Multiagent Evolutionary Computation for Complex Problems

NANYANG TECHNOLOGICAL UNIVERSITY

A thesis
submitted to the Nanyang Technological University
in fulfilment of the requirement
for the degree of
Doctor of Philosophy of

JIANG SIWEI

April 2014

Abstract

Multiagent evolutionary computation (MAEC) is a new paradigm to efficiently solve a range of complex problems, by combining the advantages of *evolutionary computation* (EC) and *multiagent systems* (MAS). In general, there are three categories in MAEC: 1) "agent based EC" incorporates characteristics of intelligent agents, and adopts techniques in MAS to enhance the performance of EC; 2) "evolutionary computation based MAS" utilizes evolutionary techniques to design self-learning agents, to improve the efficiency of interactions between agents, and to increase the effectiveness of the whole system; 3) "the sequential or embedded approach of EC and MAS" employs EC and MAS in either a sequential order or an embedded fashion. The MAEC approaches have been successfully applied to solve benchmark and real world problems. However, most studies in the first category of MAEC focus only on single-objective optimization problems. In the second group of MAEC, the information communicated between agents is often assumed to be truthful. That is, the agents in MAS are assumed to be honest and cooperative. In addition, the two components of EC and MAS have not been integrated effectively in the third direction of MAEC. Thus, the ability of MAEC to solve complex problems has not been fully exploited by the previous studies.

In view of this, this thesis proposes three approaches in the respective categories of MAEC for solving three complex problems, which are multiobjective optimization problems (MOPs), robustness of trust in MAS based e-marketplaces and traffic management in transportation systems.

Firstly, a multiagent evolutionary framework based on trust is proposed to solve MOPs. This work lies into the first category of MAEC. In particular, the trust concept that is popular in MAS is adopted for measuring the dynamic competency of services (*i.e.*, the pairs of evolutionary operators and control parameters) from generation to

generation and on different problems. Then, candidate solutions modeled as intelligent agents will select the services with the probabilities correlated to the trustworthiness of the services. Experimental results demonstrate that the proposed framework significantly improves the performance of the state-of-the-art multiobjective evolutionary algorithms (MOEAs).

Secondly, a multiagent evolutionary trust model (MET) is proposed to address the robustness problems in MAS based e-marketplaces, where advisor agents may not provide truthful information about seller agents. This work falls into the second group of MAEC. In MET, each buyer agent evolves its trust network (consisting of information about which advisor agents to include in the network and their trustworthiness) over time and finally constructs accurate and robust trust networks. Experimental results on a multiagent-based e-marketplace testbed show that MET is more robust than existing trust models in resisting typical attacks.

Thirdly, a multiagent pheromone-based traffic management framework is proposed for reducing traffic congestion. This study belongs to the third direction of MAEC. In particular, a new digital "pheromone" inspired by EC is defined for bridging MAS based vehicle rerouting and traffic light control. Once congestion is predicted for a road, a proactive vehicle rerouting algorithm is designed for assigning alternative routes to cars before they enter the congested road. At the same time, an online traffic light control strategy is proposed to assign long time duration of green traffic lights to the roads with a large amount of pheromone. Experimental results show that the proposed framework significantly alleviates traffic congestion, saves travel time, and reduces air pollution and fuel consumption.

To summarize, with the three kinds of effective MAEC algorithms for solving complex problems, this thesis serves to elicit more research efforts for building powerful MAEC approaches by delicately fusing the high level knowledge and advantages of EC and MAS, and applying them to solve large, open, dynamic, distributed complex problems. This work will also nourish the two related fields towards more intelligent EC and effective MAS.

Acknowledgments

I would like to express my sincere thanks to my supervisors Prof. Jie Zhang and Prof. Yew-Soon Ong from School of Computer Engineering, who have continuously encouraged, inspired and supported me throughout my Ph.D. research career.

My immense gratitude goes to my wife (Ms. Dan Zeng), my parents and my two elder sisters for their great love and continuous encouragement in both my study and life.

I also want to thank my colleagues, Ms. Yuan Liu, Ms. Hui Fang, Mr. Guibing Guo, Ms. Athirai Aravazhi Irissappane, Mr. Lizi Zhang, Mr. Chang Xu, Ms. Huanhuan Zhang, Ms. Xuan Liu and Mr. Qin Li. Discussions with them were really fruitful and pleasant. Without them, the research cannot be such an enriching and enjoyable experience. Also, many thanks to my good friends, Mr. Qi Mao, Mr. Liang Feng, Mr. Mingkui Tang, Mr. Yu Zhang, Mr. Hao, Yang, Mr. Yiteng Zai, Mr. Tianyi Zhou, Mr. Jianbo Yang, Mr. Qiaoliang Xiang, Mr. Xinming Zhang, Mr. Chunwei Seah and Mr. Giduthuri Sateesh babu for their friendship and supports.

Last but not least, I would also like to thank the technicians, Mr. Kesavan Asaithambi and Ms. Linda Ang, for their technical supports and enthusiastic help in the Center for Computational Intelligence (C2I) which I have been associated with.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many real world systems become very complex, involving multiple design parameters and a large number of entities with various characteristics (*e.g.*, capabilities, available actions, roles, etc.). These are often open systems, meaning that the entities can join or leave the systems at any time. Some of these systems are also distributed, without a centralized authority to control everything. In addition, these systems may exhibit many dynamics caused by the constant changes of environments and involved entities. Consequently, the complexity of the systems impose complex problems to be effectively solved.

As the first example, multiobjective optimization problems (MOPs) include several conflicting objectives to be optimized simultaneously [2–16]. Many real world problems can be formulated as MOPs [2–7] that are now prevalent in the fields of engineering, finance, logisticss, game design, control, etc. In comparison to single-objective optimization problems that search a *unique optimal solution* in the direction of maximization or minimization, MOPs often obtain an *optimal solution set* comprising a number of solutions that are fair equivalent or incomparable according to the Pareto dominance concept [2–7]. For instance, car companies wish to produce a car satisfying two conflicting goals, such as maximizing performance while minimizing fuel consumption. For customers, they make decisions of whether to purchase a car by considering the above two criteria, *e.g.*, choosing either a good performance car with high fuel consumption or an economical car with low fuel consumption. The challenge in MOPs is to find a *Pareto Set* (PS) involving non-dominated solutions that are evenly scattered along the *Pareto Front* (PF) [2–16]. Taking Fig. 1.1 as an illustrative example, the Pareto set for minimizing a bi-objective MOP is PS $= \{A, B, C\}$, where solutions are non-dominated by each other. For instance, point A is non-dominated by point B, since $f_1(A) < f_2(B)$ and $f_2(A) > f_2(B)$. On the other hand, the solutions in PS should be diversified, which is benefit for providing multiple choices to users.

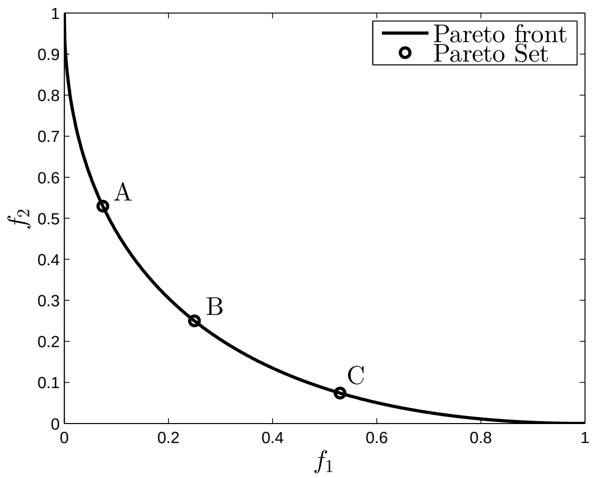


Figure 1.1: The example of minimizing a bi-objective MOP, PS = $\{A, B, C\}$ along PF.

As the second example, trust and reputation play a critical role in e-commerce systems, *e.g.*, taobao[1], ebay[2], amazon[3] [17–30]. When a buyer wants to purchase products from a potential seller, the buyer will rely on its own experience built from the past transactions with the seller. However, it is often the case where the buyer has little or no experience with the seller. Thus, the buyer has to ask opinions from other buyers (called advisors). In the large, open, dynamic and distributed e-commerce systems, advisors may exhibit various behavior patterns, such as honestly sharing information with others, randomly posting opinions about the products that they do not care, maliciously providing unfair ratings to sellers, etc. For instance, a seller may hire some advisors to provide unfairly high ratings ("ballot stuffing") to promote its own products, or give unfairly low ratings ("bad-mouthing") to degrade its competitors. Another interesting example is that a buyer re-enters e-marketplaces with a newly created account to whitewash its past

[1] `http://sea.taobao.com/`
[2] `http://www.ebay.com/`
[3] `http://www.amazon.com/`

dishonest behaviors. According to [25, 26], there are several types of unfair rating attacks in e-marketplaces, *e.g.*, Constant attack, Camouflage attack, Whitewashing attack, Sybil attack, Sybil Camouflage attack, Sybil Whitewashing attack, etc. The challenge in e-marketplaces is thus to build an effective and robust trust model in resisting various strategic attacks.

As the third example, traffic congestion is a key problem to cause driver frustration, air pollution, economical problems in today's urban areas [31–42]. Broadening roads brings high cost for deconstructing existing buildings. A more attractive solution is to develop effective computer algorithms for rerouting vehicles and controlling traffic lights. In particular, vehicle rerouting searches alternative paths for cars when traffic congestion is found or predicted. However, it is nearly impossible to apply traditional network analysis tools [43, 44] into real transportation systems, since they need to build mathematical formulations for modeling a large number of cars in a big road network. On the other hand, traffic light control automatically sets color phases (*i.e.*, red, yellow and green) and time duration of these phases according to dynamic traffic conditions. However, the fixed traffic light control rules built from expert knowledge [39] face problems for adapting dynamic traffic scenarios. For instance, traffic light control rules should be different in rush hours and off-peak hours, days and nights, working days and holidays, etc. In addition, the two components, *i.e.*, vehicle rerouting and traffic light control, are correlated with each other. For instance, to find optimal paths, vehicle rerouting should consider the waiting time when traffic lights are red. To maximize road capacities, traffic light control needs to assign more/less time duration of green traffic lights to roads with higher/lower traffic volume. Thus, it is challenging to design a unified framework for simultaneously optimizing vehicle rerouting and traffic light control.

1.1 Solutions from Artificial Intelligence

To date, a plethora of techniques in *artificial intelligence* (AI), *e.g.*, evolutionary computation (EC) and multiagent systems (MAS), have been proposed for tackling complex problems [45]. The definitions of EC and MAS are briefly introduced below.

Evolutionary computation (EC) is a kind of optimization methodology inspired by the mechanism of biological evolution and the behaviors of living organism [1, 46]. It maintains a population of candidate solutions to a certain problem. Every solution consists of several decision variables. In general, the initial searching population is generated by simple random, advanced techniques (*e.g.*, experimental design methods [47]), etc. An evaluation function or *fitness function* is designed to measure the quality of each candidate solution. From generation to generation, EC applies the Darwin's principle of "the survival of fittest" among population with the stochastic processes of crossover, mutation, and/or local search, etc [48, 49]. In essence, the population of candidate solutions is seen to adapt the problem space.

A *multiagent system* (MAS) is a system composed of multiple interacting intelligent agents [50]. The agents are designed as intelligent/autonomous entities such as software programs, robots, humans, human teams or their combination. The interactions of agents can be either cooperative or selfish. For a complex problem, MAS often decomposes it to multi-aspect problems that can be addressed by the combination of a number of simple, interacting agents. The global solution is seen as the emerged behaviors out of interactions among agents, and between agents and the environment.

In the research areas of EC and MAS, there are a large number of applications for solving the above complex problems. For instance, multiobjective evolutionary algorithms (MOEAs), as a variant of EC, have been well established as efficient methods to deal with MOPs [2, 4–16]. By imitating the natural process of biological evolution, the population-based MOEAs are able to find a set of non-dominated solutions in a single run. However, the performance of MOEAs heavily relies on the choice of evolutionary operators and control parameters.

Various trust models [18–24] in the filed of MAS have been proposed to cope with the robustness problem in MAS based e-marketplaces, particularly the unfair rating problem. For instance, dishonest advisors' ratings are either filtered out (*e.g.*, BRS [21] and iCLUB [18, 19]) or discounted (*e.g.*, TRAVOS [20], ReferralChain [22] and Personalized [23, 24]) before aggregating advisors' ratings to estimate seller reputation. Nonetheless, these models are not completely robust against various strategic attacks [25, 26].

Table 1.1: The similar characteristics of EC and MAS.

Evolutionary Computation (EC)	Multiagent Systems (MAS)
Candidate solutions	Agents
Evolutionary operators	Actions or behaviors
Gene exchange	A kind of communications between agents
Fitness function	Utility function
Variable/objective spaces	Environments

Agent-based techniques [37, 43, 44, 51–70] in the field of intelligent transportation systems (ITS) have been proposed for rerouting vehicles with less travelling cost and controlling traffic lights to maximize road capacities. However, they either consider only one of the two components (*i.e.*, vehicle rerouting and traffic light control) or validate their approaches on the small scale and simulated traffic scenarios. There is still a gap to apply them [37, 43, 44, 51–70] into real transportation systems.

Recently, the combination of various techniques in different subareas of AI has gained attraction and becomes a hot topic. *Multiagent evolutionary computation* (MAEC) is a new paradigm to efficiently solve a range of complex problems, by combing the advantages of *evolutionary computation* (EC) and *multiagent systems* (MAS) [71]. To explore possible combinations of EC and MAS, the similar characteristics of these two components are discussed below.

From the definitions of EC and MAS, both of them have the similar characteristics summarized in Table 1.1. For instance, a candidate solution or individual can be represented as an agent, evolutionary operators are similar to agents' actions or behaviors, gene exchange among individuals can be considered as a kind of communications or interactions between agents, fitness functions are like utility functions for driving agents towards final goals, variable or objective spaces are living environments for agents, etc. Such common characteristics make the combination of EC and MAS to be feasible and coherent, which becomes a new trend.

Multiagent evolutionary computation (MAEC) is a hybrid scheme to combine the two related domain knowledge together. The studies of combing EC and MAS have potential to contribute to both the fields. In general, there are three research directions in MAEC.

- "**Agent based EC**" [72–87]. In this category, a candidate solution is represented as an intelligent agent with the ability of autonomy, self-learning, interactions, etc.

- "**Evolutionary computation based MAS**" [88–97]. In order to improve agents' learning abilities and enhance the efficiency of interactions, evolutionary techniques are considered as complementary approaches to be used in MAS.

- "**The sequential or embedded approach of EC and MAS**" [32–34, 70, 98–100]. In this group, EC and MAS are employed either in either a sequential order or an embedded fashion.

As a new paradigm, MAEC becomes a hot topic to solve a range of complex problems, such as global numeric optimization [72, 78, 84–87], reactive power dispatch [73], rule-based knowledge extraction [74], constraint satisfaction [75, 79], combinatorial optimization [76], dynamic optimization [77], image feature extraction [80, 81], 7000-queen problems [82], server job scheduling [88], robot soccer in Keepaway [89, 90], iterated prisoner's dilemma [91, 92], trust management in wireless senor networks (WSNs) [95], mobile ad hoc networks (MANETs) [96] and vehicular ad hoc networks (VANETs) [97], fashion design problems [100], traffic congestion forecasting [32–34, 70], etc.

The studies in MAEC have been demonstrated as effective approaches for dealing with benchmark and real world problems [32–34, 70, 72–100]. They are considered as the first attempt to combine EC and MAS. However, the ability of MAEC for solving complex problems has not been fully developed and well exploited. The insufficiency of the existing MAEC approaches is briefly discussed below.

In the first category "agent based EC", most approaches only utilize part of characteristics of intelligent agents (*e.g.*, local views of agents) [72–83]. In addition, many methods design special actions for the particular problems, which limit their generality on other problems [78–83]. On the other hand, a plethora of proposals adopt the fully-developed techniques in MAS (*e.g.*, reinforcement learning, Q–learning, multi-armed bandit) to enhance the performance of EC [84–87]. But these studies put emphasis on either operator adaption or parameter adaption. Few studies have been carried out to select operators and parameters simultaneously in the research area of "agent-based EC". Furthermore, most of the previous studies focus only on single-objective optimization problems, whereas they do not consider complex problems such as multiobjective optimization problems (MOPs).

In the second group "evolutionary computation based MAS", a number of algorithms enhance a single agent's leaning abilities by adopting "evolutionary function approximator" [88–90], co-evolutionary algorithms [91, 92, 97], etc. On the other hand, some studies improve agents' communications by evolutionary algorithms, such as particle swam optimization (PSO) [93], genetic algorithms (GAs) [94], etc. In general, these studies use the traditional EC approaches to design intelligent agents. Some advanced EC approaches (*e.g.*, memetic algorithm [1, 49], multiobjective evolutionary algorithms [2, 4–16]) are not utilized in MAS. Thus, the capability of EC is not fully utilized in these studies. Furthermore, most of these studies assume that agents are benevolent and willing to share their experience and knowledge with other agents. But under some special circumstances (*e.g.*, e-marketplaces), the communication probably involves untruthful information. The trust issues thus need to be carefully considered in the implementation of interactions between agents in these multiagent-based scenarios.

In the third direction "the sequential or embedded approach of EC and MAS", most studies [98, 99] employ EC and MAS in a sequential order. However, they can be considered as loose coupling methods, which only iteratively utilize the final results from the black box of EC or MAS. On the other hand, some studies embed partial knowledge of EC, such as evolutionary operators [100] and bio-inspired concepts [32–34, 70], into MAS for improving the capability of problem solving. Thus, these studies [32–34, 70, 98–100] do not integrate EC and MAS effectively and delicately.

1.2 Proposed Approaches

In view of the limitations of the existing MAEC approaches, this thesis proposes three approaches, each of which falls into one of the three categories of MAEC. To demonstrate the effectiveness and efficiency, they are respectively applied to cope with three types of complex problems. The characteristics and challenges of these three complex problems are summarized as follows:

- **Multiobjective optimization problems (MOPs)** [2–6, 9, 11, 101–104]. MOPs often have several conflicting objectives, and it is difficult to simultaneously optimize all the objectives. For Pareto optimal solutions, the improvement on one objective leads to the decrement of at least one other objective. The core challenge of MOPs is to attain evenly distributed Pareto optimal solutions along Pareto fronts.

- **Robustness of trust in e-marketplaces** [17–20, 26–30, 105–109]. There exist dishonest information and various types of strategic attacks in the large, open and dynamic e-marketplaces. To gain its own maximum interest, an agent may be selfish, malicious or dynamically change its behaviors. It is thus challenging to design a robust trust model against various attacks.

- **Traffic management problem** [37, 43, 44, 51–70]. In a transportation system, a larger number of cars can enter or leave roads in large scale networks by following their own routes. On the other hand, traffic lights are used to regulate the move or stop of cars at road intersections. It is nearly impossible to find optimal routes for all cars and control traffic lights in a large traffic map only relying on a central solver. Thus, it is a challenging problem to design an effective, distributed approach for rerouting vehicles and controlling traffic lights.

The three proposed algorithms for respectively dealing with the above three complex problems are briefly discussed below.

Firstly, a multiagent evolutionary framework based on trust is proposed in Chapter 3 for coping with MOPs [3]. The intuition of this work is that different evolutionary operators configured with various control parameters are suitable to deal with different types of problems. Thus, adaptively selecting evolutionary operators and control parameters has great potential to improve the performance of evolutionary algorithms. The goal is to overcome the limitation of the first category of MAEC, where agents lack of the ability of simultaneously selecting evolutionary operators and control parameters. In general, previous studies [84–87] select evolutionary operators and control parameters based on human experience for particular problems. For other new problems, operators and parameters have to be identified by the trial-and-error procedure. Such mechanism is time consuming and inefficient when the problem is cost sensitive and highly dynamic. In this study, the trust concept, which is popular in MAS, is adopted to estimate the dynamic competence of evolutionary operators and control parameters from generation to generation and on different problems. Every agent (*i.e.*, candidate solution) automatically selects operators and parameters based on the probabilities correlated to their competence. To demonstrate the effectiveness of the proposed framework, MOPs are selected as testing instances. Experimental results on 58 benchmark MOPs demonstrate that the proposed framework significantly improves the performance of the state-of-the-art multiobjective evolutionary algorithms (MOEAs).

Secondly, to address robustness problem in e-marketplaces, a multiagent evolutionary trust model (MET) is designed in Chapter 4. The intuition behind this work is to utilize the special characteristic of robustness in evolutionary algorithms, which enables individuals (agents) to survive in a variety of surrounding environments. The purpose is to resolve the insufficiency of the second MAEC group, where communications between agents are assumed to be honest and benevolent. Taking the e-marketplace as an illustrative example, to gain their maximum interests, different agents (*e.g.*, buyers, advisors and sellers) adopt different strategies with cooperative, competitive and collusive behaviors. For instance, three colluding men positively rated each other several times and later sold a fake painting for a very high price [110]. To take transactions with a potential seller, a buyer faces great risks when it has little or no experience about the seller and has to rely on advisors' suggestions. In the survey of [109], there exist various strategic attacks from advisors (*e.g.*, Constant attack, Camouflage attack, Whitewashing attack, Sybil attack, Sybil Camouflage attack, Sybil Whitewashing attack) in e-marketplaces. As demonstrated in [26], none of the classical trust models are completely robust against all the attacks under rigorous tests. In the proposed MET, every buyer builds up its own trust network to model the trustworthiness of advisors. Based on the Darwin principle of "survival of fittest", each buyer evolves its trust network to capture dynamic behavior patterns of advisors and sets approximate trustworthiness to advisors by distributed evolutionary techniques. Experimental results on a multiagent-based e-marketplace testbed show that MET is more robust than the existing trust models (*i.e.*, BRS [21, 106], iCLUB [18, 19], TRAVOS [20], Personalized approach [107, 110] and ReferralChain [22, 111, 112]) in resisting unfair rating attacks.

Finally, a multiagent pheromone-based traffic management framework is proposed in Chapter 5 for reducing traffic congestion. The intuition of this work is to implement indirect communications between vehicles by mimicking nature species' behaviors. For instance, ants use trail pheromone (intensity of flavors) to guide other ants on attracting routes for food foraging. By modeling vehicles as ants, a vehicle collects digital pheromone information dropped by other vehicles to avoid traffic congestion. The aim is to remedy the inadequacy of the third direction of MAEC, where EC and MAS have not been effectively integrated. The previous studies [32–34, 70] only use partial information of EC (*i.e.*, bio-inspired concepts) into multiagent-based transportation systems. In addition, these studies only consider vehicle rerouting while ignoring another importance aspect, *i.e.*, traffic light control. On the other hand, the trail-and-error procedure in evolutionary algorithms [113] is only designed for simulation, since the cars' movement in realistic transportation systems is irrevocable. In the proposed framework, a novel "pheromone" is defined for bridging vehicle rerouting and traffic light control. In particular, the road infrastructure agents fuse pheromone to forecast traffic conditions. Then, a proactive vehicle rerouting algorithm is designed for assigning alternative routes to cars before they enter congested roads. At the same time, traffic light control agents use an online strategy for assigning color phases and time duration of these phases based on the amount of pheromone. Experimental studies on the platform of simulation of urban mobility (SUMO) [114] show that the proposed framework significantly alleviates traffic congestion and improves driver experience on two testing road networks.

1.3 Contributions of the Thesis

In conclusion, the proposed approaches in MAEC have three major contributions:

(i) The multiagent evolutionary framework based on trust enables agents to automatically select evolutionary operators and control parameters, and significantly improves the performance of the state-of-the-art MOEAs.

(ii) The multiagent evolutionary trust model (MET) assists buyers to construct robust trust networks. In addition, MET is robust than the existing trust models in resisting typically unfair rating attacks in e-marketplaces.

(iii) The pheromone-based traffic management framework simultaneously optimizes vehicle rerouting and traffic light control, which significantly alleviates traffic congestion, saves travel time, and reduces air pollution and fuel consumption.

To summarize, with the three kinds of effective MAEC algorithms for solving complex problems, this thesis serves to elicit more research efforts for building powerful MAEC approaches by delicately fusing the high level knowledge and advantages of EC and MAS, and applying them to solve large, open, dynamic, distributed complex problems. This work will also nourish the two related fields towards more intelligent EC and effective MAS.

1.4 Organization of the Thesis

The organization of this thesis is visualized in Fig. 1.2. In particular, Chapters 1–2 introduce the research topic of multiagent evolutionary computation (MAEC) and discuss the related two domains: evolutionary computation (EC) and multiagent systems (MAS). Then, three new approaches, each of which falls into one of the three categories of MAEC, are proposed for solving MOPs (Chapter 3), addressing robustness of trust in e-marketplaces (Chapter 4), and reducing traffic congestion in transportation systems (Chapter 5). At last, the conclusions and future work are given in Chapter 6.

The rest of chapters in this thesis are summarized as follow:

Chapter 2. Literature Review: This chapter firstly provides an overview of related research studies on evolutionary computation (EC) and multiagent systems (MAS). Then, the studies in MAEC are divided into three categories, and the limitations of the existing MAEC approaches are clearly pointed out, which motivate the research in this thesis.

Chapter 3. A Multiagent Evolutionary Framework based on Trust for MOPs: In this chapter, a multiagent evolutionary framework based on trust is proposed to cope with MOPs. At first, the existing adaptive mechanisms in EC are surveyed. Secondly, the background of MOEAs is briefly discussed. Then, the novel framework is proposed, where when designing a learning agent, the trust concept is

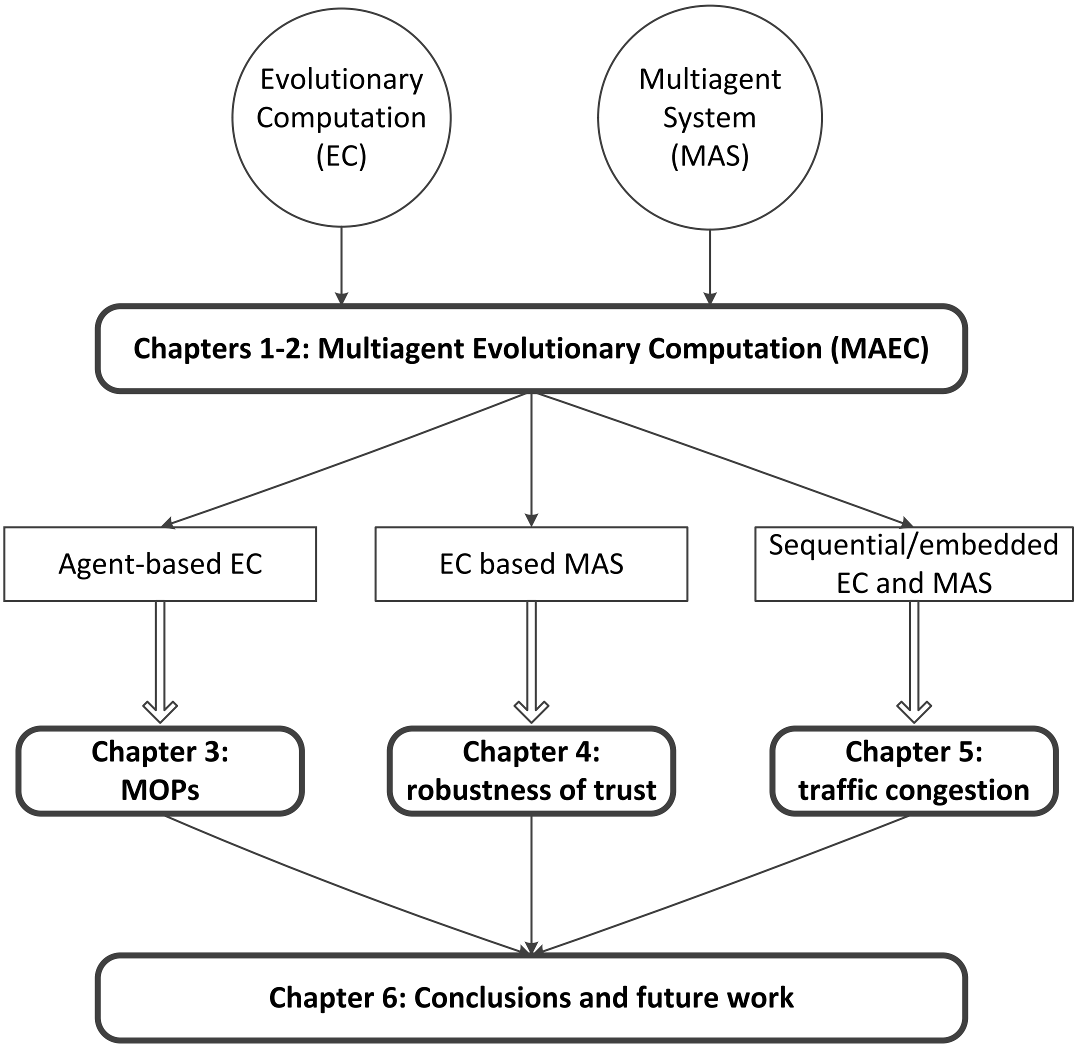


Figure 1.2: The orgainization of the thesis.

adopted to estimate the dynamic competence of evolutionary operators and control parameters from generation to generation and on different problems. Experimental results on 58 MOPs demonstrate that the proposed framework significantly improves the performance of the state-of-the-art MOEAs.

Chapter 4. A Multiagent Evolutionary Model for Constructing Robust Trust Networks: The goal of this chapter is to address the robustness problem in e-marketplaces when part of advisor agents are selfish or malicious. At first, the existing attack models and trust models are discussed in detail. Then, a novel multiagent evolutionary trust model (MET), inspired by EC, is proposed to assist buyers to model the trustworthiness of advisors. Experimental results on a multiagent-based e-marketplace testbed confirm that MET is more robust than BRS, iCLUB, TRAVOS, Personalized approach and ReferralChain in resisting the investigated attacks. At last, the conclusions and future work are given.

Chapter 5. A Multiagent Pheromone-based Traffic Management Framework for Reducing Traffic Congestion: This chapter aims to reduce traffic congestion in transportation systems. At first, the existing approaches in the field of intelligent transportation systems (ITS) have been classified into vehicle rerouting, traffic light control, and the combination of the above two components. Then, a pheromone-based traffic management framework is proposed for simultaneously searching optimal routes and controlling traffic lights. Experimental results on SUMO confirm its superiority than the existing algorithms (*i.e.*, TLC-PSO, Rerouting-DUA, Combination–PSO) for distributing traffic volume and maximizing road capacities. Finally, this chapter provides discussions on potential future work in this direction.

Chapter 6. Conclusions and Future Work: Based on the current studies, three possible directions for future work are identified. 1) For the first work: extend the current framework to a distributed evolutionary framework; incorporate more evolutionary operators and local search operators into the framework; and apply it to more challenging problems. 2) For the second study: develop new trust models to cope with multi-dimensional multi-nominal unfair ratings; detect the cheating behaviors from sellers; and put it into practice to solve complex problems with dishonest information. 3) For the third method: verify the current work on more complex road networks; develop an incentive mechanism for attracting cars to report their real driving information; and extend it to vehicle ad hoc network (VANET) scenarios.

Chapter 2

Literature Review

This chapter surveys related work in relevant research areas. Specifically, the surveys on evolutionary computation (EC) and multiagent systems (MAS) are given in Section 2.1 and Section 2.2, respectively. Then, Section 2.3 introduces the research topic *multiagent evolutionary computation* (MAEC), which integrates the characteristics of EC and MAS. Three categories of MAEC are identified: "agent based EC" in Section 2.3.1, "evolutionary computation based MAS" in Section 2.3.2 and "the sequential or embedded approach of EC and MAS" in Section 2.3.3. The limitations of the existing MAEC approaches are pointed out in Section 2.4. At last, the summary of this chapter is given in Section 2.5.

2.1 Evolutionary Computation

Evolutionary Computation (EC) is a kind of optimization methodology inspired by the mechanism of biological evolution and the behavior of living organism [1, 46]. Over the last three decades, EC has shown tremendous success in solving complex problems in the fields of biocomputing, cellular programming, evolvable hardware, game playing, job-shop scheduling, management sciences, non-linear filtering and timetabling, etc [115].

EC techniques mostly involve metaheuristic optimization algorithms. In general, it covers genetic algorithm (GA), genetic programming (GP), evolutionary programming (EP), evolution strategy (ES), genetic expression programming (GEP), memetic algorithm (MA), differential evolution (DE), swarm intelligence (SI), cultural algorithms (CA), learning classifier systems (LCS), estimation of distribution algorithm (EDA), artificial immune system (AIS), etc [49, 116].

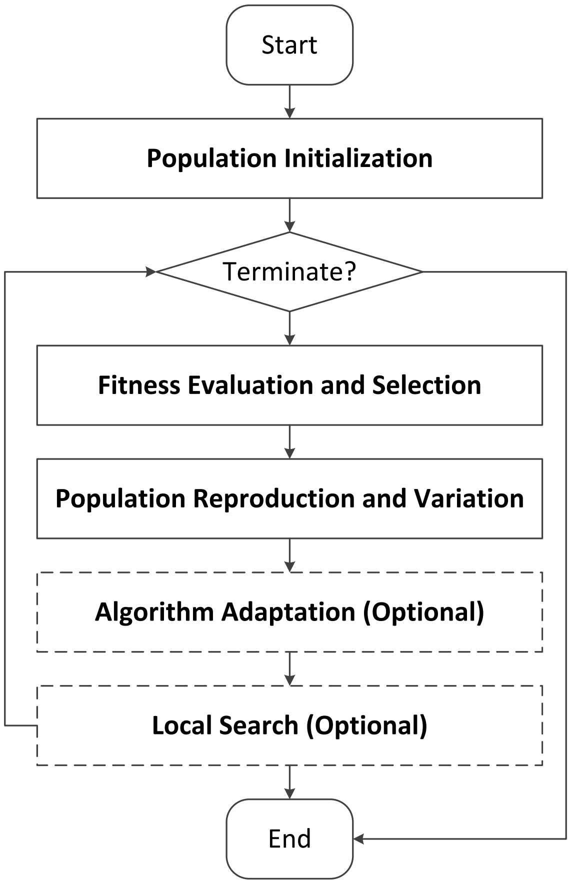


Figure 2.1: The general framework of Evolutionary Computation [1].

The framework of EC is well-defined and well-accepted by the research community. As shown in Fig. 2.1, the general framework of EC is composed of five basic components: population initialization, fitness evaluation and selection, population reproduction and variation, algorithm adaption, and local search [1]. The last two items are optional according to the types of EC. Based on this framework, the existing approaches for enhancing the effectiveness of EC can be generally categorized into the following five groups.

- Population initialization. In EC, the population consists of a number of candidate solutions as well as individuals. For exploring potential searching spaces, the frequently used encoding methods for individuals include the simple binary, real, string, ordered information, etc. Other advanced encoding approaches are trees, graphs, neural structures, Bayesian models, hidden Markov models, state-transition system models, etc [49, 104, 117, 118]. In general, the initial population is randomly generated. Other techniques, such as orthogonal and uniform initialization in the field of design of experiment (DOE) [47, 103], are adopted in EC for evenly distributing individuals into the whole searching space.

- Fitness evaluation and selection. To evaluate the quality of individuals, *fitness function* is designed to guide the direction of the evolutionary process. It decides which individuals can survive to the next generation under environmental selection pressure. In most cases, there is only one fitness function in an optimization problem. However, multiple fitness functions are also possible in some special cases. In addition, some fitness functions may be mutually conflicting and/or restricted by a number of special conditions. This type of problems is generally categorized into the multiobjective optimization problems (MOPs). On the other hand, many real world problems are hard to be formulated as mathematical models or represented by fitness functions. For instance, the semi-structure problems are not easy to be expressed in an analytical form of decision variables or the analytical formula is too costly to evaluate. Therefore, surrogate-assisted models, such as polynomial regression, support vector machine (SVM), etc., have been adopted to approximate fitness functions or reduce computational burden [48, 119–122].

- Population reproduction and variation. The process of population reproduction and variation (*evolutionary operators*) is used to produce the next generation of individuals. A plethora of evolutionary operators have been proposed to obtain high quality of new candidate solutions, such as simulated binary crossover (SBX), polynomial mutation [9, 123], strategies in differential evolution [124], methods in particle swarm optimization (PSO) [125], orthogonal crossover [126, 127], etc. To enhance evolutionary operators' searching abilities, a number of techniques in machine learning, *e.g.*, opposition-based learning (OBL) [128], clustering, principal component analysis (PCA) [129, 130], SVM [131], etc., have been adopted as auxiliary methods to produce high quality solutions.

- Algorithm adaption. It mainly includes operator adaption and parameter adaption. In particular, operator adaption refers to the selection of one or multiple evolutionary operators from a set of candidate operators as algorithm processing [132]. On the other hand, parameter adaption utilizes the collected historical feedback to control EC's parameters and strengthen its searching ability [133–135]. For instance, the accumulated fitness reward calculated by the classical methods (*e.g.*, reinforcement leaning) is used to select crossover operators in GAs [136] and choose mutation operators in EP [86].

- Local search. The local search (LS) aims to refine a solution within its neighborhood searching space, which focuses on exploitation. On the other hand, due to the random mechanism from the procedure of population initialization and reproduction, traditional EC has a strong ability of exploration. By combining the advantages of LS and traditional EC, memetic algorithm (MA) is an effective mechanism for problem solving. The classical MA is considered as a synergy of population-based evolutionary approaches with separate individual learning or local improvement procedures for problem search [48, 49, 137–140]. In order to effectively utilize LS operators, various techniques, *e.g.*, automatically select individuals to perform LS, adaptively control LS intensity or search depth [138], adaptively select one or many LS operators from a LS pool [137], self generate new LS operators [49, 141], have been proposed to improve the searching abilities of MA.

From the analysis of EC, it can be found that most of them are centralized control approaches, which limits their ability to deal with large scale and real world problems. Additionally, the communication between individuals is implemented by the simple gene/chromosome exchange. The high level knowledge (*e.g.*, the competence of evolutionary operators, the feasible searching space) is not exchanged among individuals. Furthermore, the individuals have limited capability to self learning, extract knowledge and react to changes in environments. These limitations make the classical EC not practical when they are applied to cope with large scale, mutually conflicting, expensive and dynamic problems.

2.2 Multiagent Systems

A *multiagent system* (MAS) is a system composed of multiple interacting intelligent agents [50]. Multiagent systems have a large number of applications covering the fields such as electronic marketplaces, computer games, coordinated defense systems, scheduling problems, transportation systems, logistics, disaster response problems, modeling social structures, etc [50, 71, 81, 107, 110, 142–144].

In multiagent systems, an *agent* or *intelligent agent* is an autonomous entity which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals [144]. The agents have several common characteristics, summarized as follows.

- Autonomy. An agent is at least partially autonomous, which controls its own actions without intervention of human or others.
- Local views. No agent has a full global view of the system, or the system is too complex for an agent to make practical use of such knowledge.
- Decentralization. There is no designated controlling agent (or the system is effectively reduced to a monolithic system).
- Learning or adaptive. An agent is able to change its behavior based on its experience and sensing information from the environment.
- Social ability or communicative. An agent communicates with other agents, humans or their combination using agent-communication languages.
- Reactivity. An agent responds in a timely fashion to changes in the environment.
- Pro-activity. An agent does not simply act in response to its environment. It is able to exhibit goal-oriented behaviors by taking the initiative.
- Mobile. An agent is able to transport itself from one environment to another.
- Flexible. An agent's actions or behaviors are not hard coded.
- Character. An agent has its own believable "personality" and emotion states.

Depending on various applications, researchers design agents by choosing part of the above characteristics. On the other hand, the agents' characteristics are not restricted to the above list. In future, other additional properties may increase agents' abilities to effectively solve problems.

Based on the degree of intelligence and capability, agents are broadly divided into five groups [144]:

- Simple reflex agent. This agent group only remains the current perception and ignores the historical information. The action function is described as condition-action rules (*e.g.*, if certain conditions are satisfied, then a specifical action will be performed). This type of agents only succeeds when they can fully observe the environment.

- Model-based reflex agent. A model-based agent can handle a partially observable environment. It designs abstract structures to explicitly represent, and symbolic model the environment. In essence, this type of agents chooses actions in the same way as the reflex agent.

- Goal-based agent. This kind of agents is model-based and stores the "goal" information describing the desirable situation. A good example is the agent with the Belief Desire Intentions (BDI) model which has capability to plan action sequences to achieve its goal or multiple possible goals [143, 144].

- Utility-based agent. This type of agents uses utility functions to guide their behaviors. An utility function maps a state to a measure of the utility of the state. Some techniques, such as dynamic programming, temporal difference learning, Q–learning, SARSA, etc., have been proposed to help rational agents to choose actions for maximizing expected utility [142].

- Learning agent. Learning agents initially operate in unknown environments and become more competent than what its initial knowledge alone might allow. This type of agents (*e.g.*, FALCON [145]) utilizes immediate/delayed rewards from the environment to modify its action function for obtaining high utility values in future.

Based on the real world applications, agents are also called as decision agents, input agents, processing agents, spatial agents, world agents, believable agents, physical agents, temporal agents, etc [143, 144].

A number of agents or intelligent agents interact with each other and form a multiagent system (MAS). In general, MAS considers the following factors:

- The type of agent-agent interactions (cooperative, competitive or some combinations).
- The type of agents involved (homogenous or heterogeneous).
- The number of agents involved (from a small to a large number of agents).
- The frequency, duration, importance, and variability of agent-agent interactions.

An interesting aspect of MAS is that the entire system can manifest self-organization even when the individual strategies of all their agents are simple. Interaction plays a vital role to organize simple agents together in the whole picture. In comparison to EC, MAS does not have a well-defined framework. However, it cannot be seen as a negative point for MAS because it gives the flexibility in designing MAS for a wide range of problems [50].

By analyzing the characteristics of intelligent agents, MAS is suitable to deal with large scale, open, dynamic, and uncertain problems. The studies in MAS often adopt decentralized control mechanisms, enabling MAS to have certain capabilities of fault tolerance when some of the agents are failed or shut down. However, if agents only have local views, it cannot guarantee them to find global optimal solutions when they lack of global information about the entire environment. On the other hand, from the view of the whole system, distributed agents in local environments are connected together by interactions or communications. When interactions are expensive, communications are ineffective or agents cannot reach an agreement, some complementary methods are needed to assist MAS to tackle these problems.

2.3 Multiagent Evolutionary Computation (MAEC)

In recent years, *multiagent evolutionary computation* (MAEC) becomes a new paradigm to efficiently solve a range of complex problems. This approach is a hybrid scheme which combines the advantages of *evolutionary computation* (EC) and *multiagent systems* (MAS) [71]. In general, there are three possible ways to combine EC and MAS together: 1) "agent based evolutionary computation", 2) "evolutionary computation based MAS", 3) "the sequential or embedded approach of EC and MAS".

2.3.1 Agent based Evolutionary Computation

In the first category of MAEC, "agent based EC" incorporates some characteristics of intelligent agents and adopts techniques in MAS to enhance the performance of EC.

(1) "Local views" of agents are incorporated into evolutionary computation.

In "agent based EC", an individual can be defined as an agent. Based on this assumption, every agent has its own goals, predefined actions and self-learning abilities. At the same time, all agents have *local views* and live in, for example, the lattice-like environment [72–76], two dimensional grid environment [77], ring-like environment [146], etc. In these environments, every agent is fixed on a relatively stable position and only has local views to adjacent regions. For instance, an agent in two dimensional grid only interacts with its four neighbor agents, which have the nearest distance to the agent and are located in the east, south, west and north directions, respectively.

The "local views" make agents to compete and/or cooperative with their local agents. Obviously, there is no global selection in these studies of "agent-based EC". The natural selection of the fittest agents only occurs in local environments. The global communication can be achieved by a process of information diffusion. In essence, the lattice-based population or the local view to environments is a kind of techniques for enabling a fine-grained parallel implementation of GAs [72].

In [72–77], there are four types of predefined actions for agents: the neighborhood competition operator, the mutation operator, the advanced evolutionary operator (*e.g.*, orthogonal crossover operator [126]) and the self-learning operator (*e.g.*, local search [72, 73], learning table [76], statistics based learning behavior [77]). Basically, these actions are traditionally evolutionary operators with specified parameter settings, which are proposed and verified as effective operators to solve particular problems. The authors select these traditional evolutionary operators and predefine control parameters according to their own experience. For instance, the action of "orthogonal crossover operator" is controlled by the predefined crossover probability, which is set as a fix value (*e.g.*, $p_c = 0.1$ [72]) before the algorithm starts. Thus, these agents can be classified into the simple reflex agents [72–76] or the model-based reflex agents [77].

In a word, these studies [72–77] utilize the characteristic of *local views* in intelligent agents to enhance EC's search abilities. In these studies, the agents' actions are set as the traditional evolutionary operators with predefined control parameters. Experimental results demonstrate that these approaches are effective methods for dealing with global numeric optimization problems [72], reactive power dispatch [73], rule-based knowledge extraction [74], constraint satisfaction problems [75], combinatorial optimization problems [76], dynamic optimization problem [77], etc.

(2) Specific "actions" are designed for evolutionary computation.

Besides the traditional evolutionary operators, some specific actions or evolutionary operators have been proposed to solve various types of complex problems.

In [82], Liu and Jing design three local reactive actions (*i.e.*, *random-move*, *better-move* and *least-move*) to solve constraint satisfaction problems (CSPs). In particular, the agents are distributed in a two dimensional grid-like environment. Different positions of agents in the grid represent different candidate solutions for CSP problems. At the same time, there are some constraints for restricting the location of agents. For instance, n–queen problems require that two queens cannot be located in the same rows, columns, and diagonal lines. In addition, the three kinds of agents' actions in [82] are controlled by fixed probability values. For instance, the *random-move* switch an agent to a randomly position with a probability of p_{random}. An agent uses the action, *better-move*, to arrive at a new position that has the smaller violation number than its current position. This action is controlled by the parameter p_{better}. The action *least-move* needs more computational budget than the previous two actions to find a position with the least violations. The parameter p_{least} is used to control the action *least-move*. Based on a number of experimental studies, the three parameters (*i.e.*, p_{random}, p_{better} and p_{least}) are carefully configured for different CSP problems, which keep the same values along the whole searching process.

In [78, 79], the agent-based memetic algorithm (AMA) defines four types of life span learning process (LSLP) operators as actions for agents. In particular, the first LSLP operator is a totally random method in nature. The second one is a restricted random operator that only operates on partial variables. The third one is gradient-based optimizer, and the last one is a directed search. For the first two operators, the random mechanism is used to maintain the diversity of agent-based population. For the last two operators, the directed searches try to move towards a better solution which is not necessarily in the agent's local region. As mentioned in the beginning of Section 2.3, an agent is a candidate solution consisting of a number of decision variables. In the process of LSLP operators, every variable is added or reduced with a value Δ. At the earlier stage of AMA, the value of Δ is large to encourage the exploration for searching on the whole variable space. On the other hand, Δ is set as small values for exploiting at the later stage of AMA, since agents are close to the optimum. Thus, the value of Δ should gradually decrease as the generation number increases. In particular, the value of Δ in AMA is set as $\Delta = \text{Normal}(0, 1/g)$, which is generated by the normal distribution and g is the generation number.

In [83], Socha *et al.* design two specific actions (rules) to transfer energy from one agent to other agents. In the beginning of the algorithm, every agent has an initial amount of energy that is the same as each other. Based on the competition between agents, energy will shift among the agent-based population. In particular, if an agent is dominated by its neighbors, its energy will be transferred to its local agents. The proportion of energy transformation is determined by a condition rule (*i.e.*, the domination energy transfer principle) predefined by the designers themselves. On the other hand, if a local region has too many agents, another predefined rule (*i.e.*, the crowd energy transfer principle) is designed for transferring energy of agents in crowded areas to agents in sparse regions.

To enhance the capability of agents, the studies in [78, 79, 82, 83] focus on designing specific actions for particular problems. In these studies, action selection is based on probability values [82] or predefined condition-action rules [83]. Experimental results demonstrate that these algorithms successfully solve 7000-queen problems, coloring problems [82], real-valued optimization problems [78], constrained optimization problems [79] and bi-objective multiobjective optimization problems [83].

(3) Some techniques of MAS are adopted in evolutionary computation.

The above mentioned studies focus on incorporating some characteristics of intelligent agents into EC. Other approaches in "agent based EC" utilize the fully-developed techniques in MAS to improve the effectiveness of traditional EC.

In [84], Eiben *et al.* use reinforcement learning (RL) [40] to online calibration of parameter settings in genetic algorithms (GAs). In RL, there exist three basic components (*i.e.*, states, actions and rewards), which are defined as continuous variables in this study. In particular, the *state* of RL is represented by a vector of variables that reflect the main properties of the current population (*i.e.*, a mean value of individuals' fitness, a standard deviation of individuals' fitness, etc). The *action* of RL is the parameter settings in GAs (*i.e.*, population size, the tournament proportion, mutation probabilities and crossover probabilities). The *reward* is determined by the improvement of the best fitness value found by GAs with different parameter settings. After the three components of RL are defined, Q–values are used to adjust the parameter settings in GAs. Because the states and actions are continuous variables, there is an infinite number of combinations of the pairs of states and actions. In consideration of this challenge, the regression tree is adopted to represent the *function approximator* for calculating the Q–values in RL. On the other hand, another RL–based approach, the classical SARSA algorithm (State-Action-Reward-State-Action) [40], is also adopted to adjust the control parameters (*i.e.*, step sizes of (1+1)–ES) in evolutionary strategies [85]. Experimental results on the simulation problems indicate that evolutionary algorithms (*e.g.*, GAs, ES) calibrated by RL outperforms the canonic evolutionary algorithms [84, 85].

In the adaptive evolutionary programming (EP) [86], agents use Q–learning to select operators from a set of mutation operators. In particular, there are four typical mutation operators including Gaussian mutation, Cauchy mutation, Lévy mutation and Single-point mutation. The operators selection is mapped into a reinforcement learning problem (RL). To simplify the RL optimization problem, this study [86] ignores the state description in the RL formulation. Every individual learns its optimal mutation operators using the Q–learning method based on the immediate or delayed reward (*e.g.*, fitness improvement) of mutation operators. Experimental results on numeric optimization benchmark problems demonstrate that EP with mutation operators selection is better than EP using the best of four basic mutation operators. This indicates that EP needs different kinds of mutation operators in different stages of the searching procedure.

In [87], Maturana *et al.* adopt multi-armed bandit (MAB) for selecting evolutionary operators on different problems. In particular, each operator is viewed as one arm of a MAB problem. The rewards are based on the fitness improvement brought by a corresponding operator on an individual. Experimental results on simulation problems show that MAB enable agents to select suitable evolutionary operators. At the same time, GAs with MAB is significantly better than traditional GAs to solve the global numeric optimization problems.

In [147], Lee *et al.* utilize the game theory and robust multidisciplinary design techniques to produce high quality Pareto optimal solutions in the aerospace engineering design. The Nash Game is used to decompose the complex design problems into several simple problems corresponding to the number of Nash-players. Every Nash-player optimizes part of decision variables while fixing the other decision variables. Experimental results show that the interactions between Nash-players accelerate the multiobjective optimization process by incorporating the distributed agents' local optimized results.

To summarize, "agent based EC" aims to enhance the performance of EC by utilizing the special characteristics in intelligent agents and fully-developed techniques in MAS. As more specific properties of MAS (*e.g.*, distributed skills, self-learning abilities, mobile computation, consideration of emotion states, etc.) are incorporated into EC, this category of MAEC has great potential for proposing more intelligent evolutionary algorithms that have strong capabilities to cope with complex problems.

2.3.2 Evolutionary Computation based MAS

In the second group "evolutionary computation based MAS", agents use evolutionary techniques for self-learning, improving communications and increasing the effectiveness of the whole system.

(1) "Evolutionary function approximator" is used to design self-learning agents.

In MAS, "evolutionary function approximator" is used to model the value function in temporal difference (TD) [142]. TD is a classical method for assisting agents to make decisions. In general, TD represents the value functions as tabular formulations. However, a real-world environment often has enormous pairs of states and actions. It is impossible to map all the potential combinations of states and actions to Q–values in a table. *Function approximator* is a model-based method for an agent to learn the value function. The studies in TD learning often use neural network (NN) [118] to represent the function approximator. However, it is difficult to decide the architectures or topologies of NN. Another challenge is how to determine a large number of the linking weight values between neuron nodes. In [88–90], Whiteson *et al.* adopt NeuroEvolution of Augmenting Topologies (NEAT) to automatically search the topologies and evolve the linking weights. The special characteristic of NEAT is that it adopts the classical evolutionary framework to enable NEAT to learn, via back propagation, to represent the value estimates provided by Q–learning. By testing the benchmark problems such as the mountain car task, the server job scheduling problem [88] and robot soccer in Keepaway [89, 90], experimental results show that "evolutionary function approximator" can significantly improve the performance of TD approaches.

In [91, 92], the co-evolutionary learning algorithm is adopted to deal with iterated prisoner's dilemma (IPD) in game theory. The IPD problem is more complex than the simple prisoner's dilemma problem. In the simple prisoner's dilemma game, an agent has only two choices (*i.e.*, either cooperation or defection). However, for the IPD games, every agent faces many choices including continuous levels of cooperation and defection. The crucial factor is to enable agents to select the best behavior in various game scenarios. In [91, 92], Yao *et al.* utilize a feed forward neural network to represent an agent's behavior model. The action space has a number of strategies including multiple levels of cooperation and defection. To evaluate the competence of strategies, the reputation concept [105] is introduced into this work. In particular, the reputation score of a strategy, which is calculated based on the payoffs received in the previous history, is used as an input parameter to an agent's neural network. An agent decides its current action depending not only on previous moves from direct interactions with the environment, but also from reputation scores built based on indirect interactions with other agents. Experimental results indicate that the co-evolutionary learning method with the reputation concept effectively solves the continuous IPD problem [91, 92].

(2) Evolutionary skills are adopted to improve the communication between agents.

The interactions or communications between agents play an important role in MAS, which aim to organize and coordinate simple agents in the whole systems.

In [93], particle swarm optimization (PSO) [125] is adopted for exchanging experience and knowledge, which speeds up agents' learning and improves the communications between agents. In this study, every robot agent is configured with a single-layer discrete-time artificial NN of two neurons, one for each wheel speed, with a sigmoidal output function. For the agents, the challenge is to adjust the weights of NN based on their own local information. Some studies [148] utilize evolutionary techniques (*e.g.*, GAs) to modify weights by implementing communication in group robots. In traditional GAs, a population manager (*e.g.*, a single agent as a centralized controller) must have knowledge of an entire population in order to implement the breeding techniques. For the local version of PSO formulation, each robot only needs to be aware of the state of a small subset of neighbor robots. In [93], this study applies PSO to reproduce new parameters of neural network for a robot agent by allowing it to model the best agent and neighbor agents in a distributed manner. Experimental results on group agents' learning tasks show that PSO-based communication performs well for the non-uniform robot distribution.

In [94], Feng *et al.* propose a new memetic multiagent system (MeM) towards human-like social agents with memetic automaton. The system adopts TD-FALCON [145] as learning agents. Based on interactions with the environment, an agent accumulates experience and stores knowledge as cognitive nodes in the second field of TD-FALCON. In this study [94], the concept meme is used as building blocks of information or knowledge. Then, four operators (*i.e.*, meme expression, meme assimilation, meme internal evolution and meme external evolution) are designed to allow agents to share experience and knowledge. To measure the quality of agents' knowledge, fitness scores are defined to evaluate the efficiency of solving the minefield navigation task (MNT). Every agent can select other agents from the population as models (teachers) and imitates their actions. The agents with larger fitness scores will have more chances to be selected as teachers. Based on this mechanism, the knowledge from a single agent's own experience can be shared and transferred within its local regions. Experimental results on different scales of MNT problems show that the new system MeM is significantly better than the original TD-FALCON [94].

(3) Evolutionary techniques are used to increase the effectiveness of the whole system.

To improve the effectiveness of the whole system, some studies apply the techniques in EC to wireless senor networks (WSNs), mobile ad hoc networks (MANET) and vehicular ad hoc networks (VANETs) [95–97].

In [95], Gómez *et al.* use the ant colony system (ANS), one type of EC, to find the most reliable route for propagating messages in WSNs. In the circumstance of WSNs, agents exhibit complex behavior patterns including honest, cheating and collusive actions. For instance, a malicious agent could avoid reaching its benevolent neighbors, or lead always to other malicious agents, even collusively cheat honest agents to the designated agents. In this study, ANS is used to build trust values between agents. The ants (represent agents) leave some pheromone traces that help next ants to find and follow those routes. Experimental results demonstrate that the bio-inspired techniques increase the accuracy and robustness of message propagation in WSNs.

In [96], Mejia *et al.* utilize a distributed evolutionary algorithm to foster cooperative behaviors in MANET. In the applications of MANET, a selfish mobile agent is reluctant to forward packets from other agents because it wants to reserve its own battery power and energy. If agents want to use network with high speeds, each agent needs to contribute to packet forwarding for others. Thus, a cooperation enforcement mechanism is necessary for MANET environments. By modeling the packet forwarding in MANET as a cooperative game, a bacterial-like algorithm (one type of distributed EC) is adopted to allow agents to quickly learn appropriate cooperative behaviors. Experimental results demonstrate that the proposed algorithm adapts effectively to the environmental changes in MANET.

In [97], Abdou *et al.* adopt genetic algorithms (GAs) to optimize parameter settings in VANETs. To improve the effectiveness of VANETs, an effective communication algorithm needs to consider several aspects (*i.e.*, the neighboring density, the size and shape of network, the use of a channel, and the priority levels of messages). It is challenging to determine input parameters in a real-time fashion for different VANET scenarios. In [97], every agent adopts GA to automatically optimize input parameters according to information in its local environment. Experimental results on four types of context demonstrate that GA enhances the effectiveness of broadcast algorithms in VANETs.

To summarize, "evolutionary computation based MAS" aims to enhance agents' learning abilities and improve the effectiveness of MAS by adopting evolutionary techniques. As a bio-inspired mechanism, EC has certain advantages such as self-organization, self-adaption, robustness to changes in environments, etc. Thus, the research in this MAEC group will bring more benefits to MAS by incorporating more advantages of EC.

2.3.3 The Sequential or Embedded Approach of EC and MAS

In this MAEC group, EC and MAS can be employed in either a sequential order or an embedded fashion. However, only few approaches [32–34, 70, 98–100, 113] have been proposed and studied in this direction.

In [98], Li *et al.* firstly utilize the multiagnt consultations to initialize the allocation of tasks and coordination. Then, the tasks are rescheduled by the hybrid genetic algorithm (GA) in order to achieve global optimization. For the traveling salesman problem [99], at first, GA is implemented to find the near-optimal solutions on the sub-tour problems that only consider the neighborhood agents. Then, these solutions are used as an initial multi-vehicle plan through negotiation between agents. Finally, the multiagent plan is further optimized by an evolutionary algorithm for obtaining a good overall team strategy. In addition, Bazzan *et al.* [113] adopt evolutionary algorithms (EAs) to optimize two components (*i.e.*, vehicle rerouting and traffic light control) in a sequential order, which optimizes one component based on the optimal results obtained from the other component. In general, these studies [98, 99] decompose the large scale problems into a number of subproblems, and then employ the approaches of EC and MAS in a sequential order for solving these subproblems.

In [100], Liu *et al.* propose a multiagent system where EC is used as a continuous novelty generator, not as an optimizer. In particular, EC uses a binary algebraic expression tree to form sketch shapes, and utilizes a feature based product tree to produce component combination choices. The random mechanism in EC is able to increase the diversity of candidate solutions. Experimental results on mobile phone designs show that the new approach is able to generate creative solutions. In [32–34, 70], the bio-inspired concept, *i.e.*, pheromone, has been incorporated into traffic management models. By imitating the ants' communication where interactions are accomplished by exchanging the chemical favors (called pheromone), these studies [32–34, 70] design the "digital pheromone" for implementing the communications between cars, cars to roadside infrastructures, cars to the central server, etc. The results in [32–34, 70] indicate that the digital pheromone is benefit for increasing the accuracy of traffic congestion predictions. In a word, these studies embed the partial knowledge of EC, *i.e.*, the evolutionary operators [100] and the bio-inspired concept [32–34, 70], into MAS for improving the capability of problem solving.

To summarize, "the sequential or embedded approach of EC and MAS" employ the techniques of EC and MAS either in a sequential order or an embedded fashion. In future, by effectively merging the knowledge of EC and MAS, this direction of MAEC will incubate more powerful MAEC approaches with great flexibilities for solving distribute, large scale and dynamic complex problems.

2.4 Limitations of Existing MAEC Approaches

In this section, the limitations of the existing approaches in the three categories of MAEC are illustrated and analyzed, respectively.

In the first category "agent based EC", most of the proposed approaches only utilize part of the characteristics of intelligent agents and MAS. For instance, the studies in [72–83] define a solution as an agent with *local view* (one characteristic in the intelligent agents). The actions for agents are traditionally evolutionary operators [72–77] or specifically designed actions [78–83]. In these studies, an agent is a dummy entity, which lacks of the capability of self-learning. Agents make decisions based on human experience, condition-action rules (*e.g.*, the domination energy transfer principle and the crowed energy transfer principle [83]) and predefined probabilities (*e.g.*, the agent takes the mutation operator with predefined probability 0.1 [72]). On the other hand, other studies define a solution as a learning agent by adopting Q–learning methods [84, 86], SARSA [85], multi-armed bandit [87] and techniques in game theory [147]. These approaches are located in "algorithm adaption", which is the fourth element in the general framework of EC (As shown in Fig. 2.1). They focus on either operator adaption or parameter adaption. For instance, agents in [86, 87] select evolutionary operators from the pool of candidate operators (*e.g.*, Gaussian mutation, Cauchy mutation, Lévy mutation and single-point mutation [87]). In [84, 85], these studies allow agents to adjust the control parameters (*e.g.*, the step size of Gaussian mutation in ES [85]), and adaptively set the whole parameter settings in EC (*e.g.*, population size, tournament proportion, mutation probability and crossover probability [84]). In a word, these approaches try to represent solutions as intelligent agents, and make EC to be flexible to choose actions on different problems. However, none of them select operators and parameters simultaneously. Last but not least, although these studies are demonstrated as effective methods to solve single-objective optimization problems, the ability of MAEC for dealing with the multiobjective optimization problems (MOPs) has not been investigated.

In the second group "evolutionary computation based MAS", some studies in [88–92, 97] improve agent's leaning abilities by adopting the techniques in EC. For instance, the studies in [88–90] expand the tabular value function to "evolutionary function approximator" (*i.e.*, NeuroEvolution of Augmenting Topologies). Yao *et al.* [91, 92] use the co-evolutionary learning model to optimize connective structures (*e.g.,* the weights of neural network). Abdou *et al.* [97] adopt GA to optimize parameter settings in VANETs. Basically, these approaches put emphasis on each individual agent. They lack of elaborated designs on the communication between agents. In view of this, certain studies in [93, 94] improve agents' communication based on EC's concepts. For instance, agents use PSO [93] to exchange parameters of their neural networks. In [94], an agent is able to imitate other agents' behaviors based on its observation. In general, these approaches often use the traditional EC to design intelligent agents. The configurations of EC (*e.g.*, population size, maximum number of generation, evolutionary operators, control parameters) are predefined based on the designers' experience on particular problems. Some advanced EC approaches (*e..g*, local search operators in memetic algorithm) are also not adopted in MAEC. Thus, the capability of EC is not fully developed in these studies. In addition, most of them assume that agents are benevolent and willing to share their experience and knowledge with others. However, this assumption does not hold in some real world applications. For instance, agents in e-marketplaces may be dishonest or malicious, which will provide unfair information to other agents. An example is that three colluding men positively rated each other several times and later sold a fake painting with a very high price [110]. Under this type of situations, the trust concept needs to be considered in the communication between agents in MAS.

For the third direction “the sequential or embedded approach of EC and MAS”, most of them are loose coupling methods. For instance, the studies in [98, 99] decompose a large scale problem into subproblems which are solved by agents. In essence, MAS only utilizes the final results from the black box optimizer of EC. On the other hand, the studies [32–34, 70, 100] only bring the partial knowledge of EC (*e.g.,* evolutionary operators and pheromone concept) into MAS. For instance, the evolutionary operators in EC is used to stimulate the imagination of designers [100]. In [32–34, 70], the digital pheromone is instantiated to implement the communication between agents. Thus, the characteristics of EC and MAS in these studies [32–34, 70, 98–100] have not been integrated effectively.

2.5 Summary

This chapter firstly investigates the general framework of EC including five components: population initialization, fitness evaluation and selection, population reproduction and variation, algorithm adaptation and local search. Then, the important characteristics of intelligent agents are introduced. Based on the degree of intelligence and capability, agents are divided into five groups. After that, the possible combinations of EC and MAS (*i.e.,* multiagent evolutionary computation) are identified.

There are three potential categories for combining EC and MAS: 1) “agent based EC” incorporates some characteristics of intelligent agents, and adopts the techniques in MAS to enhance the performance of EC; 2) “evolutionary computation based MAS” utilizes EC to design self-learning agents, to improve the efficiency of interactions between agents, and to increase the effectiveness of the whole system; 3) “the sequential or embedded approach of EC and MAS” employs EC and MAS in either a sequential order or an embedded fashion.

Finally, the limitations of studies in each category of MAEC are clearly pointed out. In particular, most studies in the first category of MAEC focus only on single-objective optimization problems, whereas only few of them consider the complex problems, *e.g.*, multiobjective optimization problems (MOPs). In the second group of MAEC, the information communicated between agents is often assumed to be truthful, which does not hold in MAS based e-marketplaces. In addition, the two components of EC and MAS have not been integrated effectively and delicately in the third direction of MAEC.

Chapter 3

A Multiagent Evolutionary Framework based on Trust for MOPs

As mentioned in the previous chapters, the performance of *Evolutionary Computation* (EC) is highly associated with the five basic components including 1) population initialization, 2) fitness evaluation and selection, 3) population reproduction and variation, 4) algorithm adaptation, and 5) local search [1]. This chapter will investigate the fourth critical component of "algorithm adaption" involving two aspects: operator selection and parameter adaptation.

In EC, candidate solutions to the problems play the role of individuals in a population. They produce offsprings by taking *evolutionary operators* (such as crossover and mutation) with user-specific *control parameters*. EC is well known by its generality and simplicity that they often perform well approximating solutions to all types of problems in many fields such as engineering, economics, robotics, etc. However, evolutionary operators and control parameters may vary for different problems. It is time-consuming to determine the operators and parameters by the trial-and-error procedure. In addition, the competency of operators may vary with generations. For example, crossover is often powerful in the earlier stage of EC, but mutation is effective when the solutions are similar with each other in the later stage of EC. The challenge of EC is thus how to effectively select evolutionary operators and adjust control parameters from generation to generation and on different problems.

As discussed in Chapter 2, various agent-based EC approaches [72–87], in the first category of MAEC, have been proposed for enhancing the performance of EC. However, they either select operators or adjust parameters. On the other hand, most of them put efforts for solving the single-objective optimization problems, while ignoring complex problems such as multiobjective optimization problems (MOPs).

In this chapter, a novel multiagent evolutionary framework based on trust is proposed, where each solution is represented as an intelligent agent, and the pairs of evolutionary operators and control parameters are represented as services [3]. In the proposed framework, the agents model the trustworthiness of the services, based on whether the agents' offsprings produced by using the services survive to the next generations, which represents the dynamic competency or suitability of the services from generation to generation and on particular optimization problems. The agents will then select the services with the probabilities correlated to the trustworthiness of the services. To demonstrate the value of the proposed framework, the challenging problem, *i.e.*, multiobjective optimization problems (MOPs), is selected as a case study[1]. Experimental studies on 58 benchmark MOPs confirm that the proposed framework significantly improves the performance of the state-of-the-art multiobjective evolutionary algorithms (MOEAs). This work thus represents a promising step towards the use of multiagent based paradigms in the design of novel EC by involving intelligent agents (*i.e.*, solutions) that adopt trust modeling techniques for selecting evolutionary operators and control parameters in multiagent evolutionary computation (MAEC).

The rest of this chapter is organized as follows. Section 3.1 surveys the existing work on the adaptive multiobjective evolutionary algorithms (MOEAs) and compares the proposed framework with the other existing approaches for selecting evolutionary operators and adjusting control parameters. Section 3.2 provides the brief background of MOPs and MOEAs. The multiagent evolutionary framework is described in Section 3.3. Experimental results are reported in Section 3.4. Finally, Section 3.5 summarizes the current work.

[1] The proposed framework is also generally applicable to other optimization problems.

3.1 Related Work

In the first category of MAEC, "agent based EC" defines solutions as intelligent agents and enables agents to automatically select evolutionary operators or adjust control parameters. Besides these studies in MAEC, other similar approaches, which do not call themselves as agent-based algorithms, have been proposed in the filed of evolutionary computation (EC) with the same idea of algorithm adaption. In particular, a plethora of approaches have been attempted in the last 30 years to automate the choice of evolutionary operators [87, 132, 149–152], adaptation of control parameters [134, 153–161] and configuration of local search operators [49, 137–139, 162–164] on various problems, such as single-objective optimization problems, combinatorial optimization problems, etc. For surveys on some successful research efforts of adaptive EC, the reader is referred to [48, 165, 166]. Here, this study focuses particularly on those that address the multiobjective optimization problems (MOPs).

In the field of EC, different evolutionary operators show different competence on different problems [3, 167–171]. For instance, the SBX operator does not possess the property of rotational invariance since the correlation between the location of parents is lost under the rotation of a decision space. Thus, SBX is generally preferable for dealing with MOPs with independent variables [167]. On the contrary, polynomial mutation is suitable for handling MOPs with nonlinear dependent variables [3, 168]. By analyzing the properties of various strategies in Differential Evolution (DE) [124], "DE/rand/2/bin" has better perturbation than "DE/rand/1/bin" due to the two difference vectors based operator [169–171]. In comparison to "DE/rand/1/bin" and "DE/rand/2/bin", "DE/current-to-rand/1/bin" being a rotation-invariant operator, is generally more effective for solving MOPs [169, 170].

Besides the evolutionary operators, control parameters also play an important role. Even for the same operator, control parameters can induce different search biases. For instance, a large η_c in SBX or η_m in polynomial mutation tends to generate offspring solutions that are near their parents, which is beneficial for promoting solution convergence [9, 172]. According to [3, 170, 173, 174], in the later stage of MOEAs, larger CR values in DE operators are better to find diverse solutions, and smaller F values in DE operators are more suitable to find solutions in local regions. On the other hand, sequential parameter optimization (SPO) [175–178] is a useful tool to tune qualitative and quantitative parameters on various optimization models.

To fully utilize the competence of evolutionary operators and control parameters, some studies have been conducted to investigate algorithm adaption in the field of EC. For instance, parameter adaptation utilizes static rules that adapt parameters with a Gaussian distribution [179] or a piecewise nonlinear function along with the search (often with respect to the generation count) [180]. Others follow an online approach and adapt parameters based on some credit assignment criteria [130, 173]. Based on the historical records of successful parameters, JADE [173] generates new parameter values of CR and F in "DE/current-to-$p$best/bin" by following a Normal distribution and a Cauchy distribution, respectively. In [174], jDE is a self-adaptive method that encodes parameters into chromosome and evolves them along with the evolutionary searching procedure. However, the above approaches [130, 173, 174, 179, 180] consider the adaptation only on control parameters.

On the other hand, SaDE [169] and CoDE [171] are the two representative algorithms proposed to select evolutionary operators and adjust control parameters for solving single objective optimization problems. In SaDE, operators and parameters are gradually self-adapted by learning from their previous experience in generating promising solutions. In each generation, operators and parameters are assigned to different individuals in the current population according to the selection probabilities learned from the previous generations. However, SaDE computes simple statistics on the experience only after each 50 generations. In the proposed framework, the competency or suitability of evolutionary operators and control parameters (referred to as the trustworthiness of services) is modeled by cumulating all previous experience based on well established probabilistic modeling and in a dynamic manner. Also, SaDE adjusts control parameters based on a normal distribution with a predefined mean based on the authors' prior knowledge. The CoDE algorithm randomly combines three DE operators and three predefined parameters to generate offsprings. Although CoDE obtains better performance over SaDE, its setting of the control parameters relies on some prior knowledge. All in all, SaDE and CoDE both introduce some other parameters whose values are predefined based on previous studies on single objective optimization problems, which limits their generality to other problems, *e.g.*, more complex multiobjective optimization problems (MOPs). In contrast, the selection of operators and control parameters in the proposed framework does not rely on any prior knowledge about the problems. In addition, the proposed framework is represented as a multiagent system where candidate solutions are represented as intelligent agents capable of learning, cooperation and adaptation. This design offers great flexibility and extendability for EC to employ advanced multiagent technologies for solving complex optimization problems.

3.2 Background on MOEAs

To verify the proposed framework, the complex multiobjective optimization problems (MOPs) [2, 5–16] are selected as testing instances. In particular, MOPs involve several conflicting objectives to be optimized simultaneously. A minimization of MOPs can be stated as follows:

$$\begin{array}{rcl} min \;\; \mathcal{F}(\vec{x}) & = & (f_1(\vec{x}), \ldots, f_m(\vec{x})) \\ s.t. \;\; g(\vec{x}) & \leq & 0, \; h(\vec{x}) \; = \; 0, \; \vec{x} \; \in \; \Omega, \end{array} \tag{3.1}$$

where $\vec{x} = (x_1, \ldots, x_n)$, Ω is the decision (*variable*) *space*, R^m is the *objective space*, and $\mathcal{F} : \Omega \rightarrow R^m$ consists of m real-valued objective functions with constraints $g(\vec{x}) \leq 0, h(\vec{x}) = 0$, and the feasible solution space is $\Omega = \Pi_{i=1}^{n}[LB_i, UB_i]$.

The challenge of MOPs is to find a *Pareto set* (PS) including non-dominated solutions which are evenly scattered along *Pareto front* (PF). Multiobjective Evolutionary Algorithms (MOEAs) have been well established as efficient approaches to solve various MOPs [2].

In MOEAs, the first population of solutions is randomly generated as $X_g = \{\vec{x}_{i,g} | i = 1, \ldots, NP, g = 0\}$, where NP is the population size and g is the generation index. The next population is produced by evolutionary operators. Taking the "DE/rand/1/bin" operator [49] as an illustrative example, firstly, the operator generates a vector $\vec{v}_{i,g}$ based on population X_g.

$$\vec{v}_{i,g} = \vec{x}_{r1,g} + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}), \tag{3.2}$$

where $r1, r2, r3 \in [1, NP]$ are random integer numbers and $r1 \neq r2 \neq r3 \neq i$. The control parameter F is the scaling factor which amplifies or shrinks the difference vectors.

After that, "DE/rand/1/bin" applies the binomial crossover operation to produce the offspring vectors $U_g = \{\vec{u}_{i,j,g} | i = 1, \ldots, NP, j = 1, \ldots, n\}$.

$$\vec{u}_{i,j,g} = \begin{cases} \vec{v}_{i,j,g} & \text{if } \ rand_j(0,1) \leq CR \ \text{ or } \ j = j_{rand} \\ \vec{x}_{i,j,g} & \text{otherwise,} \end{cases} \tag{3.3}$$

where $rand_j(0,1) \in [0,1]$ is a uniformly distributed random number, $j_{rand} \in [1,n]$ is a randomly chosen integer. If $\vec{u}_{i,j,g} < LB_j$, it is set to LB_j, if $\vec{u}_{i,j,g} > UB_j$, set to UB_j. The control parameter CR is the probability for crossover.

Then, MOEAs select part of offsprings to enter the next generation ($\vec{u}_{i,g} \rightarrow X_{g+1}$). MOEAs can be generally categorized into two major classes: decomposition-based (called MOEA/D) [181] and Pareto dominance-based MOEAs [9, 10].

- In MOEA/D, $\vec{u}_{i,g} \rightarrow X_{g+1}$ if $\vec{u}_{i,g} \succeq \vec{x}_{j,g+1}$ ($\forall \vec{x}_{j,g+1} \in X_{g+1}$)[2] under, for example, the Tchebycheff approach [181].

- In Pareto dominance-based MOEAs, $\vec{u}_{i,g} \rightarrow X_{g+1}$ if $\vec{u}_{i,g} \succeq \vec{x}_{j,g+1}$ ($\forall \vec{x}_{j,g+1} \in X_{g+1}$) under, for example, the crowding distance (NSGAII) [9] or the neighborhood density estimator (SPEA2) [10].

The performance of MOEAs is determined by the operators and their parameters (*e.g.*, the operator "DE/rand/1/bin", and parameters F and CR in the operator mentioned earlier). The purpose of this study is to select proper evolutionary operators and control parameters in EC (*i.e.*, MOEAs).

3.3 Proposed Framework

In MOEAs, solutions in each generation produce solutions (offsprings) by performing evolutionary operators with some control parameters. The offsprings produced by some operators and parameters may be able to survive to the next generation, but some offsprings cannot.

In the proposed multiagent evolutionary framework, each solution is represented as an agent. The pairs of evolutionary operators with corresponding control parameters are

[2] "$\succeq$" means "be better than or equal".

represented as services. In each generation, an agent selects a service to produce a new offspring agent (*i.e.*, by Eqs. 3.2-3.3), which is also a solution. The new offspring agent competes with other agents in the environment. If the offspring agent can survive to the next generation, it means that the service provides a positive outcome, otherwise, the service provides a negative outcome. The trustworthiness of services can be used to represent the competency of the services in producing positive outcomes. The larger number of outcomes a service can produce, the more suitable the service is to solve the given problem. Thus, agents in the framework model the trustworthiness of the services based on the number of positive and negative outcomes provided by the services in the past generations. The modeling results will be used by the agents to make decisions on which services to consume.

3.3.1 Probabilistic Modeling of Trustworthiness

The trustworthiness of services is normally modeled based on the number of positive and negative outcomes produced by them in the past. The terms of s and f are defined as the number of positive and negative outcomes provided by a service S, formulated as follows:

$$\begin{cases} s = s + 1 & \text{if } \vec{u}_{i,g} \to X_{g+1} \\ f = f + 1 & \text{otherwise,} \end{cases} \tag{3.4}$$

where $\vec{u}_{i,g} \to X_{g+1}$ means that the offspring $\vec{u}_{i,g}$ produced in the generation g by the service can survive to the next generation $g + 1$. Whether $\vec{u}_{i,g} \to X_{g+1}$ is determined based on different methods in MOEAs (see Section 3.2).

The Beta distribution is commonly used to model the distribution of a random variable representing the unknown probability of a binary event. The Beta probability density functions (PDF) of service S can then be formulated as:

$$\mathrm{Beta}(p(S)|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p(S)^{\alpha-1}(1-p(S))^{\beta-1}, \tag{3.5}$$

where $0 \leq p(S) \leq 1$ and $\alpha, \beta > 0$ with the restriction that $p(S) \neq 0$ if $\alpha < 1$ and $p(S) \neq 1$ if $\beta < 1$.

The trustworthiness of S is then the probability expectation value of the Beta distribution, which represents the relative frequency of positive outcomes in future events [107].

$$T(S) = \frac{\alpha}{\alpha+\beta}, \quad \text{where} \quad \alpha = s+1, \ \beta = f+1. \tag{3.6}$$

3.3.2 Trustworthiness of Service

In this section, the trustworthiness of services is formulated by the probabilistic modeling introduced in Section 3.3.1. One thing to note here is that MOEAs generally involve much randomness. An evolutionary operator configured with the same control parameters may still generate different offsprings because of the random values of $r1$, $r2$, $r3$ and $rand_j(0,1)$ in Eqs. 3.2-3.3. Due to this randomness, the trustworthiness of a service cannot be accurately estimated by a small number of outcomes produced by the service for a particular family of agents (a solution and its offsprings). Instead, in the proposed framework, it is modeled based on the outcomes produced by the service for all agents in the past generations, which is referred to as reputation [107].

A service is represented by an evolutionary operator and some control parameters. The evolutionary operator can be any operator from a list of operators $O = \{O_1, O_2, \ldots, O_{|O|}\}$ proposed in MOEAs, where $|O|$ is the number of available evolutionary operators. Given a specific operator $O_k \in O$ in the service, there will be a set of control parameters $C^k = \{C_l^k | l = 1, \ldots, |C^k|\}$ associated with the operator O_k, where $|C^k|$ is the number of control parameters. For example, the operator "DE/rand/1/bin" has two control parameters (CR and F) associated with it (see Section 3.2). Assume that a parameter C_l^k takes a continuous value in the range as $C_l^k \in [0,1]$. In order to effectively learn the performance of a control parameter, the range $[0,1]$ is quantized into a set of q disjoint segments as $L = \{[0, \frac{1}{q}), [\frac{1}{q}, \frac{2}{q}), \cdots, [\frac{q-1}{q}, 1]\}$. Thus, a service can be formally defined as a tuple (O_k, C^k) where $C^k = \{C_l^k | C_l^k = L(C_l^k), l = 1, \ldots, |C^k|\}$ and $L(C_l^k)$ is one of the segments in L for the parameter C_l^k. In another word, a service is a tuple of an evolutionary operator and a set of segments for corresponding control parameters. For simplicity, a parameter from its segment is not distinguished.

For the service (O_k, C^k), the trustworthiness of the operator O_k is computed firstly. It is modeled based on the number of positive and negative outcomes generated by the agents performing this operator in the past generations. The total number of positive and negative outcomes up to the current generation g is aggregated as follows:

$$\left\{ \begin{array}{lcl} s_g(O_k) & = & (1-\eta) \cdot s_{g-1}(O_k) + \eta \cdot N_{g,s}(O_k) \\ f_g(O_k) & = & (1-\eta) \cdot f_{g-1}(O_k) + \eta \cdot N_{g,f}(O_k), \end{array} \right. \tag{3.7}$$

where $N_{g,s}(O_k)$ and $N_{g,f}(O_k)$ are the number of positive and negative outcomes produced by the agents performing the operator O_k in the current generation g, respectively. The parameter $0 \leq \eta \leq 1$ is to determine how much to consider the current and historical information, where $\eta = 0$ means that only the historical information is considered, whereas $\eta = 1$ only the current information is utilized. After having $s_g(O_k)$ and $f_g(O_k)$, the trustworthiness of the operator O_k in the current generation g, $T_g(O_k)$, can then be computed according to Eq. 3.6.

In general, one operator is suitable for some specific types of problems, but may not work well for other types. Even for the same problem, the competency of the operator may vary in different generations. For example, the operator "DE/ran/1/bin" is suitable to multi-modal problems, which has slow convergency in the earlier stage but exhibits strong exploration in the later stage of EC. Based on this phenomenon, the trustworthiness of the operator needs to reflect the varying competency of the operator under the condition where trust is hard to build up, but easy to lose.

The aggregation function in Eq. 3.7 is then revised as:

$$\begin{cases} s_g(O_k) &= (1 - T_{g-1}(O_k)) \cdot s_{g-1}(O_k) + T_{g-1}(O_k) \cdot N_{g,s}(O_k) \\ f_g(O_k) &= (1 - T_{g-1}(O_k)) \cdot f_{g-1}(O_k) + T_{g-1}(O_k) \cdot N_{g,f}(O_k), \end{cases} \tag{3.8}$$

where $T_{g-1}(O_k)$ is the trustworthiness of operator O_k in generation $g-1$. Eq. 3.8 has two important advantages. It does not have predefined parameters, compared to Eq. 3.7 that has the parameter η. Eq. 3.8 also satisfies the above mentioned condition. When the trustworthiness of the operator in the last generation $g-1$, $T_{g-1}(O_k)$ is low, the operator needs more positive outcomes $N_{g,s}(O_k)$ to build up its trust in the current generation g. So, $T_g(O_k)$ is not easy to build up. When $T_{g-1}(O_k)$ is high, $1 - T_{g-1}(O_k)$ is low, meaning that the less consideration will be given to historical information. The trustworthiness of the operator $T_g(O_k)$ will be easy to decline when the number of negative outcomes in the current generation $N_{g,f}(O_k)$ is large.

For the service (O_k, C^k), the trustworthiness of each parameter in C^k (*i.e.*, the value range segment corresponding to each parameter) is then computed. When computing the trustworthiness of a parameter, it also needs to consider the operator that the parameter is associated with. Take the parameter C_l^k as an example. The trustworthiness of C_l^k associated with O_k in the current generation g, denoted as $T_g(C_l^k|O_k)$, can be calculated in the similar way as calculating the trustworthiness of the operator O_k (Eq. 3.8), by counting the numbers of positive and negative outcomes produced by the operator O_k with the parameter C_l^k, which are $N_{g,s}(C_l^k|O_k)$ and $N_{g,f}(C_l^k|O_k)$ respectively.

After having the trustworthiness of the evolutionary operator O_k, which is $T_g(O_k)$, and each control parameter C_l^k given O_k, which is $T_g(C_l^k|O_k)$, the trustworthiness of the service (O_k, C^k) can be computed by assuming the control parameters are independent, as follows:

$$T_g(O_k, C^k) = T_g(O_k) \cdot \prod_{l=1}^{|C^k|} T_g(C_l^k|O_k). \tag{3.9}$$

3.3.3 Trust-based Service Selection

In the proposed framework, agents select services based on the computed trust results of the services. In order to balance between exploitation and exploration, services are selected in a probabilistic manner where the probability for a service to be selected is proportional to its trust. More formally, there are $\sum_k^{|O|} |C^k| \cdot q$ services in total because there are $|O|$ evolutionary operators, each operator O_k is associated with $|C^k|$ control parameters, and each parameter is represented by one of the q value range segments. The probability for service (O_k, C^k) with the trust $T_g(O_k, C^k)$ in the current generation g to be selected in the next generation $g+1$ is:

$$p(O_k, C^k) = \frac{T_g(O_k, C^k)}{\sum^{\sum_k^{|O|} |C^k| \cdot q} T_g(O_k, C^k)}. \tag{3.10}$$

Note that after an agent selects a service, *e.g.*, (O_k, C^k), each control parameter in C^k, *e.g.*, C_l^k, is a value range segment in L, not a specific value. In order for the service to be used by the agent to produce an offspring, a specific value for the parameter C_l^k is needed. This study assumes that the values of the parameter C_l^k follow a normal distribution in the range of $L(C_l^k)$ as Normal$(\mu_g(C_l^k|O_k), \sigma)$ where $\mu_g(C_l^k|O_k)$ and $\sigma = \frac{1}{3q}$ are the mean and standard deviation, respectively, and $C_l^k \in [0, 1]$. The mean $\mu_g(C_l^k|O_k)$ is calculated as follows:

$$\begin{aligned} \mu_g(C_l^k) \;&=\; (1 - T_{g-1}(C_l^k|O_k)) \cdot \mu_{g-1}(C_l^k) \\ &\quad + T_{g-1}(C_l^k|O_k) \cdot \mathrm{Mean}(V_g(C_l^k|O_k)) \end{aligned} \tag{3.11}$$

where $V_g(C_l^k|O_k)$ is the set of the values of the parameter C_l^k, which produces positive outcomes for the agent performing the operator O_k in the current generation g. Mean$(V_g(C_l^k|O_k))$ is the mean of the values in $V_g(C_l^k|O_k)$. The rationale behind Eq. 3.11 is that the effectiveness of the parameter C_l^k measured by $T_{g-1}(C_l^k|O_k)$, reflects the appropriation of its mean $\mu_g(C_l^k|O_k)$ up to the generation $g-1$. To cope with the dynamics of the effectiveness of $\mu_g(C_l^k|O_k)$, it is formulated in a similar spirit as Eq. 3.8.

3.3.4 Pseudo-code of the Proposed Framework

The pseudo-code of the multiagent evolutionary framework based on trust is summarized in Algorithm 1.

In Lines 1-3, the trustworthiness values of evolutionary operators and control parameters are initialized as $\frac{1}{|O|}$ and $\frac{1}{q}$, respectively. The parameter q is a predefined number of disjointed segments for the control parameters. The initialized positive and negative outcomes for difference services are 1. Line 4 randomly generates the first population and the agents are evaluated in Line 5. Lines 7-8 set the number of positive and negative outcomes of services for generation g as 0, which means no feedback coming from agents. The set $V_g(C_l^k|O_k)$ is initialized as a null set, which is design to store the values of parameter C_l^k based on O_k when the service generates a positive outcome. Line 10 selects a promising evolutionary operator and a control parameter by the probabilities $p(O_k, C_l^k)$ (as stated in Eq. 3.10). The larger probability means that the service (O_k, C_l^k) has greater competence to produce better offsprings to survive in the next generation. Line 12 generates a new offspring $u_{i,g}$. In Lines 14-15, if the new offspring survives in the next generation, the positive outcomes increase and the control parameters are stored into the set $V_g(C_l^k|O_k)$. Otherwise, the negative outcomes for operators and parameters increase by 1 in Lines 16-17. Lines 18-26 update the trustworthiness scores for evolutionary operators and control parameters. Line 27 adjusts the trustworthiness of services by combing the trust values of operators and parameters. In Line 28, when termination condition is satisfied, Algorithm 1 outputs the non-dominated solutions.

Input : $O = \{O_k | k = 1, \cdots, |O|\}$, a set of evolutionary operators;
$C = \{C_l^k | C_l^k \in [0,1], k = 1, \cdots, |O|, l = 1, \cdots, |C^k|\}$, a set of control parameters with operator O_k; q, segments of control parameters;
NP, population number; G, maximum number of generations;

Output : X, the non-dominated population;

$s_0(O_k) = 1, f_0(O_k) = 1, T_0(O_k) = \frac{1}{|O|}$ where $O_k \in O$
$\mu_0(C_l^k|O_k) = \{\frac{0.5}{q}, \frac{1.5}{q}, \cdots, \frac{q-0.5}{q}\}$, $\sigma = \frac{1}{3q}$
$s_0(C_l^k|O_k) = 1, f_0(C_l^k|O_k) = 1, T_0(C_l^k|O_k) = \frac{1}{q}$ where $C_l^k \in C$
Random initial population $X_0 = \{x_{i,0} | i = 1, \cdots, NP\}$
Evaluate the population X_0
for $g = 1$ *to* G **do**
  $N_{g,s}(O_k) = 0, N_{g,f}(O_k) = 0$
  $N_{g,s}(C_l^k|O_k) = 0, N_{g,f}(C_l^k|O_k) = 0$, $V_g(C_l^k|O_k) = \phi$
  for $i = 1$ *to* NP **do**
    Select operator O_k and parameter C^k by $p(O_k, C^k)$ using Eq. 3.10
    Generate parameters $C_l^k = \mathrm{Normal}(\mu_g(C_l^k|O_k), \sigma)$
    Generate offspring $\vec{u}_{i,g}$ by service (O_k, C_l^k)
    Evaluate offspring $\vec{u}_{i,g}$
    `// Assign Positive/Negative Outcomes for Services by Different Acceptance Rules`
    if $\vec{u}_{i,g} \to X_{g+1}$ **then**
      $N_{g,s}(O_k){+}{+}$; $N_{g,s}(C_l^k|O_k){+}{+}$; $C_l^k \to V_g(C_l^k|O_k)$
    else
      $N_{g,f}(O_k){+}{+}$; $N_{g,f}(C_l^k|O_k){+}{+}$
  `// Update Trustworthiness of Evolutionary Operators`
  $\eta = T_{g-1}(O_k)$
  $s_g(O_k) = (1-\eta) \cdot s_{g-1}(O_k) + \eta \cdot N_{g,s}(O_k)$
  $f_g(O_k) = (1-\eta) \cdot f_{g-1}(O_k) + \eta \cdot N_{g,f}(O_k)$
  $T_g(O_k) = \frac{s_g(O_k)}{s_g(O_k) + f_g(O_k)}$
  `// Update Trustworthiness of Control Parameters`
  $\eta = T_{g-1}(C_l^k|O_k)$
  $s_g(C_l^k|O_k) = (1-\eta) \cdot s_{g-1}(C_l^k|O_k) + \eta \cdot N_{g,s}(C_l^k|O_k)$
  $f_g(C_l^k|O_k) = (1-\eta) \cdot f_{g-1}(C_l^k|O_k) + \eta \cdot N_{g,f}(C_l^k|O_k)$
  $T_g(C_l^k|O_k) = \frac{s_g(C_l^k|O_k)}{s_g(C_l^k|O_k) + f_g(C_l^k|O_k)}$
  $\mu_g(C_l^k|O_k) = (1-\eta) \cdot \mu_{g-1}(C_l^k|O_k) + \eta \cdot \mathrm{Mean}(V_g(C_l^k|O_k))$
  `// Update Trustworthiness of Services`
  $T_g(O_k, C^k) = T_g(O_k) \cdot \prod_{l=1}^{|C^k|} T_g(C_l^k|O_k)$
Output non-dominated population X

Algorithm 1: Multiagent evolutionary framework based on trust

Table 3.1: The characteristics of the 58 benchmark MOPs.

Problem	n	m	Constraint	Pareto front (PF)	Pareto set (PS)
ZDT1	30	2	unconstraint	high dimensionality, convex	line segment, independent
ZDT2	30	2	unconstraint	high dimensionality, concave	line segment, independent
ZDT3	30	2	unconstraint	high dimensionality, disconnected, convex	line segment, independent
ZDT4	10	2	unconstraint	21^9 local Pareto fronts, multimodality, convex	line segment, independent
ZDT6	10	2	unconstraint	concave, concave	line segment, non-uniformly, independent
DTLZ1	7	3	unconstraint	$11^5 - 1$ local Pareto fronts, hyper-plane	rectangle, independent
DTLZ2	12	3	unconstraint	the first quadrant of unite sphere, concave	rectangle, independent
DTLZ3	12	3	unconstraint	$3^{10} - 1$ local Pareto fronts, concave	rectangle, independent
DTLZ4	12	3	unconstraint	biased density, concave	rectangle, independent
DTLZ5	12	3	unconstraint	local Pareto fronts, curve	rectangle, independent
DTLZ6	12	3	unconstraint	local Pareto fronts, curve	rectangle, independent
DTLZ7	22	3	unconstraint	4 disconnected parts	rectangle
WFG1	6	2	unconstraint	convex, mixed	independent
WFG2	6	2	unconstraint	convex, disconnected	independent
WFG3	6	2	unconstraint	linear, degenerate	independent
WFG4	6	2	unconstraint	concave	independent
WFG5	6	2	unconstraint	concave	independent
WFG6	6	2	unconstraint	concave	independent
WFG7	6	2	unconstraint	concave	independent
WFG8	6	2	unconstraint	concave	independent
WFG9	6	2	unconstraint	concave	independent
WFG1D3	6	2	unconstraint	convex, mixed	independent
WFG2D3	6	2	unconstraint	convex, disconnected	independent
WFG3D3	6	2	unconstraint	linear, degenerate	independent
WFG4D3	6	2	unconstraint	concave	independent
WFG5D3	6	2	unconstraint	concave	independent
WFG6D3	6	2	unconstraint	concave	independent
WFG7D3	6	2	unconstraint	concave	independent
WFG8D3	6	2	unconstraint	concave	independent
WFG9D3	6	2	unconstraint	concave	independent
LZ09F1	30	2	unconstraint	convex	dependent
LZ09F3	30	2	unconstraint	convex	dependent
LZ09F4	30	2	unconstraint	convex	dependent
LZ09F7	10	2	unconstraint	convex	dependent
LZ09F9	30	2	unconstraint	concave	dependent
UF1	30	2	unconstraint	convex	nonlinear, dependent
UF2	30	2	unconstraint	convex	nonlinear, dependent
UF3	30	2	unconstraint	convex	nonlinear, local PS, dependent
UF4	30	2	unconstraint	concave	nonlinear, dependent
UF5	30	2	unconstraint	20 discrete points, linear	20 discrete points, local PS, dependent
UF6	30	2	unconstraint	1 isolated point, 2 disconnect parts, linear	discrete points, local PS, dependent
UF7	30	2	unconstraint	linear	nonlinear, dependent
UF8	30	3	unconstraint	concave	nonlinear, dependent
UF9	30	3	unconstraint	2 parts, hyper-plane	2 nonlinear parts, local PS, dependent
UF10	30	3	unconstraint	concave	nonlinear, dependent
UF11	30	5	unconstraint	nonseparable, multi-modal, rotated, concave	non symmetric, dependent
UF12	30	5	unconstraint	nonseparable, multi-modal, rotated, concave	non symmetric, dependent
UF13	30	5	unconstraint	separable, uni-modal, mixed, convex	non symmetric, dependent
CF1	10	2	1 constraint	10 discrete points, linear	nonlinear, dependent
CF2	10	2	1 constraint	1 isolate point, 2 disconnected parts, convex	nonlinear, dependent
CF3	10	2	1 constraint	1 isolate point, 2 disconnected parts, concave	nonlinear, local PS, dependent
CF4	10	2	1 constraint	3 connected parts, piecewise linear	nonlinear, dependent
CF5	10	2	1 constraint	3 connected parts, piecewise linear	nonlinear, dependent
CF6	10	2	2 constraints	3 connected parts, piecewise curve, convex	nonlinear, dependent
CF7	10	2	2 constraints	3 connected parts, piecewise curve, convex	nonlinear, local PS, dependent
CF8	10	3	1 constraint	5 disconnected curves, nonlinear, concave	nonlinear, dependent
CF9	10	3	1 constraint	2 disconnected 2D surfaces, 1 curve, concave	nonlinear, dependent
CF10	10	3	1 constraint	2 disconnected 2D surfaces, 1 curve, concave	nonlinear, local PS, dependent

3.4 Experimentation

The experiments are carried out on the standard platform jMetal 4.3[3], which is a Java-based framework aimed at facilitating the development of metaheuristics for solving MOPs [12]. The performance metric is set as the hypervolume indicator [4, 101, 182].

The benchmark problems include 58 test instances: 5 MOPs in the ZDTx family problems (ZDT1–4 and ZDT6 with 2 objectives), 7 MOPs in the DTLZx family problems (DTLZ1–7 with 3 objectives), 9 MOPs in the WFGx family problems (WFG1–9 with 2 objectives), 9 MOPs in the WFGx family problems (WFG1–9D3 with 3 objectives), 4 MOPs in the LZ09Fx family problems (LZ09F1, 3, 4, 7, 9 with 2 objectives)[4] [181] and 23 MOPs in the CEC2009 MOEA competition. Among the problems used in the CEC2009 MOEA competition that involves unconstrained functions, UF1–7 have 2 objectives, UF8–10 3 objectives, and UF11–13 5 objectives. In addition, the problems CF1–10 have one constraint but CF6–7 have two constraints. The *decision variables* in the Pareto sets (PSs) of ZDTx, DTLZx, WFGx and LZ09F are independent, and those in the CEC2009 MOEA competition are dependent. On the other hand, the 58 MOPs have different geometrical shapes in *objective space* such as concave, convex, linear, discrete, uni-modal and multi-modal Pareto fronts (PFs). Table 3.1 gives the details of the 58 MOPs in terms of number of *decision variables*(n), the number of objectives (m), the number of constraints, the shape of Pareto fronts (PFs) and Pareto Sets(PSs).

The major experimental settings are outlined as follows.

(i) Population size: In MOEA/D [11], the population size is decided by the number of weight vectors C_{H+m-1}^{m-1} (m is the objective number, H is a predefined integer). $NP = 100, 153, 715$ for $m = 2, 3, 5$ objectives ($H = 99, 16, 9$), respectively. Other algorithms have the same population size as MOEA/D.

(ii) Maximum function evaluations: Max_FES $= 300,000$

(iii) Independent run times: $runs = 30$

[3]http://jmetal.sourceforge.net

[4]LZ09F2, 5, 6, 8 in the LZ09F family problems are omitted because they are the same as UF1, 2, 3, 8 in the CEC2009 MOEA competition, respectively.

(iv) Evolutionary operators in NSGAII [9] and SPEA2 [10]:

- SBX: $p_c = 0.9$, $\eta_c = 20$
- Polynomial mutation: $p_m = 1/n$, $\eta_m = 20$

(v) DE operators in MOEA/D [11]: $CR = 1.0, F = 0.4$

(vi) Update approaches in decomposition-based MOEAs: the Tchebycheff approach ($m = 2$) and the Penalty-based Boundary Intersection approach ($m \geq 3$) [11]

(vii) Quantization size in the proposed approach: $q = 3$

Other parameters are set as the default values in jMetal [12].

In the experimental studies, the proposed framework is compared with the three classic MOEAs (NSGAII [9], SPEA2 [10] and MOEA/D [181]), and the other approaches (JADE [135, 173], jDE [133, 174], SaDE [169, 170], CoDE [171]) that select evolutionary operators and/or control parameters. Five evolutionary operators are considered, including "DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1/bin" [171], "SBX", and "Polynomial mutation" [9]:

(i) "DE/rand/1/bin"

$$\vec{v}_{i,g} = \vec{x}_{r1,g} + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}). \tag{3.12}$$

(ii) "DE/rand/2/bin"

$$\vec{v}_{i,g} = \vec{x}_{r1,g} + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}) + F \cdot (\vec{x}_{r4,g} - \vec{x}_{r5,g}). \tag{3.13}$$

(iii) "DE/current-to-rand/1/bin"

$$\vec{v}_{i,g} = \vec{x}_{i,g} + rand() \cdot (\vec{x}_{r1,g} - \vec{x}_{i,g}) + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}). \tag{3.14}$$

The above three DE operators apply the binomial crossover to produce offsprings as $\vec{u}_{i,g} = \{u_{i,j,g} | j = 1, \ldots, n\}$, $i \in [1, NP]$ and g is the generation index, where

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } rand_j() \leq CR \text{ or } j = j_{rand} \\ x_{i,j,g} & \text{otherwise.} \end{cases}$$

(iv) "SBX"

$$\vec{u}_{i,g} = \begin{cases} \vec{v}_{i,g} & \text{if } rand() < p_c \\ \vec{x}_{r2,g} & \text{otherwise,} \end{cases} \tag{3.15}$$

where $\vec{v}_{i,g} = \frac{1}{2}[(\vec{x}_{r1,g} + \vec{x}_{r2,g}) \pm \beta_q \cdot (\vec{x}_{r1,g} - \vec{x}_{r2,g})]$, and

$$\beta_q = \begin{cases} (2 \cdot rand())^{\frac{1}{\eta_c+1}} & \text{if } rand() \leq 0.5 \\ (\frac{1}{2\cdot(1-rand())})^{\frac{1}{\eta_c+1}} & \text{otherwise.} \end{cases}$$

(v) "Polynomial mutation"

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } rand() < p_m \\ x_{i,j,g} & \text{otherwise,} \end{cases} \tag{3.16}$$

where $v_{i,j,g} = x_{i,j,g} + (U_j - L_j) \cdot \delta_j$, and

$$\delta_j = \begin{cases} (2 \cdot rand())^{\frac{1}{\eta_m+1}} - 1 & \text{if } rand() < 0.5 \\ 1 - |2 \cdot (1 - rand())|^{\frac{1}{\eta_m+1}} & \text{otherwise.} \end{cases}$$

Here, $r1, r2, r3, r4, r5 \in [1, NP]$ are random integer numbers and $r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i$. $rand() \in [0, 1]$ is a uniform random value. The control parameter F is the scaling factor, which amplifies or shrinks the difference vectors. In SBX, $\eta_c \in [0, n]$ determines the distance of children from their parent. In polynomial mutation, $\eta_m \in [0, n]$ defines the polynomial probability distribution. CR, p_c, $p_m \in [0, 1]$ are crossover/mutation probabilities in DE operators, SBX and polynomial mutation, respectively. If the value $u_{i,j,g}$ exceeds the lower or upper bound, it is set as L_j or U_j, respectively.

All the algorithms are evaluated based on hypervolume, which is the only single set quality measure known to be strictly monotonic with regard to Pareto dominance [4, 182]. The obtained results are compared using median values and interquartile range (IQR). In order to obtain statistically sound conclusions, the Wilcoxon rank sum test with 95% confidence level is conducted on the experimental results. All the algorithms are compared under the following two aspects.

- *Robustness*. Under the Wilcoxon statistical test, the algorithm generates higher hypervolume than other approaches on different types of MOPs.

- *Ease-of-use*. A simple approach requires minimum effort to develop, as well as a minimum number of predefined parameters to be managed or specified by users.

Table 3.2: Hypervolume median and IQR of classical MOEAs and those extended by the proposed framework over 30 independent runs on 58 MOPs with 300,000 FES.

MOPs	NSGAII	NSGAII-T	SPEA2	SPEA2-T	MOEA/D	MOEA/D-T
ZDT1	6.60E-01±2.29E-04−	6.61E-01±3.15E-04−	6.62E-01±1.02E-04+	6.62E-01±2.76E-05+	6.61E-01±1.31E-04−	6.62E-01±4.95E-07
ZDT2	3.27E-01±2.30E-04−	3.27E-01±4.02E-04−	3.28E-01±1.35E-04≈	3.29E-01±3.40E-05+	3.28E-01±1.16E-04−	3.28E-01±1.18E-08
ZDT3	5.15E-01±1.72E-04+	5.15E-01±1.95E-04+	5.16E-01±8.90E-05+	5.16E-01±2.94E-05+	5.14E-01±2.01E-05−	5.14E-01±2.17E-06
ZDT4	6.61E-01±2.79E-04−	6.60E-01±3.03E-04−	6.62E-01±5.16E-05+	4.95E-01±1.67E-01≈	6.61E-01±3.35E-04−	6.62E-01±1.05E-04
ZDT6	3.98E-01±5.50E-04−	3.98E-01±3.83E-04−	4.01E-01±5.27E-04−	4.01E-01±2.74E-05+	4.01E-01±2.11E-07−	4.01E-01±4.72E-09
DTLZ1	7.78E-01±3.13E-03≈	7.80E-01±3.28E-03≈	7.97E-01±4.60E-04≈	7.97E-01±2.10E-04≈	7.93E-01±1.24E-03≈	4.51E-01±6.94E-01
DTLZ2	3.94E-01±4.22E-03−	4.01E-01±4.96E-03−	4.19E-01±2.00E-03−	4.29E-01±6.77E-04+	4.09E-01±2.46E-03−	4.29E-01±1.78E-06
DTLZ3	3.98E-01±5.79E-03+	4.00E-01±4.02E-01+	4.28E-01±8.59E-04+	4.29E-01±9.35E-04+	3.34E-01±7.03E-02−	3.93E-01±3.15E-04
DTLZ4	3.95E-01±4.41E-03−	3.97E-01±5.24E-03−	4.13E-01±2.08E-03−	4.22E-01±8.73E-04+	4.00E-01±3.33E-03−	4.22E-01±1.52E-06
DTLZ5	9.41E-02±1.38E-04+	9.44E-02±9.18E-05+	9.44E-02±5.45E-05+	9.47E-02±2.55E-05+	8.06E-02±6.92E-05+	8.04E-02±1.59E-07
DTLZ6	9.52E-02±1.02E-04+	9.52E-02±1.10E-04+	9.56E-02±3.04E-05+	9.56E-02±2.16E-05+	8.12E-02±3.30E-07≈	8.12E-02±1.20E-07
DTLZ7	2.98E-01±2.85E-03+	3.05E-01±2.59E-03+	3.07E-01±1.76E-03+	3.12E-01±1.52E-03+	2.62E-01±3.99E-03−	2.67E-01±9.47E-04
WFG1	6.30E-01±4.37E-04−	6.32E-01±3.56E-04−	6.32E-01±1.19E-01−	6.24E-01±1.86E-03−	6.27E-01±3.22E-03−	6.33E-01±6.43E-05
WFG2	5.64E-01±2.70E-03≈	5.64E-01±8.40E-05+	5.64E-01±2.79E-03≈	5.64E-01±3.46E-05+	5.63E-01±4.84E-05−	5.63E-01±2.41E-07
WFG3	4.41E-01±1.38E-04−	4.41E-01±1.64E-04−	4.42E-01±1.15E-04+	4.42E-01±1.62E-05+	4.42E-01±1.74E-05−	4.42E-01±2.41E-09
WFG4	2.18E-01±3.35E-04−	2.18E-01±3.75E-04≈	2.19E-01±2.48E-04+	2.19E-01±9.47E-05+	2.18E-01±2.07E-04−	2.18E-01±1.39E-06
WFG5	1.95E-01±5.05E-04≈	1.95E-01±4.53E-04≈	1.96E-01±1.64E-04+	1.96E-01±1.11E-04+	1.95E-01±3.87E-05−	1.95E-01±6.56E-05
WFG6	2.04E-01±5.60E-03−	2.09E-01±2.64E-04−	2.03E-01±7.41E-03−	2.10E-01±8.44E-05+	2.09E-01±3.04E-05−	2.10E-01±1.69E-07
WFG7	2.09E-01±2.12E-04−	2.09E-01±4.54E-04−	2.10E-01±2.05E-04+	2.10E-01±9.61E-05+	2.09E-01±1.88E-05−	2.10E-01±1.11E-06
WFG8	1.52E-01±8.90E-04−	1.52E-01±7.80E-04−	1.54E-01±9.13E-04−	1.51E-01±1.52E-03−	1.54E-01±9.26E-04−	2.07E-01±5.26E-02
WFG9	2.38E-01±1.99E-03−	2.38E-01±1.51E-03−	2.39E-01±2.01E-03≈	2.39E-01±1.27E-03≈	2.40E-01±1.14E-04≈	2.39E-01±1.03E-03
WFG1D3	9.08E-01±3.54E-03≈	9.20E-01±3.39E-03+	9.15E-01±1.38E-03+	7.62E-01±3.59E-02−	8.44E-01±5.55E-02−	9.07E-01±5.89E-03
WFG2D3	9.09E-01±2.51E-03−	9.13E-01±2.37E-03−	9.19E-01±1.02E-03+	9.21E-01±6.55E-04+	9.13E-01±1.23E-03−	9.16E-01±2.82E-04
WFG3D3	3.19E-01±1.68E-03+	3.20E-01±1.50E-03+	3.04E-01±4.39E-03+	3.09E-01±3.46E-03+	2.87E-01±7.18E-04−	2.87E-01±1.14E-04
WFG4D3	3.89E-01±4.73E-03−	3.89E-01±4.35E-03−	4.11E-01±3.03E-03−	4.27E-01±1.18E-03+	3.80E-01±3.05E-03−	4.17E-01±1.04E-03
WFG5D3	3.62E-01±4.77E-03−	3.66E-01±4.93E-03−	3.85E-01±1.64E-03+	3.83E-01±8.37E-04+	3.72E-01±1.64E-03−	3.81E-01±6.81E-03
WFG6D3	3.94E-01±5.60E-03−	4.01E-01±4.81E-03−	4.21E-01±1.86E-03+	4.33E-01±5.60E-04+	4.08E-01±1.70E-03−	4.17E-01±2.29E-03
WFG7D3	3.80E-01±6.79E-03−	3.83E-01±5.38E-03−	4.01E-01±3.17E-03−	4.08E-01±2.49E-03+	4.02E-01±1.94E-03−	4.06E-01±2.08E-03
WFG8D3	2.59E-01±7.27E-03−	2.59E-01±5.10E-03−	2.99E-01±5.62E-03≈	2.98E-01±4.52E-03−	2.86E-01±9.46E-02−	3.91E-01±7.33E-02
WFG9D3	3.79E-01±4.31E-03−	3.79E-01±3.88E-03−	3.95E-01±4.55E-03−	3.96E-01±2.30E-03−	3.97E-01±1.07E-03−	4.00E-01±2.94E-03
LZ09F1	6.49E-01±9.18E-04−	6.54E-01±8.68E-04−	6.55E-01±1.90E-03−	6.56E-01±6.01E-04−	6.61E-01±8.94E-05−	6.61E-01±6.62E-05
LZ09F3	6.25E-01±4.11E-03−	6.47E-01±1.95E-03−	6.20E-01±6.81E-03−	6.35E-01±3.65E-03−	6.50E-01±8.90E-03−	6.53E-01±2.55E-03
LZ09F4	6.35E-01±2.65E-03−	6.47E-01±7.24E-04−	6.32E-01±3.49E-03−	6.32E-01±4.98E-03−	6.57E-01±9.73E-04+	6.53E-01±5.27E-03
LZ09F7	4.87E-01±6.06E-02−	6.53E-01±9.90E-04−	4.72E-01±7.53E-02−	4.83E-01±5.57E-02−	6.58E-01±1.77E-03−	6.59E-01±2.37E-03
LZ09F9	2.43E-01±1.48E-02−	3.00E-01±7.86E-03−	2.24E-01±6.19E-03−	2.42E-01±1.11E-02−	3.21E-01±5.55E-03+	3.18E-01±4.89E-03
UF1	5.67E-01±3.85E-02−	6.40E-01±1.61E-02−	5.46E-01±2.26E-02−	5.83E-01±3.27E-02−	6.57E-01±1.51E-03+	6.56E-01±2.43E-03
UF2	6.34E-01±8.51E-03−	6.48E-01±1.93E-03−	6.33E-01±9.58E-03−	6.42E-01±3.45E-03−	6.45E-01±2.01E-02−	6.54E-01±2.15E-03
UF3	4.82E-01±5.66E-02−	6.35E-01±1.51E-02−	4.29E-01±6.86E-02−	4.63E-01±8.06E-02−	6.31E-01±1.95E-02−	6.50E-01±8.98E-03
UF4	2.65E-01±8.74E-04−	2.74E-01±2.39E-03−	2.71E-01±7.96E-04−	2.80E-01±3.86E-04+	2.30E-01±1.10E-02−	2.77E-01±1.51E-03
UF5	1.68E-01±8.22E-02−	3.30E-01±2.84E-02+	1.76E-01±9.89E-02−	1.04E-01±1.59E-01−	6.89E-02±1.57E-01−	2.53E-01±3.76E-02
UF6	2.35E-01±4.81E-02≈	2.50E-01±7.13E-02≈	2.38E-01±5.43E-02≈	2.17E-01±1.33E-01−	2.36E-01±9.69E-02≈	2.53E-01±5.36E-02
UF7	2.85E-01±1.99E-01−	4.76E-01±3.54E-03−	4.37E-01±1.75E-01−	4.51E-01±1.03E-02−	4.88E-01±3.79E-03≈	4.88E-01±2.45E-03
UF8	1.33E-01±1.18E-01−	1.34E-01±1.51E-01−	1.58E-01±1.91E-02−	2.55E-01±8.87E-02−	2.06E-01±9.32E-02−	3.02E-01±3.12E-03
UF9	3.83E-01±1.62E-01−	3.58E-01±2.16E-01−	5.72E-01±9.89E-02≈	5.95E-01±1.05E-02≈	5.64E-01±1.08E-01≈	5.49E-01±1.19E-01
UF10	2.20E-02±3.47E-02−	9.75E-03±4.31E-02−	3.10E-02±5.15E-02−	1.19E-01±6.66E-02≈	7.06E-02±3.71E-02−	1.32E-01±7.59E-02
UF11	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	3.08E-04±2.86E-04
UF12	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
UF13	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
CF1	4.65E-01±1.12E-03+	4.71E-01±8.56E-04+	4.62E-01±3.28E-03+	4.40E-01±6.04E-03≈	4.39E-01±1.33E-05≈	4.39E-01±2.00E-02
CF2	5.87E-01±3.13E-02−	6.10E-01±7.49E-04−	5.64E-01±3.07E-02−	5.72E-01±2.35E-02−	6.50E-01±1.35E-03−	6.51E-01±1.17E-03
CF3	9.73E-02±6.40E-02−	1.44E-01±5.14E-02≈	9.80E-02±6.24E-02−	1.28E-01±3.08E-02≈	1.31E-01±6.40E-02≈	1.46E-01±6.82E-02
CF4	1.74E-01±9.28E-02−	2.34E-01±6.50E-02−	2.94E-02±1.25E-01−	1.66E-03±8.83E-02−	5.46E-01±6.71E-03−	5.49E-01±4.00E-03
CF5	6.65E-02±1.43E-01−	1.15E-01±1.83E-01−	0.00E-00±9.04E-02−	0.00E-00±0.00E-00−	3.81E-01±1.25E-01≈	3.56E-01±1.72E-01
CF6	4.10E-01±1.19E-01−	3.77E-01±8.60E-02−	1.65E-01±2.95E-01−	1.90E-01±1.87E-01−	6.45E-01±1.56E-02−	6.58E-01±1.96E-03
CF7	1.39E-01±2.48E-01−	2.42E-01±2.10E-01−	0.00E-00±1.29E-01−	0.00E-00±0.00E-00−	4.64E-01±1.19E-01−	5.56E-01±4.47E-02
CF8	4.67E-02±9.99E-02−	1.61E-01±5.50E-02−	2.32E-01±2.39E-01≈	2.36E-01±1.69E-01≈	2.31E-01±4.65E-02+	2.04E-01±5.07E-02
CF9	1.94E-01±7.37E-02−	2.67E-01±1.52E-02−	2.88E-01±1.55E-02−	2.97E-01±1.56E-02−	2.71E-01±2.62E-02−	3.25E-01±2.91E-02
CF10	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	1.42E-01±8.81E-02−	2.07E-01±6.46E-02
Mean	3.47E-01±3.07E-02	3.71E-01±2.91E-02	3.51E-01±3.19E-02	3.52E-01±2.43E-02	3.92E-01±2.25E-02	4.03E-01±2.80E-02

+ , ≈ and − represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.

Table 3.3: Statistical results of classical MOEAs versus those extended by trust.

	NSGAII-T	SPEA2	SPEA2-T	MOEA/D	MOEA/D-T
NSGAII	35 21 2	33 14 11	37 9 12	38 8 12	44 7 7
NSGAII-T		29 6 23	32 6 20	33 7 18	41 7 10
SPEA2			33 18 7	22 10 26	31 10 17
SPEA2-T				18 12 28	25 10 23
MOEA/D					42 11 5

3.4.1 Comparison with the Classical MOEAs

In this experiment, the three classical MOEAs (NSGAII [9], SPEA2 [10] and MOEA/D [11]) are extended by the proposed framework that adapt the five evolutionary operators and control parameters based on the concept of trust. The extended versions of NSGAII, SPEA2 and MOEA/D are referred to as NSGAII-T, SPEA2-T and MOEA/D-T, respectively.

Table 3.2 reports the detailed experimental results, where each tuple tabulates the median and IQR of hypervolume over 30 independent runs on 58 MOPs with a maximum 300,000 function evaluations (FES). Table 3.3 shows the win/tie/lose ($w/t/l$) statistical results under the Wilcoxon rank sum test with 95% confidence level. Each tuple $w/t/l$ means that the algorithm at the corresponding column wins on w MOPs, ties on t MOPs, and loses on l MOPs, compared to the algorithm at the corresponding row.

In Table 3.3, the $w/t/l$ values between the three extended versions by the proposed framework and the classical MOEAs are 35/21/2, 33/18/7, 42/11/5, respectively. The performance of the extended versions shows improvements over the classical MOEAs on 55.2% of the 58 MOPs. This indicates that the proposed framework can significantly improve the *robustness* of classical MOEAs.

In addition, the *acceptance rule* that decides which solutions would survive into the next population is a key factor to influence the performance of MOEAs. From Table 3.2, SPEA2-T exhibits higher hypervolume than MOEA/D-T on DTLZ and WFG family problems, whose Pareto Fronts (PFs) share the similar geometric characteristic as $f_1^2 + \cdots + f_m^2 = 1$. The reason is that the *k-nearest neighborhood density estimator* in SPEA2 makes it more suitable to solve PFs with circular ($m = 2$) and spherical shapes ($m = 3$).

3.4.2 Comparison with the Adaptive MOEAs

In this section, several state-of-the-art adaptive DE variants (JADE [135, 173], jDE [133, 174], SaDE [169, 170], CoDE [171]) are involved for comparison and use decomposition-based MOEAs as the baseline. The goal is to verify their effectiveness against the proposed framework.

JADE and jDE only adapt the control parameters. Each of them uses a fixed evolutionary operator, namely "DE/current-to-$p$best/bin" and "DE/rand/2/bin", respectively. In JADE, the control parameters CR and F are generated by a Normal distribution ($\sigma_{CR} = 0.1$) and a Cauchy distribution ($\sigma_F = 0.1$), respectively. JADE introduces three new parameters that include another variable $c = 0.1$. In jDE, the values CR and F are encoded as part of the individuals which are co-evolved along with the search. jDE also involves four predefined parameters such as $F_l = 0.1, F_u = 0.9, \tau_1 = \tau_2 = 0.1$.

SaDE and CoDE select evolutionary operators and adapt control parameters simultaneously. Both of them design the operator pool that includes "DE/rand/1/bin", "DE/rand/2/bin" and "DE/current-to-rand/1/bin". In SaDE, the control parameters in DE are generated by Normal distribution, where $\sigma_{CR} = 0.3$, $\mu_F = 0.5$ and $\sigma_F = 0.1$. SaDE includes another predefined parameter (*i.e.*, the learning period of 50 generations). In CoDE, the three operators are combined with a set of fixed parameter settings, including $[CR = 0.1, F = 0.1]$, $[CR = 1.0, F = 0.5]$ and $[CR = 0.2, F = 0.8]$. These three parameter settings are identified based on empirical studies. The first and second settings encourage exploitation and exploration, respectively, while $[CR = 0.2, F = 0.8]$ is suitable for solving Pareto Sets (PSs) with interdependent variables [171].

In contrast to the previous adaptive approaches, a pool of evolutionary operators is considered including "DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1/bin", "SBX" and "polynomial mutation". To facilitate a fair comparison, This study also implements MOEA/D-T3 that uses the proposed framework to select among the first three operators, which is consistent with SaDE and CoDE. Furthermore, the proposed framework has only one predefined parameter (quantization size q). Based on the spirit of *ease-of-use*, this is in contrast to the previous adaptive approaches which have at least three or more parameters.

Table 3.4 reports detailed results of MOEA/D-JADE, MOEA/D-jDE, MOEA/D-SaDE, MOEA/D-CoDE, MOEA/D-T3 and MOEA/D-T. Table 3.5 on the other hand, summarizes the win/tie/lose ($w/t/l$) statistical comparison results.

In Table 3.5, the $w/t/l$ values between MOEA/D-T3 and MOEA/D-JADE, MOEA/D-jDE are 25/27/6 and 26/24/8, respectively. It indicates that operator selection is beneficial for MOEAs to solve various MOPs. The $w/t/l$ values between MOEA/D-T3 and

Table 3.4: Hypervolume median and IQR by adaptive MOEAs with ecomposition-based approaches over 30 independent runs on 58 MOPs with 300,000 FES.

MOPs	MOEA/D-JADE	MOEA/D-jDE	MOEA/D-SaDE	MOEA/D-CoDE	MOEA/D-T3	MOEA/D-T
ZDT1	6.62E-01±1.16E-05−	6.62E-01±9.96E-06−	6.62E-01±1.18E-05−	6.62E-01±1.38E-05−	6.62E-01±9.59E-07≈	6.62E-01±4.95E-07
ZDT2	3.28E-01±4.57E-06−	3.28E-01±7.88E-06−	3.28E-01±2.82E-06−	3.28E-01±1.01E-05−	3.28E-01±3.81E-08≈	3.28E-01±1.18E-08
ZDT3	5.14E-01±1.47E-05−	5.14E-01±2.28E-05−	5.14E-01±4.69E-06−	5.14E-01±9.04E-06−	5.14E-01±6.06E-06≈	5.14E-01±2.17E-06
ZDT4	3.82E-02±3.41E-01−	3.73E-03±9.99E-02−	3.41E-01±2.86E-01−	0.00E-00±3.73E-03−	1.08E-01±3.07E-01−	6.62E-01±1.05E-04
ZDT6	4.01E-01±5.94E-06−	4.01E-01±5.76E-08−	4.01E-01±4.35E-07−	4.01E-01±8.47E-08−	4.01E-01±3.80E-09≈	4.01E-01±4.72E-09
DTLZ1	7.98E-01±1.10E-07≈	7.98E-01±5.41E-07+	7.98E-01±1.94E-07+	1.04E-01±1.04E-01−	7.98E-01±6.94E-01≈	4.51E-01±6.94E-01
DTLZ2	4.29E-01±4.87E-06≈	4.29E-01±8.90E-06−	4.29E-01±4.33E-06+	4.29E-01±7.94E-06≈	4.29E-01±3.20E-06≈	4.29E-01±1.78E-06
DTLZ3	4.29E-01±4.29E-06+	0.00E-00±0.00E-00−	0.00E-00±4.29E-01≈	0.00E-00±0.00E-00−	0.00E-00±4.29E-01−	3.93E-01±3.15E-04
DTLZ4	4.22E-01±3.25E-06≈	4.22E-01±1.38E-05≈	4.22E-01±6.49E-06≈	4.22E-01±1.25E-05−	4.22E-01±2.19E-06≈	4.22E-01±1.52E-06
DTLZ5	8.04E-02±1.46E-06≈	8.04E-02±1.61E-06+	8.04E-02±6.01E-07≈	8.04E-02±8.49E-07≈	8.04E-02±1.53E-07≈	8.04E-02±1.59E-07
DTLZ6	8.12E-02±8.31E-07≈	8.12E-02±2.88E-07≈	8.12E-02±2.70E-07≈	8.12E-02±4.69E-07≈	8.12E-02±9.58E-08≈	8.12E-02±1.20E-07
DTLZ7	2.67E-01±8.26E-04≈	2.67E-01±1.16E-03≈	2.67E-01±9.64E-04−	2.67E-01±1.07E-03≈	2.67E-01±8.96E-04≈	2.67E-01±9.47E-04
WFG1	6.33E-01±2.64E-05≈	6.25E-01±1.04E-03−	6.30E-01±2.69E-03−	6.27E-01±3.75E-04−	6.33E-01±8.35E-05≈	6.33E-01±6.43E-05
WFG2	5.63E-01±2.28E-07−	5.63E-01±4.76E-07+	5.63E-01±2.84E-07≈	5.63E-01±6.73E-07+	5.63E-01±1.90E-07≈	5.63E-01±2.41E-07
WFG3	4.42E-01±2.06E-08−	4.42E-01±1.77E-06+	4.42E-01±5.57E-07+	4.42E-01±9.90E-07+	4.42E-01±3.20E-09+	4.42E-01±2.41E-09
WFG4	2.18E-01±1.66E-06≈	2.18E-01±1.45E-06≈	2.18E-01±9.82E-07−	2.18E-01±1.70E-04−	2.18E-01±1.70E-04≈	2.18E-01±1.39E-06
WFG5	1.95E-01±6.23E-05≈	1.95E-01±4.28E-05≈	1.95E-01±3.20E-05≈	1.95E-01±5.80E-05≈	1.95E-01±4.13E-05≈	1.95E-01±6.56E-05
WFG6	2.10E-01±8.83E-06−	2.10E-01±1.54E-06−	2.10E-01±2.05E-06−	2.10E-01±4.93E-06−	2.10E-01±1.42E-07+	2.10E-01±1.69E-07
WFG7	2.10E-01±1.33E-06−	2.10E-01±1.91E-06−	2.10E-01±1.47E-06−	2.10E-01±1.31E-06−	2.10E-01±5.30E-07+	2.10E-01±1.11E-06
WFG8	1.54E-01±5.29E-02−	2.06E-01±5.23E-02≈	1.55E-01±5.33E-02≈	1.54E-01±5.30E-02≈	1.54E-01±5.28E-02−	2.07E-01±5.26E-02
WFG9	2.39E-01±1.33E-03≈	2.40E-01±1.04E-03≈	2.39E-01±9.87E-04≈	2.40E-01±1.05E-03+	2.39E-01±1.08E-03≈	2.39E-01±1.03E-03
WFG1D3	8.95E-01±6.28E-03−	9.01E-01±1.92E-02−	8.96E-01±4.03E-03−	8.94E-01±1.25E-02−	9.03E-01±4.60E-03−	9.07E-01±5.89E-03
WFG2D3	9.16E-01±3.68E-04≈	9.16E-01±3.22E-04≈	9.16E-01±3.26E-04≈	9.16E-01±5.78E-03≈	9.16E-01±9.50E-05≈	9.16E-01±2.82E-04
WFG3D3	2.87E-01±2.55E-04≈	2.87E-01±3.57E-04≈	2.87E-01±5.01E-04≈	2.87E-01±4.10E-04≈	2.87E-01±1.13E-04≈	2.87E-01±1.14E-04
WFG4D3	4.15E-01±1.18E-03−	4.13E-01±1.58E-03−	4.14E-01±1.40E-03−	4.15E-01±1.21E-03−	4.17E-01±1.06E-03≈	4.17E-01±1.04E-03
WFG5D3	3.81E-01±8.19E-03≈	3.79E-01±8.72E-03−	3.81E-01±1.01E-02≈	3.78E-01±6.86E-03−	3.81E-01±4.19E-03≈	3.81E-01±6.81E-03
WFG6D3	4.13E-01±2.04E-03−	4.14E-01±1.86E-03−	4.16E-01±1.81E-03≈	4.16E-01±1.47E-03≈	4.16E-01±1.39E-03≈	4.17E-01±2.29E-03
WFG7D3	4.04E-01±2.12E-03−	4.05E-01±1.66E-03≈	4.06E-01±1.34E-03≈	4.08E-01±8.59E-04+	4.05E-01±1.45E-03≈	4.06E-01±2.08E-03
WFG8D3	2.93E-01±1.04E-01−	2.91E-01±8.98E-02−	3.90E-01±1.00E-01≈	2.97E-01±1.05E-01≈	3.91E-01±1.03E-01≈	3.91E-01±7.33E-02
WFG9D3	4.03E-01±2.96E-03+	4.00E-01±3.23E-03≈	4.03E-01±3.39E-03+	4.03E-01±3.37E-03+	3.98E-01±2.98E-03≈	4.00E-01±2.94E-03
LZ09F1	6.61E-01±5.17E-05+	6.61E-01±3.57E-05−	6.61E-01±1.01E-04−	6.61E-01±3.05E-05≈	6.61E-01±7.20E-05≈	6.61E-01±6.62E-05
LZ09F3	6.47E-01±3.03E-03−	6.53E-01±2.82E-03≈	6.46E-01±3.75E-03−	6.55E-01±4.72E-03+	6.54E-01±3.18E-03≈	6.53E-01±2.55E-03
LZ09F4	6.50E-01±6.44E-03−	6.56E-01±2.35E-03+	6.44E-01±5.66E-03−	6.57E-01±2.64E-03+	6.54E-01±5.23E-03≈	6.53E-01±5.27E-03
LZ09F7	6.59E-01±1.21E-03≈	3.30E-01±6.60E-01≈	6.59E-01±7.64E-03≈	6.61E-01±8.55E-04+	6.59E-01±1.97E-03≈	6.59E-01±2.37E-03
LZ09F9	2.61E-01±1.14E-02−	3.07E-01±4.40E-03−	2.87E-01±3.50E-03−	3.13E-01±1.08E-01−	3.16E-01±3.25E-03≈	3.18E-01±4.89E-03
UF1	6.12E-01±2.11E-02−	6.50E-01±2.41E-03−	6.37E-01±8.63E-03−	6.55E-01±3.81E-03−	6.56E-01±1.75E-03≈	6.56E-01±2.43E-03
UF2	6.50E-01±2.98E-03−	6.52E-01±2.22E-03−	6.51E-01±3.13E-03−	6.53E-01±2.09E-03−	6.52E-01±4.02E-03−	6.54E-01±2.15E-03
UF3	3.73E-01±3.84E-02−	6.41E-01±2.52E-02−	5.24E-01±1.10E-01−	6.36E-01±2.75E-02−	6.48E-01±2.07E-02≈	6.50E-01±8.98E-03
UF4	2.78E-01±9.90E-04≈	2.78E-01±9.93E-04+	2.81E-01±8.36E-04+	2.77E-01±6.67E-04−	2.78E-01±6.44E-04+	2.77E-01±1.51E-03
UF5	3.75E-02±1.30E-01−	2.91E-01±5.77E-02+	1.82E-01±7.61E-02−	3.88E-03±9.32E-02−	1.71E-01±1.88E-01−	2.53E-01±3.76E-02
UF6	2.26E-01±1.35E-01≈	2.31E-01±1.13E-02−	2.38E-01±4.13E-02≈	2.10E-01±2.35E-01≈	2.49E-01±3.59E-02≈	2.53E-01±5.36E-02
UF7	4.76E-01±8.66E-03−	4.83E-01±1.66E-03−	4.75E-01±5.77E-03−	4.86E-01±3.81E-03−	4.87E-01±2.17E-03−	4.88E-01±2.45E-03
UF8	2.27E-01±5.57E-02−	1.82E-01±2.66E-02−	1.90E-01±4.71E-02−	1.92E-01±7.40E-02−	1.89E-01±5.97E-02−	3.02E-01±3.12E-03
UF9	6.01E-01±1.37E-01≈	6.09E-01±1.04E-01≈	5.93E-01±1.36E-01≈	5.57E-01±1.14E-01≈	5.54E-01±9.90E-02≈	5.49E-01±1.19E-01
UF10	2.26E-02±1.14E-01−	5.34E-03±1.11E-01−	1.25E-02±1.52E-01−	2.76E-02±5.62E-02−	1.31E-02±5.32E-02−	1.32E-01±7.59E-02
UF11	6.24E-06±4.16E-05−	1.39E-06±1.16E-05−	1.97E-05±4.50E-05−	5.85E-06±2.96E-05−	2.56E-05±4.32E-05−	3.08E-04±2.86E-04
UF12	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
UF13	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
CF1	4.39E-01±2.00E-02+	4.39E-01±2.00E-02≈	4.49E-01±2.00E-02+	4.39E-01±2.00E-02+	4.39E-01±2.00E-02+	4.39E-01±2.00E-02
CF2	6.48E-01±5.53E-03−	6.50E-01±9.16E-04−	6.50E-01±2.78E-03−	6.50E-01±1.36E-03−	6.50E-01±9.08E-04−	6.51E-01±1.17E-03
CF3	1.26E-01±6.99E-02≈	1.53E-01±8.58E-02≈	1.18E-01±6.50E-02≈	1.43E-01±8.13E-02≈	1.50E-01±6.51E-02≈	1.46E-01±6.82E-02
CF4	4.80E-01±4.18E-02−	5.35E-01±5.02E-03−	5.17E-01±9.68E-03−	5.41E-01±1.07E-02−	5.44E-01±4.64E-03−	5.49E-01±4.00E-03
CF5	1.48E-01±2.24E-01−	3.93E-01±2.46E-01≈	3.08E-01±1.83E-01≈	2.95E-01±1.40E-01−	3.19E-01±2.05E-01≈	3.56E-01±1.72E-01
CF6	6.52E-01±1.45E-02−	6.56E-01±3.57E-03−	6.54E-01±4.53E-03−	6.56E-01±3.46E-03−	6.57E-01±1.97E-03≈	6.58E-01±1.96E-03
CF7	4.95E-01±1.60E-01≈	5.96E-01±1.84E-02+	4.74E-01±9.90E-02−	4.39E-01±2.11E-01−	5.77E-01±1.86E-01≈	5.56E-01±4.47E-02
CF8	1.79E-01±4.32E-02−	1.82E-01±4.68E-02−	1.91E-01±4.60E-02−	1.94E-01±4.83E-02≈	1.90E-01±4.84E-02≈	2.04E-01±5.07E-02
CF9	2.46E-01±1.52E-02−	2.42E-01±1.87E-02−	2.49E-01±1.32E-02−	2.59E-01±1.99E-02−	2.49E-01±1.84E-02−	3.25E-01±2.91E-02
CF10	9.57E-02±1.10E-01−	1.67E-01±1.68E-02−	1.67E-01±1.04E-02−	1.69E-01±7.48E-02−	1.68E-01±9.82E-03−	2.07E-01±6.46E-02
Mean	3.73E-01±3.27E-02	3.78E-01±3.03E-02	3.82E-01±3.36E-02	3.62E-01±2.82E-02	3.84E-01±4.56E-02	4.03E-01±2.80E-02

+ , ≈ and − represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.

Table 3.5: Statistical results of adaptive MOEAs with decomposition-based approaches.

	MOEA/D-jDE	MOEA/D-SaDE	MOEA/D-CoDE	MOEA/D-T3	MOEA/D-T
MOEA/D-JADE	24 22 12	19 31 8	22 27 9	25 27 6	33 21 4
MOEA/D-jDE		16 29 13	18 32 8	26 24 8	31 19 8
MOEA/D-SaDE			15 29 14	20 28 10	31 21 6
MOEA/D-CoDE				19 30 9	32 17 9
MOEA/D-T3					14 39 5

MOEA/D-SaDE, MOEA/D-CoDE are 20/28/10 and 19/30/9, respectively. MOEA/D-T3 not only has fewer predefined parameters than MOEA/D-SaDE and MOEA/D-CoDE, but also is more effective than them. In addition, CoDE fixes the two parameters (CR and F) before the algorithm starts. MOEA/D-T3 adapts the control parameters as the algorithm progresses. It demonstrates that parameter adaptation based on trust in the framework is able to automatically tune parameters on various MOPs.

Furthermore, the *w/t/l* values between MOEA/D-T and MOEA/D-JADE, MOEA/D-jDE, MOEA/D-SaDE, MOEA/D-CoDE, MOEA/D-T3 are 33/21/4, 31/19/8, 31/21/6, 32/17/9 and 14/39/5, respectively. MOEA/D-T emerges as the best among all the six algorithms. The selection on the two extra evolutionary operators "SBX" and "polynomial mutation" does display benefits on the adaptive MOEAs.

3.4.3 DE Operators with Parameter Adaptation by Trust

The proposed framework can be treated as a simplified version (*i.e.*, parameter adaptation by trust) when evolutionary operators are selected before the algorithms start. In this section, to investigate the effectiveness of the parameter adaptation by trust, three DE operators are adopted in MOEA/D for solving various MOPs.

In the classical MOEA/D [11], it involves two operators "DE/rand/1/bin" and "polynomial mutation". To avoid the impact of other operators, "polynomial mutation" is not used in this section. Three versions of MOEA/D are implemented named as MOEA/D1, MOEA/D2 and MOEA/D3, where "1/2/3" means MOEA/D only adopts "DE/rand/1/bin", "DE/rand/2/bin" and "DE/current-to-rand/1/bin", respectively. The DE parameters in MOEA/D1–3 are $CR = 1.0, F = 0.4$. Then, they are extended by the proposed framework, named as MOEA/D1-T, MOEA/D2-T and MOEA/D3-T, respectively. Table 3.6 shows the detailed results of MOEA/D and MOEA/D-T with the three different DE operators. Table 3.7 summarizes the win/tie/lose (w/t/l) statistical comparison results.

In Table 3.7, the *w/t/l* values between MOEA/D3 and MOEA/D1, MOEA/D2 are 41/14/3, 30/19/9, respectively. By observing the definition of Eq. 3.14, "DE/current-to-rand/1/bin" is regarded as a degenerated case of "DE/current/1/bin" when $rand() = 0$

Table 3.6: Hypervolume median and IQR by MOEA/D based on DE operators with/without parameter adaptation over 30 independent runs on 58 MOPs with 300,000 FES. "1/2/3" means "DE/rand/1/bin", "DE/rand/2/bin" and "DE/current-to-rand/1/bin", respectively.

MOPs	MOEA/D1	MOEA/D2	MOEA/D3	MOEA/D3-T	MOEA/D1-T	MOEA/D2-T
ZDT1	4.46E-01±3.26E-01−	6.59E-01±1.65E-03−	6.60E-01±1.39E-03−	6.62E-01±1.30E-05−	6.62E-01±4.29E-06≈	6.62E-01±2.61E-07
ZDT2	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	3.28E-01±1.53E-04−	3.28E-01±5.62E-06−	3.28E-01±6.84E-08−	3.28E-01±1.35E-09
ZDT3	5.13E-01±1.23E-01−	5.14E-01±8.06E-05−	5.14E-01±1.03E-04−	5.14E-01±1.22E-05−	5.14E-01±5.76E-06≈	5.14E-01±4.54E-06
ZDT4	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±2.59E-03−	0.00E-00±0.00E-00−	3.82E-02±3.08E-01≈	1.59E-01±3.29E-01
ZDT6	4.01E-01±1.95E-07−	4.01E-01±1.12E-07−	4.01E-01±6.16E-05−	4.01E-01±3.93E-06−	4.01E-01±8.01E-10+	4.01E-01±5.94E-09
DTLZ1	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±1.04E-01−	7.98E-01±6.94E-01≈	7.98E-01±5.21E-01
DTLZ2	3.92E-01±8.01E-04−	3.92E-01±9.75E-04−	3.93E-01±3.95E-04−	4.29E-01±6.98E-06≈	4.29E-01±2.52E-06≈	4.29E-01±2.09E-06
DTLZ3	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
DTLZ4	3.91E-01±3.39E-03−	3.97E-01±2.53E-03−	3.92E-01±2.94E-03−	4.22E-01±5.68E-06≈	4.22E-01±2.22E-06≈	4.22E-01±1.88E-06
DTLZ5	9.15E-02±4.27E-06+	9.15E-02±1.33E-05+	9.15E-02±3.22E-06+	8.04E-02±4.32E-07+	8.04E-02±3.84E-07≈	8.04E-02±5.35E-08
DTLZ6	9.23E-02±1.22E-06+	9.24E-02±1.55E-06+	9.24E-02±1.21E-06+	8.12E-02±2.04E-07≈	8.12E-02±1.66E-07≈	8.12E-02±5.24E-08
DTLZ7	2.17E-01±4.64E-03−	2.15E-01±3.24E-03−	2.15E-01±8.06E-04−	2.67E-01±1.30E-03−	2.67E-01±1.17E-03≈	2.67E-01±6.87E-04
WFG1	6.25E-01±2.36E-02−	6.24E-01±5.53E-03−	6.30E-01±3.58E-03−	6.33E-01±1.28E-04+	6.33E-01±3.44E-05+	6.31E-01±2.17E-03
WFG2	5.62E-01±2.89E-04−	5.63E-01±9.23E-06−	5.63E-01±1.68E-05−	5.63E-01±3.23E-07≈	5.63E-01±1.92E-07≈	5.63E-01±2.55E-07
WFG3	4.42E-01±1.21E-05−	4.42E-01±7.09E-06−	4.42E-01±9.81E-06−	4.42E-01±3.96E-07+	4.42E-01±2.95E-09≈	4.42E-01±1.78E-09
WFG4	2.14E-01±1.52E-03−	2.10E-01±2.79E-03−	2.16E-01±9.08E-04−	2.18E-01±1.70E-04≈	2.18E-01±1.71E-04≈	2.18E-01±1.69E-04
WFG5	1.95E-01±4.24E-05≈	1.95E-01±1.75E-05−	1.95E-01±6.56E-05≈	1.95E-01±8.17E-03+	1.95E-01±4.10E-05≈	1.95E-01±4.39E-05
WFG6	2.09E-01±1.04E-04−	2.09E-01±1.05E-05−	2.09E-01±1.03E-05−	2.10E-01±2.97E-06−	2.10E-01±9.46E-08≈	2.10E-01±1.34E-07
WFG7	2.09E-01±2.88E-05−	2.09E-01±8.94E-06−	2.09E-01±5.34E-06−	2.10E-01±1.80E-06−	2.10E-01±4.99E-07≈	2.10E-01±5.82E-07
WFG8	1.51E-01±1.30E-03−	1.54E-01±1.21E-03≈	1.53E-01±5.67E-04−	1.53E-01±1.14E-03≈	1.54E-01±5.31E-02≈	1.54E-01±5.32E-02
WFG9	2.40E-01±6.98E-04≈	2.38E-01±4.85E-04−	2.39E-01±7.82E-04≈	2.39E-01±9.47E-04≈	2.39E-01±7.94E-04≈	2.39E-01±9.91E-04
WFG1D3	8.81E-01±1.11E-02−	2.97E-01±9.97E-02−	8.88E-01±7.12E-03−	8.96E-01±3.78E-03−	9.04E-01±3.37E-03+	9.01E-01±4.07E-03
WFG2D3	8.89E-01±3.94E-03−	8.91E-01±3.65E-03−	8.95E-01±3.89E-03−	9.16E-01±2.21E-04≈	9.16E-01±2.79E-04≈	9.16E-01±3.77E-04
WFG3D3	3.02E-01±1.27E-04+	3.02E-01±1.85E-04+	3.02E-01±1.08E-04+	2.87E-01±3.36E-04≈	2.87E-01±1.28E-04≈	2.87E-01±2.30E-04
WFG4D3	3.63E-01±6.78E-03−	3.66E-01±7.23E-03−	3.71E-01±5.41E-03−	4.15E-01±6.71E-04−	4.17E-01±1.03E-03+	4.16E-01±1.69E-03
WFG5D3	3.31E-01±3.46E-03−	3.35E-01±3.45E-03−	3.33E-01±2.46E-03−	3.80E-01±4.74E-03≈	3.81E-01±5.71E-04+	3.81E-01±9.70E-04
WFG6D3	3.77E-01±1.31E-03−	3.77E-01±7.92E-04−	3.78E-01±1.04E-03−	4.15E-01±1.08E-03−	4.16E-01±1.71E-03≈	4.16E-01±2.62E-03
WFG7D3	3.70E-01±1.75E-03−	3.69E-01±2.30E-03−	3.69E-01±7.97E-04−	4.07E-01±1.39E-03+	4.06E-01±2.11E-03≈	4.05E-01±1.83E-03
WFG8D3	2.73E-01±4.04E-03−	2.74E-01±3.60E-03−	2.68E-01±4.88E-03−	2.93E-01±1.04E-01≈	2.94E-01±1.03E-01≈	3.90E-01±1.00E-01
WFG9D3	3.65E-01±4.57E-03−	3.65E-01±2.97E-03−	3.64E-01±5.23E-03−	4.04E-01±2.87E-03+	3.99E-01±4.33E-03≈	3.99E-01±3.49E-03
LZ09F1	4.76E-01±4.84E-02−	6.46E-01±1.42E-02−	6.29E-01±4.77E-02−	6.61E-01±1.05E-04−	6.61E-01±4.62E-05−	6.61E-01±4.88E-05
LZ09F3	3.60E-03±1.90E-02−	1.95E-01±1.44E-01−	6.23E-01±1.02E-01−	6.50E-01±4.21E-03−	6.54E-01±2.93E-03−	6.56E-01±1.93E-03
LZ09F4	3.30E-02±5.90E-02−	2.14E-01±1.05E-01−	6.09E-01±1.95E-01−	6.51E-01±9.70E-03−	6.53E-01±4.42E-03−	6.56E-01±5.31E-03
LZ09F7	6.49E-02±1.58E-01−	4.98E-02±6.35E-01−	6.55E-01±2.01E-02−	6.57E-01±7.45E-03−	6.60E-01±1.50E-03≈	6.60E-01±1.34E-03
LZ09F9	6.43E-03±2.47E-02−	6.55E-02±4.32E-02−	2.48E-01±1.37E-02−	2.98E-01±7.04E-03−	3.17E-01±3.69E-03≈	3.18E-01±3.07E-03
UF1	1.61E-01±1.06E-01−	5.12E-01±5.26E-02−	5.93E-01±3.79E-02−	6.47E-01±1.24E-02−	6.54E-01±3.78E-03−	6.56E-01±2.18E-03
UF2	5.26E-01±2.65E-02−	5.81E-01±1.21E-02−	6.19E-01±2.92E-02−	6.51E-01±4.44E-03−	6.52E-01±3.98E-03≈	6.52E-01±1.97E-03
UF3	9.99E-02±3.50E-02−	2.70E-01±8.73E-02−	3.68E-01±1.12E-01−	5.14E-01±6.89E-02−	6.43E-01±2.20E-02≈	6.48E-01±1.73E-02
UF4	2.19E-01±1.40E-02−	2.34E-01±1.24E-02−	2.29E-01±1.63E-02−	2.76E-01±1.81E-03−	2.78E-01±7.49E-04≈	2.78E-01±7.77E-04
UF5	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	1.10E-03±2.35E-02−	2.04E-01±1.50E-01−	2.76E-01±8.00E-02
UF6	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	1.03E-01±1.66E-01−	2.26E-01±1.13E-01−	2.32E-01±4.35E-02≈	2.45E-01±3.20E-02
UF7	5.96E-02±6.33E-02−	3.61E-01±9.24E-02−	4.60E-01±1.88E-02−	4.83E-01±4.85E-03−	4.86E-01±3.75E-03≈	4.86E-01±2.15E-03
UF8	1.47E-01±4.91E-02−	2.30E-01±4.72E-02≈	2.82E-01±8.54E-03+	1.90E-01±5.82E-03−	2.35E-01±6.64E-02≈	2.37E-01±1.98E-02
UF9	2.37E-01±7.16E-02−	4.84E-01±6.54E-02−	5.38E-01±9.98E-02≈	5.35E-01±1.35E-01≈	5.97E-01±1.18E-01≈	5.51E-01±1.08E-01
UF10	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±4.61E-03−	2.51E-02±3.75E-02≈	3.43E-02±1.05E-01≈	3.39E-02±7.69E-02
UF11	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	2.18E-04±1.57E-04+	4.64E-06±1.57E-05≈	1.33E-05±3.51E-05≈	1.32E-05±6.81E-05
UF12	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
UF13	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
CF1	4.39E-01±6.51E-04−	4.39E-01±2.00E-02+	4.39E-01±2.43E-05−	4.59E-01±2.00E-02≈	4.39E-01±2.00E-02−	4.39E-01±2.00E-02
CF2	6.11E-01±8.21E-02−	6.50E-01±2.04E-03−	6.45E-01±1.20E-02−	6.50E-01±1.40E-03≈	6.50E-01±7.37E-04≈	6.50E-01±8.20E-04
CF3	0.00E-00±0.00E-00−	0.00E-00±1.86E-02−	2.81E-02±8.28E-02−	1.09E-01±7.04E-02≈	1.44E-01±7.51E-02≈	1.30E-01±5.73E-02
CF4	2.64E-01±5.99E-02−	4.27E-01±7.33E-02−	4.46E-01±9.70E-02−	5.35xE-01±6.72E-03−	5.43E-01±4.59E-03≈	5.45E-01±4.95E-03
CF5	0.00E-00±0.00E-00−	9.96E-02±1.81E-01−	2.10E-01±2.40E-01−	3.31E-01±1.10E-01−	3.16E-01±2.67E-01≈	4.07E-01±2.72E-01
CF6	5.64E-01±4.25E-02−	6.54E-01±3.03E-03−	6.16E-01±2.10E-02−	6.51E-01±7.69E-03−	6.57E-01±2.79E-03≈	6.57E-01±2.23E-03
CF7	0.00E-00±0.00E-00−	3.50E-02±1.32E-01−	3.14E-02±1.45E-01−	4.13E-01±1.76E-01≈	5.59E-01±2.73E-01≈	5.62E-01±2.62E-01
CF8	2.10E-01±4.81E-02+	2.27E-01±4.84E-02+	2.30E-01±3.89E-02+	2.10E-01±4.22E-02≈	1.95E-01±4.95E-02≈	1.95E-01±3.45E-02
CF9	2.97E-01±2.29E-02+	3.00E-01±3.73E-02+	3.09E-01±3.18E-02+	2.63E-01±1.54E-02+	2.51E-01±1.47E-02≈	2.42E-01±1.71E-02
CF10	3.82E-02±6.18E-02−	1.66E-01±8.80E-02≈	1.66E-01±7.52E-02≈	1.54E-01±9.11E-02≈	1.65E-01±2.97E-02≈	1.62E-01±2.73E-02
Mean	2.41E-01±2.61E-02	2.76E-01±3.55E-02	3.31E-01±2.86E-02	3.57E-01±2.09E-02	3.82E-01±4.21E-02	3.88E-01±3.57E-02

+ , ≈ and − represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.

and "DE/rand/1/bin" when $rand() = 1$. The high perturbation of "DE/current-to-rand/1/bin" is beneficial for MOEA/D3 to maintain population diversity. The $w/t/l$ values between MOEA/D3-T and MOEA/D3 are 40/11/7, between MOEA/D1-T and MOEA/D1 are 48/5/5, and between MOEA/D2-T and MOEA/D2 are 46/6/6. The three DE operators with the proposed framework (parameter adaptation based on trust)

Table 3.7: Statistical comparison results of MOEA/D based on DE operators with/without parameter adaptation.

	MOEA/D2	MOEA/D3	MOEA/D3-T	MOEA/D1-T	MOEA/D2-T
MOEA/D1	32 20 6	41 14 3	46 7 5	48 5 5	48 5 5
MOEA/D2		30 19 9	41 9 8	46 6 6	46 6 6
MOEA/D3			40 11 7	45 6 7	44 7 7
MOEA/D3-T				27 26 5	28 23 7
MOEA/D1-T					7 46 5

obtain higher performance over the original DE operators on 66.7% of the 58 MOPs. It is evident that parameter adaptation by trust is able to enhance the *robustness* of DE operators significantly.

Furthermore, the statistical results in Table 3.7 show that the *w/t/l* values between MOEA/D1-T, MOEA/D2-T and MOEA/D3-T are 27/26/5, 28/23/7, respectively. It is worth noticing that MOEA/D3 emerges as superior among the three original DE operators, whereas MOEA/D3-T is the worst among those operators with parameter adaptation by trust. This indicates that the uncertain perturbation ($rand()$) in “DE/current-to-rand/1/bin” is less efficient than the framework of adapting control parameters based on the concept of trust.

3.4.4 Comparison of the MOEA/D Variants

In this section, two additional MOEA/D variants (called MOEA/D-Rand, MOEA/D-T’) are implemented to deal with MOPs. In contrast to the adaptive algorithms, MOEA/D-Rand just randomly selects evolutionary operators and adapts control parameters. The algorithm MOEA/D-T’ is implemented to use another version of the framework with a fixed parameter $\eta = 0.3$ (see Eq. 3.7). In MOEA/D-Rand and MOEA/D-T’, the operator pool consists of the five evolutionary operators as mentioned in the beginning of Section 3.4, and the control parameters are quantized ($q = 3$) into segments that obey the Normal distribution which is the same as MOEA/D-T. The purpose is to compare the effectiveness of the proposed framework with the random selection and the update scheme with the fixed η value.

Table 3.8 shows the detail about hypervolume results of MOEA/D-Rand, MOEA/D-T’ and MOEA/D-T over 30 independent runs on 58 MOPs with 300,000 function evo-

Table 3.8: Hypervolume median and IQR by MOEA/D variants over 30 independent runs on 58 MOPs with 300,000 FES.

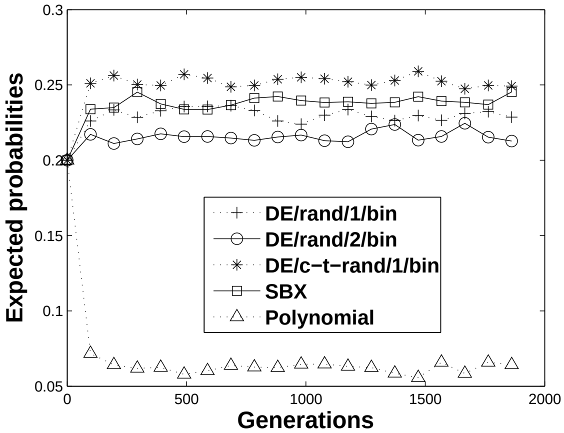
MOPs	MOEA/D-Rand	MOEA/D-T'	MOEA/D-T
ZDT1	6.62E-01±1.23E-05 −	6.62E-01±7.45E-08 +	6.62E-01±4.95E-07
ZDT2	3.28E-01±7.85E-06 −	3.28E-01±9.41E-10 +	3.28E-01±1.18E-08
ZDT3	5.14E-01±3.07E-05 −	5.14E-01±7.23E-07 ≈	5.14E-01±2.17E-06
ZDT4	6.62E-01±2.02E-05 −	3.82E-02±3.41E-01 −	6.62E-01±1.05E-04
ZDT6	4.01E-01±1.23E-07 −	4.01E-01±1.65E-09 ≈	4.01E-01±4.72E-09
DTLZ1	7.98E-01±1.65E-07 +	1.04E-01±7.72E-01 −	4.51E-01±6.94E-01
DTLZ2	4.29E-01±6.12E-06 ≈	4.29E-01±1.52E-06 ≈	4.29E-01±1.78E-06
DTLZ3	4.29E-01±1.91E-04 +	0.00E-00±0.00E-00 −	3.93E-01±3.15E-04
DTLZ4	4.22E-01±1.29E-05 ≈	4.22E-01±5.78E-07 ≈	4.22E-01±1.52E-06
DTLZ5	8.04E-02±5.50E-07 ≈	8.04E-02±2.04E-08 ≈	8.04E-02±1.59E-07
DTLZ6	8.12E-02±3.56E-07 ≈	8.12E-02±1.13E-08 ≈	8.12E-02±1.20E-07
DTLZ7	2.67E-01±5.06E-04 ≈	2.67E-01±1.16E-03 ≈	2.67E-01±9.47E-04
WFG1	6.30E-01±3.31E-03 −	6.33E-01±9.45E-05 ≈	6.33E-01±6.43E-05
WFG2	5.63E-01±4.63E-07 +	5.63E-01±1.91E-07 ≈	5.63E-01±2.41E-07
WFG3	4.42E-01±2.01E-06 +	4.42E-01±2.50E-09 ≈	4.42E-01±2.41E-09
WFG4	2.18E-01±9.77E-07 −	2.18E-01±1.70E-04 −	2.18E-01±1.39E-06
WFG5	1.95E-01±6.48E-05 ≈	1.95E-01±4.24E-05 ≈	1.95E-01±6.56E-05
WFG6	2.10E-01±4.13E-07 −	2.10E-01±2.12E-05 −	2.10E-01±1.69E-07
WFG7	2.10E-01±1.30E-06 −	2.10E-01±5.92E-07 +	2.10E-01±1.11E-06
WFG8	1.80E-01±5.34E-02 ≈	1.53E-01±5.28E-02 −	2.07E-01±5.26E-02
WFG9	2.39E-01±1.02E-03 ≈	2.39E-01±9.43E-04 −	2.39E-01±1.03E-03
WFG1D3	9.04E-01±9.88E-03 ≈	9.05E-01±5.51E-03 ≈	9.07E-01±5.89E-03
WFG2D3	9.16E-01±2.49E-04 ≈	9.05E-01±1.13E-02 −	9.16E-01±2.82E-04
WFG3D3	2.87E-01±3.85E-04 ≈	2.87E-01±1.25E-04 ≈	2.87E-01±1.14E-04
WFG4D3	4.11E-01±9.05E-04 −	4.17E-01±1.29E-03 +	4.17E-01±1.04E-03
WFG5D3	3.79E-01±1.73E-03 −	3.81E-01±2.17E-03 ≈	3.81E-01±6.81E-03
WFG6D3	4.15E-01±1.83E-03 −	4.14E-01±2.49E-03 −	4.17E-01±2.29E-03
WFG7D3	4.05E-01±1.41E-03 ≈	4.04E-01±2.36E-03 −	4.06E-01±2.08E-03
WFG8D3	3.90E-01±1.02E-01 ≈	2.93E-01±9.85E-02 −	3.91E-01±7.33E-02
WFG9D3	4.00E-01±2.93E-03 ≈	3.96E-01±2.02E-03 −	4.00E-01±2.94E-03
LZ09F1	6.61E-01±3.86E-05 −	6.61E-01±2.80E-04 −	6.61E-01±6.62E-05
LZ09F3	6.46E-01±6.00E-03 −	6.51E-01±2.25E-03 −	6.53E-01±2.55E-03
LZ09F4	6.50E-01±3.86E-03 −	6.52E-01±8.38E-03 ≈	6.53E-01±5.27E-03
LZ09F7	6.58E-01±1.86E-03 −	6.57E-01±3.98E-03 −	6.59E-01±2.37E-03
LZ09F9	2.93E-01±9.80E-03 −	3.17E-01±4.08E-03 ≈	3.18E-01±4.89E-03
UF1	6.44E-01±4.74E-03 −	6.56E-01±5.15E-03 ≈	6.56E-01±2.43E-03
UF2	6.50E-01±3.84E-03 −	6.51E-01±3.57E-03 −	6.54E-01±2.15E-03
UF3	5.64E-01±7.49E-02 −	6.50E-01±1.75E-02 ≈	6.50E-01±8.98E-03
UF4	2.75E-01±1.00E-03 −	2.78E-01±1.01E-03 ≈	2.77E-01±1.51E-03
UF5	2.20E-01±4.01E-02 −	2.14E-01±6.92E-02 −	2.53E-01±3.76E-02
UF6	2.41E-01±1.59E-01 ≈	2.33E-01±5.11E-02 ≈	2.53E-01±5.36E-02
UF7	4.82E-01±1.44E-03 −	4.85E-01±3.82E-03 −	4.88E-01±2.45E-03
UF8	1.85E-01±3.92E-02 −	1.89E-01±5.64E-02 −	3.02E-01±3.12E-03
UF9	5.26E-01±9.47E-02 −	5.52E-01±1.21E-01 ≈	5.49E-01±1.19E-01
UF10	3.70E-02±1.26E-01 −	3.99E-02±7.42E-02 −	1.32E-01±7.59E-02
UF11	0.00E-00±3.77E-07 −	3.95E-06±3.66E-05 −	3.08E-04±2.86E-04
UF12	0.00E-00±0.00E-00 ≈	0.00E-00±0.00E-00 ≈	0.00E-00±0.00E-00
UF13	0.00E-00±0.00E-00 ≈	0.00E-00±0.00E-00 ≈	0.00E-00±0.00E-00
CF1	4.39E-01±2.00E-02 −	4.39E-01±2.00E-02 ≈	4.39E-01±2.00E-02
CF2	6.50E-01±1.08E-03 −	6.50E-01±1.06E-03 −	6.51E-01±1.17E-03
CF3	1.20E-01±7.72E-02 ≈	1.29E-01±5.20E-02 ≈	1.46E-01±6.82E-02
CF4	5.30E-01±7.58E-03 −	5.36E-01±1.09E-02 −	5.49E-01±4.00E-03
CF5	3.06E-01±1.90E-01 ≈	3.32E-01±1.82E-01 ≈	3.56E-01±1.72E-01
CF6	6.56E-01±3.68E-03 −	6.54E-01±6.79E-03 −	6.58E-01±1.96E-03
CF7	5.37E-01±1.36E-01 −	4.89E-01±1.55E-01 ≈	5.56E-01±4.47E-02
CF8	2.10E-01±3.36E-02 ≈	1.94E-01±5.12E-02 −	2.04E-01±5.07E-02
CF9	2.57E-01±1.68E-02 −	2.50E-01±1.16E-02 −	3.25E-01±2.91E-02
CF10	1.67E-01±3.36E-02 −	1.44E-01±9.93E-02 −	2.07E-01±6.46E-02
Mean	3.98E-01±2.18E-02	3.67E-01±3.98E-02	4.03E-01±2.80E-02

+ , ≈ and − represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.

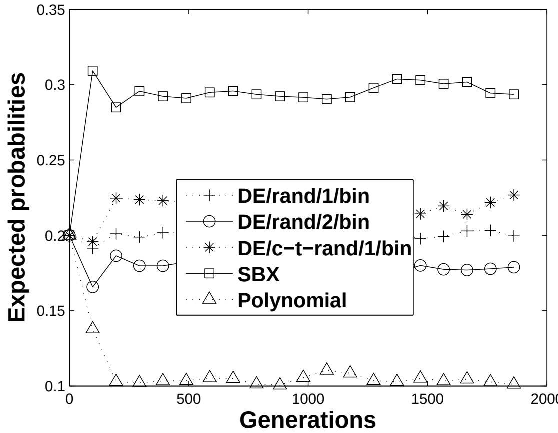
Table 3.9: Statistical comparison results of MOEA/D variants.

	MOEA/D-T'	MOEA/D-T
MOEA/D-Rand	20 26 12	34 20 4
MOEA/D-T'		27 27 4

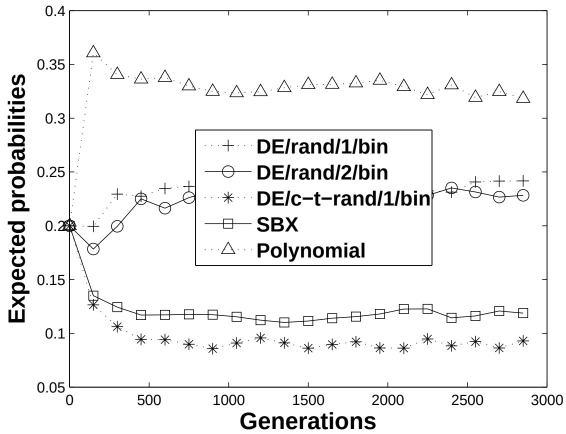
lutions. Table 3.9 summarizes the win/tie/lose (w/t/l) statistical comparison results of these three algorithms.



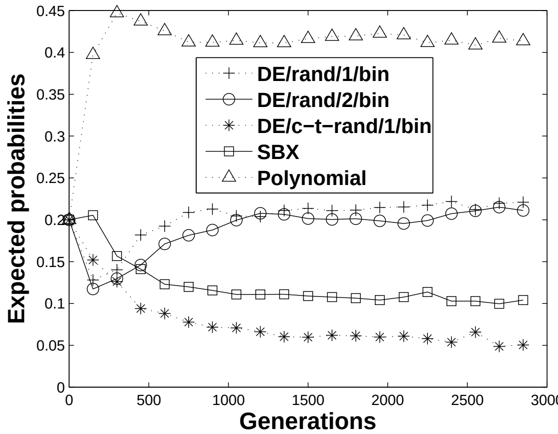
3.1.a: DTLZ1 by NSGAII-T



3.1.b: WFG1D3 by NSGAII-T



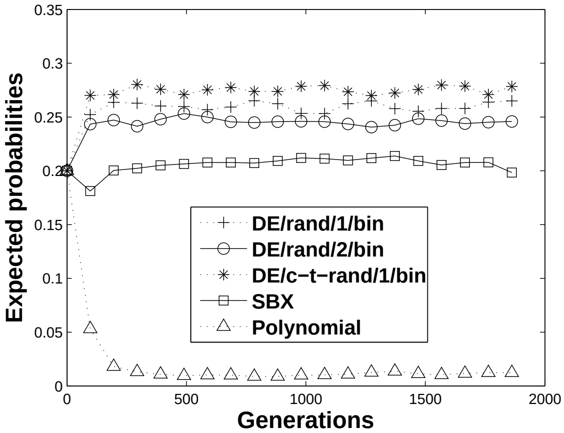
3.1.c: LZ09F1 by NSGAII-T



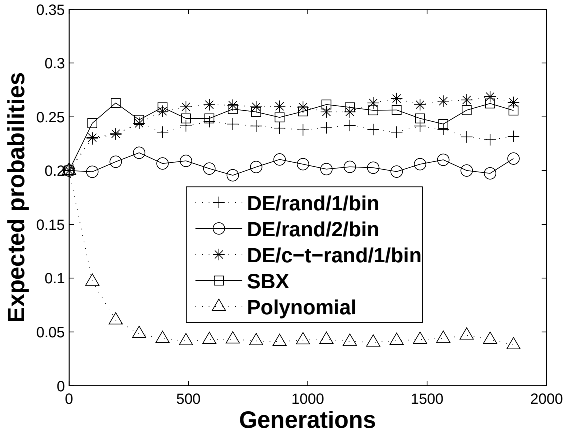
3.1.d: UF1 by NSGAII-T

Figure 3.1: The expected probabilities of operators $E_g(O)$ along generations derived by Pareto dominance-based MOEAs (NSGAII-T) on DTLZ1, WFG1D3, LZ09F1 and UF1. Operator pool is $O = \{$ “DE/rand/1/bin”, “DE/rand/2/bin”, “DE/current-to-rand/1/bin”, “SBX”, “polynomial mutation”$\}$.

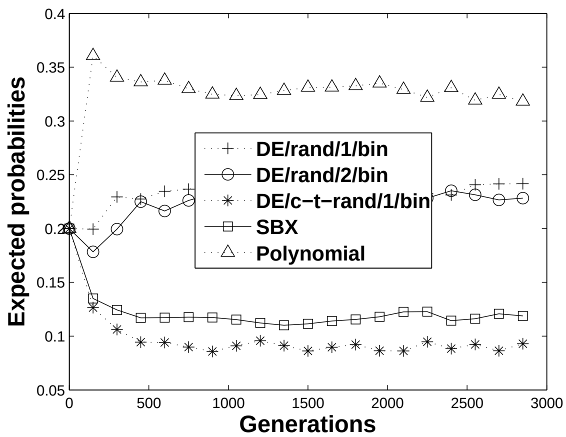
In Table 3.9, the $w/t/l$ values between MOEA/D-T and MOEA/D-Rand are 34/20/4. This indicates that the adaptive approach based on trust is better than the simple random selection. The $w/t/l$ values between MOEA/D-T and MOEA/D-T’ are 27/27/4. This demonstrates that the proposed framework is better than the update schema with the fixed η value. In addition, it is worth noticing that the framework is simpler and more efficient than MOEA/D-T' by replacing the predefined parameter η with trust scores of the previous generation $R_{g-1}(\cdot)$.



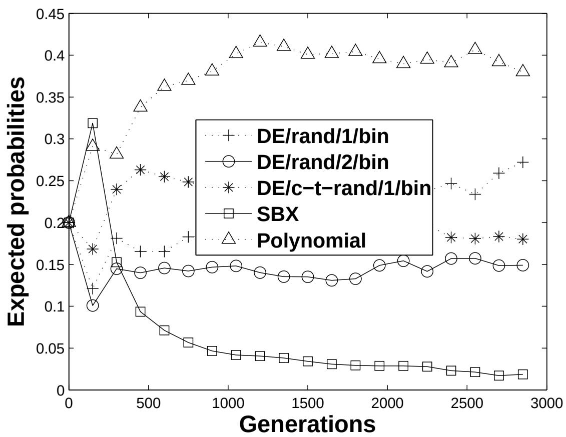
3.2.a: DTLZ1 by SPEA2-T



3.2.b: WFG1D3 by SPEA2-T



3.2.c: LZ09F1 by SPEA2-T



3.2.d: UF1 by SPEA2-T

Figure 3.2: The expected probabilities of operators $E_g(O)$ along generations derived by Pareto dominance-based MOEAs (SPEA2-T) on DTLZ1, WFG1D3, LZ09F1 and UF1. Operator pool is $O = \{$ “DE/rand/1/bin”, “DE/rand/2/bin”, “DE/current-to-rand/1/bin”, “SBX”, “polynomial mutation”$\}$.

3.4.5 Effect of Pareto Sets on Operator Selection

An important factor that affects the search ability of Pareto dominance-based MOEAs (*e.g.*, NSGAII [9], SPEA2 [10]) in *variable space* is the interdependency among the decision variables in Pareto Sets (PSs). In this experiment, the effect of typical PSs is investigated on different operators.

The algorithms NSGAII-T and SPEA2-T are tested on the four typical problems DTLZ1, WFG1D3, LZ09F1 and UF1. In the first two problems DTLZ1 and WFG1D3, decision variables in PSs are independent, but in LZ09F1 and UF1, the decision variables are interdependent [181]. Figs. 3.1-3.2 show the expected probabilities of operators $E_g(O)$ on the four typical MOPs at different generations derived by NSGAII-T and SPEA2-T, respectively. All probability values of the five operators are the mean values of 30 independent runs. From Figs. 3.1-3.2, it can be found that in Pareto dominance-based MOEAs, the evolutionary operators "SBX" and "polynomial mutation" show great contributions. It is also evident that "SBX" fares better on independent PSs (DTLZ1 and WFG1D3) than on interdependent PSs (LZ09F1 and UF1) [167], whereas, "polynomial mutation" is more suitable to deal with nonlinear variable dependencies than independent variables [168].

3.4.6 Trustworthiness of Evolutionary Operators

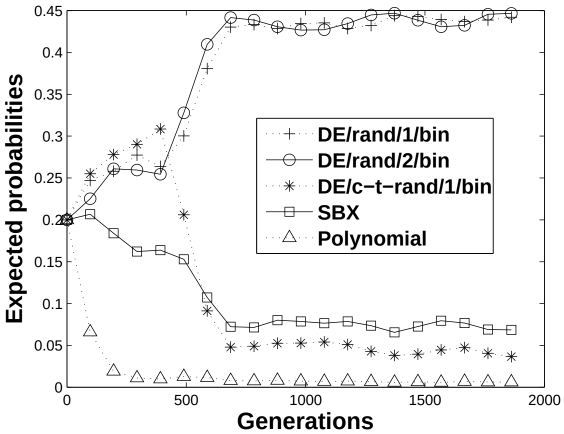
In this section, the effectiveness (trustworthiness) of evolutionary operators is investigated under decomposition-based MOEAs (*i.e.*, MOEA/D-T) on the four typical MOPs.

The operators "DE/rand/1/bin" and "DE/rand/2/bin" are quite similar. However, "DE/current-to-rand/1/bin" is different and it is formulated as follows:

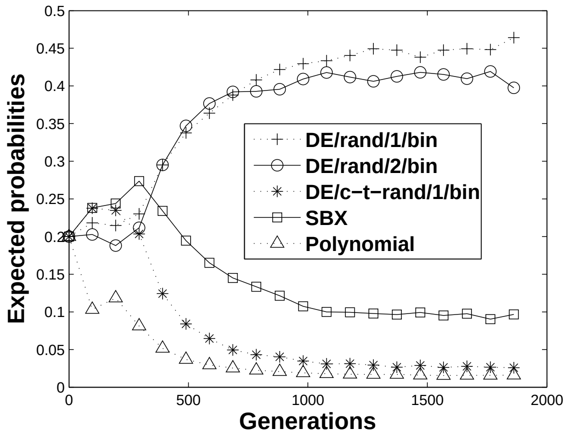
$$\vec{v}_{i,g} = \vec{x}_{i,g} + rand() \cdot (\vec{x}_{r1,g} - \vec{x}_{i,g}) + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g})$$

where $\vec{x}_i$ is the base vector, and $rand() \in [0, 1]$ is a uniform random value. The operator "DE/current-to-rand/1/bin" generates new offspring based on $\vec{x}_i$, whereas "DE/rand/1/bin" and "DE/rand/2/bin" search new offspring solutions in the global region.

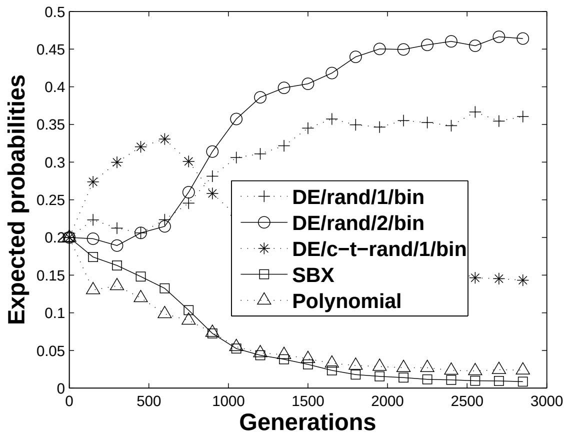
Fig. 3.3 shows the expected probabilities of the evolutionary operators $E_g(O)$ at different generations derived by MOEA/D-T on DTLZ1, WFG1D3, LZ09F1 and UF1. All



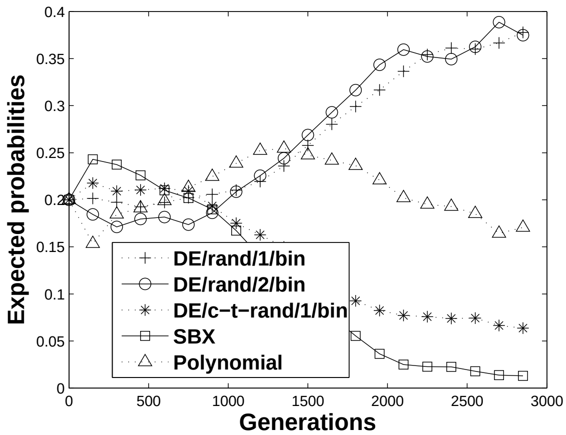
3.3.a: DTLZ1 by MOEA/D-T



3.3.b: WFG1D3 by MOEA/D-T



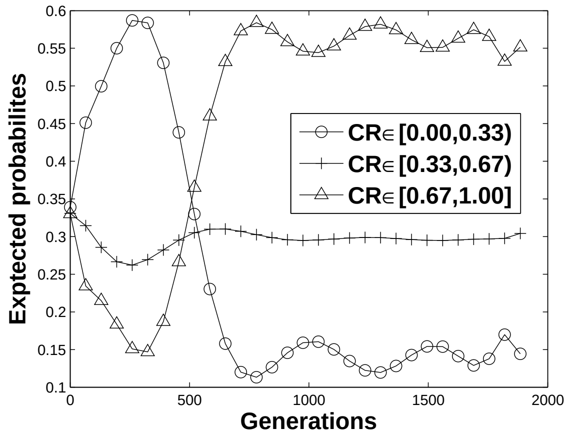
3.3.c: LZ09F1 by MOEA/D-T



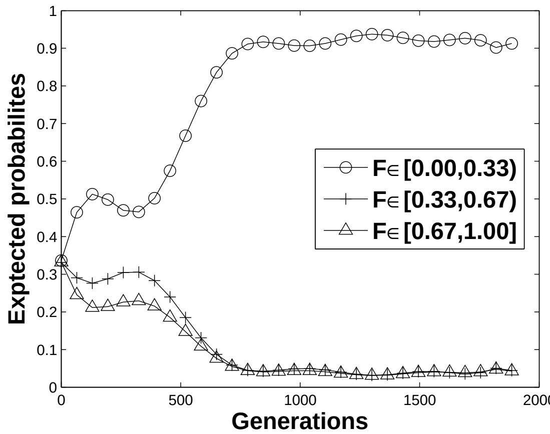
3.3.d: UF1 by MOEA/D-T

Figure 3.3: The expected probabilities of operators $E_g(O)$ along generations derived by decomposition-based MOEAs (MOEA/D-T) on DTLZ1, WFG1D3, LZ09F1 and UF1. Operator pool is $O = \{$ "DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1/bin", "SBX", "polynomial mutation"$\}$.

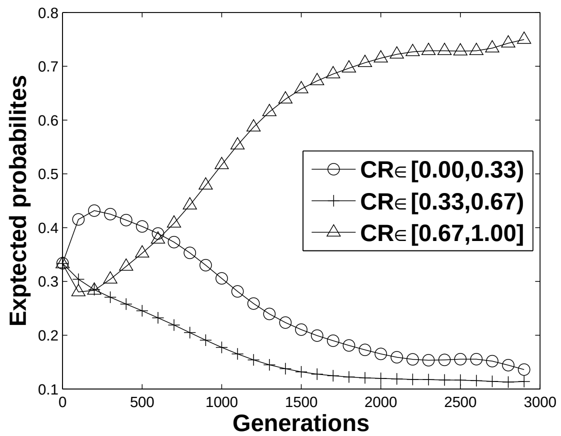
probability values of the five operators are the means of 30 independent runs. Under the decomposition-based MOEAs, "DE/rand/1/bin", "DE/rand/2/bin" and "DE/current-to-rand/1/bin" are more effective than "SBX" and "polynomial mutation". The probability of "DE/current-to-rand/1/bin" increases in the earlier stage, and then gradually decreases in the later stage, whereas the probabilities of "DE/rand/1/bin" and



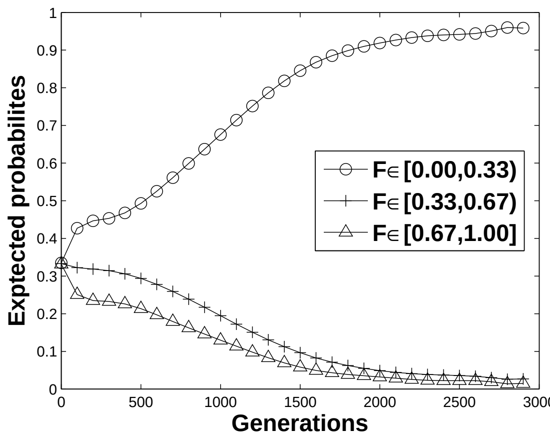
3.4.a: $E_g(CR|o)$ on DTLZ1



3.4.b: $E_g(F|o)$ on DTLZ1



3.4.c: $E_g(CR|o)$ on UF1



3.4.d: $E_g(F|o)$ on UF1

Figure 3.4: The expected probabilities of parameters $E_g(CR|o)$ and $E_g(F|o)$ along generations derived by MOEA/D-T on DTLZ1 and UF1. Two parameters CR and F are quantized into three segments, and operator is $o =$ "DE/rand/2/bin".

"DE/rand/2/bin" gradually increase as MOEA/D-T progresses. From the formulation of Eq. 3.14, "DE/current-to-rand/1/bin" has the search bias from the base vector ($\vec{x}_i$) and a large perturbation ($rand()$). In the earlier stage of MOEA/D-T, it has good performance due to the biased search. But its performance gradually deteriorates in the later stage because of the uncertain perturbation. "DE/rand/1/bin" and "DE/rand/2/bin" do not possess bias of any search directions but they have strong exploration. Their probabilities are low in the earlier stage because of their unbiased search. But in the later stage, they are more effective than "DE/current-to-rand/1/bin" due to their better exploration. Thus, the trustworthiness of the evolutionary operators modeled by the proposed framework well reflects their true effectiveness on the various MOPs.

3.4.7 Trustworthiness of Control Parameters

In this experiment, the different suitability (trustworthiness) of the control parameters is tested by MOEA/D-T on two typical MOPs. Fig. 3.4 shows the expected probabilities of the control parameters $E(CR|o)$ and $E(F|o)$ at different generations derived by MOEA/D-T on DTLZ1 and UF1, where $CR, F \in [0,1]$ are quantized into three segments. All reported probability values of the control parameters are the means of 30 independent runs. For simplicity, this study only shows the trustworthiness of control parameters for the operator $o =$ "DE/rand/2/bin".

In Fig. 3.4, the probability of $CR \in [0.67, 1.00]$ is high in the later stage on DTLZ1 and UF1. The larger CR induces the operator to search in a broad region, benefiting MOEA/D in maintaining the population diversity. The probability of $F \in [0.00, 0.33)$ gradually increases on DTLZ1 and UF1. As the algorithm progresses, the solutions spread more evenly. It means that the difference between solutions (*e.g.*, $\vec{x}_{r1,g} - \vec{x}_{r_2,g}$ in DE operators) becomes larger. So, in the later stage, the operator "DE/rand/2/bin" needs to adjust the parameter F to be small for exploitation in searching the neighboring region. Thus, the trustworthiness of the control parameters well reflects their varying suitability across generations.

3.4.8 Discussion of Parameter Quantization

As mentioned in Section 3.3.4, the proposed framework has only one predefined parameter (q) which defines the number of parameter quantization. To investigate the impact of q, the algorithm MOEA/D-T with $q \in [1, 10]$ is tested on 58 MOPs. Fig. 3.5 shows the median and IQR of hypervolume found by classical MOEAs and MOEA/D-T over 30 independent runs. It is evident that MOEA/D-T is not sensitive to the setting of the parameter q. The superior of MOEA/D-T over the classical MOEAs is due to operator selection and parameter adaptation by the concept of trust.

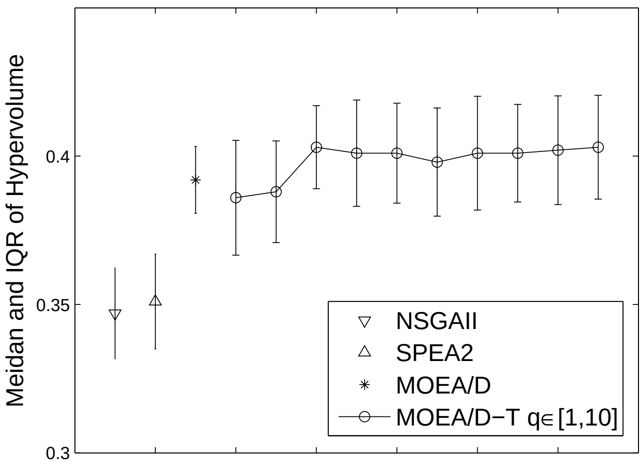


Figure 3.5: MOEA/D-T with different values of parameter quantization q.

3.5 Summary

In this chapter, a novel multiagent evolutionary framework based on trust is proposed to effectively select evolutionary operators and adjust control parameters (represented as services), for solving complex optimization problems (such as MOPs). In the framework, agents (representing solutions) automatically select services by modeling their trustworthiness based on the number of offsprings produced using them will survive to the next generation. Experiments carried out on 58 benchmark MOPs confirm that the proposed framework significantly improves the performance of the three classic MOEAs (NSGAI-I, SPEA2 and MOEA/D) and outperforms the other four adaptive approaches (JADE, jDE, SaDE, CoDE).

The contributions of the current work can be summarized as follows: 1) majority of the adaptive EC have been proposed to work with single objective optimization problems. This study is proposed as a generic framework that can also be adopted to solve MOPs and other complex optimization problems; 2) the few existing adaptive EC for MOPs adjust only control parameters, whereas agents in the proposed framework can also select evolutionary operators; 3) to the best of our knowledge, multiagent technologies have been adopted to design adaptive EC for solving MOPs for the first time; 4) the proposed framework is also the first attempt to consider the use of trust modeling for measuring the dynamic competency of evolutionary operators and control parameters.

Chapter 4

A Multiagent Evolutionary Model for Constructing Robust Trust Networks

As mentioned in Chapters 1-2, the second group of MAEC, "evolutionary computation based MAS", aims at improving agents' communication based on concepts in EC so that agents can well learn other agents' experience and knowledge in solving complex problems. However, most of the exiting studies [88–97] in this category simply assume that agents are benevolent and always willing to share their information with other agents truthfully. This assumption does not hold in many real-world situations. For example, in multiagent based e-marketplaces, buyer agents often ask advice from other buyer agents (called advisors) about seller agents. The advisors may be dishonest and provide untruthful information about sellers. Thus, it is crucial to accurately model the trustworthiness of the advisors and choose the trustworthy ones to ask for information. In this chapter, a novel multiagent evolutionary trust model (MET) is proposed to assist buyers for constructing robust trust networks which consist of trustworthy advisors and their accurate trustworthiness values [17].

More specifically, in the context of the multiagent-based e-marketplaces, when a buyer agent evaluates the reputation of a potential seller agent, it may need to ask for other buyers' opinions (advisor[1] agents' ratings) towards the seller if the buyer does not have

[1]When a buyer evaluates a seller, other buyers are that buyer's advisors. The terms *advisor* and *buyer* are used interchangeably.

much personal experience with the seller before. For the purpose of clarify, this chapter defines the following terms:

- *Honest seller*: A seller that delivers its product as specified in the contract.

- *Dishonest seller*: A seller that does not deliver its product as specified in the contract.

- *Reputation*: A value calculated by trust models to indicate whether a seller will behave honestly in the future: the higher reputation, the higher probability that the seller will behave honestly.

- *Rating*: A positive or negative rating given by a buyer/advisor to a seller indicating that the seller is an honest or dishonest seller, respectively. Besides the binary type, multi-nominal and real ratings are generally used in e-marketplaces.

- *Honest buyer/advisor*: A buyer that always provides positive ratings to honest sellers or negative ratings to dishonest sellers.

- *Dishonest buyer/advisor or Attacker*: A buyer that provides negative ratings to honest sellers or positive ratings to dishonest sellers, with the exception that some special attackers (*e.g.*, Camouflage) may strategically behave like honest buyers in a certain period.

- *Trust or Trustworthiness of buyer/advisor*[2]: A value calculated by trust models to indicate whether an advisor is honest or not: the higher trustworthiness, the higher probability that the advisor is honest.

- *Trust network*: A buyer's trust network connecting itself to other advisors with asymmetric trust values. Trust networks built by buyers can be local, global and dynamic.

[2] Generally, the terms *reputation*, *trust* and *trustworthiness* are used interchangeably in many studies. To avoid confusion, this chapter uses them to model behaviors of sellers and buyers/advisors separately.

Cheating behaviors from sellers, such as not performing the due obligations according to the transaction contract, are still possible to be sanctioned by laws if trust models fail to take effect. However, advisors' cheating behaviors, especially providing *unfair ratings* to sellers, are more difficult to be dealt with. For instance, advisors may collude with certain sellers to boost the sellers' reputation by providing unfairly positive ratings while bad-mouthing the sellers' competitors reputation with unfairly negative ratings.

Various trust models [18–24] have been proposed to cope with unfair ratings. Dishonest advisors' ratings are either filtered out (*e.g.*, BRS [21] and iCLUB [18, 19]) or discounted (*e.g.*, TRAVOS [20], ReferralChain [22] and Personalized [23, 24]) before aggregating advisors' ratings to estimate seller reputation. However, these models are not completely robust against various strategic attacks. In particular, when dishonest advisors occupy a large proportion in e-marketplaces (*i.e.*, Sybil), BRS [21] becomes inefficient and iCLUB [18, 19] is unstable because they both employ the "majority-rule". When dishonest advisors adopt strategic attacks (*i.e.*, Camouflage), TRAVOS [20] does not work well because it assumes an advisor's rating behaviors are consistent. ReferralChain [22] assigns trust value 1 to every new buyer (advisor) which provides a chance for dishonest advisors to abuse the initial trust (*i.e.*, Whitewashing). Personalized [23, 24] is vulnerable when buyers have insufficient experience with advisors and the majority of advisors are dishonest (*i.e.*, combination of Sybil and Whitewashing). Thus, there is an urgent need for building a more robust trust model.

Evolutionary computation, as a search methodology mimicking the natural organism, inherits the characteristic of robustness to enable individuals (agents) to survive in a broad variety of environments [71]. To resist various unfair rating attacks, this chapter thus proposes a multiagent evolutionary trust model (MET) for each buyer to evolve its trust network (consisting of information about which advisors to include in the network and their trustworthiness) over time and finally construct accurate and robust trust networks. Specifically, in each generation, each buyer acquires trust network information from its advisors and generates a candidate trust network using evolutionary operators [3] (*e.g.*, crossover and mutation). Based on fitness evaluation, only trust networks providing more accurate seller reputation estimations will be kept for buyers (*i.e.*, survive to the next generation). To summarize, since MET has the strong ability to explore different kinds of trust networks and is tolerant to small amount of inaccurate or attackers' information, a buyers' trust network found by MET will have high robustness against different kinds of strategic attacks.

Compared to the state-of-the-art trust models [18–24], MET has the following unique characteristics: 1) MET allows a buyer to ask its advisors about their trust network information. But, differing from trust propagation in ReferralChain [22], the buyer only uses advisors' information to generate candidate trust networks that will still go through fitness evaluation; 2) evolutionary operators enable buyers to explore diverse forms of trust networks, to alleviate the impact of false information provided by dishonest advisors; 3) different from most of the trust models that aim to accurately model each individual advisor's trustworthiness, MET finds out the optimal trust network that gives the most accurate seller reputation estimation through fitness evaluation, even certain advisors' trustworthiness values are not very accurate.

The comprehensive experiments are carried out in a simulated multiagent e-marketplace testbed where dishonest sellers try to obtain high transaction volume by colluding with dishonest advisors to launch unfair rating attacks. Experimental results show that MET is more robust than the state-of-the-art trust models and can construct more accurate trust networks to estimate seller reputation.

The rest of this chapter is organized as follows. Section 4.1 surveys the existing trust models and analyzes their potential problems in terms of robustness. The multiagent evolutionary trust model is described in Section 4.2. The extensive experiments are carried out in Section 4.3. Finally, Section 4.4 summarizes this work.

4.1 Related Work

In a large e-marketplace, direct experience between a buyer and a seller is often insufficient or even does not exist. In such a case, the buyer has to rely on indirect experience – opinions of other buyers (playing the role of advisors) towards a target seller. However, various forms of cheating behavior (attacks) from advisors have been observed and studied in the trust research community [25, 26]. This section provides brief descriptions for the typical unfair rating attacks, including **Constant** where dishonest advisors constantly provide unfairly positive/negative ratings to sellers; **Camouflage** where dishonest advisors camouflage themselves as honest advisors by providing fair ratings to build up their trustworthiness first and then give unfair ratings; **Whitewashing** where a dishonest advisor is able to *whitewash* its low trustworthiness by starting a new account with the initial trustworthiness value; **Sybil** where a dishonest advisor creates several accounts to constantly provide unfair ratings to sellers [25, 26]; **Sybil Camouflage** (the combination of Sybil and Camouflage) where a number of dishonest advisors perform Camouflage attacks together; and **Sybil Whitewashing** where a number of dishonest advisors perform Whitewashing attacks together.

Various trust models [18–24] have been proposed to handle unfair ratings from advisors, but they are vulnerable to certain attacks. Specifically, BRS [21] iteratively filters out unfair ratings based on the "majority-rule" where an advisor is considered as an outlier if its ratings are located outside of the acceptable range of all advisors' accumulated ratings. This rule renders BRS vulnerable to Sybil-based attacks because honest buyers' (the minority) ratings will be incorrectly filtered out. In [18, 19], iCLUB is proposed to handle multi-nominal ratings by applying clustering to divide buyers into different clubs. When having little evidence about sellers, a buyer relies on the club with the maximum number of advisors. In this scenario, Sybil attackers forming a club with many members will mislead the buyer to follow their opinions. Thus, iCLUB is also vulnerable to Sybil-based attacks. TRAVOS [20] discounts advisors' ratings by setting weights of their ratings according to their trustworthiness. In some cases, such weights cannot punish dishonest advisors to a large extent when a buyer has insufficient experience with the sellers which the advisors have encountered in the past. In addition, TRAVOS assumes that an advisor's behaviors are consistent, making it vulnerable to Camouflage attacks. Yu and Singh propose referral chains to propagate trust through advisors [22]. The initial trustworthiness values of advisors are set to 1 in the range of $[0, 1]$, and these values will be decreased if the advisors' ratings deviate from a buyer's direct experience. This provides an opportunity for Whitewashing attackers to abuse their initial trustworthiness values. In the Personalized approach [23, 24], the trustworthiness of advisors is calculated based on both private and public trust aspects. The private part is vulnerable to Whitewashing attacks when a buyer does not have many commonly rated sellers with a Whitewashing attacker. The public part cannot work well when the majority advisors are dishonest (Sybil attacks). Thus, this approach is vulnerable to the combination of Sybil and Whitewashing attacks.

In contrast, the proposed MET model takes advantages of the robust ability of evolutionary computation in solving dynamic and complex problems [71]. It also has several unique characteristics compared to the state-of-the-art trust models, as summarized in the previous discussion in this chapter. All these specific designs make MET robust against various unfair rating attacks, which will be demonstrated through experiments in Section 4.3.

4.2 The MET Model

To cope with possible unfair ratings from dishonest advisors, each buyer in the MET model maintains a trust network consisting of a set of advisors, each of which is assigned with a trust value[3]. The buyer then evolves its trust network over time in order to obtain a high quality trust network.

To measure the quality of trust networks, a specific fitness function is designed by considering simultaneously the following two aspects: 1) suitability of selected advisors; 2) accuracy of trust values assigned to these advisors. In each generation, each buyer asks some randomly selected advisors from its trust network for information about their trust networks and fitness values[4]. By comparing its own trust network with those provided by the advisors, the buyer selects three appropriate trust networks to produce a candidate trust network using evolutionary operators. Finally, between the candidate trust network and the buyer's current trust network, the one with higher fitness value will survive to the next generation. The details of the MET model will be provided in the following sections.

4.2.1 Fitness Function

Assume that in an e-marketplace, the set of buyers is denoted as $B = \{B_i | i = 1, \cdots, l\}$ and the set of sellers is denoted as $S = \{S_j | j = 1, \cdots, m\}$. This study also denotes the trustworthiness of an advisor $A_k \in B$ from the view of a buyer B_i as $T_{B_i}(A_k) \in [0, 1]$.

[3] For a new buyer without any knowledge of the e-marketplace environments, it can randomly select a set of other buyers as its advisors, each of which is assigned with a randomly generated trust value.

[4] How to reach a specific advisor to obtain such information as well as seller ratings is out the scope of this work.

In the buyer B_i's trust network TN_{B_i}, the trustworthiness values of advisors connected with B_i are then denoted as $T_{B_i}(A) = \{T_{B_i}(A_k)|A_k \in TN_{B_i}\}$.

A rating provided by a buyer B_i to a seller S_j is denoted as r_{B_i,S_j}, which can be a binary, multi-nominal or real value. Two types of reputation values of S_j can then be derived by buyer B_i. One type of reputation is derived based on the buyer's personal experience with the seller, denoted as $R_{B_i}(S_j)$. And, another type is calculated based on the experience with the seller (*i.e.*, ratings) shared by the advisors in the buyer's trust network, denoted as $\tilde{R}_{B_i}(S_j)$.

In the MET model, a specific fitness function is designed for buyers to measure the quality of their trust networks by comparing the two types of derived reputation values of sellers. Formally, the fitness value of buyer B_i's trust network $T_{B_i}(A) = \{T_{B_i}(A_k)|A_k \in TN_{B_i}\}$ is calculated as:

$$f(T_{B_i}(A)) = \frac{1}{m'} \sum_{j=1}^{m'} |R_{B_i}(S_j) - \tilde{R}_{B_i}(S_j)|, \tag{4.1}$$

where $R_{B_i}(S_j)$ and $\tilde{R}_{B_i}(S_j)$ are the two types of reputation of seller S_j respectively. In addition, $m' \leq m$, indicating that sellers with which either buyer B_i or its advisors have no experience will not be considered in fitness evaluation.

Suppose that the rating type is real, *i.e.*, $r_{B_i,S_j} \in [0,1]$, and the default value $r_{B_i,S_j} = 0.5$ means that B_i has no experience with S_j. The two types of reputation values of seller S_j are then calculated respectively as:

$$\begin{cases} R_{B_i}(S_j) &= \text{mean}(\{r_{B_i,S_j}\}) \\ \tilde{R}_{B_i}(S_j) &= g(T_{B_i}(A_k) \cdot \text{mean}(\{r_{A_k,S_j}\})), \end{cases} \tag{4.2}$$

where $R_{B_i}(S_j)$ is the average of B_i's ratings to S_j on different transactions[5], and $r_{B_i,S_j} \neq 0.5$. The general function $g(\cdot)$ can be a discounting or Dempster-Shafer operator [22]. For simplicity, this study uses the discounting operator as follows:

$$\tilde{R}_{B_i}(S_j) = \frac{\sum_{k=1}^{n} T_{B_i}(A_k) \times \text{mean}(\{r_{A_k,S_j}\})}{\sum_{k=1}^{n} T_{B_i}(A_k)}, \tag{4.3}$$

where $A_k \neq B_i$ (ignore B_i's own opinion), $A_k \in TN_{B_i}$, $n = |TN_{B_i}|$ is the total number of advisors in the trust network ($n \leq l-1$), and $r_{A_k,S_j} \neq 0.5$.

[5] A transaction means that a buyer consumes some goods or services from a seller.

A smaller fitness value indicates that the buyer's trust network is in higher quality, because the combination of advisors' ratings is more similar to the buyer's own opinions regarding the common sellers. In other words, the fitness function measures the suitability of the selected advisors and the accuracy of the trust values assigned to these advisors simultaneously. In addition, it is worth noting that MET does not rely on the trust transitivity or propagation despite the fact that it uses the concept of trust network. The accuracy of the trustworthiness values of all advisors in a buyer's trust network is directly measured by the buyer itself, based on the buyer's own experience with sellers.

4.2.2 Trust Network Comparison

In each generation, each buyer will ask some randomly selected advisors in its trust network to share information about their trust networks and fitness values. However, some dishonest advisors may provide false information. In some cases, honest advisors may unintentionally provide noisy or useless information, because they do not have sufficient experience with sellers in the e-marketplace in order to assign precise trust values to the advisors in its trust network. To alleviate this problem, the buyer will compare trust networks shared by advisors with its own trust network, to choose appropriate trust networks, which will further be used to generate candidate trust networks (see Section 4.2.3).

More specifically, suppose that buyer B_i has the trust network $T_{B_i}(A) = \{T_{B_i}(A_1), \cdots, T_{B_i}(A_x)\}$ and the fitness value is $f(T_{B_i}(A))$. Buyer B_i randomly chooses an advisor A_r from its trust network to ask for the information about the advisor's trust network and its fitness value. Suppose that advisor A_r provides the trust network information as $T_{A_r}(A) = \{T_{A_r}(A_1), \cdots, T_{A_r}(A_y)\}$ and the fitness value $f(T_{A_r}(A))$. The difference between the trust networks of buyer B_i and advisor A_r is calculated as follows:

$$\text{diff}(T_{B_i}(A), T_{A_r}(A)) = \frac{1}{n'} \sum_{k=1}^{n'} |T_{B_i}(A_k) - T_{A_r}(A_k)|, \tag{4.4}$$

where $n' = |TN_{B_i} \bigcup TN_{A_r}|$ is the number of advisors appearing in either the trust networks of B_i or that of A_r, that is the union of advisors in the two trust networks. In some cases, B_i and A_r may not have common advisors. If an advisor A_k appears in one trust network but not the other, the default trust value of A_k in the second trust network will be assigned as 0.5. The difference between the fitness values of B_i and A_r is:

$$\text{diff}(f(T_{B_i}(A)), f(T_{A_r}(A))) = |f(T_{B_i}(A)) - f(T_{A_r}(A))|. \tag{4.5}$$

Buyer B_i chooses to use the information shared by advisor A_r only when the following condition is satisfied:

$$(\text{diff}(T_{B_i}(A), T_{A_r}(A)) - 0.5) \times (\text{diff}(f(T_{B_i}(A)), f(T_{A_r}(A))) - 0.5) > 0. \tag{4.6}$$

The rationale of Eq. 4.6 can be explained by analyzing the following three scenarios. In the first scenario where advisor A_r provides both the real trust network and the real fitness value, Eq. 4.4 and Eq. 4.5 are smaller than 0.5 because they are similar with B_i's own experience. Then B_i will treat A_r as an honest advisor to use its information. In the second scenario where only one type of A_r's information is false, either Eq. 4.4 or Eq. 4.5 is smaller than 0.5. Then the result of Eq. 4.6 is smaller than 0. Advisor A_r will not be selected as an honest advisor by buyer B_i. In the third scenario where A_r provides both a false trust network and a false fitness value, Eq. 4.4 and Eq. 4.5 are both larger than 0.5. Then, A_r's fitness value indicates that A_r's trust network is different from B_i's own opinion, and this valuable information from A_r suggests B_i to avoid such type of trust networks.

4.2.3 Evolutionary Operators

After choosing three[6] advisors by trust network comparison, the buyer will generate a candidate trust network using evolutionary operators. By comparing the candidate trust network with the buyer's own trust network, the one with higher fitness value measured by Eq. 4.1 will survive to the next generation. Two widely used evolutionary operators (DE crossover and polynomial mutation [3]) are utilized to produce candidate trust networks in MET.

The operator "DE/ran/1/bin" in Differential Evolutionary (DE) [3] is adopted for crossover. Let us denote buyer B_i's trust network in generation $g-1$ as $T_{B_i,g-1}(A)$. At

[6]If there is an insufficient number of advisors satisfying Eq. 4.6, some advisors will be randomly selected from the buyer's trust network.

first, the crossover operator generates a new vector $V_{B_i,g}$ as:

$$V_{B_i,g}(A) = T_{A_1,g-1}(A) + F \cdot [T_{A_2,g-1}(A) - T_{A_3,g-1}(A)], \tag{4.7}$$

where F is a scaling factor which amplifies or shrinks the difference vectors. A_1, A_2, A_3 are the three advisors selected in Section 4.2.2. $T_{A_1,g-1}(A), T_{A_2,g-1}(A), T_{A_3,g-1}(A)$ are the trust networks shared by them respectively. After that, operator "DE/rand/1/bin" applies the binomial crossover operation to produce the new trust network.

$$T_{B_i,g}(A_k) = \begin{cases} V_{B_i,g}(A_k) & \text{if } rand_k() \leq CR || k = k_{rand} \\ T_{B_i,g-1}(A_k) & \text{otherwise,} \end{cases} \tag{4.8}$$

where $rand_k() \in [0, 1]$ is a uniformly distributed random number and $k_{rand} \in [1, n]$ is a randomly chosen integer. The control parameter CR is the probability for crossover. If $T_{B_i,g}(A_k) < 0$, it is set to 0. If $T_{B_i,g}(A_k) > 1$, it is set to 1.

In general, buyer B_i uses information from the advisors in its own trust network, which is considered as the *local view*. But, such mechanism may lead solutions to get stuck at local optimum. To balance between exploitation and exploration, the MET model designs a probability variable P_{local} (usually close to 1) to control the buyer's view. It means that the buyer has probability $1 - P_{local}$ to obtain information from all advisors in the e-marketplace as the *global view*.

The polynomial mutation is used to add perturbation to B_i's trust network, which is beneficial to generate different solutions and boost the evolutionary process, as follows:

$$T_{B_i,g}(A_k) = T_{B_i,g}(A_k) + \delta \quad \text{if } rand_k() \leq p_m, \tag{4.9}$$

$$\delta = \begin{cases} (2 \cdot rand())^{1/(\eta_m+1)} - 1 & \text{if } rand() < 0.5 \\ 1 - |2(1 - rand())|^{1/(\eta_m+1)} & \text{otherwise,} \end{cases} \tag{4.10}$$

where η_m is used to control the polynominal probability distribution, and p_m is the mutation probability. After evolutionary operations, the number of advisors in $T_{B_i}(A)$ is retained by discarding advisors with smaller trust values.

4.2.4 Pseudo-code Summary of MET

The pseudo-code summary of MET is given in Algorithm 2. When a buyer has some new experience with certain sellers, the buyer will evolve its trust network to capture

Input : $T_{B_i,0}(A)$, buyer B_i's current trust network;
G, the maximum number of generations;
P_{local}, probability of local view;
Output: The optimal trust network $T_{B_i}(A)$;

1 Calculate the fitness $f(T_{B_i,0}(A))$ using Eq. 4.1;
2 **for** $g = 1$ *to* G **do**
3   **if** $rand() < P_{local}$ **then**
4   Randomly select advisors A_{r1}, A_{r2}, A_{r3} that satisfy Eq. 4.6 from $T_{B_i,g-1}(A)$;
5   **else**
6   Randomly select advisors A_{r1}, A_{r2}, A_{r3} that satisfy Eq. 4.6 from all possible advisors;
7   Generate $T_{B_i,g}(A)$ by DE with A_{r1}, A_{r2}, A_{r3};
8   Apply the polynomial mutation to $T_{B_i,g}(A)$;
9   Calculate the fitness $f(T_{B_i,g}(A))$;
10   **if** $f(T_{B_i,g}(A)) < f(T_{B_i,g-1}(A))$ **then**
11   Replace $T_{B_i,g-1}(A)$ by $T_{B_i,g}(A)$;
12 Output the optimal trust network $T_{B_i,G}(A)$;

Algorithm 2: Multiagent evolutionary trust model (MET)

advisors' dynamic behavior patterns. More specifically, the buyer B_i firstly evaluates its current trust network (in generation 0) based on both the new and old experience with sellers (Line 1). The buyer then acquires trust network information from three randomly selected advisors of its own trust network with the probability P_{local} or from all possible advisors with the probability $1 - P_{local}$. The trust networks shared by all the selected advisors should satisfy Eq. 4.6 (Lines 3-6), to control the quality of the shared trust networks. With the shared trust networks, the buyer then generates a new trust network using the DE operator and polynomial mutation (Lines 7-8). If the newly generated trust network is better than the buyer's current trust network, the better one will survive to the next generation (Lines 9-11). After generations of evolution, an optimal trust network will be finally obtained for the buyer to accurately model seller reputation.

4.3 Experimentation

A set of comprehensive experiments are carried out to evaluate MET. In this section, a multiagent-based e-marketplace testbed is firstly introduced, then the robustness of MET is evaluated by simulating different strategies of advisors for sharing their trust networks, and the parameter settings on MET are examined. After that, MET is compared with five existing trust models to show its advantages of being more robust against various unfair rating attacks and more accurately modeling seller reputation and advisor trustworthiness.

4.3.1 Multiagent-based E-marketplace Testbed

As mentioned in [25, 26, 183], the existing testbeds, such as the Agent Reputation and Trust (ART) testbed, are not suitable for carrying out experiments to compare the robustness of trust models under unfair rating attacks. For example, the winning approach in the ART testbed does not consider reputation ratings from other appraisers. This decision raises concern about the importance of an approach for coping with unfair ratings in this testbed, and whether the results of comparing unfair rating detection approaches based on this testbed will be significant. Thus, a multiagent-based e-marketplace testbed is designed to incorporate different trust models and simulate unfair rating attacks from advisors. In the testbed, this study simulates a scenario of "Duopoly Market" where two sellers occupy a large portion of the total transaction volume in the market. The *dishonest duopoly seller* tries to compete with the *honest duopoly seller* to gain larger transaction volume by recruiting dishonest buyers to perform unfair rating attacks. The other sellers (*common sellers*) include 99 honest sellers and 99 dishonest sellers, and their reputation values are uniformly distributed in the range of [0, 1]. Typically, trust models are most effective when only 30% of buyers are dishonest [21]. Thus, 12 dishonest buyers (attackers) and 28 honest buyers are added into the market for non-Sybil-based attacks, and switch their numbers for Sybil-based attacks. The entire simulation lasts for 100 days. On each day, each buyer chooses to transact with one seller once, and the buyer evolves its trust network using Algorithm 2 after fulfilling the transactions in the end of the day.

Table 4.1: The key parameters in the testbed.

Key parameters	Values
Number of dishonest duopoly sellers	1
Number of honest duopoly sellers	1
Number of dishonest common sellers	99
Number of honest common sellers	99
Number of dishonest buyers ($\|B^D\|$)	12/28*
Number of honest buyers ($\|B^H\|$)	28/12*
Simulation days ($Days$)	100
Dominance ratio ($Ratio$)	0.5

* Non-Sybil-based attack/Sybil-based attack

Since most trust models are more effective when every advisor has transaction experience with many different sellers, this study assumes that buyers will transact with the duopoly sellers with the probability 0.5 while transacting with each common seller randomly. This implies that duopoly sellers occupy half of transactions in the market, which is called the *dominance ratio*. When deciding on which duopoly seller to transact with, honest buyers use trust models to calculate the reputation of the duopoly sellers and transact with the seller with the higher value, while dishonest buyers choose sellers according to their attacking strategies. After each transaction, honest buyers provide fair ratings, whereas dishonest buyers provide ratings according to their attacking strategies. The key parameters are summarized in Table 4.1.

To evaluate the robustness of trust models, this study compares the transaction volume of duopoly sellers. The robustness of a trust model (defense, Def) against an attack model (Atk) is measured as follows:

$$\mathcal{R}(Def, Atk) = \frac{|Tran(S^H)| - |Tran(S^D)|}{|B^H| \times Days \times Ratio}, \tag{4.11}$$

where $|Tran(S^D)|$ and $|Tran(S^H)|$ are transaction volume of the dishonest and honest duopoly sellers, respectively. $\mathcal{R}(Def, Atk) = 1$ or -1 means Def is *completely robust* or *completely vulnerable* to Atk, respectively. The larger value indicates that the trust model is more robust against the attack.

The mean absolute error (MAE) of seller reputation is also adopted to measure the accuracy of trust models in modeling seller reputation:

$$\text{MAE}(S_j) = \frac{\sum_t \sum_{B_i} |R^t(S_j) - \tilde{R}^t_{B_i}(S_j)|}{|B^H| \times Days}, \tag{4.12}$$

where $R^t(S_j)$ is the actual reputation of a seller S_j in day t ($t \in [0, Days]$), and $\tilde{R}^t_{B_i}(S_j)$ is the estimated reputation of S_j by a trust model based on experience of a honest buyer B_i's advisors ($B_i \in B^H$). A smaller MAE indicates that the trust model predicts seller reputation more accurately.

In the testbed, each trust model is tested against every attack over 50 independent runs. The experimental results show the mean and standard deviation ($mean \pm std$), and the best results are in bold font. The ratings to sellers are set as the real type. For parameters of trust models, the values suggested by their authors are adopted. The parameter settings of MET are outlined as following. The number of advisors in a buyer's trust network is $n = 25$. The maximum generation is $G = 10$. The probability of local view is $P_{local} = 0.8$. The DE operator has $CR = 0.6, F = 0.3$, and the polynomial mutation has $\eta_m = 20$ and $p_m = 0.05$.

4.3.2 The Influence of Trust Networks

In MET, buyers exchange information about seller ratings and trust networks. Honest buyers always provide truthful information to others. Besides unfair ratings generated by their various attacking strategies, dishonest buyers may also provide three types of trust networks to other buyers:

- Truthful trust network: a buyer provides truthful information about the trustworthiness of advisors in its trust network. Such information will help other buyers to obtain optimal trust networks quickly.
- Noisy trust network: a dishonest buyer shares a trust network with randomly generated trust values for advisors, implying that some buyers do not share their evolved trust networks but only the initial networks.
- Collusive trust network: a dishonest buyer provides false information about the trustworthiness of some advisors because they are in the same colluding group. Such collusive information is more difficult to detect.

In this section, MET is tested with the three types of trust networks. A rating to a seller from a buyer is a real value. Since BRS [21], TRAVOS [20] and Personalized [23, 24]

Table 4.2: Robustness of MET with the existence of truthful, noisy and collusive trust networks.

	Constant	Camouflage	Whitewashing
MET–truthful	**0.99±0.02**	**0.99±0.03**	**0.99±0.02**
MET–noise	0.98±0.02	**0.99±0.03**	**0.99±0.03**
MET–collusive	0.98±0.02	**0.99±0.02**	0.98±0.04
	Sybil	Sybil Cam*	Sybil WW*
MET–truthful	**0.96±0.07**	**0.99±0.07**	**0.98±0.08**
MET–noise	0.91±0.08	0.96±0.08	0.94±0.08
MET–collusive	0.87±0.15	0.94±0.06	0.82±0.11

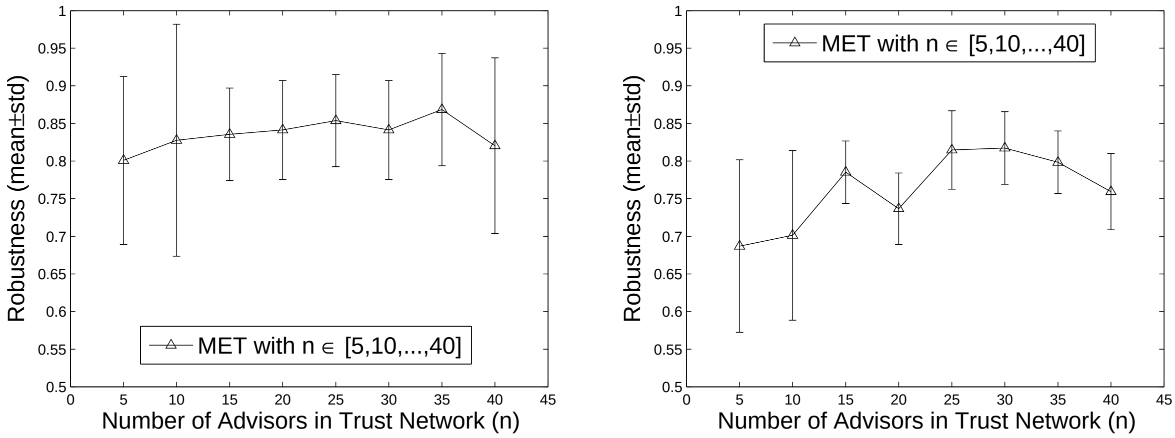
*Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

are designed to deal with binary ratings, the rating is converted to [*negative*, *positive*] when it is in the range of $[0, 0.5)$ and $[0.5, 1.0]$, respectively. The iCLUB approach [18, 19] is proposed for multi-nominal ratings, so the rating is converted to [*worst*, *bad*, *neutral*, *good*, *best*] for the ranges of $[0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), [0.8, 1.0]$, respectively.

In Table 4.2, it can be found that MET is highly robust against the Constant, Camouflage and Whitewashing attacks no matter whether dishonest buyers (advisors) provide truthful, noisy or collusive trust networks. Under the other three Sybil-based attacks, although the robustness values of MET decrease when dishonest buyers provide noisy and collusive trust networks, it still exhibits reasonably high robustness values ($\mathcal{R}(\text{MET}, Atk) \geq 0.82$). Hereafter, MET is tested under the most challenging case (collusive trust networks) unless explicitly indicated. This will also affect the ReferralChain model [22] but not other trust models.

4.3.3 Impact of Parameter Settings on MET

In this section, the impact of the parameter setting (*i.e.*, the number of advisors in a buyer's trust network n) on the performance of MET is investigated in detail. When a buyer acquires information about sellers from its trust network, a smaller/larger value of n means that the buyer can consult less/more advisors, respectively. If the buyer has fewer advisors, the evaluation of seller reputation is difficult because it receives less information on target sellers. On the other hand, when the buyer has too many advisors, MET may require more generations to evolve the buyer's trust network.



4.1.a: MET vs. Sybil   4.1.b: MET vs. Sybil WW

Figure 4.1: MET with different parameter settings.

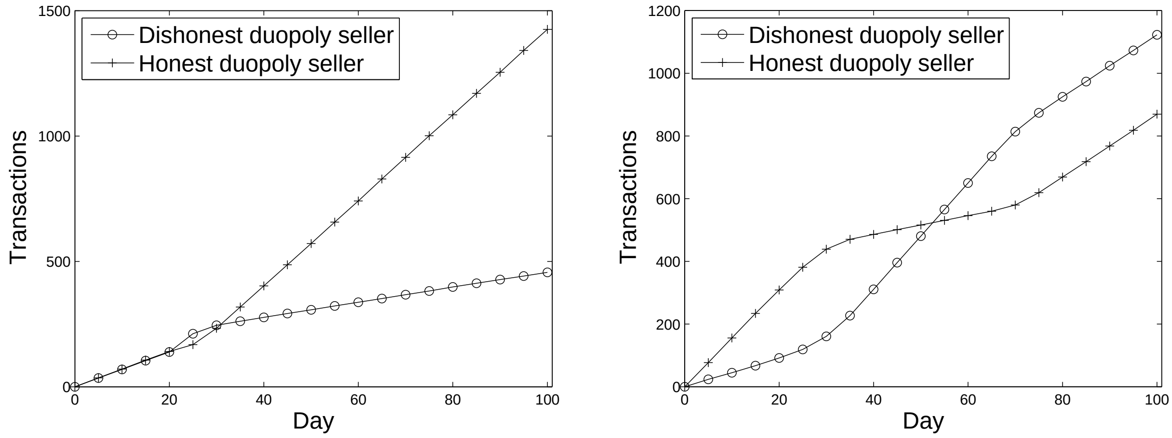
Figs. 4.1(a-b) show the average robustness of MET against Sybil and Sybil Whitewashing, respectively. MET is tested with $n \in [5, 10, \cdots, 40]$ advisors. It demonstrates that the parameter settings have certain impact on MET and $n = 25$ is a good choice in the experiments. As shown in Fig. 4.1(b), it is also worth to notice that MET with different parameters exhibits reasonably high average robustness even under the strongest attack (*i.e.*, $\mathcal{R}(\text{MET}, \text{Sybil WW}) > 0.68$).

4.3.4 Comparison of Robustness

In this section, a set of experiments are also carried out to compare the robustness of MET with that of other trust models. The results are presented in Table 4.3 and Figs. 4.2–4.4.

From Table 4.3, it can be found that all the trust models are robust against the **Constant** attack. Consistent with the authors' own experimental results [21], the table also shows that BRS is not completely robust against the Constant attack ($\mathcal{R}(\text{BRS}, \text{Constant}) = 0.87$).

The **Camouflage** attackers provide fair ratings to common sellers to establish their trustworthiness before day 20, and then give unfair ratings to all sellers. In Table 4.3, ReferralChain's robustness is low with respect to Camouflage. Before day 20, ReferralChain is unable to decrease the trustworthiness of dishonest advisors as they provide fair ratings.



4.2.a: ReferralChain vs. Camouflage

4.2.b: BRS vs. Whitewashing

Figure 4.2: Transaction volume along days for dishonest and honest duopoly sellers. (a) ReferralChain vs. Camouflage, (b) BRS vs. Whitewashing.

Buyers receive seller information from their advisors with equal chance. Thus, the two duopoly sellers obtain similar transaction volume before day 20 (see Fig. 4.2(a)). After that, ReferralChain gradually decreases the trustworthiness of dishonest advisors when they give unfair ratings. The transaction volume of the honest duopoly seller becomes larger than its competitor after day 30.

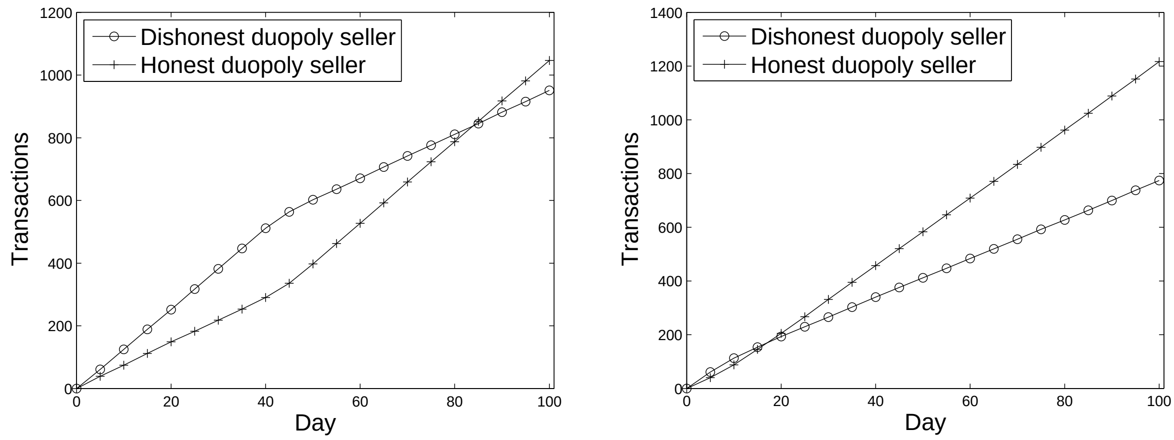
Each **Whitewashing** attacker provides one unfair rating on each day and starts with a new buyer account on the next day. In Table 4.3, the value $\mathcal{R}(\text{BRS}, \text{Whitewashing}) = -0.48$ shows that BRS is vulnerable to this attack. According to Fig. 4.2(b), the honest duopoly seller has a larger number of transactions than the dishonest one in the beginning. However, after some time (around 52 days), the dishonest duopoly seller's transaction volume exceeds its competitor. To explain, after day 52, the accumulated reputation of a seller will more easily fall in the rejection area of the beta distribution of an honest buyer rather than a Whitewashing attacker. This means that honest buyers will be incorrectly filtered out by BRS. Listening to advice from Whitewashing attackers misleads buyers to transact with the dishonest duopoly seller. ReferralChain is vulnerable to Whitewashing because the initial trustworthiness of advisors (buyers) is set to 1 in that model [22], and it is difficult for buyers to select reliable advisors between honest buyers and Whitewashing attackers.

Table 4.3: Robustness of trust models versus attacks.

	Constant	Camouflage	Whitewashing
BRS	0.87±0.03	0.89±0.02	-0.18±0.07
iCLUB	0.98±0.02	**0.99±0.02**	0.77±0.13
TRAVOS	0.97±0.02	0.82±0.03	0.87±0.03
ReferralChain	0.89±0.04	0.69±0.04	-0.95±0.08
Personalized	**0.99±0.03**	**0.99±0.03**	**0.98±0.03**
MET	0.98±0.02	**0.99±0.02**	**0.98±0.04**
	Sybil	Sybil Cam*	Sybil WW*
BRS	-0.99±0.08	-0.47±0.07	-0.30±0.07
iCLUB	0.23±0.35	0.90±0.09	0.20±0.29
TRAVOS	0.16±0.09	-0.57±0.07	-0.98±0.07
ReferralChain	0.82±0.06	0.63±0.08	-0.98±0.07
Personalized	0.74±0.45	**0.94±0.08**	-1.00±0.08
MET	**0.87±0.15**	**0.94±0.06**	**0.82±0.11**

*Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

MET allows buyers to exchange their advisor information to generate a number of candidate trust networks. Through fitness comparison using Eq. 4.1, trust networks with the most suitable advisors will be kept for buyers. It is also difficult for Whitewashing attackers to get into buyers' trust networks. Thus, MET is able to obtain the high robustness of 0.98.



4.3.a: TRAVOS vs. Sybil

4.3.b: Personalized vs. Sybil

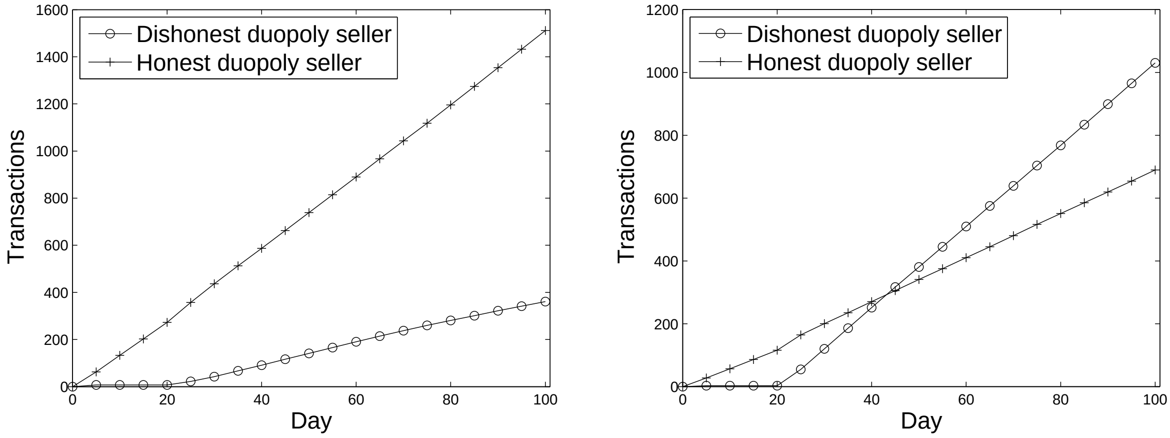
Figure 4.3: Transaction volume along days for dishonest and honest duopoly sellers. (a) TRAVOS vs. Sybil, (b) Personalized vs. Sybil.

BRS is completely vulnerable to the **Sybil** attack due to its employed "majority-rule". The robustness of iCLUB is not stable with the standard deviation of ($std = 0.35$). To explain, with the Sybil attack, the majority of buyers are dishonest. When a buyer only relies on its own experience to model the trustworthiness of advisors, it can have the accurate modeling. If it also relies on the opinions of majority advisors (which are dishonest), it will have incorrect modeling of advisors. In consequence, the modeling of seller reputation will be inaccurate. Personalized also has large standard deviation ($std = 0.45$) because it has the similar design as iCLUB.

TRAVOS is not completely robust against Sybil attacks. In the early period, TRAVOS cannot find enough reference sellers so the discounting of advisors' ratings is not effective (referred to as *soft punishment*). For instance, suppose that the trustworthiness of dishonest/honest advisors is 0.4/0.6, and each advisor (12 honest ones and 28 dishonest ones) gives one rating to a seller. An honest seller's reputation is $0.40 \approx (0.6\times12+1)/(0.4\times28+0.6\times12+2))$ and that of the dishonest seller is $0.60 \approx (0.4\times28+1)/(0.4\times28+0.6\times12+2)$. However, if a trust model is able to set dishonest/honest advisors' trustworthiness as 0.1/0.9, the evaluation of seller reputation will become more accurate. The Personalized approach, in the beginning, also suffers from the *soft punishment* when a buyer relies on the public trust to evaluate advisors' trustworthiness. As shown in Figs. 4.3(a-b), when buyers have more experience, TRAVOS and Personalized become more effective after day 80 and day 15, respectively.

MET generates diverse trust networks with several candidate trust values of advisors using evolutionary operators, and keeps the best trust network with the most accurate trust values assigned to advisors. This increases the chance for buyers to punish Sybil attackers to a large extent. Thus, MET obtains the high robustness as $\mathcal{R}(\text{MET}, \text{Sybil}) = 0.87$.

Unlike the Sybil attack, **Sybil Camouflage** is unable to render BRS completely vulnerable. This is because in the beginning attackers camouflage themselves as honest ones by providing fair ratings where BRS is always effective. After attackers stop camouflaging, the dishonest duopoly seller's transaction volume will soon exceed its competitor. Under Camouflage and Sybil Camouflage, the robustness of ReferralChain is similar because the buyer aggregates only its local advisors' ratings to predict sellers' reputation.



4.4.a: TRAVOS vs. Camouflage

4.4.b: TRAVOS vs. Sybil Camouglage

Figure 4.4: Transaction volume along days for dishonest and honest duopoly sellers. (a) TRAVOS vs. Camouflage, (b) TRAVOS vs. Sybil Camouflage.

Comparing with Camouflage, TRAVOS becomes vulnerable to Sybil Camouflage. Although TRAVOS will inaccurately promote the trustworthiness of a Camouflage attacker (most are slightly larger than 0.5), when majority of buyers are honest, the aggregated ratings from attackers are still not able to overweigh honest buyers' opinions. However, under Sybil Camouflage, when majority are dishonest buyers, these attackers' aggregated ratings will easily overweigh honest buyers' opinions and render TRAVOS vulnerable. Figs. 4.4(a-b) clearly show the difference in the robustness of TRAVOS against Camouflage and Sybil Camouflage, respectively.

Sybil Whitewashing is the strongest attack among the six investigated attacks. It can defeat BRS, TRAVOS, ReferralChain and Personalized as shown in Table 4.3. Similar to Sybil, the robustness of iCLUB against Sybil Whitewashing is still unstable. Comparing with Whitewashing, BRS is still vulnerable to Sybil Whitewashing, while TRAVOS and Personalized change dramatically from being robust to completely vulnerable. For TRAVOS, since each Whitewashing attacker provides only one unfair rating, buyers cannot find reference sellers to discount attackers' trustworthiness to a large extent. When majority are softly punished attackers, TRAVOS will always suggest honest buyers to transact with the dishonest duopoly seller. For Personalized, a buyer cannot find enough commonly rated sellers and will heavily rely on the public trust component to evaluate an advisor's trustworthiness, which is inaccurate when majority of buyers are dishonest. Thus, similar to TRAVOS, Whitewashing attackers' trustworthiness cannot be discounted to a large extent and *soft punishment* renders Personalized completely vulnerable.

ReferralChain is completely vulnerable to Sybil Whitewashing, whereas MET is sufficiently robust to this attack ($\mathcal{R}(\text{MET}, \text{Sybil WW}) = 0.82$ in Table 4.3). Although both ReferralChain and MET allow buyers to ask their advisors about other advisors, MET applies evolutionary operators (*e.g.*, crossover and mutation) to generate new trust networks. The candidate trust networks will go through the fitness evaluation. Only when the selected advisors and trustworthiness values for those advisors show better accuracy, they will be kept by buyers. Besides, unlike ReferralChain, MET does not assign high initial trust values to advisors.

In summary, experimental results show that iCLUB and the Personalized approach have large perturbation under Sybil attacks. BRS, TRAVOS and ReferralChain are vulnerable to Sybil, Camouflage and Whitewashing, respectively. It demonstrates that the proposed MET model is more robust than these other trust models against typical attacks.

Table 4.4: Mean Absolute Error (MAE) of reputation estimation for dishonest duopoly sellers.

	Constant	Camouflage	Whitewashing
BRS	0.52±0.05	0.50±0.04	0.74±0.02
iCLUB	0.83±0.21	0.73±0.14	0.80±0.09
TRAVOS	0.42±0.03	0.56±0.01	0.57±0.02
ReferralChain	0.05±0.01	0.15±0.01	0.59±0.03
Personalized	0.45±0.08	0.46±0.07	0.83±0.03
MET	**0.02±0.01**	**0.02±0.01**	**0.03±0.01**
	Sybil	**Sybil Cam***	**Sybil WW***
BRS	0.73±0.03	0.67±0.03	0.61±0.01
iCLUB	**0.06±0.01**	0.70±0.10	**0.06±0.01**
TRAVOS	0.29±0.01	0.54±0.02	0.56±0.02
ReferralChain	0.08±0.02	0.19±0.02	0.68±0.04
Personalized	0.24±0.07	0.59±0.08	0.24±0.02
MET	0.07±0.04	**0.11±0.02**	0.20±0.06

*Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

Table 4.5: Mean Absolute Error (MAE) of reputation estimation for honest duopoly sellers.

	Constant	Camouflage	Whitewashing
BRS	0.19±0.06	0.11±0.04	0.58±0.02
iCLUB	**0.01±0.00**	**0.01±0.00**	0.11±0.07
TRAVOS	0.17±0.01	0.25±0.01	0.28±0.01
ReferralChain	0.06±0.02	0.16±0.01	0.97±0.03
Personalized	0.02±0.00	**0.01±0.00**	0.06±0.01
MET	**0.01±0.00**	**0.01±0.00**	**0.05±0.03**
	Sybil	**Sybil Cam***	**Sybil WW***
BRS	0.99±0.00	0.73±0.01	0.64±0.01
iCLUB	0.35±0.17	**0.01±0.00**	0.37±0.14
TRAVOS	0.44±0.02	0.57±0.01	0.84±0.01
ReferralChain	0.10±0.02	0.19±0.02	0.99±0.01
Personalized	0.20±0.21	0.05±0.00	0.96±0.02
MET	**0.09±0.06**	0.08±0.02	**0.16±0.11**

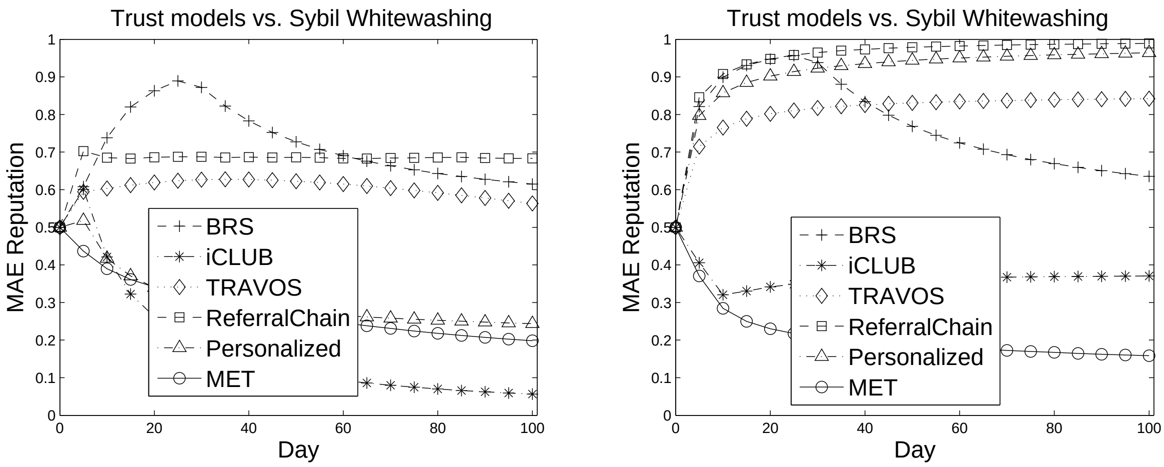
*Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

4.3.5 Accuracy of Modeling Seller Reputation

This section carries out further experiments to compare trust models in term of mean absolute error (MAE) of duopoly sellers' reputation. The smaller MAE indicates that the trust model is more accurate in modeling seller reputation.

In Table 4.4, under Constant, Camouflage and Whitewashing, MET is able to obtain the best results for both duopoly sellers' reputation. Under other Sybil, Sybil Camouflage and Sybil Whitewashing attacks, MET and iCLUB provide the best MAE values. In most cases, other trust models (except MET) obtain smaller MAE for the honest duopoly seller reputation while larger MAE for the dishonest duopoly seller reputation, implying that it is more difficult to obtain accurate reputation of dishonest sellers because they recruit attackers to perform strategic attacks.

For Sybil and Sybil Whitewashing, iCLUB obtains the best results on the dishonest duopoly seller (MAE(S^D) = 0.06), whereas it is unable to accurately estimate the honest duopoly seller's reputation (MAE(S^H) = 0.37). It is consistent with the results of robustness comparison in the previous section. To explain, when a buyer conducts enough transactions with the dishonest duopoly seller, iCLUB adopts the buyer's local knowledge to calculate the dishonest seller's reputation. However, in some cases, the buyer has little evidence about the honest duopoly seller, and iCLUB has to rely on global knowledge to calculate the honest seller's reputation. When majority of advisors are dishonest, iCLUB suggests the buyer to transact with the dishonest duopoly seller rather the honest one, and then a rating will be given to the dishonest duopoly seller. The consequence is that the buyer will still not be able to have sufficient experience with the honest duopoly seller to accurately model this seller's reputation.



4.5.a: Dishonest Duopoly Seller

4.5.b: Honest Duopoly Seller

Figure 4.5: MAE of duopoly sellers' reputation.

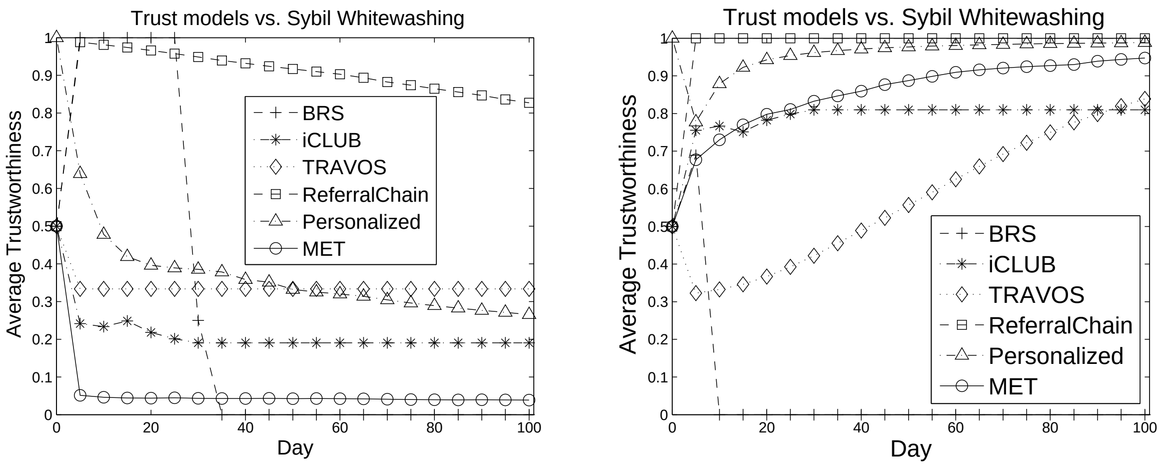
Fig. 4.5 shows the MAE of duopoly sellers' reputation over time by trust models against Sybil Whitewashing. MET and iCLUB can achieve small MAE values for modeling both duopoly sellers' reputation. Thus, iCLUB and MET are more effective than the other four trust models in predicting sellers' reputation.

4.3.6 Accuracy of Modeling Advisor Trustworthiness

This section also shows how the different trust models can accurately model the trustworthiness of both honest and dishonest advisors, under the Sybil Whitewashing attack[7].

Figs. 4.6(a-b) show the average trustworthiness of dishonest and honest advisors modeled by honest buyers, respectively. BRS assigns both dishonest and honest advisors'

[7]The Sybil Whitewashing attack is chosen because it is the strongest attack among the six typical attacks.



4.6.a: Dishonest Advisors

4.6.b: Honest Advisors

Figure 4.6: Average trustworthiness of advisors.

trustworthiness as zero after day 35 because it filters out all the advisors, which is similar to the reason why BRS is vulnerable to Whitewashing in Section 4.3.4. Although ReferralChian assigns the trustworthiness value 1 to the honest advisors, it maintains the average trustworthiness of dishonest advisors at a high level (*i.e.*, larger than 0.8 in Fig. 4.6(a)). As shown in Figs. 4.6(a-b), iCLUB, TRAVOS and Personalized are able to reduce or increase the average trustworthiness of dishonest or honest advisors, respectively. However, these three trust models cannot enforce the difference in dishonest and honest advisors' average trustworthiness to a large extent. In contrast, MET is more effective than those trust models for modeling the trustworthiness of advisors.

4.4 Summary

In this chapter, a novel multiagent evolutionary trust (MET) model is proposed for constructing robust and accurate trust networks for buyers to accurate model seller reputation in the presence of unfair rating attacks. Each buyer in the MET model evolves its own trust network by asking its advisors to provide their trust network information. By doing so, the buyer is able to select suitable advisors into its trust network, and assign accurate trustworthiness to these advisors simultaneously. Experimental studies confirm that MET is more robust and effective than the state-of-the-art trust models against various unfair rating attacks.

Thus, the MET model demonstrates contributions to the second category of MAEC "evolutionary computation based MAS" by adopting evolutionary computation techniques into the design of an effective MAS. At the same time, the MET model addresses the issue of untruthful information involved in agent communications, through the case study in the specific context of MAS based e-marketplaces. The high-level idea of this model is generally applicable to scenarios where agents have to selectively make use of advice (shared experience and knowledge) provided by other agents in solving complex problems, because some of the other agents may be malicious and share false information.

Chapter 5

A Multiagent Pheromone-based Traffic Management Framework for Reducing Traffic Congestion

As discussed in Chapters 1-2, *traffic congestion* is a key problem to cause driver frustration, air pollution and billion-dollars loss each year in today's urban cities [35, 113]. To address this complex problem, some studies [32–34, 70, 113] in the third direction of MAEC have been proposed to reroute vehicles and control traffic lights. However, they either focus on one of the two components (*i.e.*, vehicle rerouting and traffic light control) [32–34, 70] or optimize the two components in a sequential order [113]. In this chapter, a novel multiagent pheromone-based traffic management framework is proposed for reducing traffic congestion, which adopts the notion of digital "pheromone" to bridge vehicle rerouting and traffic light control [31]. This study belongs to the third direction of MAEC.

A transportation system can be formulated as an open, large, distributed and dynamic multiagent system where cars represented as agents move on road networks following their own routes. For instance, in a typical multiagent-based transportation system, the car agents may continually enquire traffic information (*e.g.*, traffic densities, mean speeds) from roadside infrastructure agents (*e.g.*, loop detectors, video cameras) to search routes with smaller traveling cost. On the other hand, all cars must obey the instructions from traffic light control agents to pass or stop at road intersections. Due to the increasing number of vehicles and the limited capacities of road networks in today's urban areas, traffic congestion becomes a major concern of governments and a complex problem in industrial applications. It also draws a lot of attention from academic [33–35, 39, 40].

To alleviate traffic congestion, various approaches [32–40, 113] have been proposed in the field of intelligent transportation systems (ITS) for rerouting vehicles with less travelling cost and controlling traffic lights to maximize road capacities. For instance, traffic congestion forecasting [32–35] is proposed to detour cars to avoid traffic jam in advance. However, these approaches ignore drivers' route intentions of changing directions in near future [36–38]. On the other hand, traffic light control [39, 40] automatically sets traffic light states and calculates the time duration of these states. Nevertheless, these studies are not suitable for large scale traffic scenarios since they need to know the global information of the testing traffic maps. In addition, Bazzan *et al.* [113] adopt evolutionary techniques in EC to optimize both vehicle rerouting and traffic light control. However, this approach does not optimize the two components in a simultaneous manner under a unified framework.

In this chapter, a novel multiagent pheromone-based traffic management framework is proposed for reducing traffic congestion, which delicately defines a notion of digital "pheromone" for the purpose of bridging vehicle rerouting and traffic light control [31]. Specifically, each car agent deposits two types of pheromone (*i.e.*, the traffic pheromone and intention pheromone representing the current and future traffic densities, respectively) on its route. Road infrastructure agents fuse the pheromone to forecast traffic conditions without resorting to a central ITS server, achieving the distributed property. Once a road congestion is predicted, the proposed framework adopts a proactive vehicle rerouting algorithm for assigning alternative routes to cars before they enter the congested road. At the same time, an online traffic light control strategy is designed to assign long time duration of green traffic lights to the roads with a large amount of pheromone.

The proposed traffic management framework is verified by the platform of simulation of urban mobility (SUMO) [114], on two road networks. Extensive experimental results show that the proposed framework significantly alleviates traffic congestion, saves travel time, and reduces air pollution and fuel consumption. Moreover, the proposed framework is robust to different levels of compliance and penetration rates.

The rest of this chapter is organized as follows. Section 5.1 surveys the existing work on reducing traffic congestion by vehicle rerouting and traffic light control. The multiagent pheromone-based traffic management framework is presented in Section 5.2, which includes traffic forecasting by pheromone, pheromone-based vehicle rerouting and pheromone-based traffic light control. The experiments are carried out on the platform SUMO in Section 5.3. Finally, Section 5.4 summarizes the current study.

5.1 Related Work

In recent years, agent-based techniques [32–40, 113, 184] have gained popularity in intelligent transportation systems (ITS) for the purpose of alleviating traffic congestion. Based on various applications, they can be generally divided into three groups: vehicle rerouting [32–38, 41], traffic light control [39, 40] and the combination of these two components [113, 184].

In large scale road networks with highly dynamic traffic flows, vehicle rerouting with a small response time is challenging. To quickly respond to route enquires, the approaches in [41] pre-compute shortest paths among transit nodes or landmarks (referred to as important nodes) on a single snapshot of a traffic map. When a driver asks for an origin-destination service from a central ITS server, it only needs to calculate partial paths, such as that from the origin/destination nodes to the important nodes. The shortcoming is that these approaches find routes without considering dynamic traffic conditions. On the other hand, commercial systems developed by companies (*e.g.*, Google, Mircrosoft and TomTom) make use of roadside infrastructures to collect dynamic traffic information and provide traffic-aware shortest routes. Although these systems are able to predict long-term traffic congestion and its duration using advanced data mining techniques, they are often used as reactive solutions that remain unable to prevent congestion. In addition, these systems may cause “route flapping” [35] where congestion switches from one road to another road when a significant number of cars follow the same route guidance. On the other hand, the above solutions heavily rely on the central ITS server, bringing up the risk of bottleneck and time delay. In contrast to these studies, the proposed traffic management framework is a decentralized approach where cars exchange traffic information with the local and distributed roadside infrastructures.

Some other vehicle rerouting approaches [32–38] proactively direct cars to avoid traffic congestion. For instance, Ando *et al.* [32, 34] introduce the “pheromone” concept from bio-inspired techniques [39] in EC to forecast short-term traffic congestion. Each car deposits three types of pheromone (*i.e.*, basic traffic pheromone, braking pheromone, distance pheromone) on its route, which represent the traffic information of speed, halting cars and distance between two cars. Kurihara [33] defines the digital pheromone as two kinds of traffic information, such as the upstream flow of traffic densities and the downstream flow of traffic congestion. Then, the distributed roadside infrastructures in [32–34] combine the multiple flavors of digital pheromone to predict short-term traffic conditions. On the other hand, Pan *et al.* [35] count the number of cars on different roads to estimate the current traffic conditions. Once congested roads are predicted, they recursively apply the k-shortest path on cars to find alternative routes. However, these approaches [32–35] forecast traffic congestion based on historical data while ignoring another important traffic information, *i.e.*, drivers' route intentions of changing directions in the near future [36–38]. For instance, Dresner [36] and Vasirani [37] design the reservation-based method to manage traffic flows according to drivers' route intentions. Yamashita *et al.* [38] adopt the similar idea that all cars report their route information to the central ITS server including their current positions, destinations and routes to destinations. Inspired by these approaches [32–38], this study defines traffic pheromone and intention pheromone for representing the current traffic densities and the future traffic densities, respectively.

On the other hand, traffic light control [39, 40] is an effective approach to manage traffic flows, which is able to automatically set traffic light states and calculate the time duration of these states. For instance, Wiering *et al.* [40] adopt the reinforcement learning method to minimize the overall waiting time of cars. However, this approach is not suitable for large scale traffic scenarios since it needs to store the global information of all traffic lights. Sánchez *et al.* [39] utilize the genetic algorithms (GAs) in EC to find the optimal combination of traffic lights. In particular, the evolutionary optimizer is an offline algorithm using the trial-and-error procedure. It is only designed for simulations, because cars' movement in realistic transportation systems is irrevocable. In comparison to [39], the traffic light control designed in the proposed framework is an online strategy, which directly calculates the time duration of traffic lights for competing roads according to the amount of digital pheromone.

Furthermore, the combination of the advantages of vehicle rerouting and traffic light control has been considered as a promising direction in ITS. For instance, Bazzan *et al.* [113] firstly adopt evolutionary techniques in EC to search optimal routes for cars. Then, they find the best traffic signals for the ITS system based on the obtained routes. However, the trial-and-error method in EC is only designed for simulation, because the evaluation of potential solutions (*i.e.*, routes and traffic signals) requires the experience of the cars after they have travelled through the streets in the optimal paths. On the other hand, the sequential optimization procedure in [113] is inefficient, since it increases the chance of divergency so that the algorithm has high probabilities of being trapped into oscillating solutions. In comparison to [113], the proposed framework in this chapter is designed as a unified framework, which fuses the two components (*i.e.*, vehicle rerouting and traffic light control) based on the deliberately designed digital "pheromone" information.

5.2 Multiagent Traffic Management Framework

The key idea of the multiagent traffic management framework is to introduce a newly defined notion of digital "pheromone" so as to simultaneously optimize vehicle rerouting and traffic light control.

5.2.1 Traffic Congestion Forecast by Pheromone

In nature, the term "pheromone" refers to communication chemical that is mutually understood by the individuals of the same species [32, 34]. For instance, ants use trail pheromone (intensity of flavors) to guide other ants on attracting routes for food foraging. Bees utilize alert pheromone (repulsive sign) to advise other bees for performing defensive actions. Inspired by this phenomenon, this study defines the multiple flavors of digital pheromone for forecasting traffic congestion. To this end, two types of pheromone, *i.e.*, traffic pheromone and intention pheromone, are designed to represent the current and future road conditions, respectively.

The *traffic pheromone* is defined to estimate the current traffic density (number of cars per meter). In a traffic map, each road has its maximum capacity. A large traffic density value means that many cars are moving on a road, and implies that traffic congestion may occur on this road in the near future. The traffic pheromone is thus formulated as follows:

$$\tau_1(p,t) = \frac{N(p,t) \times L_{\text{car}}}{L_p \times \text{lanes}(p)}, \tag{5.1}$$

where $N(p,t)$ is the number of cars on road p in time interval $(t-1,t]$, L_{car} is the mean length of cars[1], L_p is the length of road p, and lanes(p) is the number of lanes on road p. $\frac{L_p \times \text{lanes}(p)}{L_{\text{car}}}$ is the maximum number of cars that are allowed to move on road p (*i.e.*, the maximum road capacity). Thus, the traffic pheromone is in the range $\tau_1(p,t) \in [0,1]$.

When a congested road is detected, cars on this road have to find new routes. However, this reactive process may be too late since the traffic jam has already been formed and propagated to surrounding roads. In view of this limitation, it is important to forecast traffic congestion before a driver determines which road to proceed [185–187]. As a consequence, this proactive action is beneficial for distributing traffic volume onto roads with low traffic density values.

To predict traffic conditions, this study absorbs another important traffic information, *i.e.*, drivers' route intentions. Suppose that each car reports its route to the local roadside infrastructures (*e.g.*, using smart phones). The route information includes the car's intentions of changing directions on different roads. This study defines the *intention pheromone* to estimate the future traffic density, which takes the form of:

$$\tau_2(p,t+1) = \frac{(I(p,t+1) - O(p,t+1)) \times L_{\text{car}}}{L_p \times \text{lanes}(p)}, \tag{5.2}$$

where $I(p,t+1)$ and $O(p,t+1)$ are the numbers of incoming and outgoing cars on road p in time interval $(t,t+1]$, respectively. Note that these two terms describe the future road conditions (*i.e.*, in the time interval of $(t,t+1]$). Both $I(p,t+1)$ and $O(p,t+1)$ take the similar formulation. For instance, the incoming traffic volume $I(p,t+1)$ is defined as:

$$I(p,t+1) = \sum_{p' \in P_{\text{nei}}} T_g(p')V_f(p')\tau_1(p',t) \times \rho \times \frac{\text{lanes}(p')}{L_{\text{car}}}. \tag{5.3}$$

Here, $I(p,t+1)$ counts the number of cars that will move from the neighboring roads $p' \in P_{\text{nei}}$ to road p in time interval $(t,t+1]$. In Eq. 5.3, $T_g(p')$ is the time duration of the green traffic lights on road p', $V_f(p')$ is the free speed (or the maximum speed limit)

[1]The length of a car is the sum of the car length and the minimum gap between two cars.

of road p', $\tau_1(p', t)$ is the traffic density of road p' in time interval $(t-1, t]$, and ρ is the proportion of cars that have intentions of moving from road p' to road p. In a real traffic scenario, different roads have different capacities. For instance, even two roads have the same traffic density value, the road with many lanes will contain a large number of cars. Due to this consideration, the term $\frac{\text{lanes}(p')}{L_{\text{car}}}$ in Eq. 5.3 is used to align the traffic volume between road p' and road p.

The natural process of pheromone update includes two major characteristics: evaporation and propagation. Evaporation means that old pheromone will disappear as time passes. Propagation implies that pheromone will spread to surrounding environments. Based on the formulation of Eq. 5.2, the intention pheromone $\tau_2(p, t+1)$ can be considered as a type of propagation, which defines the incoming and outgoing cars in the neighboring roads. This study specifically defines an evaporation rate as follows:

$$e(p,t) = \frac{\bar{V}(p,t)}{V_f(p)} \times \frac{1}{1 + |\text{Halts}(p,t)|}, \tag{5.4}$$

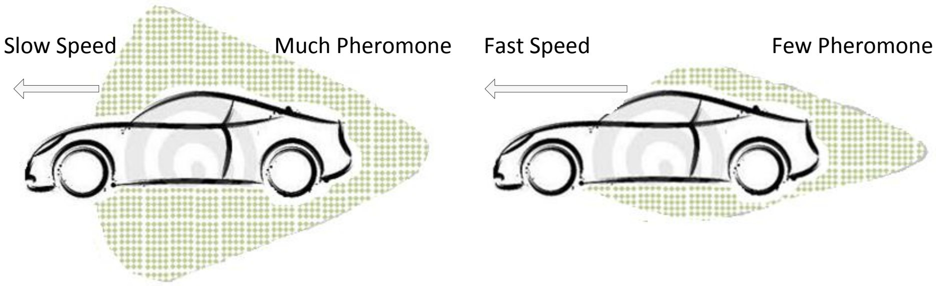
where $\bar{V}(p,t)$ is the mean speed of cars on road p in time interval $(t, t+1]$ and $|\text{Halts}(p,t)|$ is the number of halting cars on road p in time interval $(t-1, t]$. Based on this definition, the evaporation rate belongs to the range $e(p,t) \in [0,1]$.

Then, the roadside infrastructures collect the multiple flavors of digital pheromone (*i.e.*, traffic pheromone and intention pheromone) and fuse them by:

$$\tau(p, t+1) = (1 - e(p,t)) \times \tau_1(p,t) + e(p,t) \times \tau_2(p, t+1). \tag{5.5}$$

Taking Fig. 5.1 as an illustrative example to explain Eqs. 5.4-5.5, the car with slow speed will deposit much pheromone on the road in Fig. 5.1(a). On the contrary, the fast speed leads to few deposited pheromone in Fig. 5.1(b). If the mean speed $\bar{V}(p,t)$ is large enough to reach the free speed $V_f(p,t)$, $\frac{\bar{V}(p,t)}{V_f(p,t)}$ in Eq. 5.4 is equal to 1, and no traffic pheromone $\tau_1(p,t)$ will remain as detailed by Eq. 5.5. On the other hand, the old pheromone will remain as the number of halting cars increases. Thus, the evaporation rate in Eq. 5.4 involves the term $\frac{1}{1+|\text{Halts}(p,t)|}$. From the definitions of Eqs. 5.1-5.5, the digital pheromone lies in the range $\tau(p, t+1) \in [0,1]$.

The available traffic information makes it feasible to instantiate the digital pheromone into the realistic intelligent transportation system (ITS). For instance, the basic information, such as length of car L_{car}, length of road L_p, lanes of road lanes(p), and the



5.1.a: Slow speed, much Pheromone

5.1.b: Fast speed, few Pheromone

Figure 5.1: Digital pheromone with different speeds.

free speed $V_f(p)$, can be obtained by car companies and urban planning administrations. Cars can use a speed pulse to record their speed. Roadside infrastructures utilize digital cameras to count the numbers of moving cars $N(p,t)$ and the halting cars $|\text{Halts}(p,t)|$. Thus, the digital pheromone can be easily applied into ITS.

In this study, the classical ϵ-SVR in LIBSVM [188] is adopted as another model to combine the above two pheromone. In particular, the input training instances are $\{(x_1,z_1),\ldots,(x_t,z_t)\}$, where $x_t = (\tau_1(p,t-1),\tau_2(p,t))$ and $z_t = \tau(p,t) = \tau_1(p,t)$[2] are feature vector and target output, respectively. After obtaining the input pheromone information in time $(t-1,t]$ as $x = (\tau_1(p,t),\tau_2(p,t+1))$, the digital pheromone is forecasted as:

$$\tau(p,t+1) = z_{t+1} = \sum_{i=1}^{t}(-\alpha+\alpha^*)K(x_i,x)+b, \tag{5.6}$$

where α,α^*,b are estimated parameters by solving the dual problem of ϵ-SVR, $K(\cdot)$ is the RBF kernel and the training set includes $t = 60$ instances.

5.2.2 Pheromone-based Vehicle Rerouting

Based on the digital pheromone definition, short-term traffic conditions can be predicted. Once a congested road is forecasted, the roadside infrastructure will broadcast the warning information and send new routes to surrounding cars to avoid traffic jam. In the

[2]This study sets $z_t = \tau(p,t) = \tau_1(p,t)$ since digital pheromone $\tau(p,t)$ can be verified as traffic pheromone $\tau_1(p,t)$ when reaching time t.

proposed framework, a proactive vehicle rerouting algorithm is designed to search new routes with less congested roads (*i.e.*, small amount of pheromone).

Input : G, road network; δ: congestion threshold; R', old routes; t, current time;
Output: Road network with new pheromone G;

1 Update G's roads with pheromone using Eq. 5.5 or Eq. 5.6;
2 Find congested roads $p \in P_{\text{con}}$ by $\tau(p, t+1) > \delta$;
3 **while** $|P_{\text{con}}| > 0$ **do**
4   Find road $p \in P_{\text{con}}$ with maximum $\tau(p, t+1)$;
5   Get neighboring roads P_{nei} connected to road p;
6   Get nodes N_p connected to road p;
7   **for** $p' \in P_{\text{nei}}$ **do**
8   **for** $n \in N_p$ **do**
9   find new routes $R(n)$ by k-shortest path;
10   Get car agents C on road p';
11   **for** $c \in C$ **do**
12   Get $n \in N_p$ from car c's old route $r' \in R'$;
13   Set new route $r = R(n)$ to car c ;
14   Update intention pheromone $\tau_2(p', t+1)$;
15   Update pheromone $\tau(p', t+1)$ by Eq. 5.5 or Eq. 5.6;
16   Find congested roads $p \in P_{\text{con}}$ by $\tau(p, t+1) > \delta$;
17   Delete the checked road $P_{\text{con}} = P_{\text{con}} - p$;
18 Output road network with new pheromone G;

Algorithm 3: Pheromone-base Vehicle Rerouting.

The pseudo-code summary of the pheromone-based vehicle rerouting algorithm is given in Algorithm 3. Lines 1-2 find the congested roads $p \in P_{\text{con}}$ if the digital pheromone of road p is larger than the congestion threshold $\tau(p, t+1) > \delta$. In Lines 3-17, this algorithm recursively assigns new routes to cars on the neighboring roads and updates the road network with new digital pheromone. In particular, Line 4 predicts the most congested road p at time $t+1$. Line 5 and Line 6 identify the neighboring roads P_{nei} and the node set N_p that are connected to road p, respectively. In Lines 8-9, the k-shortest path[3] is used to search the alternative routes R on road network G. In Lines 12-13, the new routes R are assigned to cars C. Line 14 reports the new intention pheromone

[3] Other path optimizers (*e.g.*, Dijkstra [189], A*) can also be integrated into the pheromone-based vehicle rerouting algorithm.

$\tau_2(p', t+1)$ to the local roadside infrastructures. In Line 15, the road network is updated with new digital pheromone $\tau(p', t+1)$. Lines 16-17 find new congested roads $p \in P_{\text{con}}$ based on $\tau(p, t+1) > \delta$. To avoid searching on the same road, Line 17 deletes the checked road $P_{\text{con}} = P_{\text{con}} - p$. When all the congested roads have been checked ($|P_{\text{con}}| == 0$), the algorithm outputs the road network with new digital pheromone G in Line 18.

5.2.3 Pheromone-based Traffic Light Control

In a realistic transportation system, traffic lights are signalling devices positioned at road intersections to control competing traffic flows [39, 40]. In general, traffic light control includes two major parts: color phases and time duration. The color phase is a sequence of standard colors (*i.e.*, red, yellow/amber, and green). The time duration describes the displaying period of the color phase.

In traffic scenarios, traffic light control often adopts fixed color phases and predefined time duration regardless of dynamic traffic conditions. However, this fixed traffic light pattern is inefficient. For instance, even when the forward road is free, cars will still have to wait for a certain time period because the red traffic light is displayed.

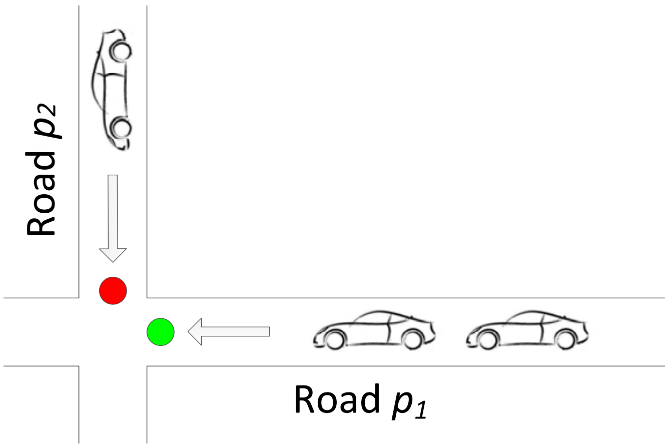


Figure 5.2: Pheromone-based Traffic Light Control.

In view of this limitation, this study proposes an online traffic light control strategy, which automatically sets color phases and calculates the time duration of these phases on competing roads according to the amount of digital pheromone. Take Fig. 5.2 as an illustrative example, and suppose that roads p_1, p_2 are one-way roads. The two roads have the competing relationship, where only one of the roads can be switched to green traffic lights at any time point. Based on their digital pheromone values, for example $\tau(p_1, t+1) > \tau(p_2, t+1)$, road p_1 should be assigned the green traffic light, and road p_2 is the red traffic light. To fully utilize road capacities, this study assumes the time duration of the green traffic light for road p_1, denoted as $T_g(p_1)$, satisfies:

$$\tau(p_1, t+1) \times (1 - \frac{T_g(p_1)V_f(p_1)}{L_{p_1}}) = \tau(p_2, t+1). \tag{5.7}$$

If the traffic density of road p_2 remains the same level, Eq. 5.7 indicates that the traffic density of road p_1 would be the same as the value of road p_2 after time duration $T_g(p_1)$. $\frac{T_g(p_1)V_f(p_1)}{L_{p_1}}$ in Eq. 5.7 is the proportion of the decreasing traffic density when cars on road p_1 travel at the free speed $V_f(p_1)$ within time duration $T_g(p_1)$. Then, the time duration of the green traffic light on road p_1 is as follows:

$$T_g(p_1) = \frac{L_{p_1} \times (\tau(p_1, t+1) - \tau(p_2, t+1))}{V_f(p_1) \times \tau(p_1, t+1)}. \tag{5.8}$$

At the same time, the time duration of the red traffic light on road p_2 is set as $T_r(p_2) = T_g(p_1)$. To avoid the situations where traffic lights change too frequently or cars wait too long, the time duration needs to be bounded. In this study, if $T_g(p_1)$ is too small or too large, it is set as the default value $T_g(p_1) = 4$s or $T_g(p_1) = 90$s, respectively.

When a traffic light device is defined to control more than two traffic flows, the pheromone-based traffic light control can be easily extended to deal with this situation. For instance, suppose that roads p_1 and p_2 are two-way roads, and other two traffic flows are introduced in Fig. 5.2. One traffic flow is moving from west to east with digital pheromone $\tau(p_3, t+1)$, and the other is from south to north with digital pheromone $\tau(p_4, t+1)$. First, the competitive relationship between these four traffic flows are analyzed. The traffic flows of (east, west) contest with the traffic flows of (south, north). Second, the time duration of green traffic lights is calculated by Eq. 5.8. Suppose that $\tau(p_i, t+1)$ and $\tau(p_j, t+1)$ are the maximum and minimum digital pheromone on the four roads, respectively. At the same time, road p_i and road p_j must retain the competitive relationship. Then, road p_i and its non-competitive roads are assigned with the green traffic lights, and other roads are set as the red traffic lights. Their time durations are calculated as $T_g(p_i) = T_r(p_j) = \frac{L_{p_i} \times (\tau(p_i, t+1) - \tau(p_j, t+1))}{V_f(p_i) \times \tau(p_i, t+1)}$.

The pseudo-code of pheromone-based traffic light control is given in Algorithm 4. First, the digital pheromone of roads $\tau(p_i, t+1), i = 1, \ldots, n$ is predicted by Eq. 5.5 or Eq. 5.6 (Line 1). Then, the roads are divided into two group (p_I, p_J) according to their competitive relationship, where $I \bigcap J = \phi, I \bigcup J = \{1, \ldots, n\}$ (Line 2). After that, two roads with maximum digital pheromone in two groups are found as $\tau(p_{i1}, t+1) = \max_{i\in I} \tau(p_i, t+1)$, $\tau(p_{j1}, t+1) = \max_{j\in J} \tau(p_j, t+1)$ (Line 3). Line 4 finds the roads with minimum digital pheromone $\tau(p_{i2}, t+1) = \min_{i\in I} \tau(p_i, t+1)$, $\tau(p_{j2}, t+1) = \min_{j\in J} \tau(p_j, t+1)$. In Lines 5-8, if roads group p_I has the largest digital pheromone, these roads $p_i \in p_I$ will be assigned as green traffic lights with time $T_g(p_i) = \frac{L_{p_{i1}} \times (\tau(p_{i1},t+1) - \tau(p_{j2},t+1))}{V_f(p_{i1}) \times \tau(p_{i1},t+1)}$. Other roads $p_j \in p_J$ will be set as red traffic lights with time $T_r(p_j) = T_g(p_i)$. On the other hand, in Lines 9-12, if roads group p_J has the largest digital pheromone, these roads $p_j \in p_J$ will be assigned as green traffic lights with time $T_g(p_j) = \frac{L_{p_{j1}} \times (\tau(p_{j1},t+1) - \tau(p_{i2},t+1))}{V_f(p_{j1}) \times \tau(p_{j1},t+1)}$. The other roads $p_i \in p_I$ will be set as red traffic lights with time $T_r(p_i) = T_g(p_j)$.

Input : p_i, $i = 1, \ldots, n$, roads controlled by a traffic light agent;
Output: The optimal time of green and red traffic lights $T_g(p_i)$ and $T_r(p_i)$;

1 Calculate pheromone on roads $\tau(p_i, t+1), i = 1, \ldots, n$ by Eq. 5.5 or Eq. 5.6;
2 Divide roads into two groups (p_I, p_J) according to competitive relationship;
3 Finds roads with maximum pheromone (p_{i1}, p_{j1}) in two groups;
4 Finds roads with minimum pheromone (p_{i2}, p_{j2}) in two groups;
5 **if** $\tau(p_{i1}, t+1) > \tau(p_{j1}, t+1)$ **then**
6   Calculate the time of green traffic lights $T_g(p_{i1}) = \frac{L_{p_{i1}} \times (\tau(p_{i1},t+1) - \tau(p_{j2},t+1))}{V_f(p_{i1}) \times \tau(p_{i1},t+1)}$;
7   Set time of green traffic lights for roads $p_i \in p_I$ as $T_g(p_i) = T_g(p_{i1})$;
8   Set time of red traffic lights for roads $p_j \in p_J$ as $T_r(p_j) = T_g(p_{i1})$;
9 **else**
10   Calculate the time of green traffic lights $T_g(p_{j1}) = \frac{L_{p_{j1}} \times (\tau(p_{j1},t+1) - \tau(p_{i2},t+1))}{V_f(p_{j1}) \times \tau(p_{j1},t+1)}$;
11   Set time of red traffic lights for roads $p_i \in p_I$ as $T_r(p_i) = T_g(p_{j1})$;
12   Set time of green traffic lights for roads $p_j \in p_J$ as $T_g(p_j) = T_g(p_{j1})$;

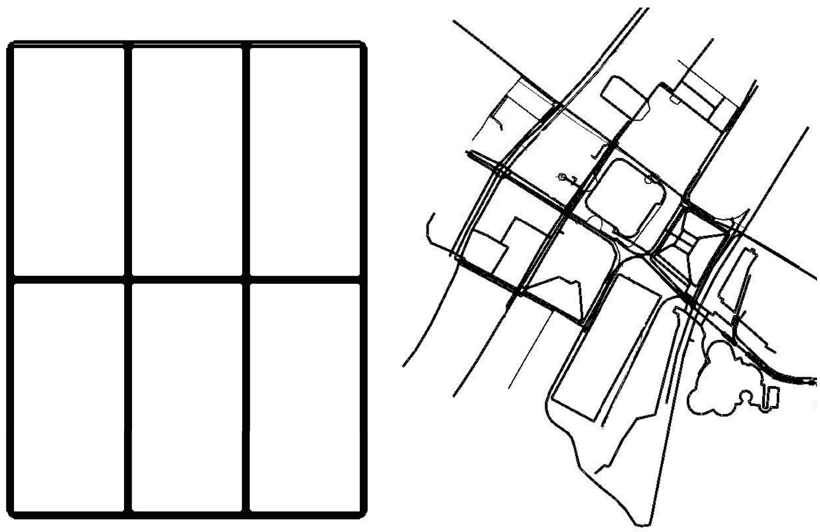
Algorithm 4: Pseudo-code of Pheromone-based traffic light control.

5.3 Experimentation

In this section, the proposed pheromone-based traffic management framework is compared with the existing approaches to show its advantages of alleviating road congestion, saving travel time, reducing air pollution and fuel consumption, and being robust to different levels of compliance rates and penetration rates.

5.3.1 Road Networks and Parameter Settings

The experiments are conducted on the simulation of urban mobility platform (SUMO) with version 0.18.0 [114], which is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks. The library of TraCI [190] provides extensive commands to control the behaviors of vehicle rerouting and manage traffic lights.



5.3.a: Grid 5.3.b: Cityhall

Figure 5.3: Two testing road networks.

The two testing road networks involved in the experiments are named as Grid and Cityhall. The Grid network is built by the abstract network generation tool in SUMO called "netgenerate" [114]. The Cityhall network is downloaded from OpenStreetMap[4]. The two maps and the properties of the two road networks are depicted in Fig. 5.3 and Table 5.1. Grid and Cityhall are designed to represent the highway and urban center traffic scenarios, respectively. Compared to Grid, vehicle rerouting and traffic

[4]http://www.openstreetmap.org/ with parameters (south, west, north, east) = (1.2904, 103.8505, 1.2963, 103.8560).

light control are more difficult in Cityhall since it involves complex roads and many traffic lights in a relatively small area.

In the simulation, the configurations of vehicles are set as the SUMO default settings: the length is 5m, the minimal gap is 2.5m, and the car following model is Krauss [114]. The number of vehicles is 2000, their trips are randomly generated, and their initialized routes are generated by the "dynamic user assignment" tool in SUMO ("duarouter") [114]. The congestion threshold is $\delta = 0.5$, the period for vehicle rerouting is 10s, the number of lanes in Grid is 2, and the maximum simulation time is 2000s. In k-shortest path, this study chooses the best one from 3 candidate routes. The parameters of ϵ-SVR in LIBSVM are set as: $C = 10$ and RBF kernel with $\gamma = 0.3$.

Table 5.1: The properties of testing road networks.

	Grid	Cityhall
Network Area	120,000m^2	65,300m^2
Number of Roads	34	507
Length of Roads	4,701m	69,453.8m
Number of Traffic Lights	16	30

To compare the performance of different traffic management models, five metrics are designed as follows: 1) the number of congested roads, $\tau_1(p,t) > \delta$; 2) the mean and standard deviation of traffic density, $\tau_1(p,t)$; 3) the number of vehicles that arrived their destinations; 4) the mean travel time of the arrived vehicles; and 5) air pollution CO_2 and fuel consumption. The first two metrics are used to assess the quality of road networks, the next two metrics are defined to evaluate driver experience, and the last one is introduced for testing environmental impact.

5.3.2 Effect of Congestion Threshold

In the proposed framework, the congestion threshold δ is designed to trigger the vehicle rerouting algorithm. To investigate the influence of this parameter, the proposed traffic management framework with $\delta \in [0.4, 1.0]$ is tested on the two road networks.

From Table 5.2, the proposed framework with congestion threshold $\delta \in [0.4, 0.6]$ obtained small mean traffic densities and large rerouting counts. When the congestion threshold is set as large values, the vehicle rerouting count becomes small. The extreme

Table 5.2: The proposed framework with different congestion thresholds $\delta \in [0.4, 1.0]$.

	Grid		Cityhall	
	Traffic Density	#Rerouting	Traffic Density	#Rerouting
$\delta = 0.4$	**0.097±0.089**	851	0.113±0.232	4003
$\delta = 0.6$	0.102±0.112	440	**0.099±0.223**	1828
$\delta = 0.8$	0.126±0.153	503	0.123±0.245	2347
$\delta = 1.0$	0.277±0.217	**0**	0.230±0.316	**0**

case of $\delta = 1.0$ means that all vehicles will adopt their initial routes (*i.e.*, #Rerouting = 0 in Table 5.2). On the other hand, the small congestion threshold will call the rerouting algorithm many times, which is able to evenly distribute vehicles onto different roads. In Table 5.2, when $\delta = 0.4$, the mean traffic density values are 0.097 and 0.113 on Grid and Cityhall, respectively. However, too many times of rerouting will disturb driving behaviors and enrage users. When $\delta = 0.4$, the rerouting counts are 851 and 4,003 on Grid and Cityhall, respectively. To balance the traffic density and the rerouting count, the congestion threshold in the proposed framework is set as $\delta = 0.5$.

5.3.3 Vehicle Rerouting with Various Weights

In vehicle rerouting algorithms, the basic requirement is to configure roads with weight information. In this section, the proposed pheromone-based vehicle rerouting algorithm is tested by configuring with four types of weights, including length of road, mean travel time, pheromone τ–Fusion by Eq. 5.5 and pheromone τ–SVR by Eq. 5.6.

Table 5.3: Results by vehicle rerouting with various weights.

	#Con. Roads	Traffic Density	#Arr. Vehs	Travel Time
Grid Network				
Length	21.2	0.660±0.197	502	307.0
Mean Travel Time	13.2	0.434±0.257	1246	248.3
Pheromone τ–Fusion	4.9	0.256±0.221	1800	170.4
Pheromone τ–SVR	**4.5**	**0.235±0.232**	**1801**	**165.2**
Cityhall Network				
Length	14.6	0.289±0.347	409	568.0
Mean Travel Time	10.4	0.224±0.332	564	463.1
Pheromone τ–Fusion	9.5	0.209±0.320	618	**446.1**
Pheromone τ–SVR	**9.1**	**0.200±0.324**	**625**	449.8

*Con. Roads: congested roads, Arr. Vehs: arrived vehicles

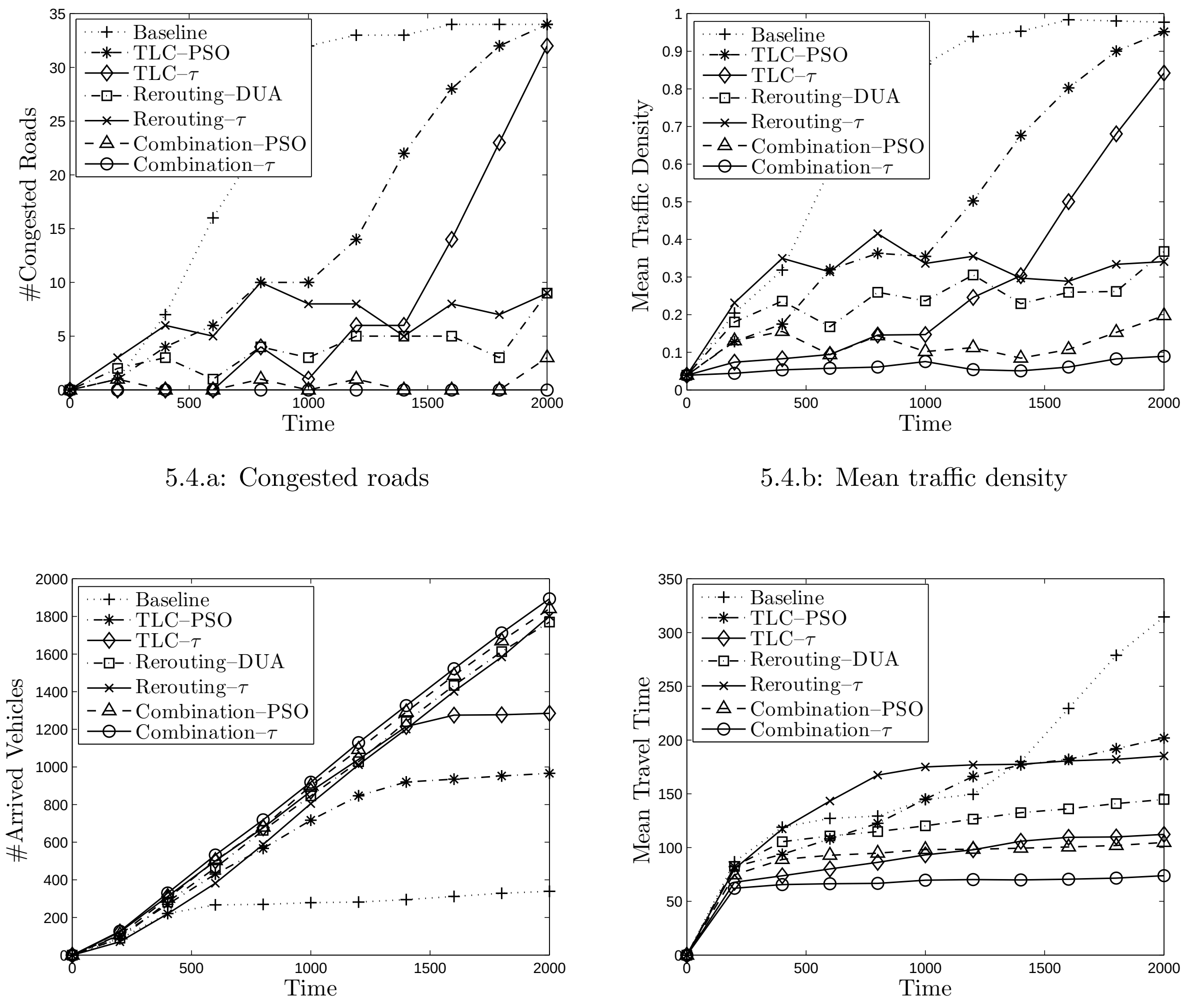
In Table 5.3, the two pheromone-based rerouting algorithms attained superior results in terms of the four metrics. On the other hand, the weight of mean travel time is better than the length information. Obviously, the length-based rerouting algorithm does not consider the dynamic future traffic conditions, which only chooses new routes for vehicles with the smallest distance. On the other hand, vehicle rerouting with travel time considers speed information. Suppose that the length of road p is L_p and the mean speed on road p in time interval $(t-1,t]$ is $\bar{V}(p,t)$. The travel time is then $\frac{L_p}{\bar{V}(p,t)}$. In addition, traffic pheromone τ_1 utilizes the traffic density [32, 34]. In comparison to the approaches that only use the current speed and traffic density information (*i.e.*, $\bar{V}(p,t)$ and $\tau_1(p,t)$ in $(t-1,t]$), the pheromone (*i.e.*, τ–Fusion and τ–SVR) is able to forecast traffic density by involving drivers' route intentions, which is $\tau_2(p,t+1)$ in $(t,t+1]$ by Eq. 5.2. The intuition behind the pheromone-based vehicle rerouting is to find alternative routes for cars with less congested roads (*i.e.*, small amount of pheromone).

Furthermore, τ–SVR is better than τ–Fusion, since SVR using Eq. 5.6 can obtain more precise prediction about the traffic condition than the simple fusion method of Eq. 5.5. In the following sections, τ–SVR is set as the default weight in the proposed traffic management framework.

From Table 5.3, the average number of congested roads by the rerouting algorithm based on pheromone τ–SVR are 4.5 and 9.1 on Grid and Cityhall, respectively. Due to the less congested traffic conditions, the arrived vehicles by the rerouting algorithm increase to 1,801 and 625 on Grid and Cityhall, respectively. In addition, the rerouting algorithm also reports better results in terms of the other two metrics (*i.e.*, the mean traffic density and the mean travel time), and very close to the best result for the mean travel time on Cityhall. These results in Table 5.3 indicate that the rerouting algorithm based on pheromone τ–SVR is more effective in alleviating traffic congestion than that using the length of road, the mean travel time and pheromone τ–Fusion.

5.3.4 Performance of the Proposed Framework

In this section, the effectiveness of the pheromone-based traffic management framework is verified on the two testing road networks. Four state-of-the-art algorithms are compared: "Baseline" without controlling routes and traffic signals; traffic light control by



5.4.a: Congested roads

5.4.b: Mean traffic density

5.4.c: Arrived vehicles

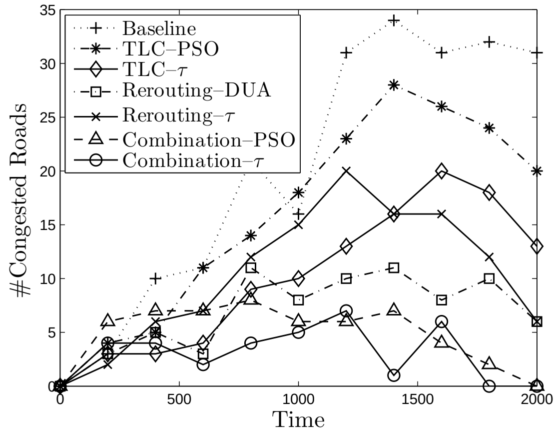
5.4.d: Mean travel time

Figure 5.4: Results on *Grid* network by vehicle rerouting and traffic light control (TLC).

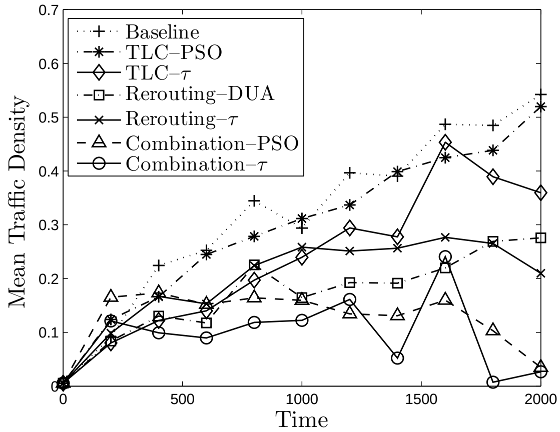
particle swarm optimization (TLC–PSO) [191]; "DUA" (dynamic user assignment)[5], by far the best rerouting algorithm to obtain the stochastic user-equilibrium (SUE) traffic states [35]; the combination of vehicle rerouting and traffic light control by PSO (Combination–PSO) [113].

Figs. 5.4-5.5 show the results of traffic management approaches on Grid and Cityhall, respectively. The two traffic light control methods give better results than Baseline in terms of the four metrics. In addition, pheromone-based traffic light control (TLC–τ)

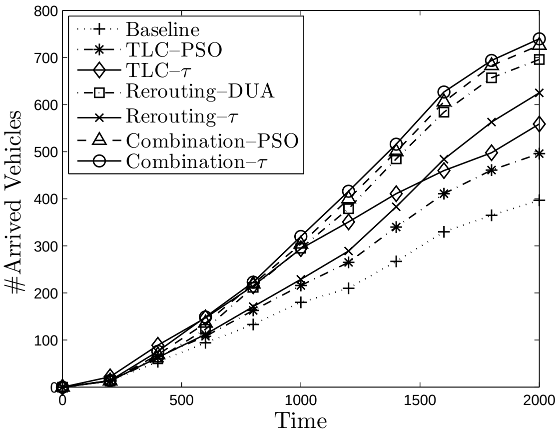
[5]http://sumo-sim.org/userdoc/Tools/Assign.html



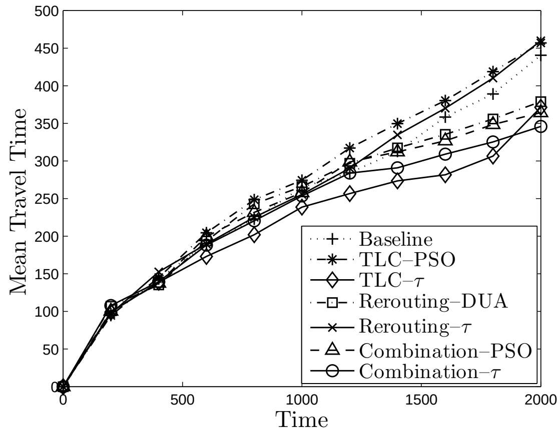
5.5.a: Congested roads



5.5.b: Mean traffic density



5.5.c: Arrived vehicles



5.5.d: Mean travel time

Figure 5.5: Results on *Cityhall* network by vehicle rerouting and traffic light control (TLC).

is superior to TCL-PSO. For instance, in Table 5.4, the average numbers of congested roads by TLC–PSO and TLC–τ are 14.0 and 7.2 on Grid, respectively. For Cityhall, these values by TLC–PSO and TLC–τ are 16.9 and 9.8, respectively. The numbers of arrived vehicles by TLC–PSO and TLC–τ are 966 and 1,285 on Grid, respectively. For Cityhall, these values by TLC–PSO and TLC–τ are 496 and 559, respectively. On the other hand, TLC–τ reports better results than TLC–PSO in terms of other two metrics (*i.e.*, the traffic density and the mean travel time). To explain, TLC–PSO only optimizes the time

Table 5.4: Results on two road networks by vehicle rerouting and traffic light control (TLC).

	#Con. Roads	Traffic Density	#Arr. Vehs	Travel Time
Grid Network				
Baseline	23.2	0.709±0.173	339	314.5
TLC–PSO	14.0	0.465±0.246	966	202.0
TLC–τ	7.2	0.277±0.217	1285	112.3
Rerouting–DUA	3.2	0.218±0.198	1770	144.7
Rerouting–τ	4.5	0.235±0.232	1801	165.2
Combination–PSO	0.9	0.142±0.146	1841	104.7
Combination–τ	**0.1**	**0.061±0.051**	**1893**	**73.9**
Cityhall Network				
Baseline	21.4	0.340±0.358	397	440.6
TLC–PSO	16.9	0.308±0.352	496	456.9
TLC–τ	9.8	0.230±0.316	559	371.5
Rerouting–DUA	7.3	0.179±0.296	696	378.8
Rerouting–τ	9.1	0.200±0.324	625	449.8
Combination–PSO	5.0	0.125±0.248	726	364.3
Combination–τ	**3.3**	**0.094±0.207**	**740**	**345.9**

*Con. Roads: congested roads, Arr. Vehs: arrived vehicles

duration of the predefined color phases [191]. On the contrary, TLC–τ dynamically sets color phases and time duration for roads according to the predicted traffic densities.

The two vehicle rerouting algorithms obtain better results than Baseline, TLC–PSO and TLC–τ. For instance, in Fig 5.4(a), TLC–τ, Rerouting–DUA and Rerouting–τ arrive at the similar number of congested roads before 1,500s. After that, the traffic volume becomes large and the traffic density increases. Only relying on traffic light control is insufficient for avoiding traffic congestion. Suppose that two competing roads have the same pheromone with a high traffic density value. TLC–τ allocates the same time duration of green traffic lights to the two roads using Eq. 5.8, and switches one road on and the other off. Thus, the strategies of traffic light control (*i.e.*, TLC–PSO and TLC–τ) are only suitable to unbalanced traffic scenarios. In comparison with traffic light control, the two rerouting algorithms (Rerouting–DUA and Rerouting–τ) search alternative routes for avoiding traffic congestion. From Fig 5.4(a), the number of congested roads by the two rerouting algorithms decreases significantly than TLC–τ after 1,500s. On the other hand, Rerouting–DUA is slightly better than Rerouting–τ in terms of the four metrics. For instance, in Table 5.4, the mean traffic densities by Rerouting–DUA and Rerouting–τ

are 0.218 and 0.235 on Grid, respectively. For Cityhall, those values by Rerouting–DUA and Rerouting–τ are 0.179 and 0.200, respectively. In essence, Rerouting–DUA is an offline algorithm, which evaluates traffic condition after vehicles have travelled through all candidate routes. Thus, it is only designed for simulation because vehicles' movement in realistic transportation systems is irrevocable. Although the proposed Rerouting–τ is designed as an online algorithm, it achieves the competitive results compared to Rerouting–DUA.

Furthermore, the two approaches combining vehicle rerouting and traffic light control produce better results than the above five algorithms. In addition, Combination–τ is superior to Combination–PSO, because Combination–PSO optimizes the two components (*i.e.*, vehicle rerouting and traffic light control) in a sequential order, which does not design its two components under a unified framework. In Table 5.4, road conditions become better with the proposed framework. The average numbers of congested roads on Grid and Cityhall are 0.1 and 3.3, respectively. The mean traffic densities on the two road networks reduce to 0.061 and 0.094, respectively. On the other hand, driver experience is also increased through the proposed framework. The numbers of arrived vehicles on Grid and Cityhall are 1,893 and 740, respectively. The mean travel time on two road networks decreases to 73.9s and 345.9s, respectively. These results in Table 5.4 indicate that the proposed traffic management framework (Combination–τ) is effective for improving the quality of road networks and saving travel time.

5.3.5 Robustness to Different Compliance Rates

In the proposed framework, the pheromone-based vehicle rerouting algorithm is able to provide alternative routes for drivers to avoid traffic congestion. In real world scenarios, not all drivers would like to follow rerouting guidance sent from the roadside infrastructure or the ITS server. In this section, the proposed rerouting algorithm (*i.e.*, Rerouting–τ) is tested by configuring with different compliance rates ($R_c \in [0, 1]$). The compliance rate means the percentage of drivers who will follow navigation guidance.

From the results on Grid in Table 5.5, Rerouting–τ with compliance rate $R_c \in [0.8, 1.0]$ only brings $6.7/34 = 19.7\%$ of edges to be congested and enables $1{,}619/2{,}000 = 81.0\%$ vehicles to arrive at their destinations. When $R_c = 0.8$, both mean traffic density and mean

Table 5.5: Results by pheromone-based vehicle rerouting with different compliance rates ($R_c \in [0, 1]$).

	#Con. Roads	Traffic Density	#Arr. Vehs	Travel Time
Grid Network				
$R_c = 0.0$	23.2	0.709±0.173	339	314.5
$R_c = 0.2$	21.8	0.666±0.183	477	283.3
$R_c = 0.4$	17.7	0.574±0.191	726	203.9
$R_c = 0.6$	11.8	0.423±0.231	1113	174.7
$R_c = 0.8$	6.7	0.251±0.243	1619	170.0
$R_c = 1.0$	**4.5**	**0.235±0.232**	**1801**	**165.2**
Cityhall Network				
$R_c = 0.0$	21.4	0.340±0.358	397	440.6
$R_c = 0.2$	20.9	0.331±0.351	429	458.1
$R_c = 0.4$	16.5	0.293±0.340	445	440.5
$R_c = 0.6$	15.0	0.274±0.341	479	**426.3**
$R_c = 0.8$	12.7	0.241±0.331	544	436.2
$R_c = 1.0$	**9.1**	**0.200±0.324**	**625**	449.8

*Con. Roads: congested roads, Arr. Vehs: arrived vehicles

travel time remain at low levels, which reach to $0.235/0.251 = 93.6\%$ and $165.2/170.0 = 97.2\%$ best performance (*i.e.*, $R_c = 1.0$), respectively.

From the results on Cityhall in Table 5.5, Rerouting–τ with compliance rate $R_c \in [0.6, 1.0]$ only causes $15.0/507 = 3.0\%$ congested roads. When $R_c = 0.6$, the arrived vehicles by Rerouting–τ is nearly 1.5 times (*i.e.*, $479/397 = 1.2$) as the result of non-compliance vehicles (*i.e.*, $R_c = 0.0$). In addition, Rerouting–τ with $R_c \in [0.6, 0.8]$ obtains competitive results in terms of the other two metrics (*i.e.*, the mean traffic density and the mean travel time). These results indicate that the pheromone-based vehicle rerouting algorithm is robust to the compliance rates of $[0.8, 1.0]$.

5.3.6 Robustness to Different Penetration Rates

In this section, the proposed traffic management framework is tested with different penetration rates ($R_p \in [0, 1]$). The penetration rate defines the percentage of drivers who will share their route information (intention pheromone τ_2 in Eq. 5.2) to the local roadside infrastructures. To avoid the influence of traffic light control, the pheromone-based vehicle rerouting (*i.e.*, Rerouting–τ) with various penetration rates is chosen to be tested on the two road networks.

Table 5.6: Results by pheromone-based vehicle rerouting with different penetration rates ($R_p \in [0, 1]$).

	#Con. Roads	Traffic Density	#Arr. Vehs	Travel Time
Grid Network				
$R_p = 0.0$	7.4	0.296±0.259	1793	209.0
$R_p = 0.2$	6.8	0.298±0.249	1766	201.3
$R_p = 0.4$	6.8	0.286±0.243	1762	194.6
$R_p = 0.6$	6.3	0.281±0.236	1769	188.5
$R_p = 0.8$	6.1	0.279±0.230	1800	187.6
$R_p = 1.0$	**4.5**	**0.235±0.232**	**1801**	**165.2**
Cityhall Network				
$R_p = 0.0$	11.6	0.220±0.332	596	465.2
$R_p = 0.2$	11.2	0.228±0.337	592	458.3
$R_p = 0.4$	10.7	**0.196±0.318**	597	450.5
$R_p = 0.6$	10.8	0.209±0.324	612	460.4
$R_p = 0.8$	10.6	0.212±0.322	618	466.2
$R_p = 1.0$	**9.1**	0.200±0.324	**625**	**449.8**

*Con. Roads: congested roads, Arr. Vehs: arrived vehicles

In Table 5.6, Rerouting–τ with penetration rate $R_p \in [0.6, 1.0]$ only brings $6.3/34 = 18.5\%$ and $10.8/507 = 2.1\%$ congested roads on Grid and Cityhall, respectively. When $R_p \in [0.6, 1.0]$, Rerouting–τ obtains competitive results in terms of the other three metrics (*i.e.*, the mean traffic density, the arrived vehicles and the mean travel time). These results indicate that the pheromone-based vehicle rerouting algorithm is robust to the penetration rates of $[0.6, 1.0]$.

Table 5.7: Results of air pollution and fuel consumption.

	Grid		Cityhall	
	CO_2 (mg)	Fuel (ml)	CO_2 (mg)	Fuel (ml)
Baseline	5.461e+008	2.177e+005	5.481e+008	2.185e+005
TLC–PSO	4.029e+008	1.606e+005	5.218e+008	2.080e+005
TLC–τ	3.994e+008	1.573e+005	5.159e+008	2.037e+005
Rerouting–DUA	3.339e+008	1.331e+005	4.546e+008	1.813e+005
Rerouting–τ	3.797e+008	1.514e+005	5.067e+008	2.020e+005
Combination–PSO	2.880e+008	1.148e+005	4.421e+008	1.762e+005
Combination–τ	**2.644e+008**	**1.054e+005**	**4.327e+008**	**1.725e+005**

5.3.7 Air Pollution and Fuel Consumption

Finally, the different traffic management approaches are tested and compared with respect to the air pollution (*i.e.*, CO_2) and fuel consumption.

From the results in Table 5.7, the two combination approaches (*i.e.*, Combination–PSO and Combination–τ) are superior to the other five algorithms. In addition, the proposed traffic management framework (*i.e.*, Combination–τ) achieves the best results. In particular, the CO_2 emission and fuel consumption by Combination–τ on Grid are 2.644e+8mg and 1.054e+5ml, respectively. These values are 4.327e+8mg and 1.725e+5ml on Cityhall, respectively. These results in Table 5.7 demonstrate that the pheromone-based traffic management framework is beneficial for reducing air pollution and saving energy consumption.

5.4 Summary

In this chapter, a novel pheromone-based multiagent traffic management framework is proposed for simultaneously rerouting vehicles and controlling traffic lights. The deliberately designed digital pheromone is introduced to predict traffic congestion. The pheromone-based vehicle rerouting is designed as a proactive solution, which searches new routes for vehicles before they enter congested roads. In addition, the pheromone-based traffic light control adopts the online strategy for allocating long time duration of green traffic lights to high traffic density roads. Experimental results confirm its superior performance on evenly distributing traffic volume and maximizing road capacities.

The proposed framework demonstrates the contributions to the third category of MAEC by effectively leveraging the advantages of both EC and MAS techniques and particularly embedding the concept of digital pheromone from the EC area into a multiagent framework for traffic management.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This section concludes the completed work so far in the research topic of *multiagent evolutionary computation* (MAEC). As a new paradigm emerging in recently years, multiagent evolutionary computation, which combines the advantages of *evolutionary computation* (EC) and *multiagent systems* (MAS), draws great attention from the two related domains. In this thesis, three novel approaches in MAEC have been proposed in Chapters 3-5 to respectively solve three types of complex problems, which are multiobjective optimization problems (MOPs), robustness of trust in e-marketplaces and traffic management in transportation systems. These three approaches respectively try to address some of the shortcomings of the existing approaches in each of the three categories of MACE: agent based EC, evolutionary computation based MAS, and the sequential or embedded approach of EC and MAS.

More specifically, a novel multiagent evolutionary framework based on trust is proposed in Chapter 3 for dealing with MOPs [3]. The purpose is to overcome the limitation of the first category of MAEC (*i.e.*, agent based EC), where agents are lack of abilities of simultaneously selecting evolutionary operators and control parameters. On the other hand, most studies in this category [72–87] focus only on single-objective optimization problems, whereas other complex problems (*e.g.*, MOPs) are not investigated. In this study, the trust concept, which is popular in MAS, is adopted into EC to estimate the dynamic competence of evolutionary operators and control parameters from generation to generation and on different problems. Every agent (*i.e.*, candidate solution) automatically selects operators and parameters based on the probabilities correlated to their competence. Experimental results on 58 benchmark MOPs demonstrate that the proposed framework significantly improves the performance of the state-of-the-art multiobjective evolutionary algorithms (MOEAs).

In Chapter 4, a new multiagent evolutionary trust model (MET) is designed to address the robustness problem in MAS-based e-marketplaces. The goal is to resolve the insufficiency of the second MAEC group (*i.e.*, evolutionary computation based MAS), where communications between agents are assumed to be truthful and the agents are benevolent. However, in some real world applications (*e.g.*, e-marketplaces), agents may provide untruthful information to others for maximizing their own interest or considering privacy issues. In this work, MET adopts evolutionary techniques into the design of an effective MAS, where buyers construct robust trust networks to accurate model seller reputation in the presence of unfair rating attacks from advisors. Experimental results on a multiagent-based e-marketplace testbed show that MET is more robust than the state-of-the-art trust models (*i.e.*, BRS [21, 106], iCLUB [18, 19], TRAVOS [20], Personalized approach [107, 110] and ReferralChain [22, 111, 112]).

In Chapter 5, a novel multiagent pheromone-based traffic management framework is proposed for reducing traffic congestion. The aim is to remedy the inadequacy of the third direction of MAEC (*i.e.*, the sequential or embedded approach of EC and MAS), where EC and MAS have not been effectively integrated. In particular, most studies in this category [32–34, 70] either optimize vehicle rerouting or traffic light control. On the other hand, the study in [113] sequentially optimizes the two components (*i.e.*, vehicle rerouting and traffic light control), which is easy to be trapped into local optimal. In this study, the concept of digital pheromone from the EC area is embedded into a multiagent-based traffic management model, which simultaneously optimizes vehicle rerouting and traffic light control under a unified framework. Experimental studies on SUMO [114] show that the proposed framework significantly alleviates traffic congestion and improves driver experience on two testing road networks.

To summarize, with the three kinds of effective MAEC algorithms for solving complex problems, this thesis serves to elicit more research efforts for building powerful MAEC approaches by delicately fusing the high level knowledge and advantages of EC and MAS, and applying them to solve large, open, dynamic, distributed complex problems. This work will also nourish the two related fields towards more intelligent EC and effective MAS.

6.2 Future Work

Based on the current studies, the future research will be conducted in the following three aspects.

6.2.1 Extensions for the Multiagent Evolutionary Framework

As stated in Chapter 3, agents in the multiagent evolutionary framework are benevolent and cooperative to solve a common problem (*e.g.*, MOPs). In essence, this framework is a centralized control system. All the agents report their true outcomes of services to a central solver agent. In future, the current work will be extended in the following five possible directions.

- **Distributed multiagent evolutionary framework**. In real world applications, many complex problems are modeling as large scale, open, dynamic and uncertain problems. For this type of problems, MAS often decomposes it to a number of multi-aspect problems that are addressed through the combination of a number of simple, interacting agents. Obviously, it is difficult to control and coordinate all the agents' behaviors by a single agent or a control center. Thus, a distributed multiagent evolutionary framework is urgently needed in order to solve complex problems (*e.g.*, MOPs). In this distributed framework, every agent should be able to collect its neighborhood agents' outcomes, analyze the competence of different services, and make decisions to select services by itself.

- **Incorporation of local search operators**. In the current work, evolutionary operators only include the three DE operators, SBX and polynomial mutation. Local search (LS) operators are not yet involved. In general, LS operators are considered as a kind of effective methods for refining solutions in their neighbor regions.

Inspired by studies in memetic computation (MC) [48, 49, 137–140], LS operators will be incorporated into the multiagent evolutionary framework, representing a promising direction for enhancing the search ability of EC approaches.

- **Credit assignment design**. The outcomes of services in the current work are from the quantity of generated non-dominated solutions. To measure the competency of operators and parameters more accurately, other *credit assignment* methods [133, 174], for example, hypervolume improvement, will be designed and explored.

- **Sequential order of evolutionary operators and control parameters**. The order of services (*i.e.*, the pairs of evolutionary operators and control parameters) is a key issue to generate high quality of agents (*i.e.*, solutions) in next generations. By considering a sequence of operators as an action, it is worthwhile to compare the existing adaption mechanisms with this kind of actions.

- **Fitness landscape consideration**. Most adaptive approaches consider rewards (*i.e.*, outcomes of evolutionary operators and control parameters) from all individual agents despite they are located in different searching spaces. Besides using the global information from the agent-based population, another interesting direction is to adopt surrogate-assisted approaches to help individual agents to select evolutionary operators and control parameters that are suitable to their local fitness landscapes [119, 121, 122, 192–195]. In future, I plan to depict trust scores regarding to fitness landscapes. Then, an agent can select operators and parameters by considering trust values in the fitness scenarios that are similar to its own fitness value.

- **Utilization of other techniques in MAS**. As stated in Chapter 3, this study incorporates the trust concept in MAS into evolutionary computation. In future, it will be interesting to investigate whether other fully developed techniques in MAS (*e.g.*, Q-learning, SARSA [142], FALCON [94, 145], etc.) are able to improve the effectiveness of evolutionary computation.

- **Other complex problems and real applications**. In future, the proposed framework will be verified by testing other complex problems, such as expensive problems, highly dynamic problems, constrained problems, etc [119, 121, 122]. Another interesting direction is to apply the current framework into practice, such as routing optimization in vehicle ad hoc networks (VANET) scenarios [196, 197].

6.2.2 Extensions for the Multiagent Evolutionary Trust Model

In Chapter 4, the novel algorithm MET is proposed to assist buyers for constructing trust networks. This work is designed as a distributed trust model where each buyer only keeps a fixed number of neighbor advisors into its trust network and communicates with them. In future, the current work will be extended in the following four potential directions.

- **Dealing with multi-dimensional multi-nominal unfair ratings**. In the trust research community, only few studies have been done to deal with multi-dimensional multi-nominal unfair ratings in e-marketplaces [27]. In practice, many e-marketplace platforms provide services consisting of a number of features that could be evaluated by buyers. For instance, a buyer gives multi-nominal ratings (*e.g.*, very bad, bad, neutral, good, excellent) to a smart phone in term of different features (*e.g.*, appearance, interface, running effectiveness, internet connection, power, compatibility, delivery time, maintenance). One future work is to deal with multi-dimensional multi-nominal unfair ratings in MAS-based e-marketplaces. In particular, firstly, a buyer evaluates the trustworthiness of advisors on different features. Then, only part of important features could be selected, and the correlation between different features will be considered. Finally, the trustworthiness of advisors is estimated by combing the selected multi-dimensional feature information.

- **Design of aggressive attack models**. Inspired by the study in [198], an interesting direction is to design more aggressive attack models with limited attacking resources, such as collusive attacks, intelligent attacks, etc. The robustness of existing trust models will be investigated under those powerful attack models.

- **Detection of the cheating behaviors from sellers**. As mentioned in Chapter 4, cheating behaviors from sellers are more difficult to detect than unfair ratings from advisors. In future, the robustness of trust models against cheating behaviors from sellers (*e.g.*, reputation lag, value imbalance, reentry, initial window [109, 199]) will be verified in complex testing simulations and real world datasets.

- **Coping with complex problems with dishonest information**. The multi-objective optimization problems (MOPs) in Chapter 3 are one type of complex problems without considering false information among agents' communications. On the other hand, in Chapter 4, robustness problems in e-marketplaces focus on solving untruthful information (*i.e.*, unfair ratings), which is considered as a single-objective optimization problem. Some real world problems (*e.g.*, route optimization and message propagation in VANETs) may exhibit the above two characteristics simultaneously. In particular, searing the best path involving multiple conflicting objectives, *e.g.*, minimizing the distance, avoiding traffic congestion, minimizing fuel consumption and saving travel time, is a typical MOP. On the other hand, to make a decision about whether to turn to a certain direction, a vehicle may rely on information providing by other vehicles (*e.g.*, accident messages, safety warnings, real-time traffic congestion and routing information). However, some vehicles may give false messages to avoid competition with other vehicles for limited road resources. To address this kind of problems, one possible future direction is to combine the first two ideas in this thesis (Chapters 3-4) by considering both the effective multiagent evolutionary framework for solving MOPs as well as the robust MET model in resisting dishonest information.

6.2.3 Extensions for the Pheromone-based Traffic Management Framework

In Chapter 5, the pheromone-based traffic management framework is proposed to simultaneously optimize vehicle rerouting and traffic light control. This study designs roadside infrastructures to collect digital pheromone deposited by surrounding cars. Thus, this work is suitable for urban areas where roadside infrastructures are fully installed and developed. In future, this work will be extended in the following directions.

- **Testing on complex road networks**. In future, the proposed framework will be verified by testing other complex road networks, which include a large number of cars and many traffic lights located in road intersections. On the other hand, the current work will be extended by considering realistic factors, including temporarily broken roadside infrastructures, fading communications by blocked buildings, weather conditions, etc [200–204]. In addition, this study can be extended and tested on rural areas, where only a few number of roadside infrastructures are installed and available [205].

- **Designing incentive mechanisms for attracting drivers to report real traffic information**. As pointed out in Chapter 5, not all the drivers will share their route information with other cars and roadside infrastructures. To address this problem, an incentive mechanism will be designed, where car agents will obtain some benefits when they share information (*e.g.*, routes, surrounding road conditions, weather information, restaurant issues, available parking lots, etc.) [23, 24, 37, 202, 206] with other cars.

- **Extending the current work to vehicular ad hoc networks (VANETs)**. In the current work, vehicle rerouting is implemented in roadside infrastructures. However, the cost for installing and maintaining a large amount of roadside infrastructures is high for governments. This problem becomes worse when it is applied to rural areas, where more electronic wires need to be installed for connecting roadside infrastructures in a large area. A possible solution is to distribute the computational cost of route optimization into cars, and exchange traffic condition by vehicle-to-vehicle (V2V) communications [200, 202–210].

Publication

(i) **Siwei Jiang**, Jie Zhang, Yew-Soon Ong, Liang Feng, " Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization", *IEEE Transactions on Cybernetics.* (Accepted, **IF: 3.236**)

(ii) **Siwei Jiang**, Jie Zhang, "A Simple and Fast Hypervolume Indicator-based Multiobjective Evolutionary Algorithm", *IEEE Transactions on Cybernetics.* (Under Review, IF: 3.236)

(iii) **Siwei Jiang**, Jie Zhang, Yew-Soon Ong, "Multiobjective Optimization Based on Reputation", *Information Sciences.* (Accepted, IF: 3.676)

(iv) **Siwei Jiang**, Jie Zhang, Yew-Soon Ong, "A Pheromone-based Traffic Management Model for Vehicle Re-routing and Traffic Light Control", *In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, France, 2014. (**AI top tier**)

(v) Athirai A. Irissappane, **Siwei Jiang**, Jie Zhang, "A Biclustering-based Approach to Filter Dishonest Advisors in Multi-criteria E-marketplaces", *In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, France, 2014. (**AI top tier**)

(vi) **Siwei Jiang**, "Towards the Design of Robust Trust and Reputation Systems", *In the Doctoral Consortium of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3225–3226, Beijing, 2013.

(vii) Athirai A. Irissappane, **Siwei Jiang**, Jie Zhang, " A Framework to Choose Trust Models for Different E-marketplace Environments", *In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 213–219, Beijing, 2013. (**AI top tier**)

(viii) **Siwei Jiang**, Jie Zhang, Yew-Soon Ong, "An Evolutionary Model for Constructing Robust Trust Networks", *In Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 813–820, America, 2013. (**AI top tier, Nomination of AAMAS 2013 Best Student Paper Award**)

(ix) Lizi Zhang, **Siwei Jiang**, Jie Zhang, Wee Keong Ng, " Robustness of Trust Models and Combinations for Handling Unfair Ratings", *In Proceedings of the IFIP WG 11.11 International Conference on Trust Management (IFIPTM)*, pp. 36–51, Indian, 2012.

(x) Chun-Wei Seah, Yew-Soon Ong, Ivor W. Tsang, **Siwei Jiang**, "Pareto Rank Learning in Multi-objective Evolutionary Algorithms", *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, Australian, 2012.

(xi) **Siwei Jiang**, Jie Zhang, Yew-Soon Ong, "A Multiagent Evolutionary Framework based on Trust for Multiobjective Optimization", *In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 299–306, Spain, 2012. (**AI top tier**)

(xii) **Siwei Jiang**, "A Multiagent Evolutionary Framework based on Trust for Multiobjective Optimization", *In the Doctoral Mentoring Program of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Spain, 2012.

(xiii) Athirai A. Irissappane, **Siwei Jiang**, Jie Zhang, "Towards a Comprehensive Testbed to Evaluate the Robustness of Reputation Systems against Unfair Rating Attack", *In Proceedings of the International Conference on User Modeling, Adaptation and Personalization (UMAP) Workshop on Trust, Reputation and User Modeling (TRUM)*, vol. 12, Canada, 2012.

(xiv) **Siwei Jiang**, Jie Zhang, Yew-Soon Ong, "Asymmetric Pareto-adaptive Scheme for Multiobjective Optimization", *In Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence (AI)*, pp. 351–360, Australian, 2011.

(xv) **Siwei Jiang**, Zhihua Cai, Jie Zhang, Yew-Soon Ong, "Multiobjective Optimization by Decomposition with Pareto-adaptive Weight Vectors", *In Proceedings of the 7th International Conference on Natural Computation (ICNC)*, pp. 1260–1264, Shanghai, China, 2011.

References

[1] J. Zhang, Z. Zhan, Y. Lin, N. Chen, Y. Gong, J. Zhong, H. Chung, Y. Li, and Y. Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 68–75, 2011.

[2] C. A. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[3] S. Jiang, J. Zhang, and Y. Ong, "A multiagent evolutionary framework based on trust for multiobjective optimization," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 299–306, 2012.

[4] S. Jiang, J. Zhang, and Y. Ong, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, 2014.

[5] S. Jiang, J. Zhang, and Y. Ong, "Asymmetric pareto-adaptive scheme for multiobjective optimization," in *Proceedings of Australasian Joint Conference on Artificial Intelligence (AI)*, pp. 351–360, Springer, 2011.

[6] S. Jiang, J. Zhang, and Y. Ong, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *Proceedings of International Conference on Natural Computation (ICNC)*, vol. 3, pp. 1260–1264, IEEE, 2011.

[7] I. W. T. Chun-Wei Seah, Yew-Soon Ong and S. Jiang, "Pareto rank learning in multi-objective evolutionary algorithms," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2012.

[8] A. Zhou, B. Qu, H. Li, S. Zhao, P. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state-of-the-art," *Swarm and Evolutionary Computation*, 2011.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proceedings of Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN)*, pp. 95–100, 2001.

[11] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[12] J. Durillo, A. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2010.

[13] C. Chen, Y. Chen, T. Shen, and J. K. Zao, "Optimizing degree distributions in LT codes by using the multiobjective evolutionary algorithm based on decomposition," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2010.

[14] Y. Chen and C. Chen, "Enabling the extended compact genetic algorithm for real-parameter optimization by using adaptive discretization," *Evolutionary Computation*, vol. 18, no. 2, pp. 199–228, 2010.

[15] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 442–446, 2012.

[16] B.-Y. Qu and P. N. Suganthan, “Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection,” *Information sciences*, vol. 180, no. 17, pp. 3170–3181, 2010.

[17] S. Jiang, J. Zhang, and Y.-S. Ong, “An evolutionary model for constructing robust trust networks,” in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 813–820, 2013.

[18] S. Liu, J. Zhang, C. Miao, Y. Theng, and A. Kot, “iCLUB: an integrated clustering-based approach to improve the robustness of reputation systems,” in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 3, pp. 1151–1152, 2011.

[19] S. Liu, J. Zhang, C. Miao, Y.-L. Theng, and A. C. Kot, “An integrated clustering-based approach to filtering unfair multi-nominal testimonies,” *Computational Intelligence*, 2012.

[20] W. Teacy, J. Patel, N. Jennings, and M. Luck, “TRAVOS: Trust and reputation in the context of inaccurate information sources,” *Autonomous Agents and Multi-Agent Systems*, vol. 12, no. 2, pp. 183–198, 2006.

[21] A. Whitby, A. Josang, and J. Indulska, “Filtering out unfair ratings in bayesian reputation systems,” in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems Workshop on Trust in Agent Societies (AAMAS)*, 2004.

[22] B. Yu and M. Singh, “Detecting deception in reputation management,” in *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems(AAMAS)*, pp. 73–80, ACM, 2003.

[23] J. Zhang, *Promoting Honesty in E-marketplacesarketplaces: Combining Trust Modeling and Incentive Mechanism Design*. PhD thesis, University of Waterloo, 2009.

[24] J. Zhang and R. Cohen, “Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings,” in *Proceedings of International Conference on Electronic Commerce (ICEC)*, pp. 225–234, ACM, 2006.

[25] A. Jøsang, "Robustness of trust and reputation systems: Does it matter?," in *Proceedings of IFIP International Conference on Trust Management (IFIPTM)*, pp. 253–262, 2012.

[26] L. Zhang, S. Jiang, J. Zhang, and W. Ng, "Robustness of trust models and combinations for handling unfair ratings," in *Proceedings of IFIP International Conference on Trust Management (IFIPTM)*, vol. 374, pp. 36–51, 2012.

[27] A. A. Irissappane, S. Jiang, and J. Zhang, "A biclustering-based approach to filter dishonest advisors in multi-criteria e-marketplaces," in *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems(AAMAS)*, ACM, 2014.

[28] S. Jiang, "Towards the design of robust trust and reputation systems," in *the Doctoral Consortium of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3225–3226, AAAI Press, 2013.

[29] A. A. Irissappane, S. Jiang, and J. Zhang, "A framework to choose trust models for different e-marketplace environments," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 213–219, AAAI Press, 2013.

[30] A. A. Irissappane, S. Jiang, and J. Zhang, "Towards a comprehensive testbed to evaluate the robustness of reputation systems against unfair rating attack.," in *Proceedings of the International Conference on User Modeling, Adaptation and Personalization (UMAP) Workshop on Trust, Reputation and User Modeling (TRUM)*, vol. 12, 2012.

[31] S. Jiang, J. Zhang, and Y. Ong, "A pheromone-based traffic management model for vehicle re-routing and traffic light control," in *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems(AAMAS)*, ACM, 2014.

[32] Y. Ando, Y. Fukazawa, O. Masutani, H. Iwasaki, and S. Honiden, "Performance of pheromone model for predicting traffic congestion," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 73–80, 2006.

[33] S. Kurihara, "Traffic-congestion forecasting algorithm based on pheromone communication model," *Ant Colony Optimization – Techniques and Applications*, pp. 163–176, 2013.

[34] O. Masutani, H. Sasaki, H. Iwasaki, Y. Ando, Y. Fukazawa, and S. Honiden, "Pheromone model: application to traffic congestion prediction," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1171–1172, ACM, 2005.

[35] J. Pan, I. Sandu Popa, K. Zeitouni, and C. Borcea, "Proactive vehicular traffic rerouting for lower travel time," *IEEE Transactions on Vehicular Technology*, 2013.

[36] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 530–537, 2004.

[37] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 617–624, ACM, 2009.

[38] T. Yamashita, K. Izumi, K. Kurumatani, and H. Nakashima, "Smooth traffic flow with a cooperative car navigation system," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 478–485, 2005.

[39] J. Sánchez, M. Galán, and E. Rubio, "Applying a traffic lights evolutionary optimization technique to a real case: "las ramblas" area in santa cruz de tenerife," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 25–40, 2008.

[40] M. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proceedings of International Conference Maching Learning (ICML)*, pp. 1151–1158, 2000.

[41] L. Wu, X. Xiao, D. Deng, G. Cong, A. D. Zhu, and S. Zhou, "Shortest path and distance queries on road networks: an experimental evaluation," in *Proceedings of*

International Conference on Very Large Data Bases (VLDB), vol. 5, pp. 406–417, 2012.

[42] G. Marfia and M. Roccetti, "Vehicular congestion detection and short-term forecasting: a new model with results," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 2936–2948, 2011.

[43] M. Smith and R. Mounce, "A splitting rate model of traffic re-routeing and traffic control," *Procedia-Social and Behavioral Sciences*, vol. 17, pp. 316–340, 2011.

[44] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers routing," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 329–342, 2004.

[45] "http://en.wikipedia.org/wiki/A.I._Artificial_Intelligence,"

[46] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.

[47] K. Fang and C. Ma, *Orthogonal and uniform design*. Science Press, 2001.

[48] Y. Ong, M. Lim, N. Zhu, and K. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 1, pp. 141–152, 2006.

[49] X. Chen, Y. Ong, M. Lim, and K. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.

[50] K. Sycara, "Multiagent systems," *Artificial Intelligence Magazine*, vol. 19, no. 2, p. 79, 1998.

[51] D. Wilkie, J. P. van den Berg, M. C. Lin, and D. Manocha, "Self-aware traffic route planning.," in *Proceedings of Conference on Artificial Intelligence (AAAI)*, 2011.

[52] T.-C. Au, N. Shahidi, and P. Stone, "Enforcing liveness in autonomous traffic management.," in *Proceedings of Conference on Artificial Intelligence (AAAI)*, 2011.

[53] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication," in *Proceedings of Vehicular Technology Conference (VTC)*, pp. 21–25, IEEE, 2007.

[54] W. Kim and M. Gerla, "NAVOPT: navigator assisted vehicular route optimizer," in *Proceedings of International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pp. 450–455, IEEE, 2011.

[55] V. T. N. Nha, S. Djahel, and J. Murphy, "A comparative study of vehicles' routing algorithms for route planning in smart cities," in *Proceedings of International Workshop on Vehicular Traffic Management for Smart Cities (VTM)*, pp. 1–6, IEEE, 2012.

[56] K. Tumer and A. K. Agogino, "Adaptive management of air traffic flow: A multiagent coordination approach.," in *Proceedings of Conference on Artificial Intelligence (AAAI)*, pp. 1581–1584, 2008.

[57] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multiagent systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 364–373, 2011.

[58] K. Tumer, Z. T. Welch, and A. Agogino, "Aligning social welfare and agent preferences to alleviate traffic congestion," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 655–662, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[59] J. Takahashi, R. Kanamori, and T. Ito, "A preliminary study on anticipatory stigmergy for traffic management," in *Proceedings of Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 3, pp. 399–405, IEEE, 2012.

[60] A. L. Bazzan and R. Junges, "Congestion tolls as utility alignment between agent and system optimum," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 126–128, ACM, 2006.

[61] C. Sommer, R. Krul, R. German, and F. Dressler, "Emissions vs. travel time: simulative evaluation of the environmental impact of its," in *Proceedings of Vehicular Technology Conference (VTC)*, pp. 1–5, IEEE, 2010.

[62] R. Kohout, K. Erol, and C. Robert, "In-time agent-based vehicle routing with a stochastic improvement heuristic," in *Proceedings of Conference on Artificial Intelligence (AAAI)*, pp. 864–869, 1999.

[63] S. Salcedo-Sanz, D. Manjarres, Á. Pastor-Sánchez, J. Del Ser, J. A. Portilla-Figueras, and S. Gil-Lopez, "One-way urban traffic reconfiguration using a multi-objective harmony search approach," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3341–3350, 2013.

[64] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz, "Self-organized traffic control," in *Proceedings of International Workshop on VehiculAr Inter-NETworking (VANET)*, pp. 85–90, ACM, 2010.

[65] F. Angius, M. Reineri, C. Chiasserini, M. Gerla, and G. Pau, "Towards a realistic optimization of urban traffic flows," in *Proceedings of International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1661–1668, IEEE, 2012.

[66] J. A. Quinn and R. Nakibuule, "Traffic flow monitoring in crowded cities.," in *Proceedings of AAAI Spring Symposium: Artificial Intelligence for Development*, 2010.

[67] M. Gramaglia, C. J. Bernardos, and M. Calderon, "Virtual induction loops based on cooperative vehicular communications," *Sensors*, vol. 13, no. 2, pp. 1467–1476, 2013.

[68] Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, "Advanced intersection management for connected vehicles using a multi-agent systems approach," in *Proceedings of Intelligent Vehicles Symposium (IV)*, pp. 932–937, IEEE, 2012.

[69] X.-F. Xie, S. F. Smith, L. Lu, and G. J. Barlow, "Schedule-driven intersection control," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 168–189, 2012.

[70] K. Tawara and N. Mukai, "Traffic signal control by using traffic congestion prediction based on pheromone model," in *Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 1, pp. 27–30, IEEE, 2010.

[71] R. A. Sarker and T. Ray, *Agent-basedased Evolutionary Search.* Springer, 2010.

[72] W. Zhong, J. Liu, M. Xue, and L. Jiao, "A multiagent genetic algorithm for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 2, pp. 1128–1141, 2004.

[73] B. Zhao, C. Guo, and Y. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1070–1078, 2005.

[74] H. Wang, S. Kwong, Y. Jin, W. Wei, and K. Man, "Agent-based evolutionary approach for interpretable rule-based knowledge extraction," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 143–155, 2005.

[75] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for constraint satisfaction problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 1, pp. 54–73, 2006.

[76] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for combinatorial optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 1, pp. 229–240, 2010.

[77] Y. Yan, S. Yang, D. Wang, and D. Wang, "Agent based evolutionary dynamic optimization," *Agent Based Evolutionary Search*, pp. 97–116, 2010.

[78] A. Barkat Ullah, R. Sarker, D. Cornforth, and C. Lokan, "AMA: a new approach for solving constrained real-valued optimization problems," *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 8, pp. 741–762, 2009.

[79] A. Barkat Ullah, R. Sarker, and C. Lokan, "Handling equality constraints with agent-based memetic algorithms," *Memetic Computing*, vol. 3, no. 1, pp. 51–72, 2011.

[80] J. Liu, Y. Tang, and Y. Cao, "An evolutionary autonomous agents approach to image feature extraction," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 141–158, 1997.

[81] J. Liu, *Autonomous agents and multi-agent systems: explorations in learning, self-organization, and adaptive computation.* World Scientific Pub Co Inc, 2001.

[82] J. Liu, H. Jing, and Y. Tang, "Multi-agent oriented constraint satisfaction," *Artificial Intelligence*, vol. 136, no. 1, pp. 101–144, 2002.

[83] K. Socha and M. Kisiel-Dorohinicki, "Agent-based evolutionary multiobjective optimisation," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, pp. 109–114, 2002.

[84] A. Eiben, M. Horvath, W. Kowalczyk, and M. Schut, "Reinforcement learning for online control of evolutionary algorithms," *Engineering Self-Organising Systems*, pp. 151–160, 2007.

[85] S. Muller, N. Schraudolph, and P. Koumoutsakos, "Step size adaptation in evolution strategies using reinforcement learning," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, pp. 151–156, 2002.

[86] H. Zhang and J. Lu, "Adaptive evolutionary programming based on reinforcement learning," *Information Sciences*, vol. 178, no. 4, pp. 971–984, 2008.

[87] J. Maturana, Á. Fialho, F. Saubion, M. Schoenauer, and M. Sebag, "Extreme compass and dynamic multi-armed bandits for adaptive operator selection," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 365–372, 2009.

[88] S. Whiteson and P. Stone, "Evolutionary function approximation for reinforcement learning," *The Journal of Machine Learning Research*, vol. 7, pp. 877–917, 2006.

[89] M. Taylor, S. Whiteson, and P. Stone, "Comparing evolutionary and temporal difference methods in a reinforcement learning domain," in *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 1321–1328, ACM, 2006.

[90] S. Whiteson, M. Taylor, and P. Stone, "Critical factors in the empirical performance of temporal difference and evolutionary methods for reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 1, pp. 1–35, 2010.

[91] X. Yao and P. Darwen, "How important is your reputation in a multi-agent environment," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 575–580, 1999.

[92] S. Chong and X. Yao, "Multiple choices and reputation in multiagent interactions," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 689–711, 2007.

[93] J. Pugh and A. Martinoli, "Multi-robot learning with particle swarm optimization," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 441–448, ACM, 2006.

[94] L. Feng, Y. Ong, A. Tan, and X. Chen, "Towards human-like social multi-agents with memetic automaton," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1092–1099, IEEE, 2011.

[95] F. Gómez Mármol and G. Martínez Pérez, "Providing trust in wireless sensor networks using a bio-inspired technique," *Telecommunication Systems*, vol. 46, no. 2, pp. 163–180, 2011.

[96] M. Mejia, N. Peña, J. Muñoz, O. Esparza, and M. Alzate, "A game theoretic trust model for on-line distributed evolution of cooperation in manets," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 39–51, 2011.

[97] W. Abdou, A. Henriet, D. Dhoutaut, F. Spies, and C. Bloch, "Optimizing communications in vehicular ad hoc networks using evolutionary computation and simulation," in *Proceedings of International Conference on Soft computing as Transdisciplinary Science and Technology*, pp. 158–165, ACM, 2008.

[98] Q. Li, D. Qu, and L. Du, "Research on hybrid-genetic algorithm for mas based job-shop dynamic scheduling," in *Proceedings of IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, vol. 2, pp. 1742–1745, 2008.

[99] G. Giardini and T. Kalmar-Nagy, "Genetic algorithm for multi-agent space exploration," in *Proceedings of AIAA InfoTech at Aerospace Conference*, vol. 2, pp. 1146–1160, 2007.

[100] H. Liu and M. Tang, "Evolutionary design in a multi-agent design environment," *Applied Soft Computing*, vol. 6, no. 2, pp. 207–220, 2006.

[101] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[102] A. Hernández-Díaz, L. Santana-Quintero, C. Coello Coello, and J. Molina, "Pareto-adaptive ε-dominance," *Evolutionary Computation*, vol. 15, no. 4, pp. 493–517, 2007.

[103] S. Jiang and Z. Cai, "Faster convergence and higher hypervolume for multi-objective evolutionary algorithms by orthogonal and uniform design," *Advances in Computation and Intelligence*, pp. 312–328, 2010.

[104] S. Jiang, Z. Cai, D. Zeng, Q. Li, and Y. Cheng, "Parallel gene expression programming algorithm based on simulated annealing method," *Journal of Electronics*, vol. 33, no. 11, pp. 2017–2021, 2005.

[105] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[106] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of Bled Electronic Commerce Conference*, pp. 41–55, 2002.

[107] J. Zhang and R. Cohen, “A comprehensive approach for sharing semantic web trust ratings,” *Computational Intelligence*, vol. 23, no. 3, pp. 302–319, 2007.

[108] M. Şensoy, J. Zhang, P. Yolum, and R. Cohen, “POYRAZ: Context-aware service selection under deception,” *Computational Intelligence*, vol. 25, no. 4, pp. 335–366, 2009.

[109] A. Jøsang and J. Golbeck, “Challenges for robust of trust and reputation systems,” in *Proceedings of International Workshop on Security and Trust Management (SMT)*, 2009.

[110] J. Zhang and R. Cohen, “Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach,” *Electronic Commerce Research and Applications*, vol. 7, no. 3, pp. 330–340, 2008.

[111] B. Yu and M. Singh, “An evidential model of distributed reputation management,” in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 294–301, ACM, 2002.

[112] B. Yu and M. Singh, “Distributed reputation management for electronic commerce,” *Computational Intelligence*, vol. 18, no. 4, pp. 535–549, 2002.

[113] A. L. Bazzan, D. De Oliveira, F. Klügl, and K. Nagel, “To adapt or not to adapt–consequences of adapting driver and traffic light agents,” in *Proceedings of Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pp. 1–14, Springer, 2008.

[114] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO – simulation of urban mobility: an overview,” in *Proceedings of International Conference on Advances in System Simulation (SIMUL)*, pp. 55–60, 2011.

[115] “`http://www.cse.dmu.ac.uk/~rij/gafaq/Q2.htm`,”

[116] T. Bäck, “Evolutionary computation: toward a new philosophy of machine intelligence,” *Complexity*, vol. 2, no. 4, pp. 28–30, 1997.

[117] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (Studies in Computational Intelligence).* Springer-Verlag New York, Inc., 2006.

[118] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[119] Y. Ong, P. Nair, and A. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *The American Institute of Aeronautics and Astronautics (AIAA)*, vol. 41, no. 4, pp. 687–696, 2003.

[120] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[121] Y. Ong, P. Nair, and K. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 392–404, 2006.

[122] D. Lim, Y. Jin, Y. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[123] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[124] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[125] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, vol. 4, pp. 1942–1948, 1995.

[126] Y. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41–53, 2001.

[127] Z. Zhan, J. Zhang, and O. Liu, "Orthogonal learning particle swarm optimization," in *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 1763–1764, ACM, 2009.

[128] S. Rahnamayan, H. Tizhoosh, and M. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[129] N. Hansen, S. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[130] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.

[131] Y. Xiaotian, W. Muqing, and S. Bing, "An adaptive LS-SVM based differential evolution algorithm," in *Proceedings of IEEE International Conference on Signal Processing Systems*, pp. 406–409, 2009.

[132] W. Gong, Á. Fialho, and Z. Cai, "Adaptive strategy selection in differential evolution," in *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 409–416, ACM, 2010.

[133] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[134] J. Zhang, H. Chung, and W. Lo, "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 326–335, 2007.

[135] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[136] G. Magyar, M. Johnsson, and O. Nevalainen, "An adaptive hybrid genetic algorithm for the three-matching problem," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 135–146, 2000.

[137] Y. Ong and A. Keane, "Meta-lamarckian learning in memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.

[138] Q. Nguyen, Y. Ong, and M. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.

[139] Y. Ong, M. Lim, and X. Chen, "Research frontier: memetic computation-past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.

[140] G. Acampora, V. Loia, and M. Gaeta, "Exploring e-learning knowledge through ontological memetic agents," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 66–77, 2010.

[141] J. Smith, "Coevolving memetic algorithms: a review and progress report," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 6–17, 2007.

[142] R. Sutton and A. Barto, *Reinforcement learning: An introduction*, vol. 116. Cambridge Univ Press, 1998.

[143] M. Wooldridge, *An introduction to multiagent systems*. Wiley, 2009.

[144] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice hall, 2010.

[145] A. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 230–244, 2008.

[146] B. Liu, T. Duan, and Y. Li, "One improved agent genetic algorithm–ring-like agent algorithm for global numerical optimization," *Asia Pacific Journal of Operational Research*, vol. 26, no. 4, p. 479, 2009.

[147] D. Lee, L. Gonzalez, J. Periaux, and K. Srinivas, "Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 133–150, 2011.

[148] U. Nehmzow, "Learning in multi-robot scenarios through physically embedded genetic algorithms," in *Proceedings of International Conference on The Simulation of Adaptive Behavior: From animals to animats*, pp. 391–392, 2002.

[149] W. Du and B. Li, "Multi-strategy ensemble particle swarm optimization for dynamic optimization," *Information Sciences*, vol. 178, no. 15, pp. 3096–3109, 2008.

[150] R. Mallipeddi, S. Mallipeddi, and P. N. Suganthan, "Ensemble strategies with adaptive evolutionary programming," *Information Sciences*, vol. 180, no. 9, pp. 1571–1581, 2010.

[151] C. Chen, Y. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 209–216, 2009.

[152] F. Caraffini, F. Neri, J. Cheng, G. Zhang, L. Picinali, G. Iacca, and E. Mininno, "Super-fit multicriteria adaptive differential evolution," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1678–1685, IEEE, 2013.

[153] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.

[154] J. Schaffer, R. Caruana, L. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proceedings of Conference Genetic Algorithms*, pp. 51–60, 1989.

[155] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[156] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 1187–1194, 2007.

[157] J. Brest, V. Zumer, and M. Maucec, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 215–222, Ieee, 2006.

[158] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 19–31, 2011.

[159] S. K. Smit and A. Eiben, "Beating the 'world champion' evolutionary algorithm via REVAC tuning," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, 2010.

[160] A. Ghosh, S. Das, A. Chowdhury, and R. Giri, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," *Information Sciences*, vol. 181, no. 18, pp. 3749–3765, 2011.

[161] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Information Sciences*, vol. 181, no. 20, pp. 4515–4538, 2011.

[162] A. Caponio and F. Neri, *Integrating cross-dominance adaptation in multi-objective memetic algorithms*. Springer, 2009.

[163] F. Neri, J. Toivanen, G. L. Cascella, and Y.-S. Ong, "An adaptive multimeme algorithm for designing HIV multidrug therapies," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 264–278, 2007.

[164] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A fast adaptive memetic algorithm for online and offline control design of pmsm drives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 28–41, 2007.

[165] R. Hinterding, Z. Michalewicz, and A. Eiben, "Adaptation in evolutionary computation: A survey," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 65–69, 1997.

[166] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[167] A. Iorio and X. Li, "Rotationally invariant crossover operators in evolutionary multi-objective optimization," *Simulated Evolution and Learning*, pp. 310–317, 2006.

[168] "http://users.jyu.fi/ miettine/kurssit/jatkoksem/karthik080310.pdf,"

[169] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[170] V. Huang, A. Qin, P. Suganthan, and M. Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution algorithm," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 3601–3608, 2007.

[171] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[172] M. Hamdan, "On the disruption-level of polynomial mutation for evolutionary multi-objective optimisation algorithms," *Computing and Informatics*, vol. 29, no. 5, pp. 783–800, 2012.

[173] J. Zhang and A. Sanderson, "Self-adaptive multi-objective differential evolution with direction information provided by archived inferior solutions," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 2801–2810, 2008.

[174] Y. Wang, L. Wu, and X. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 3, pp. 193–209, 2010.

[175] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss, "The sequential parameter optimization toolbox," in *Proceddings of Experimental methods for the analysis of optimization algorithms*, pp. 337–362, Springer, 2010.

[176] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. Murphy, "Time-bounded sequential parameter optimization," in *Proceddings of Learning and Intelligent Optimization*, pp. 281–298, Springer, 2010.

[177] M. Preuss and T. Bartz-Beielstein, "Sequential parameter optimization applied to self-adaptation for binary-coded evolutionary algorithms," in *Proceddings of Parameter setting in evolutionary algorithms*, pp. 91–119, Springer, 2007.

[178] T. Bartz-Beielstein, C. W. Lasarczyk, and M. Preuß, "Sequential parameter optimization," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, pp. 773–780, 2005.

[179] H. Abbass, R. Sarker, and C. Newton, "PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, pp. 971–978, 2001.

[180] K. Tan, C. Goh, Y. Yang, and T. Lee, "Evolving better population distribution and exploration in evolutionary multi-objective optimization," *European Journal of Operational Research*, vol. 171, no. 2, pp. 463–495, 2006.

[181] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[182] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[183] A. A. Irissappane and J. Zhang, "A testbed to evaluate the robustness of reputation systems in e-marketplaces," in *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems(AAMAS)*, ACM, 2014.

[184] W. Wen, "A dynamic and automatic traffic light control expert system for solving the road congestion problem," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2370–2381, 2008.

[185] Y. Xu, Y. Wu, J. Xu, D. Ni, G. Wu, and L. Sun, "A queue-length-based detection scheme for urban traffic congestion by VANETs," in *Proceedings of International Conference on Networking, Architecture and Storage (NAS)*, pp. 252–259, IEEE, 2012.

[186] H. Arbabi and M. C. Weigle, "Monitoring free flow traffic using vehicular networks," in *Proceedings of Consumer Communications and Networking Conference (CCNC)*, pp. 272–2760, IEEE, 2011.

[187] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[188] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[189] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[190] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: an interface for coupling road traffic and network simulators," in *Proceedings of Communications and Networking Simulation Symposium (CNS)*, pp. 155–163, 2008.

[191] García-Nieto, José, E. Alba, and A. Carolina Olivera, "Swarm intelligence for traffic light scheduling: Application to real urban areas," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 274–283, 2012.

[192] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 1, pp. 66–76, 2007.

[193] Y. Ong, K.-Y. Lum, P. Nair, D. Shi, and Z. Zhang, "Global convergence of unconstrained and bound constrained surrogate-assisted evolutionary search in aerodynamic shape design," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 3, pp. 1856–1863, 2003.

[194] Y. S. Ong, P. Nair, A. Keane, and K. Wong, "Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems," in *Knowledge Incorporation in Evolutionary Computation*, pp. 307–331, Springer, 2005.

[195] Y. Tenne and S. W. Armfield, "A framework for memetic optimization using variable global and local surrogate models," *Soft Computing*, vol. 13, no. 8-9, pp. 781–793, 2009.

[196] X. Chen, L. Feng, and Y. Ong, "A self-adaptive memeplexes robust search scheme for solving stochastic demands vehicle routing problem," *International Journal of Systems Science*, 2011.

[197] J. Zhang, "A survey on trust management for VANETs," in *Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pp. 105–112, IEEE, 2011.

[198] Y. Yang, Q. Feng, Y. Sun, and Y. Dai, "RepTrap: a novel attack on feedback-based reputation systems," in *Proceedings of International Conference on Security and Privacy in Communication Netowrks (SecureComm)*, p. 8, ACM, 2008.

[199] R. Kerr and R. Cohen, "Smart cheaters do prosper: Defeating trust and reputation systems," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 993–1000, 2009.

[200] J. W. Wedel, B. Schunemann, and I. Radusch, "V2X-based traffic congestion recognition and avoidance," in *Proceedings of International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN)*, pp. 637–641, IEEE, 2009.

[201] C. Sommer, S. Joerer, M. Segata, O. Tonguz, R. L. Cigno, and F. Dressler, "How shadowing hurts vehicular communications and how dynamic beaconing can help," in *Proceedings of International Conference on Computer Communications (INFOCOM)*, pp. 110–114, IEEE, 2013.

[202] L. Tsirigotis, E. I. Vlahogianni, and M. G. Karlaftis, "Does information on weather affect the performance of short-term traffic forecasting models?," *International Journal of Intelligent Transportation Systems Research*, vol. 10, no. 1, pp. 1–10, 2012.

[203] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, "Data aggregation and roadside unit placement for a vanet traffic information system," in *Proceedings of International Workshop on VehiculAr Inter-NETworking (VANET)*, pp. 58–65, ACM, 2008.

[204] L. Zhang, D. Gao, W. Zhao, and H.-C. Chao, "A multilevel information fusion approach for road congestion detection in VANETs," *Mathematical and Computer Modelling*, vol. 58, no. 5, pp. 1206–1221, 2013.

[205] P. Patil and A. Gokhale, "Voronoi-based placement of road-side units to improve dynamic resource management in vehicular ad hoc networks," in *Proceedings of International Conference on Collaboration Technologies and Systems (CTS)*, pp. 389–396, IEEE, 2013.

[206] D. H. Stolfi and E. Alba, "Red swarm: smart mobility in cities with eas," in *Proceedings of Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 1373–1380, ACM, 2013.

[207] I.-C. Chang, H.-T. Tai, F.-H. Yeh, D.-L. Hsieh, and S.-H. Chang, "A VANET-based route planning algorithm for travelling time-and energy-efficient gps navigation app," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[208] C. Sommer, O. K. Tonguz, and F. Dressler, "Adaptive beaconing for delay-sensitive and congestion-aware traffic information systems," in *Proceedings of Vehicular Networking Conference (VNC)*, pp. 1–8, IEEE, 2010.

[209] K. Pandit, D. Ghosal, M. Zhang, and C. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks (VANET)," 2013.

[210] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.