Distributed Data-intensive Web Service Composition

by

Soheila Sadeghiram

A thesis submitted to the Victoria University of Wellington in fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science.

Victoria University of Wellington 2021

Abstract

Service-oriented architecture (SOA) encourages the creation of modular applications involving *Web services* as the reusable components. *Data-intensive Web services* have emerged to manipulate and deal with the massive data emerged from technological advances and their various applications. Distributed *Data-intensive Web Service Composition (DWSC)* is a core of *SOA*, which includes the selection of data-intensive Web services from diverse locations on the network and composes them to accomplish a complicated task. As a fundamental challenge for service developers, service compositions must fulfil functional requirements and optimise *Quality of Service (QoS)*, simultaneously. The QoS of a distributed DWSC is not only impacted by the QoS of component services and how the compositions are generated, but also by the locations of services and data transformation between services. However, existing works often neglect the impact of locations and data on service composition. The distributed DWSC has not been sufficiently studied in the literature.

In this thesis, we first define the single-objective distributed DWSC that includes communication (e.g. bandwidth), Web service (execution time) and data (data cost) attributes. To this aim, we consider bandwidth information of communication links obtained using the location information of services. Based on the problem formulation, we then address the distributed DWSC problem by developing EC-based approaches. Those EC-based approaches are designed to incorporate domain-knowledge for effectively solving the distributed DWSC problem.

Afterwards, we study the multi-objective distributed DWSC to satisfy different QoS requirements. In particular, the QoS-constrained distributed

DWSC problem and user preferences are considered. For finding trade-off solutions for those problems, new *Multi-objective Evolutionary Algorithms* (*MOEAs*) are proposed based on the current *Non-dominated Sorting Genetic Algorithm-II* (*NSGA-II*).

Furthermore, a new problem formulation for the dynamic distributed DWSC ($D^2 - DWSC$) problem with bandwidth fluctuations is proposed. An EC-based approach is developed to solve the D²-DWSC.

Finally, extensive empirical evaluations are conducted that demonstrate the high performance of our proposed methods in finding composite services with good QoS.

List of Publications

- Soheila Sadeghiram, Hui Ma, and Gang Chen, "Priority-based Selection of Individuals in Memetic Algorithms for Distributed Dataintensive Web Service Compositions", IEEE Transactions on Services Computing (2021)(DOI 10.1109/TSC.2021.3066322).
- Soheila Sadeghiram, Hui Ma, and Gang Chen, "Cluster-guided Genetic Algorithm for distributed data-intensive Web service composition," IEEE Congress on Evolutionary Computation (2018), pp. 1–7.
- 3. Soheila Sadeghiram, Hui Ma, and Gang Chen, "Composing distributed data-intensive Web services using distance-guided memetic algorithm',' International Conf. on Database and Expert Systems Applications, Springer (2019), pp. 411–422.
- Soheila Sadeghiram, Hui Ma, and Gang Chen, "A memetic algorithm with distance-guided crossover: distributed data-intensive Web service composition',' the Genetic and Evolutionary Computation Conference Companion, ACM (2019), pp. 155-156.
- Soheila Sadeghiram, Hui Ma, and Gang Chen, "Composing distributed data-intensive Web services using a flexible memetic algorithm", IEEE Congress on Evolutionary Computation, IEEE, (2019), pp. 2832–2839.
- 6. **Soheila Sadeghiram**, Hui Ma, and Gang Chen, "QoS-constrained multi-objective distributed data-intensive web service composition-

NSGA-II with repair method", the Genetic and Evolutionary Computation Conference Companion, ACM (2020), pp.105-106.

- Soheila Sadeghiram, Hui Ma, and Gang Chen, "A Novel Repairbased Multi-Objective Algorithm for QoS-Constrained Distributed Data-Intensive Web Service Composition", International Conference on Web Information Systems, Springer (2020), pp.489-502.
- Soheila Sadeghiram, Hui Ma, and Gang Chen, "A User-Preference Driven Lexicographic Approach for Multi-Objective Distributed Web Service Composition", Symposium on computational intelligence, IEEE (2020), pp.791-797.
- 9. Soheila Sadeghiram, Hui Ma, and Gang Chen, "A Distance-based Genetic Algorithm for Robust Data-intensive Web Service Composition in Dynamic Bandwidth Environment", Service Computing Conference, IEEE (2020), pp.248-255.
- 10. **Soheila Sadeghiram**, Hui Ma, and Gang Chen, "A novel linkdominance driven approach to distributed multi-objective Web service composition", ACM Transaction on the Web (under second round review).

Acknowledgments

I would like to thank my supervisors, A/Prof. Hui Ma and Dr. Aaron Chen, for their insightful comments and suggestions, as well as continuous support. They have been more than generous with their expertise and precious time. A special thanks to Prof. Mengjie Zhang, for leading the ECRG research group while providing the nurturing environment, and to Dr. Yi Mei at the ECCO subgroup, at Victoria University of Wellington. I am also grateful to my former supervisors Prof. Mohsen Ebrahimi Moghaddam and Prof. Mohammad-Shahram Moin. I would like to acknowledge all my friends and colleagues with a special thanks to Dr. Alexandre Sawczuk da Silva and Dr. Chen Wang.

I cannot begin to express my gratitude to my whole family, Mehrdad and Mehrab Azizabadi Hamedi and Roqayye Sadeghiram, who have never left my side. A special feeling of love to my son, Mehrab, the youngest but the biggest inspiration and cheer. I love you to the moon and back. Last but not least, I am dedicating this thesis to two beloved people who have meant and continue to mean so much to me. Although they are no longer of this world, their memories continue to regulate my life; to the memory of my mother, who always believed in my ability to be successful in the academic arena, and to my father – to whom I promised to dedicate this dissertation before he left this world. They are gone but their beliefs in me have made this journey possible. I would go as far as to say I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. vi

Contents

Glossary

xvii

1	Intr	roduction		
	1.1	Proble	em Statement	1
	1.2	Motiv	ations	6
	1.3	Resear	rch Goals	12
	1.4	Major	Contributions	16
	1.5	Organ	isation of Thesis	20
2	Bacl	kgroun	d and Literature Review	23
	2.1	Backg	round	23
		2.1.1	Web services	23
		2.1.2	Data-intensive Web Services	25
		2.1.3	Web Service Composition	26
		2.1.4	Distributed Data-Intensive Web Service Composition	29
		2.1.5	Motivating Example	32
		2.1.6	An Overview of EC Techniques	33
	2.2	Litera	ture Review	41
		2.2.1	Approaches for Static WSC	41
		2.2.2	Approaches for Dynamic WSC	49
	2.3	Repre	sentation and the Decoding Process	52
		2.3.1	Decoding Sequences into Workflows	54
	2.4	Chapt	er Summary	57

3	Dist	ribute	d DWSC Problem	59
	3.1	Introd	luction	59
	3.2	Chapt	er Organisation	60
	3.3	Proble	em Model	60
		3.3.1	Distribution Model	62
		3.3.2	Service and Data Model	64
		3.3.3	Objective Function	65
	3.4	Bench	mark Dataset	67
		3.4.1	Geographical Distance of Services	68
		3.4.2	Communication Bandwidth	69
	3.5	Chapt	ter Summary	70
4	Sing	gle-Obj	jective Distributed DWSC	71
	4.1	Introd	luction	71
	4.2	Chapt	er Organisation	72
	4.3 Cluster-Guided Memetic Algorithm		er-Guided Memetic Algorithm	73
		4.3.1	Evolutionary Operators for Cluster-guided MA	75
		4.3.2	Clustering of Web services	76
	4.4	Distar	nce-Guided Memetic Algorithm	77
		4.4.1	Crossover Operators for Distributed DWSC	79
	4.5	Flexib	le Local Search for DWSC	82
		4.5.1	Neighbourhood Creating Strategies	83
		4.5.2	Flexible Local Search Process	86
	4.6	Evalu	ations	86
		4.6.1	Parameter Settings and Datasets	87
		4.6.2	Evaluation of Cluster-guided and Distance-guided	
			MAs	88
		4.6.3	Discussions on Cluster-guided and Distance-guided	
			MAs	88
		4.6.4	Evaluation of Flexible Distance-guided MA	91
	4.7	Chapt	er Summary	95

viii

CONTENTS

5	Ada	aptive Single-objective Distributed DWSC 97			
	5.1	Introduction			
	5.2	Chapter Organisation			
	5.3	Overv	view of Adaptive MA	99	
	5.4	Priori	ty-based Selection method for Local Search	100	
		5.4.1	Novelty of Individual Solutions	101	
		5.4.2	Improvability of Individual Solutions	103	
		5.4.3	Selection Criteria for Local Search	105	
	5.5	Evalu	ations	106	
		5.5.1	Competing Methods	106	
		5.5.2	Parameter Settings	107	
		5.5.3	Evaluation of the Priority-based Selection Method		
			(RQ1)	108	
		5.5.4	Evaluation of Different Selection Criteria (RQ2) \ldots	109	
		5.5.5	Evaluation of Response Time and Cost (RQ3)	114	
		5.5.6	Evaluation of Computation Time (RQ4)	116	
	5.6	Chapt	ter Summary	116	
6	Mul	ti-Obj	ective Distributed DWSC	119	
	6.1	Introd	luction	119	
	6.2	Chapt	ter Organisation	121	
	6.3	Multi	-objective Distributed DWSC	122	
		6.3.1	QoS-Constrained DWSC Problem	123	
	6.4	Link I	Pareto Local Search (LPLS) for Distributed DWSC	125	
		6.4.1	Selection of Individuals and the Acceptance Criterion	126	
	6.5	Repai	r-based MOEA for QoS-Constrained DWSC	128	
		6.5.1	Selection Rules	129	
		6.5.2	Power Repair Technique	130	
	6.6	DWS	C with Lexicographic user Preferences	133	
		6.6.1	Preference-driven algorithm: Lex-NSGA-II	133	
	6.7	Evalu	ations	136	

ix

		6.7.1	Performance Metrics
		6.7.2	Evaluation of LPLS-NSGA-II
		6.7.3	Evaluation of DWSCRepair
		6.7.4	Evaluation of Lex-NSGA-II for DWSC-Lex-UR 145
	6.8	Chap	ter Summary
7	Dyr	namic I	Distributed DWSC 151
	7.1	Introc	luction
	7.2	Chap	ter Organisation
	7.3	Dyna	mic Distributed DWSC
	7.4	Dista	nce-based MA for fully-automated D^2 -DWSC 156
		7.4.1	Estimating Robustness
		7.4.2	Distance-based Algorithm
		7.4.3	Inter-service distances and the mutation operator 158
	7.5	Evalu	ations
		7.5.1	Competing Algorithms
		7.5.2	Parameter Settings and Datasets
		7.5.3	Results
	7.6	Chap	ter Summary
8	Con	clusio	ns 167
	8.1	Achie	ved Objectives
	8.2	Main	Conclusions
		8.2.1	Problem Formulations
		8.2.2	Genetic Operators
		8.2.3	Heuristics and Domain Knowledge
		8.2.4	Priority-based Selection Measure
		8.2.5	Hybridised Techniques
		8.2.6	Practical Application of Dynamic approach for Pre-
			diction
	8.3	Futur	e Work

8.3.1	Miscellaneous Dynamic Distributed DWSC Problem		
	Model and Methods		
8.3.2	Learning Techniques		
8.3.3	Miscellaneous QoS Attributes		
8.3.4	Miscellaneous Constraints		

CONTENTS

xii

List of Figures

1.1	Thesis objectives and sub-objectives	12
2.1	Functional properties of a Web service	24
2.2	Sequence construct and QoS calculation	27
2.3	Parallel construct and QoS calculation	28
2.4	Choice construct and QoS calculation	28
2.5	Loop construct and QoS calculation	29
2.6	Example of a solution to a WSC task [175]	30
2.7	Overall Architecture of the distributed DWSC	32
2.8	Example of Web service composition solution	33
2.9	Overall structure of literature review	41
2.10	Decoding process	56
3.1	Communication attributes of a composite service	63
3.2	A component service and its attributes	65
4.1	Steps of the cluster-guided GA	73
4.2	Mutation operator	75
4.3	Index crossover for WSC [54]	76
4.4	Local search operator [54]	77
4.5	Distance-guided single-point crossover	81
4.6	Distance-guided double-point crossover	82
4.7	Distance guided LCS grossover	82
	Distance-guided LC5 clossover	05

List of Tables

4.1	Fitness values of Cluster-guided and Distance-guided MAs . 89
4.2	Fitness values of MAs with flexible local search techniques . 92
4.3	Mean execution time of flexible local search
4.4	Average improvements made by each local search strategy . 93
5.1	Common parameter settings for adaptive MAs 108
5.2	Fitness values of adaptive MAs and the abstraction refinement109
5.3	Adaptive distance-guided MA with different selection pa-
	rameters
5.4	Adaptive cluster-guided MA with different selection pa-
	rameters
5.5	Cost and response time of adaptive MAs 115
5.6	Computation time of adaptive MAs
6.1	IGD values of LPLS-NSGA-II, PLS-NSGA-II and Relaxed-
	LPLS-NSGA-II
6.2	HV values of LPLS-NSGA-II, PLS-NSGA-II and Relaxed-
	LPLS-NSGA-II
6.3	Mean execution time of LPLS, PLS and relaxed PLS methods 140
6.4	IGD values of QoS-constrained methods
6.5	HV values of QoS-constrained methods
6.6	Mean execution time of QoS-constrained methods 144
6.7	Mean IGD for preference $\mathcal{P} : (cost, time)$

68	Mean HV for preference $\mathcal{D} \cdot (cost \ time)$ 147
6.0	Mapp execution time for preference $\mathcal{D} : (acet time)$ 148
6.10	We all execution time for preference P : (cost, time)
0.10	IGD values of preference-driven methods for \mathcal{P} . (<i>time</i> , <i>cost</i>). 149
0.11	HV values of preference-driven methods for P : (<i>time</i> , cost) 149
7.1	Fitness values of dynamic methods
7.2	Response time of dynamic methods
7.3	Execution cost of dynamic methods

xvi

Glossary

- **Abstract workflow** A structure showing the connections between abstract services. Abstract workflows are used to pre-define the overall structure of a composition before optimisation, which consists of selecting the best possible concrete service to fulfil each abstract service, takes place.
- **Abstract service** Abstract service A slot to be filled in by a concrete Web service that specifies the desired functionality with regards to available inputs and expected outputs.
- **Communication link** In the context of DWSC, a link between two directly connected services in a composite service represented as an edge in the corresponding workflow..
- **Communication time** In the context of distributed DWSC, the time required to send/receive data over communication links.
- **Constructs** Structures that organise services within the composition and establish how they interact. For example, the sequential construct structures services into a chain that is executed on service at a time, whereas the parallel construct structures services to be executed independently and simultaneously.
- **Cost** In the context of QoS, this indicates the financial cost associated with invoking a given service.

- **Crossover** A genetic operator that consists of two parent candidates exchanging chosen sub-parts in order to generate some offspring.
- **Data provision cost** The cost applied by the data provider, or the cost to provide the data..
- **Data-intensive WSC (DWSC)** Selecting and composing data-intensive Web services from a variety of location to generate a Web service with complex functionality. The combination of atomic dataintensive Web services that perform simpler tasks into a workflow that accomplishes a more complex task..
- **Data-intensive Web service** Web services that perform analysis on large data and also receive and produce great volumes of data.
- **Decoding** The process of translating an individual in the population (i.e. the genotype) into a corresponding solution (i.e. the phenotype that is the expression of that genotype). In some approaches presented in this thesis, sequences of services are decoded into their corresponding service workflows.
- **Differential Evolution (DE)** An EC algorithm aiming to generate a temporary individual based on individual differences within populations and then randomly restructuring population evolutionary. Variables are in continuous domains that, instead of implementing traditional crossover and mutation operators, DE applies a linear combination of several randomly selected candidate solutions to produce a new solution.
- **Dominance** A given solution *a* is said to dominate another solution b if two criteria are met. Firstly, for all objectives the values of a are at better or equal to those of *b*. Secondly, for at least one objective the value of a is better than that of the b. If these criteria are not met, the solutions are said to be non-dominated with regards to each other.

Glossary

- **Elitism** A genetic operator that consists of preserving the fittest individuals from one generation to the next. The process of calculating an individual's fitness according to one or more functions.
- **Evolutionary computation (EC)** A set of optimisation techniques that employ Darwinian evolutionary principles to produce solutions. A population of candidate solutions is initialised and these individuals are bred over a given number of generations, each time evaluating the fitness of each candidate. Fitter candidates have a higher likelihood of surviving and breeding, which encourages the overall improvement of the population's fitness.
- **Fitness** The goodness of a given individual, typically measured using an objective function. The fitter the individual, the more likely it is to survive and breed during the evolutionary process. The measure of fitness should encourage desirable traits in an individual according to the specific domain being tackled.
- **Fully automated** If a Web service composition approach is fullyautomated, it not only optimises the overall quality of compositions but also explores different workflow configurations during the process. .
- **Genetic operator** A breeding procedure used to generate offspring for a given set of parents.
- **Genetic algorithm (GA)** An evolutionary computation approach that consists of evolving vector-like candidates in order to find the one with the best possible fitness. Genetic operators are designed with the vector representation in mind.
- **Hierarchical clustering** A method of cluster analysis which seeks to build a hierarchy of clusters. Initially, all data is considered as a single cluster, in each iteration, data points are separated from the cluster which

is not similar. Each data point which is separated is considered as an individual cluster, generating n clusters, in the end..

- **Hypervolume (HV)** A performance metric for multi-objective optimisation. It measures the total volume between a given front and a chosen reference point. The larger the hypervolume, the wider the range of solution trade-offs contained in the front.
- **Initialisation** The creation of a new population for evolution. Typically, candidates can be either initialised at random or using a problem-specific method.
- **Invalid solution** In the context of constrained optimisation, solutions that breach the constraints.
- **Inverted generational distance (IGD)** A performance metric for multiobjective optimisation. It measures the distance between points in a given Pareto front and points in the ideal (i.e. best known) Pareto front. The smaller the IGD, the better the given front is at approximating the ideal solutions.
- **K-means clustering** A method to partition observations into K clusters in which each observation belongs to the cluster with the nearest mean.
- **Lexicographic ordering** A method for modelling a priority-based ranking.
- **Local search** A genetic operator that consists of creating a neighbourhood of individuals for a given candidate. Neighbours are evaluated and the fittest one replaces the original candidate in the population, provided it is also fitter than the original.
- **Mutation** A genetic operator that consists of generating offspring by performing a small modification to the structure of an individual.

Glossary

- **Neighbourhood** A set of individuals that are created based on a given candidate. Each neighbour is obtained by performing a small modification on that candidate.
- **Non-dominated sorting genetic algorithm II (NSGA-II)** A multiobjective version of GA whose principle is to independently optimise conflicting objectives. The key idea is to sort candidates by considering the dominance relationships between them.
- **Novelty Search** An evolutionary algorithm without objectives that rewards the novelty of behaviour in the population.
- **Objective function** A function that either minimises or maximises a particular numerical value related to the optimisation problem at hand.
- **Ontology** In the context of Web service composition, an ontology shows the relationships between output and input concepts. This information is used to determine whether the inputs of one service can be fulfilled by the outputs of another.
- **Pareto front** A subset of candidates from a given population that are nondominated with regards to two or more objectives.
- **Particle swarm optimisation (PSO)** An optimisation method inspired by the social behaviour of animals. Particles independently explore the search space, communicating with each other to identify promising areas for further investigation.
- **Penalty** In the context of constrained evolutionary optimisation, is a method to deal with invalid solutions by penalising their fitness value and lowering their chance to be selected for the next generation.

- **Propagation delay** In the context of distributed DWSC, is the amount of time required for the head of data to travel from the source server to the destination server.
- **Quality of service (QoS)** A set of non-functional attributes (e.g. time, cost and throughput) that indicate the expected quality of a given service. Reliability In the context of QoS, this indicates the probability that a response returned by a service is reliable..
- **Repair** In the context of constrained evolutionary optimisation, a technique to rebuild invalid solutions.
- **Repository** A repository contains information about a set of Web services, more specifically a description of each service's location and available operations..
- **Representation** The way in which a solution is modelled for optimisation. For example, a Web service composition solution may be modelled as a tree, as a directed acyclic graph, or as a sequence.
- **Semi-automated** If a Web service composition approach is semiautomated, it assumes that an abstract workflow defining the structure of the composition has been provided, and the goal is to select concrete services that fulfil the corresponding abstract services to achieve the best possible overall quality.
- Service provision cost In the context of QoS, the price charged to use service S_i and is usually specified by service providers..
- **Time** In the context of QoS, this indicates the overall time required for a service to return a response once it receives a request.
- **Transfer time** In the context of distributed DWSC, the time required by a service to put all data on the communication link and depends on the size of data and the bandwidth.

xxii

- **T-test** A statistical test that compares two samples of values with the objective of ascertaining whether the means of the two groups present a statistically significant difference. The t-test assumes that the values analysed follow a normal distribution.
- **Task** A composition request specifying the overall inputs that are initially available for executing the composition and the overall outputs that the composition is expected to produce.
- **Web service** A functionality module that provides operations and/or data and is accessible over a network by using standard communication protocols.
- **Wilcoxon rank-sum test** A statistical test that compares two samples of values with the objective of ascertaining whether the means of the two groups present a statistically significant difference. This test does not assume that the values analysed follow a normal distribution.

xxiv

Glossary

Chapter 1

Introduction

1.1 Problem Statement

Over recent years, the design and development of software systems have been undergoing extraordinary changes [155] driven by the fast advancement of the Web. Web services have been created to support the automated use of Web applications and the evolution of the Web. They provide highlevel abstractions for organising applications in large-scale environments [91]. This is because of essential features Web services hold, such as being loosely coupled, reusable, and configurable [6]. Service-oriented computing (SOC) [141] is a major paradigm which provides a set of standards and concepts for Web services, such as Web Services Description Language (WSDL), Business Process Execution Language (BPEL), and Simple Object Access Protocol (SOAP) [203]. SOC is a concept that assembles software applications to create a network of services in order to achieve rapid, lowcost, and cross-organisational distributed business process. A suitable design is provided by Service Oriented Architecture (SOA) [27] to facilitate efficient and effective use of Web services. As building blocks of SOA, Web services are widely supported by service providers such as Amazon Web Services (AWS) [215].

Existing services often cannot satisfy the diversified needs of users [91].

Web Service Composition (WSC) is an application of SOA, which creates new services via integrating existing services to accomplish advanced functionalities [62]. Individual Web services have functional attributes specified by their inputs and outputs. They require a set of inputs and produce a set of outputs after the execution. Therefore, the functional correctness of a composite service should be satisfied when combining different Web services.

While many Web services deliver the same functionality, nonfunctional *Quality of Service (QoS)* attributes, such as response time and cost (including communication time and cost), become discriminating factors. QoS requirements must be considered explicitly for an effective composition and are clearly defined in the *Service Level Agreement (SLA)*. SLA can be described as a declarative contract between users and service providers [117].

In the following, we first discuss the *distributed Data-intensive Web Service Composition* problem. Afterwards, we briefly discuss multi-objective WSC, dynamic network environment, semi- and fully-automated WSC, and existing algorithms to solve the WSC problem, respectively.

Distributed Data-intensive WSC (DWSC) focuses on the composition of *data-intensive services*. Since the available data on the Web is doubling every 18 months [72], applications and technologies must be designed to scale and meet the growing requirements. Companies can employ dataintensive Web services to perform large-scale data analysis [83, 173]. These services also generate a high volume of new data. Massive volumes of data need to be transmitted between two directly connected services (i.e. services to be executed in sequence within a composite service [173]). For example, *gen expression analysis* (see Subsection 2.1.4 for details), which has been researched for many years, employs data-intensive Web services to analyse billions of data for identification, prediction and clinical decision support [107, 130].

1.1. PROBLEM STATEMENT

Distribution of Services over the network and DWSC are two intertwined concepts; DWSC composes data-intensive Web services distributed over the network. Therefore, it requires to send/receive large amounts of data over a large computer network such as the Internet. Important features such as bandwidth, cost and propagation delay of communication, and location of services must be taken into account. The network bandwidth capacity is the bottleneck in distributed environments, especially when the volume of communication is high [33]. In particular, all of those factors increase the complexity of Web service composition problems. However, existing DWSC approaches [199, 220, 221] have mostly considered a centralised environment for Web services without addressing the network delay. Instead, we consider the distributed DWSC which is a more realistic scenario.

Multi-Objective WSC is an optimisation problem where instead of a single final solution, a set of optimal solutions, also called a *Pareto* front, is anticipated. The reason to find multiple solutions is that our problem has multiple conflicting objectives.

Dynamically changing network environment may lead to QoS fluctuations and SLA violations of any composite service. Network attributes, in particular bandwidth, are subject to dynamic changes, disrupting the expected time and cost required for executing a composite service. *Dynamic WSC* concerns about the problem of delivering composite services with reliable QoS in a dynamic environment. The problem is more challenging than the static WSC due to some major sources of uncertainties, in particular the amount of available bandwidth for the composition. For example, a composite service that is expected to complete execution in a few seconds can take hours due to the drastic change in the bandwidth. The bandwidth change can cause certain services to become unreachable by their previous services [25, 89, 103, 137]. In the following, we will discuss two categories of the WSC problem model each requiring their specific approaches.

Semi- and Fully-automated WSC are two main problem models identified in the literature for WSC approaches. *Semi-automated* WSC approaches [2, 70, 90, 124, 177, 181, 194, 200] assume that a workflow of *abstract* services has already been defined by an expert. The workflow can be represented as a *Directed Acyclic Graph (DAG)* with some *abstract services* as nodes. The abstract services specify the expected functionality in terms of available inputs and required outputs. Composition algorithms are used to find actual services (i.e., *concrete services*) from the service repository to fulfil each abstract service in the DAG. Therefore, this process focuses only on service selection.

The workflow design and service selection are closely related to each other and therefore should be jointly optimised to construct effective solutions. However, with the increasing number of services available in the service repository, it is impractical to manually design compositions' workflow. This idea gives rise to the *fully-automated* WSC where the optimisation and the construction of workflows are performed simultaneously [45, 128, 151, 157, 158, 159, 205]. Different workflow structures can be automatically generated during the composition and services can be selected to achieve optimal QoS. Finding optimal solutions for fully-automated WSC is NP-hard [74, 127, 213]. Fully-automated WSC simultaneously creates workflow and optimises QoS, which increase the complexity of fullyautomated approaches in finding composite solutions with appropriate QoS attributes. Therefore, deciding which services are the most appropriate to be included in a composite service optimise QoS is a significant challenge for distributed DWSC since both the location of services and QoS have an important impact on the QoS of the composite solution. In the following, we will first present solution methods for semi-automated WSC and then discuss those methods for the fully-automated WSC.

1.1. PROBLEM STATEMENT

Approaches for semi-automated WSC can be categorised in three different groups. The first group includes exact optimisation methods, such as Dijkstra [121] and mixed linear programming solvers [76]. Exact optimisation methods are able, at least in theory or when the searching space is small, to provide an optimal feasible solution. However, these methods are mainly restricted to small and special problems [120]. Therefore, it is not always possible or appropriate to use exact optimisation methods to solve WSC [58].

Issues of exact optimisation methods have promoted the development of the second class of solution methods. These methods, in which the processing speed is as important as the quality of the solution obtained, are called *heuristics* or *approximation algorithms* [120]. Approximation algorithms are efficient procedures in an attempt to find good solutions even though there is no guarantee that solutions are optimal. They are designed to find "good enough" and "quick" solutions [135]. Heuristic methods form the foundation of *meta-heuristic* procedures which will be described in the next paragraph.

The third group of solution methods are called *meta-heuristics*. Metaheuristics are iterative generation processes which guide a subordinate heuristic and intelligently combine different concepts for exploring and exploiting the search spaces [138]. Therefore, heuristics are different from meta-heuristics since they are developed to solve a specific problem without the possibility of generalisation or application to other similar problems [120].

Approaches for fully-automated WSC have been mainly considered in two research categories in the literature. The first research category focuses on artificial intelligence (AI) planning techniques that employ a plan-making process [136, 146]. For example, given a service composition request (consisting of provided inputs and required outputs), a composite service can be a planning process, with the inputs as the initial state and the outputs as the desired goal state. The component services as actions triggered by one state and resulted in another state. However, this technique falls under the category of exact optimisation methods that are effective only on small to medium-sized problem instances, and their time complexity grows exponentially as the problem size increases.

The second research category employs Evolutionary Computation (EC) techniques [22, 30, 46, 51, 54, 88, 129, 158, 187, 198]. EC techniques are stochastic *population-based* search meta-heuristic algorithms. They iterate over a set of solutions to fulfil some fundamental bio-inspired principles, for example, genetic inheritance and Darwin's theory for natural evolution [123] in *Genetic Algorithms (GA)* [87] and *Genetic Programming (GP)* [100], and moving flocks of birds in *Particle Swarm Optimisation (PSO)* [96].

Different representations of solutions have been investigated since they could significantly affect the performance of EC-based approaches [54]. Direct representations, such as the tree- and graph-based representations [30, 187], represent explicitly the execution workflow of composite services through displaying actual execution flows of composite services. By contrast, indirect approaches [52, 54] often represent composite services as permutations of services, which require a *decoding* process to build up actual execution workflows.

In summary, we will develop EC-based algorithms to effectively address the single-objective, multi-objective and dynamic distributed DWSC problems. We will also define a model for the problem of distribution DWSC. The motivations related to these contexts are described in Subsection 1.2.

1.2 Motivations

Motivations of this research lie in four key areas that simultaneously account for: 1.single-objective distributed data-intensive Web service composition; 2. adaptive single-objective distributed data-intensive Web service composition;

1.2. MOTIVATIONS

3. multi-objective distributed data-intensive Web service composition, and 4 dynamic distributed data-intensive Web services composition. Each area will be discussed in more details below.

Single-objective distributed data-intensive Web service composition

In a distributed environment of DWSC, data and services are spread out over several servers in a network. The distributed DWSC is a challenging combinatorial optimisation problem since transferring data between Web services takes a significant amount of time [33].

Any small changes in the selection of services in DWSC can cause a big difference in the solution's quality. For example, services with lower cost might be in a very distant location to other services in the composite service which degrades the quality of the composite service. Therefore, to effectively solve the distributed DWSC problem, we not only need a suitable objective function but also require methods which put extensive efforts on employing domain-knowledge. These methods can be assisted by effective initialisation and evolutionary operators to focus on exploring promising solution sub-spaces.

We will design EC-based approaches combined with domainknowledge. Additionally, we will hybridise EC techniques with local search techniques, which result in Memetic Algorithms (MAs) (see Chapter 2). For example, an MA hybridised with a clustering technique in Chapter 4 to produce an effective initial population, and special distancebased local search and crossover operators will be proposed.

Adaptive Single-Objective Distributed Data-Intensive Web Service Composition

MAs can further enhance the effectiveness of EC algorithms [46, 52, 160, 222], since they can effectively exploit the neighbourhood area of a current solution. The local search operator focuses on the exploitation and the EC

algorithm explores a wide area of the solution space. However, evaluating the fitness of any new solution obtained through a local search in MAs incurs a non-negligible cost and time. Accordingly, local searches cannot be applied to all individuals in the population. This leads to serious challenges in locating suitable individuals for local searches. Existing MAs ignore the importance of deliberately selecting individuals and its impact on the effectiveness of the search process [64, 101, 140, 211, 214, 216]. Most approaches only concern the fitness of solutions by performing conventional fitness-based tournament selection [64, 101, 140] (see Chapter 2). The winner is the solution with a better fitness value. Fitness-based selection methods are expected to result in faster convergence but can easily be blocked by poor local optima [106]. They do not encourage solutions from different regions of the search space to be improved by local search. Selecting individuals of similar structures may reduce the diversity of the population and therefore increase the difficulty of escaping from local optima. Therefore, MAs may bias succeeding generations towards a particular region leading to pre-mature convergence [106]. On the other hand, the random selection of individual solutions can cause a waste of local search resources on hard-to-improve solutions.

However, some individuals have a higher potential to be improved through local search operator. For instance, a solution involving higher communication cost and time can be substantially scaled up by replacing services that cause high inter-service communication costs and times by more efficient alternative services. Additionally, the effectiveness of the local search depends not only on the fitness of the chosen solution but also on whether it can contribute to the evolution of the whole population. For example, selecting an individual with a different structure (e.g., communication links with different cost distributions) from the best solution in the population can help to preserve the population diversity and, therefore, help to avoid local optima. Therefore, it is highly desirable to develop a generalised selection method that selects high-priority solutions for the

1.2. MOTIVATIONS

local search by using fitness, population and solution-related measures.

Multi-Objective Distributed Data-Intensive Web Service Composition

DWSC usually demands optimising conflicting objectives such as the execution time and cost. It can be modelled as a multi-objective optimisation problem to capture varied trade-offs among different conflicting QoS attributes. Single-objective approaches assume that users' QoS preferences can be quantified in advance [5, 109]. In multi-objective algorithms can simultaneously search for a set of compromised solutions [160, 202, 210], for when to choose from in the absence of preferences (e.g., the weight of each QoS attribute).

Multi-objective DWSC problems are generally considered harder than their single-objective versions. Multi-objective DWSC requires complex exploration mechanisms to identify a set of solutions representing the possible compromises of the underlying conflicting objectives. Finding every composition solution in the set is impractical. Therefore, a typical goal for the multi-objective DWSC is to develop efficient and effective approaches to search for a high-quality *approximation* of the set using EC algorithms.

Multi-objective evolutionary algorithms (MOEAs) (refer to Chapter 2 for details) were originally designed for solving unconstrained multi-objective optimisation problems; however, in real-world applications problems may require satisfying QoS constraints.

QoS requirements are usually specified by users as constraints [34, 63]. For example, a deadline and a budget can be set on the overall QoS by a service requester [34, 63]. This has led to another popular research theme around multi-constrained service composition where a constraint is specified on each QoS attribute and the objective is to satisfy all the QoS constraints in the best possible way [15, 30]. However, the direct use of MOEAs may not guarantee the satisfaction of any QoS constraints since the goal to search for valid solutions may pose a negative impact on the optimisation of QoS [34, 85, 133].

Existing approach on multi-objective constrained WSC [84] penalise the fitness value of invalid solutions (solutions that violate the constraints). However, repairing invalid solutions can help the search maintain more solutions in the feasible region. Solutions that violate any QoS constraints may be repaired paving the way for discovering highquality solutions in future generations. Therefore, effective repair operators must be designed for QoS-constrained WSC, which can take advantage of heuristics and domain knowledge to repair invalid solutions and to increase the diversity of the solutions in the population.

Furthermore, sometimes users may consider one QoS attribute as being more important than others (e.g., cost *vs* response time). Users' QoS preferences are crucial parts of a service composition task since they determine whether a composite service is more preferable than others. This gives rise to a group of problems known as *WSC with user-preferences (WSC-UR)* [15, 30, 84, 179]. To solve this kind of problems, the optimisation should be guided in the direction of users' preferences.

Methods to define user preferences are desired to reduce interactions with the user during the evolutionary search. Firstly, this thesis proposes a method to easily formulate user preferences. For example, if there is some basic knowledge (e.g., the preference order) about the QoS attributes, it can be utilised to guide the search for the most preferred regions in the solution space for distributed DWSC. Secondly, it proposes to practically model user preferences through the *lexicographic ordering*, without the involvement of users during the search process. Lexicographic ordering is a method for modelling a priority-based ranking [95]. For example, in a flight itinerary service with two QoS attributes of duration and cost, if a user is interested in the fastest flights, the duration (flying time) will be important than the budget. However, if the user prefers the cheapest flight, the cost of the flight will be of the highest importance. Such a relative preference helps to define the feasible area of solutions in the objective space. Therefore, preferences can be defined by a lexical ordering, eliminating
the user interactions during the search (i.e., the user only defines the order or the priority of objectives at the beginning). Consequently, it is desirable to propose an effective and efficient multi-objective DWSC method to provide users with enhanced solutions according to their preferences.

Dynamic Distributed Data-Intensive Web Service Composition

Many existing approaches to DWSC make the underlying assumption that communication networks are static and bandwidth seldom changes. However, in a distributed environment, bandwidth often changes with time and can affect the overall quality of the composition.

Research works have been conducted to solve dynamic WSC problem [42, 73, 104, 148, 168, 177, 195, 200]. None of the existing approaches has considered fluctuations of the network bandwidth, which is the main source of QoS changes [33]. Meanwhile, [42] and [177] assume that all services are located locally.

Based on the preceding discussions, designing an effective approach to handle the dynamic network environment for our problem is an open area. Further, it is crucial to study the real-world distributed DWSC, which is dynamic. New algorithms must be developed to address aspects of the problem such as the problem formulation that includes bandwidth changes, and the influence of service location on the QoS. For example, it is desirable to define a new optimisation objective to find composite services that are *robust* to network changes. A robust solution can prevent the QoS degradation during the execution and can be re-optimised easily [191]. Additionally, it is crucial to employ problem knowledge in designing the EC operators that can address the dynamic changes.



Figure 1.1: Thesis objectives and sub-objectives

1.3 Research Goals

The overall goal of this thesis is to develop *effective EC-based methods for distributed DWSC*. More specifically, the research will focus on: (1) developing single-objective EC-based approaches to solve distributed DWSC problem; (2) developing multi-objective EC-based approaches to multi-objective distributed DWSC problems; and (3) developing EC-based approaches to distributed DWSC problem in consideration of dynamic network bandwidth. Our research aims to develop hybrid EC-based methods (e.g., combined with local search) which are enhanced by heuristics. The research goal described above will be accomplished by pursuing the following set of objectives, which are also outlined in Figure 1.1.

1. To develop EC-based approaches to single-objective DWSC.

We first propose a proper QoS model (i.e., cost and time model) for distributed DWSC considering communication delay and cost, and the local QoS of each service and dataset, to improve end-to-end QoS requirements expected by the user. We employ location of services to encompass data transmission between them. Further, new benchmark datasets are set up by defining a bandwidth simulation model from a real-world dataset and obtaining location information of services are using the real-world WS-Dream dataset [226]. Afterwards, effective EC-based methods are proposed that include important domain knowledge, such as the location of services.

- (a) To develop an effective EC-based approach for distributed DWSC. Domain knowledge is necessary for improving algorithm effectiveness to find an optimised and feasible solution. The novelty of this approach (i.e., *cluster-guided MA*) is in employing distance information to produce a high-quality initial population. Services are clustered based on the geographical location and the initial population is generated within each cluster.
- (b) To develop effective crossover operators for EC-based approaches for distributed DWSC. This sub-objective is to develop crossover operators, guided by domain-knowledge, to effectively converge to global optima. The is to generate improved offspring composite services using heuristics such the location information and the *longest common sub-sequence* (see Chapter 4 for details). The longest common sub-sequence heuristic eliminates long communication links (i.e., the longest link between two adjacent services in a composite service).
- (c) To propose EC-based approaches hybridised with a flexible distanceguided local search for distributed DWSC. To achieve this goal, a new local search is designed. The local search is then combined with the EC algorithm to generate the *flexible distanceguided memetic algorithm (MA)*. Solutions are improved during the local search considering the geographical distance between their component services, to eliminate long communication links from the composite solution. Longer communication

links are replaced with shorter ones to improve the QoS.

(d) To develop a priority-based mechanism to select individuals for local search in distributed DWSC. This sub-objective is to investigate the effect of selecting individuals for local search in MAs on the solutions. Some individuals have a higher potential to be improved through local search. To this aim, a new measure called *SelectionPriority* will be developed to define a generalised selection method which selects high-priority solutions for local search by using the fitness of the solution and the populationrelated measures such as the novelty level of a solution compared to other solutions and the solution's improvability. Additionally, the cluster-guided and distance-guided MAs are combined and then scaled up with the proposed selection method.

2. To develop EC-based approaches to multi-objective DWSC.

To fulfil this objective, we first study the simultaneous optimisation of two different objectives for multi-objective distributed DWSC. The emphasis is on improving the exploration ability of the multiobjective algorithm and the use of location information in the distributed DWSC. To this aim, we develop a new MOEA based on NSGA-II. Subsequently, two different multi-objective problems are studied: QoS-constrained multi-objective DWSC and multi-objective DWSC with user preferences.

(a) To develop an effective MOEA-based approach for distributed DWSC. Link Pareto Local Search (LPLS) is designed through extending a state-of-the-art local search technique for MOEA (i.e., Paretobased local search (PLS)). LPLS improves the MOEA for DWSC by relaxing the acceptance of neighbour solutions considering the communication link attributes. A novel multi-objective MA is designed with the help of the improved PLS and a popular MOEA, i.e NSGA-II [61].

1.3. RESEARCH GOALS

- (b) To develop MOEA-based multi-objective DWSC approach with respect to QoS constraints. We aim to develop an effective repair technique to handle QoS constraints for multi-objective distributed DWSC. Our proposed technique allows *invalid solutions* to be repaired to obey user-defined QoS constraints and participate during the search to improve the convergence rate.
- (c) To develop MOEA-based multi-objective WSC approach with user preferences. Although MOEAs are targeted to find good compromises of final solutions, users often have imprecise preferences about what kind of solutions might be presented. This objective is to integrate user preferences in the MOEA using the lexicographic ordering. By utilising preferences during the optimisation process, we develop new mechanisms to both improve the quality of the final set of solutions according to user preferences and enhance efficiency through eliminating unwanted solutions.

3. To develop EC-based techniques to dynamic distributed DWSC problems.

Objectives 1 and 2 are proposed for solving WSC problems in static composition settings. However, many optimisation problems arise in a dynamic environment that demands the service composition to adaptively cope with the changes. Bandwidth changes can frequently happen in distributed service environments [33] and disrupt the expected QoS of the composite solution to maintain certain thresholds, such as costs and response time. With this research objective, we focus on the bandwidth fluctuation during the design stage through designing robust composite services. These composite services are expected to handle bandwidth changes in a robust manner at the execution stage. (a) To develop an EC-based algorithm for distributed DWSC in environments with dynamically changing bandwidth. We formulate a new problem, fully-automated dynamic distributed DWSC (D²-DWSC) with a key focus on bandwidth changes. We then design an EC-based algorithm to find composite services that are robust to network bandwidth changes. We also employ distance of services as the problem knowledge to design a distance-based mutation operator to improve the effectiveness of our algorithm.

1.4 Major Contributions

Major contributions of this thesis are as follows:

1. This thesis proposes a model for distributed DWSC which considers communication attributes, such as bandwidth, communication time and cost. The following contributions are sought in this thesis: 1) an effective problem-specific MA has been designed which uses the proposed model. The GA is hybridised with a clustering technique which employs location of services to produce high-quality initial population. 2) A new MA has been developed armed with carefully designed crossover operators to build high-quality composite services. Two major points have been verified: the importance of utilising domain knowledge in the MA through crossover operators, and the effectiveness of preserving promising sub-solutions among existing composite services to pass valuable information through generations by crossover operators. 3) Two new local search strategies have been developed to perform a flexible local search operator for the distributed DWSC problem considering location information of Web services. Through combining the flexible local search and GA, we develop a new MA (i.e., distance-guided MA) for the distributed DWSC, which also employs one of the above-mentioned crossover

operators, 4) A novel priority-based selection method for MAs is proposed that generalises the most widely used fitness-driven selection methods. In particular, this selection method is widely applicable to many memetic algorithms (MAs). A new measure (i.e., *SelectionPriority*) which includes *novelty* and *improvability* of a current solution, to evaluate the priority of solutions to be selected is proposed. The proposed method significantly outperforms existing state-of-the-art methods for DWSC problems. This contribution has been published in:

- (a) S. Sadeghiram, H. Ma, and G. Chen, "Cluster-guided Genetic Algorithm for distributed data-intensive Web service composition," Congress on Evolutionary Computation IEEE (2018), pp. 1–7.
- (b) S. Sadeghiram, H. Ma, and G. Chen, "Composing distributed data-intensive Web services using a flexible memetic algorithm," Congress on Evolutionary Computation IEEE (2019), pp. 2832–2839.
- (c) S. Sadeghiram, H. Ma, and G. Chen, "Composing distributed data-intensive Web services using distance-guided memetic algorithm," International Conf. on Database and Expert Systems Applications, Springer (2019), pp. 411–422.
- (d) S. Sadeghiram, H. Ma, and G. Chen, "A memetic algorithm with distance-guided crossover: distributed data-intensive Web service composition," the Genetic and Evolutionary Computation Conference Companion, ACM (2019), pp. 155-156.
- (e) S. Sadeghiram, H. Ma, and G. Chen, "Priority-based Selection of Individuals in Memetic Algorithms for Distributed Dataintensive Web Service Compositions," IEEE Transactions on Service Computing (under third round review after minor revision).

2. This thesis proposes MAs for multi-objective distributed DWSC in the following contributions: 1) a *link-dominance* driven local search method (LPLS) is developed to properly balance exploration and exploitation of our proposed algorithm. Our experiments show that the current local search technique for MOEAs, i.e., PLS (see Chapter 2), focuses mainly on exploiting existing solutions. However, exploration is of great importance for multi-objective WSC and is discouraged by PLS when applied to our problem. Therefore, we design the LPLS to address this problem. We subsequently propose a novel MOEA which hybridises NSGA-II with the LPLS. Through this hybridisation, we achieve the flexibility of exploration along with the exploitation during the local search while preventing inferior solutions.

2) This thesis studies QoS-constrained multi-objective distributed DWSC. A knowledge-based repair method is proposed which can be integrated with any effective multi-objective approach for the constrained fully-automated DWSC. Therefore, we combine the repair method with NSGA-II algorithm enhanced by a set of selection rules. We performed an extensive empirical analysis of the proposed knowledge-based NSGA-II and a comparison with an existing repair-based method as well as a cutting-edge constrained handling method ([84, 179]).

3) This thesis proposes an effective approach to the multi-objective distributed DWSC problem with user QoS priority preferences. We have defined the problem as a multi-objective optimisation problem with *lexicographic* preferences, *DWSC-LexUR*, where different priorities are assigned to different objectives. Afterwards, we propose an algorithm, called *Lex-NSGA-II*, to effectively solve the DWSC-LexUR problem. We verify that our method can find high-quality solutions with the help of a clustering technique, where both diversity and user preference satisfaction in Lex-NSGA-II are ensured. The fol-

lowing publications have been derived from this contribution:

- (a) S. Sadeghiram, H. Ma, and G. Chen, "A novel link-dominance driven approach to distributed multi-objective Web service composition," ACM Transaction on the Web (under second round review).
- (b) S. Sadeghiram, H. Ma, and G. Chen, "QoS-constrained multiobjective distributed data-intensive web service composition-NSGA-II with repair method," the Genetic and Evolutionary Computation Conference Companion, ACM (2020), pp.105-106.
- (c) S. Sadeghiram, H. Ma, and G. Chen, " A Novel Repair-based Multi-Objective Algorithm for QoS-Constrained Distributed Data-Intensive Web Service Composition", International Conference on Web Information Systems, Springer (2020), pp.489-502.
- (d) S. Sadeghiram, H. Ma, and G. Chen, "A User-Preference Driven Lexicographic Approach for Multi-Objective Distributed Web Service Composition", symposium on computational intelligence, IEEE (2020), pp.791-797.
- 3. This thesis proposes a new fully-automated approach, *Distance-R-MA*, to the fully-automated distributed dynamic DWSC problem (i.e., D²-DWSC). A problem model is defined in the context of dynamic bandwidth changes. We evaluate Distance-R-GA with samples based on the real-world network bandwidth changes to test the robustness of composition solutions. Our evaluation results confirm that Distance-R-MA can generate solutions that are robust to the dynamic network environment (i.e., bandwidth fluctuations). They can find composition solutions with lower execution cost and response time than solutions generated by existing algorithms (e.g., for static problems). Further, our experimental results show that inter-service

distance can significantly affect a solution's performance, but can be carefully reduced by our proposed distance-based mutation operator. This contribution has been published in:

(a) S. Sadeghiram, H. Ma, and G. Chen, "A Distance-based Genetic Algorithm for Robust Data-intensive Web Service Composition in Dynamic Bandwidth Environment", IEEE Service Computing Conference (2020), pp.248-255.

1.5 Organisation of Thesis

The rest of this thesis is structured as follows:

Chapter 2 introduces some fundamental concepts related to Web services and Web service composition. Subsequently, related works on automated service composition in single-objective, multi-objective, and dynamic contexts are reviewed.

Chapter 3 presents the single-objective DWSC problem model. This model will be used in Chapters 4 and 5. Further, this model will be extended for the multi-objective and the dynamic distributed DWSC in Chapters 6 and 7, respectively.

Chapter 4 introduces three approaches to the single-objective distributed DWSC problem. The first approach is the cluster-guided MA, which combines a GA with the k-mean clustering technique for generating high-quality initial population. Next approach proposes different problem-specific crossover operators and compares their performance in the context of MA. The third approach proposes problem-specific local search operators for the distributed DWSC.

Chapter 5 studies adaptively selecting individuals for local search in MAs for distributed DWSC problem. A priority-based selection technique is proposed to direct the evolutionary search towards regions of the search space in MAs where more interesting compromise solutions can be found.

1.5. ORGANISATION OF THESIS

Some substantial elements of the priority-based selection are studied: solutions that increase diversity, solutions which can lead to better fitness and those that can be easily improved during the local search.

Chapter 6 proposes three approaches to solve different categories of multi-objective distributed DWSC. A link-dominance concept is proposed and used during a local search combined with NSGA-II. The second problem is QoS-constrained multi-objective distributed DWSC, which includes the problem formulation for constraints on two QoS attributes (i.e., response time and cost). A repair technique with a set of heuristics is then proposed to solve the problem. The third problem is the multi-objective distributed DWSC with user preferences. Preferences are modelled using lexicographic ordering and then are incorporated during the search.

Chapter 7 introduces D²-DWSC and formulates it as the optimisation of the robustness measure. A mutation operator is also proposed as part of a GA to improve the quality of robust solutions. The solution to this problem includes the formulating of the problem as the robust novel dynamic Web service composition problem that specifically handles bandwidth fluctuations.

Chapter 8 discusses the objectives achieved in this thesis, the main conclusions reached by the contribution chapters and insights that guide the future work.

CHAPTER 1. INTRODUCTION

Chapter 2

Background and Literature Review

In this chapter, a background including the definition of WSC and DWSC problems, and algorithms are presented in Section 2.1. Afterwards, the literature review and representations for WSC are discussed in Sections 2.2 and 2.3, respectively. Finally, Section 2.3 summarises this chapter.

2.1 Background

This section provides preliminaries on Web service composition, an overview of EC techniques and representation of WSC solutions.

2.1.1 Web services

Web services are reusable software modules available over the Internet [27] that may affect some actions or change in the world. They can be discovered, located, invoked and loosely coupled across the Web to facilitate the integration of newly established applications regardless of the platforms, operation systems and programming languages. In service-oriented environments, Internet protocols, such as the Simple Object Ac-



Figure 2.1: Functional properties of a Web service.

cess Protocol (SOAP) [203], are commonly used for communication between Web services.

Web services' interfaces are described by standard description languages, such as Web Services Description Language (WSDL) [40]. WSDL is an XML-based file that determines the functionality of a Web service, in terms of the inputs and outputs. A client program connecting to a Web service can read the WSDL file to search desired functionalities. For a Web service to be executed a set of inputs should be provided and it produces a set of outputs after the execution. The functional properties of Web services are demonstrated in Figure 2.1.

Quality of Service

Apart from the functional attributes, Web services have non-functional attributes, i.e., *Quality of Service (QoS)* properties [139]. QoS attributes often refer to some quality criteria employed for ranking services [3] when several services provide the same functionality. Today, a large number of Web services offer identical or overlapping functionality, but present various QoS, included but not limited to response time, reliability, reputation, availability and execution cost. Some Web service providers publish QoS information in a *Service Level Agreement (SLA)* [182], which defines the terms and conditions of service quality that a Web service delivers to the service requester. For example, a given SLA may specify that only services with a response time of at most one second should be considered.

Some QoS attributes are as follows:

• Response time, to measure the response time of a service once it has been invoked;

2.1. BACKGROUND

- Cost, to specify the financial cost of using a given service;
- Availability, to measure the likelihood of a service being available at invocation time (this captures crash failures);
- Reliability, which is the likelihood of a service responding appropriately when invoked (this captures arbitrary failures);
- Throughput, to determine the number of service invocations per minute.

Note that some of these QoS attributes should be maximised, (e.g., availability and throughput) while others should be minimised during the composition (e.g., response time and cost). In this thesis, we consider execution cost and response time since they are the most widely used QoS attributes in the literature.

2.1.2 Data-intensive Web Services

The access to large amounts of data has opened up new exciting opportunities mainly in science and computing areas. Therefore, it is crucial to effectively and efficiently process large-scale data. The academic world and industry have started to adopt Web services and *Service Oriented Architecture (SOA)* to manage data on the Internet since service-oriented computing and Web services have achieved success in encapsulation and data integration [33, 114, 119, 218]. However, traditional functional-centric Web services cannot meet the requirements of data processing, which merely provide the functionality required by the user. On the other hand, *Dataintensive Web services* focus on providing and updating data with large amounts of data operation and exchange [218]. They generally require a massive amount of data as their inputs, perform a set of data-intensive analysis and produce a huge amount of output data.

2.1.3 Web Service Composition

Web service composition (WSC) can be considered as a combinatorial optimisation problem (COP). COPs are a group of optimisation problems that consist of finding an optimal object from a finite set of objects [41]. There are a great number and variety of COPs which come up in practice and need to be solved efficiently. Other examples of these problems are minimum spanning tree problem, the linear ordering problem and scheduling problems.

It is difficult for the service user to find and select the best services in terms of QoS parameters. The reason is that with the increasing number of Web services produced by different service providers, many services provide the same functionality but with a different QoS. Additionally, when no single Web service is able to respond to the user's request, it is necessary to compose a range of existing services according to their input/output. *Web Service Composition (WSC)* is an implementation of the (SOA) and creates new services via integrating existing services to accomplish new value-added and complex functionality [62].

On one hand, the compatibility of input(s)/output(s) of component services affects the functional correctness of the composite Web service. On the other hand, QoS attributes have a big impact on the overall QoS of the composite service and the composite service should guarantee users an acceptable level of quality for the whole process. Therefore, the simultaneous optimisation of QoS values and ensuring the functional correctness is the main characteristic of *fully-automated* WSC problems [10, 20, 46, 48, 51, 54, 55, 154, 202]. In this thesis, we focus on the response time and cost from the perspective of users.

As for WSC, there are several basic composition constructs, including the sequential construct, the parallel construct and the conditional construct [30, 48]. The aggregation value of QoS for Web services composition varies with respect to different constructs, which determines how services are associated with each other in a composite service [54]. A formal model



Figure 2.2: Sequence construct and QoS calculation of the distributed DWSC will be presented in Chapter 3.

- Sequence construct: Services are executed sequentially, which means that the outputs of a preceding service are used to fulfil the inputs of the subsequent one. The aggregated time, T, and execution cost, C, are computed as a sum of time and cost of Web services involved and the communication time and cost between them, respectively. This construct is shown in Figure 2.2. S_i is service i and L_i is the i^{th} communication link.
- *Parallel construct*: The parallel construct introduces parallelisation schema into composite services. Web services are executed concurrently. Services are executed in parallel, the inputs of each service are fulfilled independently, and their outputs are produced independently. The aggregated execution cost is calculated in the same way as those for the sequence construct, while the aggregated time, *T*, is determined by the most time-consuming path in the composite flow. This construct is presented in Figure 2.3.
- *Choice construct*: Only one service path is executed in a choice construct depending on the satisfaction of the conditions on each path. This construct is shown in Figure 2.4 and assumes that *n* branches exist in the composite service. *p*₁, ..., *p*_n denote the probability of each branch with ∑ⁿ_{k=1} *p*_k = 1. For example, the aggregated total cost *C* is obtained by summing the multiplication of the total cost of each



Figure 2.3: Parallel construct and QoS calculation



Figure 2.4: Choice construct and QoS calculation

branching (i.e., C_n and T_n) and the corresponding branch possibility, p, over all branches.

• *Loop construct*: Web services composed with a loop construct are executed repeatedly until a certain condition is satisfied. Suppose that the average number of iterations is *l*, and *t* and *c* corresponding aggregated value of the sub-path in the composite service. Therefore, aggregated response time *T* and execution cost *C* are *l*.*t* and *l*.*c*, respectively, The loop construct is illustrated in Figure 2.5.



Figure 2.5: Loop construct and QoS calculation

In this thesis, we focus on two sequential and parallel structures similar as in most fully-automated service composition works [29, 30, 32, 46, 213], where composite services can be represented as DAGs.

A classic example of WSC problem is the travel planning scenario [175], shown in Figure 2.6. In this scenario, the system is designed to automatically reserve hotels and flights according to customer preferences. The customer is required to provide preference types, such as departure date and destination city, which are the composition *input*. Similarly, the reservation outcomes, such as issued tickets and receipts, are the composition *outputs*. Available services for the composition are a set of flight booking services and hotels. This simple composition solution performs flight and hotel reservations according to a customer's information.

2.1.4 Distributed Data-Intensive Web Service Composition

The task of selecting data-intensive services from different locations and composing them to optimise QoS is called distributed *Data-intensive Web Service Composition (DWSC)*. In a real-world network environment, Web services are distributed across various servers and data centres in different locations. Further, they use different datasets from different places and therefore, the output size varies from one service to another.

Distributed DWSC aims to find a service composition with minimal cost and response time, including communication cost and time between



Figure 2.6: Example of a solution to a WSC task [175]

services. Consequently, distributed DWSC has introduced new challenges to the research area since the distance and data transfer between services on a distributed DWSC significantly affect the performance of composite services. Therefore, different from existing WSC problems, the data size and the location of services are of great importance to the distributed DWSC since they substantially influence the communication cost and time.

However, this information has been omitted from the current fullyautomated research approaches [10, 20, 29, 30, 46, 49, 54, 71, 119, 134, 153, 190, 192, 202, 204, 205, 220, 221]. Although those approaches establish various service-oriented workflows, they assume a centralised service environment for the composition, which is unrealistic for WSC and even more unsuitable for distributed DWSC where the location of data and Web services is a fundamental element.

Distributed DWSC not only demands a different objective function but also different methods than the general WSC problem. The selection of services for the distributed DWSC can have a big impact on the quality of composite services as they may involve massive data transmission. Therefore, the difference of distributed DWSC with general Web service composition approach is not only about the objective function and including communication time and cost in the objective function but in the need for designing specific methods.

We propose that when considering the distributed nature and dataintensity in the objective function as well as in the method, the resulted composite services are more effective. The main reason for believing this is that any small changes in the selection of services in distributed DWSC can create a very different solution quality than the main one as a result of the factors of distribution such as communication time and cost. In fact, the difference is not about the search space and the complexity of the objective function, but the way of selecting services for the composition as a result of the distribution and mass data requirement. Therefore, to effectively solve the distributed DWSC problem, we need to design methods using not only a suitable objective function, but also effective initialisation, evolution operations such as selection, crossover and mutation, communication link dominance and, etc.

The overall architecture of distributed DWSC is shown in Figure 2.7. A user sends a service request to the Web service composition agent. This agent is responsible for searching for a suitable Web service composition that meets the request. The Web service composition agent accesses the distributed service repository to retrieve services' functional properties, QoS, location and data information. Web services are created and provided by different service providers and registered in the repository. The Web service composition agent employs proposed distributed DWSC MAs in this thesis to generate a composite service with high performance. The recommended composite service can then be executed by the execution engine and outputs are sent to the user. Note that parallel data processing can be performed internally within a data-intensive Web service using big data architecture such as Hadoop (implementing and executing the composite service is outside of the scope of this thesis).



Figure 2.7: Overall Architecture of the distributed DWSC.

2.1.5 Motivating Example

Data-intensive Web services and SOA are used to build the *microarray experimental system* which is one of the hottest topics in the gene expression field [107, 130]. Gene expression has been researched for many years and related microarray experiments have been conducted all over the world [1], e.g., in National Cancer Institute¹. Consequently, a vast amount of microarray data sets are produced by many institutions. To enable institutions to share the data sets, data-intensive services are created to access and analyse the data. Those services can be reused to created value-added services, i.e., via service composition [83].

The *Gene Expression Analysis Services* (*GEAS*) [80] is an example of a Web service repository which provides several publicly available REST-ful Web services. Example of data-intensive services includes microarray data process services, data normalisation services, deferentially expressed genes identification services, functional analysis services and so on [80].

For a given service request (e.g., gene prediction) existing service, in particular, data-intensive Web services can be composed to perform gene prediction. Gene prediction requires data analysis using multiple Web services. An example is shown in Figure 2.8. However, with the number of

¹https://ccr.cancer.gov/



Figure 2.8: Example of Web service composition solution (data-intensive Web services are shown in red).

available services in the repository, manually finding such a composite service is impractical because the huge number of options of selections and compositions should be explored to find one that best meets both functional and QoS requirement. It is crucial to consider service locations as well as data volume between distributed services. Specifically, some of the Web services exchange big volumes of data over a communication network such as the Internet. In this thesis, we consider this as well as data communication attributes of services into account to find a near-optimal service composition. It can automatically select services from different distributed service repositories and generate near-optimal service compositions simultaneously.

2.1.6 An Overview of EC Techniques

EC techniques simulate ideas from the biological evolution and selection of individuals in a population to solve computational problems. A population of solutions (i.e., individuals) are evolved to produce final solutions.

WSC problem can be explained as a global optimisation problem [7]. Finding an appropriate solution in a large repository of Web services can be very difficult and time-consuming because there will be several possible solutions to choose from in a limited time. Web services must match to each other in terms of their input(s) and output(s) and encourage a highquality composite service in terms of the execution cost and time. EC techniques have shown to be more efficient than traditional methods for problems with large search space and, in particular, for the WSC problem when the task increases in complexity [10]. They search for areas of possible good solutions if their population diversity is enough. Moreover, they carry the ability to use and guide heuristics, which are crucial for efficiently solving combinatorial optimisation problems (COPs) such as distributed DWSC. Further, the promises of EC techniques have been proven in solving combinatorial optimisation problems [12, 156, 180], achieving proper flexibility for encoding many different problems.

In the context of single-objective WSC, EC techniques have become increasingly popular [46, 129, 189, 205]. Examples include *Particle Swarm Optimisation algorithm (PSO)* [96] in [51], *Ant Colony Optimisation (ACO)* [65] in [196] and GAs in [30].

EC techniques are particularly effective for solving the multi-objective WSC problems [30, 37, 45]. Multi-Objective Evolutionary Algorithms (MOEAs) simultaneously optimise multiple objectives with different trade-offs. Examples of MOEAs are *elitist Non-dominated Sorting Genetic Algorithm (NSGA-II)* [61], *Strength Pareto Evolutionary Algorithm 2 (SPEA2)* [229] and *Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D)* [224].

Any EC algorithm to solve a problem must have five basic components [123]: 1) Representation of solutions; 2) A way to create an initial population of solutions; 3) Evaluating solutions; 4) Operators that alter solutions during reproduction and produce offspring solutions; 5) Values for the parameters;

For example, direct or indirect representations can be employed to represent solutions. The initial population can be generated randomly. Evaluation is to determine how good or (bad) a solution is. New offspring solutions can be produced by using a crossover operator on two selected parent individuals in GA. Parameters may include operators probability, population size, number of generations and so on.

2.1. BACKGROUND

Additionally, for a successful implementation of an EC algorithm for a particular real-world problem, basic components listed above might require some additional heuristics [123]. Examples are an output from some deterministic algorithm to create the initial population, heuristics for dealing with infeasible solutions in the fitness function (e.g., in constrained problems), or incorporating problem-knowledge in the representation.

Genetic Algorithms (GAs)

Genetic Algorithm (GA) [68] generates an initial population of solutions, i.e *chromosomes*, usually in the form of vectors. At each generation, a fitness measure is employed to evaluate solutions. Solutions are evolved by operators (i.e., crossover and mutation) to form the next generation. Solutions with the best fitness value are called *elite solutions* and are directly copied to the next generation. Finally, a solution with the best fitness value at the final generation will be returned as the final solution. The structure of a simple GA is presented in Algorithm 1. One popular method to select parent solutions for the operators is the *tournament selection*, where two randomly chosen solutions are compared based on their fitness value. The winner is the solution with better fitness value.

An example of the representation for GA is the binary representation which is frequently used for the well-known 0-1 knapsack problem. Binary representation handles each chromosome as a cord of bits, either 1 or 0. In the 0-1 knapsack problem, each bit in the chromosome is associated with a type of item "A". A zero in the position of item "A" means that it is not included in the knapsack, while a one indicates that the item has been counted.

Crossover and mutation operators can modify the chromosome. An example of the crossover operator on binary chromosomes is the singlepoint crossover operator, where a point on both parents' chromosomes is picked randomly and called a "crossover point". Bits to the right of that crossover point are swapped between the two parent chromosomes. This

Algorithm 1 Simple structure of Genetic Algorithm	
1: Generate initial population of random solutions;	
2: while the number of generations not met do	
3: Evaluate solutions using a fitness measure;	
4: Copy elite solutions to the next generation;	
5: Select parent solutions;	
6: Apply crossover and mutation operators to form the next generation;	
7: end while	
return SolutionWithBestFitness;	

results in two offspring, each carrying some genetic information from both parents. A simple mutation operator is to select one solution as the parent and flip the bit in a random position. Crossover and mutation operator used in this theses are explained in detail in Chapter 4.

Non-dominated Sorting Genetic Algorithm II (NSGA-II)

For multi-objective optimisation problems, a set of Pareto solutions are desired. The achieved solution set is not only the closest possible set to the Pareto front, but also required to be well spread and covering wide areas. Therefore, the best way to solve a multi-objective optimisation problem is by using an EC technique.

The notion of an optimal solution does not apply anymore when addressing a multi-objective problem. A solution x1 in the multi-objective case outperforms another solution x2 if it is better than x2 in at least one objective and not worse for any of the remaining objectives. Definition 1 introduces the *Pareto-dominance* concept. The goal of a Pareto-dominance algorithm is to return all solutions representing different trade-offs.

Definition 1. (*Pareto dominance*). A solution x_1 is said to dominate a solution x_2 ($x_1 \prec x_2$) if and only if

1) for all $k \in \{1.., m\}, f_k(x1) \leq f_k(x2)$ and

2) there exists at least one $k \in \{1.., m\}$, so that $f_k(x1) < f_k(x2)$ where f_k denotes the specific value of objective k.

Herein, we have assumed minimisation objective functions. Therefore, lower values for objective functions are desirable.

36

Definition 2. (Mutually non-dominated solutions). Solutions x1 and x2 are said to be mutually non-dominated (x1||x2) if and only if neither $x1 \prec x2$ nor $x2 \prec x1$, and $x2 \neq x1$.

The above relations will be extended to the distributed DWSC problem in Chapter 6.

The pseudocode of NSGA-II [61] is shown in Algorithm 2. NSGA-II presents a fast non-dominated sorting approach. NSGA-II starts with a random population of solutions (or individuals) ordered by non-dominated sorting approach. In this sorting procedure, NSGA-II applies an iterative process that searches for non-dominated solutions at different levels. First, for each solution x if the number of solutions dominating x is zero, then x belongs to the first front. This process is iterated for the set of solutions dominated by x (i.e., with each member in this set to form the second level and so on). This continues until all solutions are sorted. The population of parents for the next generation is identified by applying a tournament selection strategy to the current population, and then, the better one is selected according to the non-domination front.

Solutions at the same non-domination front are compared using a second measure called *crowding distance (CD)*. CD is a measure of the density of the solutions at the neighbourhood of that solution and is calculated as the average of the Euclidean distance of a solution to other solutions in the objective space. Subsequently, a population of offspring is generated by applying the genetic operators (i.e., crossover and mutation) to the population of parents. The next population is formed by taking the best solutions from the combined population of parents and offspring. The selection criteria are first the non-domination rank and then the crowding distance. The procedure is terminated when a user-defined maximum number of generations is reached.

Simplicity, effectiveness, modularity and the low number of userdefined parameters are the determining reasons for the popularity of NSGA-II among multi-objective optimisation algorithms [59]. For WSC, for instance, NSGA-II has been used to generate a set of solutions covering a diverse range of QoS values [46, 53, 150].

Additionally, a study of various evolutionary multi-objective optimisation algorithms, including NSGA-II, SPEA2 and MOEA/D, has been conducted on the semi-automated multi-objective WSC problem [44]. Different performance comparisons such as the hypervolume (see Chapter 6) indicated that NSGA-II can often outperform SPEA2 and MOEA/D on this problem.

Memetic Algorithms (MAs)

To further enhance the effectiveness of EC algorithms, hybridised EC enhanced with some local search techniques have been developed [46, 52, 160, 222], resulting in Memetic Algorithms (MAs) [132]. Through emphasising the synergy between exploration and exploitation, MAs can effectively exploit the neighbourhood area of a current solution. The local search operator focuses on the exploitation and a population-based algorithm tries to add more diversity by exploring a wide area of the solution space.

Previous studies for WSC problem have indicated that MAs contribute to excellent performance in searching high-quality solutions [46, 54, 92, 131, 211]. Their success is often attributed to the algorithms' ability to properly combine the intensification and diversification. For DWSC, it has been proven that MAs can significantly outperform GA on benchmark datasets for DWSC problems [220].

Three important points should be clarified for every local search: which individuals to apply the local search on (starting points for the local search), how to create a neighbour solution, and what criteria to use to accept/reject neighbour solutions.

Algorithm 2 NSGA-II	
1:	procedure NSGA-II (N)
2:	Randomly generate initial population P^0 with N (population size)
	individual solutions;
3:	Evaluate solutions in P^0 ;
4:	while the maximum number of generations not reached do
5:	Select solutions from P^0 using a tournament selection;
6:	Apply crossover and mutation on selected solutions to form the
	offspring solutions set;
7:	Combine P^0 and offspring solutions as the new population;
8:	Sort the new population into Pareto-fronts (PFS);
9:	Set $P^1 = \{\}$
10:	Set the current front to <i>PFS[0]</i>
11:	while $ P^1 < N$ do
12:	Sort solutions from the current front by Crowding Distance;
13:	for each solution in the ordered front do
14:	$\mathbf{if} P^1 < N$ then
15:	Include the solution in P^1 ;
16:	end if
17:	end for
18:	$P^0 = P^1;$
19:	Go to the next <i>PFS</i> .
20:	end while
21:	end while
22:	end procedure

Pareto Local Search

Pareto Local Search (PLS) [66, 142] is a simple yet effective local search technique dedicated to MOEAs. is based on the Pareto dominance concept. Existing PLS research on multi-objective WSC problems has verified that hybrid NSGA-II enhanced by PLS can significantly outperform NSGA-II [45].

Three key components of PLS, namely selection (which individuals should be selected for applying the local search), neighbourhood exploration (how to create a neighbourhood solution of the selected individual) and acceptance criterion (whether to accept or reject a neighbour solution after it has been evaluated) are crucially important for the performance of PLS [75, 116, 169].

Various versions of PLS have been proposed in the literature [67]. In one of the popular versions, PLS starts with an initial set of all nondominated solutions, called *archive*. It then explores their neighbourhoods. PLS extends iterative improvement procedures from the single-objective case to the multi-objective case by changing the acceptance criterion. In the single-objective case, an iterative improvement algorithm usually accepts a new solution if it is better than the current one according to the fitness value. In the multi-objective algorithm, PLS accepts a new solution only if it is not dominated by any solution in the archive. Further, PLS only accepts solutions that dominate their parent and have a higher crowding distance. A newly accepted solution is added to the archive. This iterative process of PLS stops when no other solution has been left (even the newly added solutions should be considered for the local search.)

The performance of NSGA-II has been investigated with the use of three different local search techniques in [143]: 1) PLS, 2) *Path-Relinking* (*PR*) [78], which starts from a high-quality solution generating a path in the neighbourhood space that leads toward another solution, and 3) Tabu search [77]. It has been shown in [143] that all memetic MOEAs can clearly outperform NSGA-II, and PLS is more effective than PR and Tabu search. However, PLS has a restrictive nature and a strict acceptance criterion.

2.2 Literature Review

In this section, we review existing works on WSC. Subsections 2.2.1 and 2.2.2 review approaches for static and dynamic WSC, respectively. The structure of the literature review is also shown in Figure 2.9.



Figure 2.9: Overall structure of literature review

2.2.1 Approaches for Static WSC

In this subsection, we review existing approaches for static WSC. This subsection discusses those approaches into five groups: non-EC approaches for WSC, EC-based approaches for semi-automated single-objective WSC, EC-based approaches for semi-automated multi-objective WSC, EC-based approaches for fully-automated single-objective WSC, and EC-based approaches for fully-automated multi-objective WSC, respectively.

Non-EC Approaches for Static WSC

In this subsection, we present existing works which are based on other techniques than EC. In particular, we review them in two groups: semiautomated and fully-automated approaches.

Semi-automated WSC has been addressed by non-EC methods. An approach for the semi-automated WSC is proposed in [161] for service composition mobile network. Uniform and proportional distribution of service in the environment have been considered, and then, the temporal distance between services are estimated. The proposed method in [161] finds the shortest path on a pre-defined graph for each request using Dijkstra's Algorithm. Another approach for semi-automated multi-objective WSC is proposed [37] which considers the correlation of QoS values of different services. A candidate pruning algorithm for removing the unpromising candidates regarding their QoS value and to find a set of services with no dependency on others is used. Afterwards, Vector Ordinal Optimisation (VOO) [86] technique is employed to find the compositions. VOO performs a global optimisation and uses Pareto-dominance concepts to compare solutions.

Integer Linear Programming (ILP) is used to address the fully-automated WSC problem [15, 28, 119]. ILP is a mathematical optimisation model with constraints and a linear objective function. For example, [15] addresses the single-objective semi-automated WSC using an ILP model. In that method, where violation in global constraints is permitted by a defined user threshold but with an associated penalty. Several mathematical models are calculated for a constraint (e.g., the aggregated value of that QoS) violation of users preference, and the degree of relaxation in violation depending on the willingness of user willingness. Afterwards, aggregated QoS values are obtained, where constraints are applied in the utility function.

Graph search is an alternative approach to EC-based approaches for solving fully-automated WSC [30, 32, 152]. Graph search works on searching composite services, which are constructed by sub-graphs or paths from a service dependency graph based on service dependencies. However, constructing such a service dependency graph could suffer a scalability issue when dealing with a large service repository with complexity service dependencies. This issue can get even worse when QoS optimisation is considered [97]. For example, a search method is employed to search composite services presented as paths, which are constructed from a sub-graph of a service dependency graph [152]. This sub-graph is extracted based on service requests. However, these works only focus on minimising the number of candidate services in the preprocessing step. Further, the scalability of those methods suffers when the service repository grows. To address this critical issue, [31] proposes a QoS-aware service composition via a scalable way of pruning dependency graphs, and a novel path-based construction and selection method. This method can efficiently construct near-optimal composite services.

A method, called abstraction refinement, is proposed in [29] which requires services to be grouped according to certain functional relationships. Afterwards, a representative is selected for each group, which is the most promising individual service within that group concerning the QoS values. The above research, then, creates composite services using the dependency graphs [206].

To consider multiple quality criteria in QoS, recent works [30, 32] propose an improved path-based search method based on [31]. Particularly, a node (i.e., an atomic service) associated with a higher rank is preferred in a path construction, and nodes are ranked based on the concept of dominance over multiple QoS quality criteria. However, all of the above methods [30, 31, 32, 206] prune services based on their QoS value and they do not have any consideration of the location and distance of services to each other.

GraphPlan [16], a well-known AI planning algorithm, has been widely used for WSC [35, 105, 202, 206]. GraphPlan contains two stages: a forward expand stage that constructs a planning graph, and a backward search stage that retrieves a solution. GraphPlan may be combined with other techniques, such as fuzzy techniques for ranking services [228], and skyline operators for filtering services [110] to reduce the number of services to be considered for the composition. However, skyline might filter available services and fail to find a solution, particularly when the filtered services are located in very close proximity to other services in distributed DWSC.

EC for Semi-automated Single-Objective WSC

A majority of existing EC-based methods aim to solve semi-automated single-objective WSC problem [9, 19, 82, 129, 197, 198, 210, 225]. A WSC with the *privacy-preserving* ability in the cloud environment is proposed in [9], where a model is defined to obtain the privacy-preserving level provided by the composite services. To solve the problem, the authors propose to cluster the composite services according to their fitness values and privacy-preserving levels. Afterwards, GA is hybridised with another EC technique, *Shuffled Frog Leaping Algorithm (SFLA)* [69], for finding near-optimal composite services.

Local and global optimisations are combined to solve the semiautomated single-objective DWSC in [197]. First, a local utility function is calculated to select a concrete service for each abstract service in the workflow. An optimised overall solution is then found among the possible composition paths in the workflow using Ant Colony Optimisation (ACO) algorithm.

Before the composition phase, data placement strategy is used to reduce the datasets transfer cost in [19]. To this aim, a dependency matrix of datasets is created, which grouping two datasets or more which are mostly used together for many services. Afterwards, dependent datasets are sorted and located in the same data centre in order to lower the frequency of datasets transfer in the network and to reduce cost and response time. GA is used to find the final solution.

Since placing data on data centres is not a matter of Web service composition but defined by data providers, this idea can be considered for other problems rather than the composition. They have also classified Web services based on their inputs and outputs to organise the search area.

EC for Semi-automated Multi-Objective WSC

Several methods have approached the semi-automated multi-objective WSC problem [8, 36, 37, 63, 88, 102, 150, 166, 167, 188]. Some of the manipulate existing EC algorithms to achieve higher performance. For example, a time-varying multi-objective PSO algorithm is proposed to address this problem with two objectives [88], service cost and multi-cloud risk view-points. Parameters of the algorithm are tuned based on elapsed time so that a good balance between exploration and exploitation is achieved.

A method proposed in [102] employs NSGA-II and MOEA/D to address a semi-automated multi-objective WSC. The approach considers two types of application workflows: professional and enterprise. Independent optimisations are performed in each generation by applying EC operators. A chromosome is divided into two parts where the first part optimises the professional service composition plan, while the second part accounts for optimising enterprise service composition plan.

A flexible NSGA-II is proposed for the semi-automated WSC problem [34], which employs a relaxed dominance relationship (i.e., ϵ -dominance) and an archive to choose parents of the crossover operator. This work reports a decent performance compared to the standard NSGA-II.

Seeding strategies have been investigated for injecting problemknowledge into the NSGA-II for the semi-automated WSC problem [36]. The seeding strategies are used to strengthen the quality of the initial population for the NSGA-II to start working with. Their results indicated that the seeding strategy improves the performance of NSGA-II. However, the implication of the number of seeds on the service composition problems is minimal.

NSGA-II is combined with a local search and the *Differential Evolution* (*DE*) algorithm [176] to perform an inter-cloud service composition in [167]. EC operators in NSGA-II are performed by DE algorithm which combines two mutation strategies and changes the evolutionary parameters adaptively.

Little attention has been paid to add the constraint handling ability to MOEAs for WSC problems [84, 111, 112, 183]. For example, A DE-based method for the multi-objective semi-automated WSC with QoS constraints is proposed in [63]. The method converts the problem into an unconstrained multi-objective problem through constraint-based service filtering and converting a local utility based on the global constraint.

In [84, 111], multi-objective QoS-constrained WSC has been addressed by employing a *penalty-based* technique, where the fitness of an infeasible solution is degraded by a penalty, depending on the amount of the constraint violation. However, it is difficult to define the penalty function properly and, therefore, these methods are highly sensitive to the choice of the penalty function [43, 164]. On the other hand, optimal solutions of most engineering problems lie on the constraint boundary [172, 209], which means that an invalid solution near the constraint boundary may be converted to a high-quality valid solution with a repair technique. For instance, random repair operators have been proposed in [163]. However, this repair operator is not effective since random changes cannot efficiently and effectively fix solutions.

EC for Fully-automated Single-objective WSC

Fully-automated single-objective WSC composition using EC techniques has been studied in the literature [50, 154, 187, 219, 222].

Some of the approaches for fully-automated WSC have employed the tree-based representation, where all composition structure types are represented as inner nodes of trees and the services as the outer nodes [50, 118, 154, 208, 219, 220]. Genetic Programming (GP) [99] is the first choice to solve the WSC problem when using this representation. While some approaches randomly initialise the tree-based composite services
[219], [154] proposes a context-free grammar for randomly initialising treebased composite services with the correct structure of composite services. The above two approaches filter incorrect solutions in their fitness functions by penalising them. GP has been hybridised with *Tabu search* [77] to overcome the premature and proneness of GP in [220], to solve the DWSC problem. Different constructs are considered in the mentioned method for predefined workflows.

WSC solutions can be directly represented as DAGs, which is done by taking into consideration the dependencies between services in the graph [49, 165, 212]. GraphEvol is introduced in [49] with graph-based genetic operators, which are utilised to evolve individuals represented by DAGs. GraphEvol is compared to the previously discussed GP-based approaches [47, 118], and their experiment shows that graph-based approaches can produce composite services with much higher QoS.

Sequence-based representations (e.g., permutations) have been employed in the literature for the fully-automated WSC [46, 51, 191]. Such permutations need to be interpreted into service composition workflows using a decoding strategy. For example, PSO in [51, 186] and GA in [54] investigate the permutation-based representation to optimise solutions. Additionally, the authors suggested an exhaustive local search operator on permutations for WSC, which produces all neighbours of a solution. Exhaustive local search operators are likely to be decoded into the same composite service. The experiments in [54] indicated that this representation with the combination of the local search and decoding strategy can achieve higher performance compared to GraphEvol [49].

EC for Fully-automated Multi-objective WSC

A limited number of EC-based works are proposed for solving fullyautomated multi-objective WSC problem [30, 45, 53, 193, 221].

Hybridised NSGA-II with coloured *Petri* nets [144] is proposed in [193]. This method considers qualitative and quantitative preferences as well as the global trust of composite services. Consequently, the authors in [193] model the service composition problem as a multi-objective optimisation where each of the considered aspects forms an objective function to be optimised. They employ the concept of relaxation degree to ensure the effective calculation for quantitative preferences.

A multi-objective approach for solving the centralised DWSC is proposed in [221], where a tree-based representation and search space reduction are adopted before optimisation using the concept of fully and partially dependent Web services. However, the size of data and the location of Web services which influence the communication distance between services and causes varied communication cost and communication time have been omitted.

Another approach is proposed in [30] to the multi-objective fullyautomated WSC problem. Similar to [221], this approach employs the search space reduction, where Web services are clustered into nonoverlapping groups based on their input(s) and output(s). Afterwards, the skyline technique is used to find a non-dominated set of services for each cluster, which will be used in the service composition (skyline is an operator frequently used for finding individuals that are not dominated by other individuals [17, 178]). This approach has introduced several new measures for evaluating composite services.

NSGA-II with a fragmented tree-based representation is proposed in [53] to fully-automated multi-objective WSC. However, this fragmented tree-based representation does not show its effectiveness for finding better Pareto solutions in their experiment, comparing to an indirect representation. To improve the performance of NSGA-II, the hybridisation of NSGA-II and other techniques are also explored. For example, the same authors in [45] later have proposed a hybrid approach, which combines NSGA-II and MOEA/D. This combination decomposes the multi-objective problem into single-objective sub-problems, producing a set of Pareto-optimal solutions with a variety of trade-offs. A single-objective local search is then applied to each sub-problem. This work has introduced the local search to MOEA/D. However, a large number of decomposed sub-problems are predefined, and a simple form of local search is not effective to be performed on each sub-problem. Despite achieving some promising results, opportunities still exist to address these limitations.

2.2.2 Approaches for Dynamic WSC

In this subsection, we present recent approaches on dynamic WSC. These approaches fall under the category of prediction-based, backup-based or re-optimisation-based approaches. Most of the existing approaches have been proposed to solve the semi-automated WSC.

An approach based on the prediction of QoS attributes has been proposed in [168, 200] for the semi-automated WSC. This approach adjusts the value of network attributes with the change of context information (such as novel path latency and workloads) during the execution to forecast more accurate network QoS while finding a composite service.

Historical QoS performance retrieved from service logs has been utilised in [42] to solve the dynamic semi-automated WSC. In this approach, services are ranked based on recently observed QoS measures for related services. Another approach using time-series prediction model and GA has been proposed in [177] to handle time-varying QoS of services in a semi-automated WSC.

On the other hand, some research resorts to the use of backup solutions [26, 56, 148, 184]. Backup solutions are identified before the execution of the service composition so that if a problem occurs, these solutions can be employed to maintain the successful execution of services. In these approaches, all backup solutions are determined before the execution without taking into account environment changes which cause the method not being efficient in substantially dynamic and high-dimensional environments. For example, [184] obtains a sufficient number of backup services for each service employed, which are distributed to prevent a task failure in case of network failure.

However, these techniques do not allow any changes to the workflow structure of any composite service. A workflow which works well in one network environment may not be effective when the environment changes. Additionally, these approaches are not efficient in substantially dynamic and high-dimensional environments since they require a great number of backup solutions to cope with all the possible changes.

Some existing approaches handle the dynamic WSC by proposing replacements of a sub-solution during the execution of a composite service [18, 115]. That is, when a composite service fails due to the failure of a single service, these approaches compare the failing service and some other services with possible candidates. For example, meta-heuristics have been used to discover substitutes for multiple failing services [18]. As another example, parts of the composite service which include the failing service are replaced with another subset that satisfies the global QoS values [115].

As another approach to the dynamic WSC problem, re-optimisation has been employed at the service execution stage [81, 133], where the frequency of re-optimisation is updated to handle the fluctuations in the composition environment that are assumed to happen periodically. For example, specific characteristics related to the presence of time-dependent QoS properties are considered [81] for developing proactive use of reoptimisation techniques in response to anticipated future changes, or reoptimisation is applied periodically [133]. Re-optimisation techniques can effectively choose another service from the service repository and replace the flawed service.

Another method discussed in [198] applies the same model as in [197], but with two improvements. First, it addresses dynamic data-intensive service composition to handle the arrival of new services, failure or stop of some services and changing in some QoS attributes. Second, this method discusses strategies for modifying the pheromone information in Ant Colony Optimisation to find a balance between preserving enough old information to speed up the search process and modifying enough new information to help the ants finding a new solution for the changing scenarios and thus prevents a sub-optimal solution.

All of the above approaches are semi-automated and require predefined workflows, rendering them not suitable for the fully-automated DWSC. Further, [42] and [177] have overlooked the distribution of services, and hence the corresponding communication attributes, assuming that all services are located locally. Therefore, the problem formulation, bandwidth simulation model, and the considerations of the influence of inter-service distances on the QoS and efforts to reduce them through a specifically designed mutation operator are all technical novelties of our method over the above methods.

The only approach that deals with the dynamic fully-automated WSC problem is proposed in [191]. This approach consists of two stages: a design stage, where a robust solution is designed; and the execution stage, where a local search technique performs a repair operator whenever a failure happens. The robust solution is designed to run in dynamic QoS environments without major performance degradation. This approach generates QoS failure scenarios at random. Additionally, a robust solution is easier to be re-optimise [191] and it does not rule out the chance for re-optimisation. Compared to other approaches, using robust solutions has the advantages of being efficient (the backup method is not efficient in substantially dynamic and high-dimensional environments).

However, most of the above approaches have been proposed to solve the semi-automated WSC. The only approach for fully-automated dynamic WSC, [191], has been proposed to solve the centralised WSC without network considerations. According to the existing works on dynamic WSC and due to the lack of enough studies, it is essential to design a new method. In particular, this method must consider the distributed environment for the WSC. Therefore, bandwidth fluctuations, as the main source of network changes, should be taken into account while designing a robust solution during the design stage.

2.3 Representation of Composite Services and the Decoding

To apply EC algorithms effectively and efficiently, proper representations of solutions need to be designed. In this section, we present three different representations which have been used in the literature: graph representation, tree representation and sequence representation. These representations can directly or indirectly represent the composite Web service. In this thesis, we apply an indirect representation (i.e., sequences). A decoding process is then required to convert each sequence to a corresponding workflow. In the following, the representations for WSC problems, and their pros and cons will be discussed. Afterwards, the decoding process which is used in combination with our representation will be presented.

Service compositions are often represented as directed acyclic graphs (DAGs) [39]. DAGs are the most natural ways to represent composite services [49]. The DAG representation facilitates the enforcement of dependencies between services because it directly represents the solution, making it straightforward to check feasibility. In DAGs, one node represents the start point of the composition, S_0 , and another node serves as the endpoint, S_{p+1} (p is the number of component services in the composite service). All other nodes represent Web services. An edge expresses the output of one service feeding into the input of another.

In tree-based representations, solutions are represented using trees where inner nodes consist of various workflow constructs (see details in Subsection 2.1.3) directing the flow of the composition, and leaf nodes consist of Web services used as basic components [47, 220].

Tree-based representation presents all composition structure types as

inner nodes of trees and the services as the outer nodes [50, 118, 154, 208, 219, 220]. Firstly, they allow for the creation of compositions with variable lengths and configurations, reflected in the breadth and depth of the tree structure. Secondly, they allow for the inner and leaf nodes of the tree to adopt different meanings according to the representation scheme used. Finally, during the evolutionary process tree-based representations can be modified by genetic operators relatively simply, since it is possible to isolate changes to a given sub-tree within the overall structure. Despite these advantages, there may be multiple leaf nodes representing the same candidate service, and the dependencies with other nodes must be correct for all of them. This makes it more difficult to ensure the feasibility of a solution.

When compared to the tree representation, the graph-based structure simplifies the process of verifying the functional correctness of a given solution, since each candidate service is represented by a single node only and the connections between nodes are explicitly represented. However, the modification of graph-based representations using genetic operators can be difficult, as removing any given node in the graph may impact a chain of successors. However, both tree and graph representations require sophisticated EC operators which also must ensure the functional correctness of the composite service when manipulating them.

Sequence representations [30, 46, 160] do not show the final service composition solutions directly. Sequences must be decoded to produce an executable workflow of service composition solutions. A sequence representation for Web service composition owns the benefit of simplifying the evolutionary process by separating the quality optimisation from the enforcement of functional correctness constraints and being memoryefficient because of the linear structure [52]. Additionally, the performance of indirect representation has been compared with the DAG representation in previous work [52] on a very similar problem (i.e., fully-automated WSC) with the same decoding process. That work has confirmed that the sequence representation combined with the decoding process (Subsection 2.3.1) outperforms the graph representation in terms of both efficiency and effectiveness.

In this thesis, we consider two levels of representations. The DAG representation is used for evaluation, while the sequence representation is used for searching (EC operators are applied on sequences). We will use a separate step, called the decoding process, to convert a sequence to the corresponding workflow which must ensure the functional correctness.

2.3.1 Decoding Sequences into Workflows

The decoding process is a graph-building algorithm [54] and automatically transforms sequences into a corresponding executable service composition, i.e., a feasible workflow [158]. The decoding process is a crucial part of our fully-automated DWSC and is responsible for the functional correctness of the composite service. This process starts form the end node and constructs the graph until it reaches to the start node.

We use the decoding algorithm proposed in [51] which employs *layer* information of services. Layer information of a node specifies the minimum depth of predecessors between it and the start node and is obtained by a simple discovery algorithm [185]. The algorithm identifies a set of layers, \mathcal{L} , ensuring that each service $S_i \in S\mathcal{R}$ belongs to at most one layer ($S\mathcal{R}$ is the service repository). In other words, for all layers $\mathcal{L}_k, \mathcal{L}_m \in \mathcal{L}$ where $1 \leq k, m \leq |S\mathcal{R}|, \mathcal{L}_k \neq \mathcal{L}_m \land S_i \in \mathcal{L}_k \rightarrow S_i \notin \mathcal{L}_m$. The decoding approach works from the end node towards the start node, progressively fulfilling pending service inputs. Dangling nodes do not occur when building compositions from the end node, because services must provide useful outputs to be included in the composition [51]. In this context, the layer information helps to prevent cycles from forming during the decoding process. The decoding process is shown in Algorithm 3, which is also illustrated in Example 1.

Algorithm 3 The decoding algorithm [51]
INPUT: a sequence(\mathcal{Z}), a set of layers \mathcal{L}
OUTPUT: a composite solution (CS)
1: Set the set of concepts \mathcal{O}_0 to include the inputs of the task ($\mathcal{O}_0=I(\mathcal{T})$);
2: Set CS to include the end node;
3: while not all the concepts in \mathcal{O}_0 have been satisfied do
4: Get the next service, S , in Z
5: if outputs of <i>S</i> satisfy at least one concept in O_0 , and <i>S</i> layer < current
service layer then
6: Update \mathcal{O}_0 by removing that concept and adding the inputs of \mathcal{S} ;
7: Connect S to CS
8: end if
9: end while
return CS

In principle, running a depth-first search (DFS) on the reverse DAG starting from the end node traverses the reverse DAG in level order and obtains the sequence. This is the concept that is used in the backward decoding process. Therefore, every possible composite service for the given service composition problem can be obtained by using our indirect representation and the decoding process. This means that our algorithm has the possibility to evolve a service sequence that can be converted to a composite service in the DAG form with optimal performance.

Example 1. Figure 2.10 (a) shows a sequence of services, a task $T = \langle \{a\}, \{g\} \rangle$, (with $\{a\}$ as the input and $\{g\}$ as the output). A table indicating inputs and outputs of each service is also shown in the figure. The decoding process checks the sequence from left to right and generates a workflow until the task is satisfied. It starts with the output of the service task, $End(g, \emptyset)$ and looks for services in the sequence that provide the input of End. Since service S_4 outputs $\{g\}$, which is required by End, the decoding process connects it to End. Afterwards, the decoding process continues to scan the sequence for services that can satisfy $I(S_4)=\{d,e\}$. Since $O(S_1) = \{e\}$, and $O(S_6) = \{d\}$, the union of the two services S_1 and S_6 satisfies the input of service S_4 . Afterwards, the decoding process

looks for service(s) whose outputs satisfy the inputs of service S_1 and S_6 . Since service S_2 outputs $\{c, b\}$, which satisfy the inputs required by both services S_1 and S_6 , service S_2 is connected to both S_1 and S_6 as their predecessor. The process continues by connecting service S_5 (i.e., until Start is reached). The original sequence is then modified to contain only those services used in the workflow.



Figure 2.10: Decoding process: (a) a sequence representing a solution which is converted to a DAG during a backward decoding process, and (b) the sequence after decode.

As it is shown in Figure 2.10 (b), after the decoding process, the length of a sequence may be reduced to include only those services used during the decoding. This results in shorter sequences compared to their parents. We call this *variable-length* sequence representation [54]. As we will see in Subsection 4.4.1 and Section 4.5, EC operators that manipulate services in the sequence might add extra services to the sequence. If the length of the sequence is not controlled, extremely long sequences can be created with many redundant and repeated service. In particular, the variable-length technique is useful because it allows us to use flexible EC operators which are not restricted to the length of the sequence and removes redundant services after the decoding. However, when using general EC operators [54], which does not add other services to the sequence but re-order services in the sequence, fixed-length representation can be used which does not require refining the sequence after decode.

2.4 Chapter Summary

In this chapter, we presented a background on Web services and WSC. In particular, we briefly discussed the distributed DWSC and provided a motivating example of the real-world application of DWSC. We also discussed existing solution methods for the WSC problem in EC-based and non-EC groups.

We then reviewed current approaches on WSC. Most of existing WSC approaches do not deal with the distributed environment for services. Therefore, there is a need for effective problem definition for service composition approaches that take consideration of distributed nature and data-intensity of the component services. In particular, we need to define our problem in a way that our method can produce service compositions that involve the least amount of data transportation between services while selecting services with good quality. Our proposed approaches in this thesis can effectively handle the single-objective, multi-objective and dynamic distributed DWSC.

In addition to updating the optimisation objective to include distribution attributes, specific methods should be designed for the distributed DWSC. For example, some research has dealt with the distributed DWSC problem only through changing the objective function to include datarelated attributes [220]. However, the QoS of the composite services is poor due to the use of general approaches and ignoring problem-specific knowledge, such as the location of services. To find effective composite services, we must design new methods to the distributed DWSC.

A small collection of approaches has also been proposed to solve the

fully-automated multi-objective WSC [45, 53, 221]. However, existing research works have ignored the distributed nature of Web services. For the multi-objective distributed DWSC, developing methods which both improve the quality of solutions and alleviate the computational complexity of the local search is still an open problem. Accordingly, this has motivated us to use problem-specific techniques to develop efficient and effective algorithms to solve the distributed DWSC.

Further, the current capability of MOEAs to handle constraints in DWSC should be extended to: 1) perform the challenging task of simultaneously satisfying QoS constraints, 2) find a functionally correct composition with a considerable number of existing services, and 3) optimise QoS with trade-offs between them. The focus will be on proposing an effective repair technique to repair infeasible solutions.

Additionally, based on the preceding discussions, existing research on WSC problems lacks designing an efficient composite service to handle the dynamic network environment. Motivated by the large-scale distributed DWSC problem, the aim of this thesis is to propose an efficient fully-automated GA-based method for solving the distributed DWSC problem, considering the distributed nature of data-intensive services.

Chapter 3

Single-Objective Distributed DWSC Problem

3.1 Introduction

In this chapter, we define a formal model for the static single-objective distributed DWSC problem. This model will be extended to multi-objective distributed DWSC and dynamic distributed DWSC in Chapters 6 and 7, respectively. As we described in Chapter 1, we need to model the location information of services and data as well as the bandwidth for each communication link. Existing approaches have mainly considered a centralised service environment for WSC without any considerations of realworld bandwidth or location of services [46, 220]. However, the distribution of services and communication attributes are crucial factors in the final QoS of composite solutions.

In this chapter, we focus on the problem model and benchmark datasets. The following objectives are sought in this chapter:

- 1. To propose a model that explicitly considers communication, data and service attributes that affect the QoS of a composite service;
- 2. To propose an objective function for the single-objective distributed

DWSC;

- 3. To propose a model to obtain location information of services and their geographical distance. We can obtain latitude and longitude of services. Afterwards, we calculate the pairwise distance between services using the haversine formula; This formula calculates the distance between two points on a sphere (e.g., earth surface) using their latitudes and longitudes;
- 4. To propose a bandwidth simulation model based on the real-world observations, using Curtin dataset [145]. W obtain bandwidth dynamics for each communication link on the network for evaluating dynamic distributed DWSC.

3.2 Chapter Organisation

The remainder of this chapter is organised as follows: Section 3.3 presents the problem model. Section 3.3.3 discusses the objective function of the single-objective distributed DWSC problem. Section 3.4 presents the benchmark dataset. Finally, Section 3.5 summarises this chapter.

3.3 Problem Model

This subsection presents concepts and terminology used in the distributed DWSC. In particular, this subsection discusses the distribution model, the service and data model and the objective function of the single-objective distributed DWSC problem. We introduce the following definitions:

Definition 3. A data-intensive Web service is a tuple $S_i = \langle I(S_i), O(S_i), D(S_i), QoS(S_i), h_j \rangle$, where $I(S_i)$ is a set of inputs required for the execution of S_i , and $O(S_i)$ is a set of outputs produced by S_i . $D(S_i) = \langle D_{i,1}, ..., D_{i,m} \rangle$ is a set of m data items required by S_i . h_j denotes the server

which is hosting S_i . S_i obtains the data from a data centre and can receive other data from a previous service in the composition workflow through its inputs. $QoS(S_i) = \langle Q^1(S_i), Q^2(S_i), ..., Q^q(S_i) \rangle$ is an associated tuple of quality attributes describing the non-functional properties of S_i .

In this thesis, for each Web service, we suppose $QoS(S_i) = \langle T_{service}(S_i), C_{service}(S_i) \rangle$, indicating the two frequently used quality attributes (i.e., the total time and cost of executing service S_i). They are the most commonly used QoS attributes in most WSC works [30, 48, 191, 195, 227].

Definition 4. A service repository is denoted as SR and consists of a finite collection of Web services S_i , $i \in \{1, ..., n\}$, with n as the number of services in the repository [50].

Definition 5. The geographical location of a server, h_j , is identified by its longitude and latitude ($\langle Lat_j, Long_j \rangle$). Services in a service repository are distributed over servers in a distributed environment. Any two services on the same server share the same location.

In this thesis, we assume that all data and services have been deployed on servers distributed across the network and have location information.

Definition 6. A service request (or a task) is defined as $\mathcal{T} = \langle I(\mathcal{T}), O(\mathcal{T}) \rangle$, where $I(\mathcal{T})$ is a set of inputs provided to the composition by a user, and $O(\mathcal{T})$ is a set of task outputs expected by the user to be produced by the composite service [50].

Definition 7. A composite service, CS, is essentially a workflow of p component services specifying the order in which they should be run, as well as the necessary communication between them, to reach the desired outputs. Two special services can be used to describe the overall composition's inputs and outputs: Start with $I(Start) = \emptyset$ and $O(Start) = I(\mathcal{T})$, and End, with $I(End) = O(\mathcal{T})$ and $O(End) = \emptyset$. Our composed workflows consist of a combination of parallel and sequence construct, mirroring the common structures offered in Web service composition languages such as Business Process Execution Language (BPEL4WS) [23].

Constructs can be recalled from Chapter 2. For a given task T, we aim to find a composite service (CS) that takes I(T), and produces O(T).

Definition 8. A composition is functionally correct, or feasible if (1) the input(s) of each of the component services is (are) fulfilled by the outputs of its preceding services or by $I(\mathcal{T})$, and (2) the output of the composition produces $O(\mathcal{T})$ (node End is satisfied). A direct edge which connects two services S_i and S_j in a workflow indicates that there must be a communication link and data dependency between them in the network [50].

3.3.1 Distribution Model

In this subsection, we define communication link attributes for the static distributed DWSC, which are shown in Figure 3.1. However, these attributes for the dynamic distributed DWSC will be introduced in Chapter 7, where the communication attributes are time-dependent.

Definition 9. A communication link, l_k , is a logical link between two directly connected services in a composite service, and represented as an edge in the corresponding workflow. A communication link is defined as the tuple $l_k = \langle S_{head}, S_{tail}, B_{l_k} \rangle$, where S_{head} is the source service of l_k and S_{tail} is the destination service of l_k . In a composite Web service, a point-to-point communication happens between S_{head} and S_{tail} if communication link l_k in the workflow connects S_{head} and S_{tail} . B_{l_k} is the bandwidth of l_k . The set of all communication links in CS is denoted by CL(CS).

Definition 10. The inter-service distance between two services S_i and S_j is the length of the physical link or a combination of physical links if there is no direct but a multi-hop link between S_i and S_j in the network.





We employ the inter-service distance concept to design crossover, local search and mutation operator in Chapters 4 and 7, respectively.

Definition 11. Communication cost, $C_L(l_k)$, is the cost associated with each communication link l_k and is defined as the cost to transfer a data over l_k (between two communicating Web services). Additionally, data can move from a data centre to a Web service, where the communication cost is denoted by C_{LD} .

In this thesis, we suppose that the communication cost depends on the length of the communication link according to the literature [171, 227].

Definition 12. Transfer time, $T_{tr}(l_k)$, is the required time for a service to put all bits of data $D_{i,m}$ over l_k and depends on the bandwidth, B_{l_k} and the size of data to be transferred over l_k . Transfer time is obtained using Equation (3.1):

$$T_{tr}(l_k) = \frac{size(D_{i,m})}{B_{l_k}}.$$
(3.1)

Definition 13. Communication time, $T_L(l_k)$, is the time required for communicating between two services through communication link l_k . Communication time consists of the transfer time and the propagation delay $T_P(l_k)$ (Definition 14). The source server is either a data centre hosting the data or a service in the composition which produces data required by other services. In the second case, we denote the communication time by T_{LD} . **Definition 14.** Propagation delay is the duration required for the first byte of the data to travel from the source to the destination over the communication link l_k .

According to the literature [227], in this thesis, we assume that the propagation delay depends on the communication link length. Therefore, in this thesis, we obtain a propagation delay for each communication link by generating a random number between zero and the value of the geographical distance between its S_{head} and S_{tail} . The reason behind this is that studies have verified the communication time mainly depends on the communication link's length, e.g. [168, 227].

The QoS of a composite solution is derived from aggregating the corresponding QoS attributes of all of the component services involved, and the communication time and cost between those services. In the following, we introduce the service and data model, and the communication model, respectively.

3.3.2 Service and Data Model

In this subsection, we calculate $C_{service}$ and $T_{service}$ which are cost and time of a data-intensive Web service. An example of a workflow with all the required terms for calculating $C_{service}$ and $T_{service}$ (T_{proc} , C_{sp} , C_{dp} , T_{LD} , C_{LD}) is illustrated in Figure 3.2. These terms are introduced below:

Definition 15. Service execution time $(T_{proc}(S_i))$ is the time for executing S_i [197].

Definition 16. Service provision cost, $C_{sp}(S_i)$, is the price charged to use service S_i and is usually specified by service providers.

Definition 17. Data provision cost, $C_{dp}(D_{i,m})$, is the cost applied by the data provider, *i.e.*, the cost to provide the data.

We can calculate the cost of a component service, S_i , by aggregating its data access and execution cost, as shown in Equation (3.2) and Figure 3.2:



Figure 3.2: A component service and its attributes.

$$C_{service}(S_i) = C_{sp}(S_i) + \sum_{D_{i,m} \in D(S_i)} (C_{dp}(D_{i,m}) + C_{LD}(D_{i,m}))$$
(3.2)

where $D(S_i)$ is the set of data items required by S_i and $D_{i,m}$ is the m^{th} element in $D(S_i)$. A Web service obtains these data items from the data centre.

Similarly, the execution time for a component service is obtained by Equation (3.3) and Figure 3.2:

$$T_{service}(S_i) = Tproc(S_i) + \sum_{D_{i,m} \in D(S_i)} T_{LD}(D_{i,m})$$
(3.3)

where $T_{LD}(D_{i,m})$ is the communication time for service S_i to access $D_{i,m}$.

3.3.3 Objective Function

In addition to the service-related cost and time, a Web service receives data from a previous service in a composition through its inputs. Therefore, the total cost of a composite service is obtained using Equation (3.4).

$$C_{total}(\mathcal{CS}) = \sum_{S_i \in \mathcal{SR}} C_{service}(S_i) + \sum_{k \in \mathcal{CL}} C_L(l_k)$$
(3.4)

where SR is the service repository and CL is the set of component services in CS.

As mentioned in Chapter 2, in the presence of the parallel construct, the execution time is calculated in a different way than the cost. A *path* form *Start* to *End* is a sequence of services in the workflow and communication links between them. Suppose that δ is the set of all paths in a composition *CS*. Δ_j is the execution time of path *j*. Δ_j includes the time for executing component services and the communication time of each link on path *j*. Δ_j can be obtained through Equation (3.5):

$$\Delta_j = \sum_{S_i \in Q} T_S(S_i) + \sum_{l_k \in L} T_L(l_k)$$
(3.5)

where Q and L are the set of component services and communication links (between two consequence services) on path j, respectively. Hence, T_{total} , the overall response time of CS, is calculated as the time of the most timeconsuming path in the composition as it is shown in Equation (3.6):

$$T_{total}(\mathcal{CS}) = \max_{j \in \delta} \left\{ \Delta_j \right\}$$
(3.6)

The longest path time-wise from the start node to the end node is identified using the Bellman-Ford algorithm [24].

Finally, the goal is to minimise the function in Equation (3.7) by finding the optimal composite services composed of individual services over a given repository. Accordingly, the best solution is a composition with the minimum value of F.

$$F(\mathcal{CS}) = w\hat{T}_{total}(\mathcal{CS}) + (1 - w)\hat{C}_{total}(\mathcal{CS})$$
(3.7)

where \hat{T}_{total} and \hat{C}_{total} are normalised values of T_{total} and C_{total} , respec-

66

3.4. BENCHMARK DATASET

tively. *w* is a positive real weight.

The normalised attributes are obtained through following Equations:

$$\hat{T}_{total} = \frac{T_{total} - minT_{total}}{maxT_{total} - minT_{total}}$$
(3.8)

$$\hat{C}_{total} = \frac{C_{total} - minC_{total}}{maxC_{total} - minC_{total}}$$
(3.9)

The lower bounds (i.e., $minC_{total}$ and $minT_{total}$) are set to zero. The upper bounds (i.e., $maxC_{total}$ and $maxT_{total}$) are obtained by adding the cost and time of all communication links and services in the repository, respectively.

The user of the composite service can set the weight (w in Equation (3.7) to specify the relative importance of total time and cost according to their wish. Following other WSC research works [20, 54, 125, 204], we set weight preferences to w = 0.5, to put equal importance on the time and cost in the fitness function (note that the value of w can be chosen freely in practice).

3.4 Benchmark Dataset

Experiments will be performed in Chapters 4, 5, 6 and 7 using WSC-2008 [11] and WSC-2009 [98] benchmark datasets. WSC-2008 and WSC-2009 contain eight and five repositories, respectively, with a varying number of services. A taxonomy of concepts is provided for determining which inputs and outputs are compatible and several service composition tasks are also given per repository [11, 98]. These datasets were chosen because they are the largest benchmarks that have been broadly exploited in the WSC literature [30, 49, 54, 104, 159, 220];

To extend existing benchmark datasets for WSC (WSC2008 [11] and WSC2009 [98]) and to add the location and bandwidth information, we have employed two other datasets, i.e., *WS-Dream* [227] and *Curtin* [145].

We have used the latitude and longitude of real-world services in WS-Dream to specify a location to each Web service. Additionally, we have used this information to obtain the geographical distance on the earth surface between each pair of services in Subsection 3.4.1. Curtin dataset includes packets inter-arrival time and packets length and will be used to define a bandwidth model in Subsection 3.4.2.

Since the original WSC datasets did not include QoS information for any service, we have further obtained QoS settings, e.g. $T_{proc}(S_i)$ and $C_{sp}(S_i)$, from the QWS dataset [4]. C_{dp} is generated randomly in the interval (0,1].

Finally, according to the previous discussion in this chapter, we require two types of data: the data from a data centre and the data from a previous Web service output. The size of the first one is generated randomly and assigned a location. The second one is set based on the input/output number of each Web service, which is provided in benchmark datasets WSC2008 and WSC2009.

3.4.1 Geographical Distance of Services

In this thesis, we employ geographical distance of services as a heuristic, to design effective algorithms. Additionally, we obtain location information of services and extend benchmark datasets to include this information. We use the geographical distance, $d_{j,k}$, between servers h_j and h_k on the earth surface as the length of the communication link. Therefore, we employ a simplified version of the *haversine* formula, shown in Equation (3.10) (Note that several servers can be located on the same server).

$$d_{j,k} = 2r \times \arcsin \times \left(\sqrt{\sin^2\left(\frac{Lat_k - Lat_j}{2}\right) + \cos\left(Lat_j\right)\cos(Lat_k)\sin^2\left(\frac{Long_k - Long_j}{2}\right)}\right)$$
(3.10)

3.4. BENCHMARK DATASET

where r is radius of the earth, Lat_j and Lat_k are latitudes of servers h_j and h_k , and Lat_j and Lat_k are longitudes of the two servers, respectively.

3.4.2 Communication Bandwidth

In this section, we propose a method to obtain bandwidth values. We calculate the bandwidth variations at one-minute intervals across a full day using *Curtin* dataset [145], which includes packets inter-arrival time and packets length. Thereafter, the maximum and the minimum value of the observations are recorded. Maximum and minimum values indicate the maximum and minimum available bandwidth for a communication link (which correspond to the lowest and highest workloads, respectively). Based on the maximum and minimum bandwidth observed, we set 20% and 70% of the bandwidth range as thresholds on the bandwidth to determine the percentage of time for a communication link to provide low, medium and high bandwidth, which can be indicated respectively as p_l , p_m and p_h . Details on the calculation of the percentages are given in Example 2 below:

Example 2. Suppose that $B^m = 0$ and $B^M = 10$ are the minimum and maximum observed bandwidth values, respectively. There exist 1440 (24*60=1440) observations per day. The bandwidth range is $B^M - B^m = 10$. Therefore, the threshold values are two and seven, and the low, average and high bandwidth ranges are [0,2], (2,7] and (7,10], respectively. If the number of observations within [0,2], (2,7] and (7,10] bandwidth ranges are 700, 300 and 440, respectively, then p_l , p_m and p_h are calculated as follows: $p_l = 700/1440 = 0.486$, $p_m = 300/1440 = 0.21 p_h = 440/1440 = 0.305$

To obtain a bandwidth value, a random number, r, is generated according to the percentages p_l , p_m and p_h obtained. Each communication link will have a bandwidth value according to the percentage obtained. The link is assigned with the minimum available bandwidth if $r \leq p_l$. Similarly, the maximum and the average bandwidth are assigned to the link if $p_l < r \le p_m$ and $p_m < r \le p_h$, respectively. However, two servers might be connected through a multi-hop link. To obtain the bandwidth of a multihop communication link, we independently obtain bandwidth values for each sub-link and their minimum is treated as the main communication link's bandwidth. The number of hops between each pair of servers has been obtained from the network topology map of *Sprint*. In our bandwidth calculation, a multi-hop communication link is more likely to have low bandwidth. This is consistent with the findings in the literature [168]. After the bandwidth has been sampled for each communication link, the transfer time can be calculated using Equation (3.1).

The bandwidth value obtained in this subsection, will be employed during the static WSC. However, for the dynamic WSC, a set of bandwidth samples will be generated (details can be found in Chapter 7).

3.5 Chapter Summary

In this chapter, we defined the distributed DWSC problem. In particular, we proposed a model for the distributed DWSC problem that considers the location of services and the communication among them. We explained how we obtained communication, data, and service-related attributes. To set up benchmark datasets for the distributed DWSC for experimental evaluations, we proposed a method to obtain location information form a real-world dataset as well as a simulation model to obtain the bandwidth values of each communication link.

Chapter 4

Single-Objective Distributed DWSC Methods

In this chapter, we will propose new and effective methods to the singleobjective distributed DWSC. We will design Memetic Algorithms (MAs), which employ domain knowledge during the initialisation (*Cluster-guided MA*) of the EC algorithm, during the local search and within crossover operators (*Distance-guided MA*).

4.1 Introduction

Existing approaches to WSC employ general EC operators [51, 54]. They assign crossover points randomly without considering any problem-knowledge in particular knowledge regarding which part of the solutions should be preserved to the next generation. The following objectives will be considered in this chapter:

- 1. To propose an effective initialisation for the GA using a clustering technique;
- 2. To propose effective crossover operators for GA which can find highquality composite services through reducing time while executing

composite services. Based on these crossover operators, we will propose the *distance-guided MA* for distributed DWSC. Two different heuristics, namely the *distance between services* and the *longest common sub-sequence*, will be developed to create crossover operators;

- 3. To propose two efficient local search strategies by exploiting the distance information. A Memetic Algorithm, called *flexible distanceguided MA*, will be proposed based on the two strategies. Thorough comparisons of the two strategies and with an existing local search for a very similar problem (i.e., fully-automated WSC)) will be conducted to show the effectiveness of the proposed Memetic Algorithm;
- 4. To perform empirical experiments that demonstrate the effectiveness of explicitly considering communication distance between any related services while optimising the composite service. We will specifically study the initial population in cluster-guided MA and EC operators in distance-guided and flexible distance-guided MAs. The newly developed crossover and local search operators, including the distance-guided crossover and the flexible local search, will be explored further in Chapters 5, 6 and 7.

4.2 Chapter Organisation

The chapter is organised as follows: The cluster-guided MA is presented in Section 4.3. Section 4.4 discusses the distance-guided MA with new crossover operators. New local search procedures and neighbourhood finding strategies are presented in Section 4.5. Evaluations and results are presented in 4.6. Finally, Section 4.7 summarises this chapter.

4.3 Cluster-Guided Memetic Algorithm

In the literature, the initial population of MA is mainly generated at random. The cluster-guided MA exploits the distributed nature of services to generate the initial population for the MA. To build up the initial population, services are clustered according to their geographical locations using k-means clustering. Afterwards, the MA creates solutions in each cluster. In this way, the algorithm encourages composite services to use locally located component services to find composite services with low communication cost and time (by using services which are located near each other). This is important because communication delay and cost depend on the distance between services. To this aim, we propose to build the initial population within each cluster instead of the repository.

The outline of the proposed approach is shown in Figure 4.1. Sequences of services are created within each cluster only by using those services in the same cluster. Each sequence has the same size as that cluster. Each sequence is then given to the decoding process which converts them to the corresponding workflow (Subsection 2.3). The fitness is then calculated for each workflow and the most appropriate candidates are selected. Finally, the whole process is repeated after applying the crossover, mutation and local search operators on the corresponding sequences (Subsection 4.3.1).

clustering initialisation decoding selection



Figure 4.1: Steps of the cluster-guided GA

The pseudo-code of cluster-guided MA for distributed DWSC is presented in Algorithm 4. After the clustering is performed, the mean fitness value is calculated for each cluster (the mean fitness of all solutions' workflows). Clusters are sorted according to their mean fitness values and then are given weights so that clusters with better fitness value gain more weight and more chances for their individuals to be selected for crossover, mutation and local search operators.

Algorithm 4 Cluster-guided MA for service composition

INPUT: Service Repository

OUTPUT: A Service Composition Solution, BestFitness

- 1: Apply clustering to the service repository and generate *n* clusters;
- 2: Generate permutations (solutions) within each cluster;
- 3: Decode permutations into workflows (Subsection 2.3);
- 4: Calculate mean fitness of each cluster (mean value of all workflows inside the cluster) and assign the value to permutations in that cluster;
- 5: while the maximum number of generations not reached do
- 6: Among all clusters, select permutations based on their assigned fitness value using roulette wheel selection operator [79];
- 7: Apply crossover, mutation and local search operators to generate new permutations;
- 8: Decode all newly generated permutations and calculated fitness;
- 9: **if** *fitness* is better than *Bestfitness* **then** *Bestfitness=fitness;*
- 10: **end if**
- 11: end while

return Composite service with *Bestfitness*.

The probability of a cluster being selected is obtained by Equation (4.1):

$$p_{i} = \frac{f_{i}}{\sum_{j=1}^{C} f_{j}}$$
(4.1)

where C is the number of clusters and f_i is the average fitness for cluster

74

i.

Crossover, mutation and local search operators are discussed in Subsection 4.3.1. The crossover operator is first applied among clusters to interchange services between clusters. In the next generations, solutions are selected based on their individual fitness values.

4.3.1 Evolutionary Operators for Cluster-guided MA

In this subsection, we present EC operators used in the cluster-guided MA for the distributed DWSC. These operators mainly focus on modifying the order of the services in the sequence to produce better solutions.

For the mutation operator, one solution is selected from the population and then two services are swapped in the solution sequence. Consider Figure 4.2 as an example of the mutation operator, where services S_3 and S_8 are randomly selected on the parent and swapped to produce the child.



Figure 4.2: Mutation operator, two random Web services are chosen from *parent* and swapped to generate *child*.

We use the *index crossover* [57] employed for a very similar problem (i.e., fully-automated WSC) in [54], where a random crossover point for each parent is determined (i.e., indices are independently chosen for the two parents). Subsequently, each parent is split from the index point into two sub-sequences (i.e., a *prefix* and a *suffix*). To generate offspring sequences, a parent is embedded within the *prefix* and the *suffix* of the other parent. Figure 4.3 illustrates an example of this crossover operator.

The local search operator, illustrated in Figure 4.4, repeatedly creates neighbours of the current sequence [51]. The pseudocode of the local search is presented in Algorithm 5. The input of the local search procedure is a Web service composition sequence. This procedure randomly chooses a service in the sequence. Subsequently, it swaps the selected service with all of the other services in the sequence, each time creating a different sequence. Each sequence is decoded into a composite Web service DAG and evaluated. The DAG with the best fitness is chosen to replace the original sequence.



Figure 4.3: Index crossover for WSC [54].

4.3.2 Clustering of Web services

We have used a modified version of the standard K-means algorithm to produce clusters of equal size, which is presented in Algorithm 6. This version of K-means tries to produce clusters with a similar size (a cluster with a small number of services most likely produces functionally infeasible solutions). The algorithm regularly checks whether each cluster is full or not.





Algorithm 5 Local Search procedure

INPUT: a service sequence (\mathcal{Z}) **OUTPUT:** Best sequence

- 1: Best fitness = $F(\mathcal{Z})$;
- 2: Best sequence=Z;
- 3: Randomly select a service, *S*, in *Z*;
- 4: **for** each service S' in Z **do**
- 5: Swap S with S' to obtain sequence Z';
- 6: Decode and evaluate \mathcal{Z}' ;
- 7: **if** $F(\mathcal{Z}') < Best fitness$ **then**
- 8: Best $fitness = F(\mathcal{Z}');$
- 9: Best sequence= \mathcal{Z}' ;
- 10: **end if**
- 11: end for
 - return \mathcal{Z}' ;

4.4 Distance-Guided Memetic Algorithm

GAs rely heavily on crossover operators to derive an offspring population by combining parts of existing solutions. In this section, we design distance-guided crossover operators to generate new offspring composite services that eliminate *bottleneck* communication links (i.e., the longest link between two adjacent services in a composite service). In distributed Algorithm 6 Equal-size K-means clustering for cluster-guided MA.

INPUT: Repository, Pairwise haversine distance of services **OUTPUT:** A set of non-overlapped clusters

1: Initialisation phase

- 2: Set the cluster size;
- 3: Initialise the mean of each cluster by random services;
- 4: Calculate δ for each service (δ = distance to the nearest cluster mean distance to the farthest cluster mean);
- 5: Order services by δ value;
- 6: Go through the ordered services and assign each service to its nearest cluster (if a cluster is full, consider the next nearest cluster).

Iteration phase

- 7: **for** each cluster **do**
 - Compute the new mean;
- 8: end for
- 9: for each service do

Compute δ ;

10: end for

Sort services based on δ ;

- 11: **for** each service based in the sorted list **do**
- 12: **for** each other cluster, by service gain, unless already moved **do**
- 13: **if** there is a service wanting to leave the other cluster and this swap yields and improvement **then**
 - Swap the two services;
- 14: **end if**
- 15: **if** the service can be moved without violating size constraints **then** Move it;
- 16: **end if**
- 17: **if** the service was not moved **then** Add to outgoing transfer list;
- 18: end if
- 19: end for
- 20: **end for**
- 21: **if** no more transfers were done (or max iteration threshold was reached) **then** Terminate.
- 22: **end if**

DWSC, bottleneck links determine to a large extent the QoS of a composite service [33]. We further develop a service composition algorithm with new crossover operators that not only resolves bottleneck links but also preserves promising common sub-components in existing solutions while using them to build new composite services. To this aim, we propose a heuristic named the *longest common sub-sequence (LCS)*. All of the above crossover operators will be examined in Section 4.6.

In this subsection, we propose three different crossover operators for the distributed DWSC and study their performance in MA. These crossover operators are distance-guided single-point, distance-guided double point and distance-guided LCS.

Algorithm 7 presents the pseudocode of the distance-guided MA. The initial population is created by randomly ordering all the services in the repository to form sequences. At line 2, the decoding process is performed (see Subsection 2.3), after which solutions are evaluated using the fitness function (Section 3.3). Elite solutions are copied to the next generation (line 4). Solutions chosen by a tournament selection (line 5) are involved in the crossover operator (line 6). One of the crossover operators (see Subsection 4.4.1) is used to produce offspring solutions. Subsequently, the mutation and local search are applied to the offspring solutions (lines 7 and 8). These two operators are the same for cluster-guided and distance-guided MAs (for details see Subsection 4.3.1). Distance-guided crossover operators can produce long sequences with duplicated services. Therefore, redundant services are removed from the sequence of newly generated solutions after the decoding process. The above steps are repeated until the predefined number of generations is reached.

4.4.1 Crossover Operators for Distributed DWSC

In this section, we first introduce the *distance-guided single-point* crossover operator. We then define the *longest common sub-sequence (LCS)* heuristic

Algorithm 7 Distance-guided MA for distributed DWSC

INPUT: Service Repository (\mathcal{R}) , Task (\mathcal{T}) , Number of Generations (\mathcal{G}) **OUTPUT:** A Service Composition Solution

- 1: Generate *N* sequences with randomly ordered services in \mathcal{R} ;
- 2: Decode each sequence into a workflow and calculate the fitness;
- 3: while number of iterations $< \mathcal{G} do$
- 4: Find *e* elite solutions and copy them to the next generation;
- 5: Use tournament selection to select individuals based on their fitness values;
- 6: Apply a distance-guided crossover operator to the tournament winners, to generate N offspring solution;
- 7: Apply mutation operator to the offspring solutions;
- 8: Apply local search operator to the offspring solutions;
- 9: Decode all newly generated solutions (remove redundant services) and calculate their fitness;
- 10: end while return Sequence With the Best Fitness;

and enhance the crossover with the LCS heuristic to preserve good subsolutions (or building blocks) in offspring solutions.

New crossover operators are created based on the Index Crossover [54] (for details, see Subsection 4.3.1). This crossover is chosen because it has been effective in WSC with sequence representation [54]. However, this crossover does not employ any heuristics to produce effective offspring solutions. Moreover, this crossover selects the crossover point randomly without any consideration of domain knowledge.

Three distance-guided crossover operators are developed to utilise the distance information. It is expected that our proposed MA with these crossover operators outperforms existing algorithms in terms of both effectiveness and efficiency.

Distance-guided single-point crossover: this crossover uses the location of services as the key factor in determining the crossover point. As illustrated in Figure 4.5, this crossover is very similar to the index crossover. However, in this crossover, the crossover point is selected based on the distance of services to each other. The largest distance in *Parent1* is 150km which is between service S_3 and service S_4 , and 140km for *Parent2* be-

tween service S_{10} and service S_{11} . This crossover aims to enhance the fitness of the offspring solutions by eliminating the longest links of parents. To achieve this goal, the crossover point is set at the longest communication link expecting to reduce the inter-service distance and improving the QoS.



Figure 4.5: Distance-guided single-point crossover

Distance-guided double-point crossover: the crossover point in each parent is chosen based on the first and second longest distances between any pair of consecutive services. Using these two crossover points, each parent is split into three parts. To produce children (offspring solutions), parts of the first parent are combined with those of the second parent, one in between. An example of the distance-guided double-point crossover operator is illustrated in Figure 4.6. Two offspring solutions differ from each other in the order of combination. The diversity in the population is expected to increase by combining different-size parts.

Distance-guided LCS crossover: as illustrated in Figure 4.7, a heuristic is incorporated in this crossover operator to preserve the promising part of each parent. This heuristic uses the longest sub-sequence of services which appears in both parents and, therefore, is called the *longest common sub-sequence (LCS)*. The LCS is inherited directly by their children with-



Figure 4.6: Distance-guided double-point crossover

out any change. The crossover point is selected in the same way as the distance-guided single-point crossover operator except that this point cannot be inside the LCS. This is to avoid any change to LCS so that children can easily preserve good sub-solutions embedded in the LCS. Resulted sequences have redundant services which will be removed after the decoding process.

4.5 Flexible Local Search for Distributed DWSC

Despite the recent success of MA on WSC problems [46], only primitive forms of local search borrowed from algorithms designed for other relevant problems have been applied to DWSC [94, 149, 220]. A neighbourhood structure of local search specifies how to manipulate a solution to generate a neighbour solution which can outperform the original solution. We design problem-specific approaches for the distributed DWSC to address those issues.

In this section, we propose new local search strategies to be combined with the distance-guided MA. The GA algorithm is effectively hybridised


Figure 4.7: Distance-guided LCS crossover (the longest common subsequence is identified with grey).

with local search techniques designed based on the location information of Web services. Two neighbourhood creating strategies are designed using the location information of services giving rise to three different local search processes for the distributed DWSC problem, (i.e., DistanceCR-LCS-I, DistanceCR-LCS-II and DistanceCR-LCS-Flexible). The first and second local search processes employ one neighbourhood creating strategy. DistanceCR-LCS-II allows big changes to be applied to previously evolved composite services. The DistanceCR-LCS-Flexible employs a combination of the two strategies. The local search processes designed in this section replaces the local search step in Algorithm 7.

4.5.1 Neighbourhood Creating Strategies for Distributed DWSC

In this subsection, the local search aims to avoid long communication links, which lead to an increase in the overall communication time and cost. To demonstrate the proposed local search strategies an example of a solution is illustrated in Figure 4.8, where, the longest communication link



Figure 4.8: An example of a composite service and the bottleneck

of the solution and its two corresponding services, i.e., Web services S_1 and S_6 , are shown inside the dashed area. If we can eliminate this link or replace it and the two services by another alternative sub-path that performs the same task but with a shorter communication link, we can enhance the total quality of the solution.

The first strategy, *Strategy-I*, is to replace service S_6 in the example in Figure 4.8 by an alternative service provided that its output(s) can satisfy the input(s) of S_1 . If at least some of those services are probably located in a shorter distance to S_1 , the local search improves the quality of the solution. Suppose that there are four services in the repository that can perform the same functionality as S_6 (i.e., S_{11} , S_{15} , S_{16} and S_{19}). New neighbours are created each time by replacing S_6 with one of those services, as shown in Figure 4.9(a). Moreover, a prefix including random services from the repository (except those in the sequence) is attached to the beginning of neighbour sequences. As shown in [54], adding the prefix to the neighbour solutions is important for diversifying the composite services in the neighbourhood.

Referring to Figure 4.9(b), Strategy-II is to consider alternatives for the whole sub-path identified within the red dashed area in Figure 4.8, which includes S_1 , S_6 and their communication link. Rather than replacing Web service S_6 , which forces us to consider only feasible services that can follow immediately service S_1 , we first replace Web service S_1 . The justification of this local search is that if S_1 itself is located on a very distant server, (which is most likely), solution quality can be improved by avoiding that. Therefore, in this local search, alternative services are discovered to re-



Figure 4.9: Local search strategies (a) Strategy-I, and (b) Strategy-II.

place S_1 . This causes changes in a relatively larger part of a solution rather than a single Web service. Recall that the backward decoding process (see Chapter 2) generates a DAG from *End* to *Start*. In this example (Figure 4.9(b)), service S_7 is added to the composite service first. Next, service S_4 and service S_5 are connected. Afterwards, service S_3 is added. However, to eliminate the bottleneck link, service S_1 must be avoided. Therefore, the local search algorithm considers alternative services of S_1 that can also satisfy the inputs of service S_4 . Hence, upon creating neighbours, the decoding process creates four new solutions. Each solution is created by connecting one of the four services to service S_4 .

4.5.2 Flexible Local Search Process for Distributed DWSC

We examine both strategies (see Subsection 4.5.1) in the local search of the memetic algorithm for distributed DWSC. The local search procedure is represented in Algorithm 8. A composite service (current solution) is sent to the local search procedure after it has been selected. Exploring the neighbourhood of the solution is performed either by one of the strategies (explained in Algorithm 8) or by a combination of them where each strategy generates half of the neighbour solutions. Neighbours are evaluated and the one with the best fitness value replaces the original solution. If there are no improvements in neighbours, no replacement is performed. The hybridisation of the three local search techniques with distance-guided MA results in DistanceCR-LCS-I, DistanceCR-LCS-II and DistanceCR-LCS-Flexible, which will be examined in Section 4.6.

Algorithm 8 Flexible Local Search Process for distributed DWSC

INPUT: Current Solution, N_l (NeighbourhoodSize), Service Repository **OUTPUT:** Neighbouring Solution

- Generate N_l neighbours for *Current Solution* using Strategy-I (or Strategy-II) neighbourhood strategy, or N_l/2 neighbours for *Current Solution* using Strategy-I and N_l/2 neighbours for *Current Solution* using Strategy-II neighbourhood strategy (flexible local search);
- 2: Evaluate all neighbours;
- 3: if Best Neighbour's fitness is better than Current Solution's fitness then
- 4: Current Solution=Best Neighbour;
- 5: **end if**

```
return Current Solution;
```

4.6 Evaluations

In this section, we present the evaluation settings in Subsection 4.6.1. We then evaluate and discuss cluster-guided and distance-guided MAs in Subsections 4.6.2 and 4.6.3, respectively. Finally, the evaluation and the

discussion on flexible MA is provided in Subsection 4.6.4.

4.6.1 Parameter Settings and Datasets

For all methods in this chapter, the size of the initial population and number of generations are 100 and 30, respectively. Local search, mutation and crossover rates are 0.05, 0.05 and 0.95, respectively. These parameters' values were chosen based on popular settings discussed in the literature [99, 170, 204]. Following the literature [99], tournament selection of size two is adopted as the strategy for choosing candidate solutions for local search, mutation and crossover operators in all methods. Elitism with size two is applied.

Since EC algorithms are stochastic, all algorithms are run 30 times, each time with a different random seed. We then employ the Wilcoxon ranksum statistical test (U-test) with a significance level of 5% to confirm the observed difference in the average of the fitness value. We have implemented all competing algorithms in Java 8. Experiments were conducted on a desktop computer with 8 GB RAM and an Intel Core i7-4790 and 3.6GHz processor running Linux Arch 4.9.6. We used the implementation of GA provided by the ECJ toolkit (Java evolutionary computation)¹. Github links for the full implementation of all proposed algorithms have been provided ².

We employ the datasets set up in Chapter 3 for empirical evaluations, which include bandwidth value, data, and location information of services.

¹https://cs.gmu.edu/eclab/projects/ecj/

²https://github.com/soheilasadeghi

4.6.2 Evaluation of Cluster-guided and Distance-guided MAs

To examine the importance of using domain-knowledge in generating an effective initial population, we first compare the cluster-guided MA against the *Simple-MA* [54]. Simple-MA was first proposed to address the centralised WSC in [54]. However, we update it to use our objective function. Therefore, the difference between the two algorithms is that the cluster-guided MA employs the clustering for producing the initial population, which is expected to result in a better performance.

For the cluster-guided MA, various sizes of clusters (20, 30, 40, 50 and 100) have been considered and the best results have been reported in Table 4.1. Additionally, we examine the three newly designed crossover operators in Section 4.4. Therefore, four MAs which only differ in their crossover operator are compared: Simple-MA (employs index crossover in Subsection 4.3.1), DistanceCR-I (MA with single-point distance-guided crossover operator), and DistanceCR-LCS (MA with the distance-guided LCS crossover operator). These experiments also help us to identify the most effective crossover operator for the distributed DWSC problem and employ that in future experiments (Note that we compare and combine cluster-guided and distance-guided MAs in Chapter 5).

4.6.3 Discussions on Cluster-guided and Distance-guided MAs

Comparing cluster-guided and simple-MA in the first two columns of Table 4.1 indicates that cluster-guided MA which employs domainknowledge in generating the initial population achieves better results. Simple-MA randomly generates the initial population. The next generations are built based on the initial population. If the initial population is carefully created, as in cluster-guided MA, the algorithm is more likely to find high-performance composite services.

Table 4.1: Mean fitness values and standard deviations per 30 runs. Significantly better values in all four comparisons are highlighted (U-test with the significance level 0.05) (Note: the lower the fitness the better).

Task (SR size)	Simple-MA[54]	Cluster-MA	DistanceCR-I	DistanceCR-II	DistanceCR-LCS
WSC8-1 (158)	0.55 ± 0.09	$0.46 \pm 0.04 +$	0.45 ± 0.01	0.42 ± 0.04	0.41 ± 0.12
WSC8-2 (558)	0.52 ± 0.1	$0.44 \pm 0.14 +$	0.46 ± 0.03	0.42 ± 0.05	0.42 ± 0.02
WSC8-3 (604)	0.54 ± 0.1	0.53 ± 0.03	0.51 ± 0.04	0.48 ± 0.01	0.44 ± 0.02
WSC8-4 (1041)	0.53 ± 0.2	0.5 ± 0.09	0.48 ± 0.1	0.44 ± 0.08	0.4 ± 0.01
WSC8-5 (1090)	0.54 ± 0.1	$0.52 \pm 0.2 +$	0.5 ± 0.04	0.49 ± 0.1	0.46 ± 0.16
WSC8-6 (2198)	0.58 ± 0.13	$0.56 \pm 0.08 +$	0.57 ± 0.01	0.55 ± 0.02	0.46 ± 0.22
WSC8-7 (4113)	0.57 ± 0.01	$0.55 \pm 0.236 +$	0.59 ± 0.01	0.58 ± 0.04	0.54 ± 0.02
WSC8-8 (8119)	0.54 ± 0.08	$0.49 \pm 0.04 +$	0.53 ± 0.02	0.44 ± 0.09	0.45 ± 0.05
WSC9-1 (572)	0.59 ± 0.03	$0.54 \pm 0.23 +$	0.55 ± 0.03	0.57 ± 0.07	0.53 ± 0.02
WSC9-2 (4129)	0.56 ± 0.01	$0.51 \pm 0.04 +$	0.5 ± 0.002	0.46 ± 0.1	0.47 ± 0.02
WSC9-3 (8138)	0.55 ± 0.09	0.54 ± 0.04	0.52 ± 0.06	0.52 ± 0.09	0.5 ± 0.06
WSC9-4 (8301)	0.539 ± 0.08	0.51 ± 0.04	0.525 ± 0.01	0.48 ± 0.01	0.47 ± 0.09
WSC9-5 (15211)	0.58 ± 0.09	$0.46 \pm 0.04 +$	0.49 ± 0.02	0.51 ± 0.02	0.42 ± 0.03

According to Table 4.1, the cluster-guided MA for distributed DWSC obtains significantly better values compared to the MA for distributed DWSC for 69 % of the tasks (Significantly better values are shown with "+" for each task). For example, for WSC8-6, the cluster-guided MA has achieved 0.56 while the simple-MA has achieved the average fitness of 0.58.

Table 4.1 also presents results for distance-guided MA. Bold values in each column indicate that the corresponding method has outperformed all the other four methods. Two bold columns for a task state that both methods equally outperform the other three methods. For example, for task WSC8-3, DistanceCR-LCS is significantly better than the four algorithms; however, on WSC8-2, both DistanceCR-LCS and DistanceCR-II significantly outperform the other three methods. This also demonstrates that there was no significant difference between DistanceCR-LCS and DistanceCR-II on WSC-8-2.

Overall, results show that utilising distance-guided crossover opera-



Figure 4.10: Mean fitness values over 30 runs for tasks WSC8-3 (left) and WSC9-4 (right). (Cluster-MA evaluation is truncated at 5000 instead of displaying up to 10,000 as the variation in values is minimal after this point.)

tors in general performed better than Simple-MA and Cluster-MA. Table 4.1 clearly demonstrates the importance of eliminating bottleneck and preserving good sub-solutions using crossover operators. Further, DistanceCR-LCS method performed the best among the five methods, because of its capability of maintaining valuable sub-solutions embedded in the LCS and simultaneously employing distance information.

Additionally, DistanceCR-II performed mostly better than DistanceCR-I. This implies that increasing the diversity between offspring chromosomes and their parents can improve the effectiveness of MAs. The combination of cluster-guided MA and distance-guided MA will be investigated in Chapter 5.

Figure 4.10 illustrates the mean fitness values over 30 runs for tasks WSC8-3 and WSC9-4, where, for both tasks, Cluster-MA outperformed Simple-MA in the early stages of the evolutionary process. Task WSC9-4 includes more services resulting in longer service sequences to be evaluated which demand more evaluations during the local search process.

4.6.4 Evaluation of Flexible Distance-guided MA

After confirming the distance-guided LCS as the most effective crossover operator among other proposed methods, in Subsection 4.6.2, it is desirable to examine distance-guided local search operators introduced in Section 4.5. Therefore, in this subsection, we use three competing algorithms: 1) DistanceCR-LCS-I, which only employs Strategy-I for building up neighbour solutions in the local search; 2) DistanceCR-LCS-II, which only employs Strategy-II for building up neighbour solutions in the local search; 3) the flexible distance-guided MA for distributed DWSC (DistanceCR-LCS-Flexible), which employs both Strategy-I and Strategy-II; and 4) the distance-guided LCS MA (DistanceCR-LCS-0), which does not use any distance-guided local search but the local search introduced in Subsection 4.3.1. In fact, DistanceCR-LCS-0 is the same as DistanceCR-LCS in Subsection 4.6.2. All of the methods employ the same mutation operator as explained in Subsection 4.3.1 and distance-guided LCS crossover operator. Therefore, the three methods only differ in their local search operator.

Neighbourhood size for local search is 20 for DistanceCR-LCS-I, DistanceCR-LCS-II and DistanceCR-LCS-Flexible; however, DistanceCR-LCS-0 does not have a fixed neighbourhood size because it depends on the length of the sequences, i.e., the number of services in the sequence. Other parameters share the settings described in Subsection 4.6.1.

Table 4.2 shows the mean solution fitness and standard deviation for the 30 independent runs of each approach. The quality of solutions produced by using DistanceCR-LCS-Flexible is generally surpassing the results of the other MAs. Further, comparing DistanceCR-LCS-I and DistanceCR-LCS-II indicates that DistanceCR-LCS-II outperforms DistanceCR-LCS-I in terms of fitness. This can be explained in the local search that DistanceCR-LCS-I uses, which is more restrictive. On the other hand, DistanceCR-LCS-Flexible has achieved the best results due to its use of flexible operators. Table 4.2: Mean fitness values and standard deviations per 30 runs. The significantly better values are shown in bold for each task (U-test with significance level of 0.05 is applied). (Note: the lower the fitness the better)

Task	DistanceCR-LCS-0	DistanceCR-LCS-I	DistanceCR-LCS-II	DistanceCR-LCS-Flexible
WSC8-1	0.41 ± 0.12	0.42 ± 0.02	0.4 ± 0.1	0.39 ± 0.03
WSC8-2	0.42 ± 0.02	0.46 ± 0.04	0.41 ± 0.01	0.41 ± 0.02
WSC8-3	0.44 ± 0.02	0.48 ± 0.02	0.44 ± 0.03	0.43 ± 0.01
WSC8-4	0.4 ± 0.01	0.41 ± 0.01	0.41 ± 0.03	0.4 ± 0.06
WSC8-5	0.46 ± 0.16	0.48 ± 0.04	0.46 ± 0.2	0.45 ± 0.22
WSC8-6	0.46 ± 0.22	0.51 ± 0.15	0.47 ± 0.03	0.47 ± 0.05
WSC8-7	0.54 ± 0.02	0.56 ± 0.02	0.53 ± 0.09	0.52 ± 0.2
WSC8-8	0.45 ± 0.05	0.48 ± 0.08	0.46 ± 0.2	0.46 ± 0.1
WSC9-1	0.53 ± 0.02	0.56 ± 0.05	0.51 ± 0.03	0.5 ± 0.02
WSC9-2	0.47 ± 0.02	0.5 ± 0.01	0.46 ± 0.2	0.45 ± 0.2
WSC9-3	0.499 ± 0.06	0.523 ± 0.001	0.48 ± 0.05	0.48 ± 0.16
WSC9-4	0.47 ± 0.09	0.51 ± 0.21	0.5 ± 0.2	0.46 ± 0.1
WSC9-5	0.42 ± 0.03	0.48 ± 0.06	0.46 ± 0.1	0.46 ± 0.12

Further, results of DistanceCR-LCS-0 in Table 4.2 show that this is the second-best MA among all competing methods. For example, on task WSC8-1, the fitness achieved by the DistanceCR-LCS-Flexible was significantly better than the other methods. For WSC8-3, there was no significant difference among all algorithms. However, according to Table 4.2, on WSC8-8, WSC9-4 and WSC9-5, DistanceCR-LCS-0 has produced better results. This is because of the large neighbourhood size of DistanceCR-LCS-0 since that local search swaps a service from all other services in the sequence and each time creates a new neighbour solution. Therefore, the sequences Neighbourhood size in this method is not fixed but depends on the number of Web services in the sequence (unlike the two other methods). For example, for task WSC9-5 the average neighbourhood size for 30 runs is 90. On the other hand, a large neighbourhood size can cause an increase in the execution time. Therefore, we have also compared methods in terms of both execution time and the results in Table 4.3. This table demonstrates that DistanceCR-LCS-0 requires extra execution time compared to other methods.

Task	DistanceCR-LCS-0	DistanceCR-LCS-I	DistanceCR-LCS-II	DistanceCR-LCS-Flexible
WSC8-1	0.5 ± 0.06	0.42 ± 0.12	0.45 ± 0.06	0.43 ± 0.03
WSC8-2	0.57 ± 0.13	0.42 ± 0.14	0.42 ± 0.02	0.4 ± 0.02
WSC8-3	1.32 ± 0.07	1.1 ± 0.42	1.2 ± 0.04	1.2 ± 0.02
WSC8-4	0.7 ± 0.02	0.71 ± 0.01	0.73 ± 0.04	0.71 ± 0.02
WSC8-5	0.98 ± 0.11	1.11 ± 0.04	1.01 ± 0.35	1.02 ± 0.36
WSC8-6	5.52 ± 0.49	4.1 ± 0.15	4.45 ± 0.15	4.45 ± 0.17
WSC8-7	2.64 ± 0.23	2.16 ± 0.02	2.16 ± 0.05	2.2 ± 0.19
WSC8-8	5.68 ± 0.48	5.13 ± 0.31	5.4 ± 0.2	5.18 ± 0.12
WSC9-1	0.46 ± 0.01	0.45 ± 0.02	0.46 ± 0.09	0.44 ± 0.01
WSC9-2	4.18 ± 0.32	4.02 ± 0.01	4.1 ± 0.2	4.02 ± 0.02
WSC9-3	4.32 ± 0.2	4.32 ± 0.001	4.15 ± 0.3	4.09 ± 0.02
WSC9-4	29.85 ± 0.48	4.8 ± 0.21	5.1 ± 0.3	4.04 ± 0.39
WSC9-5	6.14 ± 0.19	4.65 ± 0.36	4.2 ± 0.1	4.15 ± 0.12

Table 4.3: Mean execution time [seconds] for each approach per 30 runs.

Table 4.4: Average improvements over 30 runs, made by each local search strategy in DistanceCR-LCS-Flexible .

Task	Local Search	Local Search	No
	Strategy-I	Strategy-II	Improvements
WSC8-1	4%	52%	44%
WSC8-2	1%	47%	52%
WSC8-3	3%	51%	46%
WSC8-4	2%	39%	59%
WSC8-5	4%	47%	49%
WSC8-6	3%	50%	47%
WSC8-7	1%	39%	60%
WSC8-8	1%	44%	55%
WSC9-1	4%	40%	56%
WSC9-2	3%	35%	62%
WSC9-3	3%	56%	41%
WSC9-4	4%	42%	54%
WSC9-5	0%	50%	50%

Discussions on the Flexible Distance-Guided MA

Our experimental evaluations in Tables 4.2 and 4.3 show that DistanceCR-LCS-Flexible produces better results than other existing methods.

Additionally, the effectiveness of the two local search strategies (Strategy-I and Strategy-II) in DistanceCR-LCS-Flexible has been compared in Table 4.4 in terms of the number of improvements they made. According to Table 4.4, Strategy-II has made significantly more improvements than Strategy-I. This demonstrates that it has been able to produce



Figure 4.11: Mean fitness values for WSC8-3 (left) and WSC9-4 (right) (Note: the smaller the fitness the better)

enhanced neighbours for the majority of the time. For example, for WSC8-1, in 47% of local recalls, improvements have been made by Strategy-II. Only over 4% of the cases the local search Strategy-I has made improvements. In 51% of cases, neighbour solutions outperformed the current solution. Strategy-II does not restrict itself to include the Web service in the solution which serves as the source node of the longest communication link. If the Web service is located on a very distant place, including it in the composition can prevent the actual improvement in the solution.

Unlike DistanceCR-LCS-0, DistanceCR-LCS-I, DistanceCR-LCS-II and DistanceCR-LCS-Flexible employ the distance information as the heuristic in local search. Moreover, according to Table 4.2, DistanceCR-LCS-0 outperforms DistanceCR-LCS-I in most cases. The neighbourhood size for local search in the DistanceCR-LCS-0 is large, which leads to a good performance at the cost of execution time. Meanwhile, DistanceCR-LCS-II, which combines Strategy-I and Strategy-II strategies, outperforms both DistanceCR-LCS-0 and DistanceCR-LCS-I. Additionally, DistanceCR-LCS-I has a shorter execution time than DistanceCR-LCS-0. WSC8-3 and WSC9-4 have been used to compare all methods in terms of their speed of convergence as shown in Figure 4.11.

An example solution for WSC8-2 before and after undergoing local search Strategy-II is illustrated in Figure 4.12. The Web service S_9 was replaced by the combination of two services, S_{10} and S_{11} . This shows that the



Figure 4.12: Example of a solution (a) before (b) after, applying local search strategy Strategy-II.

link between S_9 and S_1 in Figure 4.12(a) has been identified as the longest communication link in the composite service. Although the number of services has grown in Figure 4.12(b) after applying local search, the total communication cost and time are improved. Therefore, this local search has led to an improvement in fitness value.

4.7 Chapter Summary

In this chapter, we proposed effective MAs for the single-objective distributed data-intensive Web service composition problem. The location information was used for a variety of purposes. For example, for generating initial population in cluster-guided MA in Section 4.3, constructing effective crossover operators in distance-guided MA in Section 4.4, or designing problem-specific local search strategies in DistanceCR-LCS-0 in Section 4.5.

In the cluster-guided MA, the initial population, based on which the rest solutions are generated, is carefully designed. That is, instead of blindly generating a set of random initial solutions, the location information is used to design a sensible initial population. In this way, the communication time and cost among services is reduced, which in turn, helps to improve the overall QoS value of the composite service. Experimental evaluations showed that the cluster-guided MA outperforms traditional MA. Particularly for larger datasets, this method was more effective since it is a problem-specific method developed particularly for a distributed environment.

For the distance-guided MA, we proposed novel crossover operators. Those crossover operators (i.e., distance-guided crossover operators) were designed to utilise the location information from the problem. The idea behind using them was to eliminate long communication links from a composite service. Further, one crossover operator employed a heuristic to preserve decent sub-solutions from the parents. This crossover operator is called distance-guided LCS crossover. Experimental evaluations confirmed that the MA when combined with the distance-guided LCS operator outperforms other techniques for the distributed DWSC. This crossover operator will then be used all through this thesis.

Finally, two local search techniques were proposed using domainknowledge of the distributed DWSC. They replaced the long communication link along with its services with another Web service in the repository. Subsequently, in the DistanceCR-LCS-Flexible was designed to employ those local search strategies. Our experimental evaluation using benchmark datasets showed that the DistanceCR-LCS-Flexible outperforms other existing MAs when applied to the distributed DWSC problem. This high performance is presumably due to its higher flexibility which allows a path replacement in the composition graph.

In the next chapter, we will expand our study to further improve our proposed algorithm. We will consider the deliberate selection of individual solutions for the local search. We will also study combining the clusterguided and distance-guided MAs.

Chapter 5

Adaptive Memetic Algorithm with Priority-based Selection for DWSC

In this chapter, we address the problem of adaptively selecting solutions for the local search for the single-objective distributed DWSC. We propose Memetic Algorithms (MAs), which employ a new priority measure for effectively selecting solutions.

5.1 Introduction

This chapter aims to develop a new measure called *SelectionPriority* (see Equation (5.5)) for priority-based selection of individuals for local search. Based on *SelectionPriority*, we further develop memetic algorithms for distributed DWSC.

To fulfil this goal, we propose a novel framework for adaptive use of local search during the global search process regarding the population, which is widely applicable to many MAs for the distributed DWSC. For example, it can be integrated with cluster-guided and distance-guided MAs (see Chapter 4) to significantly improve their performance. The following objectives will be considered in this chapter:

- 1. Two new measures, *novelty* and *improvability*, are proposed to prioritise solutions in a population. Specifically, the *novelty* is measured by comparing statistical values over all communication links in a solution with other archived solutions. The *improvability* is measured by the maximum possible improvement in fitness (the execution time and cost of a composite service) achievable by removing the *bottleneck link* in the solution.
- 2. To enhance the most widely used fitness-driven selection methods for local search, a novel priority-based selection method (i.e., *SelectionPriority*) is proposed. In *SelectionPriority*, we jointly consider the solution, population and problem-related information to select appropriate solutions, adaptively. To this aim, we use the *novelty, improvability* and fitness measures (fitness is obtained through the weighted sum of the execution cost and time of the composite service). The priority of a solution depends not only on its fitness but also on its similarity to other solutions. Therefore, we adaptively adjust the probability of selecting any solution based on the population. No existing work has ever studied the use of similar measures to guide the local search. Furthermore, this selection method is widely applicable to many memetic algorithms (MAs).
- 3. Cutting-edge adaptive MAs for DDWSC (i.e., adaptive clusterguided, adaptive distance-guided and adaptive Combined clusterdistance-guided MAs (Com-C-D)) are developed by integrating recent MAs for DDWSC with our priority-based selection method. Empirical experiments show that our proposed MAs significantly outperform existing methods for DDWSC problems, including the abstraction refinement method [29], cluster-guided MA [157] (Section 4.3) and flexible distance-guided MA [158] (Section 4.5), as

well as their combined version. Additionally, empirical evidence strongly demonstrates that both novelty and improvability measures play irreplaceable roles in *SelectionPriority* and must be used jointly to achieve high performance. We show that integrating with *SelectionPriority* helps MAs further enhance their effectiveness.

5.2 Chapter Organisation

The remainder of this chapter is organised as follows: An overview of the adaptive MA for the distributed DWSC is provided in Section 5.3. Afterwards, the priority-based selection method is presented in Section 5.4. Subsequently, evaluations and experiments are provided in Section 5.5. Section 5.6 summarises this chapter.

5.3 Overview of Adaptive MA

In this section, we present an overview of proposed adaptive MAs for distributed DWSC, which aims to optimise the objective function defined in Section 3.3.

We propose to use *SelectionPriority* to decide the chance of each individual solution to be selected for the local search.

The pseudocode of the proposed MA methods for distributed DWSC is given in Algorithm 9. At line 3, the decoding process is performed (see Subsection 2.3), after which solutions are evaluated by the fitness function (Section 3.3). Crossover and mutation operators are applied to individual solutions chosen by a tournament selection (lines 5-8). The *SelectionPriority* is calculated for each individual, which is used by the local search (line 9). Calculating *SelectionPriority* will be explained in detail in Subsection 5.4. The *SelectionPriority* is then used to select individuals for the local search (line 10). Finally, a local search operator is applied to the selected individual. We consider two different local search

operators employed in cluster-guided and distance-guided MAs (for details see Sections 4.3 and 4.5, respectively). The above steps are repeated until the predefined number of generations is reached.

5.4 Priority-based Selection method for Local Search

Through adaptively selecting individuals for local search, we direct the evolutionary search towards regions of the search space in MAs, where more interesting compromise solutions can be found. Some important factors need to be taken into account to achieve this goal: 1) solutions that increase diversity, 2) solutions that lead to better fitness, and 3) solutions that can be easily improved during local search.

One way to approach desirable diversity is to employ a diversity-based method. For example, the method might select solutions that are far apart from each other. Inspired by the *novelty search* [108], we take solutions *novelty* into account. The *novelty* represents the degree to which a solution is dissimilar to existing solutions. The definition of *novelty* varies from one problem to another and remains an important yet difficult task to be determined. Although *novelty search* only prioritises solutions that are behaviorally different from the rest of the population, we employ this characteristic along with the objective-based measure. The aim is to purposefully drive search to the most ambitious objectives for identifying which solutions or individuals to choose for the neighbourhood search. To enable the explicit use of domain-knowledge, we define our *novelty* measure based on the communication links of composite services.

Further, in this chapter, we go beyond *novelty* and consider another method of biased selection for local search. We define the *improvability* to determine the likelihood that a local search will improve the solution. This requires us to incorporate knowledge regarding when a problem-specific Algorithm 9 Adaptive Memetic Algorithm for distributed DWSC **INPUT:** Composition Task (\mathcal{T}) , Service Repository $(S\mathcal{R})$ **OUTPUT:** A Service Composition Solution 1: Generate *N* permutations of services in \mathcal{R} , as the initial population; 2: while the number of iterations not reached do Decode sequences into composite services and evaluate them; 3: 4: Find elite solutions and copy them to the next generation; 5: Select solutions using a fitness-based tournament selection; 6: Apply crossover operator to generate N offspring solutions; 7: Select five percent of the offspring solutions; Apply mutation operator to the selected offspring solutions; 8: 9: Calculate *SelectionPriority* of all offspring solutions; 10: Select individuals using the priority-based tournament selection (which employs SelectionPriority); 11: Apply local search; 12: end while return Individual With Best Fitness

local search operator is most effective. For example, this can be implemented by including the length of communication links in the composite service. This information can be used as the estimate of future effectiveness, along with the *novelty*. Therefore, less crowded regions in the search space can be adaptively cultivated and emphasised by controlling the local search based on the fitness function and communication links. *novelty* and *improvability* are explained in the following subsections.

5.4.1 Novelty of Individual Solutions

To direct the search toward any unexplored part of the solution space and maintain a high solution diversity in the GA population, we introduce a new diversity measure. A solution is considered for the local search if it is more novel than the effective solutions achieved so far. For this reason, an archive of previously discovered good solutions (regarding fitness values) must be constantly maintained. The archive is updated at each generation to include the fittest solutions (the number of solutions depends on the archive size). For each solution, the archive stores the statistics of communication links (e.g., how often they appear in the DAG). A vector, called *freq*, is generated to store the frequency of communication links (*freq* is calculated for any new composite service CS). The same communication link may appear multiple times in one composite service.

Example 3. Figure 5.1 shows a composite service, CS, and the corresponding freq vector. Suppose that four servers, i.e., h_1 , h_2 , h_3 and h_4 , are used to host all the nine services contained in CS. The frequency of each communication link is shown in Fig. 4 (b). For example, the communication link between h_2 and h_3 has been used three times: (S_9, S_{13}) , (S_{13}, S_6) and (S_{13}, S_7) . The freq vector of CS is compared to the freq vector of all archived composite services one at a time, based on their Euclidean distance, which is one of the simplest and most commonly used measures of similarity [207, 217].

Subsequently, the average Euclidean distance of CS to its k nearest neighbours in the archive is calculated as *novelty*, as shown in Equation (5.1):



1

	_		_	(a)				_	-	
Servers	h_1, h_1	h_1, h_2	h_1, h_3	h_1, h_4	h_2, h_2	h_2, h_3	h_2, h_4	h_3, h_3	h_3, h_4	h_4,h_4
Links	-	$\begin{array}{c}(S_1,S_6)(S_9,S_1)\\(S_9,S_3)(S_5,S_8)\\(S_9,S_5)(S_5,S_8)\\(S_5,S_7)\end{array}$	-	-	(S_6, S_7) (S_8, S_7)	$(S_9, S_{13}) \ (S_{13}, S_6) \ (S_{13}, S_7)$	$(S_9, S_{16}) \ (S_{16}, S_8)$	-	-	-
Frequency	0	7	0	0	2	3	2	0	0	0
(b)										

Figure 5.1: Example of the novelty measure: (a) a composite service, *CS*, (top right) and the location of services (top left), (b) *freq* vector.

$$Novelty(\mathcal{CS}) = (1/k) \sum_{i=1}^{k} \mathcal{D}(freq(\mathcal{CS}), freq(\mathcal{CS}_i))$$
(5.1)

where CS_i is an archived solution evolved previously. CS_i is the i_{th} nearest neighbour of CS.

A solution with a higher *novelty* has a higher priority for the neighbourhood search. The main rationale behind utilising this distance metric is to generate new genotypes and put more effort into exploiting the neighbourhood of novel solutions to avoid premature convergence. Additionally, this novelty measure facilitates incorporating problem-specific knowledge in the selection method.

5.4.2 Improvability of Individual Solutions

The second challenge in selecting individuals for local search is how to find solutions that can be easily improved. Because fitness evaluation is computationally expensive, it is not worth performing a local search if a better neighbouring solution cannot be found through the timeconsuming neighbourhood search.

we introduce a measure for the priority-based selection method. We utilise domain-knowledge to find individuals that have the potential to be enhanced during the local search, through the *improvability* measure. The *improvability* measure is developed to match the local search for the flexible distance-guided MA (see Section 4.5).

It is desirable to apply local search on solutions with diverse communication links and high average communication distance. In other words, when the solution includes links with significantly different lengths, the links are diversified, and services are farther from each other. This is also to be taken into account for individual selection.

To consider the length diversity of communication links in a composite service, standard deviation and mean of the lengths are calculated for each composite solutions. Afterwards, one z_value for the mean and another for the standard deviation are calculated. z_value is a numerical measurement used in statistics to determine a value's relationship to the average or standard deviation of the population. The z_value of deviations is obtained using Equation (5.2):

$$z_value(\sigma) = \frac{\sigma_{cs} - \mu_{\sigma}}{\sigma_{\sigma}}$$
(5.2)

where $z_value(\sigma)$ is the z_value of the deviation of the current solution to determine how different its deviation is from the deviation of the population. σ_{CS} represents the deviation of the current solution, σ_{σ} and μ_{σ} are the mean and standard deviation of all solution's deviations in the population.

Similarly, to determine the difference of the current solution's mean value and the population's mean value, we calculate the z_value for the mean, as in Equation (5.3):

$$z_value(\mu) = \frac{\mu_{CS} - \mu_{\mu}}{\sigma_{\mu}}$$
(5.3)

where $z_value(\mu)$ is the z_value for the mean of the current solution's communication links length (μ_{CS} is the mean of its communication links length), μ_{μ} and σ_{μ} are the standard deviations and the mean over all solution's mean in the population. Note that this measure takes the diversity in the length of all links into account. The calculation of z_value is exemplified in Example 4.

CS1		CS2		CS3	
Communication link	Length	Communication link	Length	Communication link	Length
l_1	0.5	l_1	0.3	l_1	0.7
l_2	0.8	l_2	0.9	l_2	0.5
l_3	0.1	l_3	0.7	l_3	0.4
l_4	0.2	l_4	0.6	l_4	0.5
$\mu_{CS} = 0.4$	4	l_5	0.8	l_5	0.3
$\sigma_{CS} = 0.3162$		μ _{CS} =0.66		$\mu_{CS}=0.48$	
		$\sigma_{CS}=0.$	23	$\sigma_{CS} = 0.148$	

Figure 5.2: Length of communication links for three composite services, with averages and standard deviations.

Example 4. Consider a population of three composite services, as shown in Fig. 5.2. The links and their lengths are shown in the associated table for each composite service. The last two columns demonstrate the average and standard deviation of the link's length for each composite service. According to the value in Fig. 5.2, we can obtain means and standard deviations as follows: $\mu_{\mu} = 0.513$ (the mean of 0.4, 0.66 and 0.48), $\sigma_{\mu} = 0.133$ (the standard deviation of 0.4, 0.66 and 0.48), $\sigma_{\sigma} = 0.23$ (the mean of 0.3162, 0.23 and 0.148), $\sigma_{\sigma} = 0.084$ (the standard deviation of 0.3162, 0.23 and 0.148). Therefore, the *z*_value for CS2, for example, is calculated as shown below:

$$z_value(\mu) = \frac{0.66 - 0.513}{0.133} = 1.1$$

$$z_value(\sigma) = \frac{0.23 - 0.23}{0.084} = 0$$

Finally, the *improvability* of CS is calculated as the sum of two z_values :

$$Imp(\mathcal{CS}) = z_value(\mu) + z_value(\sigma)$$
(5.4)

A higher value of Imp(CS) represents a higher average length and deviation of communication links than the population.

5.4.3 Selection Criteria for Local Search

Finally, a selection priority of CS for the local search is obtained using Equation (5.5):

$$SelectionPriority(CS) = w_1 \widehat{Novelty}(CS) + w_2 \widehat{Imp}(CS) + (5.5) + w_3 F(CS)$$

where F(CS) is defined by Equation (3.7) (Section 3.3), and $\widehat{Novelty}$ and \widehat{Imp} are normalised values of Novelty(CS) and Imp(CS), respectively. w_1 , w_2 and w_3 are integer weight parameters in $\{0, 1\}$, where $w_1 + w_2 + w_3 \neq 0$. These parameters are changed during the experiments to show the contribution of each measure to the performance of MAs. If a weight parameter is zero, the corresponding criterion does not contribute to the selection priority.

5.5 Evaluations

In this section, a series of experiments are performed to answer the following research questions (RQ):

- RQ1: Can the proposed MAs in this chapter, including adaptive cluster-guided, adaptive distance-guided, and adaptive *Combined Cluster-Distance-guided* (*Com-C-D*) MAs, outperform other MAs for distributed DWSC?
- RQ2: Which of the novelty, improvability and fitness measures play a more important role in MAs?
- RQ3: Does the improvement in fitness value of solutions found by our algorithms lead to the enhancement in both response time and cost?
- RQ4: What are the computation time of each method?

These research questions are addressed in Subsections 5.5.3, 5.5.4, 5.5.5, and 5.5.6, respectively.

5.5.1 Competing Methods

Four methods are used as the competing methods: 1) cluster-guided MA (Section 4.3); 2) flexible-distance-guided MA (Section 4.5); 3) *Com*-

bined Cluster-Distance-guided MA that combines the cluster-guided and the distance-guided approaches. This algorithm is named (*Com-C-D*) where the cluster-guided approach helps the GA find a better quality initial population while the distance-guided approach improves the fitness of evolved solutions by using distance-guided crossover and local search operators; and 4) abstraction refinement method [29], which is an existing state-of-the-art method (introduced in Section 2.2, Chapter 2). We make a few modifications to the abstraction method to make it suitable for the distributed DWSC. Abstraction refinement is chosen mainly since it is one of the recently proposed algorithms for fully-automated WSC.

We also integrate the proposed priority-based selection method with the above three MAs (cluster-guided, flexible distance-guided and combined-cluster-distance MAs), which results in adaptive distanceguided MA, adaptive cluster-guided MA and adaptive Com-C-D MA.

5.5.2 Parameter Settings

All MAs (adaptive and non-adaptive) share the same parameter settings as summarised in Table 5.1, Column 3. After deciding on the operator, a conventional fitness-driven tournament selection with size two chooses candidate solutions for the mutation and crossover operators. Additionally, as shown in Table 5.1, all adaptive MAs use tournament size two over the *SelectionPriority*, which is updated in each generation, archive size of 10 and neighbourhood size of three for calculating *novelty*.

We have experimentally evaluated the abstraction refinement method introduced in [29] using the test dataset WSC-2009. Our WSC test problem is the same as the distributed DWSC except that we do not consider the communication characteristics and data characteristics in the objective function. In fact, while evaluating the abstraction refinement method, we assume that all services are located close to each other. Thereafter, we adopt it for distributed DWSC.

Parameter	Value	Methods
Generations	100	All MAs
Elitism size (e)	2	All MAs
Population Size	30	All MAs
Selection (for mutation and crossover)	Tournament (size=2)	All MAs
Crossover Probability (p_c)	0.95	All MAs
Data size	3	All MAs
Local Search Probability	0.05	All MAs
Mutation Probability (p_m)	0.05	All MAs
SelectionPriority-based tournament size	2	Adaptive MAs
k (Neighbours)	3	Adaptive MAs
Archive Size	10	Adaptive MAs

Table 5.1: Common parameter settings for adaptive MAs

Similar to Chapter 4, we employ the datasets set up in Chapter 3 for empirical evaluations, which include bandwidth value, data, and location information of services.

5.5.3 Evaluation of the Priority-based Selection Method (RQ1)

To answer the first research question, RQ1, and to investigate the effectiveness of the new priority-based selection method, we compare our MAs against existing methods.

Results on distributed DWSC are shown in Table 5.2 and compared to (adaptive) cluster-guided and (adaptive) distance-guided MAs. A pairwise one-sample statistical t-test at 0.05 significance level (p-value = 0.05) was performed between each pair of algorithms in Table 5.2.

Results demonstrate that adaptive MAs outperform the abstraction refinement method on all tasks on distributed DWSC. For example, for WSC8-1 adaptive distance-guided MA is significantly better than the abstraction refinement method. The cluster-guided method, however, does not significantly outperform the abstraction refinement method on WSC8-1. The abstraction refinement and cluster-guided methods focus on grouping Web services according to only one aspect of their quality. The former groups Web services based on their functional characteristics and then selects the one with the best QoS, while the latter clusters Web services considering locations of Web services exclusively and prefers those Web services located close to other Web service used for the composition.

Table 5.2 shows that Com-C-D outperforms both cluster-guided and distance-guided MAs. We further apply our selection method to Com-C-D. As shown in Table 5.2, considering Com-C-D and adaptive distance-guided MA, Com-C-D shows higher performance on four tasks, WSC8-6, WSC8-7, WSC9-1 and WSC9-3. Additionally, adaptive Com-C-D indicates significantly higher performance than all other algorithms on all tasks. These results indicate that our proposed priority-based selection method can be applied to many existing MAs for distributed DWSC and improve their performance.

Table 5.2: Comparison of the abstraction refinement for DWSC, (adaptive) distance-guided, (adaptive) cluster-guided, and (adaptive) Com-C-D MAs. (Note: the lower the fitness the better). All three factors have been considered in the selection priority. Significantly better values are shown in bold for each task, with the significance level of 0.05.

Task	abstraction ref. [29]	cluster -guided MA	distance -guided MA	adaptive cluster-guided MA	adaptive distance-guided MA	Com-C-D MA	adaptive Com-C-D MA
WSC8-1	0.45	0.45 ± 0.01	0.39 ± 0.03	0.42 ± 0.02	0.36 ± 0.02	0.36 ± 0.09	0.35 ± 0.01
WSC8-2	0.42	0.43 ± 0.01	0.41 ± 0.02	0.4 ± 0.04	0.36 ± 0.05	0.37 ± 0.15	0.36 ± 0.04
WSC8-3	0.49	0.52 ± 0.06	0.44 ± 0.01	0.44 ± 0.13	0.41 ± 0.05	0.43 ± 0.02	0.41 ± 0.07
WSC8-4	0.51	0.5 ± 0.02	0.39 ± 0.06	0.46 ± 0.02	0.36 ± 0.09	0.36 ± 0.12	0.35 ± 0.08
WSC8-5	0.52	0.53 ± 0.03	0.45 ± 0.22	0.49 ± 0.06	0.41 ± 0.01	0.44 ± 0.07	0.4 ± 0.01
WSC8-6	0.56	0.56 ± 0.1	0.47 ± 0.05	0.51 ± 0.1	0.44 ± 0.14	0.43 ± 0.05	0.42 ± 0.15
WSC8-7	0.55	0.55 ± 0.27	0.52 ± 0.02	0.51 ± 0.23	0.49 ± 0.1	0.47 ± 0.1	0.46 ± 0.1
WSC8-8	0.48	0.49 ± 0.15	0.46 ± 0.1	0.45 ± 0.09	0.41 ± 0.09	0.43 ± 0.11	0.4 ± 0.2
WSC9-1	0.54	0.54 ± 0.21	0.5 ± 0.02	0.48 ± 0.02	0.46 ± 0.16	0.45 ± 0.12	0.46 ± 0.02
WSC9-2	0.52	0.51 ± 0.2	0.45 ± 0.2	0.48 ± 0.2	0.41 ± 0.13	0.41 ± 0.17	0.4 ± 0.15
WSC9-3	0.53	0.54 ± 0.2	0.48 ± 0.16	0.5 ± 0.16	0.45 ± 0.17	0.42 ± 0.18	0.41 ± 0.2
WSC9-4	0.51	0.5 ± 0.05	0.48 ± 0.1	0.46 ± 0.2	0.43 ± 0.2	0.45 ± 0.12	0.43 ± 0.2
WSC9-5	0.48	0.46 ± 0.1	0.46 ± 0.12	0.45 ± 0.2	0.42 ± 0.16	0.43 ± 0.2	0.42 ± 0.2

5.5.4 Evaluation of Different Selection Criteria (RQ2)

To answer the second research question, RQ2, we examine the influence of different selection criteria. To examine how MAs can benefit from considering improvability and novelty, the effectiveness of each measure in the *SelectionPriority* in Equation (5.5) are assessed separately.

Table 5.3: Mean and standard deviation of fitness values per 30 runs for distance-guided MA with different selection parameters. Significantly better values are shown in bold for each task (Note: the lower the fitness the better).

Task	$w_1 = w_2 = 0, w_3 = 1$	$w_1 = 0, w_2 = w_3 = 1$	$w_2 = 0, w_1 = w_3 = 1$	$w_1 = 1, w_2 = w_3 = 0$	$w_1 = w_2 = w_3 = 1$
WSC8-1	0.39 ± 0.03	0.38 ± 0.12	0.36 ± 0.11	0.39 ± 0.18	0.36 ± 0.11
WSC8-2	0.41 ± 0.02	0.39 ± 0.08	0.37 ± 0.06	0.39 ± 0.08	0.36 ± 0.04
WSC8-3	0.44 ± 0.01	0.43 ± 0.02	0.41 ± 0.02	0.42 ± 0.03	0.41 ± 0.08
WSC8-4	0.39 ± 0.06	0.38 ± 0.04	0.36 ± 0.05	0.37 ± 0.12	0.36 ± 0.14
WSC8-5	0.45 ± 0.22	0.41 ± 0.1	0.41 ± 0.15	0.44 ± 0.09	0.4 ± 0.14
WSC8-6	0.47 ± 0.06	0.46 ± 0.12	0.45 ± 0.05	0.45 ± 0.1	0.43 ± 0.09
WSC8-7	0.52 ± 0.02	0.51 ± 0.17	0.48 ± 0.07	0.5 ± 0.1	0.48 ± 0.1
WSC8-8	0.46 ± 0.01	0.45 ± 0.08	0.44 ± 0.2	0.45 ± 0.02	0.41 ± 0.13
WSC9-1	0.5 ± 0.02	0.49 ± 0.04	0.48 ± 0.02	0.49 ± 0.04	0.47 ± 0.1
WSC9-2	0.45 ± 0.02	0.43 ± 0.12	0.41 ± 0.2	0.44 ± 0.1	0.4 ± 0.8
WSC9-3	0.48 ± 0.16	0.46 ± 0.06	0.46 ± 0.04	0.48 ± 0.01	0.45 ± 0.2
WSC9-4	0.48 ± 0.01	0.47 ± 0.1	0.47 ± 0.08	0.48 ± 0.09	0.43 ± 0.14
WSC9-5	0.46 ± 0.12	0.47 ± 0.1	0.46 ± 0.15	0.48 ± 0.09	0.45 ± 0.9

Table 5.4: Mean fitness values and standard deviations per 30 runs for the cluster-guided MA with different selection parameters. Significantly better values are shown in bold. The significance level is 0.05 (Note: the lower the fitness the better).

Task	$w_1 = w_2 = 0, w_3 = 1$	$w_1 = 0, w_2 = w_3 = 1$	$w_2 = 0, w_1 = w_3 = 1$	$w_1 = 1, w_2 = w_3 = 0$	$w_1 = w_2 = w_3 = 1$
WSC8-1	0.45 ± 0.01	0.45 ± 0.02	0.42 ± 0.2	0.44 ± 0.18	0.42 ± 0.09
WSC8-2	0.43 ± 0.01	0.42 ± 0.1	0.39 ± 0.1	0.41 ± 0.07	0.4 ± 0.07
WSC8-3	0.52 ± 0.06	0.52 ± 0.08	0.46 ± 0.1	0.48 ± 0.1	0.45 ± 0.06
WSC8-4	0.5 ± 0.09	0.49 ± 0.01	0.48 ± 0.01	0.49 ± 0.02	0.46 ± 0.03
WSC8-5	0.53 ± 0.03	0.52 ± 0.23	0.5 ± 0.2	0.52 ± 0.1	0.49 ± 0.2
WSC8-6	0.56 ± 0.01	0.56 ± 0.09	0.51 ± 0.2	0.55 ± 0.15	0.51 ± 0.2
WSC8-7	0.55 ± 0.27	0.54 ± 0.2	0.51 ± 0.3	0.53 ± 0.3	0.51 ± 0.3
WSC8-8	0.5 ± 0.02	0.51 ± 0.04	0.49 ± 0.1	0.5 ± 0.1	0.49 ± 0.1
WSC9-1	0.54 ± 0.03	0.53 ± 0.05	0.5 ± 0.17	0.54 ± 0.1	0.5 ± 0.2
WSC9-2	0.49 ± 0.15	0.48 ± 0.05	0.44 ± 0.09	0.45 ± 0.2	0.44 ± 0.2
WSC9-3	0.54 ± 0.01	0.53 ± 0.1	0.5 ± 0.1	0.52 ± 0.22	0.5 ± 0.19
WSC9-4	0.5 ± 0.09	0.5 ± 0.09	0.47 ± 0.01	0.49 ± 0.03	0.46 ± 0.06
WSC9-5	0.46 ± 0.1	0.46 ± 0.1	0.44 ± 0.1	0.45 ± 0.1	0.43 ± 0.2

Measurements for RQ2 We evaluate the following combinations for the priority-based selection: fitness with novelty, fitness with improvability, and, fitness with novelty and improvability. We conduct experiments with five separate weight configurations for Equation (5.5) (i.e., C1 to C5). C1 indicates the initial algorithm which only employs fitness value during the selection ($w_1 = w_2 = 0$ and $w_3 = 1$). Configuration C2 implies that the contribution of novelty in the priority-based selection is zero ($w_1 = 0$ and $w_2 = w_3 = 1$), while $w_2 = 0$ and $w_1 = w_3 = 1$ in C3 mean that only novelty and fitness affect the selection process. Configuration C4 is for applying

only novelty ($w_1 = 1$ and $w_2 = w_3 = 0$). Finally, in C5, all three criteria are applied ($w_1 = w_2 = w_3 = 1$).

Results Results of the priority-based selection method integration with distance-guided and cluster-guided MAs are shown in Table 5.3 and Table 5.4, respectively. The Wilcoxon statistical test with p-value = 0.05 has been performed on the average fitness values. Priority-based selection of individuals has significantly enhanced the quality of composite services. The first column in each table represents the task. SThe second columns of tables indicate the original algorithm that relies purely on the fitness value for selecting individuals for the local search. The rest columns in both tables present results obtained by using one criterion along with the fitness value in priority-based selection. In most of the tasks, the prioritybased selection method produces better results compared to the original algorithm. For example, for WSC8-1, the fitness of the best solution found by the original algorithm is 0.4. However, when applying priority-based selection (in columns 3-6) we obtain solutions with better fitness. This applies for WSC8-4, WSC8-5, WSC8-6, WSC8-7, WSC9-1 and WSC9-5, as well. The Wilcoxon statistical test reveals that the method presented in the last column which utilises all measures for priority-based selection outperforms all other methods for task 08-1. Overall, the configurations C2, C3, C4 and C5 outperformed the original algorithms (i.e., configuration C1) on 92%, 62%, 54%, and 100% of the tasks, respectively.

For C2, we can witness a slight improvement in performance, when the improvability of a solution (Imp) is utilised along with the fitness value (F) in selecting solutions for local search. For example, on tasks WSC8-1, WSC8-2 and WSC8-5 as well as all WSC9 tasks, the final fitness value of the solution is lower (better) than the final fitness value for C1. However, for the cluster-guided MA, this improvement is not significant (i.e., C2) has not yielded to better results than C1. In other words, for the cluster-guided MA, C2 outperforms the original configuration C1 only on a small

number of tasks. For example, in task WSC8-2 the solution's fitness with C1 is 0.44 and with C2 is 0.43 which is marginally better, while in task WSC8-3 the value is the same (0.53) for both C1 and C2.

We obtain higher quality results using both *Novelty* criterion and the fitness value in configuration C3. We subsequently considered the effect of *Novelty* without *F* and *Imp*, which is in association with the fifth column (i.e., C4) in both tables. For about half of the tasks, e.g. WSC8-2, WSC8-8, WSC9-2 and WSC9-4, we obtained solutions with similar quality to the original method (column two). However, for the rest of the tasks, marginally superior solutions were achieved (supported by WSC8-1, WSC8-3, WSC8-4, etc.). On task WSC8-8 for the cluster-guided MA, the novelty was able to find even fitter solutions compared to the combined use of novelty and fitness. Comparing the configurations C2 and C3 in Table 5.3 and Table 5.4, we find out that *Novelty* (C3) plays a more important role than *Imp* (C2) in enhancing the performance of MAs.

Finally, in association with the last columns of Table 5.3 and Table 5.4, the performance using all the three criteria (i.e., *Novelty*, *Imp* and *F*) is studied for selecting individuals for the local search. An overall improvement on all tasks for distance-guided MA and on all tasks except WSC8-2 for the cluster-guided MA can be confirmed.

Figure 5.3 shows the convergence of each method for tasks WSC8-8 and WSC9-5. For example, for WSC8-8, the blue plot associated with the green plot distance-guided MA has converged to a local optimum and, therefore, failed to discover a satisfactory solution compared to other methods which are all adaptive. The blue is achieving improvements in initial generations. Overall, the blue plot which employs the combination of fitness, novelty and improvability, outperformed all other techniques.

Figure 5.4 shows the convergence of each method for tasks WSC8-8 and WSC9-5 for the cluster-guided MA. For both tasks, methods that include novelty and fitness together (i.e., blue and purple plots), outperform other methods.



Figure 5.3: Mean fitness over evolution numbers for tasks WSC8-8 (left) and WSC9-5 (right) for adaptive distance-guided MA (the lower the fitness the better).



Figure 5.4: Mean fitness over evolution numbers for tasks WSC8-8 (left) and WSC9-5 (right) for adaptive cluster-guided MA (the lower the fitness the better).

5.5.5 Evaluation of Response Time and Cost (RQ3)

To answer the third research question, RQ3, and to demonstrate the practical significance of the improvement made by our MAs, we separately examine the response time and cost besides fitness values.

We have compared the performance of different algorithms based on the weighted sum in Equation (3.7) (Section 3.3) so far, following the common practice in the literature [29, 49, 189]. The weighted sum approach provides the user with a single final solution. Therefore, unlike in multiobjective Pareto-based approaches, the user is not required to select one solution out of a possibly large number of compromising solutions in the Pareto front. However, to prove the practical significance of our method, the improvements in time and cost should be investigated separately.

In view of this, additional evaluation results regarding the response time and cost achieved by the evolved composite services are shown in Table 5.5, for distance-guided and Com-C-D MAs. These two methods are chosen because they have achieved higher performance compared to other baselines as shown in Table 5.2.

According to Table 5.5, the savings in the time and cost achieved by the proposed MAs are significant. The average improvements in response time and cost over all benchmark problems for distance-guided MA are 13% and 3%, respectively and for Com-C-D (columns four and five) are 5% and 10%, respectively. For example, comparing columns two and three of both tables for WSC8-1 shows that both response time and cost have been reduced significantly along with improvement in fitness, i.e., 16% for the response time and 26% for the cost (the response time reduces from 32251.9 ms to 27001.6 ms and the cost reduces from \$159.3 to \$103.9). For another example, comparing Com-C-D and adaptive Com-C-D on task WSC8-8, the reduction in cost and time are 19% and 6% respectively. Additionally, the case where one objective degrades in favour of the other objective has only appeared occasionally on a few tasks when fitness cannot be improved further. For example, for task WSC8-3, columns two and three have the same fitness value but the cost degrades in favour of the response time. Results confirm that the improvement in the fitness value is usually a consequence of improvements in both objectives.

Table 5.5: Mean and standard deviation of cost and response time per 30 runs for distance-guided, Com-C-D, adaptive distance-guided and adaptive Com-C-D MAs. Significantly better values are shown in bold. The significance level is 0.05.

	distance-gu	ided MA	adaptive distan	ce-guided MA	Com-C-E	O MA	adaptive Co	m-C-D MA
Task	Fitne	ess	Fitn	ess	Fitne	SS	Fitn	ess
	Time	Cost	Time	Cost	Time	Cost	Time	Cost
WSC8-1	0.39 ±	0.03	0.36 ±	0.02	0.36 ± 0	0.09	$0.35 \pm$	0.01
W3C0-1	32251.9 ± 1247	159.3 ± 21	27001.6 ± 599	103.9 ± 9	26169.14 ± 799	110.1 ± 11	26001.1 ± 809	101.6 ± 15
WSC8-2	0.41 ±	0.02	$0.36 \pm$	0.05	0.37 ± 0	0.15	$0.36 \pm$	0.04
W3C0-2	24701.5 ± 395	180.1 ± 14	18515.8 ± 365	99.6 ± 8.5	18951.98 ± 501	101.1 ± 13	17999.6 ± 998	101.9 ± 13
WSC8-3	0.44 ±	0.01	$0.41 \pm$	0.05	0.43 ± 0	0.02	$0.41 \pm$	0.07
W3C0-5	75661.65 ± 1010	219.1 ± 15	72093.1 ± 587	196.45 ± 16	75913.59 ±599	214.09 ± 17	72893.9 ± 580	192.9 ± 15
WEC9 4	0.39 ±	0.06	0.36 ±	0.09	$0.36 \pm$	0.12	$0.35 \pm$	0.08
W3C0-4	17121.54 ± 109	126.8 ± 11	12709.12 ± 114	99.9 ± 6	12569.45 ± 101	101.1 ± 7	11061.9 ± 75	95.4 ± 5
WECS 5	0.45 ±	0.22	0.41 ±	0.01	$0.44 \pm$	0.07	$0.4 \pm$	0.01
W3C8-5	34121.18 ± 312	255.45 ± 33	30911.19 ± 379	209.9 ± 16	32660.9 ± 481	234.33 ± 15	29966.1 ± 111	204.1 ± 20
WSC8-6	0.47 ± 0.01		0.44 ± 0.04		0.43 ± 0.08		$0.42 \pm$	0.15
W3C0-0	31213.19 ± 387	388.35 ± 15	27429.25 ± 263	300.14 ± 15	26901.18 ± 362	299.9 ± 16	25422.5 ± 299	281.95 ± 16
WSC8-7	$0.52 \pm$	0.02	0.49 =	± 0.1	0.47 ±	0.1	0.46	0.1
W3C0-7	34301.8 ± 299	301.23 ± 59	29496.1 ± 150	199.09 ± 19	27859.2 ± 343	192.1 ± 25	27099.8 ± 198	185.64 ± 45
WSC8-8	0.46 ±	0.01	0.41 ±	: 0.09	0.43 ± 0	0.11	$0.4 \pm$	0.2
W3C0-0	58022.74 ± 454	313.92 ± 33	50126.9 ± 329	239.13 ± 20	53319.1 ± 341	260.3 ± 40	49995.15 ± 409	211.9 ± 31
WSC9-1	0.5 ±	0.1	$0.46 \pm$	0.16	0.45 ± 0.12		$0.46 \pm$	0.02
WSC)-1	24713.7 ± 913	112.93 ± 11	20195.4 ± 629	92.1 ± 5.3	19610.15 ± 659	90.01 ± 6	19002.11 ± 491	90.5 ± 10
WSC9-2	0.45 ±	: 0.2	0.41 ±	0.13	0.41±0	0.17	$0.4 \pm$	0.15
W3CJ-2	48876.15 ± 6875	304.13 ± 36	44381.19 ± 360	214.9 ± 21	43819.9 ± 399	218.17 ± 23	42706.9 ± 6699	211.57 ± 37
WSC9-3	0.48 ±	0.16	0.45 ±	0.17	0.42 ± 0	0.18	0.41	-0.2
WSCJ-5	31760.56 ± 115	293.43 ± 39	27114.5 ± 69	204.19 ± 22	25719.5 ± 77	202.11 ± 15	24013.13 ± 119	199.08 ± 34
WSC9-4	0.48 ±	: 0.1	$0.43 \pm$	0.02	0.45 ± 0	0.12	$0.43 \pm$	0.02
110094	74340.9 ± 2314	699.49 ± 125	66439.5 ± 1309	491.13 ± 78	71117.1 ± 1599	596.31 ± 132	65042.2 ± 1211	506.74 ± 108
WSC9-5	0.46 ±	0.12	0.42 ±	0.16	0.43 ±	0.2	0.42	0.2
11009-5	66505.9 ± 990	491.89 ± 68	61650.92 ± 681	389.89 ± 29	60070.32 ± 401	406.35 ± 28	59777.92 ± 250	362.9 ± 13

Table 5.6: Mean and standard deviation of computation time (seconds) for each algorithm over 30 runs (Wilcoxon sum rank test with significance level of 0.05 is applied).

Task	abstraction ref. [29]	cluster -guided MA	distance -guided MA	adaptive cluster-guided MA	adaptive distance-guided MA	Com-C-D MA	adaptive Com-C-D MA
WSC8-1	0.4	3.7 ± 0.14	0.43 ± 0.03	4.2 ± 0.54	1.19 ± 0.15	4.6 ± 0.03	4.9 ± 0.04
WSC8-2	0.33	3.6 ± 0.12	0.4 ± 0.03	4.11 ± 0.5	1.2 ± 0.04	4.36 ± 0.6	5.02 ± 0.08
WSC8-3	1.23	7.13 ± 0.83	1.28 ± 0.02	7.49 ± 1.01	2.1 ± 0.14	8.08 ± 1.03	8.37 ± 1.07
WSC8-4	1.01	5.5 ± 0.09	0.74 ± 0.02	6.01 ± 0.14	2.46 ± 1.03	6.64 ± 0.22	6.87 ± 0.3
WSC8-5	2.12	7.63 ± 2.1	1.02 ± 0.07	7.94 ± 1.04	1.9 ± 0.39	8.75 ± 0.9	8.94 ± 1.02
WSC8-6	4.66	11.56 ± 2.08	4.45 ± 0.17	11.94 ± 2.1	5.91 ± 1.12	12.86 ± 3.3	12.95 ± 3.4
WSC8-7	3.28	7.15 ± 1.21	2.42 ± 0.19	8.52 ± 0.8	3.51 ± 0.09	9.94 ± 1.2	10.23 ± 1.19
WSC8-8	4.98	9.49 ± 1.04	5.18 ± 0.12	10.63 ± 0.07	6.95 ± 1.16	13.24 ± 1.3	13.86 ± 1.4
WSC9-1	0.97	4.98 ± 1.02	0.44 ± 0.02	6.02 ± 0.2	0.98 ± 0.18	6.43 ± 0.3	7.03 ± 0.32
WSC9-2	3.98	6.95 ± 0.7	4.05 ± 0.02	7.08 ± 0.12	4.39 ± 0.09	7.58 ± 0.19	7.78 ± 0.18
WSC9-3	4.45	8.54 ± 1.1	4.08 ± 0.03	5.98 ± 1.06	5.05 ± 0.6	6.71 ± 2.1	7.02 ± 1.9
WSC9-4	3.01	12.5 ± 3.11	4.16 ± 0.09	14.8 ± 0.73	4.95 ± 0.04	15.9 ± 1.4	16.65 ± 1.7
WSC9-5	2.99	13.68 ± 4.2	4.15 ± 0.12	15.02 ± 1.08	5.92 ± 1.0	15.73 ± 1.36	16.13 ± 1.47

5.5.6 Evaluation of Computation Time (RQ4)

To answer RQ4, we compare all algorithms in term of their execution time. Table 5.6 indicates that the abstraction refinement [29] is the fastest algorithm.

For example, it requires only 0.33 seconds to provide solutions for 08-2. The cluster-guided MA is always slower than the distance-guided MA. For example, the cluster-guided MA spends 3.6 seconds while the distance-guided MA requires only 0.4 seconds on 08-2. Therefore, the combination of cluster-guided MA and distance-guided MA (i.e., Com-C-D MA) requires slightly more computation time than the cluster-guided MA and substantially more time than the distance-guided MA. All adaptive MAs take slightly more computation time than their non-adaptive version. Finally, adaptive cluster-guided and adaptive distance-guided MAs are faster than Com-C-D MA.

5.6 Chapter Summary

In this chapter, we proposed a priority-based selection method to select individuals for local search in MAs. Our proposed method not only adopted the fitness value as a selection criterion but also the two novel measures (i.e., the solution novelty and improvability). We proposed methods to carefully calculate the novelty and the improvability. We then defined the selection priority using the fitness value, the improvability and the novelty measures. The fitness component of *SelectionPriority* ensures that good solutions still maintain a reasonable chance for local search.

For preventing premature convergence of MAs to a local optimum, it is essential to be able to identify if the GA is converging to an optimum. One possible way of detecting convergence is to observe novelty concerning the solutions in the archive. Solutions with lower fitness but higher novelty still have a chance to be improved by local search. While the high fitness solutions aid in the convergence of the GA, the low fitness solutions prevent the MA from getting stuck at a local optimum.

Moreover, the improvability of a solution can be taken into account while selecting a solution for the neighbourhood search.

We hybridised all MAs with *SelectionPriority* and results showed that MAs integrated with *SelectionPriority* (i.e., adaptive MAs) can scale up a wide range of MAs with conventional fitness-based selection.

We also examined different components of *SelectionPriority* separately. Results confirmed that including all components together is superior compared to other combinations of the components.

In conclusion, we conducted in-depth experiments to answer multiple research questions. Experimental results clearly showed that our proposed priority-driven selection method can significantly enhance the performance of some cutting-edge MAs for DWSC. First, we discovered that combining the two methods, (i.e., cluster-guided and distanceguided MAs) results in an algorithm with a higher performance than the two methods (i.e., Com-C-D). Additionally, our developed adaptive MAs in this chapter, including adaptive cluster-guided, adaptive distanceguided and adaptive Combined Cluster-Distance-guided MAs (Com-C-D) outperformed cluster-guided and distance-guided MAs for distributed DWSC. In particular, we discovered that both the novelty and improvability measures play irreplaceable roles in the SelectionPriority. They jointly enable our MAs to substantially outperform several algorithms on a range of benchmark DWSC problems. However, careful examinations disclosed that the novelty measure plays a more important role than the improvability in enhancing the performance of MAs. Finally, the results confirm that the improvement in the fitness value is usually the result of improvements in both objectives (i.e., cost and time). This indicates that optimising the weighted sum of the objectives as the single objective of our EC algorithms does not degrade the performance.

118CHAPTER 5. ADAPTIVE SINGLE-OBJECTIVE DISTRIBUTED DWSC
Chapter 6

Multi-Objective Distributed DWSC Methods

In previous chapters, we proposed approaches for single-objective distributed DWSC to generate composite solutions with a given task and optimised QoS. In this chapter, we address the multi-objective distributed DWSC problem, to generate a set of compromised solutions with tradeoffs between time and cost.

6.1 Introduction

Armed with a population-based search strategy, the ability for EC algorithms to simultaneously handle multiple objectives associated with WSC problems has been widely demonstrated for the centralised WSC [30, 34, 44, 45, 53, 54, 110, 150].

In this chapter, we study three problems. Motivated by Subsection 1.2, the first problem is to design a local search method for NSGA-II. Building on the standard PLS (Pareto Local Search for MOEAs [93]) introduced in Section 2.1.6 of Chapter 2, we develop a new knowledge-driven local search method. We specifically introduce some flexibility to PLS for relaxing the acceptance of neighbour solutions considering the communication link attributes. We propose the *link dominance* concept and employ it in designing a local search technique, *Link Pareto Local Search (LPLS)*. LPLS is embedded into NSGA-II in Section 6.4 to effectively solve the multi-objective distributed DWSC. Our new technique relaxes the acceptance criteria of PLS. In addition to solutions with better fitness values, LPLS accepts solutions that outperform their parent in terms of communication attributes. This helps the algorithm to explore more diverse solutions, in particular, solutions with promising communication attributes. We experimentally show that the PLS focuses mainly on exploiting existing solutions; however, the exploration of the search space with promising new solutions is discouraged, affecting the algorithm performance.

The second problem that we solve in this chapter is the QoSconstrained multi-objective distributed DWSC. We develop an effective MOEA supported by a repair method, called *Power repair*. The power repair enjoys a set of heuristics obtained from domain-knowledge. The power repair is then combined with NSGA-II to effectively search for Pareto-optimal service compositions that satisfy QoS constraints. This repair method facilitates the construction of constraint-obeying composite services. Experimental results show that our algorithm outperforms existing NSGA-II based approaches for this problem on standard benchmark datasets.

The third problem that we aim to address in this chapter is DWSC with user preferences. Users often have QoS preferences, e.g. one QoS attribute is deemed more important than the others. QoS preferences are crucial parts of a service composition request. User preferences have not been considered in the current composition research. However, in the literature, approaches for other multi-objective problems considered using a user's *region of interest (ROI)* in the objective space to increase the density of solutions in this area [2, 13, 14, 173, 174]. Through the extensive exploration of the ROI, these algorithms can achieve higher performance within the available algorithm running time.

To model users preferences easily, we define the service composition with user preferences as a multi-objective optimisation problem with *lexicographic* preferences, *DWSC-LexUR*. In DWSC-LexUR, different priorities are assigned to different objectives.

The objectives of this chapter are as follows:

- 1. To develop a new local search, *Link Pareto Local Search (LPLS)* and combined it NSGA-II to form an effective memetic NSGA-II, (*LPLS-NSGA-II*) for the distributed DWSC;
- 2. To propose an effective repair method, *power repair*, for solving the QoS-constrained fully-automated DWSC. The power repair is enhanced by a set of heuristics that guide the repair method; It is combined with NSGA-II to form *DWSCRepair* method for effective search of valid solutions;
- 3. To develop *Lex-NSGA-II* method to solve the distributed DWSC-LexUR problem effectively. Lex-NSGA-II models users' QoS preferences using lexicographic ordering. The method also features the use of a clustering technique to ensure both diversity and preference satisfaction.

Experimental evaluations are conducted first to examine the impact and importance of the LPLS and then to analyse DWSCRepair method for the QoS-constrained DWSC. DWSCRepair is also compared with an existing repair-based method as well as a cutting-edge constrained handling method [84, 179]. Finally, experiments are conducted to examine the proposed Lex-NSGA-II for the DWSC-Lex-UR problem.

6.2 Chapter Organisation

The remainder of this chapter is organised as follows: Section 6.3 outlines the multi-objective distributed DWSC. Proposed methods for the multiobjective distributed DWSC are discussed in Section 6.4. DWSCRepair for QoS-constrained DWSC is presented in Section 6.5. Section 6.6 discusses the user-preference driven lexicographic approach and Lex-NSGA-II method. Subsequently, experiments are presented in Section 6.7. Finally, Section 6.8 summarises this chapter.

6.3 Multi-objective Data-Intensive Web Service composition Problem

Every WSC problem can be modelled as a single-objective or a multiobjective optimisation problem. In the single-objective formulation introduced in Chapter 3, a weighted sum of C_{Total} and T_{total} has been calculated as the objective function. However, in this chapter, we employ two objective functions, $f_1(CS) = T_{total}$ and $f_2(CS) = C_{total}$, where C_{Total} and T_{total} are obtained by Equations (6.1) and (6.3) (also obtained in the same way for single-objective DWSC in Chapter 3).

$$C_{total}(\mathcal{CS}) = \sum_{S_i \in \mathcal{SR}} C_{service}(S_i) + \sum_{k \in \mathcal{CL}} C_L(l_k)$$
(6.1)

where SR is the service repository and CL is the set of component services in CS.

Suppose that δ is the set of all paths in a composition CS. Δ_j is the execution time of path j. Δ_j includes the time for executing component services and the communication time of each link on path j. Δ_j can be obtained through Equation (6.2):

$$\Delta_j = \sum_{S_i \in Q} T_S(S_i) + \sum_{l_k \in L} T_L(l_k)$$
(6.2)

where Q and L are the set of component services and communication links (between two consequence services) on path j, respectively. Hence, T_{total} ,

the overall response time of CS, is calculated as the time of the most timeconsuming path in the composition as it is shown in Equation (6.3):

$$T_{total}(\mathcal{CS}) = \max_{i \in \delta} \left\{ \Delta_i \right\}$$
(6.3)

Accordingly, the formal definition of the Pareto dominance in Section 2.1 can be extended to the WSC problem with two objectives as follows:

Definition 18. (*Strict dominance*) A composite service CS_1 dominates solution CS_2 ($CS_1 \prec CS_2$) if and only if:

$$f_1(\mathcal{CS}_1) \le f_1(\mathcal{CS}_2) \quad and \quad f_2(\mathcal{CS}_1) \le f_2(\mathcal{CS}_2) \quad and (f_1(\mathcal{CS}_1) < f_1(\mathcal{CS}_2) \quad or \quad f_2(\mathcal{CS}_1) < f_2(\mathcal{CS}_2))$$
(6.4)

where $f_1(CS_1)$ and $f_1(CS_2)$ refer to the first objective (total execution time) of composite services CS_1 and CS_2 , respectively. Similarly, $f_2(CS_1)$ and $f_2(CS_2)$ refer to the second objective (total execution cost) of the corresponding composite services.

According to Definition 2 in Section 2.1, CS_1 and CS_2 are mutually nondominated if they are two different solutions which do not dominate the other. In this chapter, we use the term *strict dominance* for Definition 18 in order to differentiate it from *communication link dominance* for distributed DWSC, which will be defined in Section 6.4.

Definition 19. (*Pareto optimal*). A composite service CS^* is Pareto optimal if and only if it is not dominated by any other solution, CS, in the solution set.

6.3.1 QoS-Constrained DWSC Problem

In this subsection, we define the QoS-constrained DWSC problem. To this aim, we update the definition of the service request in Chapter 3 to include constraints as follows:

Definition 20. A composition request (or a task) for the QoS-constrained DWSC problem is defined as a triple $\mathcal{T} = (I(\mathcal{T}), O(\mathcal{T}), \mathcal{G}(\mathcal{T}))$ where $I_{\mathcal{T}}$ is a set of inputs provided to the composition by a user, and $O_{\mathcal{T}}$ is a set of task outputs expected by the user to be produced by the composite service. For task \mathcal{T} , we should find a composition (CS) that takes $I(\mathcal{T})$, produces $O(\mathcal{T})$ with optimised QoS and satisfies global QoS constraint $\mathcal{G}(\mathcal{T})$.

The QoS-constrained multi-objective distributed DWSC problem can be written as follows:

$$\min_{\mathcal{CS}} f(\mathcal{CS}) = \{ f_1(\mathcal{CS}), f_2(\mathcal{CS}) \},\$$

s.t. $f_1(\mathcal{CS}) \le k_1 \text{ and } f_2(\mathcal{CS}) \le k_2.$ (6.5)

where k_1 and k_2 are constants provided by the user ($\mathcal{G}(\mathcal{T}) = \{k_1, k_2\}$). The objective space in which $f_1(\mathcal{CS}) \leq k_1$ and $f_2(\mathcal{CS}) \leq k_2$ is called the *valid region*. This definition and the constraints model for the WSC problems has been employed in the literature for semi-automated WSC problem [34, 63]. In [63] the problem is converted into an unconstrained multi-objective problem through constraint-based service filtering, where the constraints are applied locally on each Web service. [34] converts the constrained problem into the unconstrained multi-objective problem through formulating constraints as one objective.

In order to measure the quality of solutions regarding constraint satisfaction, we define *constraint violation degree* in Definition 21.

Definition 21. *Constraint violation degree is a measure to indicates to which extent a composite service violates constraints and is obtained by Equation (6.6):*

$$\mathcal{CV} = max((f_1(\mathcal{CS}) - k_1), 0) + max((f_2(\mathcal{CS}) - k_2), 0)$$
(6.6)

that is, if CV = 0, no constraint is violated.

Definition 22. A composite service is invalid if CV > 0 for that service composition.

6.4 Link Pareto Local Search (LPLS) for Distributed DWSC

We propose a new local search technique, *link Pareto local search (LPLS)*, through extending PLS by the *link dominance* concept, for the distributed DWSC. The link dominance serves as a selection criterion for the local search. The proposed local search is then hybridised with NSGA-II (i.e., *LPLS-NSGA-II*).

Major steps of our proposed LPLS-NSGA-II are presented in Algorithm 10. Solutions are selected by a tournament selection which employs the strict dominance concept introduced in Definition 18 (line 4). Crossover and mutation operators are applied (line 5). We employ the distance-guided LCS crossover operator in Subsection 4.4.1 and the mutation operator introduced in Subsection 4.3.1. LPLS is applied at line 9 (see Section 6.4).

Algorithm 10 LPLS-NSGA-II for distributed DWSC.INPUT: SR, TOUTPUT: a set of CS

- 1: Randomly initialise population P with N(population size) individual solutions;
- 2: Evaluate solutions by decoding them into DAGs;
- 3: while the maximum number of generations not reached do
- 4: Select candidates from the pool using a tournament selection;
- 5: Apply crossover and mutation on selected candidates;
- 6: Combine mating pool and offspring as the new population;
- 7: Calculate crowding distance and perform non-dominated Sorting, putting nondominated solutions in the first front;
- 8: Select initial individuals for local search;
- 9: Apply Link Pareto Local Search (LPLS) local search.
- 10: Select *N* individuals for the next generation.
- 11: end while
- 12: Return non-dominated set of solutions in the final generation.

We first define the *link dominance* concept regarding the communication link attributes. We then propose *Link Pareto Local Search (LPLS)* based on the PLS and the link dominance to address the limitations of PLS mentioned in Section 6.1.

6.4.1 Selection of Individuals for LPLS and the Acceptance Criterion

LPLS starts with a set of solutions (initially including those solutions in the Pareto front) that have not been explored yet. A neighbour solution is then created using the flexible distance-guided local search (Section 4.5). The local search is iteratively applied to every solution in the set to produce offspring solutions. Afterwards, an acceptance criterion based on the link dominance (to be discussed in the following) is applied. Thereafter, offspring solutions are added to the set if they are accepted according to the acceptance criterion. This process stops when all solutions in the set, including the newly added offspring, have been explored. The local search procedure is presented in Algorithm 11. The complexity of LPLS mostly depends on the utilised neighbourhood definition strategy.

To define the acceptance criterion of LPLS, the *link dominance* concept, regarding the communication components of each composite service is defined as follows:

Definition 23. (*Link dominance*). A composite service CS' link dominates solution CS ($CS' \prec_L CS$) over communication-centric objectives if and only if:

$$f_{cTime}(\mathcal{CS}') \leq f_{cTime}(\mathcal{CS}) \quad and$$

$$f_{cCost}(\mathcal{CS}') \leq f_{cCost}(\mathcal{CS}) \quad and$$

$$(f_{cTime}(\mathcal{CS}') < f_{cTime}(\mathcal{CS}) \text{ or } f_{cCost}(\mathcal{CS}') < f_{cCost}(\mathcal{CS}))$$

$$(6.7)$$

where f_{cTime} and f_{cCost} are the communication time and communication cost of the corresponding composite service, respectively.

6.4. LINK PARETO LOCAL SEARCH (LPLS) FOR DISTRIBUTED DWSC127

Suppose that solution CS' has been generated by applying the local search on a solution CS. In LPLS, (Algorithm 11, line 5), the acceptance criterion initially checks for the strict dominance relationship. If it is not satisfied, the LPLS gives one more chance to CS' by considering only communication attributes (line 9), where the link dominance relationship is checked. In all cases, the neighbour solution must have a higher crowding distance (CD) than the initial solution.

 $Acceptance(CS', CS) = \begin{cases} CS' \prec \mathcal{CS} \land (CD(CS') < CD(CS)) & True \\ CS' ||\mathcal{CS} \land CS' \prec_L CS \land (CD(CS') < CD(CS)) & True \\ Otherwise & False \end{cases}$ (6.8)

Most importantly, Equation (6.8) emphasises that the solutions CS and CS' must be mutually non-dominated (refer to Definition 2) to be considered for the link dominance relationship. Therefore, the procedure only accepts neighbour solutions that strictly dominate the initial solution (parent solution), or are non-dominated by the parent while dominating it via the link dominance relationship.

Considering the neighbourhood definition strategy, LPLS is expected to increase the exploration ability by allowing the local search to carry on with every new promising solution discovered and are most likely to be improved by the above local search. LPLS encourages more explorations towards promising solutions due to the use of the link dominance, and therefore, effectively increases the chance of discovering promising new solutions.

Algorithm 11 LPLS algorithm for the distributed DWSC

```
INPUT: S (A set of start solutions)
OUTPUT: Updated solution set
 1: procedure LPLS(S)
2:
        for each \mathcal{CS} \in \mathcal{S} do
         Find neighbours of CS(N(CS))
            for each \mathcal{CS}' \in \mathcal{N}(CS) do
3:
                if CrowdingDistance(CS') > CrowdingDistance(CS) then
4:
                    if (CS' \prec CS) then // Strict dominance
5:
                        Accept(\mathcal{CS}', \mathcal{CS})
6:
                        S = Update(S, CS')
7:
8:
                    else
9:
                        if (CS' \preceq_L CS) then //Link dominance
                            Accept(\mathcal{CS}', \mathcal{CS})
10:
11:
                            S = Update(S, CS')
12:
                        end if
13:
                    end if
                end if
14:
15:
            end for
16:
        end for
17: end procedure
```

6.5 Repair-based Multi-Objective Algorithm for QoS-Constrained DWSC

In this section, we propose a repair method (i.e., power repair) to be combined with NSGA-II to solve the QoS-constrained DWSC problem. The resulted algorithm is called DWSCRepair method.

Algorithm 12 shows an overview of DWSCRepair method. Parents are selected for the crossover and mutation operators, (line 4), using the selection rules (see Subsection 6.5.1). After evaluating all offspring solutions, invalid solutions are identified (line 7). These solutions are sent to the power repair technique, which aims to repair solutions using a set of heuristics (line 8). All solutions are sorted using the non-dominated sorting and the best solutions are transferred to the next generation (line 9). The above process continues for the maximum number of generations. Each key component of this algorithm will be explained in detail in the following subsections.

Algorithm 12 NSGA-II with power repair (DWSCRepair)

INPUT: Task (*T*), Service Repository (SR), population size (*N*) **OUTPUT:** Solution set

- 1: Randomly initialise population P from SR by creating N solutions;
- 2: Decode individuals into DAGs and evaluate them;
- 3: while the maximum number of generations is not reached do
- 4: Select solutions using a tournament selection based on the *Selection rules* (see Subsection 6.5.1);
- 5: Apply crossover and mutation on selected solutions to generate offspring;
- 6: Decode and evaluate offspring solutions;
- 7: Identify invalid solutions (see Definition 22);
- 8: Apply the power repair technique on invalid solutions (Algorithm 13);
- 9: Perform non-domination sorting using *Selection rules*;
- 10: Select *N* individuals for the next-generation.
- 11: end while
- 12: Return non-dominated set of solutions in the final generation;

6.5.1 Selection Rules

A tournament selection, based on the pairwise comparison of solutions, is proposed in [122] for a single-objective optimisation problem. That selection uses *selection rules* to rank solutions in the population where both valid and invalid solutions exist. The flowchart of this selection is illustrated in Figure 6.1 for two composite services, CS1 and CS2. These selection rules are as follows: 1) any valid solution is preferred to any invalid solution; 2) between two valid solutions, the one with better objective function value (or better front and crowding distance in case of the multi-objective method) is preferred; and 3) between two invalid solutions, the one with the smallest violation of constraints, CV (Equation (6.6)), is preferred. If used alone, these rules can discard many invalid solutions which might carry useful information in their structure. However, optimal solutions to the unconstrained problems often lie on constraint boundaries [147]. Therefore, simply discarding close-to-boundary solutions makes it difficult to discover optimal solutions and reduces the population diversity. To tackle the above-mentioned shortcoming, we design a repair method that enhances these rules, in Section 6.5.2.



Figure 6.1: Flowchart of the selection rules

6.5.2 Power Repair Technique

In this subsection, we propose to repair invalid solutions and create enhanced neighbouring solutions without destroying valuable genetic information. It is proven that solutions with moderate violation of constraints enjoy the nice chance of producing good valid solutions in their neighbourhood [21, 60, 122, 126, 201] (this is also confirmed by our experiments in Subsection 6.7). As shown in the example in Figure 6.2, only a small portion of non-dominated solutions is located in the valid region. However, two invalid solutions are located in a closer distance to the valid region on the Pareto-front. If repaired properly, e.g., in the direction of minimising one of the objective functions in this example, good valid non-dominated solutions are produced through exploiting these invalid solutions, which will help the algorithm to discover high-quality solutions.

A set of seven novel heuristics (i.e., *selection heuristics set*) are defined to locate services in a composite service for repair. We are interested in using structural information, such as the communication links used to connect two services and the QoS of component services, to repair and improve invalid solutions with minor violation of constraints. The power repair



Figure 6.2: Feasible region

technique aims to reduce the constraint violation degree (CV) of any invalid solution (see Definition 21).

The power repair method tries to address both time and cost constraints. This is specifically achieved by swapping a Web service S with all other services in the same layer as S (details on layers can be found in Section 2.3).

We define the heuristic set for choosing the Web service for replacement. Hence, for each invalid composite service, up to seven component services can be selected. Selection heuristics jointly cover the majority of chances for repair as follows: 1) the longest execution time among all services in the workflow; 2) the highest cost among all services in the workflow; 3) the highest sum of execution time and cost; 4) the longest link in the workflow as its output; 5) the longest link in the workflow as its input; 6) the most expensive link in the workflow as its output; 7) the most time-consuming link in the workflow as its output.

Algorithm 13 presents the pseudo-code of the proposed repair method. A simple example is also illustrated in Figure 6.3. Suppose that service S_1 is one of the selected services by the *selection heuristic set* and there exist two other services, S_3 and S_4 , in the same layer of S_1 [54]. Therefore, two solutions are generated by swapping S_1 with S_3 and S_4 , respectively. Neighbour solutions are decoded and evaluated, and the one with the smallest CV is chosen to replace the original solution if the corresponding

Algorithm 13 Power repair technique.

- 1: while Not all invalid solutions are investigated for the repair do
- 2: Select an invalid composite service CS;
- 3: Select a set of component services in *CS* using the *selection heuristic set*;
- 4: **for all** selected services in *CS* **do**
- 5: Swap the selected service with other services in the same layer;
- 6: end for
- 7: Decode and evaluate the new permutations;
- 8: Calculate *CV* for each newly generated solution;
- 9: Select a solution from the neighbours with the smallest CV to replace the original solution.
- 10: end while

Services in the same layer:									
		la	yer 3	: S ₅ , S	6 6	lay	er 4: 6	$S_{11}^{11}, S_{11}^{11}, S_{11}^{11}$	$, S_{4}$
a	assumption: service S ₁ is selected by selection heuristics set to be modified.								
S ₆	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$						S 3		
		01	riginal	linval	id solı	ition			
S_6	$S_6 S_7 S_{13} S_{11} S_8 S_2 S_4 S_1 S_5 S_3$						S 3		
neighbor 1: service S_1 swapped with service S_4									
S ₆	$S_6 S_7 S_{13} S_{11} S_8 S_2 S_3 S_4 S_5 S_1$								
	neigh	ibor 2:	servi	ce <mark>S</mark> 1 s	swapp	ed wit	h serv	rice <mark>S</mark> 3	

Figure 6.3: Example of the power repair operator.

CV is smaller than or equal to that of the original solution. Simultaneous enhancing both objectives in Equation (6.5) during the local optimisation is achieved by eliminating an expensive communication link that appeared in the composite service, and/or a costly and time-consuming component (single Web service).

6.6 A User-Preference Driven Lexicographic Approach for Multi-Objective Distributed Web Service Composition

In this section, we address the WSC problem with user preferences. Popular MOEAs with user preferences, such as *reference solution-based nondominated sorting Genetic Algorithm (r-NSGA-II)* [162], are based on one or more reference points predefined by the user. However, we address the problem in a way that no reference points are required. The lexicographic user preference is defined as follows:

Definition 24. Given two QoS attributes of the final composite service (two objective functions in our problem), f_1 and f_2 , a lexicographical user preference, $\mathcal{P} : (f_1(\mathcal{CS}), f_2(\mathcal{CS}))$ is an ordered set of QoS attributes, which indicates that $f_1(\mathcal{CS})$ has higher priority than $f_2(\mathcal{CS})$.

According to the above discussion, the lexicographic ordering that does not require any significant interaction with the user is a suitable method of presenting user preferences. The user is only required to establish a predefined ordering of the competing objectives.

6.6.1 Preference-driven Algorithm for DWSC: Lex-NSGA-II

In this Subsection, we describe our proposed algorithm, Lex-NSGA-II. Lex-NSGA-II employs a clustering technique for selecting solutions for the next generation. The clustering technique used in this chapter is a hierarchical clustering which repeatedly divides solutions into two groups and then selects one solution from each group. This is to ensure diversity and to provide a mechanism for the algorithm for incorporating user preferences.

Lex-NSGA-II is presented in Algorithm 14. Solutions are sorted into different fronts (line 5). To apply mutation and crossover operators, a tournament-based selection is used to first select parent solutions for reproduction (line 5). This tournament selection compares solutions based on their fronts: a solution in a lower front (closer to the first front) is always preferred to a solution in a higher front. However, if two solutions are in a similar front, as a secondary criterion, the lexicographic user-preferences (see Definition 24) are used to prioritise solutions with the higher priority objective. The joint population (i.e., the pool of parents and offspring solutions) is sorted into fronts again (lines 9 and 10). Solutions are then moved to the next generation starting from the best front. However, if only a portion of solutions can survive to the next generation, an extra step is taken to decide which solutions from that front should survive to the next generation. In the standard NSGA-II, crowding distance is employed to select solutions in this step. However, in our approach, we employ a clustering technique that allows us to employ the combination of user preferences and the crowding distance.

A divisive hierarchical clustering technique is employed to help the Lex-NSGA-II algorithm select appropriate solutions for the next generation while preserving the diversity of the solution space. Performing a topdown process, this clustering technique starts by including all solutions in the front in a single large cluster. A representative solution for the cluster is chosen to be moved to the next generation, which is then removed from the current cluster. This representative is selected according to user preferences. For example, the value of the most important objective (f_1) is taken into account to compare two solutions. If two solutions have equal f_1 , they are compared regarding the second most important objective and the solution with the better f_2 is selected and so on. Each cluster is then split into two other clusters and the process of selecting representative, moving it to the next generation and creating new clusters are iterated until the next generation is filled. To divide a cluster into two clusters, two solutions that are the most distant to other solutions in terms of the crowding distance are selected as the seeds of two new clusters. The rest of the solutions are placed in one of the new clusters based on their proximity to each seed. This clustering is very simple, and in our algorithm, the clustering iteration stops when the next generation is filled.

Algorithm 14 Proposed Lex-NSGA-II algorithm for distributed DWSC-LexUR.

INPUT: Service Repository (SR) Task(T), N(population size) **OUTPUT:** A Set of Service Composition Solution

- 1: Generation counter $t \leftarrow 1$;
- 2: Randomly initialise population P_t by creating N permutations;
- 3: Decode individuals into DAGs and evaluate them;
- 4: while the maximum number of generations not reached do
- 5: Sort P_t (the parent population) into different non-dominated fronts;
- 6: Select solutions and apply crossover and mutation on them until $|Q_t| = |P_t|$ offspring solutions are generated;
- 7: Decode and evaluate offspring solutions;
- 8: Combine parents and offspring solutions into the joint population $(R_t = P_t \cup Q_t)$;
- 9: Sort R_t into fronts $F_i, 1, ..., p$, by non-dominated sorting;
- 10: $P_{t+1} \leftarrow \emptyset;$
- 11: Add all solutions of the smallest fronts to P_{t+1} provided that the total number of individuals of those fronts does not exceed $|P_{t+1}|$;
- 12: **if** $|F_i| > N |P_{t+1}|$ then
- 13: while $|P_{t+1}| < N$ do
- 14: **Perform/continue divisive hierarchical clustering on** *F*_{*i*};
- 15: Identify one solution in each of the newly generated clusters as the representative;
- 16: Add the representatives to P_{t+1} ;
- 17: **Remove added representatives from front** *F*_{*i*};
- 18: end while
- 19: **end if**

```
20: t \leftarrow t + 1;
```

```
21: end while
return P<sub>t</sub>;
```

The clustering serves an important role in our algorithm. It helps the algorithm maintain the diversity of solutions in the new population along with encouraging alignment with user preferences. The clustering groups solutions based on their distances in the objective space and ensures that at least one solution from each group is involved in forming the next generation. Furthermore, the clustering process is where we employ user preferences during the search, which could not be achieved only by using the crowding distance in the original NSGA-II (the lexicographic user preference information is used twice: first when selecting solutions for the EC operators, and second when selecting the representative of each cluster). This helps the algorithm find high-quality solutions in the ROI. Finally, compared to many other clustering techniques, our clustering does not require any predefined parameters, such as the cluster size, and stops when the next generation is completely formed.

6.7 Evaluations

In this section, performance metrics for our multi-objective methods are presented in Subsection 6.7.1. Experimental Evaluations for the LPLS-NSGA-II, DWSCRepair and Lex-NSGA-II are presented in Subsections 6.7.2, 6.7.3 and 6.7.4, respectively.

6.7.1 Performance Metrics

Methods are compared according to their execution time, as well as the inverted generational distance (IGD) and hypervolume (HV) [113]. Those metrics provide a comprehensive measure of the performance of each method [113] and are commonly used for evaluating multi-objective approaches [54]. IGD is the average distance of all Pareto solutions to the reference solution set. In this thesis, the reference solution set for IGD is obtained using an aggregated Pareto front consisting of the non-dominated solutions obtained from all runs of all approaches of a given dataset. Hypervolume is defined as the area enclosed by a reference point (obtained in a different way for each method, see the following subsections for detail) and the Pareto optimal solutions.

We ran all algorithms 30 times each with a different random seed. We

136

then employed Wilcoxon rank sum statistical test (U-test) with a significance level of 5% to verify the observed difference in the mean of all runs on IGD, HV and execution time values.

For all methods, a population of size 500 was evolved for 51 generations.

6.7.2 Evaluation of LPLS-NSGA-II

In this Subsection, parameter settings, competing algorithms, results and analysis for MA with LPLS are presented.

Parameter Settings

In this method, (1,1) is used as the reference point for calculating the HV because in the worst case a composite solution's objectives are (1,1).

Competing Algorithms

Experiments were conducted to compare the performance of LPLS-NSGA-II and PLS-based NSGA-II (*PLS-NSGA-II*) [143]. Additionally, a relaxed version of LPLS was implemented (i.e., *Relaxed-LPLS-NSGA-II*). In *Relaxed-LPLS-NSGA-II*, a newly generated solution is accepted if it dominates its parent regarding the link dominance relationship even if it is dominated by the parent regarding the strict dominance relationship. The difference between the conditions in *LPLS-NSGA-II* and *Relaxed-LPLS-NSGA-II* is further clarified through a diagram in Figures 6.4. An example is also provided in Figure 6.5. It is crucial to compare our proposed method to the relaxed version in order to verify if accepting an offspring that link dominates its parent (but might be strictly dominated by the parent) can promote effective local search. This difference is shown through a real example of a dominated offspring solution which is accepted by *Relaxed-LPLS-NSGA-II*.



Figure 6.4: Acceptance conditions in (a) LPLS, (b) Relaxed LPLS.

Parent:	Offspring:
$f_{TotalTime} = 0.624402198, f_{TotalCost} = 0.510041684$	$f_{TotalTime} = 0.6400714559, f_{TotalCost} = 0.52736972$
$f_{cTime} = 0.483754622, f_{cCost} = 0.340999999$	$f_{CTime} = 0.433754622, f_{CCost} = 0.29333333$

Figure 6.5: A parent composite service (left) and the offspring generated by Relaxed-LPLS-NSGA-II (right). Although dominated by the parent, the offspring is accepted and the parent is discarded in the Relaxed-LPLS-NSGA-II.

outperforms *Relaxed-LPLS-NSGA-II* in the final results, the hypothesis will be rejected.

Table 6.1: Mean IGD for 30 independent runs (the lower IGD the better).

PLS-NSGA-II	Relaxed-LPLS-NSGA-II	LPLS-NSGA-II
0.03 ± 0.01	0.12 ± 0.03	0.06 ± 0.01
0.2 ± 0.04	0.25 ± 0.03	0.12 ± 0.02
0.06 ± 0.02	0.07 ± 0.02	0.01 ± 0.01
0.01 ± 0.01	0.09 ± 0.02	0.05 ± 0.02
0.12 ± 0.1	0.16 ± 0.1	0.02 ± 0.01
0.04 ± 0.03	0.07 ± 0.03	0.03 ± 0.01
0.1 ± 0.04	0.11 ± 0.05	0.03 ± 0.01
0.06 ± 0.02	0.11 ± 0.03	0.05 ± 0.02
0.03 ± 0.01	0.04 ± 0.02	0.01 ± 0.01
0.08 ± 0.01	0.11 ± 0.01	0.09 ± 0.02
0.01 ± 0.02	0.02 ± 0.01	0.02 ± 0.01
0.09 ± 0.05	0.15 ± 0.04	0.02 ± 0.01
0.01 ± 0.02	0.05 ± 0.03	0.05 ± 0.02
	$\begin{array}{c} \text{PLS-NSGA-II} \\ \hline \textbf{0.03 \pm 0.01} \\ 0.2 \pm 0.04 \\ 0.06 \pm 0.02 \\ \hline \textbf{0.01 \pm 0.01} \\ 0.12 \pm 0.1 \\ \hline \textbf{0.04 \pm 0.03} \\ 0.1 \pm 0.04 \\ 0.06 \pm 0.02 \\ 0.03 \pm 0.01 \\ \hline \textbf{0.08 \pm 0.01} \\ \hline \textbf{0.01 \pm 0.02} \\ 0.09 \pm 0.05 \\ \hline \textbf{0.01 \pm 0.02} \end{array}$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $

Dataset	PLS-NSGA-II	Relaxed-LPLS-NSGA-II	LPLS-NSGA-II
WSC8-1	0.3196 ± 0.01	0.3044 ± 0.01	0.312 ± 0.01
WSC8-2	0.3715 ± 0.03	0.3454 ± 0.03	0.3943 ± 0.01
WSC8-3	0.3439 ± 0.04	0.3385 ± 0.03	0.3517 ± 0.22
WSC8-4	0.3685 ± 0.01	0.3565 ± 0.05	0.3612 ± 0.01
WSC8-5	0.3333 ± 0.02	0.3286 ± 0.03	0.3539 ± 0.03
WSC8-6	0.3289 ± 0.03	0.3215 ± 0.03	0.3298 ± 0.02
WSC8-7	0.3542 ± 0.04	0.3421 ± 0.03	0.3727 ± 0.03
WSC8-8	0.3133 ± 0.04	0.3113 ± 0.03	0.3197 ± 0.22
WSC9-1	0.3617 ± 0.01	0.3599 ± 0.03	0.3637 ± 0.01
WSC9-2	0.3111 ± 0.01	0.3026 ± 0.01	0.3109 ± 0.01
WSC9-3	0.3991 ± 0.01	0.3966 ± 0.03	0.3985 ± 0.05
WSC9-4	0.3365 ± 0.03	0.3333 ± 0.01	0.3366 ± 0.01
WSC9-5	0.3493 ± 0.04	0.3366 ± 0.02	0.3496 ± 0.02

Table 6.2: Mean HV for 30 independent runs (the higher HV the better).

Results

Tables 6.2 and 6.1 show the IGD and HV observed for *LPLS-NSGA-II*, *PLS-NSGA-II* and *Relaxed-LPLS-NSGA-II*. It is evident from those tables that *LPLS-NSGA-II* achieves significantly better values of IGD and HV for about 70 % of the tasks. *PLS-NSGA-II*, however, merely achieves significantly better values of both IGD and HV for 46% and 53% of the tasks, respectively, while *Relaxed-LPLS-NSGA-II* gains the worst IGD and HV values for all tasks. This is because *Relaxed-LPLS-NSGA-II* introduces redundant solutions to the population and accepts many worse steps, which is unnecessary. Essentially, using link dominance alone without considering true Pareto dominance does not appear to be effective.

Mean execution times for *PLS-NSGA-II*, *Relaxed-LPLS-NSGA-II* and *LPLS-NSGA-II* are shown in Table 6.3. While *Relaxed-LPLS-NSGA-II* and *LPLS-NSGA-II* are comparable to each other for each task in WSC08 and WSC09, *PLS-NSGA-II* is consistently faster (e.g., on WSC8-1). This is because *PLS-NSGA-II* has a strict condition for accepting a solution after applying local search, which makes this method restrictive compared to *Relaxed-LPLS-NSGA-II* and *LPLS-NSGA-II* and *LPLS-NSGA-II*. This characteristic, in turn, narrows the number of solutions generated by the local search and leads to a smaller set of initial solutions for the next local searches, and results in a

shorter execution time. In fact, in *Relaxed-LPLS-NSGA-II* and *LPLS-NSGA-II*, the local search has been applied on a larger set of solutions (see Section 6.4) which grows rapidly in LPLS compared to that in PLS. *PLS-NSGA-II* shows a shorter execution time among the three methods. However, the decrease in the execution time is not significant enough to compensate for the low performance of *PLS-NSGA-II*.

Task (size)	PLS-NSGA-II	Relaxed-LPLS-NSGA-II	LPLS-NSGA-II
WSC08-1 (158)	2.9 ± 0.09	3.4 ± 0.06	3.2 ± 0.17
WSC08-2 (558)	3.1 ± 0.13	5.08 ± 0.13	5.08 ± 0.52
WSC08-3 (604)	10.2 ± 1.3	14.38 ± 1.2	14.13 ± 1.79
WSC08-4 (1041)	4.93 ± 0.09	5.08 ± 0.21	5.6 ± 0.17
WSC08-5 (1090)	15.53 ± 0.041	26.61 ± 2.16	24.06 ± 1.31
WSC08-6 (2198)	42.33 ± 4.15	62.55 ± 0.03	51 ± 6.42
WSC08-7 (4113)	79.03 ± 6.79	136.53 ± 0.03	128.8 ± 13.23
WSC08-8 (8119)	182.8 ± 25.39	208.48 ± 35.74	201 ± 20.87
WSC09-1 (572)	2.54 ± 0.23	2.97 ± 0.22	2.93 ± 0.17
WSC09-2 (4129)	52.88 ± 4.07	74.33 ± 6.14	64.12 ± 5.33
WSC09-3 (8138)	97.54 ± 11.27	137.33 ± 18.06	123.15 ± 15.87
WSC09-4 (8301)	398.5 ± 50.33	478.5 ± 82.91	442.36 ± 75.39
WSC09-5 (15211)	200 ± 35.54	263 ± 46.98	245 ± 47.31

Table 6.3: Mean execution time [seconds] for the three methods.

Further Analysis and Discussions

We further evaluate the convergence rates of IGD and HV over 30 runs. We use WSC8-3 as an example to show this evaluation in Figure 6.6. The convergence is faster for *LPLS-NSGA-II* than *PLS-NSGA-II* and *Relaxed-LPLS-NSGA-II*. Therefore, *LPLS-NSGA-II* achieves significantly better IGD and HV values with an identical number of evaluations.

6.7.3 Evaluation of DWSCRepair for QoS-Constrained Distributed DWSC

In this subsection, we present experimental settings and results for the DWSCRepair.



Figure 6.6: Mean HV (left) and IGD (right) over the number of evaluations for 30 runs for non-dominated solutions for task WSC8-3. Smaller values of IGD and larger values of HV demonstrates better quality.

Competing Algorithms

To show the effectiveness of DWSCRepair, we choose two constraint handling techniques, Penalty [84] and Crossover repair [179], proposed for WSC problem, and combine them with NSGA-II to form Penalty-NSGA-II and CrRepair-NSGA-II, respectively.

1) Penalty-NSGA-II: a penalty mechanism employed in [84] for multiobjective WSC. We combine it with NSGA-II and use it in our problem as in Equation (6.9):

$$f_1'(\mathcal{CS}) = (f_1(\mathcal{CS}) - k_1) + f_1(\mathcal{CS})$$

and $f_2'(\mathcal{CS}) = (f_2(\mathcal{CS}) - k_2) + f_2(\mathcal{CS})$ (6.9)

where $f_1 - k_1$ and $f_2 - k_2$ are the amount of constraint violations for objective functions f_1 and f_2 , respectively. f'_1 and f'_2 are new fitness values for the solution (Note that all values have been normalised between 0 and 1).

2) Crossover repair (CrRepair-NSGA-II): generally, a repair can be performed through crossover operators. A crossover repair operator has been employed in [179] for the constrained Web service selection problem to reduce the chance for a composition solution to violate any constraints. Following [179], two invalid solutions sequences are selected as parents p_1 and p2, and the two-point crossover is applied to generate two offspring sequences c1 and c2. One of the crossover points is determined using p1, based on the location of the longest communication link in p1. The second crossover point is determined using p2 in the same way as p1. The sub-sequence of p1 and p2 between the two crossover points are swapped while ensuring no service duplication.

Parameter Settings

A tournament selection based on the selection rules (Subsection 6.5.1) selects candidate solutions for the mutation and crossover operators. The repair rate in CrRepair-NSGA-II is 0.95. The above parameter settings were based on popular settings discussed in the literature [45, 99].

Since existing datasets for WSC do not include constraints on QoS, we manually determined constraints on each objective. First, we carried out experiments using the unconstrained version of our problem to obtain the values of f_1 and f_2 for each task. We then set the constraint values (i.e., k_1 and k_2) to be 0.75 of the corresponding QoS's worst value according to [84]. For example, if the worst response time for the task is 0.5 when running without constraints, we set the time constraint equal to 0.375. This means the response time should not be higher than 0.375. These constraints set a higher bound to the objective functions.

Results

Tables 6.7 and 6.8 show the IGD and HV observed for Penalty-NSGA-II [84], CrRepair-NSGA-II [179] and DWSCRepair. A pairwise Wilcoxon rank sum statistical test with a significance level of 5% is conducted on the average of these values over the 30 runs for the three algorithms at 0.05 significance level (*p*-value = 0.05).

It is evident from Tables 6.4 and 6.5 that the DWSCRepair achieves significantly better values of IGD and HV for all of the tasks compared to

6.7. EVALUATIONS

Table 6.4: Mean IGD for 30 runs, for Penalty-NSGA-II, CrRepair-NSGA-II and DWSCRepair.

Task	Penalty-NSGA-II [84]	CrRepair-NSGA-II [179]	DWSCRepair
WSC8-1	0.19 ± 0.02	0.22 ± 0.03	0.06 ± 0.01
WSC8-2	0.22 ± 0.03	0.2 ± 0.05	0.14 ± 0.04
WSC8-3	0.05 ± 0.003	0.11 ± 0.001	0.01 ± 0.001
WSC8-4	0.09 ± 0.001	0.07 ± 0.002	0.02 ± 0.001
WSC8-5	0.21 ± 0.02	0.24 ± 0.1	0.038 ± 0.02
WSC8-6	0.055 ± 0.01	0.057 ± 0.001	0.056 ± 0.002
WSC8-7	0.11 ± 0.03	0.12 ± 0.03	0.12 ± 0.03
WSC8-8	0.07 ± 0.01	0.07 ± 0.01	0.05 ± 0.01
WSC9-1	0.03 ± 0.01	0.02 ± 0.02	0.01 ± 0.01
WSC9-2	0.091 ± 0.005	0.098 ± 0.004	0.03 ± 0.001
WSC9-3	0.02 ± 0.001	0.03 ± 0.001	0.02 ± 0.001
WSC9-4	0.1 ± 0.001	0.23 ± 0.001	0.08 ± 0.001
WSC9-5	0.09 ± 0.002	0.13 ± 0.001	0.06 ± 0.001

Table 6.5: Mean HV for 30 runs, for penalty-NSGA-II, CrRepair-NSGA-II and DWSCRepair.

Task	Penalty-NSGA-II [84]	CrRepair-NSGA-II [179]	DWSCRepair
WSC8-1	0.29 ± 0.02	0.3022 ± 0.03	0.341 ± 0.02
WSC8-2	0.3113 ± 0.01	0.3163 ± 0.01	0.3511 ± 0.01
WSC8-3	0.332 ± 0.004	0.314 ± 0.004	0.359 ± 0.004
WSC8-4	0.318 ± 0.002	0.317 ± 0.01	0.339 ± 0.004
WSC8-5	0.326 ± 0.005	0.32 ± 0.006	0.358 ± 0.001
WSC8-6	0.3242 ± 0.004	0.3227 ± 0.01	0.3238 ± 0.009
WSC8-7	0.33 ± 0.008	0.33 ± 0.008	0.336 ± 0.009
WSC8-8	0.3121 ± 0.001	0.3124 ± 0.001	0.313 ± 0.009
WSC9-1	0.361 ± 0.03	0.3631 ± 0.04	0.3692 ± 0.05
WSC9-2	0.34 ± 0.008	0.335 ± 0.009	0.34 ± 0.009
WSC9-3	0.302 ± 0.001	0.306 ± 0.005	0.32 ± 0.005
WSC9-4	0.34 ± 0.005	0.31 ± 0.005	0.357 ± 0.005
WSC9-5	0.33 ± 0.001	0.27 ± 0.005	0.342 ± 0.004

the other two methods. CrRepair-NSGA-II outperforms Penalty-NSGA-II on 30% of tasks. Unlike CrRepair-NSGA-II which uses random repair, the proposed power repair technique explicitly considers the distribution of services and effectively remove the longest communication links from those solutions that violate any QoS constraints. The average execution time and the corresponding standard deviation in seconds over 30 independent runs for each approach are shown in Table 6.6. Table 6.6 demonstrates that the proposed method has the longest execution time since attempts to repair all invalid solutions.

Penalty-NSGA-II is faster than the other methods. The reason is that,



Table 6.6: Mean execution time [seconds] over 30 runs for penalty-NSGA-II, CrRepair-NSGA-II and DWSCRepair.

Figure 6.7: Mean IGD (left) and HV (right) on task WSC9-5.

unlike the repair-based methods, Penalty-NSGA-II does not require extra evaluations. In repair-based methods, all repaired solutions are decoded and evaluated so that the one with better objective functions is defined.

Further Analysis

We further evaluate the convergence behaviours concerning both IGD and HV over 30 runs using task WSC9-5 as an example. Figure 6.7 demonstrates the evolution of the mean values of the IGD and HV over the number of evaluations for the proposed method, CrRepair-NSGA-II and



method CrRepair * DWSCRepair

Figure 6.8: Pareto solutions for WSC9-5 (left) and WSC9-2 (right) (note that only valid solutions and region are shown in this figure).

Penalty-NSGA-II, where, DWSCRepair converges slower than the two other methods but continues to improve the solutions. We present a plot of the Pareto front valid solutions of WSC9-2 and WSC09-5 obtained by the three methods over 30 independent runs in Figure 6.8. The best Pareto optimal solutions are identified based on the combined results of all 30 runs of each method. It is evident from Figure 6.8 that the Pareto front generated by the Penalty-NSGA-II is more widely distributed for both tasks.

Solutions generated by DWSCRepair approach dominate solutions from the other two approaches in the central region. This demonstrates that DWSCRepair is effective in generating both quality and valid solutions. To be specific, DWSCRepair produces valid solutions with better quality than other methods.

6.7.4 Evaluation of Lex-NSGA-II for DWSC-Lex-UR

In this subsection, competing algorithms, experimental settings and results for the Lex-NSGA-II are discussed.

Competing Algorithms

We employ two multi-objective baseline algorithms for comparisons: first, the standard NSGA-II [61], where final solutions should be refined to fit in the ROI; and second, r-NSGA-II [162] since this is the most widely used multi-objective algorithm with user preference [38, 162, 223].

Settings

Although the standard r-NSGA-II has only one reference point, it can be modified to address multiple reference points. In this thesis, we replace the calculation of weighted Euclidean distance in standard r-NSGA-II by the distance to the closest reference point. However, exactly defining the reference point in real-world problems might be difficult. Herein, we choose five reference points: (0, 0.2), (0.04, 0.2), (0.08, 0.2), (0.12, 0.2) and (0.16, 0.2). These reference points are chosen because they are located in an evenly-spaced manner on a line which is drawn from x = y (the border of the ROI) to the y-axis (response time) and all of them are in the ROI.

However, we investigate the algorithm's sensitivity to preference orders by studying different orders of the objectives.

In the presence of user preferences, the reference solution set for IGD is obtained using an aggregation of final solutions of all runs which reside in the ROI. Point (0.5,1) has been used as the reference point to calculate HV since the preference order is \mathcal{P} : (*cost*, *time*). We carried out experiments twice, each time for the different order of the QoS attributes as the preferences.

Results

Tables 6.7 and 6.8 show IGD and HV when the cost has a higher priority than time. The ROI for this case is shown in Figure 6.10 (note that the objectives have been normalised). This model can be extended to include any number of objectives simply by defining the preference order

Table 6.7: Mean IGD over 30 runs for preference $\mathcal{P} : (cost, time)$.

Task	NSGA-II	r-NSGA-II	Lex-NSGA-II
WSC08-1	0.039 ± 0.004	0.033 ± 0.002	0.029 ± 0.001
WSC08-2	0.038 ± 0.002	0.032 ± 0.009	0.022 ± 0.001
WSC08-3	0.081 ± 0.01	0.077 ± 0.004	0.058 ± 0.001
WSC08-4	0.032 ± 0.001	0.03 ± 0.02	0.031 ± 0.03
WSC08-5	0.031 ± 0.002	0.031 ± 0.001	0.029 ± 0.001
WSC08-6	0.041 ± 0.003	0.04 ± 0.004	0.032 ± 0.002
WSC08-7	0.13 ± 0.06	0.084 ± 0.02	0.085 ± 0.006
WSC08-8	0.0381 ± 0.001	0.0352 ± 0.002	0.031 ± 0.002
WSC09-1	0.019 ± 0.003	0.018 ± 0.008	0.014 ± 0.001
WSC09-2	0.063 ± 0.001	0.062 ± 0.003	0.041 ± 0.001
WSC09-3	0.043 ± 0.02	0.043 ± 0.02	0.039 ± 0.03
WSC09-4	0.034 ± 0.005	0.031 ± 0.006	0.018 ± 0.002
WSC09-5	0.12 ± 0.002	0.06 ± 0.003	0.06 ± 0.007

Table 6.8: Mean HV for NSGA-II, r-NSGA-II and Lex-NSGA-II over 30 runs for preference \mathcal{P} : (*cost*, *time*).

Task	NSGA-II	r-NSGA-II	Lex-NSGA-II
WSC08-1	0.27 ± 0.06	0.28 ± 0.07	0.3 ± 0.01
WSC08-2	0.299 ± 0.023	0.318 ± 0.01	0.323 ± 0.001
WSC08-3	0.302 ± 0.001	0.311 ± 0.001	0.3265 ± 0.001
WSC08-4	0.245 ± 0.002	0.282 ± 0.001	0.2716 ± 0.006
WSC08-5	0.301 ± 0.003	0.3 ± 0.004	0.315 ± 0.003
WSC08-6	0.291 ± 0.01	0.301 ± 0.001	0.3238 ± 0.01
WSC08-7	0.291 ± 0.003	0.318 ± 0.002	0.313 ± 0.003
WSC08-8	0.3 ± 0.002	0.309 ± 0.01	0.3142 ± 0.012
WSC09-1	0.289 ± 0.003	0.292 ± 0.003	0.313 ± 0.001
WSC09-2	0.243 ± 0.002	0.253 ± 0.001	0.2856 ± 0.004
WSC09-3	0.283 ± 0.006	0.284 ± 0.002	0.2917 ± 0.007
WSC09-4	0.251 ± 0.001	0.271 ± 0.001	0.2891 ± 0.002
WSC09-5	0.214 ± 0.001	0.275 ± 0.004	0.2731 ± 0.004

of them. Tables 6.7 and 6.8 indicate that Lex-NSGA-II outperforms other methods on 77% of test cases for both IGD and HV. This method enjoys a higher diversity due to the use of clustering. On the other hand, NSGA-II produces the worst results because the algorithm does not incorporate user preference information during the search. r-NSGA-II uses a modified domination relation. However, Lex-NSGA-II employs clustering to ensure diversity. Figure 6.9 demonstrates the evolution of the mean values of the IGD and HV over the number of generations on WSC08-6 for the three algorithms, where Lex-NSGA-II continues to improve the solutions.

The average execution time in seconds over 30 independent runs are shown in Table 6.9. This table shows that r-NSGA-II is slightly faster than Lex-NSGA-II since it does not perform clustering.



Figure 6.9: IGD (left) and HV (right) of three methods on WSC8-6.

Table 6.9: Mean execution time [seconds] over 30 runs for preference \mathcal{P} : (*cost*, *time*).

Task	NSGA-II	r-NSGA-II	Lex-NSGA-II	
WSC08-1	1.05 ± 0.1	1.1 ± 0.07	1.4 ± 0.17	
WSC08-2	1.09 ± 0.1	1.26 ± 0.12	1.3 ± 0.52	
WSC08-3	2.9 ± 0.4	3.1 ± 1.26	3.2 ± 1.79	
WSC08-4	1.8 ± 0.3	2 ± 0.03	2.1 ± 0.13	
WSC08-5	4.9 ± 1.6	5.1 ± 1.9	5.2 ± 1.13	
WSC08-6	5.1 ± 4.7	6.9 ± 1.41	10.33 ± 3.42	
WSC08-7	14.6 ± 2.9	16.09 ± 8.43	18.8 ± 11.33	
WSC08-8	68.6 ± 14.17	76 ± 16.7	78 ± 20.87	
WSC09-1	1.19 ± 0.25	1.63 ± 0.19	1.93 ± 0.17	
WSC09-2	11.9 ± 3.2	13.11 ± 3.93	15.12 ± 5.33	
WSC09-3	24.11 ± 12.14	25.14 ± 13.17	27.15 ± 11.87	
WSC09-4	80.39 ± 23.7	85.66 ± 35.21	93.36 ± 35.39	
WSC09-5	72.9 ± 25.41	78 ± 22.75	81 ± 31.21	



Figure 6.10: User-preferred region (ROI) ($f_1 < f_2$). The dashed curve shows the Pareto-front.

Table 6.10: Mean IGD over 30 runs for preference \mathcal{P} : (*time*, *cost*).

Task	NSGA-II	r-NSGA-II	Lex-NSGA-II
WSC08-1	0.028 ± 0.002	0.029 ± 0.001	0.024 ± 0.002
WSC08-2	0.044 ± 0.001	0.04 ± 0.001	0.038 ± 0.001
WSC08-3	0.034 ± 0.002	0.031 ± 0.001	0.03 ± 0.002
WSC08-4	0.052 ± 0.003	0.049 ± 0.01	0.046 ± 0.02
WSC08-5	0.047 ± 0.004	0.043 ± 0.003	0.044 ± 0.003
WSC08-6	0.035 ± 0.003	0.032 ± 0.001	0.033 ± 0.002
WSC08-7	0.049 ± 0.003	0.046 ± 0.002	0.044 ± 0.002
WSC08-8	0.043 ± 0.001	0.034 ± 0.002	0.036 ± 0.002
WSC09-1	0.045 ± 0.004	0.043 ± 0.002	0.04 ± 0.002
WSC09-2	0.06 ± 0.004	0.058 ± 0.004	0.049 ± 0.003
WSC09-3	0.023 ± 0.003	0.022 ± 0.004	0.019 ± 0.001
WSC09-4	0.051 ± 0.005	0.049 ± 0.004	0.05 ± 0.004
WSC09-5	0.09 ± 0.002	0.07 ± 0.002	0.08 ± 0.002

Table 6.11: Mean HV for NSGA-II, r-NSGA-II and Lex-NSGA-II over 30 runs for preference \mathcal{P} : (*time*, *cost*).

Task	NSGA-II	r-NSGA-II	Lex-NSGA-II
WSC08-1	0.25 ± 0.06	0.27 ± 0.07	0.27 ± 0.06
WSC08-2	0.32 ± 0.01	0.331 ± 0.02	0.338 ± 0.02
WSC08-3	0.313 ± 0.001	0.322 ± 0.001	0.324 ± 0.001
WSC08-4	0.291 ± 0.003	0.302 ± 0.003	0.305 ± 0.005
WSC08-5	0.321 ± 0.002	0.332 ± 0.002	0.317 ± 0.002
WSC08-6	0.297 ± 0.003	0.318 ± 0.006	0.311 ± 0.007
WSC08-7	0.312 ± 0.004	0.322 ± 0.003	0.333 ± 0.003
WSC08-8	0.301 ± 0.002	0.317 ± 0.02	0.312 ± 0.02
WSC09-1	0.282 ± 0.004	0.286 ± 0.005	0.328 ± 0.01
WSC09-2	0.274 ± 0.005	0.285 ± 0.004	0.287 ± 0.002
WSC09-3	0.251 ± 0.02	0.262 ± 0.01	0.289 ± 0.01
WSC09-4	0.231 ± 0.01	0.281 ± 0.01	0.272 ± 0.01
WSC09-5	0.233 ± 0.001	0.252 ± 0.001	0.251 ± 0.002

Additionally, we conducted experiments for \mathcal{P} : (*time*, *cost*) where response time was proffered to cost. Results are shown in Tables 6.10 and 6.11. Lex-NSGA-II achieved significantly better IGD and HV among the three algorithms for 70% of tasks. This confirmed that our method can be applied effectively to handle different orders of preferences.

6.8 Chapter Summary

In this chapter, we propose a memetic NSGA-II (i.e., LPLS-NSGA-II) for effectively solving the distributed DWSC problem. In particular, we proposed a novel local search (i.e., LPLS) for NSGA-II that utilises the features of distributed DWSC. Our experimental evaluation using benchmark datasets showed that our proposed memetic NSGA-II finds Pareto front solutions with better IGD and HV compared to existing hybrid MOEAs. This is presumably due to its exploration flexibility along with the exploitation during the local search, and still being decent enough to prevent inferior solutions. We developed another variation of our proposed method to conduct in-depth experiments. This variant relaxed the acceptance criteria of local search to study the influence of accepting dominated solutions.

We further studied the problem of QoS-constrained multi-objective distributed DWSC. We designed a novel knowledge-based repair technique, the power repair, which takes advantage of a set of heuristics for the distributed DWSC. We then defined DWSCRepair by combining power repair with NSGA-II. Results demonstrated that DWSCRepair can achieve significantly better performance in all of the benchmarks compared to the existing relevant DWSC methods while searching valid composition solutions.

We finally studied the distributed DWSC problem with QoS priority preferences and used a convenient way of expressing users' desires in the lack of certain user preferences. We employed lexicographic ordering and proposed a multi-objective Pareto-based optimisation algorithm (i.e., Lex-NSGA-II). We combined the concepts of Pareto dominance and lexicographic ordering, as well as a hierarchical clustering technique to enhance the effectiveness of Lex-NSGA-II. We compared Lex-NSGA-II with two other algorithms in this context. Experimental results showed that our proposed method is the most effective in satisfying the user's preferences.

Chapter 7

Dynamic Distributed Web service Composition

In this chapter, we study the problem of *dynamic distributed DWSC* (D^2 -DWSC). We specifically address the distributed DWSC in dynamic bandwidth environment (motivated in Section 1.2 and Subsection 2.2.2). We define a problem model to formally define the robustness of composite services in the presence of bandwidth changes. We then propose an algorithm to generate robust solutions for D²-DWSC. Finally, we propose a mutation operator to enhance the algorithm performance. Experimental results and empirical comparisons to existing algorithms demonstrate the effectiveness of our method.

7.1 Introduction

The dynamically changing network environment can cause QoS fluctuations and SLA violations. Network attributes, in particular bandwidth, are subject to dynamic changes, which disrupts the expected time and cost required to execute a composite service. However, existing approaches have not considered the impact of a dynamically changing network environment on distributed DWSC (see Subsection 2.2.2). In previous chapters, we assumed that the bandwidth of each communication link remains fixed over time.

It is important and desirable to build service compositions that are *robust* to network changes. A robust composite service is more likely to cope with the dynamically fluctuating network bandwidth without major degradation in the QoS.

In this chapter, the following contributions are achieved:

- We provide a model for formulating the problem. Based on the model, we design a new GA-based algorithm (i.e., Distance-R-MA) to find composite services that are robust to network bandwidth changes.
- For our GA-based algorithm, we introduce a distance-based mutation operator to improve the effectiveness of our algorithm as well as a fitness measure to efficiently and effectively estimate the robustness. The proposed distance-based mutation operator is effective in reducing inter-service distance and therefore can improve the quality of solutions.

Extensive experiments using real-world network bandwidth data show that our Distance-R-MA can find composition solutions with lower execution cost and response time than solutions generated by existing algorithms.

7.2 Chapter Organisation

This chapter is organised as follows: an introduction to the objectives of this chapter is given in Section 7.1. Section 7.3 discusses the problem formulation. Our proposed approach for D^2 -DWSC is presented in Section 7.4. Section 7.5 presents experimental results. Finally, Section 7.6 summarises the chapter.

152

7.3 Dynamic Distributed Data-Intensive Web Service Composition Problem

In this section, we first present basic terms such as the bandwidth sample and robustness, which are necessary for D^2 -DWSC. We also update the definition of the communication link and the transfer time for the D^2 -DWSC. Afterwards, we present our novel model of dynamic service composition to include the impact of the dynamic network on service compositions.

In this chapter, we consider bandwidth values which change over time. Therefore, the bandwidth value of a communication link depends on the time. We update Definition 9 (Section 3.3) of the communication link as follows:

Definition 25. A communication link, l_k is a logical link between two directly connected services in a composite service. It is represented as an edge in the corresponding workflow. A communication link is defined as the tuple $l_k = \langle S_i, S_j, B_{l_k}(t) \rangle$. In a composite Web service, a point-to-point communication happens between S_i (source service) and S_j (destination service) if communication link l_k in the DAG connects S_i and S_j .

As we can see in Definition 25, the bandwidth of the communication link varies with time (i.e., $B_{l_k}(t)$ depends on the time).

Additionally, the definition of transmission time for the static distributed DWSC (Section 3.3) is updated as follows:

Definition 26. Transmission time, $T_{tr}(l_k, B_{l_k}(t))$, is the time required for communicating between two services through communication link l_k when the available bandwidth for l_k at time t is $B_{l_k}(t)$. T_{tr} changes over time (every one minute in this thesis), subject to the change of B_{l_k} .

Communication cost, $C_L(l_k)$, and the overall cost of a composite service CS (i.e., $C_{total}(CS)$) are obtained in the same way for the static single-

objective distributed DWSC (see Subsection 3.3.3 in Chapter 3 for more detail).

Definition 27. A sample, ϵ_t , defines a set of sampled bandwidth in a composite service, one bandwidth value for each communication link. t is the discrete sampling time which also shows the sample number.

For example, in sample $\epsilon_t = \{B_{l_l}(t), B_{l_2}(t), ..., B_{l_{|\mathcal{CL}|}}(t)\}, B_{l_1}(t)$ is the bandwidth value for link l_1 in the t^{th} sample. In D²-DWSC, the transmission time for each link varies from time to time as the bandwidth changes dynamically. Therefore, the transmission time for link l_k at time t can be calculated as shown in Equation (7.1):

$$T_{tr}(l_k, B_{l_k}(t)) = \frac{size(D_{i,m})}{B_{l_k}(t)}.$$
(7.1)

where $size(D_{i,m})$ is the size of data, $D_{i,m}$, to be transferred between two services through l_k (see Section 3.3). The total response time of a composite service is the longest execution time of all the paths from the *Start* to the *End* node.

Therefore, the execution time Δ_j of path j in a composition CS can be obtained through Equation (7.2). Δ_j includes the time for executing component services and the communication time of each link on path j:

$$\Delta_j = \sum_{S_i \in Q} T_S(S_i) + \sum_{l_k \in L} T_{tr}(l_k, B_{l_k}(t))$$
(7.2)

where Q and L are the set of component services and communication links (between two consequence services) on path j, respectively. Therefore, T_{total} can be obtained by Equation (7.3):

$$T_{total}(\mathcal{CS}, \epsilon_t) = \max_{i \in \delta} \left\{ \Delta_i \right\}$$
(7.3)

Finally, the quality of a composite service is obtained in Equation (7.4):

154
$$F(\mathcal{CS}, \epsilon_t) = w\hat{T}_{total}(\mathcal{CS}, \epsilon_t) + (1 - w)\hat{C}_{total}(CS)$$
(7.4)

where T_{total} and C_{total} are normalised values (between 0 and 1) (see [54]) of T_{total} and C_{total} , respectively, and w is a positive weight (see Subsection 3.3.3 in Chapter 3 for more detail). We define the expected F in the following. It is highly desirable to find a service composition with optimal quality (i.e., minimal expected $F(CS, \epsilon)$) in the context of a network with dynamic bandwidth changes.

Definition 28. The robustness of a composite service is defined as the expected average QoS of the composite service subject to dynamic bandwidth changes.

We consider dynamic bandwidth changes regarding all possible samples of link bandwidths. Suppose that $E = \{\epsilon_1, \epsilon_2, ..., \epsilon_{|E|}\}$ is a full set of bandwidth samples for all communication links in a composite service, where all bandwidth samples have the same probability. The robustness of *CS* can be then defined as the expectation on the samples in Equation (7.5), where the average QoS stays at a high level (i.e., we consider the robustness of composite services by regarding the impact of dynamic bandwidth changes):

$$Robustness(\mathcal{CS}) = \mathbb{E}_{\epsilon}[F(\mathcal{CS}, \epsilon)] = \sum_{i=1}^{|E|} Pr(\epsilon_i)F(\mathcal{CS}, \epsilon_i)$$
(7.5)

The aim of D^2 -DWSC is then to find a service composition with high robustness, which yields better expected QoS than other compositions at the execution stage. Since *E* refers to the set of all samples that can be extremely large. Therefore, the robustness cannot be precisely and efficiently evaluated. We will estimate the robustness using an average function in Section 6.6.1.

7.4 Distance-based MA for fully-automated D²-DWSC

Since the distance between services has a significant impact on the quality of services, the longer the distance between component services, the greater the likelihood that the network dynamics affect the composite services.

In this section, we propose to employ the inter-service distance (see Definition 10 in Chapter 3) in search of service compositions. Inter-service distances carry important problem knowledge [168] (e.g., reducing the inter-service distance in a composite service can enhance the service performance). Additionally, inter-service distances usually do not change over time and hence are easy to be used. To this aim, we propose an MA, Distance-based-Robust MA (Distance-R-MA). To reduce the inter-service distance, our method proposes a distance-based mutation operator, and then, utilises simulation models (see Subsection 3.4.2 in Chapter 3) for obtaining bandwidth samples. Finally, our method generates a robust composite service. Our method differs from the two-stage robust method proposed in [191] in several aspects: 1) we address the network-aware composition problem, where network attributes are in the objective function (D²-DWSC); 2) We propose a mutation operator for D²-DWSC which employs distance information of services; 3) We consider fluctuations in QoS caused by changes in the communication links bandwidth rather than random scenarios of service failures in [191]. This is particularly important because in a real-world network environment bandwidth fluctuations cause the majority of failures [168]. Bandwidth fluctuations are critical reasons that make services unreachable within the maximum response time in a distributed environment, causing them to be treated as failed services [168, 200]. Any individual service is often used to support many composite services simultaneously. The collective demands from all users and composite services often imply that the service location should remain stable for a sufficiently long time. During this time, it is desirable to optimise the robustness of composite services considering the current location of services.

7.4.1 Estimating Robustness

Since the number of samples can be extremely large, it is impossible to precisely evaluate the robustness of *CS* (i.e., calculating robustness across all bandwidth samples is impossible). Therefore, in our GA-based approach, we employ a random subset of samples (i.e., M) to approximate Equation (7.5) ($M \subset E$). We define the fitness function in Equation (7.6) that gives an unbiased estimation of robustness defined in Equation (7.5), as the average QoS of the composite services across samples in M:

$$Fitness(\mathcal{CS}) = \frac{1}{|M|} \sum_{\epsilon_i \in M} F(\mathcal{CS}, \epsilon_i)$$
(7.6)

where |M| is the number of samples in M.

Estimating the fitness of composite services using several samples, e.g. 200 samples, has the benefit that the solutions are not sensitive to the fitness measure used to determine the quality of each evolved composite service [191]. The objective of our algorithm is to minimise fitness in Equation (7.6) by producing a suitable composite service, constructed from services in a repository \mathcal{R} .

7.4.2 Distance-based Algorithm

The pseudocode of the *distance-based method for producing robust solutions for* D^2 -DWSC, Distance-R-MA, at the design stage is shown in Algorithm 15. The population is randomly initialised at line 1 (individuals of the population are sequences of services). Sequences are converted into workflows using the decoding process to form composite services. Afterwards, a set of samples of network bandwidth is generated (unlike the static DWSC)

where each link has one static bandwidth value) for each composite service to evaluate its fitness through Equation (7.6). We employ our proposed *distance-based mutation operator* (see Subsection 7.4.3), which utilises location information to reduce the inter-service distance. We adopt the crossover and local search operators of the flexible distance-based MA (see Section 4.5).

Algorithm 15 Distance-R-MA Algorithm
INPUT: Composition Task (T), Service Repository (\mathcal{R})
OUTPUT: A Robust Composition Solution
Set generation counter $g \leftarrow 0$;
Initialise population P^0 by random individual solutions, Π s, and decode all Π s
to workflows.;
while $g \le predefined$ number of iterations do
for each Π in current population do
Generate $M = \{\epsilon_i\}$ (<i>M</i> is a subset of bandwidth samples);
for each ϵ_i in M do
Evaluate Π using the bandwidth values specified in ϵ_i
Calculate $F(CS, \epsilon_i)$ (Equations (7.1) and (7.4)).
end for
Estimate the fitness of Π using Equation (7.6);
end for
Select solutions based on their <i>Fitness</i> ;
$g \leftarrow g + 1;$
Apply crossover operator;
Apply distance-based mutation operator (Subsection 7.4.3);
Apply neighbourhood search (local search);
Decode and evaluate all newly generated sequences;
end while
return Sequence With Best Fitness

7.4.3 Inter-service distances and the mutation operator

In this subsection, we propose the distance-based mutation operator of the Distance-R-MA algorithm for a distributed service environment.

The inter-service distance in a composite service is the key factor affecting QoS. According to the bandwidth model obtained in Subsection

158

3.4.2, and since the bandwidth changes of physical communication links affect the bandwidth available to the logical communication link, the interservice distance has a strong impact on the communication time. Further, as shown in previous chapters, the inter-service distance is closely related to the communication cost. Consequently, a mutation operator can be designed to decrease the maximum inter-service distance in a composite service while increasing the diversity of the solution space. Our mutation operator aims to reduce the inter-service distance and hence the QoS, through eliminating long communication links. Long communication links increase the communication attributes such as response time and cost [159], which in turn, degrade the QoS of the composite service. However, for fully-automated WSC, only primitive forms of mutation in GAs have been exploited by existing studies [54, 220, 191].

This mutation operator has two steps. The first step follows the same idea as the mutation for WSC [54], which includes inserting a prefix and appending a suffix to the original sequence (Figure 7.1). This prefix consists of k services chosen at random from the relevant services in the repository (the parameter k is set before the execution). The suffix appended consists of a randomly generated sequence comprising all services in the repository. This is to prevent the operation from rendering the corresponding composition non-functional [54]. In the second step, we employ the distance information to the mutation operator. If designed properly, e.g. by employing domain-knowledge, mutation operators can increase the performance of the GA.

In this mutation operator, two services from the resulted sequence are swapped. This mutation selects the service with the highest average distance to its directly connected services in the workflow, as the first service to be swapped (Figure 7.2). However, the second service is chosen at random to encourage diversity. That is, one of the two services to be swapped is determined on the workflow based on the distance information, i.e., the service with the highest distance to other component services



Figure 7.1: Distance-based mutation operator.

and another is determined randomly (although services are selected based on workflows, we change their corresponding positions in the indirectly represented service sequences).

Figure 7.2 illustrates an example of determining the first service for the swap in the mutation operator. The physical distance between each pair of connected services is shown on each edge on the workflow in Figure 7.2a. The average distance for each Web service is calculated in Figure 7.2b. Therefore, the distance-based mutation operator swaps Web service S_f with a randomly chosen service in the corresponding sequence.

7.5 Evaluations

In this section, we present competing algorithms in Subsection 7.5.1. We then discuss parameter settings and results in Subsections 7.5.2 and 7.5.3, respectively.

7.5.1 Competing Algorithms

We carry out experiments to evaluate the performance of the Distance-R-MA method. Two algorithms are selected to be baseline algorithms for comparing with our proposed method. The first one is flexible distance-



Figure 7.2: Identifying a service for swap: (a) A workflow with the distance information between directly connected services, (b) calculation of the average distance for each service and determining one with the largest one to others.

guided MA (*F-MA*) (see Section 4.5). As it can be recalled from Chapter 4, F-MA achieved a higher performance for the static DWSC compared to other algorithms. The second competing algorithm is a robust version of F-MA, *R-MA*. *R-MA* is chosen because it differs from our algorithm in that it does not use the distance-based mutation operator. All competing algorithms search for one solution at the design stage. The F-MA generates a solution desiring to optimise the weighted sum function in Equation (7.4), while *R-MA* and *Distance-R-MA* optimise the robustness measure. Through comparing with these algorithms, the effect of the proposed mutation operator and the proposed formulation for the robustness optimisation can be determined.

7.5.2 Parameter Settings and Datasets

A population of size 30 are evolved for 100 generations for all methods. Local search, mutation and crossover operators' rate are set to 0.05, 0.05, and 0.95, respectively. Parameter n in mutation is set to three. The tourna-

ment selection size was two. Elitism with size two is applied. The parameter w in Equation (7.4) is set to 0.5.

We employ the benchmark dataset developed in Section 3.4. However, in previous chapters, each communication link had a fixed bandwidth value. In this chapter, instead of one value, we generate the bandwidth value several times. For *R-MA* and *Distance-R-MA* the sample size at the design stage is 100 (M = 100) (see Algorithm 15). We subsequently employ 300 bandwidth samples on the result of each run during the execution stage. Note that a large number of sampled scenarios (e.g., 300) is taken into account for testing while a smaller number of sampled scenarios (e.g., 100) is used at the design stage. This difference is important for the design stage in order to remain highly efficient, whereas we want to accurately measure the robustness of any composite service during the execution stage [191].

7.5.3 Results

Algorithms have been compared at the execution stage and the response time and cost of solutions. We verify the significant difference in corresponding values of the three algorithms using a Wilcoxon signed-rank test.

The mean fitness values of 30 runs with different random seeds are shown in Table 7.1, where *Distance-R-MA* produces solutions with robust quality at the design stage which achieves higher QoS at the test stage. That is, composite services produced by our proposed method show significantly better performance in the presence of bandwidth fluctuations. Although the *F-MA* method for the distributed DWSC provides solutions with good fitness at the design stage, as verified in [158], there is no guarantee that the composite services can remain robust. Significantly better values concerning the statistical test have been highlighted in Table 7.1. Additionally, *Distance-R-MA* outperforms *R-MA* in 11 out of 13 tasks, which indicates that reducing inter-service distances, achieved by the proposed mutation operator, significantly improved the QoS of the solutions.

Table 7.2 shows the average response time of the solutions found by each algorithm. The response time has great importance in D^2 -DWSC since it is directly affected by the bandwidth changes. Solutions produced by *Distance-R-MA* have significantly better response time than solutions generated by the two other algorithms. Finally, Table 7.3 presents the cost of the composite service found by the three algorithms. These observations verify that our proposed method improves the cost and response time of composite services.

Table 7.1: Mean and standard deviations of fitness values, at the execution stage, based on the baseline solution. Wilcoxon test with the significance level of 0.05 is applied.

Task	F-MA [158]	R-MA	Distance-R-MA
WSC08-1	0.52 ± 0.12	0.49 ± 0.05	0.48 ± 0.04
WSC08-2	0.62 ± 0.14	0.58 ± 0.16	0.57 ± 0.1
WSC08-3	0.64 ± 0.2	0.63 ± 0.06	0.59 ± 0.18
WSC08-4	0.51 ± 0.19	0.5 ± 0.09	0.5 ± 0.09
WSC08-5	0.62 ± 0.09	0.62 ± 0.09	0.61 ± 0.01
WSC08-6	0.62 ± 0.3	0.62 ± 0.3	0.61 ± 0.23
WSC08-7	0.7 ± 0.05	0.69 ± 0.27	0.68 ± 0.22
WSC08-8	0.61 ± 0.09	0.59 ± 0.07	0.60 ± 0.19
WSC09-1	0.65 ± 0.12	0.64 ± 0.11	0.6 ± 0.11
WSC09-2	0.63 ± 0.1	0.63 ± 0.2	0.59 ± 0.3
WSC09-3	0.65 ± 0.2	0.61 ± 0.1	0.60 ± 0.2
WSC09-4	0.62 ± 0.1	0.61 ± 0.1	0.59 ± 0.07
WSC09-5	0.68 ± 0.2	0.65 ± 0.2	0.65 ± 0.15

7.6 Chapter Summary

We proposed a fully-automated approach, Distance-R-MA, to the D²-DWSC problem in the context of dynamic bandwidth changes, as well as a model to evaluate the robustness of distributed services. We evaluated Distance-R-MA with samples based on the real-world network bandwidth changes to test the robustness of composition solutions. Our evaluation re-

Task	F-MA [158]	R-MA	Distance-R-MA
WSC08-1	36865.05 ± 1421	35947.31 ± 1873	34251.9 ± 1549
WSC08-2	29945 ± 1582	29353.48 ± 1396	28751.12 ± 1201
WSC08-3	85491.72 ± 1251	84521.05 ± 1121	82661.65 ± 1020
WSC08-4	19876.24 ± 391	18031.19 ± 242	18134.82 ± 219
WSC08-5	37176.21 ± 334	37011.34 ± 506	35335.75 ± 498
WSC08-6	32545.32 ± 553	33211.45 ± 532	32291.35 ± 412
WSC08-7	44276.23 ± 786	43369.78 ± 812	35303.7 ± 498
WSC08-8	62513.08 ± 1223	59513.08 ± 1023	60070.75 ± 923
WSC09-1	30979 ± 2752	30903.98 ± 2878	26810.24 ± 1392
WSC09-2	55098.29 ± 1165	54786.08 ± 1090	50380.06 ± 9899
WSC09-3	36571.79 ± 1905	32571.66 ± 1011	33450.85 ± 1329
WSC09-4	79598.91 ± 7659	77894.48 ± 5109	76314.09 ± 5487
WSC09-5	74121.09 ± 1953	73904.24 ± 2431	69107.3 ± 2313

Table 7.2: Mean response time [ms] at the execution stage. Wilcoxon test with the significance level of 0.05 is applied.

Table 7.3: Mean cost at the execution stage. Wilcoxon test with significance level of 0.05 is applied.

Task	F-MA [158]	R-MA	Distance-R-MA
WSC08-1	223.54 ± 49	207.73 ± 43	173.3 ± 31
WSC08-2	264.01 ± 15	235.81 ± 15	190.82 ± 12
WSC08-3	279.81 ± 15	276.56 ± 21	231.91 ± 16
WSC08-4	143.79 ± 11	146.24 ± 9	138.54 ± 7
WSC08-5	340.1 ± 23	316.03 ± 55	265.18 ± 49
WSC08-6	567.05 ± 26	560.11 ± 21	394.35 ± 13
WSC08-7	459.11 ± 74	359.11 ± 52	307.92 ± 61
WSC08-8	363.05 ± 31	318.91 ± 34	335.046 ± 29
WSC09-1	232.34 ± 35	172.34 ± 29	119.92 ± 16
WSC09-2	391.78 ± 41	379.94 ± 39	311.44 ± 37
WSC09-3	497.73 ± 105	307.82 ± 99	427.73 ± 115
WSC09-4	850.01 ± 24	871.11 ± 79	705.59 ± 115
WSC09-5	576.09 ± 71	537.05 ± 65	502.43 ± 68

sults verified that Distance-R-MA can generate composite services that are robust to the dynamic network environment (i.e., bandwidth fluctuations). Distance-R-MA can find composition solutions with lower execution cost and response time than solutions generated by existing algorithms. Further, our results showed that our proposed distance-based mutation operator is effective in reducing inter-service distance and therefore, can improve solutions. In the future, we will consider different objectives, e.g., minimising service cost, as well as re-optimisation techniques. CHAPTER 7. DYNAMIC DISTRIBUTED DWSC

166

Chapter 8

Conclusions

The overall goal of this thesis was to propose effective solutions for distributed data-intensive Web service composition problems optimised according to their QoS and communication attributes (see the goal set in Subsection 1.3). This thesis has successfully achieved this goal in several aspects. This thesis has prepared datasets for the distributed DWSC problem formulation in Chapter 3. A bandwidth simulation model on an existing real-world dataset, Curtin, and a formulation on WS-Dream have been used to determine the location information, and consequently, the distance information between services. The bandwidth information was employed to produce communication time. The location information was used to generate the communication cost and, as a heuristic, in guiding several approaches proposed in the next contributions. This thesis has proposed new problem formulations regarding QoS and communication attributes (Chapter 3). During the execution of a data-intensive composite service in a distributed environment, significant time and cost are required to transfer and access data. One of the key importance of this model was dealing with several aspects of the distributed DWSC, such as the location of services. Additionally, this new problem model was extended to define multi-objective distributed DWSC with user preferences, QoSconstrained multi-objective distributed DWSC and dynamic distributed

DWSC. To solve all the modelled problems, several approaches were proposed through the use of specialised initialisation strategies, genetic operators and local search.

The remainder of this chapter is organised as follows. Section 8.1 outlines the objectives that were achieved in this thesis. Section 8.2 presents the main conclusions reached in this work. Section 8.3 explores possible future work directions based on the contributions in this thesis.

8.1 Achieved Objectives

The overall goal of this thesis has been achieved through the following objectives:

- 1. This thesis has proposed effective approaches for performing singleobjective fully automated distributed DWSC (Chapter 4). The first objective of this thesis was to study the use of distribution information of services with GA. This was to design more effective and stable composite services for the single-objective distributed DWSC in Chapter 4. To fulfil this objective, a clustering technique was combined with the location information of services to create an enhanced initial population based on which high-quality solutions were built. Secondly, a heuristic was designed to preserve promising knowledge from one generation to another. This heuristic was then used in several proposed distance-guided crossover operators. Thirdly, multiple distance-guided local search strategies were designed and investigated. These investigations established the distance-based crossover and local search as the most promising EC operators in a single-objective distributed DWSC context. Based on these findings, we further proposed adaptive memetic algorithms (Chapter 5).
- 2. This thesis has proposed a priority-based selection method to select appropriate solutions for the local search in single-objective dis-

8.1. ACHIEVED OBJECTIVES

tributed DDWSC. The priority function was adaptively updated according to the population. This selection method enabled us to develop highly effective algorithms for distributed DWSC by explicitly considering novelty, borrowed from the *novelty search* [108], and improvability of solutions. Similar selection methods that can be consistently integrated with any MA for distributed DWSC had never been performed previously.

Experimental results demonstrated that the priority-based selection method can be hybridised with a range of MAs to find high-quality solutions effectively upon considering the internal structure of solutions, the diversity among them and the likelihood of a solution improvement. Results also showed that the use of communication information in both EC operators and initialisation allowed the MAs to work effectively and improve the quality of solutions.

3. This thesis has proposed multi-objective fully-automated distributed DWSC approaches (Chapter 6). The first approach was to design a memetic NSGA-II to the distributed DWSC for obtaining highperformance composite services. This was achieved by introducing a new local search (i.e., LPLS) to NSGA-II which employed the communication link dominance concept. This investigation established LPLS as the most effective local search technique in the context of the multi-objective distributed DWSC. After developing the effective memetic NSGA-II for the general distributed DWSC problem, the aim was to explore the QoS-constrained distributed DWSC problem. Therefore, DWSCRepair was proposed. This method relies on a set of heuristics to repair solutions. DWSCRepair was also designed to preserve the diversity of the solution space. Finally, this thesis aimed to investigate users' QoS preferences. The lexicographic ordering was proposed to model user preferences, while merged with NSGA-II in several steps to effectively incorporate the user preferences during the evolutionary search. More specifically, hierarchical clustering was employed to both utilise the preferences and improve diversity. This particular problem has never been extensively researched previously. In particular, the proposed approach did not rely on any reference points or user's fine-grained QoS preferences unlike existing similar algorithms proposed for other problems.

4. This thesis has proposed the dynamic distributed DWSC problem, $D^2 - DWSC$ (Chapter 7), which accounts for bandwidth changes. In the static distributed DWSC (Chapters 4 to 6) the bandwidth value of each communication link has been considered fixed. In the dynamic distributed DWSC, the bandwidth of each communication link changes with time. To solve $D^2 - DWSC$, this thesis proposed to formulate it as the problem of optimising the robustness of composite services. Based on the formulation, the proposed algorithm for $D^2 - DWSC$ found a robust composite service during the design stage. Additionally, a novel mutation operator was specifically designed for the distributed DWSC to enhance the service's performance. Experimental results demonstrated that our approach significantly outperformed similar approaches, while both the mutation operator and the robustness optimisation contributed to the better QoS when exposed to bandwidth fluctuations.

8.2 Main Conclusions

This section outlines the main conclusions reached in the major contribution Chapters presented in this thesis (Chapter 3 to Chapter 7).

8.2.1 Problem Formulations

This thesis proposed novel ways to define distributed DWSC in the context of single-objective, multi-objective and dynamic distributed DWSC, which can be properly adjusted to formulate both service-related and network-related attributes. Models were established to determine bandwidth and location information, the two crucial but missing elements in many WSC approaches (see Chapter 2). In practice, it is crucial to consider communication characteristics for many service composition problems. Effective methods for these problems demand proper problem formulation. This model is important for the dynamic distributed DWSC to properly accommodate the fluctuating bandwidth. Additionally, for the multi-objective distributed DWSC with user preferences, where users do not have clear QoS preferences, it is important to formulate the problem using the lexicographic ordering technique. This technique does not require any fine-grained preference information. This idea was straightforward and good enough to represent those preferences, eliminating the need for complex methods that repeatedly deal with users during the search or sophisticated ways to find reference points.

8.2.2 Genetic Operators

This thesis proposed novel approach-specific genetic operators, to be used during the evolutionary process, for modification strategies. In the case of the sequence representation used in this thesis, operations are relatively easy to design, since services can be re-ordered, deleted or inserted in a sequence without violating the correctness of compositions. Additionally, a sequence representation does not consider any service connections since the functional correctness is ensured by the decoding process. Therefore, it is important to design new problem-specific operators that can enhance the performance of the composite service through the proper use of domain-knowledge. A fundamental conclusion is that for the distributed DWSC, it is necessary to define several heuristics that can provide meaningful information to operators and guide the search for promising solutions. The underlying constraint is how to modify the sequence so that the domain-knowledge is used for producing decent offspring solutions.

8.2.3 Heuristics and Domain Knowledge

This thesis proposed the use of heuristics and domain knowledge, the necessary information for finding an optimised and feasible solution. Upon considering this information, we expect to achieve higher-performance methods compared to the general methods proposed for various problems. One of the heuristics was to use the longest common sub-sequence (LCS) of parent solutions in the crossover operator to preserve good subsolutions. Improving the inter-service distance, and hence the QoS, was another heuristic proposed in this thesis. The principal conclusion in crossover operators was to use both the inter-service distance and LCS. However, in the case of mutation and local search operators, where only one parent solution is involved, the inter-service distance was sufficient. Finally, a set of several heuristics was used to implement the repair technique in the QoS-constrained DWSC, which collectively boosted the effectiveness of DWSCRepair method. This heuristic set helped the repair technique choose a Web service in the sequence to be swapped with another service.

8.2.4 Priority-based Selection Measure

This thesis proposed to design a priority-based selection method for local search, which established the adaptive use of local search during the global search process. It is effective and efficient to apply local search on solutions in the population by focusing on the problem itself and utilising information on the problem. Therefore, it is essential to design a method to select solutions for the local search. This problem, which is a predominant difficulty in the local search, has been addressed in this thesis by considering novelty and improvability of solutions. These two measures along with the fitness value of composite services significantly enhanced different local search performances. This priority-based selection method can be applied to many WSC algorithms.

8.2.5 Hybridised Techniques

First of all, this thesis proposed to hybridise EC with the clustering technique in several ways, where clustering supported the use of problemknowledge. For example, the clustering technique produced proper initial solutions in the cluster-guided MA, which consequently was employed in Com-C-D (Chapter 5) to further enhance the performance. In the second place, the clustering was employed to increase the diversity in the multi-objective as well as to deliver users' QoS preferences. The idea of hybridising EC with clustering was simple yet effective and versatile to help algorithms achieve their required goal.

8.2.6 Practical Application of Dynamic approach for Prediction

This thesis proposed a new dynamic distributed DWSC problem with a key focus on bandwidth fluctuations. In this problem, the bandwidth value changes with time. The novelty of this problem lies in that at the design stage bandwidth values are different from the execution stage. This scenario is very likely to happen in real-world applications. Our solution to this problem involved obtaining several bandwidth values through a simulation model and generating robust composite services at the design stage, which are likely to continue to work without considerable degradation in their QoS at the execution stage.

This method can be used in combination with any other methods for dynamic distributed DWSC, such as prediction-based methods, to achieve even better composite services. For example, a robust solution is first generated at the design stage. Afterwards, this composite service is executed with a new bandwidth value at the execution stage, which has been obtained using prediction techniques.

8.3 Future Work

This section discusses some of the open research directions to be tackled by future works in this area, which were identified based on the work presented in this thesis.

8.3.1 Miscellaneous Dynamic Distributed DWSC Problem Model and Methods

Many real-world DWSC problems deal with the dynamic environment, making the dynamic DWSC a prone area to be explored. For example, although the bandwidth is one of the crucial network attributes in the dynamic DWSC [168], many other sources of uncertainty, such as server failure, can be considered in the problem. Another aspect that can be considered is the dynamic DWSC with bandwidth fluctuations, which is the time that the composite solution will be executed. Therefore, a new problem definition to include the time input and new methods to address this problem can be researched. One other aspect worth the investigation is the use of prediction of QoS and network attributes in combination with the designing robust solutions proposed in this thesis.

8.3.2 Learning Techniques

A range of EC operators were designed in this thesis. Multiple local search and crossover operators were employed and examined. Future work in this area could incorporate learning techniques into the composition process, aiming to identify which local search or crossover technique leads to the best possible overall QoS. For example, the priority-based selection method discussed in Chapter 5 would be a good starting point; however, effectively learning and reusing during generations through a *self-adaptive* process could be beneficial to the overall optimisation process.

8.3.3 Miscellaneous QoS Attributes

In this thesis, two QoS attributes are considered, one focused on the service execution cost and the other focused on the execution time in both single-objective and multi-objective problems. Simultaneously optimising more independent functions would require the investigation of appropriate many-objective techniques, which can cope with the additional complexity. Future work could investigate methods for optimising more QoS attributes in both single-objective and multi-objective problems.

8.3.4 Miscellaneous Constraints

In this thesis, QoS-constrained distributed DWSC was examined. A diverse range of other factors may pose constraints on the distributed DWSC problem. Mutual constraints, specifying services that cannot be jointly used, and quantity constraints, that defining an upper bound on the number of services in a composition are worthwhile to be pursued.

CHAPTER 8. CONCLUSIONS

176

Bibliography

- [1] AGAPITO, G. Computer tools to analyze microarray data. In *Microarray Bioinformatics*. Springer, 2019, pp. 267–282.
- [2] AGARWAL, S., AND LAMPARTER, S. User preference based automated selection of Web service compositions. In *ICSOC Workshop on Dynamic Web Processes* (2005), vol. 12, sn, pp. 1–12.
- [3] AGARWAL, S., LAMPARTER, S., AND STUDER, R. Making Web services tradable: A policy-based approach for specifying preferences on Web service properties. *Journal of Web Semantics* 7, 1 (2009), 11–20.
- [4] AL-MASRI, E., AND MAHMOUD, Q. H. Investigating Web services on the World Wide Web. In *Proceedings of the 17th international conference on World Wide Web* (2008), ACM, pp. 795–804.
- [5] ALHADITHY, H., AND AL-SHARGABI, B. Web service composition in cloud: A fuzzy rule model. *Recent Advances in Computer Science* and Communications (Formerly: Recent Patents on Computer Science) 13, 3 (2020), 446–453.
- [6] ALONSO, G., CASATI, F., KUNO, H., AND MACHIRAJU, V. Web services. In *Web Services*. Springer, 2004, pp. 123–149.
- [7] ALRIFAI, M. Distributed and scalable QoS optimization for dynamic Web service composition. In *ICSOC PhD Symposium* (2008).

- [8] ARUNACHALAM, N., AND AMUTHAN, A. A survey on QoS aware global optimization for dynamic Web service composition. In 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN) (2019), IEEE, pp. 1–6.
- [9] ASGHARI, P., RAHMANI, A. M., AND JAVADI, H. H. S. Privacyaware cloud service composition based on QoS optimization in Internet of Things. *Journal of Ambient Intelligence and Humanized Computing* (2020), 1–26.
- [10] AVERSANO, L., DI PENTA, M., AND TANEJA, K. A Genetic Programming approach to support the design of service compositions. *International Journal of Computer Systems Science & Engineering* 21, 4 (2006), 247–254.
- [11] BANSAL, A., BLAKE, M. B., KONA, S., BLEUL, S., WEISE, T., AND JAEGER, M. C. WSC-08: continuing the Web services challenge. In E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on (2008), IEEE, pp. 351–354.
- [12] BEHMANESH, R., RAHIMI, I., AND GANDOMI, A. H. Evolutionary many-objective algorithms for combinatorial optimization problems: A comparative study. *Archives of Computational Methods in Engineering* (2020), 1–16.
- BENOUARET, K., BENSLIMANE, D., HADJALI, A., AND BARHAMGI,
 M. Top-k Web service compositions using fuzzy dominance relationship. In 2011 IEEE International Conference on Services Computing (2011), IEEE, pp. 144–151.
- [14] BENOUARET, K., BENSLIMANE, D., HADJALI, A., BARHAMGI, M., MAAMAR, Z., AND SHENG, Q. Z. Web service compositions

with fuzzy preferences: A graded dominance relationship-based approach. *ACM Transactions on Internet Technology (TOIT)* 13, 4 (2014), 12.

- [15] BHARATHAN, S., RAJENDRAN, C., AND SUNDARRAJ, R. Penalty based mathematical models for Web service composition in a geodistributed cloud environment. In 2017 IEEE International Conference on Web Services (ICWS) (2017), IEEE, pp. 886–889.
- [16] BLUM, A. L., AND FURST, M. L. Fast planning through planning graph analysis. *Artificial intelligence* 90, 1-2 (1997), 281–300.
- [17] BORZSONY, S., KOSSMANN, D., AND STOCKER, K. The skyline operator. In *Data Engineering*, 2001. Proceedings. 17th International Conference on (2001), IEEE, pp. 421–430.
- [18] BOUDRIES, F., SADOUKI, S., AND TARI, A. A bio-inspired algorithm for dynamic reconfiguration with end-to-end constraints in Web services composition. *Service Oriented Computing and Applications* 13, 3 (2019), 251–260.
- [19] BOUSRIH, A., AND BRAHMI, Z. Optimizing cost and response time for data intensive services' composition based on ABC algorithm. In *Information and Communication Technology and Accessibility (ICTA)*, 2015 5th International Conference on (2015), IEEE, pp. 1–6.
- [20] BOUSSALIA, S. R., AND CHAOUI, A. Optimizing QoS-based Web services composition by using quantum inspired cuckoo search algorithm. In *International Conference on Mobile Web and Information Systems* (2014), Springer, pp. 41–55.
- [21] CAI, Z., AND WANG, Y. A multiobjective optimization-based evolutionary algorithm for constrained optimization. *IEEE Transactions* on evolutionary computation 10, 6 (2006), 658–675.

- [22] CANFORA, G., DI PENTA, M., ESPOSITO, R., AND VILLANI, M. L. An approach for QoS-aware service composition based on Genetic Algorithms. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation* (2005), ACM, pp. 1069–1075.
- [23] CARDOSO, J., SHETH, A., MILLER, J., ARNOLD, J., AND KOCHUT, K. Quality of service for workflows and Web service processes. *Journal of web semantics* 1, 3 (2004), 281–308.
- [24] CAVENDISH, D., AND GERLA, M. Internet QoS routing using the Bellman-Ford algorithm. In *High Performance Networking* (1998), Springer, pp. 627–646.
- [25] CENGGORO, T. W., AND SIAHAAN, I. Dynamic bandwidth management based on traffic prediction using deep Long Short Term Memory. In 2016 2nd International Conference on Science in Information Technology (ICSITech) (2016), IEEE, pp. 318–323.
- [26] CHAFLE, G., DOSHI, P., HARNEY, J., MITTAL, S., AND SRIVASTAVA, B. Improved adaptation of Web service compositions using value of changed information. In *IEEE International Conference on Web Services* (*ICWS 2007*) (2007), IEEE, pp. 784–791.
- [27] CHANNABASAVAIAH, K., HOLLEY, K., AND TUGGLE, E. Migrating to a service-oriented architecture. *IBM DeveloperWorks* 16 (2003), 727–728.
- [28] CHATTOPADHYAY, S., AND BANERJEE, A. Qscas: QoS-aware Web service composition algorithms with stochastic parameters. In 2016 IEEE International Conference on Web Services (ICWS) (2016), IEEE, pp. 388–395.
- [29] CHATTOPADHYAY, S., AND BANERJEE, A. QoS constrained large scale Web service composition using abstraction refinement. *IEEE Transactions on Services Computing* (2017).

- [30] CHATTOPADHYAY, S., AND BANERJEE, A. QoS-aware automatic Web service composition with multiple objectives. *ACM Transactions on the Web (TWEB)* 14, 3 (2020), 1–38.
- [31] CHATTOPADHYAY, S., BANERJEE, A., AND BANERJEE, N. A scalable and approximate mechanism for Web service composition. In 2015 IEEE International Conference on Web Services (2015), IEEE, pp. 9–16.
- [32] CHATTOPADHYAY, S., BANERJEE, A., AND BANERJEE, N. A fast and scalable mechanism for Web service composition. *ACM Transactions on the Web (TWEB)* 11, 4 (2017), 26.
- [33] CHEN, C. P., AND ZHANG, C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big data. *Information sciences* 275 (2014), 314–347.
- [34] CHEN, F., DOU, R., LI, M., AND WU, H. A flexible QoS-aware Web service composition method by multi-objective optimization in cloud manufacturing. *Computers & Industrial Engineering* 99 (2016), 423–431.
- [35] CHEN, M., AND YAN, Y. QoS-aware service composition over graphplan through graph reachability. In 2014 IEEE International Conference on Services Computing (2014), IEEE, pp. 544–551.
- [36] CHEN, T., LI, M., AND YAO, X. On the effects of seeding strategies: a case for search-based multi-objective service composition. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2018), pp. 1419–1426.
- [37] CHEN, Y., HUANG, J., LIN, C., AND SHEN, X. Multi-objective service composition with QoS dependencies. *IEEE Transactions on Cloud Computing* (2016).

- [38] CHENG, R., JIN, Y., OLHOFER, M., AND SENDHOFF, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 5 (2016), 773–791.
- [39] CHIFU, V. R., SALOMIE, I., POP, C. B., NICULICI, A. N., AND SUIA, D. S. Exploring the selection of the optimal Web service composition through ant colony optimization. *Computing and Informatics* 33, 5 (2015), 1047–1064.
- [40] CHRISTENSEN, E., CURBERA, F., MEREDITH, G., WEERAWARANA, S., ET AL. Web Services Description Language (wsdl) 1.1, 2001.
- [41] CHU, P. C., AND BEASLEY, J. E. A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics* 4, 1 (1998), 63–86.
- [42] CHU, V. W., WONG, R. K., CHEN, F., AND CHI, C.-H. Service selection based on dynamic QoS networks. In 2016 IEEE International Conference on Services Computing (SCC) (2016), IEEE, pp. 98–105.
- [43] COELLO COELLO, C. A. Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion* (2016), ACM, pp. 563– 587.
- [44] CREMENE, M., SUCIU, M., PALLEZ, D., AND DUMITRESCU, D. Comparative analysis of multi-objective Evolutionary Algorithms for QoS-aware Web service composition. *Applied Soft Computing 39* (2016), 124–139.
- [45] DA SILVA, A. S., MA, H., MEI, Y., AND ZHANG, M. A hybrid memetic approach for fully-automated multi-objective Web service composition. In 2018 IEEE International Conference on Web Services (ICWS) (2018), IEEE, pp. 26–33.

- [46] DA SILVA, A. S., MA, H., MEI, Y., AND ZHANG, M. A survey of evolutionary computation for Web service composition: A technical perspective. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2020).
- [47] DA SILVA, A. S., MA, H., AND ZHANG, M. A GP approach to QoSaware Web service composition and selection. In *Asia-Pacific Conference on Simulated Evolution and Learning* (2014), Springer, pp. 180– 191.
- [48] DA SILVA, A. S., MA, H., AND ZHANG, M. A graph-based Particle Swarm Optimisation approach to QoS-aware Web service composition and selection. In *Evolutionary Computation (CEC)*, 2014 IEEE Congress on (2014), IEEE, pp. 3127–3134.
- [49] DA SILVA, A. S., MA, H., AND ZHANG, M. Graphevol: a graph evolution technique for Web service composition. In *International Conference on Database and Expert Systems Applications* (2015), Springer, pp. 134–142.
- [50] DA SILVA, A. S., MA, H., AND ZHANG, M. Genetic programming for QoS-aware Web service composition and selection. *Soft Computing* 20, 10 (2016), 3851–3867.
- [51] DA SILVA, A. S., MEI, Y., MA, H., AND ZHANG, M. A memetic algorithm-based indirect approach to Web service composition. In *Evolutionary Computation (CEC), IEEE Congress on* (2016).
- [52] DA SILVA, A. S., MEI, Y., MA, H., AND ZHANG, M. Particle swarm optimisation with sequence-like indirect representation for Web service composition. In *European Conference on Evolutionary Computation in Combinatorial Optimization* (2016), Springer, pp. 202–218.
- [53] DA SILVA, A. S., MEI, Y., MA, H., AND ZHANG, M. Fragment-based genetic programming for fully automated multi-objective Web ser-

vice composition. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2017), ACM, pp. 353–360.

- [54] DA SILVA, A. S., MEI, Y., MA, H., AND ZHANG, M. Evolutionary computation for automatic Web service composition: an indirect representation approach. *Journal of Heuristics* 24, 3 (2018), 425–456.
- [55] DA SILVA, A. S., MOSHI, E., MA, H., AND HARTMANN, S. A qos-aware Web service composition approach based on genetic programming and graph databases. In *International Conference on Database and Expert Systems Applications* (2017), Springer, pp. 37–44.
- [56] DAI, Y., YANG, L., AND ZHANG, B. QoS-driven self-healing Web service composition based on performance prediction. *Journal of Computer Science and Technology* 24, 2 (2009), 250–261.
- [57] DAVIS, L. Handbook of Genetic Glgorithms. CumInCAD, 1991.
- [58] DE GIOVANNI, L., AND PEZZELLA, F. An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. *European journal of operational research 200*, 2 (2010), 395–408.
- [59] DEB, K. A robust evolutionary framework for multi-objective optimization. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation* (2008), ACM, pp. 633–640.
- [60] DEB, K., AND DATTA, R. A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach. *Engineering Optimization* 45, 5 (2013), 503–527.
- [61] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitist multi-objective Genetic Algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
- [62] DENG, S., HUANG, L., TAHERI, J., AND ZOMAYA, A. Y. Computation offloading for service workflow in mobile cloud computing.

IEEE Transactions on Parallel and Distributed Systems 26, 12 (2014), 3317–3329.

- [63] DENG, S., HUANG, L., WU, H., AND WU, Z. Constraints-driven service composition in mobile cloud computing. In 2016 IEEE International Conference on Web Services (ICWS) (2016), IEEE, pp. 228–235.
- [64] DINNEEN, M. J., AND WEI, K. On the analysis of a (1+ 1) adaptive memetic algorithm. In 2013 IEEE Workshop on Memetic Computing (MC) (2013), IEEE, pp. 24–31.
- [65] DORIGO, M., BIRATTARI, M., AND STUTZLE, T. Ant Colony Optimization. *IEEE computational intelligence magazine* 1, 4 (2006), 28–39.
- [66] DRUGAN, M. M., AND THIERENS, D. Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies. *Journal of Heuristics* 18, 5 (2012), 727–766.
- [67] DUBOIS-LACOSTE, J., LÓPEZ-IBÁNEZ, M., AND STÜTZLE, T. Combining two search paradigms for multi-objective optimization: Twophase and pareto local search. In *Hybrid Metaheuristics*. Springer, 2013, pp. 97–117.
- [68] EIBEN, A. E., RAUE, P.-E., AND RUTTKAY, Z. Genetic algorithms with multi-parent recombination. In *International Conference on Parallel Problem Solving from Nature* (1994), Springer, pp. 78–87.
- [69] EUSUFF, M., LANSEY, K., AND PASHA, F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization 38*, 2 (2006), 129–154.
- [70] FAN, S.-L., DING, F., GUO, C.-H., AND YANG, Y.-B. Supervised Web service composition integrating multi-objective QoS optimization and service quantity minimization. In *International Conference* on Web Services (2018), Springer, pp. 215–230.

- [71] FANJIANG, Y.-Y., SYU, Y., WU, C.-H., KUO, J.-Y., AND MA, S.-P. Genetic algorithm for QoS-aware dynamic Web services composition. In 2010 International Conference on Machine Learning and Cybernetics (2010), vol. 6, IEEE, pp. 3246–3251.
- [72] GANTZ, J., AND REINSEL, D. As the economy contracts, the digital universe expands. *IDC Multimedia White Paper* (2009).
- [73] GAO, H., ZHANG, K., YANG, J., WU, F., AND LIU, H. Applying improved particle swarm optimization for dynamic service composition focusing on Quality of Service evaluations under hybrid networks. *International Journal of Distributed Sensor Networks* 14, 2 (2018), 1550147718761583.
- [74] GAREY, M. R., JOHNSON, D. S., AND SETHI, R. The complexity of flowshop and jobshop scheduling. *Mathematics of operations research* 1, 2 (1976), 117–129.
- [75] GEIGER, M. J. Decision support for multi-objective flow shop scheduling by the pareto iterated local search methodology. *Computers & industrial engineering* 61, 3 (2011), 805–812.
- [76] GHOBAEI-ARANI, M., AND SOURI, A. LP-WSC: a linear programming approach for Web service composition in geographically distributed cloud environments. *The Journal of Supercomputing* 75, 5 (2019), 2603–2628.
- [77] GLOVER, F. Tabu search—part I. ORSA Journal on computing 1, 3 (1989), 190–206.
- [78] GLOVER, F., LAGUNA, M., AND MARTÍ, R. Fundamentals of scatter search and path relinking. *Control and cybernetics* 29, 3 (2000), 653– 684.
- [79] GOLDENBERG, D. E. Genetic Algorithms in search, optimization and machine learning, 1989.

- [80] GUARDIA, G. D., PIRES, L. F., VÊNCIO, R. Z., MALMEGRIM, K. C., AND DE FARIAS, C. R. A methodology for the development of RESTful semantic Web services for gene expression analysis. *PloS one 10*, 7 (2015), e0134011.
- [81] GUIDARA, I., AL JAOUHARI, I., AND GUERMOUCHE, N. Dynamic selection for service composition based on temporal and QoS constraints. In 2016 IEEE International Conference on Services Computing (SCC) (2016), IEEE, pp. 267–274.
- [82] GUO, H., AND SHU, M. The design of correlation-aware service composition system. *Journal of Physics: Conference Series* 1634, 1 (2020), 012124.
- [83] HABICH, D., RICHLY, S., PREISSLER, S., GRASSELT, M., LEHNER, W., AND MAIER, A. BPEL DT—Data-aware extension for dataintensive service applications. In *Emerging Web Services Technology*, *Volume II*. Springer, 2008, pp. 111–128.
- [84] HAN, X., YUAN, Y., CHEN, C., AND WANG, K. QoS-aware multiobjective optimization algorithm for Web services selection with deadline and budget constraints. *Advances in Mechanical Engineering* 6 (2014), 361298.
- [85] HASSINE, A. B., MATSUBARA, S., AND ISHIDA, T. A constraintbased approach to horizontal Web service composition. In *International semantic Web conference* (2006), Springer, pp. 130–143.
- [86] HO, Y.-C., ZHAO, Q.-C., AND JIA, Q.-S. Vector ordinal optimization. Ordinal Optimization: Soft Optimization for Hard Problems (2007), 93–112.
- [87] HOLLAND, J. H. Genetic algorithms. *Scientific american* 267, 1 (1992), 66–73.

- [88] HOSSEINI SHIRVANI, M. Bi-objective Web service composition problem in multi-cloud environment: a bi-objective time-varying Particle Swarm Optimisation algorithm. *Journal of Experimental & Theoretical Artificial Intelligence* (2020), 1–24.
- [89] HU, L., AND CHE, X. Design and implementation of bandwidth prediction based on grid service. In 2008 10th IEEE International Conference on High Performance Computing and Communications (2008), IEEE, pp. 45–52.
- [90] HUANG, J., LIU, G., DUAN, Q., AND YAN, Y. QoS-aware service composition for converged network-cloud service provisioning. In 2014 IEEE International Conference on Services Computing (2014), IEEE, pp. 67–74.
- [91] HUHNS, M. N., AND SINGH, M. P. Service-oriented computing: Key concepts and principles. *IEEE Internet computing* 9, 1 (2005), 75–81.
- [92] ISHIBUCHI, H., HITOTSUYANAGI, Y., WAKAMATSU, Y., AND NO-JIMA, Y. How to choose solutions for local search in multiobjective combinatorial memetic algorithms. In *International Conference on Parallel Problem Solving from Nature* (2010), Springer, pp. 516–525.
- [93] ISHIBUCHI, H., AND YOSHIDA, T. Hybrid evolutionary multiobjective optimization algorithms. In *HIS* (2002), pp. 163–172.
- [94] JULA, A., OTHMAN, Z., AND SUNDARARAJAN, E. A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition. In *Memetic Computing (MC)*, 2013 *IEEE Workshop on* (2013), IEEE, pp. 37–43.
- [95] JUNGJIT, S., AND FREITAS, A. A lexicographic multi-objective Genetic Algorithm for multi-label correlation based feature selection.

In Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation (2015), pp. 989–996.

- [96] KENNEDY, J. Particle swarm optimization. In *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.
- [97] KLEIN, A., ISHIKAWA, F., AND HONIDEN, S. Efficient heuristic approach with improved time complexity for QoS-aware service composition. In 2011 IEEE International Conference on Web Services (2011), IEEE, pp. 436–443.
- [98] KONA, S., BANSAL, A., BLAKE, M. B., BLEUL, S., AND WEISE, T. WSC-2009: a quality of service-oriented Web services challenge. In *Commerce and Enterprise Computing*, 2009. CEC'09. IEEE Conference on (2009), IEEE, pp. 487–490.
- [99] KOZA, J. R. *Genetic Programming: on the programming of computers by means of natural selection*, vol. 1. MIT press, 1992.
- [100] KOZA, J. R., ET AL. Genetic programming II, vol. 17. MIT press Cambridge, 1994.
- [101] KRASNOGOR, N., AND SMITH, J. A memetic algorithm with selfadaptive local search: TSP as a case study. In *Proceedings of the* 2nd Annual Conference on Genetic and Evolutionary Computation (2000), Morgan Kaufmann Publishers Inc., pp. 987–994.
- [102] KUMAR, S., BAHSOON, R., CHEN, T., LI, K., AND BUYYA, R. Multitenant cloud service composition using evolutionary optimization. In 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS) (2018), IEEE, pp. 972–979.
- [103] KURDOGLU, E., LIU, Y., WANG, Y., SHI, Y., GU, C., AND LYU, J. Real-time bandwidth prediction and rate adaptation for video calls

over cellular networks. In *Proceedings of the 7th International Conference on Multimedia Systems* (2016), pp. 1–11.

- [104] LALEH, T., PAQUET, J., MOKHOV, S., AND YAN, Y. Constraint adaptation in Web service composition. In 2017 IEEE International Conference on Services Computing (SCC) (2017), IEEE, pp. 156–163.
- [105] LAMINE, R. B., JEMAA, R. B., AND AMOR, I. A. B. Graph planning based composition for adaptable semantic Web services. *Procedia Computer Science* 112 (2017), 358–368.
- [106] LAND, M. W. S., AND BELEW, R. K. Evolutionary algorithms with local search for combinatorial optimization. Citeseer, 1998.
- [107] LARRIBA, Y., RUEDA, C., FERNÁNDEZ, M. A., AND PEDDADA, S. D. Microarray data normalization and robust detection of Rhythmic features. In *Microarray Bioinformatics*. Springer, 2019, pp. 207– 225.
- [108] LEHMAN, J., AND STANLEY, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation* 19, 2 (2011), 189–223.
- [109] LI, C., LI, J., AND CHEN, H. A meta-heuristic-based approach for QoS-aware service composition. *IEEE Access 8* (2020), 69579–69592.
- [110] LI, J., YAN, Y., AND LEMIRE, D. Scaling up Web service composition with the skyline operator. In 2016 IEEE International Conference on Web Services (ICWS) (2016), IEEE, pp. 147–154.
- [111] LI, K., CHEN, R., FU, G., AND YAO, X. Two-archive evolutionary algorithm for constrained multi-objective optimization. *IEEE Transactions on Evolutionary Computation* 23, 2 (2018), 303–315.
- [112] LI, L., YANG, P., OU, L., ZHANG, Z., AND CHENG, P. Genetic Algorithm-based multi-objective optimisation for QoS-aware Web
services composition. In *International Conference on Knowledge Science, Engineering and Management* (2010), Springer, pp. 549–554.

- [113] LI, M., AND ZHENG, J. Spread assessment for evolutionary multiobjective optimization. In *International conference on evolutionary multi-criterion optimization* (2009), Springer, pp. 216–230.
- [114] LI, Y., AND LIN, C. QoS-aware service composition for workflowbased data-intensive applications. In Web Services (ICWS), 2011 IEEE International Conference on (2011), IEEE, pp. 452–459.
- [115] LI, Y., LU, Y., YIN, Y., DENG, S., AND YIN, J. Towards QoS-based dynamic reconfiguration of SOA-based applications. In 2010 IEEE Asia-Pacific Services Computing Conference (2010), IEEE, pp. 107–114.
- [116] LIEFOOGHE, A., HUMEAU, J., MESMOUDI, S., JOURDAN, L., AND TALBI, E.-G. On dominance-based multi-objective local search: design, implementation and experimental analysis on scheduling and Traveling Salesman Problems. *Journal of Heuristics* 18, 2 (2012), 317– 352.
- [117] LUDWIG, H., KELLER, A., DAN, A., KING, R. P., AND FRANCK, R. Web service level agreement (WSLA) language specification. *Ibm corporation* (2003), 815–824.
- [118] MA, H., WANG, A., AND ZHANG, M. A hybrid approach using Genetic Programming and greedy search for QoS-aware Web service composition. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XVIII*. Springer, 2015, pp. 180–205.
- [119] MA, H., ZHU, H., LI, K., AND TANG, W. Collaborative optimization of service composition for data-intensive applications in a hybrid cloud. *IEEE Transactions on Parallel and Distributed Systems* 30, 5 (2018), 1022–1035.

- [120] MARTÍ, R., AND REINELT, G. Heuristic methods. In *The Linear Ordering Problem*. Springer, 2011, pp. 17–40.
- [121] MENA, F. M., UCAN, R. H., CETINA, V. U., AND RAMIREZ, F. M. Web service composition using the bidirectional Dijkstra algorithm. *IEEE Latin America Transactions* 14, 5 (2016), 2522–2528.
- [122] MEZURA-MONTES, E., AND COELLO, C. A. C. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary computation* 9, 1 (2005), 1–17.
- [123] MICHALEWICZ, Z. Heuristic methods for Evolutionary Computation techniques. *Journal of Heuristics* 1, 2 (1996), 177–206.
- [124] MIN, X., XU, X., AND WANG, Z. Combining von Neumann neighborhood topology with approximate-mapping local search for ABCbased service composition. In 2014 IEEE International Conference on Services Computing (2014), IEEE, pp. 187–194.
- [125] MISTRY, S., BOUGUETTAYA, A., DONG, H., AND QIN, A. K. Metaheuristic optimization for long-term IaaS service composition. *IEEE Transactions on Services Computing* 11, 1 (2016), 131–143.
- [126] MIYAKAWA, M., SATO, H., AND SATO, Y. A study for parallelization of multi-objective evolutionary algorithm based on decomposition and directed mating. In *Proceedings of the 2019 3rd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence* (2019), ACM, pp. 25–29.
- [127] MOGHADDAM, M., AND DAVIS, J. G. Service selection in Web service composition: A comparative review of existing approaches. In *Web Services Foundations*. Springer, 2014, pp. 321–346.
- [128] MOHR, F., JUNGMANN, A., AND BÜNING, H. K. Automated online service composition. In 2015 IEEE International Conference on Services Computing (2015), IEEE, pp. 57–64.

- [129] MOHSNI, T., BRAHMI, Z., AND GAMMOUDI, M. M. Data-intensive service composition in cloud computing: State-of-the-art. In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA) (2016), IEEE, pp. 1–7.
- [130] MONACO, A., PANTALEO, E., AMOROSO, N., BELLANTUONO, L., LOMBARDI, A., TATEO, A., TANGARO, S., AND BELLOTTI, R. Identifying potential gene biomarkers for Parkinson's disease through an information entropy based approach. *Physical Biology* (2020).
- [131] MOSCATO, P., COTTA, C., AND MENDES, A. Memetic Algorithms. In *New optimization techniques in engineering*. Springer, 2004, pp. 53– 85.
- [132] MOSCATO, P., ET AL. On evolution, search, optimization, Genetic Algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report 826* (1989), 1989.
- [133] MOSTAFA, A., AND ZHANG, M. Multi-objective service composition in uncertain environments. *IEEE Transactions on Services Computing* (2015).
- [134] MUCIENTES, M., LAMA, M., AND COUTO, M. I. A Genetic Programming-based algorithm for composing Web services. In 2009 Ninth International Conference on Intelligent Systems Design and Applications (2009), IEEE, pp. 379–384.
- [135] NGUYEN, S., MEI, Y., AND ZHANG, M. Genetic Programming for production scheduling: a survey with a unified framework. *Complex* & Intelligent Systems 3, 1 (2017), 41–66.
- [136] OLAIFA, M., AND ZUVA, T. Causal path planning graph based on semantic pre-link computation for Web service composition. In 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS) (2019), IEEE, pp. 1–5.

- [137] OPITZ, S. J., TODTENBERG, N., AND KÖNIG, H. Mobile bandwidth prediction in the context of emergency medical service. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments* (2014), pp. 1–7.
- [138] OSMAN, I. H., AND KELLY, J. P. Meta-heuristics: an overview. In *Meta-heuristics*. Springer, 1996, pp. 1–21.
- [139] O'SULLIVAN, J., EDMOND, D., AND TER HOFSTEDE, A. What's in a Service? *Distributed and Parallel Databases* 12, 2-3 (2002), 117–133.
- [140] ÖZCAN, E., DRAKE, J. H., ALTINTAŞ, C., AND ASTA, S. A selfadaptive multimeme memetic algorithm co-evolving utility scores to control genetic operators and their parameter settings. *Applied Soft Computing* 49 (2016), 81–93.
- [141] PAPAZOGLOU, M. P. Service-oriented computing: Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, 2003. WISE 2003. (2003), IEEE, pp. 3–12.
- [142] PAQUETE, L., CHIARANDINI, M., AND STÜTZLE, T. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In *Metaheuristics for multiobjective optimisation*. Springer, 2004, pp. 177–199.
- [143] PÁRRAGA-ALAVA, J., DORN, M., AND INOSTROZA-PONTA, M. Using local search strategies to improve the performance of NSGA-II for the multi-criteria minimum spanning tree problem. In 2017 IEEE Congress on Evolutionary Computation (CEC) (2017), IEEE, pp. 1119– 1126.
- [144] PETERSON, J. L. Petri nets. *ACM Computing Surveys (CSUR)* 9, 3 (1977), 223–252.

- [145] PRANDL, S., LAZARESCU, M., PHAM, D. S., SOH, S. T., AND KAK, S. An investigation of power law probability distributions for network anomaly detection. In 2017 IEEE Security and Privacy Workshops (SPW) (2017), IEEE, pp. 217–222.
- [146] PUROHIT, L., CHOUHAN, S. S., AND JAIN, A. Dynamic Web service composition using AI planning technique: Case study on Blackbox planner. In *Social Networking and Computational Intelligence*. Springer, 2020, pp. 183–195.
- [147] RAHI, K. H., SINGH, H. K., AND RAY, T. Investigating the use of sequencing and infeasibility driven strategies for constrained optimization. In 2019 IEEE Congress on Evolutionary Computation (CEC) (2019), IEEE, pp. 1642–1649.
- [148] RAJENDRAN, V., CHUA, F.-F., AND CHAN, G.-Y. Self-healing in dynamic Web service composition. In 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud) (2017), IEEE, pp. 206–211.
- [149] RAJESWARY, C. A survey on efficient evolutionary algorithms for Web service selection. *International Journal of Managment, IT and En*gineering 2, 9 (2012), 177–191.
- [150] RAMÍREZ, A., PAREJO, J. A., ROMERO, J. R., SEGURA, S., AND RUIZ-CORTÉS, A. Evolutionary composition of QoS-aware Web services: a many-objective perspective. *Expert Systems with Applications* 72 (2017), 357–370.
- [151] RAO, J., AND SU, X. A survey of automated Web service composition methods. In *International Workshop on Semantic Web Services and Web Process Composition* (2004), Springer, pp. 43–54.
- [152] RODRIGUEZ-MIER, P., MUCIENTES, M., AND LAMA, M. Automatic Web service composition with a heuristic-based search algorithm.

In 2011 IEEE International Conference on Web Services (2011), IEEE, pp. 81–88.

- [153] RODRIGUEZ-MIER, P., MUCIENTES, M., AND LAMA, M. Hybrid optimization algorithm for large-scale QoS-aware service composition. *IEEE transactions on services computing* 10, 4 (2015), 547–559.
- [154] RODRIGUEZ-MIER, P., MUCIENTES, M., LAMA, M., AND COUTO, M. I. Composition of Web services through Genetic Programming. *Evolutionary Intelligence* 3, 3-4 (2010), 171–186.
- [155] ROSEN, C. A new approach to software systems development. In *Guide to Software Systems Development*. Springer, 2020, pp. 1–16.
- [156] ROSS, N., KEEDWELL, E., AND SAVIC, D. Human-derived heuristic enhancement of an Evolutionary algorithm for the 2D bin-packing problem. In *International Conference on Parallel Problem Solving from Nature* (2020), Springer, pp. 413–427.
- [157] SADEGHIRAM, S., MA, H., AND CHEN, G. Cluster-guided Genetic Algorithm for distributed data-intensive Web service composition. *Evolutionary Computation (CEC)*, 2018 IEEE Congress on (2018).
- [158] SADEGHIRAM, S., MA, H., AND CHEN, G. Composing distributed data-intensive Web services using a flexible memetic algorithm. In 2019 IEEE Congress on Evolutionary Computation (CEC) (2019), IEEE, pp. 2832–2839.
- [159] SADEGHIRAM, S., MA, H., AND CHEN, G. Composing distributed data-intensive Web services using distance-guided memetic algorithm. In *International Conference on Database and Expert Systems Applications* (2019), Springer, pp. 411–422.
- [160] SADEGHIRAM, S., MA, H., AND CHEN, G. A novel repair-based multi-objective algorithm for QoS-Constrained distributed data-

intensive Web service composition. In *International Conference on Web Information Systems Engineering* (2020), Springer, pp. 489–502.

- [161] SADIQ, U., KUMAR, M., PASSARELLA, A., AND CONTI, M. Service composition in opportunistic networks: A load and mobility aware solution. *IEEE Transactions on Computers* 64, 8 (2014), 2308–2322.
- [162] SAID, L. B., BECHIKH, S., AND GHÉDIRA, K. The r-dominance: a new dominance relation for interactive evolutionary multi-criteria decision making. *IEEE transactions on Evolutionary Computation* 14, 5 (2010), 801–818.
- [163] SALCEDO-SANZ, S. A survey of repair methods used as constraint handling techniques in evolutionary algorithms. *Computer science review* 3, 3 (2009), 175–192.
- [164] SAMANIPOUR, F., AND JELOVICA, J. Adaptive repair method for constraint handling in multi-objective Genetic Algorithm based on relationship between constraints and variables. *Applied Soft Computing 90* (2020), 106143.
- [165] SAWCZUK DA SILVA, A., MA, H., AND ZHANG, M. A graph-based QoS-Aware method for Web service composition with branching. In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (2016), pp. 131–132.
- [166] SEGHIR, F., AND KHABABA, A. A hybrid approach using Genetic and fruit fly optimization algorithms for QoS-aware cloud service composition. *Journal of Intelligent Manufacturing* 29, 8 (2018), 1773– 1792.
- [167] SHE, Q., WEI, X., NIE, G., AND CHEN, D. QoS-aware cloud service composition: A systematic mapping study from the perspective of computational intelligence. *Expert Systems with Applications* 138 (2019), 112804.

- [168] SHEN, Y., ZHU, J., WANG, X., CAI, L., YANG, X., AND ZHOU, B. Geographic location-based network-aware QoS prediction for service composition. In 2013 IEEE 20th International Conference on Web Services (2013), IEEE, pp. 66–74.
- [169] SHI, J., ZHANG, Q., DERBEL, B., LIEFOOGHE, A., AND SUN, J. Parallel Pareto local search revisited: first experimental results on biobjective UBQP. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2018), ACM, pp. 753–760.
- [170] SHI, Y., ET AL. Particle swarm optimization: developments, applications and resources. In *evolutionary computation*, 2001. Proceedings of the 2001 Congress on (2001), vol. 1, IEEE, pp. 81–86.
- [171] SHIRVANI, M. H. Web service composition in multi-cloud environment: A bi-objective Genetic optimization algorithm. In 2018 Innovations in Intelligent Systems and Applications (INISTA) (2018), IEEE, pp. 1–6.
- [172] SINGH, H. K., ISAACS, A., RAY, T., AND SMITH, W. Infeasibility driven evolutionary algorithm (IDEA) for engineering design optimization. In *Australasian Joint Conference on Artificial Intelligence* (2008), Springer, pp. 104–115.
- [173] SOHRABI, S., AND MCILRAITH, S. A. Preference-based Web service composition: A middle ground between execution and search. In *International Semantic Web Conference* (2010), Springer, pp. 713–729.
- [174] SOHRABI, S., PROKOSHYNA, N., AND MCILRAITH, S. A. Web service composition via generic procedures and customizing user preferences. In *International Semantic Web Conference* (2006), Springer, pp. 597–611.

- [175] SRIVASTAVA, B., AND KOEHLER, J. Web service composition-current solutions and open problems. In *ICAPS 2003 workshop on Planning for Web Services* (2003), vol. 35, pp. 28–35.
- [176] STORN, R., AND PRICE, K. Differential Evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.
- [177] SUN, X., CHEN, J., XIA, Y., HE, Q., WANG, Y., LUO, X., ZHANG, R., HAN, W., AND WU, Q. A fluctuation-aware approach for predictive Web service composition. In 2018 IEEE International Conference on Services Computing (SCC) (2018), IEEE, pp. 121–128.
- [178] TAN, K.-L., ENG, P.-K., OOI, B. C., ET AL. Efficient progressive skyline computation. In *VLDB* (2001), vol. 1, pp. 301–310.
- [179] TANG, M., AND AI, L. A hybrid Genetic Algorithm for the optimal constrained Web service selection problem in Web service composition. In *IEEE Congress on Evolutionary Computation* (2010), IEEE, pp. 1–8.
- [180] TAYARANI-N, M.-H. Novel operators for quantum Evolutionary algorithm in solving timetabling problem. *Evolutionary Intelligence* (2020), 1–25.
- [181] TRANG, M. X., MURAKAMI, Y., AND ISHIDA, T. Policy-aware optimization of parallel execution of composite services. In 2015 IEEE International Conference on Services Computing (2015), IEEE, pp. 106– 113.
- [182] VERMA, D. C. Supporting service level agreements on IP networks. Citeseer, 1999.
- [183] WADA, H., SUZUKI, J., YAMANO, Y., AND OBA, K. E³: a multiobjective optimization framework for SLA-aware service composition. *IEEE Transactions on Services Computing* 5, 3 (2012), 358–372.

- [184] WAGNER, F., ISHIKAWA, F., AND HONIDEN, S. Robust service compositions with functional and location diversity. *IEEE Transactions* on Services Computing 9, 2 (2013), 277–290.
- [185] WANG, A., MA, H., AND ZHANG, M. Genetic Programming with greedy search for Web service composition. In *International Conference on Database and Expert Systems Applications* (2013), Springer, pp. 9–17.
- [186] WANG, C., MA, H., CHEN, A., AND HARTMANN, S. Comprehensive Quality-aware automated semantic Web service composition. In Australasian Joint Conference on Artificial Intelligence (2017), Springer, pp. 195–207.
- [187] WANG, C., MA, H., CHEN, A., AND HARTMANN, S. GP-based approach to comprehensive Quality-aware automated semanticWeb service composition. In *Asia-Pacific Conference on Simulated Evolution* and Learning (2017), Springer, pp. 170–183.
- [188] WANG, C., MA, H., CHEN, A., AND HARTMANN, S. Knowledgedriven automated Web service composition—an EDA-based approach. In *International Conference on Web Information Systems En*gineering (2018), Springer, pp. 135–150.
- [189] WANG, C., MA, H., AND CHEN, G. Eda-based approach to comprehensive quality-aware automated semantic Web service composition. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (2018), pp. 147–148.
- [190] WANG, C., MA, H., AND CHEN, G. Using EDA-based local search to improve the performance of NSGA-II for multiobjective semantic Web service composition. In *International Conference on Database and Expert Systems Applications* (2019), Springer, pp. 434–451.

- [191] WANG, C., MA, H., CHEN, G., AND HARTMANN, S. Towards robust Web service composition with stochastic service failures based on a genetic algorithm. In *Australasian Joint Conference on Artificial Intelligence* (2019), Springer, pp. 445–459.
- [192] WANG, D., YANG, Y., AND MI, Z. A genetic-based approach to Web service composition in geo-distributed cloud environment. *Comput*ers & Electrical Engineering 43 (2015), 129–141.
- [193] WANG, H., ZOU, B., GUO, G., YANG, D., AND ZHANG, J. Integrating trust with user preference for effective Web service composition. *IEEE Transactions on Services Computing* 10, 4 (2015), 574–588.
- [194] WANG, H., ZOU, B., GUO, G., ZHANG, J., AND YANG, Z. Optimal and effective Web service composition with trust and user preference. In 2015 IEEE International Conference on Web Services (2015), IEEE, pp. 329–336.
- [195] WANG, L., AND SHEN, J. Multi-phase Ant colony system for multiparty data-intensive service provision. *IEEE Transactions on Services Computing* 9, 2 (2016), 264–276.
- [196] WANG, L., AND SHEN, J. Data-intensive service provision based on Particle Swarm Optimization. *International Journal of Computational Intelligence Systems* 11, 1 (2018), 330–339.
- [197] WANG, L., SHEN, J., AND BEYDOUN, G. Enhanced ant colony algorithm for cost-aware data-intensive service provision. In *Services* (SERVICES), 203 IEEE Ninth World Congress on (2013), IEEE, pp. 227– 234.
- [198] WANG, L., SHEN, J., AND LUO, J. Impacts of pheromone modification strategies in ant colony for data-intensive service provision. In *Web Services (ICWS)*, 2014 IEEE International Conference on (2014), IEEE, pp. 177–184.

- [199] WANG, L., SHEN, J., LUO, J., AND DONG, F. An improved genetic algorithm for cost-effective data-intensive service composition. In *Semantics, Knowledge and Grids (SKG), 2013 Ninth International Conference on* (2013), IEEE, pp. 105–112.
- [200] WANG, X., ZHU, J., AND SHEN, Y. Network-aware QoS prediction for service composition using geolocation. *IEEE Transactions on Services Computing* 8, 4 (2014), 630–643.
- [201] WANG, Y., AND CAI, Z. A dynamic hybrid framework for constrained evolutionary optimization. *IEEE Transactions on Systems*, *Man, and Cybernetics, Part B (Cybernetics)* 42, 1 (2011), 203–217.
- [202] WANG, Z., CHENG, B., ZHANG, W., AND CHEN, J. Q-graphplan: QoS-aware automatic service composition with the extended planning graph. *IEEE Access 8* (2020), 8314–8323.
- [203] WEERAWARANA, S., CURBERA, F., LEYMANN, F., STOREY, T., AND FERGUSON, D. F. Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. Prentice Hall PTR, 2005.
- [204] WEISE, T., BLEUL, S., COMES, D., AND GEIHS, K. Different approaches to semantic Web service composition. In 2008 Third International Conference on Internet and Web Applications and Services (2008), IEEE, pp. 90–96.
- [205] WU, Z., LIN, P., HUANG, P., PENG, H., HE, Y., AND CHEN, J. A user constraint awareness approach for QoS-based service composition. In *International Conference on Web Services* (2019), Springer, pp. 48–62.
- [206] XIA, Y.-M., AND YANG, Y.-B. Web service composition integrating QoS optimization and redundancy removal. In 2013 IEEE 20th International Conference on Web Services (2013), IEEE, pp. 203–210.

- [207] XIANG, Y., ZHOU, Y., YANG, X., AND HUANG, H. A manyobjective evolutionary algorithm with Pareto-adaptive reference points. *IEEE Transactions on Evolutionary Computation* (2019).
- [208] XIAO, L., CHANG, C. K., YANG, H.-I., LU, K.-S., AND JIANG, H.-Y. Automated Web service composition using Genetic Programming. In 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops (2012), IEEE, pp. 7–12.
- [209] XU, B., ZHANG, H., ZHANG, M., AND LIU, L. Differential evolution using cooperative ranking-based mutation operators for constrained optimization. *Swarm and Evolutionary Computation* 49 (2019), 206–219.
- [210] YAGHOUBI, M., AND MAROOSI, A. Simulation and modeling of an improved multi-verse optimization algorithm for QoS-aware Web service composition with service level agreements in the cloud environments. *Simulation Modelling Practice and Theory* (2020), 102090.
- [211] YAN, J., LI, M., AND XU, J. An adaptive strategy applied to memetic algorithms. *IAENG International Journal of Computer Science* 42, 2 (2015).
- [212] YAN, L., MEI, Y., MA, H., AND ZHANG, M. Evolutionary Web service composition: a graph-based memetic algorithm. In 2016 IEEE Congress on Evolutionary Computation (CEC) (2016), IEEE, pp. 201–208.
- [213] YAN, Y., AND CHEN, M. Anytime QoS-aware service composition over the GraphPlan. Service Oriented Computing and Applications 9, 1 (2015), 1–19.
- [214] YANNIBELLI, V., AND AMANDI, A. Project scheduling: A memetic algorithm with diversity-adaptive components that optimizes the effectiveness of human resources. *Polibits*, 52 (2015), 93–103.

- [215] YAO, J., CHEN, S., NEPAL, S., LEVY, D., AND ZIC, J. Truststore: Making Amazon S3 trustworthy with services composition. In 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (2010), IEEE, pp. 600–605.
- [216] YAO, T., YAO, X., HAN, S., WANG, Y., CAO, D., AND WANG, F. Memetic algorithm with adaptive local search for capacitated arc routing problem. In *Intelligent Transportation Systems (ITSC)*, 2017 *IEEE 20th International Conference on* (2017), IEEE, pp. 836–841.
- [217] YI, J., BAI, J., HE, H., PENG, J., AND TANG, D. ar-MOEA: A novel preference-based dominance relation for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 23, 5 (2018), 788–802.
- [218] YU, D., LI, C., YIN, Y., ET AL. Optimizing Web service composition for data-intensive applications. *International Journal of Database Theory and Application 7*, 2 (2014), 1–12.
- [219] YU, Y., MA, H., AND ZHANG, M. An adaptive genetic programming approach to QoS-aware Web services composition. In *Evolutionary Computation (CEC), 2013 IEEE Congress on* (2013), IEEE, pp. 1740–1747.
- [220] YU, Y., MA, H., AND ZHANG, M. A hybrid GP-Tabu approach to QoS-aware data-intensive web service composition. In *Asia-Pacific Conference on Simulated Evolution and Learning* (2014), Springer, pp. 106–118.
- [221] YU, Y., MA, H., AND ZHANG, M. F-MOGP: A novel many-objective evolutionary approach to QoS-aware data intensive Web service composition. In *Evolutionary Computation (CEC)*, 2015 IEEE Congress on (2015), IEEE, pp. 2843–2850.

- [222] YU, Y., MA, H., AND ZHANG, M. A genetic programming approach to distributed execution of data-intensive web service compositions. In *Proceedings of the Australasian Computer Science Week Multiconference* (2016), pp. 1–9.
- [223] YUAN, Y., XU, H., WANG, B., AND YAO, X. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 1 (2015), 16– 37.
- [224] ZHANG, Q., AND LI, H. MOEA/D: A multiobjective Evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.
- [225] ZHENG, S., FENG, D., AND YU, J. The algorithm of Web services composition of group decision-making based on fuzzy numbers. In *International Conference on Computer Science and Application Engineering* (2019), ACM, pp. 1–7.
- [226] ZHENG, Z., AND LYU, M. R. WS-dream: A distributed reliability assessment mechanism for Web services. In *Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on* (2008), IEEE, pp. 392–397.
- [227] ZHENG, Z., ZHANG, Y., AND LYU, M. R. Investigating QoS of real-world Web services. *IEEE transactions on services computing* 7, 1 (2014), 32–39.
- [228] ZHU, M., FAN, G., LI, J., AND KUANG, H. An approach for QoSaware service composition with graphplan and fuzzy logic. *Procedia Computer Science* 141 (2018), 56–63.
- [229] ZITZLER, E., LAUMANNS, M., AND THIELE, L. SPEA2: Improving the strength Pareto Evolutionary algorithm. *TIK-report 103* (2001).