	–
HE1, HE2	10–5, 10–10, 10–20, 10–30, 4–5, 4–10, 4–20, 4–30, 2–5, 2–10, 2–20, 2–30	–
FDA1, FDA3, JY1, JY2, DIMP1	20–10, 20–25 40–20	20–20*, 10–20*, 10–10*, 5–20*, 2.85–20* 10–5, 5–5, 5–2, 2.85–10, 2.85–5, 2–5, 2–10, 2–20
DIMP2, JY6, JY9, DMOP2, ZJZ	100-30, 40-30	10–20, 10–10, 10–5, 4–20, 4–10, 4–5, 2–20, 2–10, 2–5
JY10	40-30†, 10-30†, 4-30†, 2-30†	10–20, 10–10, 10–5, 4–20, 4–10, 4–5, 2–20, 2–10, 2–5

The HE1 & HE2 problems in Table 3.3 do not have recommended challenging combinations as the patterning in the frequency-severity combination space is non-uniform. Further investigation should be conducted to establish the impacts of severity and frequency that lead to the observed trends and to find appropriate recommendations for examining these dynamic instances in future.

Recommendations for static MOEAs as baseline algorithms

The performance of the MOEAs across different frequency and severity combinations showcased in these results highlights several key points. Firstly, unmodified MOEAs are capable of handling some of the dynamic instances of DMOP benchmarks that have been examined in the literature. Secondly, there are intuitive differences in the ability of the MOEAs but establishment of the baseline capability of these algorithms provides a foundation for meaningful progress and informed design of experiments going forward. Any of the algorithms investigated here can provide a reasonable baseline comparison algorithm without the addition of a random-restart mechanism after each change. However, simple modification of algorithms to include random solutions after a change event may prove useful in some cases (lower frequency, higher severity instances) but unmodified algorithms have been shown to be effective across many of the commonly used frequency-severity instances. From the presented results, the importance of informed selection of frequency and severity parameters is compounded by choosing the correct baseline algorithm, for which an understanding of its capability is essential.

There are many other MOEA algorithms that could be used in comparison such as Indicator-based Evolutionary Algorithm (IBEA) [296], the Pareto Archived Evolution

Strategy (PAES) [297], [298] or an implementation of Cooperative-Competitive principles in Evolutionary Algorithms (CoEA)[76]. Adopting any baseline comparison algorithm should be done so with knowledge of its capability and limitations in terms of the frequency-severity combination instances of DMOP benchmarks.

Alternatively, a clearer or more specific conditioning of conclusions on the exact frequency-severity parameter combinations used and on the benchmark problems examined can achieve similar clarity. However, this may reveal the narrow scope and limited relevance of conclusions if the existing approach to frequency-severity parameter selection is maintained.

The parameters of the MOEAs themselves contribute to their ability to handle the dynamic instances, for the cases examined here a combination of popular usage (e.g. the number of decision variables in each problem, $n = 20$) or the default values as prescribed in PlatEMO (e.g. the mutation and crossover frequencies). To promote reproducibility, in addition to the provision of the DPTP code, all algorithm parameters are provided in Table 3.1.

Recommendations for Design of Experiments

Generally, the more of the frequency–severity combination space is covered in experimentation, the greater the strength of conclusions for the problems considered. Any performance results for novel algorithms should condition their conclusions based on the parameter combinations examined. Alternatively, an informed and stratified sampling of the frequency–severity space or use of the suggested combinations in Table 3.3 will ensure a reasonable amount of information is obtained for each problem. A more informative and parsimonious approach could involve an iterative active learning method that builds a model in the combination space, using algorithm performance as the driving information for classification of instances with different severity and frequency parameters. Automated selection of the parameter combination that reduces uncertainty or strengthens the confidence the most will be recommended for subsequent experimentation. The boundary of algorithm performance, as illustrated by the heatmaps can be determined without the need to examine the entire combination space.

3.7 Limitations and Further Research

The results presented provide a baseline in performance by examining the coverage across the possible dynamic instance space for a number of static MOEAs commonly used as comparison methods. Performance for a number of simple dynamic response mechanisms; the addition or random and mutated solutions, the random re-initialization method and a population prediction strategy (algorithm independent) are also provided here. This provides the basis for further investigation using ‘state of the art’ dynamic MOEAs and methods that have been specifically engineered to cope with problems with dynamic changes. This allows for the practical application of the established baselines, the determination of the quantifiable improvement provided by the DMOEAs and a corroboration of previous conclusions and claims on algorithm performance.

The selection of DMOP benchmarks with provided recommendations in Table 3.3 is limited; there remains a plethora of other DMOP benchmarks that the comprehensive testing methodology can be applied to, particularly some of the more recently proposed suites. Problems with more than two objectives also require frequency-severity recommendations, however the general testing methodology can be extended to these in future work.

3.8 Conclusions

A comprehensive methodology for investigating the impacts of the parameters controlling the frequency and severity of dynamic change events is presented here. Where previous inconsistency in experimental approaches is identified, the proposed framework uses the parameters controlling dynamic changes to provide a more coherent and complete understanding of the possible dynamic instance space that DMOP benchmarks can provide.

A baseline of expectable performance for a number of well-known MOEAs is established across a comprehensive range of frequency and severity combinations for seventeen frequently used and recently defined DMOP benchmarks with a variety of characteristics.

An important comparison is presented for values of frequency and severity used within literature on DMOP benchmarks and the achieved performance of the unmodified MOEAs. This highlights the possible narrowness, scope and relevance of previous conclusions and the importance of correctly conditioning results.

The suitability and veracity of the commonly used post-change 'random-restart' mechanisms is determined through the proposed frequency-severity combination testing. The performance of unmodified MOEAs is better than the random restart method across the considered DMOP benchmarks; providing evidence against its future use as a baseline for comparison.

The additional provision of source code, together with clear explanation of methodology, collection of measurement data and visualization processes highlights the importance of reproducibility in this domain.

Recommendations are provided generally for DMO and specifically for the DMOP benchmarks given the previously employed incoherence in selecting frequency and severity parameters for experiments and the possible impacts on the relevance of previous conclusions. Generally, recommendations include ensuring the informed selection of dynamic instances of DMOP benchmarks for meaningful composition of test instances and the observation of differences in impacts of frequency and severity across different problems. Specifically, for the examined DMOP benchmarks, the challenging and easily-handled frequency-severity combinations are identified based on the performance coverage for the examined MOEAs.

The coherent design of experiments within DMO is important for meaningful progress to occur. Establishing the tools for determining reproducible baselines of performance using static algorithms and identifying improvements in comparison techniques, will enable better targeted development of future experiments and the acceleration of DMO research generally.

Chapter 4

A Realistic Combinatorial Dynamic Multi-Objective Benchmark Problem: The Dynamic Travelling Thief Problem (DTTP)

4.1 Introducing realistic dynamics to the TTP

There exists a multitude of dynamic benchmark problems in the continuous domain, including for multi-objective and complex search landscapes. Many of the more recent contributions in this area also exhibit some direct justification from real world problems. The same progress has been absent from the combinatorial problem domain, with some limited formulations for dynamic multi-objective versions of TSP and KP problems. Realistic features have been incorporated into dynamic vehicle routing problems and dynamic arc-routing problems, however few formulations are justifiably multi-objective. Therefore, a gap remains for a problem with more than one objective that contains dynamic events that can be justified in realistic scenarios. The bi-objective Travelling Thief Problem provides the basis for a number of dynamic formulations to be presented. Drawing on generalizable logistics scenarios, the potential application of this problem framework provides a bridge between previously simplistic mathematical representations and the specific end-use, deployment scenarios. In many cases where the optimal set of solutions is unknown, proposition of novel problems becomes a more complex task; effectively finding good solutions must be combined with meaningfully evaluating their quality. Directed and logical development of methods to find solutions in the dynamic instances is grounded in observation-informed design of response mechanisms. Comprehensive experimentation, sensible comparison, and the explanation of statistically-observant performance reporting ensures the clarity of conclusions for seeding response mechanisms presented here for the dynamic travelling thief problem.

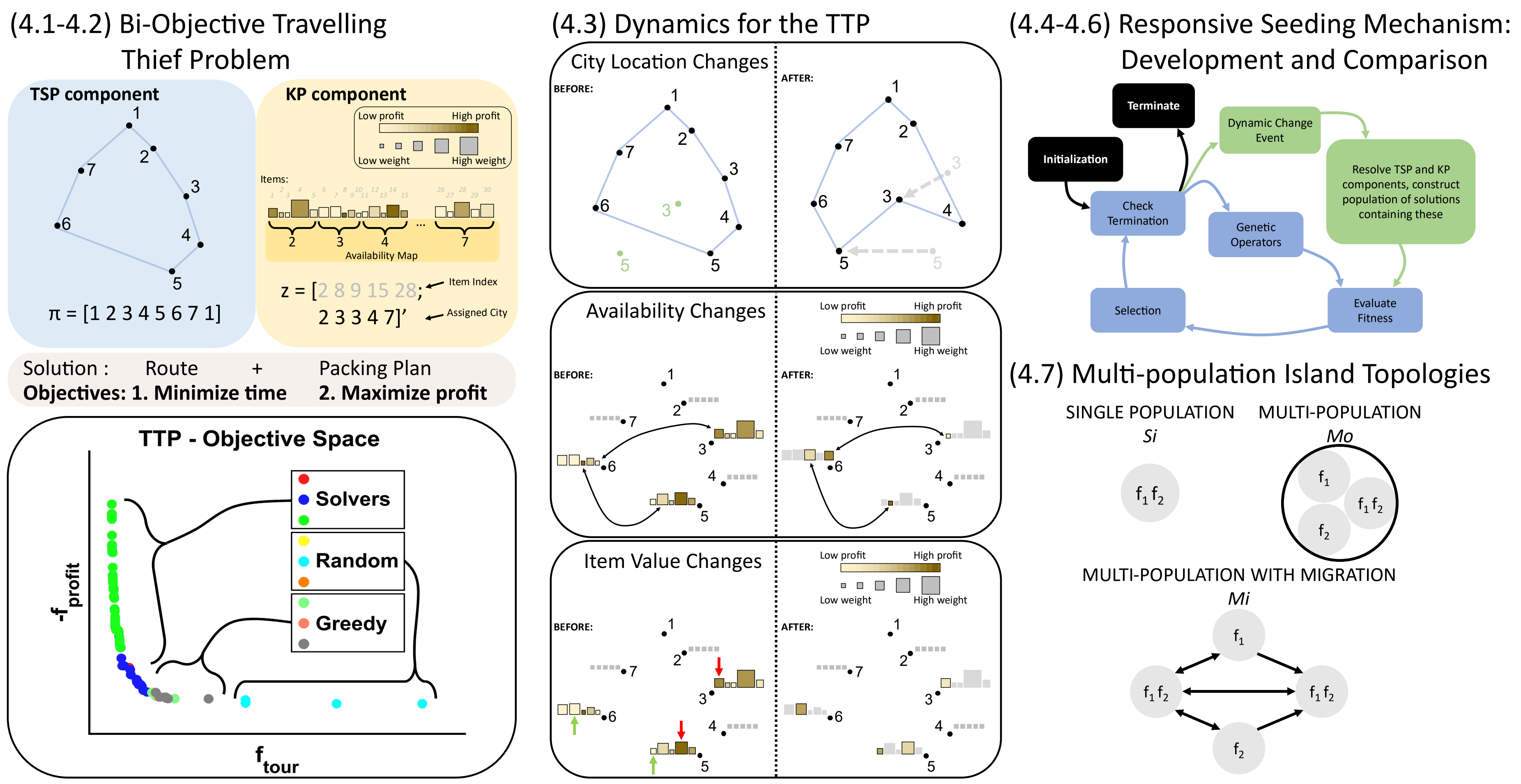


Figure 4.1: Summary diagram providing (left) illustrative context in the form of simple diagrams to explain the Travelling Thief Problem basis and (center & right) accessible depictions of the key contributions: the proposed dynamics for the TTP and the algorithm methods used to improve the finding of solutions in these instances.

The Travelling Thief Problem (TTP) was first introduced nearly a decade ago by Bonyadi et al [137] and has received much attention from the evolutionary computation community. The TTP represents a complex and realistic problem framework, with interconnected components, a complex search landscape and unknown optimal solutions. Considering these characteristics as being indicative of more realistic scenarios, the TTP provides a perfect candidate to develop dynamic multi-objective (in this case bi-objective) optimisation benchmark problems in the combinatorial domain. Current and former examples of this are limited, either in their scope of scenarios and simplicity (dynamic MO-Knapsack) [28] or the realistic justification for incorporating dynamics (dynamic MO-TSP) [96], [101], [182], [299]. Few examples also exist of real world discrete problems that can be generalized into benchmark scenarios. For example the work by Colson et al. [274] considers a specific dynamic power management example.

The majority of DMOP literature considers problems in the continuous domain, with fewer combinatorial problems considered despite the applicability to important sectors such as mobile and network communications, energy and transport. A few works consider the DMO-TSP problem [69], [182] and in [28] a dynamic Knapsack formulation is presented.

The proposal of Dynamic Travelling Thief Problem (DTTP) scenarios provides a bridge between overly simplistic benchmark problems and the innate complexities and specific nature of real world problem formulations. Instead, heuristics and dynamic response strategies can be examined in a complex, but customisable problem environment that is flexible, so to allow experimentation across a diverse set of instances with diverse characteristics and dynamic events.

We therefore propose, and investigate the responses of algorithm approaches to, three formulations of the Bi-objective Dynamic Travelling Thief Problem. These changes occur separately in the locations of the cities, the item-city assignment (known as the availability map) and in the profit values associated with each item. Each of these can be justified in the context of a general logistics optimization problem where a courier or vehicle must collect the items across a graph of cities or locations subject to the objective functions and constraints of the problem. Examples include waste collection, salt gritting/refilling, road sign/traffic cone/apparatus recovery. Being able to track a non-dominated set of high-quality solutions (as the Pareto Front is unknown, we can only provide an approximate set of good solutions) through the dynamic intervals of a problem, enables selection of an actionable solution by a decision maker subject to external information, such as poor weather conditions; extreme traffic, emergencies or road closures; or any prioritization of one objective given unforeseen events.

The dynamics are proposed for the two-objective problem as there are exploitable behaviours linked to initialization that can be observed in the objective space contribution of solutions. Furthermore, combinatorial problems with dynamics and more than one objective function have limited representation in the previous literature. The iterative development of methods to find good solutions to the proposed dynamic TTP instances employs multi-population topologies and migration events for exploitation of the localized evolution possible in the objective space.

4.2 Background – The Travelling Thief Problem

4.2.1 The Bi-objective Travelling Thief Problem

The TTP constitutes a superposition of Travelling Salesperson and Knapsack Problem components, interconnected by the behaviours of the 'thief'. The thief completes a Hamiltonian tour of the cities (x) and must collect items distributed across the cities according to a packing plan (z). The components are connected through knapsack-usage-dependent velocity and time-dependent degradation of item profits. The conflicting objectives are given in Eqn. 4.1 and reflect a minimization of tour-time and maximization of item profit.

$$G(x,z) = \begin{cases} min \quad f(x,z) = \sum_{i=1}^{n-1}(t_{x_i,x_{i+1}}) + t_{x_n,x_1} \\ \qquad\quad x = (x_1,\dots,x_n) \\ max \quad g(x,z) = \sum_{j\in z} p_j Dr^{\left\lceil \frac{T_j}{C} \right\rceil} \end{cases} \tag{4.1}$$

where:

$$t_{x_i,x_{i+1}} = \frac{d_{x_i,x_{i+1}}}{v_c}, v_c = (v_{max} - W_c \frac{v_{max} - v_{min}}{W}) \tag{4.2}$$

and:

$$C = \frac{ln\,(Dr) * E_t}{v_{min} * ln\left(\frac{rl}{u}\right)} \tag{4.3}$$

where $f(x,z)$ is the travel time of the tour accounting for item selection; $g(x,z)$ is the sum of the selected items' profits at the end of the tour; $d_{x_i,x_{i+1}}$ is the Euclidean distance between successive cities in the tour permutation; v_c is the current speed of travel; W_c and W are the current weight and maximum capacity of the thief's knapsack; v_{min} and v_{max} are the minimum and maximum travel velocity (0.1, 1); Dr is the dropping rate (0.9); T_j and p_j are the total time item j is carried during the tour and its profit value respectively. The constant C is calculated using the equation in [137] with $r = 0.45$ so as to generate reproducible results. Here, l and u are the minimum and maximum profit values across all items and E_t represents the shortest inter-city distance in the distance matrix D.

The distribution of items in the *availability map* is defined as part of the problem instance; it contains information about which items are available for selection from each city. A single solution is comprised of a permutation of the city indexes as a tour (x) and a KP solution (y) that respects the capacity constraint W. This is converted into a packing plan (z) based on the availability map. The packing plan is a vector of the same length as the number of items; the index of the selected items contains the city index number if it is part of the solution and zero otherwise. The TTP as shown, provides the basis for the dynamic bi-objective TTP presented herein.

There is a wealth of literature devoted to the TTP on major subjects including theoretical understanding of interconnectedness of the problem components as well as development of exact and approximate methods for its optimization.

Mei et al [300] provides theoretical and empirical research on the interconnectedness of the problem components. As the objective function is not additively separable into the TSP and KP components, a combination of solutions to the TSP and KP components solved in isolation is less effective than consideration of the whole TTP problem, as posited in the definitive work [137].

Faulkner et al. [5] defines approximate heuristic methods, including the S5 heuristic, for solving the single objective TTP that are initialized with tours using the Chained Lin-Kernighan TSP solver [301]. Exact methods have also been employed in [6] to determine the performance, by comparison, to a range of approximate methods applied to TTP problems. Similarly, Dynamic Programming (DP) is used to find an optimal packing plan in the fixed tour scenario of the TTP: the Packing While Travelling (PWT) problem [138], [302], [303].

Many meta-heuristic approaches have also been applied to the TTP. These works use simulated annealing [304] and hill climbing [305] or evolutionary algorithms [300], [306]–[308], including MATLS [309]. Ant Colony Optimization [310], [311], Cooperative Coevolution [300], [312] and Local Search [307] methods have also been applied to the single objective TTP. More information on all these algorithms can be found in [313].

Despite the reported applicable range of TTP instance sizes for EAs and the scale of the comprehensive benchmark set proposed [307], the existing literature is mostly limited to problems with 101 cities or fewer. The work of Mei et al [300], specifically addresses large scale problems with at least 10,000 cities and 1,000,000 items, whilst the ACO-based MMAS approach in [311] is applied to instances with at most 1000 cities and 10,000 items.

Relatively few works address the bi-objective version of the problem [303], [308], [314]–[316]. Blank et al's approach uses solvers combined with low level heuristics to provide solutions to a limited set of TTP instances [308]. The Empirical Attainment Function (EAF) [260], [261] was used to report the 'median Pareto Fronts' for the TTP. Yafrani et al. demonstrated the utility of considering the bi-objective problem and the applicability of found solutions to the single-objective formulation [314] whilst Wu et al. [303] investigated the sequential optimization of a tour then a packing plan for the bi-objective TTP. Chagas et al. [315] investigated the impacts of hyperparameters as well as using optimal solutions to the components of the TSP and KP problems. Chagas & Wagner [316] proposed a weighted-sum approach to modify existing TTP heuristics to provide improved performance of the single-objective problem instances.

Beyond the proposed dynamics for the TTP presented here, there are a number of open questions regarding the existing TTP literature. Attention is focused on the single objective scalarization of the TTP, whilst there remain opportunities to exploit behaviours observed in the objective space of the bi-objective problem in order to provide a diverse set of solutions for any decision maker. Validation of findings on the comprehensive benchmark TTP set [307] for the bi-objective problem and a construction of novel heuristics to exploit the *DropRate* feature remain open research tasks.

4.2.2 The Dynamic Travelling Thief Problem: Existing Work

Studies addressing DTTPs are greatly limited [317], [318] despite the range of possible formulations and utility of generating a realistic and complex dynamic combinatorial problem. An attempt to introduce dynamics into the single-objective formulation of the Travelling Thief Problem was suggested by Sachdeva et al. [318]. The proposed dynamic events were a toggling of cities and items, meaning that after a change event some items cannot be selected or some cities cannot be visited. These definitions lacked contextual justification and comparison with other methods and pose additional issues for comparability; between successive dynamic intervals the total number of items and cities is different.

4.2.3 Dynamic Combinatorial Benchmark Problems

A simplistic multi-objective TSP problem with dynamics was provided by Farina et al. [28]. The proposed problem consists of cities equally spaced on the circumference of a circle. Weights chosen such that the solutions that are circularly connected and those with a maximum number of diametric connections provide the extremes of the non-dominated set in the objective space. The dynamic changes consist of the swapping of cities, either randomly or between ‘sectors’ that divide the overall circle. This provides the template for many experimental benchmarks since, however due to its simplistic nature the potential for meaningful results using this format is limited.

The work of Yang and coauthors [69], [96], [319] examines a Dynamic Multi-objective TSP problem to find the optimal tour across a number of fixed locations (ground cities) and a small number of satellites in orbit which dynamically changing position. The problem is bi-objective with the first objective being the sum of the tour length using a ‘distance matrix’, however, there is limited explanation for the second objective, derived from a ‘cost matrix’ with some objective values being probabilistically inverted from the distance matrix.

Li & Feng [182] describes scenarios of a DMOTSP with a scalable number of objectives. The difference between these objective functions is in the cost matrix used to calculate the fitness of the tour; each objective has its own cost matrix. Three types of dynamics are defined: increases or decreases in the cost between cities as changes in a number of weights in the cost matrices; a change in the number of cities in the problem; and a change in the number of objective functions (similar to [31], the number of objectives changes by 1, in $M \in 2, 3$). Missing is an accompanying justification for how this corresponds to a realistic scenario.

Cordero et al. [97] utilized real world data in the form of (near) real-time data streaming of traffic information for a number of nodes in New York city to inform the dynamic changes to the problem. In addition a synthetic problem is adapted from TSPLIB [98] and in both cases the problems are bi-objective, using distance and travel-time as the optimization goals. The provided results illustrate concave fronts of non-dominated solutions such that long-distance solutions have shorter travel times, whilst the converse (longer travel times are associated with shorter distances). Whilst this geometry may be a result of the traffic present within the graph network, there is little discussion of this and many specific details are missing regarding the dynamic changes in both the synthetic and real-world problem and additionally in the reproducibility of the results.

Liu et al. [101] provides another example of a real world problem scenario framed as a DMOTSP. The task is route planning for low-earth-orbit debris removal using multi-satellite swarms with the objectives, the cost of orbital transfer (change in velocity), the number of satellites required for removal and the priority of the debris item, scalarised to a single objective problem. This means that a non-dominated set of trade-off solutions is not found for the problem and cannot be used to interpret the interaction between the objectives.

Gupta & Nanda [299] offer recent expansion of dynamic TSP to include many objectives. The examined problem is derived from the simplistic 16-city, circular arrangement proposed by Farina et al. and described previously and in contrast to the title and motivations of the paper has only two objectives, with no description of scalability. The proposed prediction-improved dynamic NSGA-III-based algorithm is also applied to continuous benchmark problems from the JY set [86], however again these are described in a bi-objective format with no results presented beyond these cases. In [320] four objectives are proposed for the 16-city problem, with the first two as before and two additional objectives defined for the delivery of ‘letters’ and ‘gifts’.

There are also similar problems in the dynamic formulations of the Capacitated Arc-Routing Problem (DCARP) [321]–[323] as well as in dynamic vehicle routing problems (DVRP) [104], [324], [325] and dynamic pickup and delivery problems (DPDP) [108], [109]. Other examples extend these to realistic scenarios for routing emergency vehicles

[326]. However, across all of these problem types examples of multi-objective instances are limited.

For the dynamic knapsack problem and its multi-objective derivations, there are few examples of useful, specific or widely applicable benchmarks. Farina et al. [28] suggests a generic dynamic knapsack problem with multiple knapsacks representing the different objectives in the problem. A similar instance is suggested for multiple knapsacks by Perry & Hartman [128], however a weighted sum of the reward is taken to give a single fitness value. In terms of real world applications, in [128] the optimization is for the allocation of manufacturing capacity and the work of Ceran [143] extends the dynamic knapsack problem to allocation of renewable energy resources. Fundamental works for the single-objective dynamic knapsack problem can be found in [124] & [327] and additional time-varying knapsack examples can be found in [328], [329].

4.2.4 Dynamic Travelling Thief for Real World Scenarios

There are a number of specific scenarios that fit the schema of the DTTP. Some of which have been examined in static scenarios and some have yet to be addressed. One example is the optimization of routes for gritting vehicles in icy weather, where the items could represent refills of the gritting material. Dynamics most naturally present as extreme or adverse weather conditions, or road closures, both of which could alter the distance matrix. Dynamic updates to the quantity of material or the priority of locations could be represented in the changes in the availability maps or in the item values across the nodes. Similarly, the collection of household or construction waste or recycling materials fit in the model of the DTTP, the vehicles have a limited capacity, may have to service multiple locations or regions and can ascribe profit to the materials for collection such that similar materials could be collected from different sites. As with the previous example, dynamics can present as road closures, adverse weather or problems with accessing a pickup location or additional pickup locations being provided. In emergency scenarios, including for major incidents where multiple injured persons or damaged objects must be serviced simultaneously, the profit values now correspond to a priority and the capacity of the vehicles is intuitive. Dynamics could present in an update of information, such as new reported injuries or a shift in priorities. Considering the recovery of road maintenance equipment, including items such as temporary barriers, traffic cones, stop signals; vehicles will have a limited capacity and may wish to consolidate items of a particular type which could be prescribed in the profit values. These examples demonstrate that there is wide ranging applicability for the DTTP formulations that are presented in the subsequent section. The flexibility of the dynamic components in order to simulate a vast variety of possible realistic scenarios is a key design feature for the proposed dynamics. In all of these scenarios, the automation of these tasks with robots or drones will especially rely on effective provision of an optimal or approximate set of solutions that is dynamically updated as conditions change.

4.2.5 Seeding, Initialization and the use of Multi-population Techniques

Due to the components of the TTP, meaning the TSP and KP parts of a solution, having extensive research and efficient established methodologies, there is the natural incentive to exploit this information for the TTP instances. However, it is noted in [137] and the associated literature mentioned previously, that the combination of solved TSP & KP components does not guarantee good solutions to the TTP instance. The default practice for most evolutionary algorithms and stochastic heuristics is to randomly initialize a solution or population of solutions. However, for the TTP, exploiting the information obtained by solving the TSP and KP components provides additional and valuable information compared with random initialization and is common practice in the TTP literature.

The extent to which the benefits of this different initial solution information impacts the quality and diversity of the terminal solutions obtained is presented within the results of this work. Within a given specified generation budget, there is observable localization of non-dominated solutions in the objective space based on the solution information content (the construction methods) of the initial population of solutions. There is extensive literature regarding the 'seeding' of solutions into population based algorithms for multi-objective problems [4], [330], [331], and based on the localization observations the development of seeding based response mechanism for dynamic instance of the TTP is proposed.

This is further developed using multi-population topologies with biased objectives and migration between populations to evolve a diverse, good-quality set of solutions based on a directed and focused optimization methodologies for the DTTP. There is a plethora of research on applying multi-population approaches to dynamic optimization benchmarks, covering a diverse range of methodologies [174], [195], [332]–[339], however the application to real world problems and industrial optimization tasks [340], [341] and realistic scenarios is limited. In this scenario, the observed localization can be exploited differently across separate populations in a multi-population topology, whilst solution information can be shared via migration events (as in island topologies [342]–[346]) for overall improvements in obtained solutions.

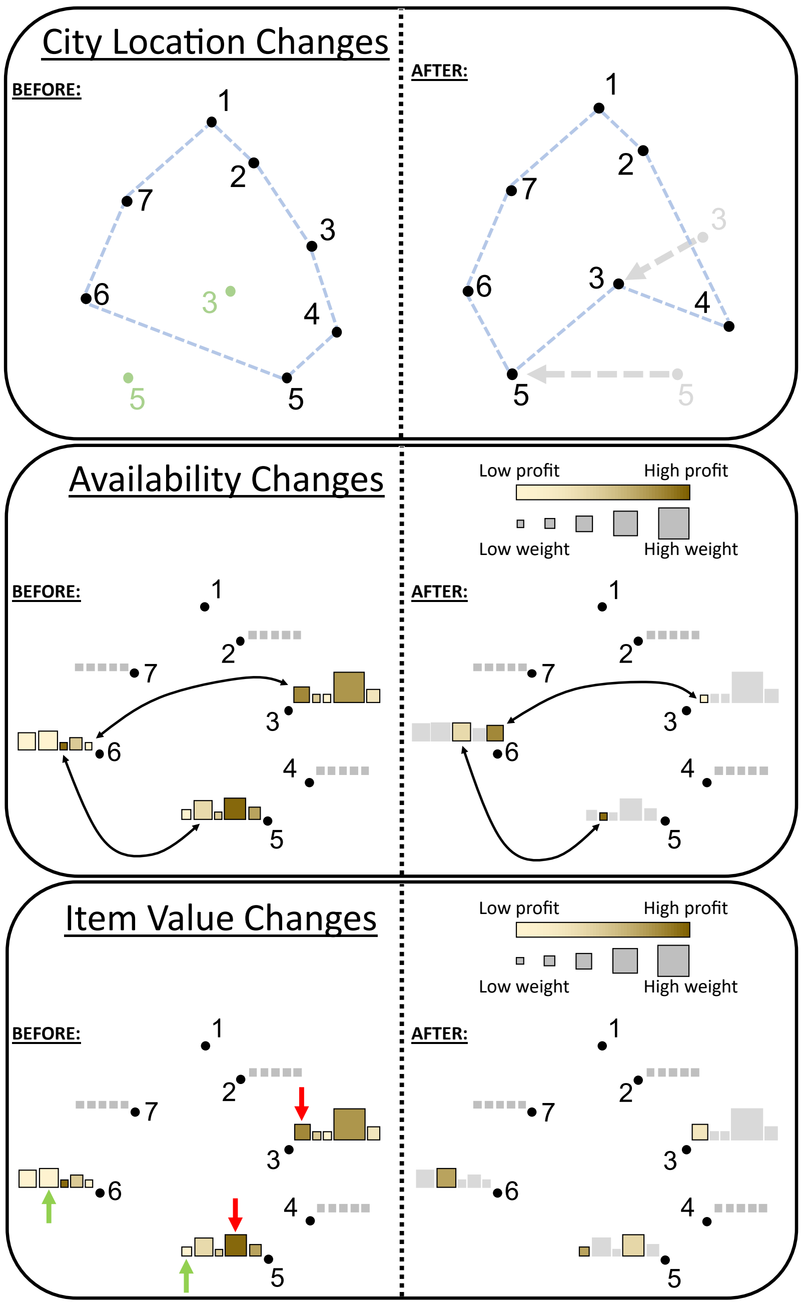


Figure 4.2: Diagrams illustrating the three types of proposed dynamic changes: (top) City location changes, with example routing before and after changes; (centre) Item availability changes for randomly selected items between randomly selected destinations; (bottom) Item value changes, upwards green and downwards red arrows represent an increase and decrease in the profit value of the indicated items in the 'after' panel, as explained in the key.

4.3 Proposed dynamics and problem impacts

The diagrams in Figure 4.2 illustrate the composition of dynamic Travelling Thief Problem instances, beginning at the TSP and KP components, their combination into a TTP instance and the inclusion of dynamic change events of different type. These different types of changes are defined and justified in the context of events that may occur in a general logistics scenario. Including this context in the definitions is important for the applicability of the investigation beyond the experiments conducted here and facilitates a clearer understanding of motivations. The construction of the continuous benchmark problem set by Jiang et al [86] followed a similar protocol for justifying the dynamics based on events observable in real world scenarios.

Three different types of dynamic events are proposed: changes in the locations of a number of cities, changes in the availability map which contains the item-city assignments, and changes in the profit values associated with a number of items.

4.3.1 City Location Change (Loc)

Within the DTSP literature, dynamics are added by the changing locations of cities [243], [347]–[350], adding or removing cities [347] or altering specific distances between cities to simulate traffic in the network [244], [351]. We focus on the first of these for the TTP, opting to use a context-driven justification for the dynamics proposed.

Where a city (x_i) is represented with Cartesian coordinates ($x_{\mathrm{x},i}, x_{\mathrm{y},i}$), and a distance matrix, D is constructed, the row and column elements of D must be updated when a city's location changes. The allowable translation limits for a city's location, the initial range in x and y directions is symmetrically increased by 5% in each direction whilst maintaining non-negative coordinate values (denoted as ϵ_{x}, ϵ_{y}). Given the model in Eqns 4.1, we redefine Eqn. 4.2 to account for the changing city locations.

$$t_{x_i,x_{i+1}} = \frac{\sqrt{(x^{i+1}_{\mathrm{x}(\tau)} - x^{i}_{\mathrm{x}(\tau)})^2 + (x^{i+1}_{\mathrm{y}(\tau)} - x^{i}_{\mathrm{y}(\tau)})^2}}{v_c}$$

The new city locations are calculated via:

$$x^{i}_{\mathrm{x}(\tau+1)} = r_{\mathrm{x}} \quad , \quad x^{i}_{\mathrm{y}(\tau+1)} = r_{\mathrm{y}}$$

where τ is the dynamic interval counter and r_{x}, r_{y} are drawn at random from the respective uniform discrete distributions:

$$r_{\mathrm{x}} \sim unif_{\mathrm{x}}\{\min(\min_{i=1,\ldots,N}(x^{i}_{\mathrm{x},0}) - \epsilon_{\mathrm{x}}, 0), \max_{i=1,\ldots,N}(x^{i}_{\mathrm{x},0}) + \epsilon_{\mathrm{x}}\}$$

$$r_{\mathrm{y}} \sim unif_{\mathrm{y}}\{\min(\min_{i=1,\ldots,N}(x^{i}_{\mathrm{y},0}) - \epsilon_{\mathrm{y}}, 0), \max_{i=1,\ldots,N}(x^{i}_{\mathrm{y},0}) + \epsilon_{\mathrm{y}}\}$$

A number of cities $d_{N,Loc}$ (magnitude of the change) are updated to feasible randomly generated locations and the distance matrix is updated accordingly.

Given a courier that must collect items along its route, a change in a city location can be interpreted as an alternative depot (with the same items) being chosen. Some example motivations for this could be a closure or road incident preventing access to the original location.

4.3.2 Item Availability Change (Ava)

The availability map defines the allocation of items across the cities. The work of Sachdeva et al. [318] considers a simple toggling of item availability, however realistic supply chains are likely to contain some level of redundancy across networks. Considering Eqn. 4.1; as the packing plan z is constructed from the KP solution component according to the availability map; dynamic changes can be represented in the model as the time-dependent packing plan $z(\tau)$:

$$G(x,z) = \begin{cases} min \quad f(x,z(\tau)) = \sum_{i=1}^{n-1}(t_{x_i,x_{i+1}}) + t_{x_n,x_1} \\ \qquad\qquad x = (x_1,\dots,x_n) \\ max \quad g(x,z(\tau)) = \sum_{j\in z(\tau)} p_j Dr^{\left\lceil \frac{T_j}{C} \right\rceil} \end{cases}$$

Here a dynamic change in the availability map corresponds to a change in item-city assignments (I_{city}) for a number of items. This means the city index at which an item can be selected is altered in the new dynamic interval. Figure 4.2 provides an illustration of this process together with the other types of change:

$$I_{city,\tau+1} = r$$

where r is a random city index drawn from the uniform discrete distribution $unif\{1,N\}$. The magnitude is controlled as a percentage of the items (as $d_{N,\textbf{Ava}}$) which undergo an assignment change (since the number of items varies across problem subtypes).

A change in the availability map can be contextually interpreted as stock shortages or discontinued items at the item's original city index, and therefore a switch to an alternative depot location or a competitor offering the same items.

4.3.3 Item Value Change (Val)

Dynamic Knapsack problems can have non-static capacities, numbers of items, weights or profits. In the context of a courier collection problem, only the non-static number of items and item profit make logical sense. Since varying the number of items requires updating the availability map, we consider novel dynamics in the item profits only. As with the availability map dynamics, the magnitude of each change is determined by the percentage of item profits that change $d_{N,\textbf{Val}}$. Additionally, the *change factor, cf* gives the percentage and sign (chosen uniformly at random for each item) of change in each item's profit. The existing model can be reformulated as:

$$G(x,z) = \begin{cases} min \quad f(x,z) = \sum_{i=1}^{n-1}(t_{x_i,x_{i+1}}) + t_{x_n,x_1} \\ \qquad\qquad x = (x_1,\dots,x_n) \\ max \quad g(x,z) = \sum_{j\in z} p_j(\tau) Dr^{\left\lceil \frac{T_j}{C} \right\rceil} \end{cases}$$

with all parameters as in the initial definition. A subset of item's profit values p are subject to change between dynamic intervals, represented here by the time-dependency of the profit value of item j: $p_j(\tau)$. An item's profit value is updated as:

$$I_{p,\tau+1} = (1 + cf) \times I_{p,\tau}$$

Realistically, item profit values can change due to a number of factors. These include, global and local stock levels, competition factors, inflation and for perishable items, this can relate to the freshness or quality of items.

4.4 Solver Exploitation & Outcomes of Different Initialization Methods

4.4.1 Initial Set Construction Methods

As mentioned, constructing the initial set based on information about the problem components has been performed by a number of works for the TTP [5], [6], [308]. It has been observed that optimal solutions to the components of the problem do not necessary yield optimal TTP solutions [137]. However, solver-based solutions (***s***), greedily constructed solutions (***g***), and randomly generated solutions (***r***) are examined here in various combinations.

Solver-based Initialization

Several exact TSP solvers exist, for example Concorde [352] and Branch and Bound methods [353]. To remain consistent with the methods in [308], we use the Lin-Kernighan heuristic (LKH v2.0.9, [354]) for the optimal tour component. A simple dynamic programming approach solves the KP prior to the optimization, given sufficient memory resources. The optimal KP solution is transformed into a packing plan using the availability map. Dynamically recalculating an optimal KP solution for the largest problems is thus far computationally infeasible and greedy or random KP solutions are used instead. A population of unique solutions is then constructed using the optimal solution components by employing the mutation operators; Bitflip for KP solutions and Single Swap Mutation for the TSP tour.

Greedy Initialization

Greedy methods can provide near-optimal solutions in some cases and little better than randomized solutions in others. Tours are constructed by iteratively, from the first city (which has no items), selecting the minimum distance to any other city until a complete tour is formed. For the KP solution, the items are sorted in descending order of their profit/weight ratio and the first subset of items with a cumulative weight below the knapsack capacity W comprise the greedy solution. Similarly to the solver-based tours, the single solution is mutated using the algorithm operators to form a population.

Random Initialization

The randomized population consists of random tour permutations and a random item permutations, truncated at the point where their cumulative weight exceeds the maximum capacity of the knapsack.

Algorithm Parameters

A version of the non-dominated sorting algorithm (NSGA-II) was modified to handle the solution representation for the TTP in order to determine the impacts of solution initialization. This algorithm was further developed for the dynamic instances proposed in later sections and is shown in Algorithm 4. 5. The application of evolutionary operators within the algorithm follows [308]; the offspring population is generated by combining equal proportions of solutions with operators applied to the tour, the packing plan and both components. An edge recombination crossover [355], [356] and a single swap mutation [357] are used for tours and bitflip mutation and single point crossover [358] for the KP solutions before converting to packing plans. After each dynamic change the initial tours constructed using the solver and greedy methods are updated; an Inver-Over Repair operator [347] is used as fast alternative to resolving.

Population size is fixed at 90 and a single mutation and crossover event are guaranteed for offspring. Tournament size for selecting crossover parents is $\frac{1}{10} * popsize$. Unless otherwise stated, the maximum iterations are 1000 with five dynamic intervals.

Initialization Impacts for the TTP

The results in Figure 4.3 illustrate the localization of non-dominated sets of solutions achieved by different population initialization on the 52A TTP instance.

Each of the series in Figure 4.3 represents the aggregated set of 30 repeats, each run for 1000 generations. The localization of the non-dominated sets appears driven primarily by the construction method used for the tour component of the initial set. Solver-based tour initialization achieves a high density coverage in the minimum-tour region of the objective space, whilst greedily constructed solutions with longer tours achieve higher profits. Representation from the sets with randomly initialized tours is limited to a few solutions with high profits and much longer tours. It should be noted this coverage is absent in most examined problems with larger KP or TSP components, whilst similar localization of greedy and solver-based solutions is generally present.

Based on these observations, we construct eight different initialization methods to be deployed in response to dynamics changes. These differently exploit problem information to try to maximize post-change algorithm performance. The base Seeding EA algorithm is shown in Algorithm 4. 5 and uses NSGA-II-style mechanisms such as Pareto-dominance and crowding-distance based tiebreaking replacement. By replacing the offspring gener-

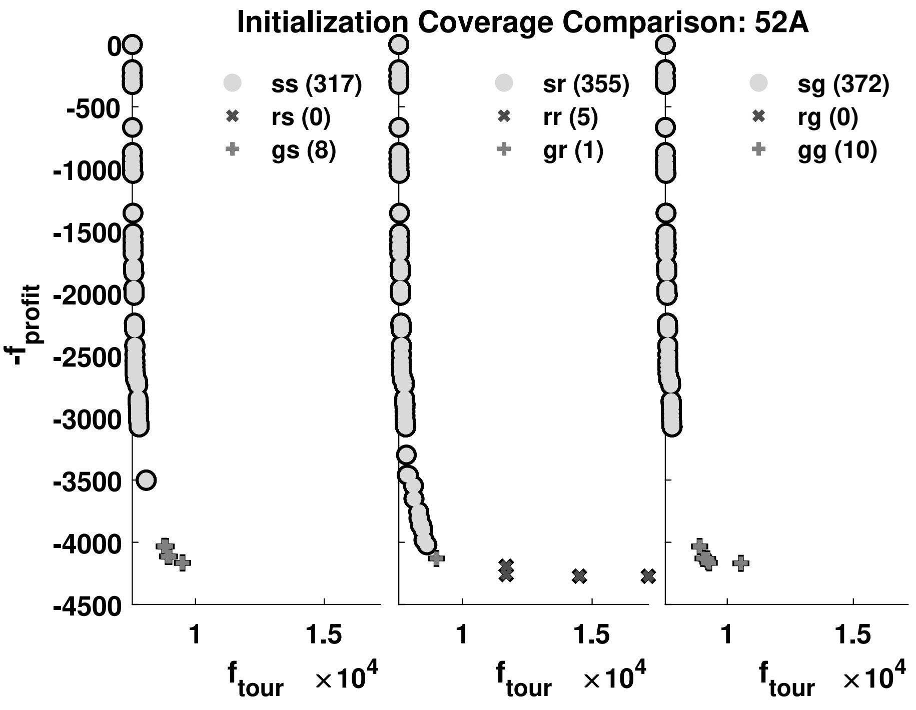


Figure 4.3: Comparison of non-dominated set (NDS) coverage with different initial population compositions. Each series corresponds to an initial set construction method (solver, random or greedy) for tours (first character) and KP-solutions (second character). For example, **ss** solutions are contributions to the aggregated NDS that come from an initial population comprised of **s**olver-based tours and **s**olver-based packing plans and **gr** solutions come from an initial set of **g**reedy tours and **r**andom packing plans. For **ss(317)**, this means that 317 unique solutions are present in the aggregated NDS which originated from an initial population generated using only solver-based information. Results featured for 30 independent runs of each initialization on the *52A* problem. Localization can be seen most clearly depending on tour initialization method; from left to right solver, greedy and random initial tours enable localized sets of non-dominated solutions.

Table 4.1: Responsive population construction methods for tour and packing plan solution components employed for the DTTP instances. *The *mN* algorithm uses the *mC* initialization but has no response mechanism to dynamic changes.

Response Strategy	**TSP solution**	**Packing Plan**
pS	solver	solver
pG	greedy	greedy
pR	random	random
mS	solver	solver/greedy/random
mG	greedy	solver/greedy/random
mR	random	solver/greedy/random
mC	solver/greedy/random	solver/greedy/random
mN	none*	none*

ation step with a set-construction step after a dynamic change, the population is seeded with solutions that will be relevant to the new dynamic interval. The goal is to mitigate the impacts of change and improve coverage of the non-dominated set in each interval. The eight different seeding mechanisms are described in Table 4.1.

Algorithm 5 Seeding EA - responsive seeding algorithm.

```
1: Require: $popsize, maxiter, SeedingMethod, d_N,$ ;
2:         $DynType$
3: $P \leftarrow SeedingMethod$   \\ Initialize population
4: $EvaluateFitness(P)$
5: $H.0 \leftarrow RecordHypervolume(P)$
6: $iter \leftarrow 1$
7: while $iter \leq maxiter$ do
8:     if $iter\%(maxiter/d_N) == 0$ then
9:         Apply Dynamic Change of type $DynType$
10:        Update Problem Information
11:            \\Recalculate solver/greedy solution
12:            \\components if required
13:        $Q \leftarrow SeedingMethod$
14:    else
15:        $Q \leftarrow GeneticOperators(P)$
16:            \\Tournament selection, three part genetic
17:            \\operator application (see Section 4.4.1)
18:    end if
19:    $EvaluateFitness(Q)$
20:    $P \leftarrow P \cup Q$
21:    $RankAndNonDominatedSort(P)$
22:    Crop $P$ to $popsize$
23:    $H.iter \leftarrow RecordHypervolume(P)$
24:    $iter++$
25: end whilereturn $P, H$
```

Table 4.2: KP component subtypes in problem set, see [359]–[361] for details.

Label	Items per City	Knapsack Capacity	Weight/Profit Rel.
A	1	(low) $\frac{1}{11}\sum I_w$	strongly correlated
B	5	(med) $\frac{5}{11}\sum I_w$	similar
C	10	(high) $\frac{10}{11}\sum I_w$	uncorrelated

4.5 Impacts of Dynamics in the DTTP

4.5.1 Problem Instance Generation for the DTTP

The set of problems examined follows the format instances used in the competitions for the TTP [359]–[361]. A subset of the comprehensive TSPLIB-based benchmark set proposed in [307] is used[1] with three KP-component subtypes. These subtypes are given in Table 4.2; we denote these as *A, B & C* for clarity. From [307], evolutionary algorithms solving TTP instances can effectively cope with problems of up to 3000-5000 cities, however Wagner [311] suggests focusing on performance improvements on smaller instances first. Therefore our sample of problems fits this range ($berlin52, a280, rat783, u2319$). Problem variants are referred to by the number of cities and KP type (e.g. *52A*).

The schedule of dynamic changes is fixed at $d_{freq} = 200$ generations to allow population stability to return and for clearer observation of change impacts. A number of reproducible patterns of changes are generated using fixed pattern seeds to ensure consistent instances and improve the reproducibility of these experiments (examples of these

[1]See: cs.adelaide.edu.au/~optlog/CEC2014COMP_InstancesNew/

can be found in the code locations provided in the front matter). The magnitude of change is fixed at $d_{N,Loc} = 2, d_{N,Ava/Val} = 5\%$ to determine the preliminary impacts of the dynamics. The impacts of frequency and severity and the specific challenges for the DTTP are discussed within Chapter 5.

4.5.2 Performance Measurement & Statistical Testing

Measurements

Hypervolume has been used as a performance measure for the bi-objective TTP [303], [308], but no consistent nadir point is used. We use a reference point calculated as $(f^{\dagger}_{tour}, f^{\dagger}_{profit})$=$[\overline{D} \times |D|, 0]$ to allow disparate coverage of the achieved sets to be compared, whilst removing a portion of the poor quality objective space. Any solution set that does not dominate this point is given a hypervolume of zero.

Profiles and Ranking: The Composite Median

The following procedure is applied separately to each problem to effectively communicate performance across the sample of dynamic instances. For 10 patterns of dynamic changes (instances), the mean hypervolume across 30 repeats is calculated. For plotting the hypervolume profile, the median of these 10 instance-means is taken, an example of which is shown in Figure 4.4. Rankings are calculated comparing like-instances for all response methods on the 10 patterns. The median rank for each response is reported as the *composite median* in Fig.4.5 and in following results.

Statistical Testing for Quantitative Comparisons of Performance

A single value for comparative performance of algorithms that observes statistically relevant differences is calculated. For each algorithm a number of repeats are performed for each dynamic pattern, with the hypervolume recorded in each generation. At each numbered time step (generation), the measurements from repeats form the sample set. These are used in pairwise one-tailed t-test comparisons, using Bonferroni correction ($n = 2$) and $\alpha = 0.05$. For example, using three methods *J, K* & *L*, a total of six tests are performed at each generation with the following alternative hypotheses:

- (1) $\mu_{t,J} > \mu_{t,K}$; (2) $\mu_{t,K} > \mu_{t,J}$;
- (3) $\mu_{t,J} > \mu_{t,L}$; (4) $\mu_{t,L} > \mu_{t,J}$;
- (5) $\mu_{t,K} > \mu_{t,L}$; (6) $\mu_{t,L} > \mu_{t,K}$;

where $\mu_{t,J}$, $\mu_{t,K}$ & $\mu_{t,L}$ are the means of the sample for algorithm *J, K* & *L* at generation t, respectively. The binary outcomes of the tests are recorded.

Tests are repeated for every generation and the percentage of positive results (alternative hypothesis is accepted) is recorded. In each pair of comparisons (*J* vs. *K*, *K* vs. *J*) the

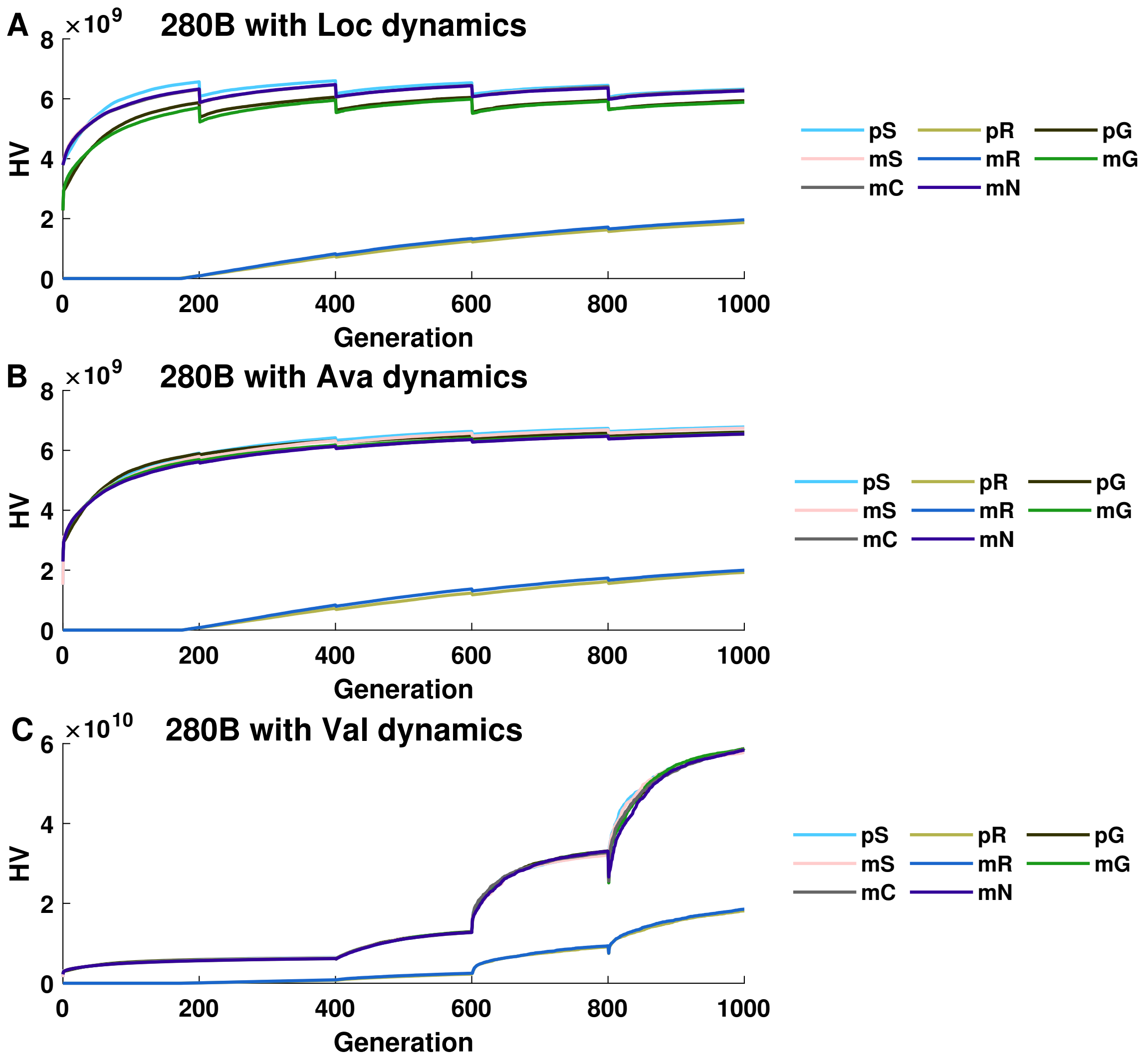


Figure 4.4: Median hypervolume profiles across 10 patterns of dynamic changes (with 30 repeats for each) for eight different dynamic responses on the *280B* problem. Change frequency is 200 generations. (A) Response mechanism hypervolumes for the **Loc** dynamics. (B) Response mechanism hypervolumes for the **Ava** dynamics. The profiles of **Loc** and **Ava** hypervolumes are characteristic DMOP measurement curve and an attenuated-impact version, respectively. (C) Response mechanism hypervolumes for the **Val** dynamics. Each type of dynamic changes generates a visibly different effect on the problem but signed item profit changes (**Val**) show marked impact on existing solutions with successively increasing magnitudes.

maximum sum of these values will be 100% but is likely to be less than this as there are generations where performance between two algorithms is not significantly different. If algorithm J is better in more generations than the comparison algorithm K, a larger value will be seen for the J vs. K comparison.

4.5.3 Visualising Impacts of Dynamic Changes in the DTTP

Hypervolume (HV) measurement profiles can illustrate the high level features of the impacts of problem dynamics; each novel dynamic formulation affects the problem with different character and severity. Figure 4.4 illustrates the HV profiles achieved for each of the eight population seeding methods for the *280B* problem. The profiles plotted are the composite median; the median of the ten dynamic instances, for each of which the mean of 30 repeats is calculated.

The different types of dynamics affect the ability of seeding responses to mitigate the impacts of dynamics. As these impacts are reliant on not just the type of dynamics, but their frequency and magnitude, the insights drawn here are preliminary and indicative of trends rather than concrete assessments. Typically characteristic-in-shape decreases in hypervolume after each of the five changes, can be seen in the **Loc** dynamics, whilst an attenuated version of these is seen in the HV profiles for the **Ava** dynamics. This is intuitive from the 'relative directness' of the tour and packing plan solution components; a change in the city locations impacts every solution in the population (since a valid solution must visit all cities), whilst a change in item availability immediately affects only those solutions containing the items that have been altered.

For the **Val** dynamics, the impact appears more varied and the performance of the responses more volatile. Similarly to the **Ava** dynamics, as every item is not necessarily included in a current solution (in the population) it is expected that the impacts of changes is lessened and relatively low in magnitude. However, it appears that even a small change to the values of a small percentage of randomly selected items can evince a large change in the solution set's hypervolume.

4.6 Difference in Responsive Solution Set Construction Methods

The polar plots in Figure 4.5 illustrate the performance of each responsive seeding method across the set of problems with different types of dynamics. The varying performance of the responsive seeding methods is visible across the types of dynamics and the problem component combinations.

A prominent feature of these results is the poor performance of the random seeding *pR*. Together with *mR*, these consistently achieve the lowest ranks on every problem and for each type of dynamics. The consistency between *pR* and *mR* indicates that diversifying the packing plan information (*mR* uses random, solver-based and greedily constructed knapsack solutions together with random tours) does not substantially improve performance. This reiterates the relative control the solution components excise on the optimization; good performance is primarily driven by good TSP solution components.

These results also indicate that the TSP-component of the examined problem can influence the most effective responsive seeding method. This is demonstrated in *pS* and *pG* methods for the *52A,B&C* and *280A,B&C* problems. Responsive seeding derived purely from solver solutions (*pS*) achieves better performance on the *280A&B* problems, whilst greedy-constructed problems are better for the *52A,B&C* problems. These statements apply across all three types of dynamics. The best response method for the *280C* problem

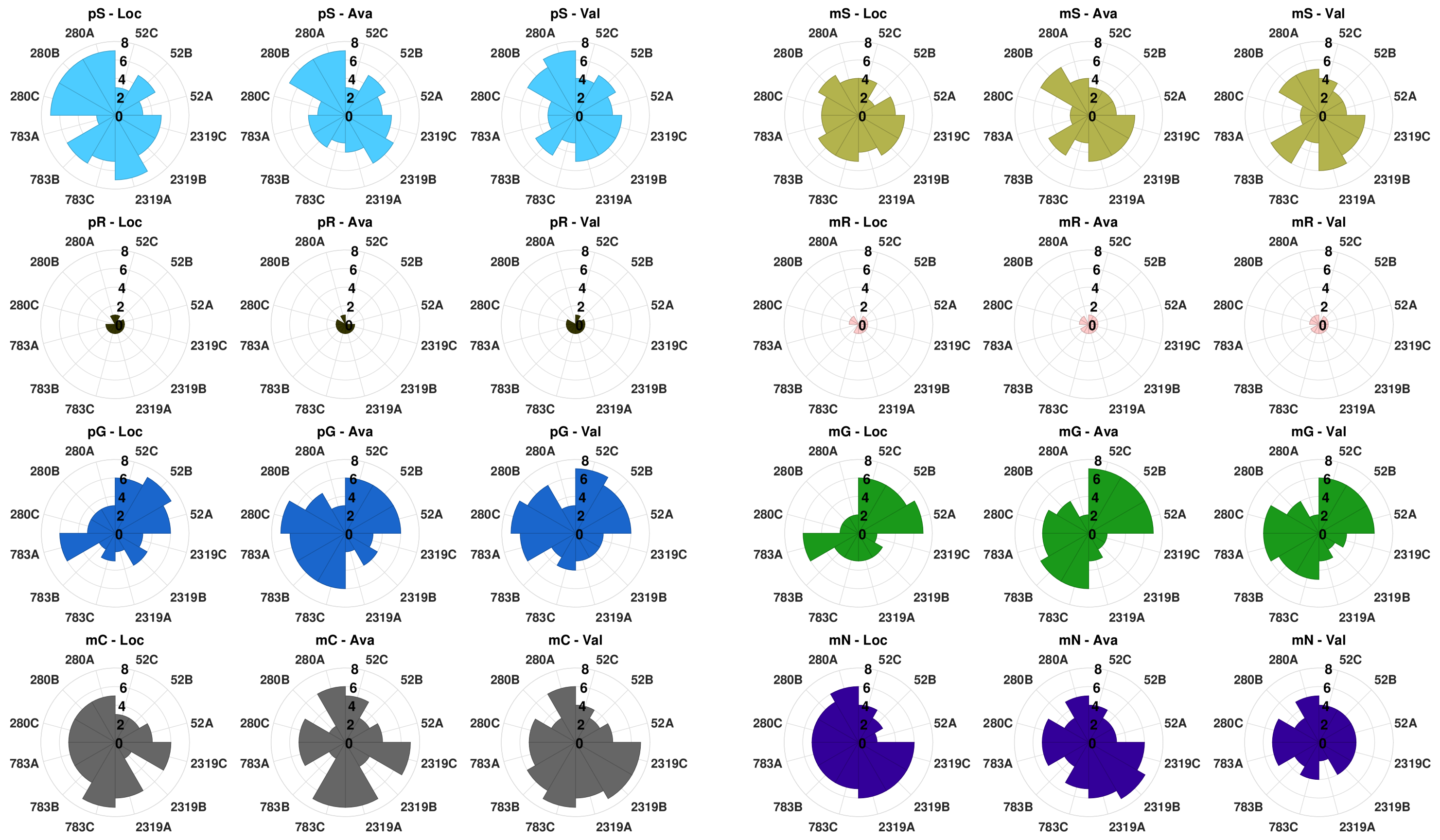


Figure 4.5: Composite-median end of interval hypervolume rankings for the three types of dynamics, grouped into the eight different seeding methods (as listed in Table 4.1) on the 12 problems in the test set (plotted radially on each axes as the number of cities and KP type). Ranks are relative to other seeding methods. See Section 4.5.2 for the calculation of the composite-median. The **Loc**, **Ava** & **Val** labels indicate the type of dynamics in the problems. Solver-based responses (first row) show suitability for all 280 variants and some *2319* variants; randomized initialization (second row) consistently achieves the worst ranks on all problems; greedily-constructed seeding (third row) gives the best response on *52* variants and others depending on the type of dynamics. Combined (fourth row, left) and passive (fourth row, right) methods have complex trends depending on the type of dynamics and problem components.

appears to change with the type of dynamics; *pS* in **Loc** dynamics, and *pG* in **Ava** & **Val** dynamics.

Compared with each of the *pS* and *pG*, the diversification of the packing plan solution components has mixed results. The *mS* strategy achieves lower ranks than *pS* for most problems; implying that solved sub-components are preferable to more diverse packing plan solution components. For *mG*, for *52A,B&C* with **Ava** dynamics, the maximum rank is achieved over *pG*. For other problems, *mG* ranks are below those achieved by *pG*; only for some problems a diversified packing plan set is a beneficial.

The *mN* method provides important insights into the comparative difficulty of each presented dynamic formulation. The *mN* method is 'passive' such that, the population is not seeded in response to a change. Initialization with a diverse population of both tours and packing plans, as in the *mC* method, is used. This enables determination of the relative benefit of each response method compared with a 'do nothing' approach. As established earlier, the informed selection of a baseline comparison allows for more meaningful conclusions to be made.

Clearly visible in the **Loc** dynamics (leftmost of bottom-right triplet in Fig. 4.5), is a steady increase in ranks as the size of the TSP-component grows. For problems with the same TSP-size, there is a similar increase between *A,B&C* (1, 5 & 10 items per city respectively). The other response methods increasingly struggle to do better than a passive approach as the size of both problem components increases. It is important to note that for **Loc** dynamics, the magnitude of the change is constant regardless of the TSP-component size. For the **Ava** dynamics, a similar trend is present with *mN* achieving high ranks on the largest problems. The ranks achieved for the **Val** dynamics imply they have a non-uniform effect on the range of problems without a clearly discernible trend.

Finally, the rankings achieved by the *mC* response method (bottom-left triplet in Fig. 4.5) inform on the utility of diversity in response to dynamic changes of different types. On the smaller and mid-sized problems (all *52 & 280* variants and all types of dynamics), *mC* achieves reasonable rankings, however the performance of *mC* is deflated by the good performance of the *pG, mG & pS* methods on these problems. Consistently high rankings can be seen for the larger problems in the set, (*783 & 2319* variants). Interestingly, the *u2319B* problem is ineffectively handled by *mC* under both **Loc** and **Ava** dynamics. Lower rankings are also achieved on all B-type problems with **Ava** dynamics as well. This type of KP-problem has 5 items per city with similar weights and profits. Further investigation may elucidate the interactions between types of dynamics and KP-components. As the *mC* method provides reasonable performance across the problem set, we compare it with adapted methods for the static TTP.

4.7 Comparison with TTP heuristics

We compare the mC strategy, named succinctly here as 'SeedEA', on 12 DTTP instances each with 10 patterns of city location changes, with two methods for the static TTP: S5 [5] and MATLS [300] methods. Designed for non-dynamic single objective TTP problems, there are no existing comparable methods for the bi-objective DTTP. Both also employ TSP solvers and therefore the minimum time TTP solution is trivial compared with the high-profit solutions. S5 works on iterative improvement over a single solution via a parameter search; we allow the same number of iterations as generations for SeedEA. Whilst MATLS maintains a population of solutions, there is no dominance assessment compatible to simultaneously optimize for the bi-objective case. Therefore, both methods preferentially replace for high-profit solutions. As MATLS is population based (with *popsize* equal to SeedEA), we examine two versions; in **r**MATLS, the population is reinitialized at the beginning of each dynamic interval, whereas in **k**MATLS the population is kept in the next dynamic interval.

4.7.1 S5 Heuristic

The S5 heuristic was constructed from its description and the pseudocode provided in [5]. A score is calculated for each item based on the benefit and impact on the remainder of the tour. Items are then selected iteratively and the scores updated until no further improvement can be found. The scoring parameters are varied during the heuristic to narrow the search for the best packing plan. Initialization occurs using the Chained Lin-Kernighan TSP solver for the tour component. The KP component is constructed iteratively by the heuristic.

4.7.2 MATLS (rMATLS and kMATLS)

The Memetic Algorithm with Two-Stage Local Search method was also constructed based on the pseudocode and description in [300]. The basic format involves a local search step for the TSP and then the KP component of a small population of solutions. The tours are initialised using the Chained LKH TSP solver. The KP components are generated by an heuristic algorithm within the methodology proposed in the original paper.

4.7.3 SeedEA

The SeedEA is effectively Algorithm 4. 5 with specifically the *mC* seeding response strategy from 4.1. Both the S5 and MATLS methods utilise a heuristic packing algorithm designed to select the best items after a tour is generated. To allow for competitive results, after a change event an approximate high profit solution is constructed by first building the terminal end of the tour based on the highest profit solutions consecutively chosen in reverse, accounting for the time-adjusted-profit and weight-adjusted-time of item selec-

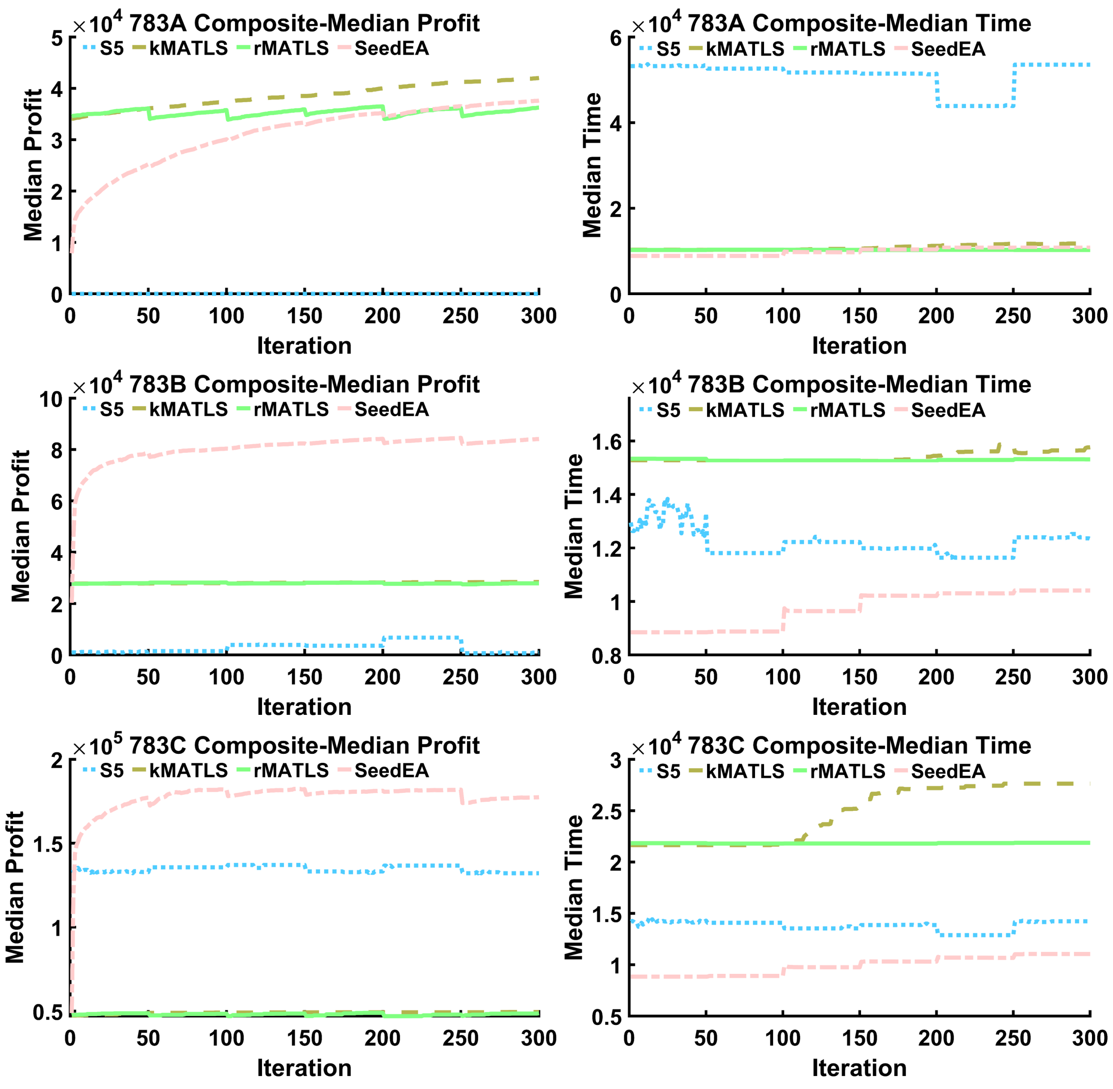


Figure 4.6: (top row, left) Composite median highest profit objective value in each iteration achieved by each method on the 783A problem. (top row, right) as left but for lowest time objective value. (middle row, left) Composite median profit profiles for algorithms on 783B. (middle row, right) Composite median time profiles for algorithms on 783B. (bottom row, left) Composite median profit profiles for algorithms on 783C. (bottom row, right) Composite median time profiles for algorithms on 783C.

tion. The shortest path for the remaining cities is found by LKH solver and combined with the terminal end of the solution. This was added to the post change offspring population.

Figure 4.6 illustrates the attainment of the two objectives. Since the comparison methods were not designed for the bi-objective version of the problem, measurements such as hypervolume cannot fairly be compared. Moreover, since the use of exact TSP solvers is common to all methods, finding the minimum-time solution is trivial and alone, does not provide insight. Here however, we show the minimum tour solution alongside the maximum profit solution in each generation (left vs. right subplots) to highlight that the SeedEA method can effectively find a high-profit solution whilst maintaining a lower-time tour simultaneously.

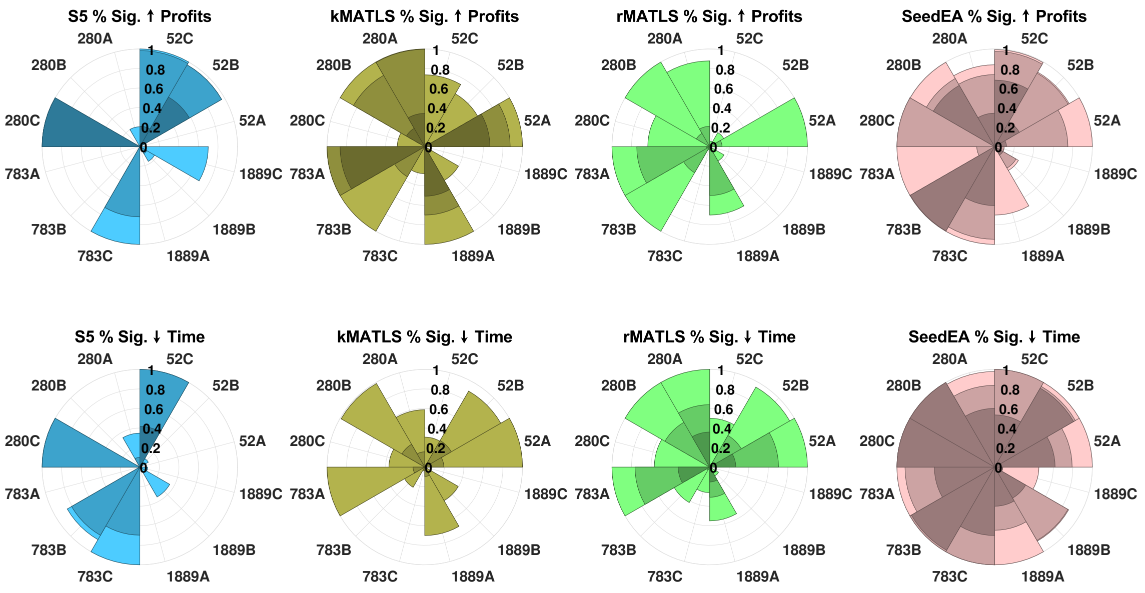


Figure 4.7: Plots showing the problem set coverage of adapted TTP methods and SeedEA in terms of the proportion of iterations containing solutions with significantly higher profit (top row) and significantly lower time objective values (bottom row). The three shades on each plot correspond to one, two and three significant improvements (from lightest to darkest) in the comparison to the other methods. Statistical test are carried out separately different dynamic patterns for each problem and aggregated in the percentage calculation.

Within Figure 4.6 SeedEA performs well across the example B-type KP variant, producing a high profit solution and maintaining a low-time solution simultaneously, where the other methods cannot. The rMATLS, kMATLS and SeedEA methods perform well on the smallest, A-type TTP problems, however on the B & C type problem, the performance of the MATLS method is much lower. The S5 methods appears more suitable for the B & C variants, but still does not compete with SeedEA. These trends are consistent across the problems but the results are collated into ranking for clearer interpretation.

SeedEA maintains pressure on achieving high profit solutions whilst preserving the diversity to achieve good objective space coverage. The additional diversity of this method allows for improved exploration of the high-profit solutions through exploitation of decision variable diversity contained in the population. The construction of the approximate high profit solution heuristic reveals the importance of the *dropConstant* in achieving high profit solutions. This value represents weight intervals at which the velocity of the thief is slowed by the weight of items; it may be possible to optimize to each threshold for a period of the tour – it may provide an exploitable feature for future heuristics. In terms of the SeedEA, by maintaining a greater diversity of solutions that comprise tours with items selected at a variety of locations, exploiting this unhindered velocity threshold may be more common. Wagner [311] also states that high-profit solutions may require longer tours, of which the SeedEA maintains a selection.

Within Figure 4.7 we compare performance of the different methods across DTTP problem instances. We calculate the proportions of significant 'wins' per iteration of each method over the others across each pattern of dynamics and for each problem. A Bonferroni correction of $n = 3$ is applied. Due to persistent TSP solver divergence issues, the 2319 problems were substituted for variants of the rl1889 problem.

The radial bars represent the significant improvements over the other methods; the radial axis represents the proportion of iterations (across all repeats and patterns) in which a method achieves a significantly higher profit (top row) or significantly lower time (bottom row). The shading represents a significantly better solution over one (lightest), two (mid shade) or all three (darkest) of the other methods. For example, for kMATLS on the 1889A problem, in 100% of iterations, the profits are significantly better than one of the other methods (we can infer it as the S5 method). In approximately 70% of iterations, the profits are significantly better than two methods (S5 and rMATLS) and in 30% iterations kMATLS finds significantly higher profits over all three other methods.

Generally, the MATLS methods perform well on the A-type variant problems, but their performance declines greatly on B- and C-type problems. Results for kMATLS are better than rMATLS in achieving higher profits, whereas achieving both higher profit and low time solutions is better handled by rMATLS. S5 shows the opposite with no competitive performance on A-type problems but some limited significant achievement in profits and times in the B- and C-type variants.

The very good performance of kMATLS on the A-type variants reduces the overall significant performance of the SeedEA's darkest area for profit - most notably on the A-type problems. S5's performance on 280C is unrivalled and it achieves good performance on 52B and 52C as well; these also contribute to reducing SeedEA's darkest shade area.

Given the comparison methods were not designed for the dynamic or multi-objective problem, they still provide significant improvements in high profit solutions on a limited and varied subset of the total problems examined. Another important consideration is the relative execution time and objective function evaluations that these methods consume.

The number of function evaluations, using 280A,B & C problems as an example in Figure 4.8, remains similar for SeedEA, and relatively lower than for rMATLS and kMATLS. For these, the value increases 10-fold from the A-type (1 item-per-city) to B-type (5 items-per-city), then 3-fold from B-type to C-type. For S5, the number of function evaluations remains relatively low across the different KP types.

The MATLS methods involve two local search stages, one for each of the TSP and KP problem components. With more items in the problem, the neighbourhood of the local search increases and therefore the number of required evaluations to search it is inflated as the problem size increases. SeedEA's mechanism is independent of the problem size and therefore the number remains relatively constant across the problems.

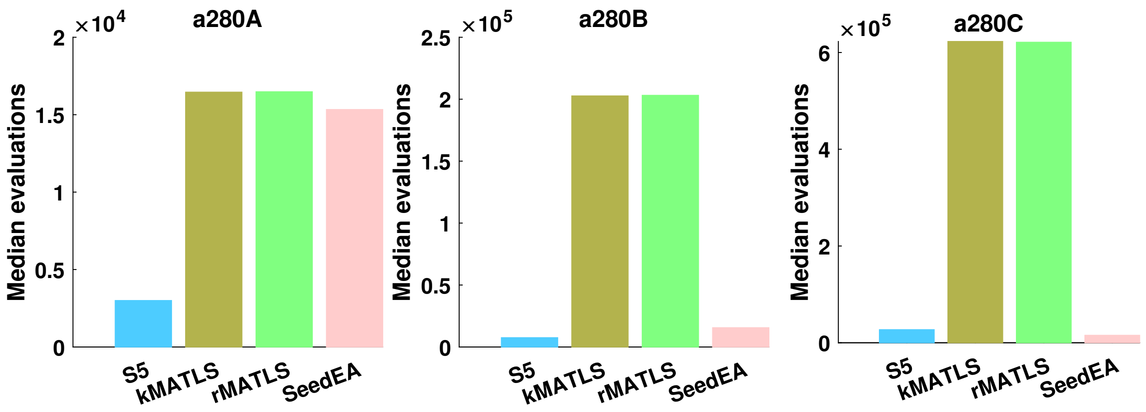


Figure 4.8: The number of objective function evaluations; the median across the (up to) 30 repeats for each of 10 patterns of dynamics for the 280 TSP component with A, B and C KP variants for each of the four methods being compared.

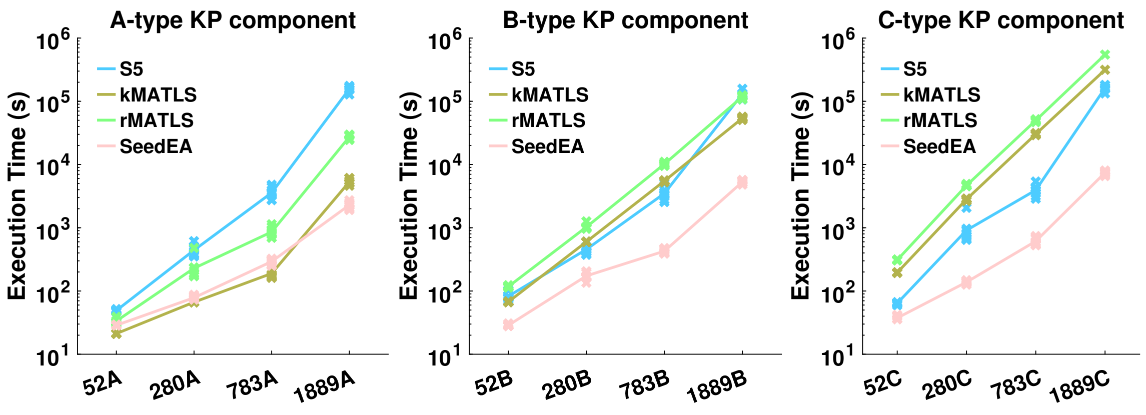


Figure 4.9: Execution times for each comparison method. The mean for each dynamic pattern is given as a separate point and the median of these points for each problem is connected for clarity. The problems are grouped by their KP-component.

In addition to function evaluations, each method requires differing numbers of TSP and KP solver calls. For example, the populations for the MATLS methods were initialized according to the origin paper's protocol – for the TSP components 10 solutions initialized by LKH solver, and 40 via Minimum Spanning Tree method. For rMATLS, this process is repeated after each dynamic change, greatly increasing the relative execution time.

Figure 4.9 shows the execution times of each method with the line joining the median points for each problem. All experiments were run using MATLAB 2018b, an Intel-i7 processor (3.80GHz) and 16GB Memory.

On A-type problems, kMATLS has similar execution times to SeedEA and Figure 4.7 depicts good signficant improvements on these problems. However, as shown by Figure 4.8 as the size of the KP component increases (from A- to B- to C-type), SeedEA and the MATLS methods become increasingly different. For example, the largest of the problems 1889C, the median execution time for a single run of SeedEA was 2.7 hours, for rMATLS this was 138 hours. Comparison methods have redundant calls of TSP solvers, which can result in extreme inflation of execution times. Also greatly increasing the number of objective function evaluations, as in rMATLS and kMATLS also contributes to this difference.

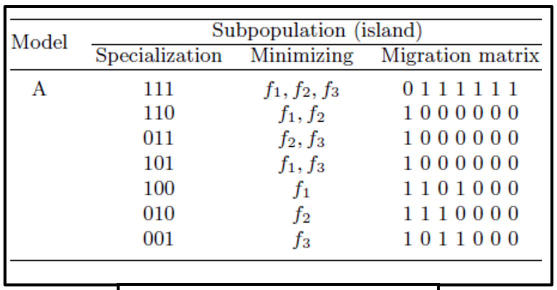
4.8 Development of Multi-Population Topologies with Migration and Mixing for the DTTP

The preliminary results displayed in Figure 4.3 highlight the localization of obtained solutions based on the composition of the initial set of solutions. There is a natural opportunity to develop and exploit this using multi-population methods to improve the coverage and diversity of high-quality non-dominated solutions achieved for the DTTP instance proposed. Together with this, quantification of the difficulty of information relevance of solver-based exploitation as the problem state changes in these dynamic instances is explored.

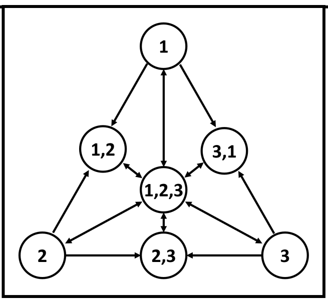
A summary of relevant multi-population and island model topologies is given earlier in this section, (see Section 4.2.5). Using the work of Xiao et al. [332] as inspiration, an iterative development and testing of connected population topologies resulted in a simple multi-population structure with carefully selected migration patterns to maximize the trade-off between exploration and exploitation of found solution information. This is compared with a single population approach and a simple unconnected multi-population approach.

This development of multi-population methods spans three phases, however in the interest of space, the details of the first two are summarised briefly here as they do not contribute useful results. Initially, an interconnected hierarchical topology of sub-populations including all response methods (herein used to refer to the different solution construction methods) was constructed. A diagram of this, together with the proposed structure from [332] is provided in Figure 4.10. Having many populations however, requires a large global population size in order for effective/efficient evolution to occur within each of the sub-populations. As this also results in an inflation in the number of function evaluations and a decrease in the optimization speed, combined with the observable coverage overlap between some of the response methods, a smaller model topology was constructed instead.

Secondly, the definition of an objective bias filtering step allows for a single response mechanism to be employed across a smaller number of more intelligently connected sub-populations. The use of an objective bias allows for the focused isolated evolution towards specific regions of the objective space, however a comparison to select the base response mechanism was required. To provide results with the greatest context to the previous TTP literature, the multi-population approaches were applied to problems with



Model	Subpopulation (island)		
	Specialization	Minimizing	Migration matrix
A	111	f_1, f_2, f_3	0 1 1 1 1 1 1
	110	f_1, f_2	1 0 0 0 0 0 0
	011	f_2, f_3	1 0 0 0 0 0 0
	101	f_1, f_3	1 0 0 0 0 0 0
	100	f_1	1 1 0 1 0 0 0
	010	f_2	1 1 1 0 0 0 0
	001	f_3	1 0 1 1 0 0 0



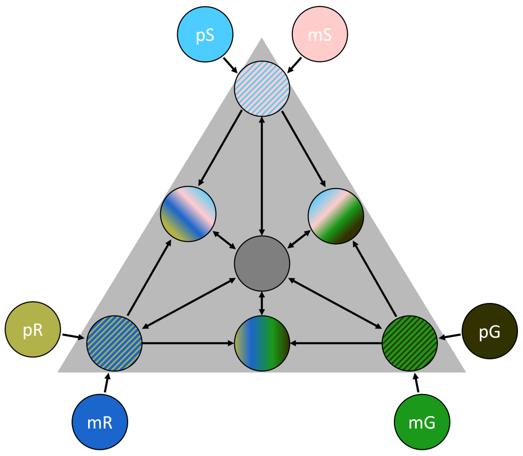


Figure 4.10: (left) Multi-population island model with migration and objective biasing as proposed by Xiao et al [332]. (right) Adaptation of multi-population structure to incorporate information from all solution construction methods. Each circle corresponds to a sub-population and the arrows indicate the directions of migration of solutions. The colours of each sub-population indicate the source of solution information, with the central population providing en environment for mixing of all solutions. A modified version of this model with a structure refined based on utility and solution contributions was adopted for experimentation - see Figure 4.11.

smaller TSP components. The *pS* response achieved the best hypervolume and extent in both objectives (best 'extreme' solutions in each objective) across the static instances (the first dynamic interval) of the problems, with the 'mC' response adopted in the SeedEA providing similar performance. From the outcomes of these first two stages, the final multi-population model topology was constructed and tested, the details of which are to follow.

4.8.1 Seeding-Dependent Terminal Solution Localization and Island Structures

The proposed multi-population island topology to tackle the DTTP contains three key features: four sub-populations each with a separate optimization goal and purpose; the migration events between sub-populations with specific frequency and occurrence; and solver-solution-based responsive seeding mechanism as seen before. This mechanism is triggered in response to the dynamic change events and functions the same as in the SeedEA. Each sub-population is effectively running a separate version of the SeedEA including the replacement of offspring with a (diversified) solver-based population of solutions in the generation of the dynamic change.

Sub-population Specialization: Objective Biasing

The previous results highlight the differential coverage of the objective space when using different initialization sets and responsively generated solution sets. In these cases, the optimizing algorithm has no particular focus for finding solutions in any particular objective, other than how the topology of the search landscape guides the search. This is the expectation in a multi-objective optimization problem – both objectives are prioritised and we can refer to this unbiased or 'neutral'. However, we can also guide the optimization by biasing the search in each of the objectives separately and the distributed nature of multi-population methods facilitates this. By adding a custom filtering stage in some of the sub-populations we can prioritise optimization towards solutions in a particular region of the search space (similar to a reference point based guided search [362]–[364]).

Firstly, for each of the sub-populations we label which objective is the priority or if there is no bias, we describe the sub-population as neutral. For each of the biased populations within the general evolutionary loop, before the application of genetic operators an additional population filtering step is performed. The population is sorted in descending order for the prioritised objective's fitness values and the worst 90% of solutions are discarded. Offspring are then generated as normal using tournament selection and mutation and crossover operators to generate a full $2N$ set. For example, using $popsize = 90$ the best nine solutions will be kept after filtering, then 171 solutions would be generated using only the filtered solutions as the parent pool. This generates additional evolutionary pressure towards good solutions in a particular region of the objective space, but relies on a logical connectivity of solutions within the decision space (such that neighbourhoods are relatively conserved between the decision and objective spaces).

Having a combination of biased and neutral sub-populations within an island topology intuitively allows for a greater total diversity in the non-dominated set. Connecting these and sharing information between the populations may facilitate further drive towards good solutions.

Island Topologies with Migration Events

In the proposed multi-population topology with migration depicted in Figure 4.11, a structure with four sub-populations is defined. There are two biased sub-populations, one focusing on each of the objectives for the bi-objective DTTP, and two neutral populations. One of the neutral populations carries out a default approach to optimization, this is referred to as the 'neutral population'. The other neutral population allows for the mixing of solutions from all other populations and serves a temporal buffer. This is referred to hence as the 'mixing population'. Migration routes are defined by a matrix based on the utility of the solutions in the destination; there is no migration from the mixing population to the focused-objective populations. The frequency of migration events is also controlled such that migration between the focused and neutral populations happens more rapidly than the migration to and from the mixing population. The frequency, phase and connectivity were selected to avoid stagnation and maximize information sharing whilst maintaining isolated evolution and preventing annihilation of the endemic solutions by

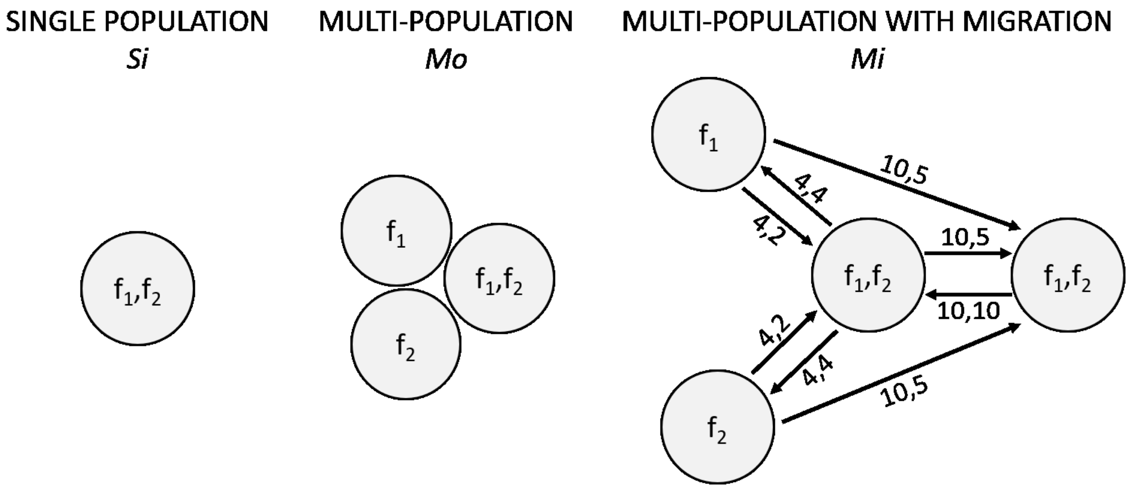


Figure 4.11: Population model topologies applied to the Dynamic Travelling Thief Problem instances. (left) A single-population model, represented as one circle, represented the default approach employed by most algorithms; no bias is asserted towards solutions that better satisfy a particular objective. (middle) A multi-population topology where three groups of solutions evolve in isolation; the objective function labels indicate the bias towards solutions that better satisfy an objective (or both – neutral) therefore exploring particular regions of the objective space. (right) The proposed multi-population topology with migration between populations. Two sub-populations each with a focus for a particular objective and two 'neutral' sub-population are connected with migration patterns as indicated by arrows. The direction indicates migrant solution flow, with first number on each arrow indicating the frequency of the migration and the second indicating the generation of the first migration event.

migrants. Despite a high frequency of migrations being known to influence evolutionary trajectory (leading to premature convergence) [343], the objective biasing in the focused populations and the regular refreshing of the neutral population from the mixing population prevents this.

For brevity the connectivity matrix is not included here, however the edge weights in Figure 4.11 indicate the frequency and the first occurrence of migration for each of the connections.

Migrants are selected as an evenly distributed sample of the (preferentially) non-dominated solutions in the origin population. The migrant set size is equal to 20% of the origin population size (as in [340]). Assimilation of migrants into the destination population occurs through normal ranking and replacement based on crowding distance. Measurements for this model are taken using the neutral population only.

The other population models depicted in Figure 4.11 allow for comparison and evaluation of the multi-population model with migration. The first of these is a simple single population model, abbreviated as *Si*, which is the default approach adopted in the previous experiments in this chapter. The second is simple multi-population model *Mo*, with three sub-populations: one biased for evolution in each of the two objectives and the third a neutral population with no objective filtering biasing subroutine. The proposed multi-population model with migration is abbreviated as *Mi*.

4.8.2 Reporting Performance and Incorporating Statistical Analysis

As mentioned previously reporting performance can be challenging in dynamic and dynamic multi-objective optimization problems and algorithms. This is compounded for multi-population models unless a clear paradigm is defined and followed. For each of the population models compared here, the performance is reported as follows:

- *Si:* The hypervolume measurement is measured for the non-dominated set of solutions in the population in each generation.
- *Mo:* The non-dominated solutions in each sub-population are aggregated and the hypervolume of the resulting set after another round of non-dominated sorting is taken as the measurement.
- *Mi:* The hypervolume measurement is taken on the non-dominated set within the neutral population only.

The inclusion of meaningful statistical comparisons is also difficult in this domain as the composition of samples varies depending on the frequency of measurements and the motivation and approach to performance measurement in general. Here we extend the approach used previously, in the evaluation of seeding responses and in the comparison to TTP heuristics.

The hypervolume is measured at in every generation of the optimization and these measurements form the samples. A pairwise one-tailed t-test, with an alternative hypothesis for greater mean, is then performed to provide a statistical comparison between any two methods (as there are three methods, a Bonferroni correction of $\alpha = 2$ must be applied) for each generation of the optimization. This process is repeated for each of the 10 patterns of dynamic changes examined. For example, if the optimization runs for 1000 generations (independent of the number of dynamic change events) there will be 1000 (generations) $\times 3$ (methods) $\times 2$ (comparisons for greater mean) $\times 10$ (patterns of dynamic changes) statistical tests performed. The results of these are collated to provide, for each model in every pairwise combination, the percentage of the generations, across all patterns of changes, in which it can provide a significantly higher hypervolume.

A sample of problem instances is used to evaluate the population topologies. Using the City Location dynamics, 10 patterns of changes, each with 10 change events and a fixed magnitude of $d_N = 2$ were examined, with 30 repetitions conducted for each. For the algorithm parameters, the global population size is set to 360, with the crossover frequency set to 0.9 for both solution components and the mutation frequency set to $2/N_{items}$ and 2 for the bitflip and swap mutations respectively. The value of γ is 10%. The other problem parameters are as follows: the value of r in Eqn. 4.2 is fixed at 0.45 (midpoint of random interval) in order to generate reproducible problem scenarios. The Dr, v_{min} and v_{max} parameters are set to 0.9, 0.1 & 1.0 respectively, as in [137].

In summary, for each of the 18 problems, 20 repeats are performed on each of 10 patterns of dynamics. A similar format for the specific instance was employed here, using the *52A* notation to denote the number of cities in the TSP component and the type of the KP component (see Table 4.2 for more information).

4.8.3 Impacts of Information Currentness

An additional goal of these experiments is to highlight the importance of the 'currentness' of the information being used to seed the population in response to dynamic changes. Information relevancy in terms of its currentness describing its applicability to the current problem state, represents a clear challenge in the dynamic problem case. Here, we quantify the difference in performance achieved by all examined topologies when the solver-calculated solution components are and are not updated after a dynamic change event. Intuitively, successive changes compound the difference between the initial and current problem states, meaning the relative utility of the initially-optimal TSP and KP solution components decreases over time. Quantifying these impacts helps to illustrate the difficulty of applying methods that exploit problem information for dynamic instances.

4.8.4 Evaluation of Multi-population Island Topology with Migration (Results)

Impacts of Information Relevance Degradation (Currentness)

The hypervolume profiles in Figure 4.12 illustrate the impacts of resolving and non-resolving the TSP and KP components after each dynamic change event, separately illustrated for each of the model topologies. To clarify, the suffix '-NR' refers to the practice of post change event seeding using the initial solver-based solutions for the TSP and KP components. Regular practice sees the seeding of the population, as previously, with diversified solutions based on the resolved or repaired solution components for the TSP and KP parts of the problem - this means the seeding solution information is kept up-to-date for the new problem state.

For all three of the population models, there is a clear difference in the hypervolume achievement when using more relevant seeding information; higher hypervolumes are unanimously achieved after each change event when the TSP and KP components are updated.

Over the course of the optimization, the magnitude of the difference between the median achieved hypervolumes for the *Si* and *Si-NR* models appears loosely to grow. A similar pattern presents for the *Mo* and *Mo-NR* methods. Interestingly, the difference between the *Mi* and *Mi-NR* median hypervolumes are smaller than for the other models. This implies that the focused optimization, migration and mixing present within the base *Mi* model topology may alleviate some of the difficulty introduced by not updating the seeding information.

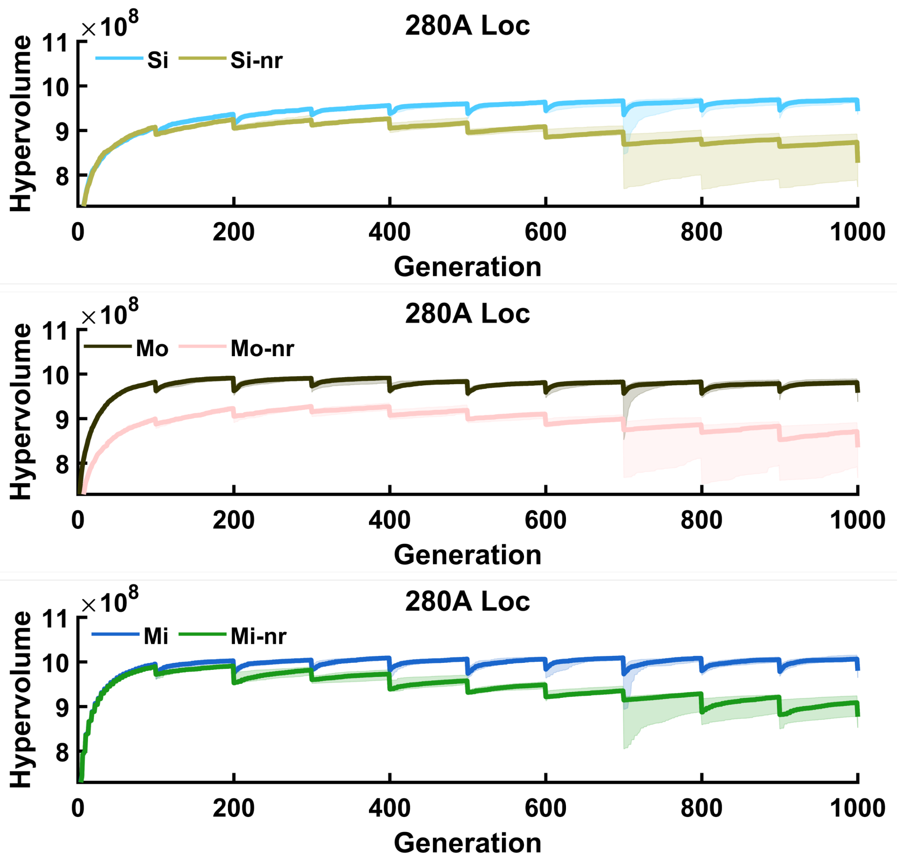


Figure 4.12: Composite median (median of ten pattern-means) hypervolume profiles for the 280A problem with city location changes, comparing the resolving and non-resolving variants of the different population models. (upper) The median hypervolume for the *Si*, single population model with resolving of minimum time tour and maximum profit solution components for responsive seeding is compared to the non-resolving single population model *Si-nr* (using initial state information only). (middle) The median hypervolume profile of the multi-population model, *Mo*, is compared to the non-resolving version of this model *Mo-nr*. (lower) The median hypervolume profile of the multi-population model with migration, *Mi*, is compared to the non-resolving version of this model *Mi-nr*.

A summary of the proportion of generations with significantly higher hypervolumes between the models and their '-NR' counterparts is presented for a selection of problems in Table 4.3. For the *Mo* and *Mi* models we can see that significantly higher hypervolumes are achieved on every problem when resolving the problem components, with many having significant improvements in upwards of 95% of generations. For the *Si* model, the problems with the largest KP components, the C-type, see a large proportion of generations with significantly higher hypervolumes occur when not updating the seeding information. These results informed the remainder of the experiments: the seeding solutions were generated based on resolved or repaired TSP and KP solutions. As the comparisons in Table 4.3 are between only each model and its non-resolving variant, there is no specific disadvantage on the *Si* from adopting a resolving approach for the remaining experiments (and is demonstrated in the following results).

Table 4.3: Percentage of generations with significantly higher hypervolume measurements: comparison resolving and non-resolving problem component optima for responsive population seeding. Bold values indicate largest value in each comparison.

	Si vs Si-nr		Mo vs Mo-nr		Mi vs Mi-nr	
berlin52A	**0.614**	0.083	**0.939**	0.000	**0.493**	0.076
kroA100A	**0.574**	0.003	**0.893**	0.000	**0.457**	0.004
eil101A	**0.890**	0.004	**0.999**	0.000	**0.872**	0.009
pr144A	**0.793**	0.034	**1**	0	**0.846**	0.002
a280A	**0.904**	0.002	**1**	0	**0.870**	0.000
lin318A	**0.503**	0.003	**0.994**	0.000	**0.357**	0.019
berlin52B	**0.672**	0.020	**0.994**	0.001	**0.420**	0.012
kroA100B	**0.427**	0.134	**0.971**	0.000	**0.463**	0.011
eil101B	**0.649**	0.065	**0.982**	0.000	**0.756**	0.008
pr144B	**0.382**	0.170	**0.978**	0.002	**0.480**	0.001
a280B	**0.875**	0.000	**0.961**	0.000	**0.901**	0.001
lin318B	0.295	**0.379**	**0.922**	0.003	**0.866**	0.001
berlin52C	0.030	**0.648**	**0.989**	0.000	**0.111**	0.046
kroA100C	0.053	**0.610**	**0.959**	0.001	**0.644**	0.004
eil101C	0.037	**0.752**	**0.966**	0.002	**0.846**	0.000
pr144C	0.256	**0.367**	**0.947**	0.002	**0.685**	0.001
a280C	**0.866**	0.000	**0.946**	0.004	**0.855**	0.003
lin318C	0.052	**0.550**	**0.879**	0.002	**0.829**	0.000

Comparison of Population Topologies

Figure 4.13 shows the hypervolume profiles achieved by the three population topologies across the 280-city DTTP variants with *A*, *B* & *C* type KP components. The data series correspond to the median across 10 patterns of dynamic changes of the mean hypervolume attainment, calculated from 30 repetitions on each pattern. It should be noted that the global population size for all models was equated to reduce the inequality in function evaluations from model-specific processes.

Generally, the profiles indicate the clearer separation of hypervolume achievement from the 280A to 280C instances. This implies that the complexity of the KP-component impacts the performance population models acutely; the single population model becomes less suitable as more items must be considered.

We can see from the *280A* plot that there is relatively similar hypervolume attainment by all three models, however the *Mi* model (multi-population with migration) achieves a higher hypervolume consistently after the first dynamic change event. The *Mo* model achieves progressively lower hypervolumes compared with the *Si* and *Mi* model as the optimization progresses and the dynamic changes compound. The apparent similarity of achieved hypervolumes is distinguished using the statistical analysis provided in Table 4.4.

A greater distinction in achievement between the topologies presents for the 280B problem, with the *Mi* model achieving greater hypervolumes than *Mo*, which in turn achieves greater hypervolumes than the *Si* model. There is little to no overlap between the different model's achievement for this problem.

For the *280C* problem, there is some overlap in the performance achieved by the *Mo* and *Mi* models within the first few dynamic intervals. However, after successive dynamic change events, the *Mi* model consistently achieves greater hypervolumes. The *Si* model does not produce competitive hypervolume measurements with the other population models. A summary of these interaction can be obtained through the statistical analysis described previously and is given in Table 4.4.

Table 4.4 shows the results of the pairwise comparison of hypervolume measurements taken in each generation across the set of dynamic patterns and for each combination of the population models. The first columns represents the proportion of generations in which the *Si* model achieves significantly higher mean hypervolume than the *Mo* model. The second column represents the inverse of this comparison: the proportion of generations in which the *Mo* model achieves significantly higher mean hypervolume over the *Si* model. The remaining pairs of columns follow this format for pairwise comparison. The bold values indicate the highest value in each column pair.

For the A-type problems in the upper section of Table 4.4, the *Si* model can provide a comparable proportion of generations with significantly higher hypervolumes on a limited range of instances compared with the *Mo* model. Compared with the *Mi* model, only on the *101A* problem does the *Si* model achieve significantly higher hypervolumes in approximately 30% of generations. Across the same sample of generation measurements, the *Mi* model achieves significantly greater hypervolumes in 25% of generations. This highlights a consistent challenge in reporting performance on DMOPs; the rich temporal behaviour of algorithms cannot effectively be represented in a single value without some loss of information. As an ongoing research topic, the effective and meaningful communication of dynamic performance remains a major difficulty within the domain. On these A-type problems, the *Mi* model performs significantly better in up to 50% of generations than the *Mo* model.

The results for the B-type problems are shown in the middle section of Table 4.4 and across all the problems here, the *Mo* and *Mi* models provide significantly higher hypervolumes in at least 60% of generations when each is compared with the *Si* model. For some problems in this section, the *Mi* model can achieve significantly better hypervolume in more than 90% of generations – from the profiles in Figure 4.13 we can deduce that the remainder of the generations without significant differences must be early in the optimization.

When comparing the performance for the *Mo* and *Mi* models on the B-type problems (the final pair of columns in the middle section), the *Mi* model achieves significantly higher mean hypervolumes across most generations for some of the problems (*280B* & *318B*), a moderate proportion for others (*100B* & *101B*) and for the remainder, the *Mo* model achieves a larger proportion of generations with higher hypervolumes. Further

Table 4.4: Percentage of generations with significantly higher hypervolume measurements: comparison of different population models. Bold values indicate larger value in each comparison.

	Si vs Mo		Si vs Mi		Mo vs Mi	
berlin52A	0.230	**0.258**	0.060	**0.560**	0.036	**0.530**
kroA100A	**0.151**	0.001	0.045	**0.154**	0.011	**0.278**
eil101A	**0.556**	0.134	**0.303**	0.252	0.085	**0.234**
pr144A	0.263	0.231	0.055	**0.469**	0.035	**0.392**
a280A	0.323	0.320	0.029	**0.594**	0.069	**0.400**
lin318A	0.037	**0.498**	0.064	**0.523**	0.041	0.074
berlin52B	0.029	**0.884**	0.057	**0.650**	**0.579**	0.060
kroA100B	0.234	**0.603**	0.143	**0.637**	0.123	**0.217**
eil101B	0.103	**0.736**	0.072	**0.901**	0.103	**0.488**
pr144B	0.038	**0.856**	0.036	**0.867**	**0.164**	0.099
a280B	0.046	**0.892**	0.013	**0.970**	0.002	**0.983**
lin318B	0.052	**0.874**	0.008	**0.978**	0.013	**0.940**
berlin52C	0.031	**0.953**	0.009	**0.965**	**0.396**	0.058
kroA100C	0.053	**0.871**	0.010	**0.958**	0.042	**0.504**
eil101C	0.039	**0.955**	0.010	**0.981**	0.062	**0.494**
pr144C	0.064	**0.831**	0.013	**0.952**	0.083	**0.591**
a280C	0.063	**0.902**	0.008	**0.956**	0.131	**0.677**
lin318C	0.115	**0.834**	0.006	**0.986**	0.000	**0.988**

development of the *Mi* model may consolidate its superior performance for this type of problem.

These comparisons for the C-type problem (rightmost columns in the lower section of the table) are more straightforward. For the instances with at least 100 cities, the *Mi* model achieves significantly higher hypervolumes in at least approximately 50% of generations. Only on the *52C* problem does the *Mo* model achieve a higher proportion. Compared to the *Si* model, the *Mo* and *Mi* models achieve significantly higher hypervolumes in at least 83% and 95% of generations respectively.

These results can be clarified in terms of a relational summary of performance across the examined problem set. Using the 280A profile in Figure 4.13 and the corresponding row in Table 4.4 we can establish a relational ordering of the topology performance. In the hypervolume profile we can see that for some of the optimization the *Si* and *Mo* are better than each other, with a swapping after the fourth dynamic change. The *Mi* topology consistently achieves the best hypervolume. Relating these observations to the 280A row of Table 4.4, we see that there are similar values for the *Si* and *Mo* models and that the *Mi* model has higher values than both. This gives rise to the relation: $Si \approx Mo < Mi$. A summary of these relations and their frequency across the problem set is given in Table 4.5. The majority of the problems see the *Mi* model achieve the best results.

4.9 Summary of Findings for the Dynamic Travelling Thief Problem

The Dynamic Travelling Thief Problem, like its static counterpart, represents a challenging problem. The quantification of this challenge is also difficult as the Pareto Set and Pareto Front remain unknown, meaning we must make comparative analyses on the per-

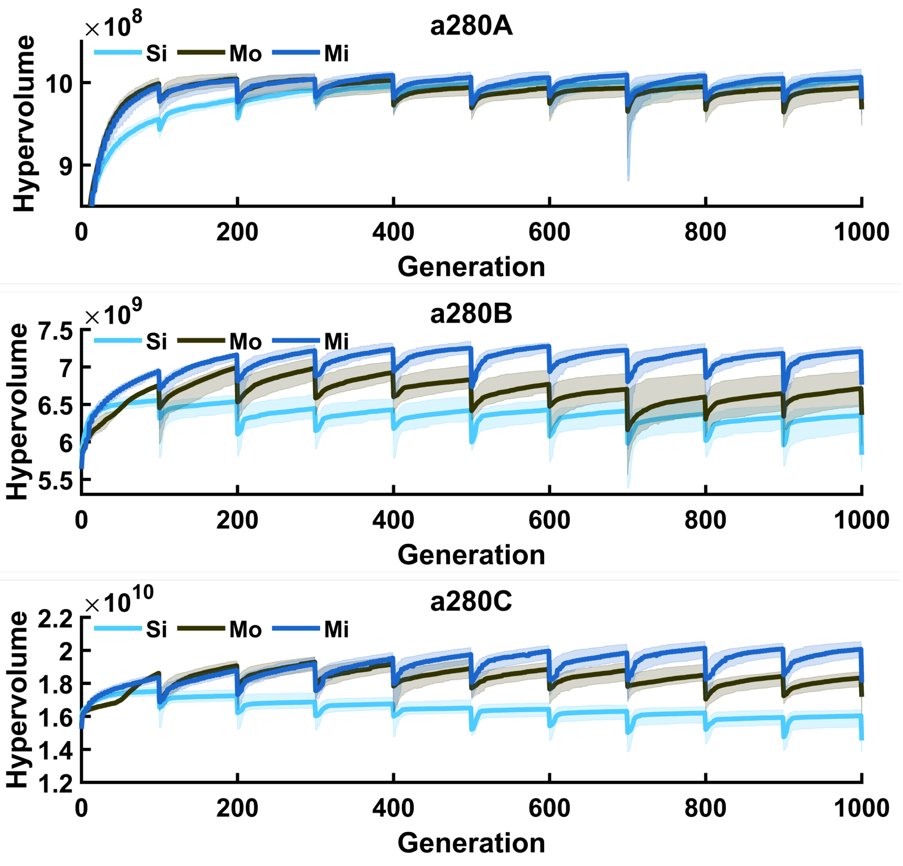


Figure 4.13: Composite median (median of pattern means) hypervolume profiles for variants of the 280-city DTTP problem with city location changes with magnitude $d_N = 2$. (upper) The 280A problem with median hypervolume profiles, together with the interquartile range (IQR), for the *Si*, *Mo*, and *Mi* models. (middle) Median hypervolume profiles IQR for the *Si*, *Mo*, and *Mi* on the 280B problem. (lower) Median hypervolume profiles with IQR for the *Si*, *Mo*, and *Mi* models on the 280C problem.

Table 4.5: Performance relation orderings for population topologies based on the proportion of generations with significantly higher mean hypervolumes.

Performance Relation	Frequency
$Si < Mo < Mi$	11
$Si \approx Mo < Mi$	2
$Si < Mi < Mo$	3
$Mo < Si < Mi$	1
$Mo < Mi < Si$	1

formance of applied algorithms. This work has provided a number of these; for the definition of an evidenced mechanism for dynamic response; in comparison to adapted heuristics for the static problem; and in the cogent implementation of multi-population topologies to drive the search for better solutions.

The initial results detail the observation of the difference in objective space coverage achieved when the initial solution set contains a variety of solution information. This notion is adapted to a variety of solution construction mechanisms using a variety of solver-based information, greedy and randomized solution components for the TSP and KP parts of the problem. A comprehensive range of combinations of this information were tested as dynamic response mechanisms in an attempt to mitigate the impacts of changes and drive evolution of good solutions after a change event.

Whilst seeding appears to be an effective response mechanism, across the sampled problem space, both diversity in the seeding set and the inclusion of solver based information can be useful. The response mechanism with the maximal diversity and variety in combination of solution information (the *mC* response) retains good performance as the size of the problem grows, whereas the good performance of the other responses remains competitive mostly on the problems with smaller TSP components.

A combination of the interconnectedness of the TTP components and the performance of the randomized seeding responses indicate that comparison with off-the-shelf DMOEAs may not provide competitive results. Similarly, many of the heuristics for the static TTP have be custom-built to handle the complexity of the problem. Therefore an adaptation of these heuristics for the dynamic problem provides an insightful comparison into the attainment, utility and efficiency (in terms of both time and evaluations) of the proposed EA with seeding (*SeedEA*) approach for DTTP instances.

Finally, development of multi-population topologies to drive isolated evolution and sharing of fitness through migration events, provides improvements across many of the examined DTTP cases with the changing city location dynamics. Demonstration of both the importance of updating the seeding information and for the inclusion of migration events between sub-populations is evidenced in the presented results. The use of a statistically observant technique to extract and distill the behaviour of algorithms throughout the dynamic optimization can be applied beyond this work.

There are a number of challenges that have been highlighted in the experiments on instances of the DTTP. These are limitations that raise additional questions for future research to address. They include the difficulties of reporting absolute performance when the Pareto Front is unknown and the large number of possible ways of executing change events and the impact this has on generalizing reported performance.

4.10 Conclusions

A novel framework for generating dynamic problem instances is defined using events inspired by realistic dynamics intuitively present in real world problems. The flexibility of the problem parameters and the dynamics parameters provides the tools to extensively test algorithms on a range of challenging problems which are more indicative of realistic scenarios than some existing simplistic benchmarks in the dynamic combinatorial and multi-objective problem space. Three types of dynamics are proposed, each with specific justifications for its inclusion and each with distinct impacts visible via performance measurements.

An observance of the link between the specific solution information provided to a typical dominance based multi-objective evolutionary algorithm and the achievable set of non-dominated solutions provided the basis for a dynamic response mechanism. By seeding the solutions with solution information for the tour and item-selection components that make the Travelling Thief Problem, instead of a random or default DMOEA response, allows for an effective amelioration of the impacts of the dynamic change events. The results indicate that across a variety of construction methods and a range of problem sizes relevant to previous literature, both diversity and solver-based solution information enable better solutions to be found after dynamic change events.

From previous literature on static problems, performance comparison with a number of adapted heuristics highlights the relative efficiency and better performance of the responsive seeding approach. Further development using a multi-population model with migration, developed through an iterative observation of solution localization and exploitation of objective biasing, indicates that there is significant opportunity to find better solutions through informed design.

The presentation of results in the experiments conducted here addresses a significant challenge in the dynamic and dynamic multi-objective optimization area. The provided methodologies and the inclusion of statistical comparisons using meaningful samples of data provide a widely applicable and accessible way to consolidate a large volume of results for these types of problem.

A novel dynamic multi-objective problem type in the combinatorial domain is proposed here, together with fundamental and developed versions of a responsive dynamic algorithm and accompanying contextual justifications, comparative performance analysis and methods to provide statistical evidence.

Chapter 5

Multi-Dynamic Optimization Problems: New Challenges and Potential Approaches

5.1 Introduction

Dynamic problems in research draw motivations from realistic cases and real world problems that feature time-varying characteristics. For example, the numerous continuous benchmarks detailed in Chapter 3 are formulated to model characteristics in a simplified problem setting. Similarly, the Dynamic Travelling Thief instances proposed in Chapter 4 provide an example of problem formulation based of characteristics of real systems. However, in many real cases, for example in energy dispatch problems where the renewable generation and the demand can both be dynamic; a single, synchronised schedule of change events is a naive approximation of the true scenario. Formulation of test cases of real-world scenarios remains an ongoing research task [38], [269].

A characteristic so-far undocumented in EMO, is multi-dynamic problems; that is dynamic problems with multiple components that are independently (of each other) dependent on time. The 'components' could be any of those noted within Chapter 2; the number of decision variables, the number of objectives or elsewhere in the problem. However, to provide preliminary insights on this novel problem class, the focus here is on multiple dynamic terms in the objective functions of multi-objective problems.

Whilst some existing DMO benchmark problems already feature multiple terms that are dependent on time t, multi-dynamic problems must have separate t values, one for each time-dependent term. Since the value and schedule of t are how change events are implemented, this also means that each distinct t value has its own associated frequency and severity parameters.

Across both continuous and combinatorial problem domains there are similarities and differences which affect algorithm evaluation and possible conclusions. The consideration of both domains highlights the disparity in available tools for experimentation as well as the difference in the challenges that arise when incorporating these realistic characteristics.

The proposed methods in Chapter 3 for establishing a baseline of performance across the possible dynamic instance space of continuous benchmarks are extended here to identify the key challenges for multi-dynamic instances of these problems. Similarly, the impacts of frequency and severity parameters for the DTTP and the efficacy of the responsive seeding algorithm for instances with a combination of multiple types of changes provides insights for the combinatorial domain. Using the specific examples examined, the limitations and challenges for continuous and combinatorial problems are explored. A discussion of these challenges highlights their wider applicability and the limitations of existing methods for effectively reporting on these novel problem instances. Recommendations for multi-dynamic problems are provided based on the experiments conducted, particularly concerning experimental design and measurement recording.

Multi-dynamic problems therefore present a novel and difficult challenge for established practices within DMO. The contributions can be summarised as follows:

- A prefatory analysis of the impacts of frequency and magnitude of change is provided for the Dynamic Travelling Thief Problem with city location changes.
- Instances of the DTTP with multiple dynamic components with coincidental change events are defined and the effectiveness of the responsive seeding mechanisms designed in Chapter 4 are re-evaluated.
- Further instances of multi-dynamic TTPs are considered with non-coincidental changes (different change frequency between components) providing illustrative examples of significant challenges present in multi-dynamic problems.
- Adaptation of existing continuous dynamic multi-objective optimization problems to provide relevant examples of multi-dynamic problems from which instances can be generated.
- Identification of the observable motifs in terms of performance across the possible dynamic instance space are given using the defined continuous instances.
- Consolidation of the major challenges facing continuous multi-dynamic instances including, measurement difficulties are given using instances with non-coincidental change schedules.
- Recommendations for experimental design and future investigations are provided.

The remainder of this chapter is organised as follows. A sample of pertinent background information and relevant areas of EMO research is provided in Section 2, together with an illustrative example to demonstrate the utility of multi-dynamic problem instances. Section 3 focuses on the impacts of frequency and severity in the DTTP as well as co-incidental changes, including definitions and results of different response methods defined in Chapter 4. Section 4 handles the non-coincidental changes, examining the impacts and observations from components with different change frequencies. The extension of this investigation to continuous problems is given in Section 5. Finally, a summary of recommendations, limitations and future considerations is presented in Section 6 with conclusions in Section 7.

5.2 Background

There is limited applicable previous work addressing multi-dynamic problems - they present a novel challenge for the field. There are numerous areas from which practices may be borrowed or insights can be shared however. For clarity, the multi-dynamic problems that are the focus of this investigation are problems containing multiple, independent types of dynamic change, for example there is more than one term within the objective functions of the problem that is dependent on t. Multi-component and interconnected problems, meaning problems with distinct aspects such that a solution is composed of multiple parts, provide a parallel in terms of static examples. A good example for these types of problems is the previously studied Travelling Thief Problem. Compositional problems commonly solved by co-evolutionary algorithm techniques [365] and bi-level optimization studies [366]–[368] may also provide some relevant similarities for which the contributions of this investigation may apply.

In terms of multi-dynamic problems however, there are some unique challenges that have limited similarity to any other class of problem. With the definition or separation of independently changing components of a problem, there is introduction of additional parameters. Importantly, the parameters that define the nature of the changes, the frequency and severity have been shown in Chapter 3 to have significant impacts on the difficult of problem through the combination of different values to generate different dynamic instances. With multiple dynamic components, this expands the possible dynamic instance space exponentially with each additional discrete parameter or infinitely if they are considered to be continuous (e.g. real time frequency or continuous severity).

There are limited references to multiple frequencies of change in the literature. The benchmark suite defined in Biswas et al. [80] has probabilistic selection of multiple t variables, however these are not considered as multi-dynamic instances. Yazdani et al. [39] makes reference to the characteristics of real-world problems having continuously changing environments and Chen et al. [71] notes non-fixed frequency and severity parameters for a vehicle refueling scenario. The work of Richter and Dietel [369] suggests single-objective problem instances where the constraints and the objective functions change asynchronously.

Some other research areas have used similar language to describe different types of problems or evaluation methods. For example, distributed and parallel algorithms with asynchronous evaluations use multiple populations with fitness evaluation and updating at different times [370], [371]. The Artificial Bee Colony algorithm proposed by Akay et al. [372] compares synchronous and asynchronous updating of algorithm mechanisms. These share the definition of asynchrony and the notion of complex temporal characteristics with dynamic optimization however they are mostly focused on single-objective problems and are not directly relevant to the investigations undertaken here.

Recent comprehensive work have been conducted on instance space analysis (ISA) for static multi-objective problems [373], [374]. Definition of a multitude of characteristics across many problems, known as features, help to define the instance space and allow for insights to be drawn based on the footprint of algorithm performance across the feature space [375]–[378]. A parallel can be drawn with the approach taken in these works; the dynamics parameters are used to define the space and the footprint is measured similarly as the performance over different regions of this space. Extension of the established effective and comprehensive ISA frameworks to dynamic and dynamic multi-objective optimization problems remains a open research area. Whilst the dynamic intervals of a problem (periods between change events) can be treated as separate problems, there are ‘meta-impacts’ from the adjacency of different problem states (consecutive intervals) as well as the impacts from the parameters controlling the dynamics, the frequency and severity of changes. Therefore, the extension of ISA techniques to multi-dynamic problems is a step further removed and an informed and justified investigative approach used for ISA is the key influence in the presented methodology.

Before examining real world problems however, understanding the key challenges and obstacles presented by increasing the scope of dynamic formulation is required. Therefore a simplified model is necessary to pioneer the exploration of this problem class. Below in Equation 5.1, is an example of a simplified multi-dynamic system to formally illustrate the distinction from previous DMOP benchmarks.

$$\begin{aligned} \vec{x} &= [x_1, x_2, \dots, x_n]; \\ f_1(\vec{x}, \mathbf{t_1}) &= 1 - g(\vec{x}, \mathbf{t_1}) \\ f_2(\vec{x}, \mathbf{t_2}) &= h(\vec{x}, \mathbf{t_2}) \end{aligned} \tag{5.1}$$

where $\vec{x}$ represents the decision vector as a single solution to the problem, formulated as the objective functions f_1 & f_2. The t_1 and t_2 parameters here represent the dynamic state variable as calculated in Equations 5.2. The sub-functions g and h represent the specific calculations of the objective functions and in this simplified model are considered as separate across the objective functions. Similarly, there may be (potentially dynamic) equality or inequality constraints present for the problem but these are omitted for clarity.

$$t_1 = \frac{1}{n_{t_1}} \left\lfloor \frac{\tau}{\tau_{t_1}} \right\rfloor ; \qquad t_2 = \frac{1}{n_{t_2}} \left\lfloor \frac{\tau}{\tau_{t_2}} \right\rfloor ; \tag{5.2}$$

The consideration of multiple change components within the same problem is a representation of realistic characteristics that present in real world scenarios. Before the comparative analysis of DMOEAs can be undertaken for multi-dynamic problems, the fundamental properties and challenges must be determined. The following investigations detail the application of structured experimentation to highlight the key challenges for both combinatorial and continuous multi-dynamic instances. Identification of the key types of multi-dynamic instances in terms of co-incidental and non-coincidental changes provides insights into the difficulty of experimental design and the complexity of the possible instance space.

5.3 Combinatorial Multi-Dynamic Problems: Coincidental Changes

The results presented in Chapter 3 detail the impacts of frequency and severity for a number of continuous benchmarks, however similar investigation for combinatorial instances remains. For the Dynamic Travelling Thief Problem (DTTP), the novel definition of multiple types of changes lends itself to the consideration of multi-dynamic instances by simple combination of the dynamic components. However, for clarity the introductory experiments presented in Chapter 4 dealt with instances with fixed frequency and severity in order to determine an effective dynamic response strategy.

The results presented in Figures 5.1 and 5.2 illustrate the impacts of different frequency and severity parameter values on the DTTP. Here instances with location changes are examined and the performance measurements are recorded for the Responsive Seeding Algorithm presented in Chapter 4.

There are several observations to be made from Figure 5.1, both within each of the subplots and across them. Within each subplot, each group of boxplots corresponds to the distribution of hypervolume measurements recorded for identical change events but with a varying number of generations between the events from $\tau_t \in [20, 200]$. Each subplot shows the same measurements for a different value of the severity parameter, $n_t \in [1, 10]$. Change Index 0 corresponds to the first change event after the initial *dynamic onset delay* which is constant across all series at 50 generations. This serves as a control check and illustration of the variability in performance as the distributions of all series should be similar for this index.

The key observations are as follows:

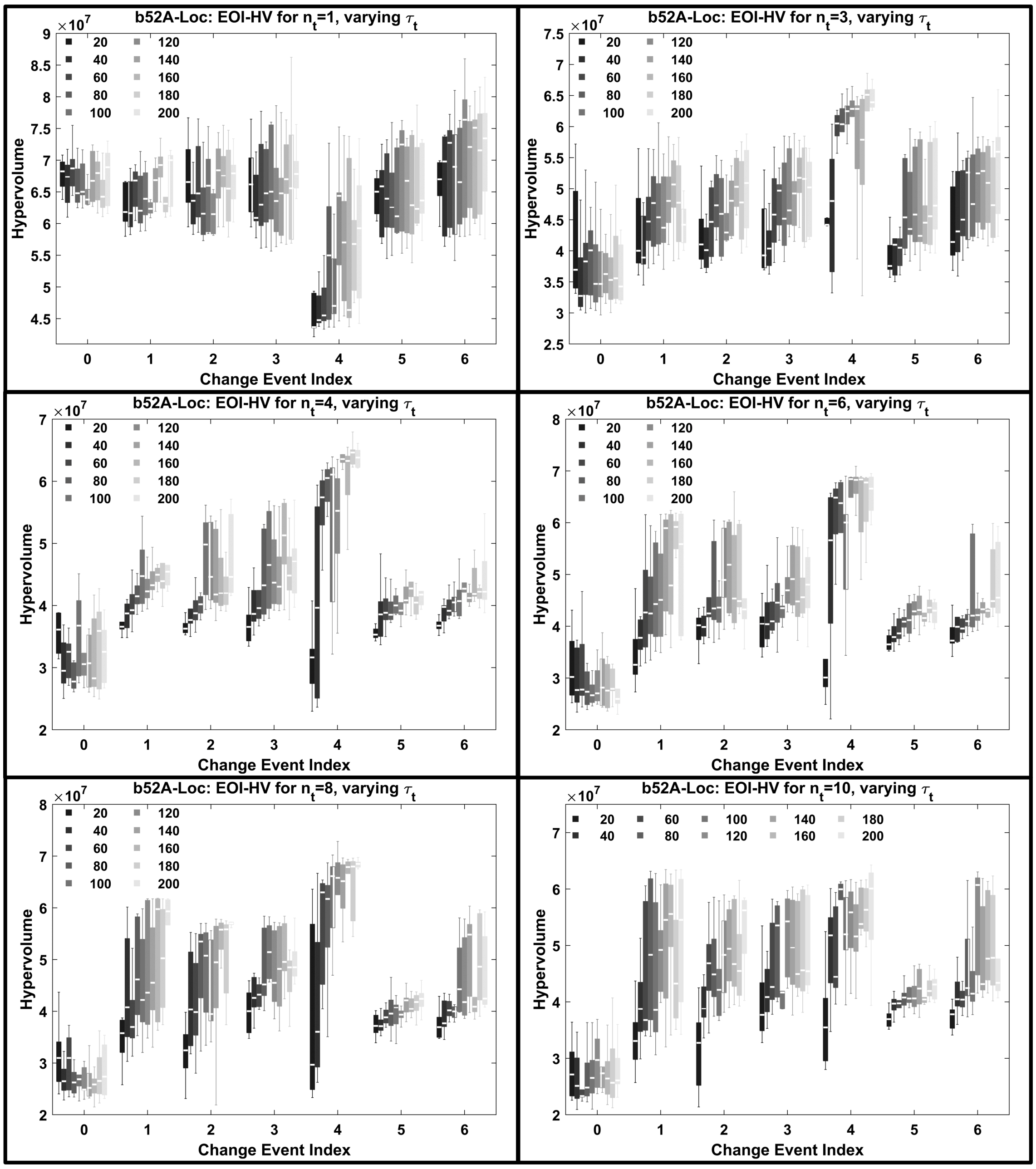


Figure 5.1: Impacts of different frequency parameter values on the 52A DTTP instance with location changes. Each panel is a different severity value (n_t=1,3,4,6,8,10) and each group of boxplots corresponds to one change event. The individual boxplots correspond to the distribution of hypervolume measurements at the end of each dynamic interval for each of value of the frequency parameter τ_t between 20 and 200.

1. In the first subplot, the hypervolume measurement distributions for the $n_t = 1$ setting, there are similar distributions despite different frequency parameter values. As the change index increases, there is a slight separation between the distributions for the fastest and slowest change measurements.

2. Between $n_t = 1$ and $n_t = 2$ there is a clear increase in the hypervolume as the τ_t parameter increases; lower medians are observed when the changes are rapid ($\tau_t = 20, 40$) than for the slower occurrences ($\tau_t = 200$). This trend becomes

more apparent with successive change events (as the change index increases) and when value of n_t increases in the subsequent panels.

3. Across the subplots showing results for $n_t = 1, 3, 4, 6\&8$, change index 4 appears to have a strong impact on the problem, more clearly highlighting the difference in achievable hypervolume when the change frequency is rapid (black boxplot) or slower (lightest grey boxplot). This impact is not apparent in the final panel showing $n_t = 10$ results, possibly as the 9^{th} and 10^{th} changes at each event index have a similarly strong impact.
4. Generally, these impacts are expected: slower changes result in better measurements than rapid changes; the optimizing algorithm has more time to find good solutions. The difference in this capability is exaggerated as the severity of each change event is increased.

Alternatively, the severity parameter impacts can be visualized for fixed frequency as in Figure 5.2. Here, each subplot corresponds to the measurements recorded at a particular change frequency from $\tau_t \in [20, 200]$ and each series of boxplots within these corresponds to a different severity parameter value. The black series are the distributions of hypervolume measurements for the Responsive Seeding Algorithm applied to an instance with change events with severity $n_t = 1$ (meaning one city location is changed at each event). The lightest grey series in each subplot corresponds to the highest severity parameter setting examined, where $n_t = 10$ city locations are altered at each change event. Unlike the previous figure, we cannot expect a similar distribution for change index 0 across the different data series. The key observations from these plots are given below:

1. Within the first subplot, there is a clearly visible decrease in the achieved hypervolume as the severity of each change is increasing. This is illustrated by the median and spread of the measurement distributions decreasing within each group of boxplots at each change event.
2. The succession of change events, which are additive has little visible effect besides a narrowing of the distribution for the highest severity series at the later change indexes.
3. In the subsequent subplots, a similar pattern to the first subplot is present; the highest hypervolume measurements consistently recorded for the instance with the lowest change severity. However, the margin between the distributions for the highest and lowest severity settings is smaller as the frequency parameter τ_t, increases (longer time between changes) throughout the subplots.
4. As in the previous figure, change index 4 has a varied impact on the measurement distributions potentially indicating a performance sensitivity to changes in particular city locations, or to the concatenation of changes particular to the examined pattern of change events.

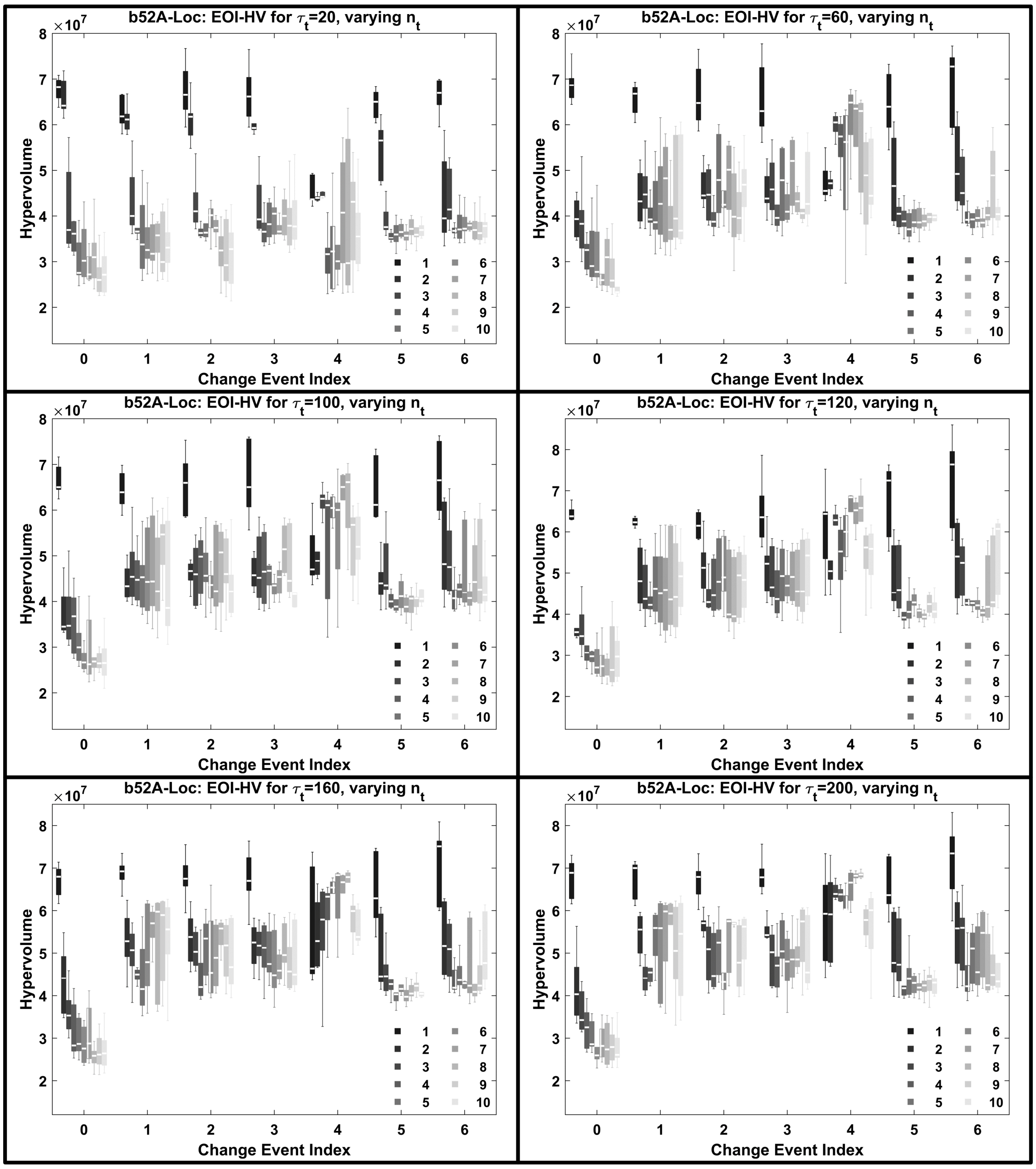


Figure 5.2: Impacts of different severity parameter values on the 52A DTTP instance with location changes. Each panel is a different frequency value (τ_t=20,60,100,120,160,200) and each group of boxplots corresponds to one change event. The individual boxplots correspond to the distribution of hypervolume measurements at the end of each dynamic interval for each of value of the severity parameter n_t between 1 and 10.

5. The illustrated impacts are expected: lower change severity results in higher hypervolume attainment, compared with the higher severity change events. These impacts are attenuated as the frequency of change events decreases.

These results are illustrative of the impacts of frequency and severity, confirming the expectation of differences in these parameters. Due to the disparate problem states the result from problem instances with different severity of change parameters, statistical testing is omitted for these results. Together with the scale of the potential dynamic instance space, caused by the number of unique ways of executing dynamic changes, statistical testing offers little additional insights in this case.

Again, the specific pattern of changes considered is illustrative rather than the comprehensive approach adopted in Chapter 3. The unknown Pareto Front for the static TTP prevents the summary of absolute performance across the multiple problem states in the DTTP instance. Nevertheless, frequency and severity are important in the combinatorial case as in the continuous case.

The previous chapter's results provide the insights into the impacts of frequency and severity parameters on the DTTP with city location changes. To consider the Multi-Dynamic instances of the DTTP, the dynamic components must be combined; the city location changes, the item availability changes and the item value changes can be incorporated into the same instances. Figure 5.3 provides a summary of the different responsive seeding architectures proposed in Chapter 4 for each combination of the dynamic change types (city **Loc**ations, item **Ava**ilability and item **Val**ues). The rankings for the hypervolume and Zitzler's Maximum Spread metric (MS) are reported for a range of instances between 52 and 783 cities and with 1, 5 or 10 items per city (*A,B* & *C* types respectively). Zitzler's Maximum Spread metric [291] is used here to provide additional information about the achieved solution sets using each responsive seeding method. In addition to the HV measurement as a metric of general solution set performance, the MS metric measures the distribution of the obtained solutions and can indicate the better coverage of the high-profit region of the objective space. Calculation of MS for the bi-objective problem is given in Equation 5.3 and is the length of the diagonal hyperbox between the non-dominated solutions in the extremes of each objective [291].

$$MS = \sqrt{(f_1^{max} - f_1^{min})^2 + (f_2^{max} - f_2^{min})^2}$$

where, for example, f_1^{max} corresponds to the maximum fitness value in the f_1 objective belonging to a non-dominated solution.

These results provide an interesting and relevant insights for the investigation of multi-dynamic problems. The extension of the established methodology for testing efficacy of response solution construction highlights the persistence of good methods in the multi-dynamic case. However, these results correspond to instances where the severity is fixed for both components and the change events are co-incidental, meaning that the changes, for example, to the city locations and in the item availability, occur in the same generation. It is important to consider this as a subset of multi-dynamic cases and comparing the performance of response strategies using the context of the previous single-dynamic instances confirms the expectation of performance trends. These can be summarised as follows:

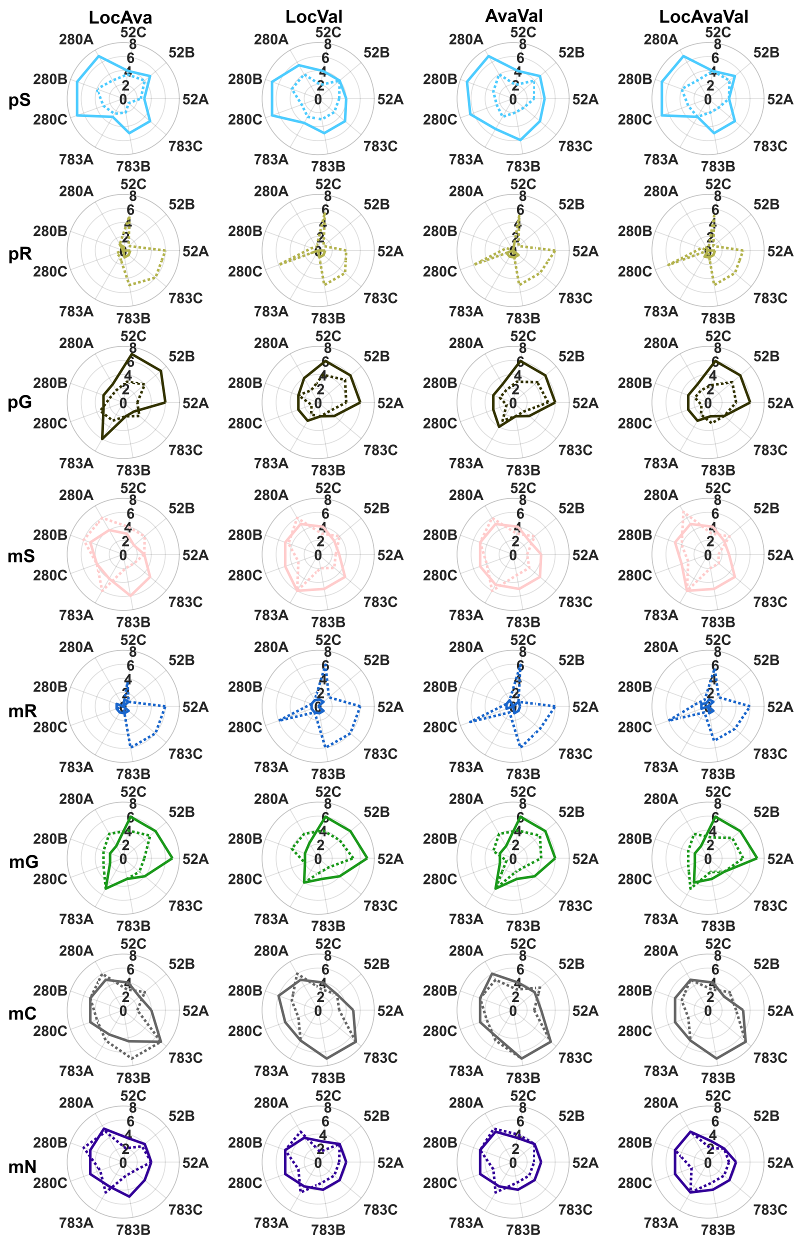


Figure 5.3: Polar plots for responsive seeding algorithm with different combinations of solutions of coincidental changes at fixed frequency and severity. Each column of plots corresponds to a different combination of the dynamic components (eg. LocAva: Location changes and Item Availability changes). Each row of plots corresponds to a different responsive seeding method, with *S, R, G, C* respectively referring to **S**olver-based, **R**andom, **G**reedy, **C**ombined construction of seeding solutions sets. *N* is the algorithm with **N**o seeding response to change events and the prefixed *p* and *m* correspond to the packing plan construction method (see Chapter 4 for more details on these methods). The solid lines each polar axis correspond to the ranking compared to the other seeding response methods, of the median hypervolume on a particular instance. The dashed lines are the ranking for the spread metric [291].

.

1. The random responses *pR* and *mR* show poor performance across the instances of different size (radial ticks) and across the different combinations of dynamic changes (columns). the largest instances have a high ranking for spread, but with a poor hypervolume rank, this means these solutions are not competitive with those produced by the other methods.

2. The purely solver based method which builds a population from a solver-based tour and a dynamic-programming based packing plan achieves high ranks on the 280 city instances across all combinations of changes, and when availability and item value are combined on some of the 783 city instances. However, these high hypervolume ranks are accompanied by poor spread rankings. Previous observations have shown a poor coverage of the high-profit region of the objective space (including solutions with long tours), when using only solver-based information and this is reflected in the multi-dynamic cases too. The more diverse *mS* strategy achieves better spread rankings but worse hypervolume rankings across the instances and the dynamic combinations.
3. The greedy-based methods *pG* and *mG* have good performance on a limited range of instances, as noted in the single dynamic case in Chapter 4.
4. Across all of the methods, there are very similar patterns of rankings between the different combinations of dynamic change types. This indicates that the performance of different response methods is not strongly coupled to the types of changes that are occurring and that the established hierarchy of responses from the single-dynamic cases persists here. The results presented are comparative, meaning the relative impacts between the different combinations of dynamic changes are not explicitly investigated here. However, the similarity in the performance profiles of the different seeding responses highlights the similarity between the co-incidental multi-dynamic cases and the single-dynamic cases for the DTTP.
5. The most diverse seeding method, *mC* performs the best on the largest instances achieving similar hypervolume and spread rankings across the different combinations of dynamic changes. There is close competition on the smallest instances with some of the other seeding methods, however the *mC* seeding method, (also referred to as the SeedEA algorithm) remains a robust choice for the multi-dynamic instances of the DTTP where the change events are co-incidental.

However, the combination of multiple dynamic components in this way, such that the change events occur simultaneously could be viewed as an increase in the total magnitude of the change, spread across changes in different components of the problem. This effectively reduces the co-incidental case to a single-dynamic problem instance; changes occurring at different frequencies is therefore a major consideration for multi-dynamic problems.

5.4 Combinatorial Multi-Dynamic Problems: Non-coincidental Changes

The more complex scenario, where the dynamic changes in each component are non-coincidental (meaning they occur at different frequencies or on different schedules) is a more realistic scenario for this class of problems.

In order to effectively navigate the space of multi-dynamic problems with non-coincidental dynamic changes some understanding of the impacts of the dynamic changes is required. The impacts of frequency and severity are discussed for continuous problems in Chapter 3, with a comprehensive assessment of baseline performance across the possible dynamic instance space. It is important to understand a key difference between combinatorial and continuous instances that makes the direct application of the same methodology impossible.

5.4.1 Experimental Challenges for Problems with Non-Coincidental Changes

The representation of combinatorial problems commonly follows the form of a permutation of indexes or a binary string denoting the selection and non-selection of indexes. In the case of the DTTP considered here, a solution is comprised of both of these components, a tour and a packing plan (item selection component). The disparity between continuous and combinatorial instance spaces is not exclusively linked to either one of these representations, but is common to both.

Scale of the Possible Dynamic Instance Space

When describing the nature of dynamics in a problem, the frequency (τ_t) and the severity (n_t) are the most consistently employed and are used to provide some level of comparability or relation between experiments. However, in combinatorial problems, the execution of a change event is different to continuous problems. As seen in Chapter 3, the magnitude of a change is presented to the dynamic aspects of the problem through the t variable. In combinatorial problems, the objective functions may depend on t in exactly the same way, however more commonly some aspect of the problem is altered, such as the city locations or the distance matrix (or additionally the availability map or the item values for the DTTP). For example, the number of possible unique ways to execute a city location dynamic change with a magnitude of n_t is: ${}^{N}C_{n_t}$, where N is the number of cities in the problem and C is the combinatoric 'choose' operator. For a 100 city problem and $n_t = 2$, this is equal to 4950 possible problem states after this change. Consideration of successive changes in a dynamic problem increases the scale of the possible dynamic instance space even further. This remains an open problem for which there is not an apparent solution yet, however possible areas for further research into this issue are addressed at the end of this chapter.

Reproducibility and Statistical Evidence

The previous experiments presented in Chapter 4 attempted to mitigate this impact by taking a median of performance measurements across 30 different patterns of changes. This provides an estimate of the distribution of problem states in terms of their difficulty by taking a sample of the possible states and patterns that result from changes. There

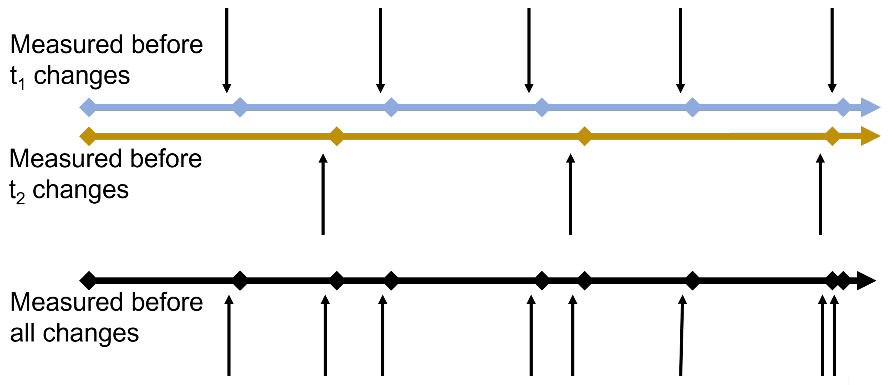

Measured before t_1 changes
Measured before t_2 changes
Measured before all changes

Figure 5.4: Illustration of possible performance measurement recording schedules in a problem with multiple dynamic components (blue and orange lines) experiencing non-coincidental changes (demarcations on each horizontal line). (upper) The relatively faster changing dynamic component generates more measurements than the relatively slower changing component (middle), however each of these sets of recordings are equally space whereas in the combined measurement schedule (lower - after any change) the time between measurements is uneven.

is an additional reproducibility issue, such that in order to provide statistical evidence, repetitions must be performed on the same pattern such that an additional level of experiments is necessary and the seeds/pattern generation method should be reported together with results. These limitations are acutely applicable to multi-dynamic problems with non-coincidental changes as there are a greater number of distinct problems states than in an instance with coincidental changes.

Performance Measurement Challenges

In problem instances with non-coincidental change events, even the measurement of performance becomes more difficult. A common way to record measurements that provide insight into the achievements of an optimization algorithm is to take a measurement recording immediately before a change event. Often, the mean of these (often normalized) pre-change measurements are reported as a single value of performance. The extension of this established practice to non-coincidental multi-dynamic instances is not straightforward. Figure 5.4 highlights the potential approaches to performance measurement in these scenarios.

The key observations that can be made are as follows:

1. There are unequal interval lengths when measuring before every change. This means incomparable dynamic interval lengths and therefore averaging of measurements may not be appropriate.
2. Measuring before only one change may not effectively capture the optimizing algorithms behaviour as changes in the other component may impact perceived efficacy from measurements.
3. A solution to the correct measurement method is unclear, besides recording on both change event timescales separately and reporting both.

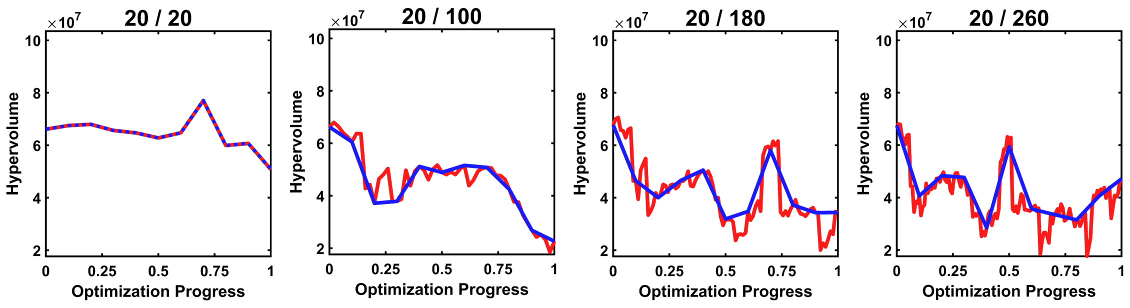


Figure 5.5: A sample of mean hypervolume measurement profiles (10 repeats) for instances of the multi-dynamic DTTP with non-coincidental changes. The sample of instances have increasing change frequency from left to right with the change frequencies displayed above each subplot as (item availability / city location) respectively. Within each panel the red line is mean hypervolume measurements before city location changes and the blue line is mean hypervolume measurement before item availability changes.

4. If the Pareto front is known and the measurement can be represented as a percentage of perfect attainment, then a single-value can be reported. However in the DTTP, as the exact Pareto Front is unknown and this makes absolute performance reporting difficult.

The final point in the observations above highlights that this issue is not limited to combinatorial problems, but continuous multi-dynamic problem instances may face similar challenges.

5.4.2 Non-coincidental Changes in the Dynamic Travelling Thief Problem

To provide a specific example and illustrate these identified challenges, an instance of the DTTP with non-coincidental city location changes and item availability changes is considered. Figure 5.5 shows the hypervolume measurements recorded after each change event as two separate series across four instances with different levels of disparate frequency between the dynamic changes.

The difference in the frequency of change between the plots in Figure 5.5 is noted in the title of each axis; in the first panel the changes are co-incidental as the frequency of change is the same so a single line for the hypervolume measurement before changes is visible. In the second panel, the frequency of the location changes remains consistent at 20 generations and the measurement before each change is shown by the red line. The same pattern of item changes is executed at a lower frequency and the hypervolume measurements before each change are shown by the blue line. As the total number of generations is increased due to the lower frequency of change, the x-axis of each plot is normalized for clarity. A greater disparity between the frequency of changes is visible in the third and fourth panels.

The key observation that can be made from comparing these results is that each of the dynamic components has a strong impact in the achievable hypervolume; despite the same number and pattern of item availability changes, the city location changes have sufficient (cumulative) impact to alter the solution space and therefore the achievable hypervolume in each case. Beyond this however, the previously mentioned difficulties limit concrete conclusions from being made. Unknown Pareto Fronts for any of the problem states in any DTTP instance and unreconciled measurement protocols mean the presented results are comparative rather than absolute. The results therefore are mostly illustrative of the significant challenges that multi-dynamic combinatorial problems present for established methodologies in optimization.

We can observe a more comprehensive range of disparate frequencies in multi-dynamic DTTP within Figure 5.6. General observations can be based on holistic appreciation of the differences across the considered dynamic instance space here. For example, the lower right hand corner of the figure shows plots with lower frequencies in both dynamic components, whilst the first row and column show the most extreme disparities with the most rapid changes in one component being compared with increasing slower changes in the other component. When comparing the lower right plots on the diagonal with those around them, there are smaller differences between the two measurements compared with the noisy deviations noted in the more extreme cases. Whilst this highlights that lower (or more generally that relatively similar) frequencies of non-coincidental changes may provide less of a challenge as there are fewer problems states, the deviations from the measurements in the coincidental case (the diagonal plots) are significant enough to alter the achievable hypervolume and result in different measurement profiles. This highlights the importance of this problem class generally; even in this case with relatively low severity of change, slight differences in the frequency of change of dynamic components in a problem can have a noticeable impact compared with the co-incidental case. This means that the simplification of real-world multi-dynamic problems to a case with co-incidental changes may not be appropriate and obtained results may not be useful or relevant for the true case.

5.4.3 Summary of Challenges for Combinatorial Multi-dynamic Problems

The presented results for the multi-dynamic DTTP instances with non-coincidental changes showcase three major challenges for combinatorial examples of this problem class. Each of these challenges defines an additional research direction for which the solutions are not immediately apparent.

Firstly, the recording of measurements is made difficult as the existing practice is not directly transferable when there are multiple changing components in the problem. Measurements taken immediately before change events or averaged across an optimization are inconsistent in the length of dynamic intervals compared to established reporting procedure. Further work can address reconciling different approaches to measurement and their relative advantages for multi-dynamic instances.

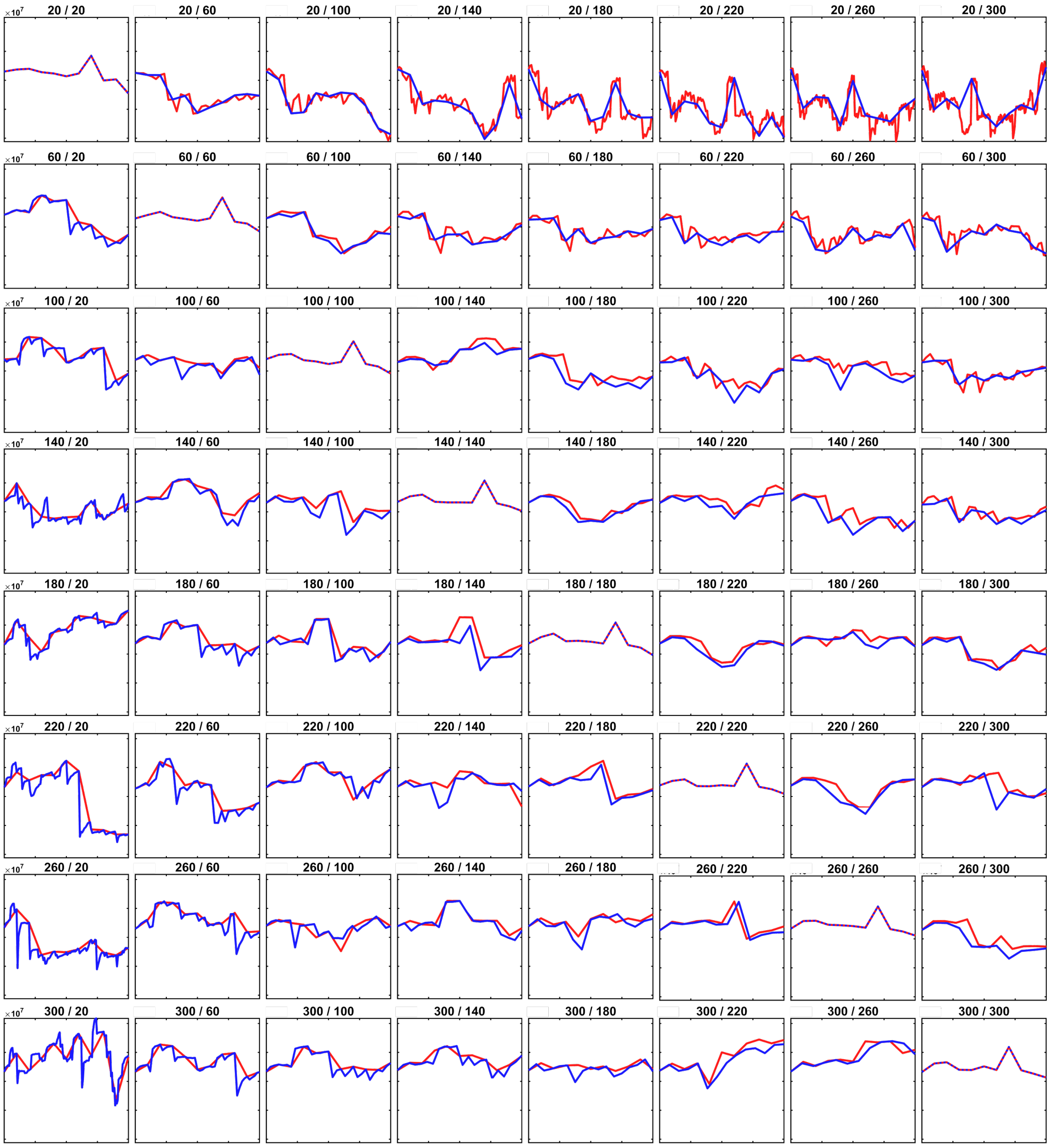


Figure 5.6: Composite of hypervolume measurements from instances with different frequency parameters for non-coincidental city location and item availability changes in the DTTP problem. Each panel corresponds to the measurement profile of hypervolume measurements for the combination of change frequency values above it. The plots are arranged such that the coincidental cases are on the left-to-right diagonal and the off-diagonals correspond to the non-coincidental cases with more rapid frequency in one of the two dynamic components. Within each plot, the two data series correspond to the measurement of hypervolume before changes in each of the components; (red) city location dynamics & (blue) item availability changes. The most rapid changes in both dynamic components can be seen in the upper left of the figure, whilst the instances with slower changes are in the lower right of the figure.

Secondly, in cases where the Pareto Front and/or Pareto Set are unknown, measurements cannot be compared to an achievable maximum, meaning that unless comparative, results risk having limited relevance, wider applicability strength of conclusions. Pareto front estimation methods or approximated Pareto-optimal solutions, for both the static and dynamic TTP, if consistently adopted, may allow for more coherent experimental progress in this area.

Finally, the scale of the potential dynamic instance space is a challenge in single-dynamic problems due the number of unique ways to execute each change event in a combinatorial representation. This scale is therefore magnified further when multiple components are changing independently. Methodologies for defining representative sets of instances or relevant combinations of changes may collapse the extreme scope of possible instances into a more manageable and useful scale.

The first, and to a lesser extent the second of these challenges, are also significant challenges for continuous problems. Whilst the size of the dynamic instance space is also larger considering different frequencies of change and severity parameters, the scale of this expansion is smaller than with the combinatorial examples.

5.5 Continuous Multi-Dynamic Problems

This section has two sets of experimental results that attempt to better identify the challenges presented by multi-dynamic instances of continuous benchmark problems. The first attempts to extend the established methodology from Chapter 3 to provide an illustration of algorithmic performance across the dynamic instance space generated by considering different change frequencies between dynamic components in the same problem. The second part of the results highlights the effectiveness of some existing DMOEA algorithm methods on a subset of instances. These instances are selected through an attempt to parameterise the relative difference in the frequency of change as a way to condense the possible dynamic instance space for feasible experimentation.

The experiments, as previously established for combinatorial instances, highlight the severe challenge that realistic incorporation of multiple dynamic components poses to existing procedures and experimental protocols. A summary and discussion of these challenges, together with recommendations across both combinatorial and continuous domains is provided at the end of this chapter.

5.5.1 Examples of Continuous Multi-Dynamic Benchmark Problems

To define continuous multi-dynamic problem instances existing DMOP instances are developed that naturally allow for the separation of a single dynamic change into multiple dynamic changes. Three examples from the experiments conducted in Chapter 3 are extended here; the JY2 [29], dMOP2 [76] and ZJZ [79] problems. The equations for these problems are given below: note the occurrence of two time variables t_1 and t_2 within the objective functions.

Multi-dynamic JY2 (Type II):

$$\begin{aligned} f_1(\mathbf{x}, t_1, t_2) &= (1 + g(\mathbf{x_{II}}))(x_I + A_t\text{sin}(W_t\pi x_I); \\ f_2(\mathbf{x}, t_1, t_2) &= (1 + g(\mathbf{x_{II}}))(1 - x_I + A_t\text{sin}(W_t\pi x_I); \\ g(\mathbf{x_{II}}) &= \textstyle\sum_{x_i \in \mathbf{x_{II}}}(x_i - G(t_1))^2; \\ G(t_1) = \text{sin}(0.5\pi t_1); \quad A_t &= 0.05; \quad W_t(t_2) = \lfloor 6\text{sin}(0.5\pi(t_2 - 1)) \rfloor; \\ \mathbf{x_I} = (x_1)\ ; \quad x_I \in [0,1]\ ; \quad & \mathbf{x_{II}} = (x_2, \ldots, x_n); \quad x_{II} \in [-1,1]; \end{aligned} \tag{5.3}$$

Multi-dynamic dMOP2 (Type II):

$$\begin{gathered} f_1(\mathbf{x_I}) = x_1; \quad f_2(\mathbf{x_{II}}, t_1, t_2) = gh \\ g(\mathbf{x_{II}}) = 1 + 9\textstyle\sum_{x_i \in \mathbf{x_{II}}}(x_i - G(t_1))^2; \quad h = 1 - \left(\frac{f_1}{g}\right)^{H(t_2)} \\ G(t_1) = sin(0.5\pi t_1); \quad H(t_2) = 0.75sin(0.5t_2) + 1.25; \\ x_i \in [0,1]; \quad \mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \ldots, x_n) \end{gathered} \tag{5.4}$$

Multi-dynamic ZJZ (Type II):

$$\begin{gathered} f_1(\mathbf{x_I}) = x_1; \quad f_2(\mathbf{x_{II}}, t_1, t_2) = gh \\ g(\mathbf{x_{II}}) = 1 + \textstyle\sum_{x_i \in \mathbf{x_{II}}}(x_i - G(t_1) - x_1^{H(t_2)})^2; \quad h = 1 - \left(\frac{f_1}{g}\right)^{H(t_2)} \\ G(t_1) = sin(0.5\pi t_1); \qquad H(t_2) = 1.5 + sin(0.5\pi t_2) \\ x_I \in [0,1]; \quad x_{II} \in [-1,2]; \quad \mathbf{x_I} = (x_1); \quad \mathbf{x_{II}} = (x_2, \ldots, x_n) \end{gathered} \tag{5.5}$$

In each of these cases, t_1 is contained within the G component of the problem definitions which determines changes in the Pareto-optimal set. In the original definitions of these problems, there are multiple occurrences of t, therefore a natural opportunity to separate these incidences into multiple dynamics presents itself. The t-values, as in single-dynamic problems, correspond to a change in the problem state and signal the start of a new dynamic interval. However, in these cases, since there are multiple occurrences of t, there are multiple prompts to begin a new dynamic interval. As mentioned previously, the introduction of multiple independent dynamic changes within a problem creates difficulties in applying previous practices for recording performance measurements. Figure 5.4 illustrates this problem for combinatorial instances, however the same issues arise in the continuous case. Previously measurements are recorded in the generation or time step immediately before a change to report an algorithms best capability in each of the dynamic intervals of the problem instance. However, in the multi-dynamic case, changes in which component should prompt measurements to be taken? If both changes prompt measurements, this results in unequal dynamic interval lengths which may skew or invalidate average values for performance. This remains an open problem and the presented results highlight the possible observations when using different measurement schedules.

In line with the results presented earlier in the chapter, coincidental changes with different severity parameter values for each component can be considered as complex variant

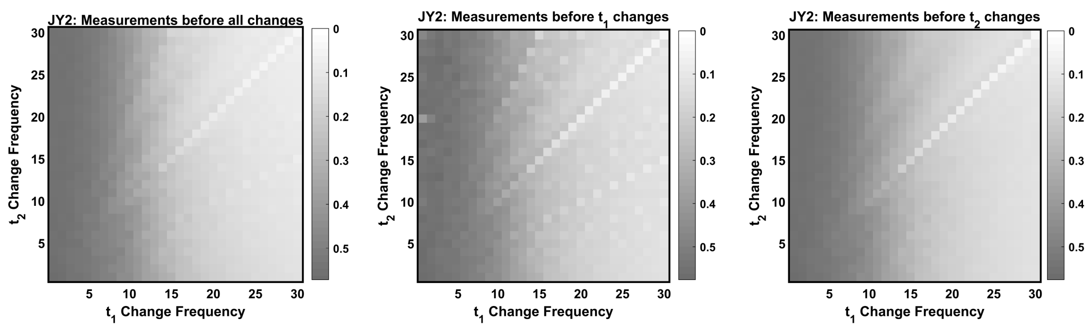


Figure 5.7: Performance of the NSGA-II algorithm across the dynamic instance space for the JY2 problem with varying frequency for each component.

of a single-dynamic problem and therefore the more challenging case of non-coincidental changes (different frequency parameter values between dynamic problem components) is explored for the continuous examples above. Borrowing from the methodology presented in Chapter 3 for visualizing the possible dynamic instance space across different frequency and severity parameter combinations, allows for a similar illustration of the impacts of different frequency parameter values between the dynamic problem components.

5.5.2 Illustrating the Dynamic Instance Space in Multi-Dynamic Continuous Benchmark Problems

To showcase the complex challenge of navigating the continuous multi-dynamic instance space, heatmaps of algorithm performance are generated in a similar way to those presented earlier for the establishment of baseline performance across the possible dynamic instance space for different benchmark problems. A similar outcome of the comprehensive experimentation to the design of future experiments is afforded here, however the distinction lies in the verification and consolidation of previous practices versus the pioneering navigation of the dynamic instance space of a novel class of problems.

Figure 5.7 shows the performance across the dynamic instances created by varying the frequency of change of the t_1 and t_2 parameters. Each of the three panels corresponds to a different measurement protocol: (left) measurements before both changes, (centre) measurements before t_1 changes and (right) measurements before t_2 changes. As before, each cell corresponds to the average hypervolume difference (HVD) across five repetitions each with at least 30 change events in each of t_1 and t_2. This means each cell contains the mean of 150 measurements of changes with the same severity and frequency. The HVD is the difference between the hypervolume of the obtained solution set and the hypervolume of an optimal solution set that lies on the Pareto Front; this means that lower values, depicted by lighter shades of grey in the heatmaps, indicate better performance. Producing heatmaps in this way is both data intensive and computationally expensive, requiring 4500 algorithm runs to produce each one. Nevertheless, the comprehensive approach to exploring the dynamic instance space provides insights that may otherwise be difficult to illustrate. The key observations from Figure 5.7 can be summarised as follows:

1. Visible across all the panels, the diagonal cells represent performance on instances with coincidental changes (meaning the frequency of change is equal for t_1 and t_2. These cells are noticeably lighter than any of the cells around them, indicating that any degree of asynchrony (non-coincidental) in the changes has an impact on the achievement of an otherwise capable optimization algorithm. Rapid change frequencies in both dynamic components (lower right corner of each heatmap) are the most difficult instances, with poor performance regardless of when measurements are recorded (any non-coincidental change is visibly more difficult than coincidental changes).

2. There is a clear left-to-right gradient of poor performance across all frequencies for t_2 and improving at frequencies between 10 and 15 for t_1. This indicates that algorithm performance worsens across all instances with more rapid change frequencies than this in t_1, ignoring the frequency of t_2 changes. This 'motif' in the performance across the instance spaces illustrates two important concepts for multi-dynamic problems. Firstly, there can be 'dominant' dynamic components, such that changes in one do not impact the difficulty (or changes are not perceptible to an optimizing algorithm) when another changes. Secondly, that there are limits to change frequency that determine when this dominance becomes an active factor.

3. The impacts of 'factor frequency' on performance is another visible motif, most clearly seen in the center heatmap. Specifically the cases where changes in t_1 or t_2 are double or half of the other, are visible as lighter cells on the 'half-diagonals' across the instance space. The visualization of this motif confirms that the difficulty of non-incidental multi-dynamic instance is eased when the frequencies of change are factors; as some change events are coincidental, fewer total unique problem states are presented to the optimizing algorithm than if the frequencies are non-factors.

4. The difference in the visibility of the identified motifs across the three heatmaps highlights the difficulties in correct procedure for recording measurements in problem instances with non-coincidental changes. The factor frequency clearly visible when recording measurements before t_2 changes is less pronounced in the other heatmaps. Beyond understanding algorithm behaviour across this novel range of instances, this difference reinforces the importance of developing a robust measurement procedure for this class of problems. Without a cohesive approach to performance measurement, as explained in Chapter 3 in terms of instance se-

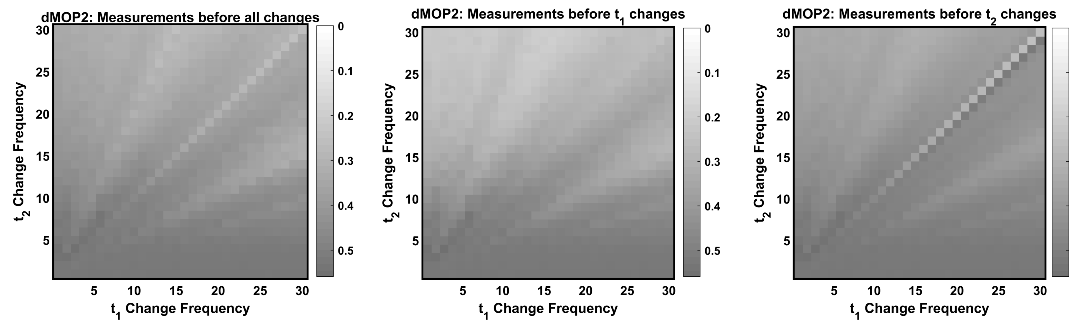


Figure 5.8: Performance of the MOEA/D algorithm across the dynamic instance space for the dMOP2 problem with varying frequency for each component.

lection, there is the potential for conclusions on algorithm performance to lack strength.

For the second of the defined problems, Figure 5.8 shows the performance of the MOEA/D algorithm across the multi-dynamic instances introduced by different change frequencies of the values of t_1 and t_2. As before the severity of change is fixed at $\frac{1}{n_t} = 0.1$ for all instances. The MOEA/D algorithm performs slighty better on this problem compared with the NSGA-II algorithm, making the identified motifs more visible. Building upon those noted for the JY2 problem, the key observations here can be summarised as follows:

1. The dominant component motif is less prevalent across this instance space, however at high frequencies of the t_2 changes, poor performance becomes uniform across instances with any change frequency for t_1. This is opposite to the dominance displayed in the performance for the JY2 problem, indicating that the POS-linked change component does not dominate the instance difficulty in cases with rapid changes.
2. As a result of attenuated dominance motif, the factor frequency motif is more visible here. In addition to the 'half-diagonals' with slight improvements to performance (lighter grey streaks on the 'half-diagonals') there are secondary patterns of improvement when change frequency in one component is one-third of the frequency of the other component.

Generally, the performance across much of the dynamic instance space for these multi-dynamic problem examples is poor. Generally, the introduction of multiple dynamic changes shortens the overall length of each dynamic interval and increases the number of unique problem states during the optimization. This combination, although mimicking the realistic occurrence of this type of dynamic characteristic, visibly increases the difficulty of the problem; the previous reasonable performance on the coincidental cases does not directly extend to the non-coincidental cases.

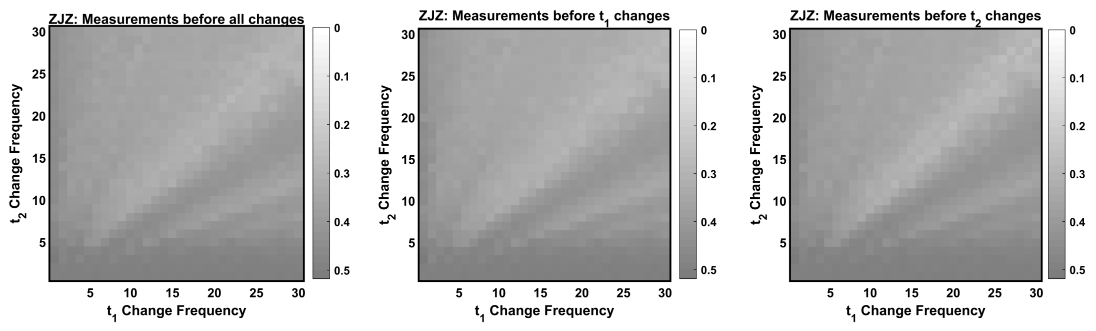


Figure 5.9: Performance of the NSGA-II algorithm across the dynamic instance space for the ZJZ problem with varying frequency for each component.

The final example, the multi-dynamic ZJZ problem, highlights this observation further together with the variability of the identified motifs from the other problems. As before, the severity is fixed at $\frac{1}{n_t} = 0.1$ for all dynamic instances created by the combination of different frequencies of change in t_1 and t_2. The MOEA/D algorithm performs inconsistently across the dynamic instance space for the single-dynamic ZJZ problem, therefore the results reported here are using the NSGA-II algorithm. The key observations from Figure 5.9 are reported below:

1. The consistent poor performance of the NSGA-II algorithm across the multi-dynamic instance space highlights the challenge that this class of problems presents generally. A similar goal identified from the results in Chapter 3, is to apply state-of-the-art algorithms against these obtained baselines to determine comprehensively the utility of more complex and current optimization methods.

2. The dominance over performance of the rapid changes in t_2 (visible across all three heatmaps) is similar to the observation from Figure 5.8. However, the factor frequency motif appears dissimilar. Previously, the slight improvements in performance attributed to the instances with half or third relation between the two change frequencies were symmetrical around the diagonal (the cases with equal frequency and therefore coincidental changes). Here, there are several notable differences. Firstly, when the change frequency of t_2 is half that of t_1, these instances appear to be more difficult (algorithm has worse performance) than the instances closer to the diagonal, coincindental cases. Secondly, when the change frequency of t_2 is third that of t_1, the algorithm's performance is better. Thirdly, there is no clear performance improvement for cases where the frequency of changes in t_1 is half or a third that of t_2. Generally, this further illustrates the complexity of the multi-dynamic instance space when considering non-coincidental changes; the utility of the identified performance motifs are not consistent across problems. This makes navigating the design of experiments a more challenging task for multi-dynamic

problems since the selection or avoidance of particular instances based on the insights from the motifs may not always be possible.

The application of this approach, the visualization of baseline performance across the dynamic instance space, has enabled the identification of these motifs where other methodologies may not have. The comprehensive testing of different combinations of change frequency parameters for the multi-dynamic example problems shows that there are significantly complex impacts between the different dynamic components in these example problems. The heatmaps themselves provide an accessible visualization that facilitates explanation of complex trends across large volumes of data.

The NSGA-II algorithm is an algorithm that was not designed for dynamic multi-objective optimization, much less multi-dynamic instances. However, previous findings rebuke the use of random restart methods and the baseline of achievable performance obtained by NSGA-II provides a more useful starting point for directed future experimentation on these problems.

Future experimentation can also exploit the identified motifs, the factor frequency instances provide a subset of multi-dynamic instances which have partial coincidence of dynamic changes. However, as the results show, the varying characteristics of performance motifs across different problems must be carefully considered in the construction fo test sets in future.

The contributions of these experiments come with a number of limitations. However, many of these are opportunities for future investigation that will enable refining of appropriate techniques for multi-dynamic problems. For example, the results are presented with fixed severity and the focus, as stated, is on differing frequencies of change. The consideration of all possible aspects of these dynamic changes presents even more complex instances that can only be effectively addressed after solutions to the challenges in these simpler cases are resolved. Likewise, there are numerous other possibilities for multi-dynamic instances, including cases where the frequencies of change are equal but non-coincidental. The variety in possible definitions and diverse potential to compose multi-dynamic problems in different ways highlights this class of problems as an area of research that will be significant in the journey towards more realistic dynamic test functions.

The scale of the instance space however may still be infeasible for effectively comparing algorithm performance and therefore a potential way to condense the range of instances is based on the relative frequency of changes. To a limit, algorithm performance depends on time, such that better solutions can be found given more generations or time. Changes occurring on different time scales, or asynchronously, is a general way to describe multi-dynamic instances, however as the presented results indicate, the scale of the instance space precludes straightforward comparisons of performance. Therefore, assessing algorithm performance over a structured set of instances where asynchronously occurring dynamic changes can be described clearly along a scale, may give insights on multi-dynamic problems without the comprehensive experimentation required to produce the heatmaps seen previously.

5.5.3 DMOEA Performance on Instances with Relative Asynchrony

The FDA3 and FDA5 problems [28] provide additional opportunity to define more examples of continuous multi-dynamic benchmark problems. The separate dynamic change events in both problems concern the transformation of the Pareto Front and the change in the spread or density of solutions across the Pareto Front, thus both problems are Type II. The multi-dynamic versions of the equations are given below, note the occurrence of the t_1 and t_2 variables within the objective functions.

Multi-Dynamic FDA3:

$$\begin{gathered} f_1(\mathbf{x}, t_1) = \frac{1}{|\mathbf{x_I}|}\sum_{x_i \in \mathbf{x_I}} (x_i^{F(t_1)}) \qquad f_2(\mathbf{x}, t_1, t_2) = g * h \\ g(\mathbf{x_{II}}) = 1 + G(t_2) + \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t_2))^2; \\ G(t_2) = |\sin(0.5\pi t_2)|; \quad F(t_1) = 10^{2sin(0.5\pi t_1)}; \quad h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}; \\ \mathbf{x_I} = (x_1)\ ; \quad x_I \in [0,1]\ ; \quad \mathbf{x_{II}} = (x_2, \ldots, x_n); \quad x_{II} \in [-1,1]; \end{gathered} \tag{5.6}$$

Multi-Dynamic FDA5:

$$\begin{gathered} f_1(\mathbf{x}, t_1, t_2) = (1 + g(\mathbf{x_{II}}, t_2)) \prod_{i=1}^{M-1} cos(\frac{\pi y_i(t_1)}{2})\ ; \\ f_k(\mathbf{x}, t_1, t_2) = (1 + g(\mathbf{x_{II}}, t_2))(\prod_{i=1}^{M-k} cos(\frac{\pi y_i(t_1)}{2}))sin(\frac{\pi y_{M-k+1}(t_1)}{2})\ ; \\ f_M(\mathbf{x}, t_1, t_2) = (1 + g(\mathbf{x_{II}}, t_2))sin(\frac{\pi y_1(t_1)}{2})\ ; \forall k = 2 : (M-1) \\ g(\mathbf{x_{II}}, t_2) = G(t_2) + \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t_2))^2\ ; \\ y_i(t_1) = x_i^{F(t_1)}; \quad \forall i = 1 : (M-1); \\ G(t_2) = |\sin(0.5\pi t_2)|; \quad F(t_1) = 1 + 100^{sin(0.5\pi t_1)^4}; \\ \mathbf{x_I} = (x_1) \quad \mathbf{x_{II}} = (x_M, \ldots, x_n)\ ; \quad x_i \in [0,1] \quad \forall i = 1 : n; \end{gathered} \tag{5.7}$$

where M is the number of objectives and n is the number of decision variables.

A range of possible problem states for each of the types of changes are illustrated in Figure 5.10 for FDA3 and examples of the different solution densities and PF offsets are shown for FDA5 in Figure 5.11.

Parameterization of Relative Asynchrony

The number of combinations of different change frequency parameters for each of the dynamic components results in a large possible dynamic instance space; the grid of cells presented in the heatmaps represents the comprehensive investigation of this space. Naturally, this type of investigation requires substantive computational and time resources. In order to make experimentation for multi-dynamic problems more feasible, the results of

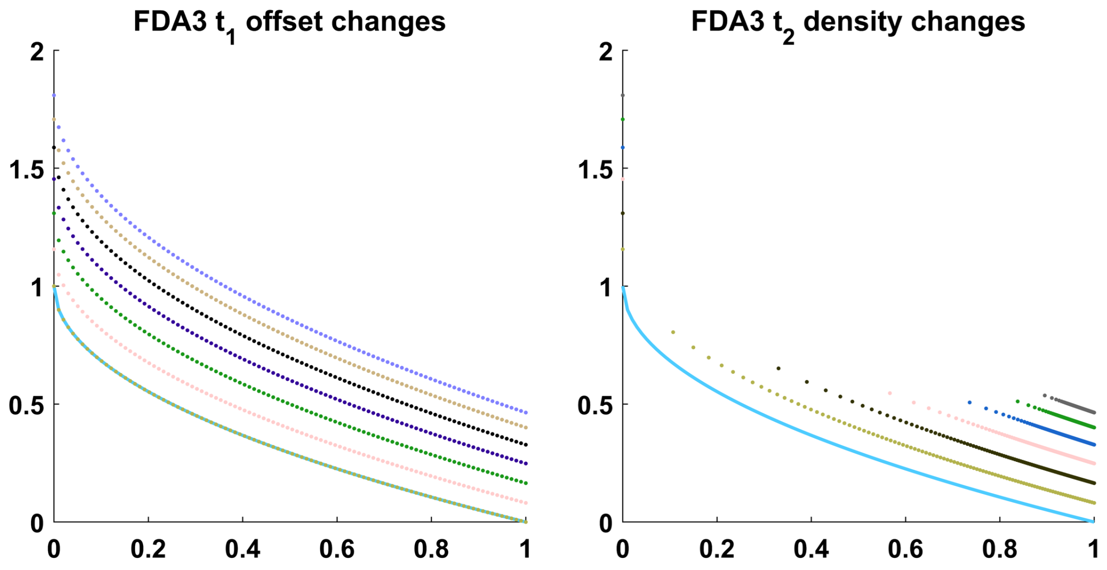


Figure 5.10: Examples of the continuous POF changes in the FDA3 benchmark problem [28] with multiple dynamic components in solution density and offset.

an exploratory investigation into collapsing the possible dynamic instance space based on the relative frequency of changes between components is presented here. The factor frequency motif noted for the multi-dynamic instances in Section 5.5.2 indicates that better algorithm performance can be expected for instances with a natural relationship between the frequencies (partial coincidence) than those with no relation. Therefore a subset of instances with different relative frequencies of change may provide a reasonable test bed for more feasible experimentation across the large instance space of multi-dynamic problems.

Using a parameter A it is possible to define the *relative asynchrony* between the frequencies of change. Additionally, introducing non-fixed frequency in the form of interval length drawn from a distribution with a mean of τ_t instead of using a fixed value is a small step towards more realistic dynamic and multi-dynamic test instances.

$$\begin{aligned} \mu_{\tau_T,t_1} &= 40; \qquad \mu_{\tau_T,t_2} = \tfrac{\mu_{\tau_T,t_1}(e-1)}{(e^{A^{0.435}}-1)}; \\ \tau_{T,t_1} &\sim N(\mu_{\tau_T,t_1},1); \qquad \tau_{T,t_2} \sim N(\mu_{\tau_T,t_2},1); \end{aligned} \tag{5.8}$$

A range of values for the parameter A are selected logarithmically spaced around $A = 1$ with extreme values such that the relative asynchrony between changes is approximately 10 times more often and 10 times less often. Table 5.1 has a summary of the asynchrony parameter values used together. The mean of the distribution of the change frequency (μ_{τ_T,t_1}) of the t_1 variable is held constant at 40 generations, whilst the change frequency of the t_2 as (μ_{τ_T,t_2}) is calculated using Equation 5.8. This formulation was selected such

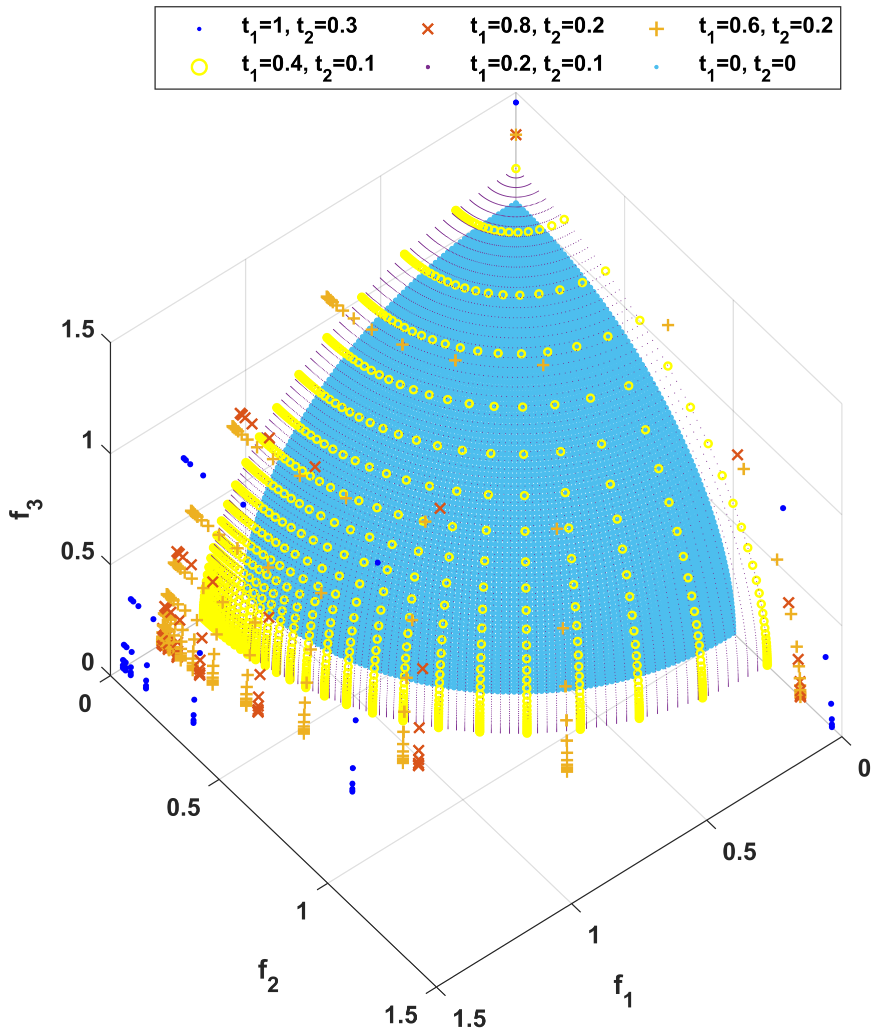


Figure 5.11: Examples of the continuous POF changes in the FDA5 benchmark problem [28] with multiple dynamic components in solution density and offset.

Table 5.1: Asynchrony parameter values and their resultant mean frequency of change for t_2

A	μ_{τ_T,t_2}	A	μ_{τ_T,t_2}
0.01	475.92	1.2589	34.021
0.0167	374.44	1.5849	28.718
0.0278	293.29	1.9953	24.037
0.0464	228.45	2.5119	19.925
0.0774	176.68	3.1623	16.336
0.1292	135.40	3.9811	13.228
0.2154	102.57	5.0119	10.560
0.3594	76.552	6.3096	8.2955
0.5995	56.037	7.9433	6.3982
1	40	10	4.8330

that approximate relative frequencies could be achieved using the desired parameter range for A.

For each value of A a pattern of changes is (reproducibly) generated using the distributions in Equation 5.8. Interval lengths are sampled and rounded to integer generation numbers from the distribution with mean (μ_{τ_T,t_1}) and (μ_{τ_T,t_2}) (and $\sigma^2 = 1$ for both) for t_1 and t_2 respectively. Using these values of A, an instance generated using $A = 0.01$ experiences changes in t_2 approximately 10 times slower than in t_1. Likewise an instance generated using $A = 10$ results in t_2 changes 10 times faster than t_1. For both FDA3 and FDA5, the changes in t_1 and t_2 correspond to changes in the solution density in the objective space and the Pareto Front offset respectively.

DMOEA Algorithms Applied to Asynchrony DMOPs

To provide insights into the suitability of dynamic response mechanisms in line with the previously presented results, six algorithms are applied to the instances with different relative asynchrony of changes. These comprise the unmodified NSGA-II algorithm to provide a performance baseline; NSGA-II with a standard re-initialization technique and four additional algorithms with dynamic responses from the literature.

DNSGA-II-A and **DNSGA-II-B** are commonly employed as comparison methods due to the simplicity of the response mechanism. The performance across the dynamic instance space for DMOP benchmarks was illustrated in Chapter 3 for these algorithms and they are employed here to determine any visible difference in performance attainment from the baseline algorithms in the multi-dynamic case. The mechanism of the dynamic response for the DNSGA-II-A/B algorithms is to replace percentage of the offspring population in the generation of change with (A) randomly generated solutions or (B) mutated solutions. Commonly, 20% of the population is replaced [141] and so the same value is adopted here.

Population Prediction Strategy (PPS) is a prediction-based DMOEA introduced by Zhou et al. [77]. As seen in Chapter 3, PPS could not provide competitive results across the dynamic instance space when given a consistent number of dynamic intervals presented to other algorithms. The nature of the complex change environment in the multi-dynamic case provides an additional challenge and the effectiveness of a prediction strategy such as PPS is useful. In brief, the PPS mechanism reduces the obtained solutions in a given interval to a centroid (in the decision space) and a manifold (in the objective space) and produces an auto-regressive moving average (ARMA) model for each. Using these models, when a change event occurs, a predicted centroid and manifold are used to re-initialize the population potentially closer to the new optimal set.

The **Dy-NSGA-II** algorithm is a hybrid, adaptive population management strategy formulated by Azzouz et al. [53]. It is a more complex DMOEA for continuous instances that shares similarities in terms of structured diversity introduced in response to changes, with the *SeedEA* algorithm proposed in Chapter 4 for the combinatorial DTTP instances . In response to changes, a population is constructed using a combination of methods

Table 5.2: Algorithm parameters and settings for parameterised relative asynchrony experiments.

Parameter	FDA3	FDA5
Population Size	100	250
Number of objectives	2	3
Number of decision variables	20	
Maximum generations	1000	
Maximum Archive Size	1000	
Polynomial Mutation probability	1/n	
Mutation distribution index	5	
Simulated Binary Crossover probability	0.9	
Crossover distribution index	15	
(DNSGA-II-A/B) Replacement percentage	20	
(PPS) Number of previous centroids stored	5	
(PPS) Number of previous manifolds stored	2	
(PPS) AR model order	3	
(Dy-NSGA-II) Polynomial Mutation probability	0.01	
(Dy-NSGA-II) Mutation distribution index	20	
(Dy-NSGA-II) Number of trials in Local Search	50	
(Dy-NSGA-II) Cell density partitioning parameter	10	
(Dy-NSGA-II) Local Search Contribution	0.4*popsize	

to generate solutions. Solutions retrieved from an archive, solutions obtained via a local search and randomly generated solutions are combined to form a new population for the new problems state. The proportion of these contributions stems from an estimation of the change severity.

Performance Measurements and Experimental Setup

Performance of the six algorithms is measured using two measurements. Firstly, the percentage of the maximum attainable hypervolume (%HV) at the end of each dynamic interval is recorded, using a reference set of *popsize* solutions on the POF. Second, the Generational Distance (GD) is calculated for the same generation using a reference set of 10^4 points on the Pareto Front. Therefore, in these experiments, performance measurements are taken before every change event. For both measurements, as in the DTTP experiments the *composite median* is reported; the mean taken within each run, as there are different numbers of measurements for different A values, then the median of these is reported across 100 repeat runs. The combination of these measurements provides information on both the convergence and spread of solutions. The other experimental parameters, including those specific to the different algorithms are stated in Table 5.2.

Statistical comparisons are carried out between the two baseline algorithms (default NSGA-II and NSGA-II with random re-initialization) and each of the other methods. One-tailed Wilicoxon ranksum tests are carried out with $p = 0.01$ to determine significant increases in %HV and significant decreases in GD achieved by the DMOEAs.

Visualizing the profiles of recorded performance measurements on DMOP instances provides much needed context to tables of numerical results. Figure 5.12 illustrates the median hypervolume profile for the random-restart, DNSGA-II-B, PPS and Dy-NSGA-II algorithms across three examples of different values of the relative asynchrony parameter

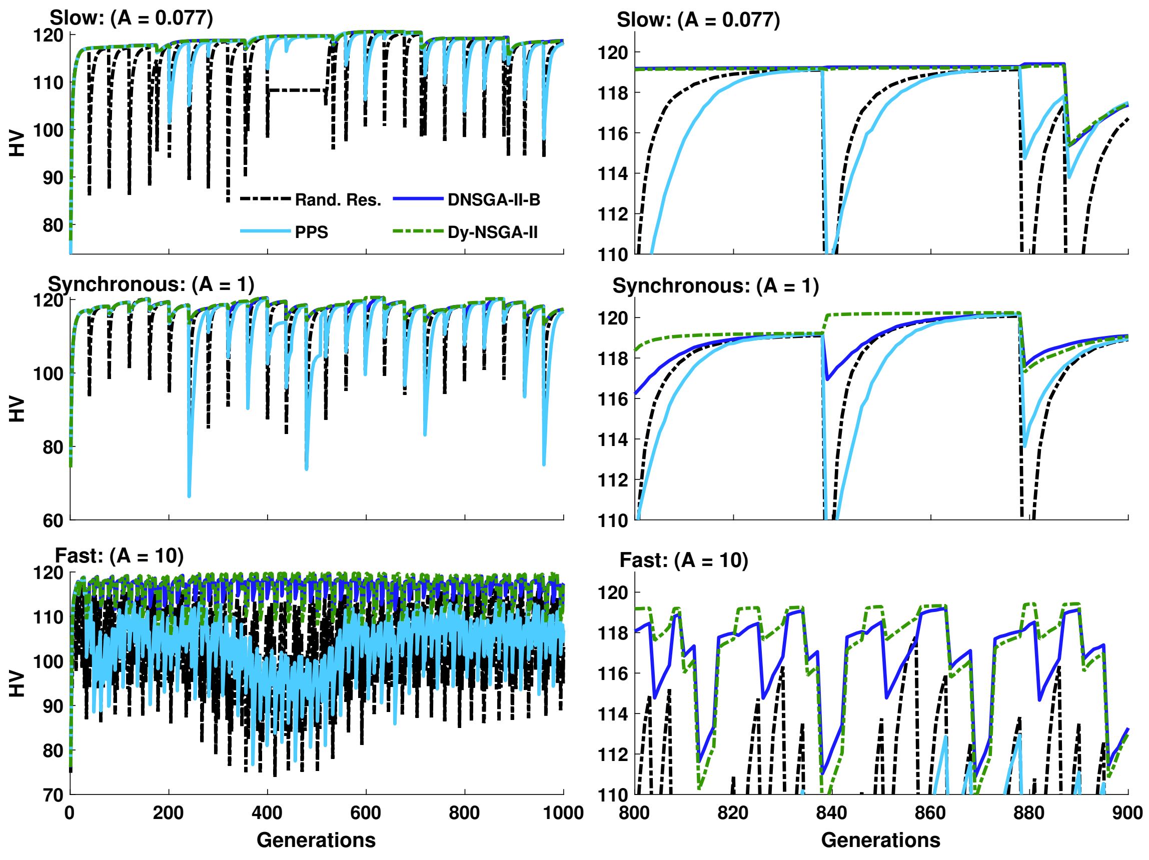


Figure 5.12: Hypervolume profiles for four of the algorithms applied to the FDA3 problem. (top-left) Slow relative asynchronous changes where A=0.077; (top-right) zoomed section of top-left sub-plot between generations 800 and 900. (middle-left) Synchronous case where A=1; (middle-right) zoomed section of middle-left sub-plot between generations 800 and 900. (lower-left) Fast relative asynchronous changes where A=10; (lower-right) zoomed section of lower-left sub-plot between generations 800 and 900.

A on the FDA3 problem. The profiles for NSGA-II and DNSGA-II-A are omitted for clarity; both algorithms almost exactly match the profile of DNSGA-II-B throughout the optimization. There are several key observations that can be made from Figure 5.12:

1. The left column of subplots highlights the direct impact of different values of the A parameter. The default t_1 interval length of 40 generations is visible as the regular downward spiking pattern of the random restart and PPS measurement profiles in the top and middle subplots. The relatively slower changes are more visible in the performance profile of the Dy-NSGA-II algorithm (as the t_1 changes do not appear to have significant impact). In contrast, the instance with the most rapid relative asynchrony, shown in the lower-left panel contains performance profiles that appear much 'noisier' as the changes in t_2 are much more rapid.

2. The right column of subplots show a magnified section of the profiles to illustrate the separation in algorithm performance across instances with different relative asynchrony. The slow relative asynchrony case (small A value) in the upper subplot highlights the performance difference between the algorithms in terms of

the rate of improvement after each change event. The Dy-NSGA-II algorithm achieves the largest hypervolume without a visible impact on performance from the dynamic change events. The DNSGA-II-B algorithm also appears to track the maximum hypervolume but suffers visible impact after some changes. The random-restart method sees large decreases in hypervolume immediately after each change, as does the PPS method, however the random-restart method recovers more rapidly.

3. The middle and lower subplots in the right column illustrate this separation of performance further; the change in the A parameter resulting in the more rapid changes in t_2 relative to the changes in t_1 separate the algorithms by performance. The same order is preserved, with Dy-NSGA-II achieving the best hypervolumes, followed by DNSGA-II-B, then random-restart and PPS. In the fastest relative asynchrony instance (A=10), the Dy-NSGA-II and DNSGA-II-B algorithms achieve similar hypervolume values.
4. These performance profiles more generally help to illustrate three general challenges for DMO research. Firstly, the use of raw hypervolume values to display these hypervolumes highlights the challenge of unequal hypervolume between dynamic intervals. Secondly, the profiles provide additional insights and context compared with the reporting of single values for performance; good performance in some intervals can be offset by poor performance in other when averaging measurements to a single-value. Finally, in the multi-dynamic case, the presented profiles highlight three of the 20 instances that comprise these experiments and in this case the severity is constant. If we consider different severity values, despite the use of the relative asynchrony parameter, the range of possible instances still makes the clear and accessible presentation of results challenging.

The results for all algorithms across a greater range of the instances are given for FDA3 in Tables 5.5.3 & 5.3, and for FDA5 in Tables 5.4 & 5.5. The significant differences are marked within these tables and cell shading is used to illustrate the key features in the results. Results for some values of A are omitted for clarity and do not alter the overall observations.

The following key observations can be made for the results across both problems and are divided into the synchronous, slow relative and fast relative asynchronous cases for clarity:

Synchronous Case ($A = 1$):

- Very little difference between the DNSG-II-A and DNSG-II-B performances for the synchronous case ($A = 1$), on FDA3; both achieve significant improvements over the random-restart method and similar performance in both %HV and GD to the NSGA-II algorithm.

Table 5.3: Median HV% attainment across different values of the asynchrony parameter on **FDA3** (2-objective). Asterisks indicate significant increases over NSGA-II (top row) value and bold values indicate significant increase over the random restart (second row) value (one-tailed Wilcoxon rank-sum, p<0.01). Cell shading illustrates gradient of performance from best (lightest) to worst (darkest) across all obtained values.

(Slow) Asynchrony Parameter Value (A) (Fast)

Algorithm	0.010	0.017	0.028	0.077	0.215	0.359	0.599	1.000	1.259	1.995	2.512	3.162	3.981	5.012	7.943	10.00
NSGA-II	0.9969	0.9977	0.9975	0.9965	0.9964	0.9947	0.9936	0.9966	0.9930	0.9909	0.9915	0.9887	0.9862	0.9843	0.9818	0.9790
Rand. Res.	0.9741	0.9793	0.9813	0.9823	0.9782	0.9722	0.9686	0.9891	0.9663	0.9646	0.9541	0.9604	0.9465	0.9489	0.9326	0.9177
DNSGA-II-A	**0.9969**	**0.9977**	**0.9975**	**0.9965**	**0.9964**	**0.9947**	**0.9936**	**0.9966**	**0.9929**	**0.9908**	**0.9915**	**0.9887**	**0.9861**	**0.9842**	**0.9816**	**0.9788**
DNSGA-II-B	**0.9969**	**0.9977**	**0.9975**	**0.9966**	**0.9964**	**0.9948**	**0.9937**	**0.9966**	**0.9930**	**0.9910**	**0.9916**	**0.9889**	**0.9863**	**0.9845**	**0.9819**	**0.9791**
PPS	**0.9964**	**0.9972**	**0.9968**	**0.9956**	**0.9931**	**0.9844**	0.9699	0.9843	0.9535	0.9467	0.9414	0.9468	0.9331	0.9285	0.9071	0.8846
Dy-NSGA-II	**0.9968**	**0.9976**	**0.9974**	**0.9964**	**0.9966**	**0.9952**	**0.9944**	**0.9969**	**0.9941**	**0.9930**	**0.9931**	**0.9898**	**0.9887**	**0.9862**	**0.9845**	**0.9810**

Table 5.4: Median GD attainment across different values of the asynchrony parameter on **FDA3** (2-objective). Asterisks indicate significant decrease over NSGA-II (top row) value and bold values indicate significant decrease over the random restart (second row) value (one-tailed Wilcoxon rank-sum, p<0.01).

(Slow) Asynchrony Parameter Value (A) (Fast)

Algorithm	0.010	0.017	0.028	0.077	0.215	0.359	0.599	1.000	1.259	1.995	2.512	3.162	3.981	5.012	7.943	10.00
NSGA-II	0.0032	0.0014	0.0014	0.0034	0.0025	0.0060	0.0092	0.0009	0.0114	0.0177	0.0118	0.0246	0.0419	0.0496	0.0789	0.1387
Rand. Res.	0.2768	0.0092	0.0472	0.0404	0.1199	0.4851	0.9288	0.0006	0.9799	0.9075	1.5934	1.2021	2.4493	1.4833	2.0045	2.5888
DNSGA-II-A	**0.0033**	**0.0014**	**0.0014**	**0.0035**	**0.0026**	**0.0061**	**0.0117**	0.0009	**0.0227**	**0.0193**	**0.0131**	**0.0359**	**0.0614**	**0.0680**	**0.0917**	**0.1625**
DNSGA-II-B	**0.0031**	**0.0013**	**0.0013**	**0.0034**	**0.0024**	**0.0057**	**0.0091**	0.0009	**0.0113**	**0.0175**	**0.0115**	**0.0237**	**0.0411**	**0.0488**	**0.0797**	**0.1429**
PPS	**0.0031**	**0.0015**	**0.0013**	**0.0042**	**0.0181**	**0.2128**	**0.2966**	0.0017	**0.8948**	**0.7164**	**0.8474**	**0.5728**	**0.9804**	**0.8683**	**1.3690**	**2.1478**
Dy-NSGA-II	**0.0031**	**0.0012**	**0.0014**	**0.0034**	**0.0021**	**0.0057**	**0.0075**	0.0007	**0.0061**	**0.0149**	**0.0102**	**0.0268**	**0.0417**	**0.0507**	**0.0715**	**0.1431**

Table 5.5: Median HV% attainment across different values of the asynchrony parameter on **FDA5** (3-objective). Asterisks indicate significant increases over NSGA-II (top row) value and bold values indicate significant increase over the random restart (second row) value (one-tailed Wilcoxon rank-sum, p<0.01).

(Slow) Asynchrony Parameter Value (A) (Fast)

Algorithm	0.010	0.017	0.028	0.077	0.215	0.359	0.599	1.000	1.259	1.995	2.512	3.162	3.981	5.012	7.943	10.00
NSGA-II	0.9988	0.9990	0.9990	0.9985	0.9984	0.9980	0.9974	0.9982	0.9967	0.9964	0.9960	0.9954	0.9945	0.9944	0.9937	0.9931
Rand. Res.	0.9962	0.9979	0.9959	0.9967	0.9961	0.9945	0.9932	0.9985	0.9913	0.9913	0.9892	0.9912	0.9805	0.9862	0.9802	0.9770
DNSGA-II-A	**0.9988**	**0.9990**	**0.9989**	**0.9985**	**0.9982**	**0.9966**	**0.9974**	0.9983	**0.9954**	**0.9963**	**0.9961**	**0.9955**	**0.9934**	**0.9945**	**0.9930**	**0.9929**
DNSGA-II-B	**0.9988**	**0.9990**	**0.9990**	**0.9985**	**0.9984**	**0.9980**	**0.9974**	0.9982	**0.9967**	**0.9964**	**0.9961**	**0.9954**	**0.9946**	**0.9945**	**0.9937**	**0.9932**
PPS	**0.9983**	**0.9986**	**0.9986**	**0.9979**	0.9928	0.9944	0.9903	0.9981	0.9703	0.9650	0.9548	0.9708	0.9509	0.9695	0.9539	0.9456
Dy-NSGA-II	0.9955	0.9963	0.9962	0.9952	0.9942	0.9894	0.9922	0.9951	**0.9924**	0.9916	**0.9919**	0.9915	**0.9910**	**0.9909**	**0.9887**	**0.9879**

Table 5.6: Median GD attainment across different values of the asynchrony parameter on **FDA5** (3-objective). Asterisks indicate significant decrease over NSGA-II (top row) value and bold values indicate significant decrease over the random restart (second row) value (one-tailed Wilcoxon rank-sum, p<0.01).

(Slow) Asynchrony Parameter Value (A) (Fast)

Algorithm	0.010	0.017	0.028	0.077	0.215	0.359	0.599	1.000	1.259	1.995	2.512	3.162	3.981	5.012	7.943	10.00
NSGA-II	0.0113	0.0049	0.0051	0.0139	0.0096	0.0262	0.0401	0.0078	0.0535	0.0685	0.0583	0.1074	0.1248	0.1652	0.2214	0.2620
Rand. Res.	0.2589	0.0502	0.2784	0.1497	0.1678	0.2500	0.3469	0.0045	0.5553	0.5001	0.6260	0.4226	1.3762	0.8243	1.4484	1.5640
DNSGA-II-A	**0.0116**	**0.0049**	**0.0097**	**0.0144**	**0.0131**	**0.0317**	**0.0376**	0.0066	**0.0586**	**0.0685**	**0.0503**	**0.1003**	**0.1190**	**0.1636**	**0.2375**	**0.2534**
DNSGA-II-B	**0.0110**	**0.0046**	**0.0050**	**0.0134**	**0.0095**	**0.0255**	**0.0390**	0.0077	**0.0517**	**0.0665**	**0.0561**	**0.1048**	**0.1189**	**0.1605**	**0.2175**	**0.2589**
PPS	**0.0158**	**0.0050**	**0.0049**	**0.0196**	0.4074	0.3504	0.6241	0.0065	3.2866	3.5779	6.1369	2.7321	6.5076	2.7454	4.7021	5.4482
Dy-NSGA-II	**0.0118**	**0.0021**	**0.0044**	**0.0117**	**0.0184**	**0.0536**	**0.0896**	0.0118	**0.1077**	**0.0513**	**0.1069**	**0.1030**	**0.1425**	**0.1585**	**0.2741**	**0.3006**

- Again for the synchronous case ($A = 1$) Dy-NSGA-II is better than NSGA-II and the random restart on FDA3 with significant improvements in %HV over both and a lower GD than NSGA-II. Where GD is not improved over the random restart method this implies very good convergence, but a worse spread of of solutions.

These successes for Dy-NSGA-II are not apparent in the FDA5 problem: worse GD and HV% than both NSGA-II and the random restart. Complexity of the archive management mechanism requires refinement to better scale with an increasing number of objectives as this increased run times greatly.

- PPS achieves similar results to both the static NSGA-II and random restart methods, with only a significant improvement on the GD in FDA5 over NSGA-II.

Slow Relative Asynchrony ($A < 1$):

- When observing how these relative performances change with the value of A there are some useful insights for suitable methods in different cases of asynchronous (non-coincidental) multi-dynamic problem instances. Observing the performance of the controls, the NSGA-II and random restart method (first and second rows in each table respectively), for the slow relative asynchrony cases ($A < 1$) shows similar findings to earlier results. Any introduction of multiple dynamic components results in an increased difficulty of the problem instance, reflected in the worse %HV and GD attainment even when one of the components is changing more slowly. As non-coincidence results in a greater number of total dynamic change events, naturally the performance achievable by the random-restart method will be worse. Comparatively, the NSGA-II algorithm achieves competitive results to the other methods across these instances (the lack of significant improvements by the other methods is evidence of this). This provides additional evidence for the conclusions in Chapter 3 about the unsuitability of the random-restart method as a baseline comparison method when unmodified static algorithms, such as NSGA-II, provide a more useful baseline.
- All of the other methods, the DNSGA-II-A, DNSGA-II-B, PPS and Dy-NSGA-II, all achieve significant improvements over the random restart method across the majority of the instances (bold values) for both FDA3 and FDA5. Only Dy-NSGA-II and PPS in cases with values of A close to 1, worse than the random restart in terms of the %HV measurements. Both the DNSGA-II-A and DNSGA-II-B algorithms consistently achieve significant improvements in both %HV and GD measurements over the random restart method for both FDA3 and FDA5, with DNSGA-II-B additionally outperforming NSGA-II on GD measurements for FDA5.

Fast Relative Asynchrony ($A > 1$):

- These instances add additional difficulty and help to separate the performance of the algorithms further. As the colour gradient is based on the values within each table, the darker cells towards the right of each table as the value of A increases, indicate this overall increase in difficulty; the performance of all algorithms is decreased. This observation is only possible because of the absolute nature of these results - we know what the optimum is and therefore can clearly observe when deviation from this takes place. Comparing the NSGA-II and random restart algorithms (first and second rows in each table respectively), we again see that the decline in the performance of the random restart method is severe compared to NSGA-II, reinforcing the unsuitability of random restart comparisons.

- The PPS method shares a similar marked decline in performance as the value of A increases and is unsuitable for any of these instances for both FDA3 and FDA5 problems. Dy-NSGA-II achieves good performance on these instances for the FDA3 problem, with significant improvements over NSGA-II and the random restart for many of the values of A. However, for the FDA5 problem, whilst significant improvements in both %HV and GD remain over the random restart, the improvements are less consistent over NSGA-II. This implies that further adaptation or balancing of the response is required for problems of higher dimension.
- As in the slow relative cases, the DNSGA-II-A and DSNGA-II-B algorithms consistently improve over the random restart method, achieving significantly better %HV and GD in every instance across both FDA3 and FDA5 problems. There are also several instances where significant improvements over NSGA-II are present, however the D-NSGA-II-B algorithm achieves these more consistently. This highlights the important of exploitation of existing solutions in rapidly changing environments, however, the fixed and relatively low severity of changes in these instances may be a contributing factor to the distinction between these two methods.

The observations from the above results contribute both specifically and generally to the investigation of multi-dynamic problems. Specifically, the examined DMOEA techniques highlight the need for dedicated development to tackle the challenges intrinsic to multi-dynamic instances. This is particularly apparent for prediction strategies, where model construction methods based around a single dynamic component (or co-incidental / synchronous) changes are intuitively not likely to extend performance to multiple, independently changing dynamic components. Adaptive mechanisms such as the Dy-NSGA-II, face similar challenges for multi-dynamic instances; the adaptivity of the response mechanism must be able to handle the impacts of change events in two (or more) components of the problem, which may be different from each other. There are many similarities between the Dy-NSGA-II algorithm and the SeedEA algorithm proposed for the combinatorial DTTP instances defined in Chapter 4. The combination of a variety of solution information to provide ‘structured diversity’ may be the key to the success of these algorithms on complex and difficult dynamic instances, however additional development or tuning may be required to allow for these performance gains on problems with more than two objectives.

More generally, the presented methodology for the parameterisation of the relative asynchrony of multiple dynamic changes provides insights into potential experimental approaches for this type of problem. The definition of the A parameter provides a useful way to collapse the large possible dynamic instance space shown previously, into a more feasible instance set, where the relationship between instances remains intuitive and clearly defined. It is also easy to compare algorithm performance across this subset of instances in a way that is more difficult for the heatmap representations. However, this simplification of the possible multi-dynamic instance space brings with it multiple limitations which reflect and compound the previously identified challenges for dynamic problems. Selection of a subset of instances using the employed range of A parameter values prevents the potential exploitation or avoidance of the identified motifs in the dynamic instance space (component dominance and factor frequency). However alternative ranges of A can allow for only factor-frequency instances to be investigated.

Informed instance selection was a key recommendation from Chapter 3 and the reduction of the possible dynamic instance space should include an observation of the different impacts of frequency and severity on different problems. This means that a single scale of A values, whilst easy for comparison, may not be suitable for comparing algorithm performance across different problems. The selection of an appropriate 'default frequency' is also challenging and limits the general relevance and comparability of results. Similar to the observation of conclusions in the DMO literature based on experimentation with a single combination of frequency and severity, conclusions made with a specific value of this 'default frequency' offer limited and narrow relevance.

Finally, the fixed severity of the instances investigated limit the overall generality of conclusions possible. This remains a contributing factor to the scale of the possible dynamic instance space and similarly to the conclusions of Chapters 3 & 4, explicit conditioning of results should also be a foundation of experiments on multi-dynamic problems.

5.6 Summary of Challenges and Recommendations for Multi-Dynamic Optimization

The preceding experiments across combinatorial and continuous domains, considering co-incidental and non-coincidental changes in multi-dynamic example problems have allowed for the identification of several key challenges and an understanding of the limitations of current methods for tackling this type of complex problems. Each of these challenges is addressed below and a recommendation based on the interpretation of experimental results is provided.

1. The size of the possible dynamic instance space had been a recurring observation across all of the examined multi-dynamic problem examples and is an issue that also applies to single-dynamic combinatorial problems. The introduction of

multiple dynamic components requires the definition of additional parameters that define the frequency and severity of these additional changes. Varying these new parameters generates unique instances based on each combination of values and presents an explosion in the scale of the possible dynamic instance space.

- The informed or deliberate selection of relevant instances is one potential way to counter the size of the possible dynamic instance space. Drawing from realistic cases and the construction of subsets of instances that reflect useful and relevant characteristics may be one way to improve the feasibility of multi-dynamic problem experiments. However, as mentioned previously, this requires conditioning of the results such that an understanding of their exact relevance is provided alongside any conclusions.

2. For combinatorial instances, the reproducibility of results in multi-dynamic instances should be a key focus. As the number of unique ways to apply a change of magnitude $n_t = 2$ is large even for problems with few indexes, and each of these results in a unique problem state, both the pattern of changes and their order must be carefully reported to ensure the reproducibility of results. This is a challenge identified for single-dynamic combinatorial, however with multiple dynamic changes and a larger number of successive, unique problem states in non-coincidental cases, its impact is magnified.

    - Careful reporting of patterns of changes, together with the explicit statement of pattern construction can ensure that reproducibility standards are upheld for these problems. This is closely linked to the correct conditioning of problems and also highlights the limitations of experiments by considering a small sample of the potential dynamic instance space. Similarly to the recommendation to construct experimental test sets based on specific investigation goals, the construction of patterns of changes for combinatorial instances can be specific to relevant changes characteristics observed in real systems.

3. The accurate recording and effective reporting of performance is an ongoing research task within DMO. However, the multi-dynamic instances examined provide an additional challenges to those outstanding for dynamic multi-objective problems. Identified across both continuous and combinatorial instances, the application of existing practices for recording performance measurements before each change event are confused by the introduction of multiple dynamic components. Experimental data for the continuous benchmarks indicate different observations can be made depending on when performance measurements are made.

    - Current recommendations to address this challenge are based on the observations across combinatorial and continuous examples examined. The measurement strategy, i.e. before each change event in each component or before both, should be carefully considered, and more importantly, clearly stated when carrying out experiments on multi-dynamic instances. Experiments on a subset of instance selected for relevance to a real world characteristic may be suited to a particular measurement strategy, however understanding that there are alternative ways to record this performance is also important. Additionally, mixed visualization techniques (such as including profiles of performance measurements as well as tables of averaged performance values) provide additional context and help to illustrate complex temporal behaviours of algorithms, such as differential performance across intervals, that would otherwise be unexplained by single-valued performance reporting.

4. Particularly relevant to the combinatorial instances examined, unknown Pareto Fronts make it difficult to draw experimental conclusions without using comparison (as in Chapter 4). The results presented for the coincidental changes in the DTTP focus on comparison between response techniques, however to the DTTP and the static TTP the unknown Pareto Front imposes a limitation on the conclusions that can be made around performance. Without an absolute limit to the achievable performance, as in the case for the continuous problems for which heatmaps of the deviation from this maximum can be constructed, conclusions rely on comparison alone.

   - Specifically for the TTP and its derivations, additional research toward accurate and efficient approximations of the Pareto Front (such that the successive problem states in dynamic case can also benefit) is required. More generally, in combinatorial and continuous multi-dynamic problems where the Pareto Front is unknown there are two options for determining the performance of algorithms. Either comparison to performance achieved by a baseline algorithm or an understanding of performance goals provided by external context, such as through selection of a reference point in the objective space or preference incorporation techniques that have been used elsewhere in DMO [213], [220].

5. The complexity of impacts of multiple dynamic changes across different problems is an additional challenge. Noted in Chapter 3, the same observations can be made here; the same combinations of frequency parameter values on different problems can result in instances of different difficulty

   - Informed selection is imperative when constructing experimental sets of test instances. The importance of problem selection and ranges of dynamic characteristics among test sets is noted previously [29], [32], [41], [255]. However, an understanding-driven experimental design approach is necessary when the range of potential dynamic instances is large as in the multi-dynamic case. Previous experimental approaches which simply compose a test set of single or a limited range of instances from different problems cannot be directly applied to the multi-dynamic case.

Many of these identified challenges and recommendations have relevance beyond specifically the multi-dynamic instances examined here. For example the informed selection of instances with patterns of relevant changes is particularly pertinent for problems with a changing number of objectives or decision variables. Since there are many ways to execute these changes, i.e. which objectives/variables are added/removed, and many potential patterns of changes, experimental approaches should follow one of two paths. Either the comprehensive consideration of potential patterns and instances should be evaluated, or a subset of instances that reflect specific real world cases should be investigated. Furthermore, the development of effective and meaningful performance measurements remains a challenge for problems with other types of complex dynamics within DMO. Careful consideration should be paid as to the comparability and reproducibility of results across instances and across studies.

Collectively, these complex dynamics represent characteristics of real world scenarios and investigating how algorithms perform on simplified problem instances containing these characteristics provides us with important information. A greater understanding of the difficulties inherent in different industrial applications can be gleaned through the informed selection of problem instances. Establishing expectable baselines in performance also helps to drive development of algorithms and response mechanisms as resources need not be wasted on finding solutions obtainable by existing methods. What remains is for the development of a coherent central repository for these baselines in performance across a range of different continuous and combinatorial problems and dynamic change types.

5.7 Conclusions

A number of experiments are performed to demonstrate and assist in the identification of the key challenges, characteristics and suitability or approaches for multi-dynamic problems. Example multi-objective problems are defined for both the combinatorial and continuous domains and challenges common to both and unique to each are realised. The impacts of frequency and severity on the Dynamic Travelling Thief Problem are visualised and the distinction between multi-dynamic instances with coincidental and non-coincidental changes is explained. Methods defined in Chapter 4 are evaluated in the coincidental multi-dynamic case for the DTTP and results indicate more significant challenges arise from the non-coincidental case.

The definition of a number of multi-dynamic continuous benchmark problems enables the comprehensive experimental procedure defined in Chapter 3 to be applied here. Patterns of performance, termed 'motifs', are visible across the dynamic instance space generated by different combinations of frequency parameters which vary the incidence of change events. Identification of these patterns, the factor frequency motifs and dominant component motifs, highlight features of the dynamic instance space that future experimental approaches may exploit or avoid.

A case study approach to handling the increase in the number of possible instances is proposed through collection of a subset of instances with different relative asynchrony, meaning the non-coincidence of changes in one dynamic component is relative to the fixed occurrence of the other. Whilst the results verify intuitive beliefs about the difficulty of instances with rapid and slower changes, the relevance of the results is limited by the specificity of the fixed occurrence and definition of the relative asynchrony of instances.

The key challenges are consolidated for multi-dynamic problems across both domains and recommendations are made to enable more informed experimental design for multi-dynamic instances. Major challenges are identified in the application of traditional performance measurement practices; the explosion of the possible instance space due to the unique impacts of the frequency and severity parameters in different dynamic components. Challenges identified in previous chapters are magnified in the multi-dynamic case; reproducibility of results for combinatorial patterns of changes, the determination of absolute performance on problems with unknown Pareto Fronts and the differential impacts of, now multiple, frequency and severity parameters across different problems.

General recommendations on the importance of informed selection of test instances are relevant to the multi-dynamic case. The reduction of the possible instance space through the identification of relevant subsets that are representative of realistic characteristics is a future goal that will enable directed and meaningful experimentation on multi-dynamic problems.

This exploratory research into the novel and as yet unconsidered class of multi-dynamic multi-objective optimization problems provides significant contribution towards refining experimental approaches for these problems. As more realistic test problems become the focus of optimization approaches, prefatory experimentation such as these help to guide the establishment of efficient practices and cohesive research in complex areas that are relevant to real world scenarios.

Chapter 6

Conclusions and Future Directions

A variety of work has been presented which addresses some of the key gaps within dynamic multi-objective optimization research. The contributions span continuous and combinatorial domains in attempt to provide a more complete picture of reproducibility, realism and extracting meaning from results on previous, current and future problems within the field. A reminder of the research questions that this thesis addresses, as set out in Section 1.2, are included below.

> **RQ1:** *How can we characterise the impacts of dynamic parameters in DMOPs for more meaningful benchmark evaluation?*

> **RQ2:** *How can we incorporate dynamics into existing static problems to generate and find solutions for realistic combinatorial test instances that better reflect characteristics observable in real-world systems?*

> **RQ3:** *How can we apply similar methodologies for both continuous and combinatorial domain problems to define the key features, challenges and potential approaches to multi-dynamic multi-objective optimisation problems?*

These questions drive the understanding of distinct aspects of dynamic multi-objective optimisation that have been so far overlooked. Collectively, they span the challenging of established practice, the expansion of currently available test problems and a forward-looking approach toward novel types of problems of interest in the field and industry. The combination of these questions and the resulting interlinked nature of the methods, optimization techniques and conclusions seen throughout the presented results, highlight the wider applicability of outcomes of specific research across DMO.

6.1 Summary of Contributions

Each the preceding chapters has addressed the research questions set out in Chapter 1, providing results and conclusions that attempt to answer these key topics. Detailed conclusions are given in each of Chapter 3, 4 & 5 and so a summary of the major contributions is provided here.

1. To effectively address **RQ1** as stated in Chapter 1, a comprehensive experimental approach to the possible instance space of continuous dynamic multi-objective optimization problems (DMOPs) is presented using the frequency and severity of dynamic changes to define the space. Specific recommendations are given for the examined benchmark sample, the baseline comparison algorithms in terms of challenging problems and instances that should be the target of future works. General recommendations are made for improving experimental practices in DMO, in terms of reproducibility, comparability and relevance of results. These recommendations are based on the proposed practice for establishing baseline performance using static algorithms and basic dynamic response mechanisms. To facilitate the adoption of these practices, an easy to use platform (the DPTP: Dynamic Parameter Testing Platform) has been made available and through which the presented results can be easily reproduced.
2. Three types of novel dynamic instances of a complex combinatorial bi-objective problem are proposed to provide answers to **RQ2**, adding to the scope of available realistic DMOPs in this domain. Observations from previous exploitation of problem knowledge to improve the solution quality and algorithm efficacy, enabled the development of a reactive seeding algorithm. The determination of effective combinations of seeding information for the different dynamic instance types across a range of problem sizes is presented. Comparison with static heuristics that similarly exploit problem information is provided. Further development of the seeding algorithm improves the coverage of the non-dominated set through isolated evolution of multiple, specialised populations of solutions and the evaluation of simple migration policies to boost convergence.
3. **RQ3** is addressed through several threads of investigation. Consideration of the impacts of multiple coincidental changes in the DTTP and the evaluation of solution information reactively provided by the previously proposed seeding algorithm in these cases. The impacts of frequency and severity on the mono-dynamic DTTP and challenges raised by disparate performance across different patterns of changes and an unknown POF. Investigation into example multi-dynamic continuous problems with differential frequency and severity and the insights gained: factor frequency, dynamic component dominance and changes to the difficulty of regions of the dynamic instance space defined in Chapter 3. Exploratory experi-

ments applying DMOEA methods to a subset of multi-dynamic instances defined by their relative difference in frequency of changes are presented, together with a critical evaluation of such an approach. Recommendations across both problem domains are made for the future design of experiments focusing on this novel class of dynamic instances.

These contributions have wider applicability to DMO; generally using dynamics parameters to define the possible instance space rather than a minimal selection of instances without consideration of how representative they are, highlights the more informed and comprehensive experimentation required for meaningful DMO research. The attention paid to the reproducibility of these results is something that must become common practice in DMO for cohesive and verifiable progress to be made. Towards more realistic benchmark problems and test instances; using an iterative development process of algorithms for these complex problems, together with justifications and meaningful evaluation and comparison showcases the motivations and strengthens possible conclusions. Approaching novel problem classes or formulating original systems as optimization problems can be challenging, however contributions need not always be solution focused: identifying the key challenges and the recommendation of experimental practices provides useful information. In contrast, the limitations of the experiments and their results provide perspective and highlight areas for further work.

6.2 Evaluation and Future Work

For each of the chapters, the key limitations that have been identified are presented in turn here, together with the opportunities for further investigation arising from them.

6.2.1 Establishing Baseline Performance and Experimental Protocols for DMOP Benchmarks:

The results are presented for a number of well-known MOEA algorithms, for a sample of commonly used bi-objective DMO benchmarks across 630 combinations of frequency and severity. Due to the volume of results required to demonstrate the impacts of frequency and severity across the instance space on each problem, a primary limitation, and subsequent opportunity, is in the application of this framework to the complete set of DMOP benchmark suites present in the literature (as documented in Chapter 2).

Extension of the protocols provided here will allow a comprehensive library of expectable performance to be compiled that is directly relevant and comparable to future works. Additionally, as noted in Chapter 3, the development of a mechanism for the automated assay (using surrogate models or SVM) of the dynamic instance space, will be useful to reduce the computational burden of evaluating performance across the entire window of the dynamic instance space as considered here. A similar framework for combinatorial benchmarks is difficult due to the nature of and reproducibility of exacted dynamic changes and the often unknown Pareto Fronts making absolute performance measurement difficult. However the establishment of accessible and explainable protocols to generate reproducible results for combinatorial DMOPs is a remaining task.

6.2.2 The Dynamic Travelling Thief Problem and its Solutions:

Several limitations, mostly in terms of comparability and reproducbility, are identified in terms of the proposed dynamic Travelling Thief Problem instances. The largest instances included in these experiments (more than 2000 cities with 10 items-per-city) are costly in terms of computational resources. Furthermore, a general limitation for combinatorial problems that remains unsolved so far; the nature of dynamic changes in most combinatorial problems is such that the patterns of possible changes or the combination or sequence of the changes generates a vast instance space. This challenge is not as significant in continuous problems, however randomized changes provide a similar challenge. To clarify the limitation further, unless sufficiently many patterns are considered, providing a general picture of the possible dynamic instance space, the results must be conditioned on the specific patterns employed and their reproducibility must be ensured. Selection of a subset of patterns that effectively represents the possible dynamic instance space is an opportunity for further research in this area. Comparisons with established DMOEA methods are omitted in favour for a comparison to adapted TTP-specific heuristics. The exploitation of problem knowledge (using solved problem components) employed in these provides significant advantages which many algorithms that employ randomized initialization may struggle to achieve competitive results in similar time/number of evaluations. Therefore future development of a standardised, algorithm-independent initialization method may be used to compare existing and established DMOEA mechanisms.

Transfer learning mechanisms and further generalised models for partial solutions may help to address the computational burden of larger problems by reducing the number of evaluations required to reach good solutions. Specific exploitation of the TTP problems structure, such as the impact of the *dropRate* parameter may yield additional insights into response mechanisms for the dynamic instances. Finally, a more general goal for dynamic combinatorial problems is to streamline reporting for a generalized or justified set of dynamic patterns to improve comparability of results.

6.2.3 Combinatorial and Continuous Multi-Dynamic Problems:

Consideration of a novel class of problems comes with discovering the challenges they present. Through the prior scope of the possible dynamic instance space established for continuous problems early in this thesis, the consideration of each additional dynamic component results in additional dimensions to the instance space. Therefore, insights are presented in a series of preliminary and exemplary cases rather than consideration of the entire dynamic instance space as this would be infeasible. Furthermore, the challenges identified for combinatorial problems remain relevant and potent in the multi-dynamic case, with the observation-based recommendations advising on general practice.

A significant task facing multi-dynamic, and more generally dynamic combinatorial problems is the effective representation of the possible dynamic instance space within the sampled instances used in experiments. As the number of unique possible change events is exponential in the number of changeable indices, the efficient representation of the possible sequences of changes, the overall patterns and the sets of patterns used in test sets is a challenging future task. Data-driven or bespoke and informed construction of sets of instances that reflect useful combinations of problem characteristics is one potential solution to this curse of dimensionality.

6.2.4 Example Future Research Topics within DMO:

There are opportunities to develop the investigation presented here in a variety of ways to provide significant contributions to the field of DMO. A number of examples of these are presented below:

- The application of the presented comprehensive methodology for performance over a range of dynamic instances to consider different algorithm operators, or additional parameters such as population size or the number of decision variables.
- The expanded investigation to explain definitively how different algorithm mechanisms (e.g. dominance-based versus decomposition-based) enable different patterns of performance across the dynamic instance space of different problems.
- The definition of additional types of dynamic changes for the TTP or for other combinatorial benchmarks such as multi-vehicle routing problems, knapsack problems or bin packing problems.
- The development of coherent standardised practices for performance measurement for dynamic, multi-objective combinatorial problems.
- The extension of existing instance space analysis techniques for dynamic problems, specifically in the reconciling of measurements over different dynamic intervals into a useful and informative feature measurement.
- The definition of investigative practices and relevant instance selection for the next generation of complex dynamic problems that more closely reflect real world scenarios.

Whilst there are many specific developments for each of the presented research questions, there are also several more general potential future areas of investigation. For example, the collection and cataloguing of performance baselines for continuous problems into a resource that enables rapid development of new methods or to enable a querying service for the informed selection of instance to evaluate. This notion extends beyond the continuous benchmarks examined here; a cohesive method to rapidly compare performance results is a limiting factor on the rate of research in DMO, and more widely across optimization.

6.3 Closing Remarks

Multiple topics across DMO are presented here spanning the establishing of experimental protocols and determining the impacts of dynamics parameters; the justified construction of novel dynamic, combinatorial benchmarks and pioneering investigations into novel features of realistic problem formulation. At the confluence of these topics is the establishing of reproducible, explainable and accessible methodologies, benchmarks and exploratory experimentation approaches for DMO. A framework for the reproducible experimentation on continuous DMOP benchmarks is proposed using the frequency and severity of change parameters to define the possible dynamic instance space. A baseline of expectable performance using common MOEAs with no, or simple dynamic response mechanisms is established and provides insights on the impacts of the the frequency and severity parameters on the relative difficulty of different benchmarks. Specific recommendations are made based on the obtained results for the DMOP problems examined and general recommendations are provided to guide more informed and meaningful practice towards evaluation of DMO benchmarks.

Three types of dynamic instance for the realistic, bi-objective, interconnected Travelling Thief Problem are proposed together with justification and complete descriptions of implementation and execution. Examples of instances with varying size, encompassing and expanding on previous scope, are evaluated in this work.

The previous usage of problem information, partial solutions and solvers to influence algorithm performance and improve the quality of obtained solutions is extended here. Using an observation-driven and explanation focused, iterative development process for algorithms that reactively insert solution information, the mitigation of dynamic change impacts is evaluated.

An open-ended investigation of the potential for multi-dynamic problems was undertaken. A key focus of this was to identify the major challenges that present when navigating the possible instance space, for finding solutions and determining problem and dynamics features that may be exploited by algorithms. Examples of continuous and combinatorial problem instances that incorporate this novel aspect of realistic systems are used, since the applicability of many real world features is far-reaching across DMO.

Through the presented work in this thesis, the topics identified by the research questions have been comprehensively addressed. Each of these investigations, whilst providing insights and answers, prompted many further questions, avenues for investigation and ideas for additional experiments. Finding an answer to a question, but in the process identifying multiple new questions highlights a cornerstone of scientific research; our work is never truly over.

Bibliography

[1] G. H. Dash and N. Kajiji, “On multiobjective combinatorial optimization and dynamic interim hedging of efficient portfolios,” *International Transactions in Operational Research*, vol. 21, no. 6, pp. 899–918, 2014, ISSN: 14753995. DOI: 10.1111/itor.12067.

[2] N. Honing and H. La Poutre, “Reducing electricity consumption peaks with parametrised dynamic pricing strategies given maximal unit prices,” *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, pp. 171–175, 2013, ISSN: 15294188. DOI: 10.1109/DEXA.2013.46.

[3] Y. Jin and J. Branke, “Evolutionary optimization in uncertain environments-a survey,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005. DOI: 10.1109/TEVC.2005.846356.

[4] T. Friedrich and M. Wagner, “Seeding the initial population of multi-objective evolutionary algorithms: A computational study,” *Applied Soft Computing Journal*, vol. 33, pp. 223–230, 2015, ISSN: 15684946. DOI: 10.1016/j.asoc.2015.04.043. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2015.04.043.

[5] H. Faulkner, S. Polyakovskiy, T. Schultz, and M. Wagner, “Approximate Approaches to the Traveling Thief Problem,” pp. 385–392, 2015. DOI: 10.1145/2739480.2754716.

[6] J. Wu, M. Wagner, S. Polyakovskiy, and F. Neumann, “Exact approaches for the travelling thief problem,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10593 LNCS, 2017, pp. 110–121, ISBN: 9783319687582. DOI: 10.1007/978-3-319-68759-9_10. arXiv: arXiv:1708.00331v1.

[7] J. H. Holland, “The Optimal Allocation of Trials,” *SIAM Journal of Computing*, vol. 2, no. 2, pp. 88–105, 1973. DOI: 10.7551/mitpress/1090.003.0008.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Bradford book). MIT Press, 1992, ISBN: 9780585038445. [Online]. Available: https://books.google.co.uk/books?id=cyV7nQEACAAJ.

[9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Pub: Addison Wesley, 1989.

[10] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning,” *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988, ISSN: 15730565. DOI: 10.1023/A:1022602019183.

[11] D. Goldberg and R. Smith, “Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy.,” *Icga*, pp. 59–69, 1987. [Online]. Available: http://illigal.org/wp-content/uploads/illigal/pub/papers/Publications/1987/nonstationary_function_opt-ICGA_1987.pdf.

[12] J. Klockgether and H. P. Schwefel, “Two-phase nozzle and hollow core jet experiments,” *Proc. 11th Symp. Engineering Aspects of Magnetohydrodynamics*, no. DECEMBER 1969, pp. 141–148, 1970.

[13] J. J. Grefenstette, “Optimization of Control Parameters for Genetic Algorithms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986. DOI: 10.1109/TSMC.1986.289288.

[14] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997. DOI: 10.1023/A:1008202821328. [Online]. Available: http://dx.doi.org/10.1023/A:1008202821328.

[15] N. Cramer, “A Representation for the Adaptive Generation of Simple Sequential Programs,” *Proc. of an Intl. Conf. on Genetic Algorithms and their Applications, Carnegie-Mellon University*, 1985.

[16] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[17] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006, ISSN: 1556-603X. DOI: 10.1109/MCI.2006.329691. [Online]. Available: http://ieeexplore.ieee.org/ielx5/10207/4129833/04129846.pdf?tp=&arnumber=4129846&isnumber=4129833%5Cnhttps://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4129846.

[18] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” *Proceedings of ICNN’95 - International Conference on Neural Networks*, pp. 1942–1948, 1995. DOI: 10.1007/978-3-319-46173-1_2.

[19] A. P. Wierzbicki, “Dynamic aspects of multi-objective optimization,” *Lecture Notes in Economics and Mathematical Systems, Eds. A. Lewandowski, V. Volkovich*, vol. 351, pp. 154–174, 1988.

[20] H. G. Cobb and J. J. Grefenstette, “Genetic algorithms for tracking changing environments,” *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 523–530, 1993. DOI: 10.1.1.48.6501.

[21] J. J. Grefenstette, “Genetic algorithms for changing environments,” *Ppsn*, vol. 2, pp. 137–144, 1992. DOI: 10.1.1.48.6501. [Online]. Available: http://www.gardeux-vincent.eu/These/Papiers/Bibli1/Grefenstette92.pdf.

[22] A. Ghosh, S. Tsutsui, and H. Tanaka, “Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals,” *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, pp. 666–671, 1998. DOI: 10.1109/icec.1998.700119.

[23] J. Branke, “Memory enhanced evolutionary algorithms for changing optimization problems,” *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, vol. 3, no. 721, pp. 1875–1882, 1999, ISSN: 02683768. DOI: 10.1109/CEC.1999.785502.

[24] W. Liles and K. D. Jong, “The usefulness of tag bits in changing environments,” in *Proceedings of the Congress on Evolutionary Computation*, P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, Eds., vol. 3, Mayflower Hotel, Washington D.C., USA: IEEE Press, 1999, pp. 2054–2060, ISBN: 0-7803-5537-7 (Microfiche).

[25] S. K. Oh, C. Y. Lee, and J. J. Lee, “A new distributed evolutionary algorithm for optimization in nonstationary environments,” *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 1, pp. 378–383, 2002. DOI: 10.1109/CEC.2002.1006264.

[26] D. Yazdani, M. N. Omidvar, R. Cheng, J. Branke, T. T. Nguyen, and X. Yao, “Benchmarking Continuous Dynamic Optimization: Survey and Generalized Test Suite,” *IEEE Transactions on Cybernetics*, pp. 1–14, 2020, ISSN: 2168-2267. DOI: 10.1109/tcyb.2020.3011828.

[27] S. Yang, “Non-stationary problem optimization using the primal-dual genetic algorithm,” *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*, vol. 3, pp. 2246–2253, 2003. DOI: 10.1109/CEC.2003.1299951.

[28] M. Farina, K. Deb, and P. Amato, “Dynamic Multiobjective Optimization Problems: Test Cases Approximation and Applications,” *Evolutionary Multi-Criterion Optimization. Second International Conference EMO 2003*, vol. 8, no. 5, pp. 311–326, 2003, ISSN: 03029743. DOI: 10.1109/TEVC.2004.831456.

[29] S. Jiang and S. Yang, “Evolutionary Dynamic Multiobjective Optimization: Benchmarks and Algorithm Comparisons,” *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, 2017, ISSN: 21682267. DOI: 10.1109/TCYB.2015.2510698.

[30] J. Ou, L. Xing, M. Liu, and L. Yang, “A Novel Prediction Strategy Based on Change Degree of Decision Variables for Dynamic Multi-Objective Optimization,” *IEEE Access*, vol. 8, pp. 13 362–13 374, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2961980.

[31] R. Chen, K. Li, and X. Yao, “Dynamic Multiobjectives Optimization with a Changing Number of Objectives,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2018, ISSN: 1089778X. DOI: 10.1109/TEVC.2017.2669638. arXiv: 1608.06514.

[32] S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor, “A Scalable Test Suite for Continuous Dynamic Multiobjective Optimization,” *IEEE Transactions on Cybernetics*, vol. 6, no. 50, pp. 2814–2826, June 2020, ISSN: 2168-2267. DOI: 10.1109/tcyb.2019.2896021. arXiv: arXiv:1903.02510v1.

[33] L. Huang, I. H. Suh, and A. Abraham, “Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants,” *Information Sciences*, vol. 181, no. 11, pp. 2370–2391, 2011. DOI: 10.1016/j.ins.2010.12.015. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2010.12.015.

[34] S. Jiang, S. Yang, X. Yao, K. C. Tan, and M. Kaiser, “Benchmark Problems for CEC2018 Competition on Dynamic Multiobjective Optimisation,” *CEC2018 Competition*, pp. 1–18, 2018.

[35] R. Azzouz, S. Bechikh, and L. B. Said, “Multi-objective Optimization with Dynamic Constraints and Objectives : New Challenges for Evolutionary Algorithms,” *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 615–622, 2015. DOI: 10.1145/2739480.2754708. [Online]. Available: http://dl.acm.org/citation.cfm?id=2754708&dl=ACM&coll=DL&CFID=596636671&CFTOKEN=56506171.

[36] K. Trojanowski and Z. Michalewicz, “Evolutionary Algorithms for Non-Stationary Environments,” *Proceedings of the Workshop Intelligent Information Systems VIII, Poland*, pp. 1–12, 1999.

[37] S. Jiang, J. Zou, S. Yang, S. Member, and X. Yao, “Evolutionary Dynamic Multi-Objective Optimisation : A Survey,” *ACM Computing Surveys*, pp. 1–19, 2021.

[38] T. Nguyen, S. Yang, and J. Branke, “Evolutionary dynamic optimization: A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 6, 2012, ISSN: 22106502. DOI: 10.1016/j.swevo.2012.05.001.

[39] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, and X. Yao, “A Survey of Evolutionary Continuous Dynamic Optimization over Two Decades-Part A,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 630–650, 2021, ISSN: 19410026. DOI: 10.1109/TEVC.2021.3060012.

[40] C. Wang, G. G. Yen, and F. Zou, “A novel predictive method based on key points for dynamic multi-objective optimization,” *Expert Systems with Applications*, vol. 190, no. October 2021, p. 116 127, 2022, ISSN: 09574174. DOI: 10.1016/j.eswa.2021.116127. [Online]. Available: https://doi.org/10.1016/j.eswa.2021.116127.

[41] M. Helbig and A. P. Engelbrecht, “Benchmarks for dynamic multi-objective optimisation,” *Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments, CIDUE 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, vol. 46, no. 3, pp. 84–91, 2013, ISSN: 03600300. DOI: 10.1109/CIDUE.2013.6595776.

[42] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Norwell, MA, USA: Kluwer Academic Publishers, 2001, ISBN: 0792376315.

[43] R. Azzouz, S. Bechikh, and L. B. Said, "Dynamic multi-objective optimization using evolutionary algorithms: a survey," pp. 31–70, 2016.

[44] Chi-Keong Goh, "Evolutionary Multi-Objective Optimization in Uncertain Environments," *PhD dissertation, National University of Singapore*, 2007.

[45] K. Trojanowski and Z. Michalewicz, "Searching for optima in non-stationary environments," *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, vol. 3, pp. 1843–1850, 1999. DOI: 10.1109/CEC.1999.785498.

[46] A. Tantar, E. Tantar, and P. Bouvry, "Design and classification of dynamic multi-objective optimization problems," *arXiv preprint arXiv:1103.4820*, pp. 177–183, 2011. DOI: 10.1145/2001858.2001918. [Online]. Available: http://arxiv.org/abs/1103.4820.

[47] E. Tantar, A. A. Tantar, and P. Bouvry, "On dynamic multi-objective optimization, classification and performance measures," *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 2759–2766, 2011. DOI: 10.1109/CEC.2011.5949964.

[48] A.-A. Tantar, E. Tantar, and P. Bouvry, "A classification of dynamic multi-objective optimization problems," *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation - GECCO '11*, p. 105, 2011. DOI: 10.1145/2001858.2001918. [Online]. Available: http://portal.acm.org/citation.cfm?doid=2001858.2001918.

[49] H. Aguirre and K. Tanaka, "Insights on properties of multiobjective MNK-landscapes," *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, pp. 196–203, 2004. DOI: 10.1109/CEC.2004.1330857.

[50] Y. Jin and B. Sendhoff, "Constructing dynamic optimization test problems using the multi-objective optimization concept," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3005, pp. 525–536, 2004, ISSN: 03029743. DOI: 10.1007/978-3-540-24653-4_53.

[51] I. O. Essiet, Y. Sun, and Z. Wang, "A novel algorithm for optimizing the Pareto set in dynamic problem spaces," *2018 Conference on Information Communications Technology and Society, ICTAS 2018 - Proceedings*, pp. 1–6, 2018. DOI: 10.1109/ICTAS.2018.8368762.

[52] A. Birashk, J. Kazemi Kordestani, and M. R. Meybodi, "Cellular teaching-learning-based optimization approach for dynamic multi-objective problems," *Knowledge-Based Systems*, vol. 141, pp. 148–177, 2018, ISSN: 09507051. DOI: 10.1016/j.knosys.2017.11.016. [Online]. Available: https://doi.org/10.1016/j.knosys.2017.11.016.

[53] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Computing*, vol. 21, no. 4, pp. 885–906, 2017. DOI: 10.1007/s00500-015-1820-4.

[54] Y. Wang and B. Li, "Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment," *2009 IEEE Congress on Evolutionary Computation*, pp. 630–637, 2009. DOI: 10.1109/CEC.2009.4983004. [Online]. Available: http://ieeexplore.ieee.org/document/4983004/.

[55] Y. Wang and B. Li, "Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization," *Memetic Computing*, vol. 2, no. 1, pp. 3–24, 2010, ISSN: 18659284. DOI: 10.1007/s12293-009-0012-0.

[56] J. Eaton, S. Yang, and M. Gongora, "Ant Colony Optimization for Simulated Dynamic Multi-Objective Railway Junction Rescheduling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2980–2992, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2017.2665042.

[57] Y. N. Guo, J. Cheng, S. Luo, D. Gong, and Y. Xue, "Robust Dynamic Multi-Objective Vehicle Routing Optimization Method," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1891–1903, 2017, ISSN: 15579964. DOI: 10.1109/TCBB.2017.2685320.

[58] R. P. Hamalainen and M. Juha, “Dynamic multi-objective heating optimization,” *European Journal of Operational Research*, vol. 142, pp. 1–15, 2002.

[59] M. M. Hasan, K. Lwin, M. Imani, A. Shabut, L. F. Bittencourt, and M. A. Hossain, “Dynamic multi-objective optimisation using deep reinforcement learning: benchmark, algorithm and an application to identify vulnerable zones based on water quality,” *Engineering Applications of Artificial Intelligence*, vol. 86, no. 2013, pp. 107–135, 2019, ISSN: 09521976. DOI: 10.1016/j.engappai.2019.08.014.

[60] W. Zheng, C. Wang, and D. Liu, “Data-driven based multi-objective combustion optimization covering static and dynamic states,” *Expert Systems With Applications*, vol. 210, no. August, p. 118 531, 2022. DOI: 10.1016/j.eswa.2022.118531. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.118531.

[61] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. Coello, “A heredity-based adaptive variation operator for reinitialization in dynamic multi-objective problems,” *Applied Soft Computing*, vol. 101, p. 107 027, 2021, ISSN: 15684946. DOI: 10.1016/j.asoc.2020.107027. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.107027.

[62] A. Aboud, N. Rokbani, R. Fdhila, *et al.*, “DPb-MOPSO: A Novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization Algorithm,” Dec. 2021. DOI: 10.36227/techrxiv.17207576.v1. [Online]. Available: https://www.techrxiv.org/articles/preprint/DPb-MOPSO_A_Novel_Dynamic_Pareto_bi-level_Multi-Objective_Particle_Swarm_Optimization_Algorithm/17207576.

[63] S. Jiang and S. Yang, “A framework of scalable dynamic test problems for dynamic multi-objective optimization,” *IEEE SSCI 2014: 2014 IEEE Symposium Series on Computational Intelligence - CIDUE 2014: 2014 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments, Proceedings*, no. Cci, pp. 32–39, 2014. DOI: 10.1109/CIDUE.2014.7007864.

[64] J. Li, R. Liu, and R. Wang, “A change type-based self-adaptive response strategy for dynamic multi-objective optimization,” *Knowledge-Based Systems*, vol. 243, p. 108 447, 2022, ISSN: 09507051. DOI: 10.1016/j.knosys.2022.108447. [Online]. Available: https://doi.org/10.1016/j.knosys.2022.108447.

[65] R. Morrison, *Designing Evolutionary Algorithms for Dynamic Environments* (Natural Computing Series). Springer Berlin Heidelberg, 2013, ISBN: 9783662065600. [Online]. Available: https://books.google.com.au/books?id=3ty9BwAAQBAJ.

[66] S. Sahmoud and H. R. Topcuoglu, “Sensor-based change detection schemes for dynamic multi-objective optimization problems,” *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, no. 1, 2017. DOI: 10.1109/SSCI.2016.7849963.

[67] L. Altin and H. R. Topcuoglu, “Impact of sensor-based change detection schemes on the performance of evolutionary dynamic optimization techniques,” *Soft Computing*, vol. 22, no. 14, pp. 4741–4762, 2018, ISSN: 14337479. DOI: 10.1007/s00500-017-2660-1.

[68] H. G. Cobb, “An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments,” *Naval Research Lab Washington Dc*, vol. ADA229159, pp. 1–19, 1990. DOI: 10.1.1.79.4834. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA229159%255Cnhttps://www.google.co.jp/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjErf-W78rRAhXDj5QKHdIPDVMQFgg4MAI&url=http%25253A%25252F%25252Fciteseerx.ist.psu.edu%2525.

[69] M. Yang, L. Kang, and J. Guan, “Multi-algorithm co-evolution strategy for dynamic multi-objective TSP,” in *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, 2008, pp. 466–471, ISBN: 9781424418237. DOI: 10.1109/CEC.2008.4630839.

[70] A. Boulesnane and S. Meshoul, “Do We Need Change Detection for Dynamic Optimization Problems?: A Survey,” *Lecture Notes in Networks and Systems*, vol. 413 LNNS, pp. 132–142, 2022, ISSN: 23673389. DOI: 10.1007/978-3-030-96311-8_13.

[71] M. Chen, Y. Guo, Y. Jin, S. Yang, D. Gong, and Z. Yu, “An environment-driven hybrid evolutionary algorithm for dynamic multi-objective optimization problems,” *Complex & Intelligent Systems*, 2022, ISSN: 2198-6053. DOI: 10.1007/s40747-022-00824-4. [Online]. Available: https://doi.org/10.1007/s40747-022-00824-4.

[72] G. Ruan, J. Zheng, J. Zou, Z. Ma, and S. Yang, “A random benchmark suite and a new reaction strategy in dynamic multiobjective optimization,” *Swarm and Evolutionary Computation*, vol. 63, no. December 2020, p. 100 867, 2021, ISSN: 22106502. DOI: 10.1016/j.swevo.2021.100867. [Online]. Available: https://doi.org/10.1016/j.swevo.2021.100867.

[73] M. Cámara, J. Ortega, and F. De Toro, “Performance measures for dynamic multi-objective optimization,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5517 LNCS, no. PART 1, pp. 760–767, 2009, ISSN: 03029743. DOI: 10.1007/978-3-642-02478-8_95.

[74] J. Mehnen, T. Wagner, and G. Rudolph, “Evolutionary Optimization of Dynamic Multiobjective Functions,” *Technical Report CI-204/06 Dortmund University*, 2006.

[75] M. Greeff and A. P. Engelbrecht, “Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation,” *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pp. 2917–2924, 2008. DOI: 10.1109/CEC.2008.4631190.

[76] C. K. Goh and K. C. Tan, “A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009, ISSN: 1089778X. DOI: 10.1109/TEVC.2008.920671.

[77] A. Zhou, Y. Jin, and Q. Zhang, “A Population prediction strategy for evolutionary dynamic multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014, ISSN: 21682267. DOI: 10.1109/TCYB.2013.2245892.

[78] W. T. Koo, C. K. Goh, and K. C. Tan, “A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment,” *Memetic Computing*, vol. 2, no. 2, pp. 87–110, 2010, ISSN: 18659284. DOI: 10.1007/s12293-009-0026-7.

[79] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, “Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization,” *Proceedings of EMO,LNCS4403*, pp. 832–846, 2007, ISSN: 03029743. DOI: 10.1007/978-3-540-70928-2_62.

[80] S. Biswas, S. Das, P. N. Suganthan, and C. A. Coello Coello, “Evolutionary Multi-Objective Optimization in dynamic environments:a set of benchmarks,” *Proc.2014 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 74–88, 2014.

[81] S. B. Gee, K. C. Tan, and H. A. Abbass, “A Benchmark Test Suite for Dynamic Evolutionary Multiobjective Optimization,” *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 461–472, 2017, ISSN: 21682267. DOI: 10.1109/TCYB.2016.2519450.

[82] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, “Multidirectional Prediction Approach for Dynamic Multiobjective Optimization Problems,” *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–13, 2018, ISSN: 21682267. DOI: 10.1109/TCYB.2018.2842158.

[83] J. Li, R. Liu, and R. Wang, “Handling dynamic multiobjective optimization problems with variable environmental change via classification prediction and dynamic mutation,” *Information Sciences*, vol. 608, pp. 970–995, 2022, ISSN: 0020-0255. DOI: 10.1016/j.ins.2022.06.095. [Online]. Available: https://doi.org/10.1016/j.ins.2022.06.095.

[84] Y. Wu, Y. Jin, and X. Liu, “A directed search strategy for evolutionary dynamic multiobjective optimization,” *Soft Computing*, vol. 19, no. 11, pp. 3221–3235, 2015, ISSN: 14337479. DOI: 10.1007/s00500-014-1477-4.

[85] S. U. Guan, Q. Chen, and W. Mo, "Evolving dynamic multi-objective optimization problems with objective replacement," *Artificial Intelligence Review*, vol. 23, no. 3, pp. 267–293, 2005, ISSN: 02692821. DOI: 10.1007/s10462-004-5900-6.

[86] S. Jiang, M. Kaiser, J. Guo, S. Yang, and N. Krasnogor, "Less detectable environmental changes in dynamic multiobjective optimisation," *Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO '18*, no. April, pp. 673–680, 2018. DOI: 10.1145/3205455.3205521. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3205455.3205521.

[87] X. Li, J. Branke, and M. Kirley, "On performance metrics and particle swarm methods for dynamic multiobjective optimization problems," *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 576–583, 2007. DOI: 10.1109/CEC.2007.4424522.

[88] C. A. Liu, Y. Wang, and A. Ren, "New Dynamic Multi-Objective Constrained Optimization Evolutionary Algorithm," *Asia-Pacific Journal of Operational Research*, vol. 32, no. 5, pp. 1–23, 2015, ISSN: 02175959. DOI: 10.1142/S0217595915500360.

[89] M. Tang, Z. Huang, and G. Chen, "The Construction of Dynamic Multi-objective Optimization Test Functions," *Advances in Computation and Intelligence. Second International Symposium (ISICA'2007)*, pp. 72–79, 2007, ISSN: 03029743.

[90] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," *Evolutionary Multi-Criterion Optimization*, pp. 280–295, 2005, ISSN: 03029743. DOI: 10.1007/978-3-540-31880-4_20.

[91] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results.," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000, ISSN: 10636560. DOI: 10.1162/106365600568202.

[92] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002. DOI: 10.1109/4235.996017.

[93] N. Hansen, A. Auger, O. Mersmann, T. Tusar, and D. Brockhoff, "COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting," pp. 1–10, 2016. arXiv: 1603.08785. [Online]. Available: http://arxiv.org/abs/1603.08785.

[94] R. Liu, N. Li, L. Peng, and K. Wu, "A special point-based transfer component analysis for dynamic multi-objective optimization," *Complex and Intelligent Systems*, 2022, ISSN: 21986053. DOI: 10.1007/s40747-021-00631-3. [Online]. Available: https://doi.org/10.1007/s40747-021-00631-3.

[95] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer Learning-Based Dynamic Multiobjective Optimization Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2018, ISSN: 1089778X. DOI: 10.1109/TEVC.2017.2771451. arXiv: 1612.06093.

[96] M. Yang, L. Kang, and J. Guan, "An Evolutionary Algorithm for Dynamic Multi-Objective TSP," *Lecture Notes in Computer Science*, pp. 62–71, 2007.

[97] J. A. Cordero, A. J. Nebro, C. Barba-González, *et al.*, "Dynamic multi-objective optimization with jMetal and Spark: A case study," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10122 LNCS, pp. 106–117, 2016, ISSN: 16113349. DOI: 10.1007/978-3-319-51469-7_9.

[98] G. Reinelt, "TSPLIB. A traveling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991, ISSN: 08991499. DOI: 10.1287/ijoc.3.4.376.

[99] M. Grötschel, M. Jünger, and G. Reinelt, "Optimal control of plotting and drilling machines: A case study," *Mathematical Methods of Operational Research*, vol. 35, pp. 61–84, May 1991. DOI: 10.1007/BF01415960.

[100] D. Izzo, C. I. Sprague, and D. V. Tailor, “Machine Learning and Evolutionary Techniques in Interplanetary Trajectory Design,” *Springer Optimization and Its Applications*, vol. 144, pp. 191–210, 2019, ISSN: 19316836. DOI: 10.1007/978-3-030-10501-3_8. arXiv: 1802.00180.

[101] Y. Liu, J. Yang, Y. Wang, Q. Pan, and J. Yuan, “Multi-objective optimal preliminary planning of multi-debris active removal mission in LEO,” *Science China Information Sciences*, vol. 60, no. 7, pp. 1–10, 2017, ISSN: 18691919. DOI: 10.1007/s11432-016-0566-7.

[102] D. Hennes, D. Izzo, and D. Landau, “Fast approximators for optimal low-thrust hops between main belt asteroids,” *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 2017. DOI: 10.1109/SSCI.2016.7850107.

[103] A. Panizo-Lledot, M. Pedemonte, G. Bello-Orgaz, and D. Camacho, “Addressing Evolutionary-Based Dynamic Problems: A New Methodology for Evaluating Immigrants Strategies in MOGAs,” *IEEE Access*, vol. 10, pp. 27 611–27 629, 2022, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3156944.

[104] H. Psaraftis, “Dynamic Vehicle Routing Problems,” in *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad, Eds., Elsevier Science Publishers, B. V., 1988, pp. 223–248. [Online]. Available: http://www.martrans.org/documents/2008/rst/dvrp%20psaraftis%2088.pdf.

[105] H. Handa, L. Chapman, and X. Yao, “Dynamic salting route optimisation using evolutionary computation,” *2005 IEEE Congress on Evolutionary Computation*, vol. 1, no. 1, pp. 158–165, 2005.

[106] M. Tagmouti, M. Gendreau, and J. Y. Potvin, “A dynamic capacitated arc routing problem with time-dependent service costs,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 1, pp. 20–28, 2011, ISSN: 0968090X. DOI: 10.1016/j.trc.2010.02.003. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2010.02.003.

[107] H. Tong, L. L. Minku, S. Menzel, B. Sendhoff, and X. Yao, “What makes the dynamic capacitated arc routing problem hard to solve: Insights from fitness landscape analysis,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO ’22, Boston, Massachusetts: Association for Computing Machinery, 2022, pp. 305–313, ISBN: 9781450392372. DOI: 10.1145/3512290.3528756. [Online]. Available: https://doi.org/10.1145/3512290.3528756.

[108] D. Muñoz-Carpintero, D. Sáez, C. E. Cortés, and A. Núñez, “A Methodology Based on Evolutionary Algorithms to Solve a Dynamic Pickup and Delivery Problem Under a Hybrid Predictive Control Approach,” *Transportation Science*, vol. 49, no. 2, pp. 239–253, 2015, ISSN: 0041-1655. DOI: 10.1287/trsc.2014.0569.

[109] D. Sáez, C. E. Cortés, and A. Núñez, “Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering,” *Computers and Operations Research*, vol. 35, no. 11, pp. 3412–3438, 2008, ISSN: 03050548. DOI: 10.1016/j.cor.2007.01.025.

[110] Z. Zhu, J. Xiao, S. He, Z. Ji, and Y. Sun, “A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem,” *Information Sciences*, vol. 329, pp. 73–89, 2016, ISSN: 00200255. DOI: 10.1016/j.ins.2015.09.006. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2015.09.006.

[111] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013. DOI: 10.1016/j.ejor.2012.08.015. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2012.08.015.

[112] U. Ritzinger, J. Puchinger, and R. F. Hartl, “A survey on dynamic and stochastic vehicle routing problems,” *International Journal of Production Research*, vol. 54, no. 1, pp. 215–231, 2016. DOI: 10.1080/00207543.2015.1043403. [Online]. Available: http://dx.doi.org/10.1080/00207543.2015.1043403.

[113] H. Kanoh and K. Hara, "Hybrid genetic algorithm for dynamic multi-objective route planning with predicted traffic in a real-world road network," *GECCO'08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation 2008*, pp. 657–664, 2008. DOI: 10.1145/1389095.1389226.

[114] H. Tong, L. L. Minku, S. Menzel, B. Sendhoff, and X. Yao, "Benchmarking Dynamic Capacitated Arc Routing Algorithms Using Real-World Traffic Simulation," *2022 IEEE Congress on Evolutionary Computation, CEC 2022 - Conference Proceedings*, 2022. DOI: 10.1109/CEC55065.2022.9870399.

[115] F. O. De França, L. C. Gomes, L. N. De Castro, and F. J. Von Zuben, "Handling time-varying TSP instances," *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pp. 2830–2837, 2006. DOI: 10.1109/cec.2006.1688664.

[116] Lishan Kang, Aimin Zhou, B. McKay, Yan Li, and Zhuo Kang, "Benchmarking algorithms for dynamic travelling salesman problems," vol. 1, no. 2, pp. 1286–1292, 2005. DOI: 10.1109/cec.2004.1331045.

[117] G. A. Godfrey and W. B. Powell, "An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times," *Transportation Science*, vol. 36, no. 1, pp. 40–54, 2002, ISSN: 00411655. DOI: 10.1287/trsc.36.1.40.572.

[118] G. A. Godfrey and W. B. Powell, "An adaptive dynamic programming algorithm for dynamic fleet management, II: Multiperiod travel times," *Transportation Science*, vol. 36, no. 1, pp. 40–54, 2002, ISSN: 00411655. DOI: 10.1287/trsc.36.1.40.572.

[119] A. K. Hutzschenreuter, P. A. Bosman, and H. La Poutré, "Evolutionary multiobjective optimization for dynamic hospital resource management," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5467 LNCS, pp. 320–334, 2010, ISSN: 03029743. DOI: 10.1007/978-3-642-01020-0_27.

[120] F. Yu, F. Tu, and K. R. Pattipati, "Integration of a holonic organizational control architecture and multiobjective evolutionary algorithm for flexible distributed scheduling," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 38, no. 5, pp. 1001–1017, 2008, ISSN: 10834427. DOI: 10.1109/TSMCA.2008.923082.

[121] Z. Bingul, A. L. I. S. Sekmen, and S. Zein-sabatto, "Adaptive genetic algorithms applied to dynamic multiobjective problems," *Applied Soft Computing*, vol. 7, no. 3, pp. 791–799, 2007.

[122] A. Haghani and S. Jung, "A dynamic vehicle routing problem with time-dependent travel times," *Computers & Operations Research*, vol. 32, no. 11, pp. 2959–2986, 2005, ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2004.04.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054804000887.

[123] E. Gazioğlu and A. S. Etaner-Uyar, "Experimental analysis of a statistical multiploid genetic algorithm for dynamic environments," *Engineering Science and Technology, an International Journal*, vol. 35, 2022, ISSN: 22150986. DOI: 10.1016/j.jestch.2022.101173.

[124] A. J. Kleywegt and J. D. Papastavrou, "The Dynamic and Stochastic Knapsack Problem," *Operations Research*, vol. 46, no. 1, pp. 17–35, 1998, ISSN: 0030-364X. DOI: 10.1287/opre.46.1.17.

[125] G. Colombo and C. Mumford, "Comparing Algorithms, Representations and Operators for the Multi-Objective Knapsack Problem," *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005*, vol. 2, pp. 1268–1275, 2005. DOI: 10.1109/CEC.2005.1554836.

[126] T. F. de Queiroz Lafetá and G. M. B. de Oliveira, "Applying dynamic evolutionary optimization to the multiobjective knapsack problem," in *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I*, Rio Grande, Brazil: Springer-Verlag, 2020, pp. 49–63, ISBN: 978-3-030-61376-1. DOI: 10.1007/978-3-030-61377-8_4. [Online]. Available: https://doi.org/10.1007/978-3-030-61377-8_4.

[127] V. Roostapour, A. Neumann, and F. Neumann, "On the Performance of Baseline Evolutionary Algorithms on the Dynamic Knapsack Problem," *Parallel Problem Solving from Nature - PPSN XV - 15th International Conference, Coimbra, Portugal, September 8-12, 2018, Proceedings, Part I*, no. October, 2018. DOI: 10.1007/978-3-319-99253-2. [Online]. Available: https://doi.org/10.1007/978-3-319-99253-2.

[128] T. C. Perry and J. C. Hartman, "Allocating Manufacturing Capacity by Solving Dynamic Stochastic Multi-Knapsack Problem," *Technical Report 04T-009, Lehigh University, Pennsylvania*, 2004.

[129] T. C. Perry and J. C. Hartman, "An approximate dynamic programming approach to solving a dynamic, stochastic multiple knapsack problem," *International Transactions in Operational Research*, vol. 16, no. 3, pp. 347–359, 2009, ISSN: 14753995. DOI: 10.1111/j.1475-3995.2008.00679.x.

[130] A. Beham, S. Leitner, J. Karder, B. Werth, and S. Wagner, "DynStack - A Benchmarking Framework for Dynamic Optimization Problems in Warehouse Operations," *GECCO 2022 Companion - Proceedings of the 2022 Genetic and Evolutionary Computation Conference*, pp. 1984–1991, 2022. DOI: 10.1145/3520304.3533957.

[131] J. Karder, A. Beham, B. Werth, S. Wagner, and M. Affenzeller, "Integrated Machine Learning in Open-Ended Crane Scheduling: Learning Movement Speeds and Service Times," *Procedia Computer Science*, vol. 200, pp. 1031–1040, 2022, ISSN: 18770509. DOI: 10.1016/j.procs.2022.01.302.

[132] J. A. Gunawardhana, H. N. Perera, and A. Thibbotuwawa, "Rule-based dynamic container stacking to optimize yard operations at port terminals," *Maritime Transport Research*, vol. 2, no. July, p. 100 034, 2021, ISSN: 2666822X. DOI: 10.1016/j.martra.2021.100034. [Online]. Available: https://doi.org/10.1016/j.martra.2021.100034.

[133] T. Park, R. Choe, Y. Hun Kim, and K. Ryel Ryu, "Dynamic adjustment of container stacking policy in an automated container terminal," *International Journal of Production Economics*, vol. 133, no. 1, pp. 385–392, 2011, ISSN: 09255273. DOI: 10.1016/j.ijpe.2010.03.024. [Online]. Available: http://dx.doi.org/10.1016/j.ijpe.2010.03.024.

[134] L. Epstein and M. Levy, "Dynamic multi-dimensional bin packing," *Journal of Discrete Algorithms*, vol. 8, no. 4, pp. 356–372, 2010, ISSN: 15708667. DOI: 10.1016/j.jda.2010.07.002. [Online]. Available: http://dx.doi.org/10.1016/j.jda.2010.07.002.

[135] D. S. J. E. G. Coffman Jr. M. R. Garey, "Dynamic Bin Packing," *SIAM Journal on Computing*, vol. 12, no. 2, pp. 227–259, 1983.

[136] A. Baykasoğlu and F. B. Ozsoydan, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3712–3725, 2014, ISSN: 09574174. DOI: 10.1016/j.eswa.2013.11.040.

[137] M. R. Bonyadi, Z. Michalewicz, and L. Barone, "The travelling thief problem: The first step in the transition from theoretical problems to realistic problems," *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, pp. 1037–1044, 2013. DOI: 10.1109/CEC.2013.6557681.

[138] S. Polyakovskiy and F. Neumann, "Packing While Traveling: Mixed Integer Programming for a Class of Nonlinear Knapsack Problems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9075, pp. 332–346, 2015, ISSN: 16113349. DOI: 10.1007/978-3-319-18008-3. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84929648636%7B%5C&%7DpartnerID=tZOtx3y1.

[139] C. E. Cortés, D. Sáez, F. Milla, A. Núñez, and M. Riquelme, "Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 757–769, 2010, ISSN: 0968090X. DOI: 10.1016/j.trc.2009.05.016. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2009.05.016.

[140] H. Handa, L. Chapman, and X. Yao, "Robust salting route optimization using evolutionary algorithms," *Studies in Computational Intelligence*, vol. 51, no. 2007, pp. 497–517, 2007, ISSN: 1860949X. DOI: 10.1007/978-3-540-49774-5_22.

[141] K. Deb, U. B. Rao N., and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4403 LNCS, pp. 803–817, 2007, ISSN: 16113349. DOI: 10.1007/978-3-540-70928-2_60.

[142] H. Xia, P. Jia, and L. Ma, "Robust Multi-Objective Optimization for Gas Turbine Operation Based on Kriging Surrogate Model," *Chinese Control Conference, CCC*, vol. 2021-July, pp. 6704–6709, 2021, ISSN: 21612927. DOI: 10.23919/CCC52363.2021.9550112.

[143] E. T. Ceran, "DYNAMIC ALLOCATION OF RENEWABLE ENERGY THROUGH A STOCHASTIC KNAPSACK PROBLEM FORMULATION FOR AN ACCESS POINT ON THE MOVE," *Masters Thesis, Middle East Technical University*, vol. 151, no. 4, pp. 1–46, 2014, ISSN: 10974172. DOI: 10.1016/j.cell.2009.01.043. [Online]. Available: http://dx.doi.org/10.1016/j.biochi.2015.03.025%0Ahttp://dx.doi.org/10.1038/nature10402%0Ahttp://dx.doi.org/10.1038/nature21059%0Ahttp://journal.stainkudus.ac.id/index.php/equilibrium/article/view/1268/1127%0Ahttp://dx.doi.org/10.1038/nrmicro2577%0Ahttp://.

[144] M. Sharafi and T. Y. ElMekkawy, "A dynamic MOPSO algorithm for multiobjective optimal design of hybrid renewable energy systems," *International journal of energy research*, vol. 38, pp. 1949–1963, 2014. DOI: 10.1002/er. arXiv: arXiv:1011.1669v3.

[145] X. Shi, G. Bao, K. Ding, and L. Lu, "Multi-Objective Optimal Dispatch Considering Wind Power and Interactive Load for Power System," *Energy and Power Engineering*, vol. 10, no. 04, pp. 1–10, 2018, ISSN: 1949-243X. DOI: 10.4236/epe.2018.104b001.

[146] S. L. Jiang, Q. Liu, I. D. L. Bogle, and Z. Zheng, "A Self-Learning Based Dynamic Multi-Objective Evolutionary Algorithm for Resilient Scheduling Problems in Steelmaking Plants," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2022, ISSN: 15583783. DOI: 10.1109/TASE.2022.3168385.

[147] Z. Zhang and S. Qian, "Multi-objective immune optimization in dynamic environments and its application to signal simulation," *2009 International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2009*, vol. 3, pp. 246–250, 2009. DOI: 10.1109/ICMTMA.2009.141.

[148] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Applied Soft Computing Journal*, vol. 8, no. 2, pp. 959–971, 2008, ISSN: 15684946. DOI: 10.1016/j.asoc.2007.07.005.

[149] E. Vellasques, R. Sabourin, and E. Granger, "A Dual-Purpose Memory Approach for Dynamic Particle Swarm Optimization of Recurrent Problems," *Studies in Computational Intelligence*, vol. 621, pp. 1–9, 2016, ISSN: 1860949X. DOI: 10.1007/978-3-319-26450-9.

[150] V. Mytilinou and A. J. Kolios, "A multi-objective optimisation approach applied to offshore wind farm location selection," *Journal of Ocean Engineering and Marine Energy*, vol. 3, no. 3, pp. 265–284, 2017, ISSN: 21986452. DOI: 10.1007/s40722-017-0092-8.

[151] F. G. Montoya, F. Manzano-Agugliaro, S. López-Márquez, Q. Hernández-Escobedo, and C. Gil, "Wind turbine selection for wind farm layout using multi-objective evolutionary algorithms," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6585–6595, 2014, ISSN: 09574174. DOI: 10.1016/j.eswa.2014.04.044. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2014.04.044.

[152] T. Ackling, C. Denison, and F. Neumann, “Fast and Effective Multi-Objective Optimisation of Wind Turbine Placement Categories and Subject Descriptors,” *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO ’13*, pp. 1381–1388, 2013.

[153] S. Rodrigues, P. Bauer, and P. A. Bosman, “Multi-objective optimization of wind farm layouts – Complexity, constraint handling and scalability,” *Renewable and Sustainable Energy Reviews*, vol. 65, pp. 587–609, 2016, ISSN: 18790690. DOI: `10.1016/j.rser.2016.07.021`. [Online]. Available: `http://dx.doi.org/10.1016/j.rser.2016.07.021`.

[154] L. Y. Pao and K. E. Johnson, “A Tutorial on the Dynamics and Control of Wind Turbines and Wind Farms,” *American Control Conference June 2009*, 2009, ISSN: 01767364. DOI: `10.1007/BF02982392`.

[155] D. Lombardi, “Dynamics of Offshore Wind Turbines,” *Masters Thesis, University of Bristol*, no. May, 2010.

[156] C. Gallego, P. Pinson, H. Madsen, A. Costa, and A. Cuerva, “Influence of local wind speed and direction on wind power dynamics - Application to offshore very short-term forecasting,” *Applied Energy*, vol. 88, no. 11, pp. 4087–4096, 2011, ISSN: 03062619. DOI: `10.1016/j.apenergy.2011.04.051`. [Online]. Available: `http://dx.doi.org/10.1016/j.apenergy.2011.04.051`.

[157] S. Vipsita and S. K. Rath, “Protein superfamily classification using adaptive evolutionary radial basis function network,” *International Journal of Computational Intelligence and Applications*, vol. 11, no. 4, pp. 1–22, 2012, ISSN: 14690268. DOI: `10.1142/S1469026812500265`.

[158] S. Y. Zeng, G. Chen, L. Zheng, *et al.*, “A dynamic multi-objective evolutionary algorithm based on an orthogonal design,” *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pp. 573–580, 2006. DOI: `10.1109/cec.2006.1688361`.

[159] C. Fonteix, S. Massebeuf, F. Pla, and L. N. Kiss, “Multicriteria optimization of an emulsion polymerization process,” in *European Journal of Operational Research*, vol. 153, 2004, pp. 350–359. DOI: `10.1016/S0377-2217(03)00157-7`.

[160] S. Massebeuf, C. Fonteix, S. Hoppe, and F. Pla, “Development of new concepts for the control of polymerization processes: Multiobjective optimization and decision engineering. II. Application of a Choquet integral to an emulsion copolymerization process,” *Journal of Applied Polymer Science*, vol. 120, no. 6, pp. 3421–3434, 2003, ISSN: 00218995. DOI: `10.1002/app.33348`.

[161] D. Izzo and M. Manuel López-Ibáñez, “Optimization challenges at the European Space Agency,” pp. 1542–1553, 2022.

[162] K. Deb, N. Padhye, and G. Neema, “Interplanetary Trajectory Optimization with Swing-Bys Using Evolutionary Multi-objective Optimization,” *ISICA 2007: Advances in Computation and Intelligence*, pp. 26–35, 2007.

[163] C. Cruz, J. R. González, and D. A. Pelta, “Optimization in dynamic environments: A survey on problems, methods and measures,” *Soft Computing*, vol. 15, no. 7, pp. 1427–1448, 2011, ISSN: 14327643. DOI: `10.1007/s00500-010-0681-0`.

[164] M. Helbig and A. P. Engelbrecht, “Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems,” *Swarm and Evolutionary Computation*, vol. 14, pp. 31–47, 2014, ISSN: 22106502. DOI: `10.1016/j.swevo.2013.08.004`.

[165] M. Mavrovouniotis, C. Li, and S. Yang, “A survey of swarm intelligence for dynamic optimization: Algorithms and applications,” *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017, ISSN: 22106502. DOI: `10.1016/j.swevo.2016.12.005`.

[166] M. Liu and W. Zeng, “A fast evolutionary algorithm for dynamic bi-objective optimization problems,” *ICCSE 2012 - Proceedings of 2012 7th International Conference on Computer Science and Education*, vol. 1, no. Iccse, pp. 130–134, 2012. DOI: `10.1109/ICCSE.2012.6295042`.

[167] S. B. Gee, K. C. Tan, and C. Alippi, "Solving Multiobjective Optimization Problems in Unknown Dynamic Environments: An Inverse Modeling Approach," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4223–4234, 2017, ISSN: 21682267. DOI: `10.1109/TCYB.2016.2602561`.

[168] S. Yang, "Memory-based immigrants for genetic algorithms in dynamic environments," *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*, p. 1115, 2005. DOI: `10.1145/1068009.1068196`. [Online]. Available: `http://portal.acm.org/citation.cfm?doid=1068009.1068196`.

[169] F. Vavak, K. Jukes, and T. C. Fogarty, "Learning the local search range for genetic optimization in nonstationary environments," *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, pp. 355–360, 1997. DOI: `10.1109/icec.1997.592335`.

[170] L. T. Bui, J. Branke, and H. A. Abbass, "Diversity as a selection pressure in dynamic environments," *GECCO 2005 - Genetic and Evolutionary Computation Conference*, pp. 1557–1558, 2005. DOI: `10.1145/1068009.1068257`.

[171] N. Mori, H. Kita, and Y. Nishikawa, "Adaptation to a changing environment by means of the thermodynamical genetic algorithm," *In: Voigt, HM., Ebeling, W., Rechenberg, I., Schwefel, HP. (eds) Parallel Problem Solving from Nature — PPSN IV. PPSN 1996. Lecture Notes in Computer Science, vol 1141. Springer, Berlin, Heidelberg.*, 1996.

[172] N. Mori, H. Kita, and Y. Nishikawa, "Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1498 LNCS, pp. 149–158, 1998, ISSN: 16113349. DOI: `10.1007/bfb0056858`.

[173] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983, ISSN: 00368075. DOI: `10.1126/science.220.4598.671`. [Online]. Available: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.4175`.

[174] J. Branke, T. Kaußler, C. Schmidt, and H. Schmeck, "A Multi-population Approach to Dynamic Optimization Problems," *Evolutionary Design and Manufacture*, no. January, 2000. DOI: `10.1007/978-1-4471-0519-0`.

[175] F. Oppacher, M. Wineberg, *et al.*, "The shifting balance genetic algorithm: Improving the ga in a dynamic environment," in *Proceedings of the genetic and evolutionary computation conference*, vol. 1, 1999, pp. 504–510.

[176] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 966–978, 2014, ISSN: 21682267. DOI: `10.1109/TCYB.2013.2278188`.

[177] X. F. Liu, Y. R. Zhou, and X. Yu, "Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization," *Applied Soft Computing Journal*, vol. 87, p. 105 988, 2020, ISSN: 15684946. DOI: `10.1016/j.asoc.2019.105988`. [Online]. Available: `https://doi.org/10.1016/j.asoc.2019.105988`.

[178] R. Liu, J. Li, J. Fan, and L. Jiao, "A dynamic multiple populations particle swarm optimization algorithm based on decomposition and prediction," *Applied Soft Computing Journal*, vol. 73, pp. 434–459, 2018, ISSN: 15684946. DOI: `10.1016/j.asoc.2018.08.015`. [Online]. Available: `https://doi.org/10.1016/j.asoc.2018.08.015`.

[179] M. Camara, J. Ortega, and F. J. Toro, "Parallel Processing for Multi-objective Optimization in Dynamic Environments," *2007 IEEE International Parallel and Distributed Processing Symposium*, pp. 1–8, 2007. DOI: `10.1109/IPDPS.2007.370433`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4228161`.

[180] M. Cámara Sola, *Parallel Processing for Dynamic Multi-objective Optimization*. 2010, ISBN: 9788469344231.

[181] M. Cámara, J. O. Ortega, and F. de Toro, "High performance computing for dynamic multi-objective optimisation," *International Journal of High Performance Systems Architecture*, vol. 1, no. 4, pp. 241–250, 2008, ISSN: 17516536. DOI: 10.1504/IJHPSA.2008.024208.

[182] W. Li and M. Feng, "Aparallel procedure for dynamic multi-objective TSP," *Proceedings of the 2012 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA 2012*, pp. 1–8, 2012. DOI: 10.1109/ISPA.2012.10.

[183] K. C. Tan, Y. J. Yang, and C. K. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 527–549, 2006, ISSN: 1089778X. DOI: 10.1109/TEVC.2005.860762.

[184] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A Similarity-Based Cooperative Co-Evolutionary Algorithm for Dynamic Interval Multiobjective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 142–156, 2020, ISSN: 19410026. DOI: 10.1109/TEVC.2019.2912204.

[185] R. Liu, Y. Chen, W. Ma, C. Mu, and L. Jiao, "A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model," *Soft Computing*, vol. 18, no. 10, pp. 1913–1929, 2014, ISSN: 14337479. DOI: 10.1007/s00500-013-1175-7.

[186] R. Liu, J. Li, J. Fan, C. Mu, and L. Jiao, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017, ISSN: 03772217. DOI: 10.1016/j.ejor.2017.03.048. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2017.03.048.

[187] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research*, vol. 202, no. 1, pp. 42–54, 2010. DOI: 10.1016/j.ejor.2009.05.005. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2009.05.005.

[188] H. Xie, J. Zou, S. Yang, J. Zheng, J. Ou, and Y. Hu, "A decision variable classification-based cooperative coevolutionary algorithm for dynamic multiobjective optimization," *Information Sciences*, vol. 560, pp. 307–330, 2021, ISSN: 00200255. DOI: 10.1016/j.ins.2021.01.021. [Online]. Available: https://doi.org/10.1016/j.ins.2021.01.021.

[189] J. Lewis, E. Hart, and G. Ritchie, "A comparison of dominance mechanisms and simple mutation on non-stationary problems," pp. 139–148, 1998, ISSN: 0302-9743 (Print) 1611-3349 (Online). DOI: 10.1007/BFb0056857. [Online]. Available: http://link.springer.com/10.1007/BFb0056857.

[190] K. P. Ng and K. C. Wong, "A new diploid scheme and dominance change mechanism for non-stationary function optimization," in *ICGA*, 1995.

[191] A. Ş. Uyar and A. E. Harmanci, "A new population based adaptive domination change mechanism for diploid genetic algorithms in dynamic environments," *Soft Computing*, vol. 9, no. 11, pp. 803–814, 2005, ISSN: 14327643. DOI: 10.1007/s00500-004-0421-4.

[192] S. Yang, "Associative memory scheme for genetic algorithms in dynamic environments," *Applications of Evolutionary Computing Lecture Notes in Computer Science*, pp. 788–799, 2006.

[193] C. Ryan, "Diploidy without dominance," *In Proc. Third Nordic Workshop on Genetic Algorithms*, pp. 63–70, 1997.

[194] E. Collingwood, D. Corne, and P. Ross, "Useful diversity via multiploidy," *Proceedings of the IEEE Conference on Evolutionary Computation*, no. June 1996, pp. 810–813, 1996. DOI: 10.1109/icec.1996.542705.

[195] A. M. Turky and S. Abdullah, "A multi-population harmony search algorithm with external archive for dynamic optimization problems," *Information Sciences*, vol. 272, pp. 84–95, 2014, ISSN: 00200255. DOI: 10.1016/j.ins.2014.02.084. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2014.02.084.

[196] P. Jócko, B. M. Ombuki-Berman, and A. P. Engelbrecht, "Multi-guide particle swarm optimisation archive management strategies for dynamic optimisation problems," *Swarm Intelligence*, vol. 16, no. 2, pp. 143–168, 2022, ISSN: 19353820. DOI: 10.1007/s11721-022-00210-3. [Online]. Available: https://doi.org/10.1007/s11721-022-00210-3.

[197] M. Helbig and A. P. Engelbrecht, "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," *Procroceeding of 2011 IEEE Ccongress on Evolutionary Computation*, pp. 2047–2054, 2011, ISSN: Pending. DOI: 10.1109/CEC.2011.5949867.

[198] Y. Wu, L. Shi, and X. Liu, "A new dynamic strategy for dynamic multi-objective optimization," *Information Sciences*, vol. 529, pp. 116–131, 2020, ISSN: 00200255. DOI: 10.1016/j.ins.2020.04.011. [Online]. Available: https://doi.org/10.1016/j.ins.2020.04.011.

[199] P. Filipiak and P. Lipinski, "Infeasibility Driven Evolutionary Algorithm with Feed-Forward Prediction Strategy for Dynamic Constrained Optimization Problems," *Proceedings - 2014 EvoApplications Conference*, 2014, ISSN: 16113349. DOI: 10.1007/978-3-662-45523-4.

[200] A. Muruganantham, Y. Zhao, S. B. Gee, X. Qiu, and K. C. Tan, "Dynamic multiobjective optimization using evolutionary algorithm with kalman filter," *Procedia Computer Science*, vol. 24, pp. 66–75, 2013, ISSN: 18770509. DOI: 10.1016/j.procs.2013.10.028. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.10.028.

[201] Q. Zhang, X. He, S. Yang, Y. Dong, H. Song, and S. Jiang, "Solving dynamic multi-objective problems using polynomial fitting-based prediction algorithm," *Inf. Sci.*, vol. 610, no. C, pp. 868–886, 2022, ISSN: 0020-0255. DOI: 10.1016/j.ins.2022.08.020. [Online]. Available: https://doi.org/10.1016/j.ins.2022.08.020.

[202] G. Li, Y. Liu, and X. Deng, "A prediction method based on fractional order displacement for dynamic multiobjective optimization," *ISA Transactions*, vol. 130, Mar. 2022. DOI: 10.1016/j.isatra.2022.03.015.

[203] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization evolutionary algorithms: a Forward-Looking approach," *Proceedings of ACM GECCO*, vol. 4, pp. 1201–1208, 2006. DOI: 10.1145/1143997.1144187.

[204] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, "A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization," *Applied Soft Computing Journal*, vol. 61, pp. 806–818, 2017, ISSN: 15684946. DOI: 10.1016/j.asoc.2017.08.004. [Online]. Available: https://doi.org/10.1016/j.asoc.2017.08.004.

[205] J. Zheng, Y. Zhou, J. Zou, S. Yang, J. Ou, and Y. Hu, "A prediction strategy based on decision variable analysis for dynamic Multi-objective Optimization," *Swarm and Evolutionary Computation*, vol. 60, no. June 2020, p. 100 786, 2021, ISSN: 22106502. DOI: 10.1016/j.swevo.2020.100786. [Online]. Available: https://doi.org/10.1016/j.swevo.2020.100786.

[206] X. Li, J. Yang, H. Sun, Z. Hu, and A. Cao, "A dual prediction strategy with inverse model for evolutionary dynamic multiobjective optimization," *ISA Transactions*, vol. 117, no. xxxx, pp. 196–209, 2021, ISSN: 00190578. DOI: 10.1016/j.isatra.2021.01.053. [Online]. Available: https://doi.org/10.1016/j.isatra.2021.01.053.

[207] H. Sun, A. Cao, Z. Hu, X. Li, and Z. Zhao, "A novel quantile-guided dual prediction strategies for dynamic multi-objective optimization," *Information Sciences*, vol. 579, pp. 751–775, 2021, ISSN: 00200255. DOI: 10.1016/j.ins.2021.08.027. [Online]. Available: https://doi.org/10.1016/j.ins.2021.08.027.

[208] C. Wang, G. G. Yen, and M. Jiang, "A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 56, no. September 2019, p. 100 695, 2020, ISSN: 22106502. DOI: 10.1016/j.swevo.2020.100695. [Online]. Available: https://doi.org/10.1016/j.swevo.2020.100695.

[209] W. Zhou, L. Feng, K. C. Tan, M. Jiang, and Y. Liu, "Evolutionary Search with Multi-View Prediction for Dynamic Multi-objective Optimization," *IEEE Transactions on Evolutionary Computation*, no. c, pp. 1–15, 2021, ISSN: 19410026. DOI: 10.1109/TEVC.2021.3135020.

[210] K. Yu, D. Zhang, J. Liang, *et al.*, "A Correlation-Guided Layered Prediction Approach for Evolutionary Dynamic Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 8, 2022, ISSN: 19410026. DOI: 10.1109/TEVC.2022.3193287.

[211] Q. Li, X. Liu, F. Wang, S. Wang, P. Zhang, and X. Wu, "A Framework Based on Generational and Environmental Response Strategies for Dynamic Multi-objective Optimization," no. Qingya Li, 2022. arXiv: 2207.04047. [Online]. Available: http://arxiv.org/abs/2207.04047.

[212] Q. Zhao, B. Yan, Y. Shi, and M. Middendorf, "Evolutionary Dynamic Multiobjective Optimization via Learning From Historical Search Process," *IEEE Transactions on Cybernetics*, pp. 1–12, 2021, ISSN: 21682275. DOI: 10.1109/TCYB.2021.3059252.

[213] T. Macias-Escobar, L. Cruz-Reyes, H. Fraire, and B. Dorronsoro, "Plane Separation: A method to solve dynamic multi-objective optimization problems with incorporated preferences," *Future Generation Computer Systems*, vol. 110, no. xxxx, pp. 864–875, 2020, ISSN: 0167739X. DOI: 10.1016/j.future.2019.10.039. [Online]. Available: https://doi.org/10.1016/j.future.2019.10.039.

[214] Y. Wang, T. Du, T. Liu, and L. Zhang, "Dynamic Multiobjective Squirrel Search Algorithm Based on Decomposition with Evolutionary Direction Prediction and Bidirectional Memory Populations," *IEEE Access*, vol. 7, pp. 115 997–116 013, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2932883.

[215] Z. Peng, J. Zheng, J. Zou, and M. Liu, "Novel prediction and memory strategies for dynamic multiobjective optimization," *Soft Computing*, vol. 19, no. 9, pp. 2633–2653, 2015, ISSN: 14337479. DOI: 10.1007/s00500-014-1433-3.

[216] Y. Guo, H. Yang, M. Chen, J. Cheng, and D. Gong, "Ensemble prediction-based dynamic robust multi-objective optimization methods," *Swarm and Evolutionary Computation*, 2019, ISSN: 22106502. DOI: 10.1016/j.swevo.2019.03.015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210650218302712.

[217] F. Wang, Y. Li, F. Liao, and H. Yan, "An ensemble learning based prediction strategy for dynamic multi-objective optimization," *Applied Soft Computing Journal*, vol. 96, p. 106 592, 2020, ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106592. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106592.

[218] S. Jiang and S. Yang, "An Improved Multiobjective Optimization Evolutionary Algorithm Based on Decomposition for Complex Pareto Fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016, ISSN: 21682267. DOI: 10.1109/TCYB.2015.2403131.

[219] Y. Chen, J. Zou, Y. Liu, S. Yang, J. Zheng, and W. Huang, "Combining a hybrid prediction strategy and a mutation strategy for dynamic multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 70, no. July 2021, p. 101 041, 2022, ISSN: 22106502. DOI: 10.1016/j.swevo.2022.101041. [Online]. Available: https://doi.org/10.1016/j.swevo.2022.101041.

[220] Y. Hu, J. Zheng, J. Zou, S. Jiang, and S. Yang, "Dynamic multi-objective optimization algorithm based decomposition and preference," *Information Sciences*, vol. 571, pp. 175–190, 2021, ISSN: 00200255. DOI: 10.1016/j.ins.2021.04.055. [Online]. Available: https://doi.org/10.1016/j.ins.2021.04.055.

[221] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Applied Soft Computing Journal*, vol. 58, pp. 631–647, 2017, ISSN: 15684946. DOI: 10.1016/j.asoc.2017.05.008. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2017.05.008.

[222] Y. Wang, K. Li, and G.-G. Wang, “Combining key-points-based transfer learning and hybrid prediction strategies for dynamic multi-objective optimization,” *Mathematics*, vol. 10, no. 12, 2022, ISSN: 2227-7390. DOI: 10.3390/math10122117. [Online]. Available: https://www.mdpi.com/2227-7390/10/12/2117.

[223] D. J. Wang, F. Liu, and Y. Jin, “A multi-objective evolutionary algorithm guided by directed search for dynamic scheduling,” *Computers and Operations Research*, vol. 79, pp. 279–290, 2017, ISSN: 03050548. DOI: 10.1016/j.cor.2016.04.024. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2016.04.024.

[224] Y. Hu, J. Ou, J. Zheng, J. Zou, S. Yang, and G. Ruan, “Solving dynamic multi-objective problems with an evolutionary multi-directional search approach,” *Knowledge-Based Systems*, vol. 194, 2020, ISSN: 09507051. DOI: 10.1016/j.knosys.2019.105175. [Online]. Available: https://doi.org/10.1016/j.knosys.2019.105175.

[225] R. Shang, L. Jiao, Y. Ren, L. Li, and L. Wang, “Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization,” *Soft Computing*, vol. 18, no. 4, pp. 743–756, 2014, ISSN: 14337479. DOI: 10.1007/s00500-013-1085-8.

[226] R. K. Ursem, “Multinational GAs: Multimodal Optimization Techniques in Dynamic Environments (2000),” *In Proceedings of the Second Genetic and Evolutionary Computation Conference*, 2000.

[227] M. Liu and Y. Liu, “A dynamic evolutionary multi-objective optimization algorithm based on decomposition and adaptive diversity introduction,” *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, no. 14, pp. 235–240, 2016. DOI: 10.1109/FSKD.2016.7603180.

[228] S. Wan and D. Wang, “A novel differential evolution for dynamic multiobjective optimization with adaptive immigration scheme,” *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, ICCSNT 2013*, pp. 502–507, 2014. DOI: 10.1109/ICCSNT.2013.6967163.

[229] J. Zou, Q. Li, S. Yang, J. Zheng, Z. Peng, and T. Pei, “A dynamic multiobjective evolutionary algorithm based on a dynamic evolutionary environment model,” *Swarm and Evolutionary Computation*, vol. 44, no. November 2017, pp. 247–259, 2019, ISSN: 22106502. DOI: 10.1016/j.swevo.2018.03.010. [Online]. Available: https://doi.org/10.1016/j.swevo.2018.03.010.

[230] X. Li, J. Branke, and T. Blackwell, “Particle swarm with speciation and adaptation in a dynamic environment,” *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO ’06*, no. January, p. 51, 2006. DOI: 10.1145/1143997.1144005. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1143997.1144005.

[231] R. Liu, J. Li, Y. Jin, and L. Jiao, “A self-adaptive response strategy for dynamic multiobjective evolutionary optimization based on objective space decomposition,” *Evolutionary Computation*, vol. 29, no. 4, pp. 491–519, 2021, ISSN: 15309304. DOI: 10.1162/evco_a_00289.

[232] L. Chen, H. Wang, D. Pan, *et al.*, “Dynamic multiobjective evolutionary algorithm with adaptive response mechanism selection strategy,” *Knowledge-Based Systems*, vol. 246, p. 108 691, Apr. 2022. DOI: 10.1016/j.knosys.2022.108691.

[233] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, “A Fast Dynamic Evolutionary Multiobjective Algorithm via Manifold Transfer Learning,” *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3417–3428, 2021, ISSN: 21682275. DOI: 10.1109/TCYB.2020.2989465.

[234] J. Li, T. Sun, Q. Lin, M. Jiang, and K. C. Tan, “Reducing Negative Transfer Learning via Clustering for Dynamic Multiobjective Optimization,” *IEEE Transactions on Evolutionary Computation*, no. c, pp. 1–15, 2022, ISSN: 19410026. DOI: 10.1109/TEVC.2022.3144180.

[235] G. Chen, N. Guo, M. Huang, D. Gong, and Z. Yu, “A domain adaptation learning strategy for dynamic multiobjective optimization,” *Information Sciences*, vol. 606, May 2022. DOI: 10.1016/j.ins.2022.05.050.

[236] M. Helbig and A. P. Engelbrecht, "Heterogeneous dynamic vector evaluated particle swarm optimisation for dynamic multi-objective optimisation," *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 3151–3159, 2015. DOI: 10.1109/CEC.2014.6900303.

[237] A. Aboud, N. Rokbani, S. Mirjalili, and A. Alimi, "A Distributed Bi-behaviors Crow Search Algorithm for Dynamic Multi-Objective Optimization and Many-Objective Optimization," 2021. DOI: 10.36227/techrxiv.16607858.v2. [Online]. Available: /articles/preprint/A_Distributed_Bi-behaviors_Crow_Search_Algorithm_for_Dynamic_Multi-Objective_Optimization_and_Many-Objective_Optimization/16607858/2.

[238] M. Greeff and A. P. Engelbrecht, "Dynamic multi-objective optimisation using PSO," *Studies in Computational Intelligence*, vol. 261, pp. 105–123, 2010, ISSN: 1860949X. DOI: 10.1007/978-3-642-05165-4_5.

[239] A. Aboud, N. Rokbani, B. Neji, Z. A. Al Barakeh, S. Mirjalili, and A. M. Alimi, "A Distributed Bi-Behaviors Crow Search Algorithm for Dynamic Multi-Objective Optimization and Many-Objective Optimization Problems," *Applied Sciences*, vol. 12, no. 19, p. 9627, 2022, ISSN: 20763417. DOI: 10.3390/app12199627.

[240] B. Ombuki-Berman, P. Joćko, and A. Engelbrecht, "Quantum Multi-guide Particle Swarm Optimisation for Dynamic Multi-objective Optimisation Problems," *Research Square*, pp. 1–27, 2022. DOI: 10.21203/rs.3.rs-1503527/v1.

[241] S. Wang, D. Ma, and M. Wu, "A Quick Search Dynamic Vector-Evaluated Particle Swarm Optimization Algorithm Based on Fitness Distance," *Mathematics*, vol. 10, no. 9, 2022, ISSN: 22277390. DOI: 10.3390/math10091587.

[242] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant Colony Optimization with Local Search for Dynamic Traveling Salesman Problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1743–1756, 2017, ISSN: 21682267. DOI: 10.1109/TCYB.2016.2556742.

[243] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors," *Applied Soft Computing Journal*, vol. 13, no. 10, pp. 4023–4037, 2013, ISSN: 15684946. DOI: 10.1016/j.asoc.2013.05.022. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2013.05.022.

[244] M. Mavrovouniotis and S. Yang, "Memory-based immigrants for ant colony optimization in changing environments," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6624 LNCS, no. PART 1, pp. 324–333, 2011, ISSN: 03029743. DOI: 10.1007/978-3-642-20525-5_33.

[245] M. Mavrovouniotis and S. Yang, "Ant algorithms with immigrants schemes for the dynamic vehicle routing problem," *Information Sciences*, vol. 294, pp. 456–477, 2015, ISSN: 00200255. DOI: 10.1016/j.ins.2014.10.002. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2014.10.002.

[246] J. Eaton, "Ant Colony Optimisation for Dynamic and Dynamic Multi-objective Railway Rescheduling Problems," *PhD Thesis, De Monfort University*, 2017. [Online]. Available: https://www.dora.dmu.ac.uk/xmlui/handle/2086/14950.

[247] S. Abolhoseini and A. Sadeghi-niaraki, "Dynamic Multi-Objective Navigation in Urban Transportation Network using Ant Colony Optimization," vol. 6, no. 1, 2018.

[248] Z. Zhang and S. Qian, "Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems," *Soft Computing*, vol. 15, no. 7, pp. 1333–1349, 2011. DOI: 10.1007/s00500-010-0674-z.

[249] R. Shang, L. Jiao, M. Gong, and B. Lu, "Clonal selection algorithm for dynamic multiobjective optimization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3801 LNAI, pp. 846–851, 2005, ISSN: 03029743. DOI: 10.1007/11596448_125.

[250] Z. Zhang, M. Liao, and L. Wang, “Immune Optimization Approach for Dynamic Constrained Multi-Objective Multimodal Optimization Problems,” *American Journal of Operations Research*, vol. 02, no. 02, pp. 193–202, 2012, ISSN: 2160-8830. DOI: 10.4236/ajor.2012.22022.

[251] Z. Zhang, S. Qian, and X. Tu, “Dynamic clonal selection algorithm solving constrained multi-objective problems in dynamic environments,” *Proceedings - 2010 6th International Conference on Natural Computation, ICNC 2010*, vol. 6, no. Icnc, pp. 2861–2865, 2010. DOI: 10.1109/ICNC.2010.5584014.

[252] K. Trojanowski and S. T. Wierzchoń, “Immune-based algorithms for dynamic optimization,” *Information Sciences*, vol. 179, no. 10, pp. 1495–1515, 2009, ISSN: 00200255. DOI: 10.1016/j.ins.2008.11.014.

[253] P. Amato and M. Farina, “An ALife-Inspired Evolutionary Algorithm for Dynamic Multiobjective Optimization Problems,” *Soft Computing: Methodologies and Applications*, vol. 125, no. 2005, pp. 113–125, 2006. DOI: 10.1007/3-540-32400-3_9.

[254] S. Yuen, T. H. G. Ezard, and A. J. Sobey, “The effect of epigenetic blocking on dynamic multi-objective optimisation problems,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO ’22, Boston, Massachusetts: Association for Computing Machinery, 2022, pp. 379–382, ISBN: 9781450392686. DOI: 10.1145/3520304.3529022. [Online]. Available: https://doi.org/10.1145/3520304.3529022.

[255] M. Helbig and A. P. Engelbrecht, “Performance measures for dynamic multi-objective optimisation algorithms,” *Information Sciences*, vol. 250, pp. 61–81, 2013, ISSN: 00200255. DOI: 10.1016/j.ins.2013.06.051.

[256] C. Raquel and X. Yao, “Evolutionary Computation for Dynamic Optimization Problems,” vol. 490, pp. 85–86, 2013. DOI: 10.1007/978-3-642-38416-5. [Online]. Available: http://link.springer.com/10.1007/978-3-642-38416-5.

[257] R. P. and Y. X., “Evolutionary Dynamic Optimization: Challenges and Perspectives.,” Y. S. and Y. X., Eds., 2013. [Online]. Available: http://www.martrans.org/documents/2008/rst/dvrp%20psaraftis%2088.pdf.

[258] B. Zheng, “A new dynamic multi-objective optimization evolutionary algorithm,” *Natural Computation, 2007. ICNC 2007. Third International Conference on*, vol. 5, no. January, pp. 565–570, 2007, ISSN: 13494198. DOI: 10.1109/ICNC.2007.91.

[259] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable Test Problems for Evolutionary Multi-objective Optimization,” *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, no. 1990, pp. 105–145, 2005. DOI: 10.1007/1-84628-137-7_6.

[260] M. López-Ibáñez, L. Paquete, and T. Stützle, “Exploratory analysis of stochastic local search algorithms in biobjective optimization,” *Experimental Methods for the Analysis of Optimization Algorithms*, pp. 209–222, 2010. DOI: 10.1007/978-3-642-02538-9_9.

[261] C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, and L. Paquete, “On the computation of the empirical attainment function,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6576 LNCS, pp. 106–120, 2011, ISSN: 03029743. DOI: 10.1007/978-3-642-19893-9_8.

[262] *ACM Trans. Evol. Learn. Optim.*, vol. 2, no. 3, 2022, ISSN: 2688-299X.

[263] W. Mauerer, S. Klessinger, and S. Scherzinger, *Beyond the Badge: Reproducibility Engineering as a Lifetime Skill.* Association for Computing Machinery, 2022, vol. 1, pp. 1–4, ISBN: 9781450393362. DOI: 10.1145/3528231.3528359. arXiv: 2203.05283.

[264] M. Helbig, K. Deb, and A. Engelbrecht, “Key challenges and future directions of dynamic multi-objective optimisation,” in *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, 2016, pp. 1256–1261, ISBN: 9781509006229. DOI: 10.1109/CEC.2016.7743931.

[265] M. Helbig, “Challenges Applying Dynamic Multi-objective Optimisation Algorithms to Real-World Problems,” in *Women in Computational Intelligence. Women in Engineering and Science.* A. Smith, Ed., Springer, 2022, pp. 353–375, ISBN: 9783030790929. DOI: `10.1007/978-3-030-79092-9_16`.

[266] C. Bu, W. Luo, T. Zhu, and L. Yue, “Solving online dynamic time-linkage problems under unreliable prediction,” *Applied Soft Computing Journal*, vol. 56, pp. 702–716, 2017, ISSN: 15684946. DOI: `10.1016/j.asoc.2016.11.005`. [Online]. Available: `http://dx.doi.org/10.1016/j.asoc.2016.11.005`.

[267] L. T. Bui, Z. Michalewicz, E. Parkinson, and M. B. Abello, “Adaptation in dynamic environments: A case study in mission planning,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 190–209, 2012, ISSN: 1089778X. DOI: `10.1109/TEVC.2010.2104156`.

[268] J. Butans, “Addressing Real-Time Control Problems in Complex Environments Using Dynamic Multi-Objective Evolutionary Approaches,” vol. PhD, 2012.

[269] T. T. Nguyen, “Continuous Dynamic Optimisation Using Evolutionary Algorithms,” *PhD Thesis, School of Computer Science, University of Birmingham*, no. October, 2010.

[270] A. Muruganantham, K. C. Tan, and P. Vadakkepat, “Evolutionary Dynamic Multiobjective Optimization Via Kalman Filter Prediction,” *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016. DOI: `10.1109/TCYB.2015.2490738`.

[271] H. Zhang, G. G. Wang, J. Dong, and A. H. Gandomi, “Improved nsga-iii with second-order difference random strategy for dynamic multi-objective optimization,” *Processes*, vol. 9, no. 6, pp. 1–23, 2021, ISSN: 22279717. DOI: `10.3390/pr9060911`.

[272] J. Branke, *Evolutionary optimization in dynamic environments*. 2002, vol. 1, pp. 11–18, ISBN: 978-1-4244-5007-7. DOI: `10.1109/ICCP.2009.5284794`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5284794`.

[273] J. Branke and H. Schmeck, “Designing Evolutionary Algorithms for Dynamic Optimization Problems,” pp. 239–262, 2003. DOI: `10.1007/978-3-642-18965-4_9`.

[274] C. M. Colson, M. H. Nehrir, and S. A. Pourmousavi, “Towards real-time microgrid power management using computational intelligence methods,” *IEEE PES General Meeting, PES 2010*, no. January 2015, 2010. DOI: `10.1109/PES.2010.5588053`.

[275] R. Allmendinger and J. Knowles, “On Handling Ephemeral Resource Constraints in Evolutionary Search,” *Evolutionary Computation (MIT)*, vol. 21, no. 4, pp. 1–34, 2012. DOI: `10.1162/EVCO`. [Online]. Available: `http://www.cs.kent.ac.uk/pubs/2010/2993`.

[276] P. J. Angeline, “Tracking extrema in dynamic environments,” in *Evolutionary Programming VI*, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 335–345, ISBN: 978-3-540-68518-0.

[277] T. Back, “On the behavior of evolutionary algorithms in dynamic environments,” in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 446–451. DOI: `10.1109/ICEC.1998.699839`.

[278] K. Weicker, “An analysis of dynamic severity and population size,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1917, pp. 159–168, 2000, ISSN: 16113349. DOI: `10.1007/3-540-45356-3_16`.

[279] Q. Chen, J. Ding, S. Yang, and T. Chai, “A Novel Evolutionary Algorithm for Dynamic Constrained Multiobjective Optimization Problems,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 792–806, 2020, ISSN: 19410026. DOI: `10.1109/TEVC.2019.2958075`.

[280] M. Helbig, “Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation,” *Congress on Evolutionary Computation (CEC)*, no. May, 2012.

[281] S. B. Gee, “Evolutionary Multi-Objective Optimization in Static and Dynamic Environments Thesis,” *PhD Thesis, NATIONAL UNIVERSITY OF SINGAPORE*, 2016.

[282] L. Cao, L. Xu, E. D. Goodman, and H. Li, "Decomposition-based evolutionary dynamic multiobjective optimization using a difference model," *Applied Soft Computing Journal*, vol. 76, pp. 473–490, 2019, ISSN: 15684946. DOI: `10.1016/j.asoc.2018.12.031`. [Online]. Available: `https://doi.org/10.1016/j.asoc.2018.12.031`.

[283] X. Fan, K. Li, and K. C. Tan, "Surrogate Assisted Evolutionary Algorithm Based on Transfer Learning for Dynamic Expensive Multi-Objective Optimisation Problems," *2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings*, 2020. DOI: `10.1109/CEC48606.2020.9185522`.

[284] Y. Guo, H. Yang, M. Chen, D. Gong, and S. Cheng, "Grid-based dynamic robust multi-objective brain storm optimization algorithm," *Soft Computing*, vol. 24, no. 10, pp. 7395–7415, 2020, ISSN: 14337479. DOI: `10.1007/s00500-019-04365-w`. [Online]. Available: `https://doi.org/10.1007/s00500-019-04365-w`.

[285] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *TIK-Report 103 May*, pp. 95–100, 2001. DOI: `10.1.1.28.7571`.

[286] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007, ISSN: 1089778X. DOI: `10.1109/TEVC.2007.892759`.

[287] F. Zou, G. G. Yen, and C. Zhao, "Dynamic multiobjective optimization driven by inverse reinforcement learning," *Information Sciences*, vol. 575, pp. 468–484, 2021, ISSN: 00200255. DOI: `10.1016/j.ins.2021.06.054`. [Online]. Available: `https://doi.org/10.1016/j.ins.2021.06.054`.

[288] F. Zou, G. G. Yen, L. Tang, and C. Wang, "A reinforcement learning approach for dynamic multi-objective optimization," *Information Sciences*, vol. 546, pp. 815–834, 2021, ISSN: 00200255. DOI: `10.1016/j.ins.2020.08.101`. [Online]. Available: `https://doi.org/10.1016/j.ins.2020.08.101`.

[289] X. Peng, D. Xu, and X. Gao, "Evolutionary algorithms for the multiple unmanned aerial combat vehicles anti-ground attack problem in dynamic environments," in May 2013, vol. 490, pp. 403–431, ISBN: 978-3-642-38415-8. DOI: `10.1007/978-3-642-38416-5_16`.

[290] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014. DOI: `10.1109/TEVC.2013.2281535`.

[291] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Ph.D. dissertation, 1999. DOI: `10.1007/s00059-002-2420-5`.

[292] D. Herring, M. Kirley, and X. Yao, "Investigation of Asynchrony in Dynamic Multi-Objective Optimization," *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, pp. 3165–3172, 2019. DOI: `10.1109/CEC.2019.8790270`.

[293] D. Chen, F. Zou, R. Lu, and X. Wang, "A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 43, pp. 147–165, 2018, ISSN: 22106502. DOI: `10.1016/j.swevo.2018.05.001`. [Online]. Available: `https://doi.org/10.1016/j.swevo.2018.05.001`.

[294] M. Chen, Y. Guo, H. Liu, and C. Wang, "The evolutionary algorithm to find robust pareto-optimal solutions over time," *Mathematical Problems in Engineering*, vol. 2015, 2015, ISSN: 15635147. DOI: `10.1155/2015/814210`.

[295] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017. DOI: `10.1109/MCI.2017.2742868`.

[296] E. Zitzler and S. Künzli, "Indicator-Based Selection in Multiobjective Search," in *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 832–842.

[297] J. Knowles and D. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation," in *Proc Congress on Evolutionary Computation*, 1999.

[298] J. Knowles and D. Corne, "M-paes: A memetic algorithm for multiobjective optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1, 2000, 325–332 vol.1. DOI: 10.1109/CEC.2000.870313.

[299] R. Gupta and S. J. Nanda, "Solving Dynamic Many-objective TSP using NSGA-III equipped with SVR-RBF Kernel Predictor," *2021 IEEE Congress on Evolutionary Computation, CEC 2021 - Proceedings*, pp. 95–102, 2021. DOI: 10.1109/CEC45853.2021.9504966.

[300] Y. Mei, X. Li, and X. Yao, "Improving efficiency of heuristics for the large scale traveling thief problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8886, pp. 631–643, 2014, ISSN: 16113349.

[301] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for Large Traveling Salesman Problems," *INFORMS Journal on Computing*, vol. 15, no. 1, pp. 82–92, 2003, ISSN: 1091-9856. DOI: 10.1287/ijoc.15.1.82.15157.

[302] S. Polyakovskiy and F. Neumann, "The Packing While Traveling Problem," *European Journal of Operational Research*, vol. 258, no. 2, pp. 424–439, 2017, ISSN: 03772217. DOI: 10.1016/j.ejor.2016.09.035. arXiv: arXiv:1512.08831v2.

[303] J. Wu, S. Polyakovskiy, M. Wagner, and F. Neumann, "Evolutionary computation plus dynamic programming for the bi-objective travelling thief problem," pp. 777–784, 2018. DOI: 10.1145/3205455.3205488.

[304] H. Ali, M. Z. Rafique, M. S. Sarfraz, M. S. A. Malik, M. A. Alqahtani, and J. S. Alqurni, "A novel approach for solving travelling thief problem using enhanced simulated annealing," *PeerJ Computer Science*, vol. 7, pp. 1–18, 2021, ISSN: 23765992. DOI: 10.7717/peerj-cs.377.

[305] M. El Yafrani and B. Ahiod, "Efficiently solving the Traveling Thief Problem using hill climbing and simulated annealing," *Information Sciences*, vol. 432, pp. 231–244, 2018, ISSN: 00200255. DOI: 10.1016/j.ins.2017.12.011.

[306] C. Wachter, "Solving The Travelling Thief Problem with an Evolutionary Algorithm," *PhD Thesis, Technische Universität Wien*, vol. 2015, 2015.

[307] S. Polyakovskiy, M. R. Bonyadi, M. Wagner, Z. Michalewicz, and F. Neumann, "A comprehensive benchmark set and heuristics for the traveling thief problem," pp. 477–484, 2014. DOI: 10.1145/2576768.2598249.

[308] J. Blank, K. Deb, and S. Mostaghim, "Solving the Bi-objective Traveling Thief Problem with Multi-objective Evolutionary Algorithms," vol. 1, p. 2577, 2017. DOI: 10.1145/1830761.1830909.

[309] Y. Mei, X. Li, and X. Yao, "On investigation of interdependence between sub-problems of the Travelling Thief Problem," *Soft Computing*, vol. 20, no. 1, pp. 157–172, 2016, ISSN: 14337479. DOI: 10.1007/s00500-014-1487-2.

[310] R. Birkedal, "Design, Implementation and Comparison of Randomized Search Heuristics for the Travelling Thief Problem," *Masters Thesis, Technical University of Denmark*, 2015.

[311] M. Wagner, "Stealing items more efficiently with ants: A swarm intelligence approach to the travelling thief problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9882 LNCS, pp. 273–281, 2016, ISSN: 16113349. DOI: 10.1007/978-3-319-44427-7_25.

[312] M. R. Bonyadi, Z. Michalewicz, M. R. Przybyłek, and A. Wierzbicki, "Socially inspired algorithms for the traveling thief problem," *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, pp. 421–428, 2014. DOI: 10.1145/2576768.2598367.

[313] M. Wagner, M. Lindauer, S. Nallaperuma, F. Hutter, and M. Mısır, “A case study of algorithm selection for the traveling thief problem,” *Journal of Heuristics*, 2017, ISSN: 1381-1231. DOI: 10.1007/s10732-017-9328-y.

[314] M. E. Yafrani, S. Chand, A. Neumann, B. Ahiod, and M. Wagner, “Multi-objectiveness in the single-objective traveling thief problem,” pp. 107–108, 2017. DOI: 10.1145/3067695.3076010.

[315] J. B. Chagas, J. Blank, M. Wagner, M. J. Souza, and K. Deb, “A non-dominated sorting based customized random-key genetic algorithm for the bi-objective traveling thief problem,” *Journal of Heuristics*, vol. 27, no. 3, pp. 267–301, 2021, ISSN: 15729397. DOI: 10.1007/s10732-020-09457-7. arXiv: 2002.04303.

[316] J. B. C. Chagas and J. Blank, “A weighted-sum method for solving the bi-objective traveling thief problem,” *Computers and Operations Research*, vol. 138, 2022. arXiv: 3923190 [arXiv:submit].

[317] D. Herring, M. Kirley, and X. Yao, “Dynamic Multi-objective Optimization of the Travelling Thief Problem,” *2002.02636v1 [cs.NE] 7 Feb 2020*, pp. 1–21, 2020. arXiv: 2002.02636. [Online]. Available: http://arxiv.org/abs/2002.02636.

[318] R. Sachdeva, F. Neumann, and M. Wagner, “The Dynamic Travelling Thief Problem: Benchmarks and Performance of Evolutionary Algorithms,” 2020. arXiv: 2004.12045. [Online]. Available: http://arxiv.org/abs/2004.12045.

[319] M. Yang, Z. Kang, and L. Kang, “A parallel multi-algorithm solver for dynamic multi-objective TSP (DMO-TSP),” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5227 LNAI, pp. 164–173, 2008, ISSN: 03029743. DOI: 10.1007/978-3-540-85984-0_21.

[320] R. Gupta and S. J. Nanda, “Solving time varying many-objective tsp with dynamic θ-nsga-iii algorithm,” *Applied Soft Computing*, vol. 118, p. 108 493, 2022, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2022.108493. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494622000448.

[321] M. Liu, “Development of Benchmarks and Algorithms for Realistic Arc Routing Problems,” 2014. [Online]. Available: http://proxy.mul.missouri.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,url,uid&db=ddu&AN=B57EBB3A26BCF464&site=ehost-live&scope=site.

[322] M. Monroy-Licht, C. A. Amaya, A. Langevin, and L. M. Rousseau, “The rescheduling arc routing problem,” *International Transactions in Operational Research*, vol. 24, no. 6, pp. 1325–1346, 2017, ISSN: 14753995. DOI: 10.1111/itor.12346.

[323] Y. Mei, K. Tang, and X. Yao, “Evolutionary computation for dynamic capacitated arc routing problem,” *Studies in Computational Intelligence*, vol. 490, pp. 377–401, 2013, ISSN: 1860949X. DOI: 10.1007/978-3-642-38416-5_15.

[324] D. Bertsimas and G. Van Ryzin, “A stochastic and Dynamic Vehicle Routing Problem in the euclidean plan,” *Operations Research*, vol. 39, no. 4, pp. 601–615, 1990.

[325] A. Larsen, “The Dynamic Vehicle Routing Problem,” *Kgs. Lyngby, Denmark: Technical University of Denmark (DTU)*, no. IMM-PHD; No. 2000-73, pp. 3–18, 2001, ISSN: 03050548. DOI: 10.1016/j.cor.2004.04.013. [Online]. Available: http://orbit.dtu.dk/fedora/objects/orbit:83345/datastreams/file%7B%5C_%7D5261816/content.

[326] B. Fontem, S. H. Melouk, B. B. Keskin, and N. Bajwa, “A decomposition-based heuristic for stochastic emergency routing problems,” *Expert Systems with Applications*, vol. 59, pp. 47–59, 2016, ISSN: 09574174. DOI: 10.1016/j.eswa.2016.04.002. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2016.04.002.

[327] A. J. Kleywegt and J. D. Papastavrou, “The Dynamic and Stochastic Knapsack Problem with Random Sized Items,” *Operations Research*, vol. 49, no. 1, pp. 26–41, 2003, ISSN: 0030-364X. DOI: 10.1287/opre.49.1.26.11185.

[328] G. Y. Lin, Y. Lu, and D. D. Yao, "The Stochastic Knapsack Revisited: Switch-Over Policies and Dynamic Pricing," *Operations Research*, vol. 56, no. 4, pp. 945–957, 2008, ISSN: 0030-364X. DOI: `10.1287/opre.1080.0555`. [Online]. Available: `http://pubsonline.informs.org/doi/abs/10.1287/opre.1080.0555`.

[329] M. Bartlett, A. Frisch, Y. Hamadi, I. Miguel, S. A. Tarim, and C. Unsworth, "The Temporal Knapsack Problem and Its Solution," *Lecture Notes in Computer Science*, vol. 3524, pp. 34–48, 2005. DOI: `10.1007/11493853_5`.

[330] A. G. Hernández-Díaz, C. A. C. Coello, F. Pérez, R. Caballero, J. Molina, and L. V. Santana-Quintero, "Seeding the initial population of a multi-objective evolutionary algorithm using gradient-based information," *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pp. 1617–1624, 2008. DOI: `10.1109/CEC.2008.4631008`.

[331] N. Padhye, L. Zuo, C. K. Mohan, and P. K. Varshney, "Dynamic and evolutionary multi-objective optimization for sensor selection in sensor networks for target tracking," *IJCCI 2009 - International Joint Conference on Computational Intelligence, Proceedings*, pp. 160–167, 2009. DOI: `10.5220/0002324901600167`.

[332] N. Xiao and M. P. Armstrong, "A specialized island model and its application in multiobjective optimization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2724, pp. 1530–1540, 2003, ISSN: 03029743. DOI: `10.1007/3-540-45110-2_24`.

[333] B. Nasiri, M. R. Meybodi, and M. M. Ebadzadeh, "History-Driven Particle Swarm Optimization in dynamic and uncertain environments," *Neurocomputing*, vol. 172, pp. 356–370, 2016. DOI: `10.1016/j.neucom.2015.05.115`.

[334] A. M. Turky, S. Abdullah, and N. R. Sabar, "A hybrid harmony search algorithm for solving dynamic optimisation problems," *Procedia Computer Science*, vol. 29, pp. 1926–1936, 2014. DOI: `10.1016/j.procs.2014.05.177`. [Online]. Available: `http://dx.doi.org/10.1016/j.procs.2014.05.177`.

[335] S. K. Nseef, S. Abdullah, A. Turky, and G. Kendall, "An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems," *Knowledge-Based Systems*, vol. 104, pp. 14–23, 2016, ISSN: 09507051. DOI: `10.1016/j.knosys.2016.04.005`. [Online]. Available: `http://dx.doi.org/10.1016/j.knosys.2016.04.005`.

[336] H. Cheng and S. Yang, "Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6024 LNCS, no. PART 1, pp. 562–571, 2010, ISSN: 03029743. DOI: `10.1007/978-3-642-12239-2-58`.

[337] J. K. Kordestani, A. E. Ranginkaman, M. R. Meybodi, and P. Novoa-Hernández, "A novel framework for improving multi-population algorithms for dynamic optimization problems: A scheduling approach," *Swarm and Evolutionary Computation*, vol. 44, pp. 788–805, 2019, ISSN: 22106502. DOI: `10.1016/j.swevo.2018.09.002`.

[338] X. Peng, K. Liu, and Y. Jin, "A dynamic optimization approach to the design of cooperative co-evolutionary algorithms," *Knowledge-Based Systems*, vol. 109, pp. 174–186, 2016, ISSN: 09507051. DOI: `10.1016/j.knosys.2016.07.001`. [Online]. Available: `http://dx.doi.org/10.1016/j.knosys.2016.07.001`.

[339] D. Yazdani, T. T. Nguyen, and J. Branke, "Robust Optimization over Time by Learning Problem Space Characteristics," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 143–155, 2019, ISSN: 1089778X. DOI: `10.1109/TEVC.2018.2843566`.

[340] M. Märtens and D. Izzo, "The asynchronous island model and NSGA-II," p. 1173, 2013. DOI: `10.1145/2463372.2463516`.

[341] L. F. Gonzalez, D. S. Lee, K. Srinivas, and K. C. Wong, “Single and multi-objective UAV aerofoil optimisation via hierarchical asynchronous parallel evolutionary algorithm,” *Aeronautical Journal*, vol. 110, no. 1112, pp. 659–672, 2006, ISSN: 00019240. DOI: 10.1017/S0001924000001524.

[342] E. Cantu-paz, “Topologies , Migration Rates , and Multi-Population Parallel Genetic Algorithms,” *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 91–98, 2000.

[343] Z. Skolicki and K. De Jong, “The influence of migration intervals on island models,” *GECCO 2005 - Genetic and Evolutionary Computation Conference*, pp. 1295–1302, 2005. DOI: 10.1145/1068009.1068219.

[344] M. Linder and I. Sekaj, “Parallel genetic algorithms,” *Mendel*, vol. 53, no. 4, pp. 9–15, 2011, ISSN: 18033814. DOI: 10.1145/3400031.

[345] M. Weber, F. Neri, and V. Tirronen, “Distributed differential evolution with explorative-exploitative population families,” *Genetic Programming and Evolvable Machines*, vol. 10, no. 4, pp. 343–371, 2009, ISSN: 13892576. DOI: 10.1007/s10710-009-9089-y.

[346] F. Lardeux and A. Goëffon, “A dynamic island-based genetic algorithms framework,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6457 LNCS, pp. 156–165, 2010, ISSN: 03029743. DOI: 10.1007/978-3-642-17298-4_16.

[347] A. Zhou, L. Kang, and Z. Yan, “Solving dynamic TSP with evolutionary approach in real time,” *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*, vol. 2, pp. 951–957, 2003. DOI: 10.1109/CEC.2003.1299769.

[348] M. Guntsch, M. Middendorf, and H. Schmeck, “An Ant Colony Optimization Approach to Dynamic TSP,” in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 860–867.

[349] M. Guntsch and M. Middendorf, “Pheromone modification strategies for ant algorithms applied to dynamic tsp,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2037, pp. 213–222, 2001, ISSN: 16113349. DOI: 10.1007/3-540-45365-2_22.

[350] C. Li, M. Yang, and L. Kang, “A New Approach to Solving Dynamic Traveling Salesman Problems,” *Simulated Evolution and Learning*, 2006, ISSN: 1343-0130. DOI: 10.20965/jaciii.2000.p0129.

[351] C. J. Eyckelhof and M. Snoek, “Ant Systems for a Dynamic TSP,” pp. 88–99, 2002. DOI: 10.1007/3-540-45724-0_8.

[352] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, “On the Solution of Traveling Salesman Problems,” *Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung, International Congress of Mathematicians*, pp. 645–656. 1998.

[353] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, “An Algorithm for the Traveling Salesman Problem,” *Operations Research*, vol. 11, no. 6, pp. 972–989, 1963.

[354] K. Helsgaun, “Effective implementation of the Lin-Kernighan traveling salesman heuristic,” *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[355] I. M. Oliver, D. J. Smith, and J. R. C. Holland, “A study of permutation crossover operators on the traveling salesman problem,” in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, 1987, pp. 224–230.

[356] L. D. Whitley, T. Starkweather, and D. Fuquay, “Scheduling problems and traveling salesmen: The genetic edge recombination operator,” in *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989, pp. 133–140.

[357] W. Banzhaf, “The Molecular Traveling Salesman,” *Biol. Cybern.*, vol. 64, pp. 7–14, 1990.

[358] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. Springer International Publishing, 2010.

[359] S. P. Wanru Gao and M. Wagner, *Optimisation of Problems with Multiple Interdependent Components*, 5May 2017. [Online]. Available: `https://cs.adelaide.edu.au/~optlog/TTP2017Comp/`.

[360] J. Blank and M. Wagner, *EMO2019 - Ranking — Thief 1.0.0 Documentation*, 14 Aug 2019. [Online]. Available: `https://www.egr.msu.edu/coinlab/blankjul/emo19-thief/`.

[361] J. Blank and M. Wagner, *GECCO2019 - Bi-objective Traveling Thief Competition Documentation*, 14 Aug 2019. [Online]. Available: `https://www.egr.msu.edu/coinlab/blankjul/gecco19-thief/%7B%5C#%7Dblank-2017-sbt-3088676-3088680`.

[362] L. Rachmawati and D. Srinivasan, "Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey," *2006 IEEE International Conference on Evolutionary Computation*, pp. 962–968, 2006, ISSN: 01692070. DOI: `10.1109/CEC.2006.1688414`. arXiv: `arXiv:1011.1669v3`. [Online]. Available: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1688414%5Cnhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1688414`.

[363] L. Li, I. Yevseyeva, V. Basto-Fernandes, H. Trautmann, N. Jing, and M. Emmerich, "An Ontology of Preference-Based Multiobjective Metaheuristics," no. Longmei Li, 2016. arXiv: `1609.08082`. [Online]. Available: `http://arxiv.org/abs/1609.08082`.

[364] K. Taylor and X. Li, "Interactive multiobjective optimisation: Preference changes and algorithm responsiveness," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '18, Kyoto, Japan: Association for Computing Machinery, 2018, pp. 761–768, ISBN: 9781450356183. DOI: `10.1145/3205455.3205624`. [Online]. Available: `https://doi.org/10.1145/3205455.3205624`.

[365] E. Popovici, A. Bucci, R. P. Wiegand, and E. D. De Jong, "Coevolutionary principles," in *Handbook of Natural Computing*, G. Rozenberg, T. Bäck, and J. N. Kok, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 987–1033, ISBN: 978-3-540-92910-9. DOI: `10.1007/978-3-540-92910-9_31`. [Online]. Available: `https://doi.org/10.1007/978-3-540-92910-9_31`.

[366] A. Sinha, P. Malo, and K. Deb, "Towards understanding bilevel multi-objective optimization with deterministic lower level decisions," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9018, pp. 426–443, 2015, ISSN: 16113349. DOI: `10.1007/978-3-319-15934-8_29`.

[367] A. Sinha, P. Malo, and K. Deb, *Evolutionary bilevel optimization: An introduction and recent advances*. 2017, vol. 20, pp. 71–103, ISBN: 9783319429786. DOI: `10.1007/978-3-319-42978-6_3`.

[368] G. Eichfelder, "Multiobjective bilevel optimization," *Mathematical Programming*, vol. 123, no. 2, pp. 419–449, 2010, ISSN: 00255610. DOI: `10.1007/s10107-008-0259-0`.

[369] H. Richter and F. Dietel, "Solving dynamic constrained optimization problems with asynchronous change pattern," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6624 LNCS, 2011, pp. 334–343, ISBN: 9783642205248. DOI: `10.1007/978-3-642-20525-5_34`.

[370] E. Alba and J. M. Troya, "Analyzing synchronous and asynchronous parallel distributed genetic algorithms," *Future Generation Computer Systems*, vol. 17, no. 4, pp. 451–465, 2001, ISSN: 0167739X. DOI: `10.1016/S0167-739X(99)00129-6`.

[371] T. Harada and K. Takadama, "Performance comparison of parallel asynchronous multi-objective evolutionary algorithm with different asynchrony," *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings*, pp. 1215–1222, 2017. DOI: `10.1109/CEC.2017.7969444`.

[372] B. Akay, "Synchronous and asynchronous Pareto-based multi-objective Artificial Bee Colony algorithms," *Journal of Global Optimization*, vol. 57, no. 2, pp. 415–445, 2013, ISSN: 09255001. DOI: `10.1007/s10898-012-9993-1`.

[373] S. Kandanaarachchi, M. A. Muñoz, and K. Smith-Miles, “Instance space analysis for unsupervised outlier detection,” *CEUR Workshop Proceedings*, vol. 2436, 2019, ISSN: 16130073.

[374] H. Alsouly, M. Kirley, and M. A. Munoz, “An Instance Space Analysis of Constrained Multi-Objective Optimization Problems,” *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022, ISSN: 1089-778X. DOI: 10.1109/tevc.2022.3208595. arXiv: 2203.00868.

[375] K. Smith-Miles and S. Bowly, “Generating new test instances by evolving in instance space,” *Computers and Operations Research*, vol. 63, pp. 102–113, 2015, ISSN: 03050548. DOI: 10.1016/j.cor.2015.04.022. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2015.04.022.

[376] K. A. Smith-Miles, “Towards insightful algorithm selection for optimisation using meta-learning concepts,” *Proceedings of the International Joint Conference on Neural Networks*, pp. 4118–4124, 2008. DOI: 10.1109/IJCNN.2008.4634391.

[377] K. A. Smith-Miles, “Cross-disciplinary perspectives on meta-learning for algorithm selection,” *ACM Computing Surveys*, vol. 41, no. 1, pp. 1–25, 2008, ISSN: 03600300. DOI: 10.1145/1456650.1456656.

[378] K. Smith-Miles, D. Baatar, B. Wreford, and R. Lewis, “Towards objective measures of algorithm performance across instance space,” *Computers and Operations Research*, vol. 45, pp. 12–24, 2014, ISSN: 03050548. DOI: 10.1016/j.cor.2013.11.015. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2013.11.015.