
Multi-Objective Hyper-Heuristics

Mashael Suliaman Maashi

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.69222>

Abstract

Multi-objective hyper-heuristics is a search method or learning mechanism that operates over a fixed set of low-level heuristics to solve multi-objective optimization problems by controlling and combining the strengths of those heuristics. Although numerous papers on hyper-heuristics have been published and several studies are still underway, most research has focused on single-objective optimization. Work on hyper-heuristics for multi-objective optimization remains limited. This chapter draws attention to this area of research to help researchers and PhD students understand and reuse these methods. It also provides the basic concepts of multi-objective optimization and hyper-heuristics to facilitate a better understanding of the related research areas, in addition to exploring hyper-heuristic methodologies that address multi-objective optimization. Some design issues related to the development of hyper-heuristic framework for multi-objective optimization are discussed. The chapter concludes with a case study of multi-objective selection hyper-heuristics and its application on a real-world problem.

Keywords: hyper-heuristics, multi-objective optimization, meta-heuristics, evolutionary algorithms, computational search problems

1. Introduction

Many real-world problems are complex. Owing to the (often) NP-hard nature of such problems, researchers and practitioners frequently resort to problem-tailored heuristics in order to obtain a reasonable solution within a reasonable amount of time. Hyper-heuristics are methodologies that operate on a search space of heuristics rather than directly searching the solution space for solving hard computational problems. One of the main aims of hyper-heuristics is to raise the level of generality of search methodologies and to automatically adapt the algorithm by combining the strength of each heuristic and making up for the weaknesses of others. References to multi-objective hyper-heuristics are scarce

as most research in this area has been limited to single-objective optimization. The purpose of this chapter is to provide an introduction to hyper-heuristic methodologies that are designed for multi-objective optimization (HHMOs). The chapter might help researchers and PhD students interested in this research area to understand and use these methods. Hyper-heuristics for multi-objective optimization is a relatively new area of research in operational research (OR) and evolutionary computation [1, 2]. Few studies have dealt with hyper-heuristics for multi-objective problems. In order to offer a better understanding of the subject, the basic concepts of multi-objective optimization and hyper-heuristics are provided in Section 2. This chapter also provides a brief survey of hyper-heuristic methodologies that address multi-objective optimization in Section 3. Some design issues related to the development of a hyper-heuristic framework for multi-objective optimization are discussed in Section 4. Additionally, a case study of multi-objective selection hyper-heuristics and its application on a real-world problem is presented in Section 5. Finally, promising research areas for future application are provided in Section 6.

2. The basic concepts and underlying issues

2.1. Multi-objective optimization

Multi-objective problems (MOPs): MOPs comprise several objectives (two or more), which need to be minimized or maximized depending on the problem. A general definition of an MOP [3] is

An MOP minimizes $F(x) = (f_1(x), \dots, f_k(x))$ subject to $g_i(x) \leq 0; i = 1, \dots, m, x \in \Omega$. The solution of the MOP minimizes the components of a vector $F(x)$, where x is an n -dimensional decision variable vector ($x = x_1, \dots, x_n$) from some universe Ω . An MOP includes n decision variables, m constraints, and k objectives. The MOP's evaluation function $F: \Omega \rightarrow \wedge$ maps decision variable vectors ($x = x_1, \dots, x_n$) to vectors ($y = a_1, \dots, a_k$).

Multi-objective optimization techniques are divided into three classes [3, 4]:

- **A priori approach** (decision making and then a search):
In this class, the objective preferences or weights are set by the decision maker prior to the search process. An example of this is aggregation-based approaches such as the weighted sum approach.
- **A posteriori approach** (a search and then decision making):
The search is conducted to find solutions for the objective functions. Following this, a decision process selects the most appropriate solutions (often involving a trade-off). Multi-objective evolutionary optimization (MOEA) techniques, whether non-Pareto-based or Pareto-based, are examples of this class.
- **Interactive or progressive approach** (search and decision making simultaneously):
In this class, the preferences of the decision maker(s) are made and adjusted during the search process.

Pareto dominance: The idea behind the dominance concept is to generate a preference between MOP solutions since there is no information regarding objective preference provided by the decision maker. This preference is used to compare the dominance between any two solutions [5]. A more formal definition of Pareto dominance (for minimization case) is as follows [4]:

A vector $u = (u_1, \dots, u_k)$ is said to dominate another vector $v = (v_1, \dots, v_k)$ (denoted by $u \leq v$) according to k objectives if and only if u is partially less than v , that is, $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$.

All non-dominated solutions are also known as the *Pareto optimal sets*. The corresponding Pareto optimal set, with respect to the objective space, is known as the *Pareto optimal front*. The quality of the obtained *Pareto optimal set* can be determined in Refs. [6, 7]: the extent of the Pareto optimal set and the distance and the distribution of the Pareto optimal front.

2.2. Hyper-heuristics

The main aim of the hyper-heuristic methodology is raising the level of generality of search techniques by producing general search algorithms that are applicable for solving a wide range of the problems in different domains [2, 8, 9]. In a hyper-heuristic approach, different heuristics (or heuristic components) can be selected, generated, or combined to solve a given optimization problem in an efficient way. Since each heuristic has its own strengths and weaknesses, one of the aims of hyper-heuristics is to automatically inform the algorithm by combining the strength of each heuristic and making up for the weaknesses of others. This process requires the incorporation of a learning mechanism into the algorithm to adaptively direct the search at each decision point for a particular state of the problem or the stage of search. It is obvious that the concept of hyper-heuristics has strong ties to operational research (OR) in terms of finding optimal or near-optimal solutions to computational search problems. It is also firmly linked to artificial intelligence (AI) in terms of machine-learning methodologies [8].

2.2.1. Generic hyper-heuristic framework

In their simplest form, hyper-heuristics is a search methodology that encompasses a high-level strategy (which could be a meta-heuristic) that controls the search over a set of heuristics (low-level heuristics) rather than controlling a search over a direct representation of the solutions. Usually, in a hyper-heuristic framework, there is a clear separation between high-level strategy and the set of low-level heuristics [10]. The purpose of domain barrier is to give the hyper-heuristics a higher level of abstraction. This also increases the level of generality of hyper-heuristics by enabling its application to a new problem without the need for a framework change. Only information relevant to the problem domain is provided from low level to high level, including cost/fitness measured by an evaluation function, indicating the quality of a solution. The high-level strategy can be a (meta-) heuristic or a learning mechanism. The task of the high-level strategy is to guide the search intelligently and adapt according to the success/failure of the low-level heuristics or combinations of heuristic components during the search process.

2.2.2. Hyper-heuristics classification

Generally, there are two recognized methodologies of hyper-heuristics: *selection* and *generation* hyper-heuristics. For both hyper-heuristic methodologies, there are two types of heuristics: (i) *constructive heuristics*, which process a partial solution(s) and build a complete solution(s) and (ii) *perturbative heuristics*, which operate on complete solution(s).

In the context of hyper-heuristics for multi-objective optimization, we could classify them into three categories:

- **Multi-objective selection hyper-heuristic** manages a number of heuristics during an iterative process for a given time. At each iteration, one of best heuristics is chosen at a decision point to operate and run. This type comprises two main stages: *heuristic selection* and *move-acceptance* strategy.
- **Multi-objective combination/hybridization hyper-heuristic** combines a number of heuristics or (components of heuristics) that operate simultaneously and adaptively to create new solutions.
- **Multi-objective generation hyper-heuristic:** To the best of the authors' knowledge, there are no studies addressing multi-objective generation hyper-heuristic in the literature.

3. Brief survey of hyper-heuristics for multi-objective optimization

The hyper-heuristics for multi-objective optimization problems is a new area of research in evolutionary computation and operational research [2, 8]. To date, few studies have been identified that deal with hyper-heuristics for multi-objective problems (see **Table 1**).

Regarding multi-objective selection hyper-heuristics, a multi-objective hyper-heuristic tabu search (TS) based (TSRoulette Wheel) was presented in Ref. [11]. In this approach, an appropriate heuristic is selected at each iteration using tabu search as a high-level search strategy. The experiments showed results with acceptable solution when applied in space allocation and timetabling problems. In Ref. [12], an online selection hyper-heuristic, Markov-chain-based hyper-heuristic (MCHH), is investigated. The Markov chain guides the selection of heuristics and applies online reinforcement learning to adapt transition weights between heuristics. In MCHH, hybrid meta-heuristics and evolution strategies were incorporated and applied to the DTLZ test problems [13] and compared to a (1+1) evolution strategy meta-heuristic, a random hyper-heuristic, and TSRoulette Wheel [11]. The comparison shows the efficacy of the proposed approach in terms of Pareto convergence and learning ability to select good heuristic combinations. Further work is needed in terms of diversity-preserving mechanisms. The MCHH was applied to the Walking Fish Group (WFG) test problems [14], and although the results from the performed tests show the ability of the approach, future work needs to be made to improve the search strategies [12]. MCHH has been applied to real-world water distribution networks and it produced competitive results [15]. A multi-objective hyper-heuristic optimization scheme for engineering system designs is presented in Ref. [16].

Component name	Application domain/test problems	Reference(s)
Tabu search	Space allocation, timetabling Traveling salesman problems	[8] [35]
Markov chain, evolution strategy	Real-world water distribution networks design/DTLZ, WFG	[12]
NSGAII	Irregular 2D cutting stock Strip packing and cutting stock	[37] [39]
NSGAII, quasi-Newton algorithm	Stacked neural network	[40]
Number of operations from NSGAII, SPEA2, and IBEA	A number of continuous multi-objective test problems	[19]
Number of selection, crossover, and mutation operations of evolutionary algorithms	Software module clustering	[23]
Hypervolume	Dynamic-mapped island-based model/WFG	[36]
Particle swarm optimization, adaptive metropolis algorithm, differential evolution	Water resource problems/a number of continuous multi-objective test problems	[41]
Memory strategy, genetic and differential operators	Dynamic optimization problems/a number of continuous multi-objective test problems	[46]
Genetic algorithm, simulated annealing, particle swarm optimization	Engineering system design problems/a number of classical multi-objective test problems	[16]
Simulated annealing	Shelf-space allocation	[34]
NSGAII, SPEA2, MOGA, choice function, great deluge algorithm, and late acceptance	WFG/the vehicle crashworthiness design problem	[24–26]
NSGAII, SPEA2, IBEA, choice function, great deluge algorithm	WFG/wind farm layout optimization	[32]
Markov model	DTLZ	[33]

Table 1. Heuristic components and application domains of hyper-heuristics for multi-objective optimization.

Simulated annealing (SA) [17], genetic algorithm, and particle swarm optimization [18] are used as low-level heuristics. A multi-indicator hyper-heuristics for multi-objective optimization is proposed in Ref. [19]. The approach based on multiple rank of indicators is taken from NSGAII [20], SPEA2 [22], and IBEA [21]. In Ref. [23], a multi-objective hyper-heuristic genetic algorithm (MHypGA) for multi-objective software module clustering problem is presented. In MHypGA, different strategies of selection, crossover, and mutation operations of genetic algorithms are incorporated as low-level heuristics. In Refs. [24–26], online-learning multi-objective hyper-heuristics are presented. These multi-objective hyper-heuristics are based on a choice function. The multi-objective choice-function-based hyper-heuristics are combined with different move-acceptance strategies including all-moves (AMs) and the great deluge algorithm (GDA) [27] and late acceptance (LA) [28]. The multi-objective hyper-heuristic controls and combines the strengths of three well-known multi-objective evolutionary algorithms (NSGAII [20], SPEA2 [22], and MOGA [30]). The performance of the proposed multi-objective

choice-function-based hyper-heuristics is evaluated on the Walking Fish Group (WFG) test suite [14] and is applied to the vehicle crashworthiness design problem [31]. The results of both benchmark test problems demonstrate the capability and potential of the multi-objective hyper-heuristic approaches in solving continuous multi-objective optimization problems. More details about these approaches are provided as a study case in Section 5. In Ref. [32], a multi-objective selection hyper-heuristics for a multi-objective wind farm layout optimization is proposed. The experimental results show that the proposed approach is successfully applied to this optimization problem. In Ref. [33], a multi-objective selection sequence-based hyper-heuristic (MOSSH) is proposed. The MOSSH algorithm employs a hidden Markov model based on a sequence of heuristics that is determined by transition probabilities on ε -dominance. The proposed approach has been applied to DTLZ test problems [13] and results showed its capability to solve it through the learning process. A multiple neighborhood hyper-heuristic for two-dimensional (2D) shelf-space allocation problem is proposed in Ref. [34]. The proposed hyper-heuristic is based on a simulated annealing algorithm. A two-stage multi-objective hyper-heuristic approach is presented in Ref. [35]. The first phase targets in producing an efficient Pareto front, and the second phase focuses on solving a given problem in a flexible way so as to drive a subset of the population to the desired Pareto front. The approach was assessed on the multi-objective-traveling salesman problems using 11 low-level heuristics. Comparison with other methods from the scientific literature revealed that the proposed approach produces high-quality results. Nevertheless, upcoming efforts are still necessary in advancing the approach. In Ref. [36], a hypervolume-based hyper-heuristic for a dynamic-mapped multi-objective island-based model is proposed. This method shows its superiority when compared to the contribution-based hyper-heuristic and other standard parallel models over the WFG test problems [14]. A new hyper-heuristic based on the multi-objective evolutionary algorithm NSGAII [20] is proposed in Ref. [37]. The main idea of this method involves the production of the final Pareto-optimal set, through a learning process that evolves combinations of condition-action rules based on NSGAII. The proposed method was tested on many instances of irregular 2D-cutting stock benchmark problems and produced promising results. A hyper-heuristic-based codification is proposed in Refs. [38, 39], for solving strip packing and cutting stock problems in order to maximize the total profit and minimize the total number of cuts. The experimental results show that outcomes of the proposed hyper-heuristic outperform single heuristics. In Ref. [40], a multi-objective hyper-heuristic for the design and optimization of a stacked neural network is proposed. The proposed approach is based on NSGAII [20] combined with a local search algorithm (Quasi-Newton algorithm).

Regarding multi-objective hybridization hyper-heuristic, an adaptive multi-method (multi-point) search called AMALGAM is proposed in Ref. [41]. It employs multiple search algorithms, NSGAII [20], PSO [18], AMS [42], and DE [43], simultaneously using the concepts of multi-method search and adaptive offspring creation. AMALGAM has been applied to a number of continuous multi-objective test problems, and it has been shown to be superior to other methods. It has also been applied to solve a number of water resource problems and has yielded very good solutions [44, 45]. A multi-strategy ensemble, multi-objective evolutionary algorithm called MS-MOEA for dynamic optimization, is proposed in Ref. [46]. The approach

combines differential operators and different strategies, including strategy and genetic, to adaptively creating offspring and increase the convergence speed. The results show that MS-MOEA obtains solution with good quality.

It is worth mentioning that multi-objective hybridization hyper-heuristics are similar to multi-objective selection hyper-heuristics in terms of the incorporation of different algorithms. However, they are different in their concepts. Multi-objective selection hyper-heuristic is based on two successive stages: a selection mechanism and an acceptance move strategy. By contrast, multi-objective hybridization hyper-heuristics is based on an adaptive creation offspring strategy. In multi-objective selection hyper-heuristics, a sequence of heuristics/meta-heuristics is executed during the search, that is, one heuristic/meta-heuristic is selected and applied at each stage (iteration/decision point) of the search. The high-level strategy in hyper-heuristics evaluates the performance of a set of heuristics/meta-heuristics in order to improve the population of solutions. By contrast, in multi-objective hybridization hyper-heuristics, multiple heuristic/meta-heuristics run concurrently. Each heuristic/meta-heuristic produces a different population of offspring, and then, all produced offspring are evaluated to evolve a new population of offspring by an adaptive creation offspring strategy.

4. Multi-objective hyper-heuristics design issues

The idea of hybridizing a number of algorithms (heuristics) into a hyper-heuristic framework is straightforward and meaningful. However, many design issues related to the development of hyper-heuristics for multi-objective optimization require more attention when designing such a framework to be applicable and effective.

The main components of the hyper-heuristic framework are low-level heuristics, selection method, learning mechanism, and move-acceptance method. The choosing of these components is critical. Each component in the hyper-heuristic framework plays a significant role in improving the quality of both the search and the eventual solution. The components of the hyper-heuristic in the context of multi-objective optimization are discussed as follows.

4.1. Low-level heuristics

The choice of appropriate low-level heuristics is not an easy task. Many questions arise here, what heuristics (algorithms) are suitable for dealing with multi-objective optimization problems? Are priori approaches or a posteriori approaches more suitable? Are non-Pareto-based or Pareto-based approaches more applicable? The author agrees with many researchers [30, 38, 47–51] that evolutionary algorithms and population-based methods such as decomposition-based approaches MOEA/Ds (e.g., [52, 53]) and indicator-based approaches (e.g., [54, 55]) are more suitable in dealing with multi-objective optimization problems because of their population-based nature, which means that they can find Pareto optimal sets (trade-off solutions) in a single run, thus allowing a decision maker to select a suitable compromise solution (with respect to the space of the solutions). In the context of multi-objective hyper-heuristics,

a decision maker here could be a selection method that decides which is the best low-level heuristic to select at each decision point (with respect to the space of the heuristics).

4.2. Selection methods

As a selection hyper-heuristic relies on an iterative process, the main questions arising here are what is an effective way to choose an appropriate heuristic at each decision point, and how to choose this heuristic, that is, which criteria can be considered when choosing a heuristic? In single-objective cases, this criterion is easy to determine by measuring the quality of the solution, such as the objective/cost value and time. However, this is more complex when tackling a multi-objective problem. The quality of the solution is not easy to assess. There are many different criteria that should be considered, such as the number of non-dominated individuals and the distance between the non-dominated front and the POF. To make a simple framework, a higher level of abstraction of the hyper-heuristics should be considered when designing such a framework. It is not necessary to provide any problem-specific information, such as the number of objectives or the nature of the solution space to the high-level strategy. More attention should be given to the performance of the low-level heuristics. This will boost the intensification element. Therefore, a heuristic with the best performance will be chosen more frequently to exploit the search area. The aim is to achieve a kind of balance between the intensification and diversification when choosing a heuristic. Selection methods based on randomization support only the diversification by exploring unvisited areas of the search space. Reinforcement learning (RL) [56] uses support intensification as a selection method by rewarding and punishing each heuristic based on its performance during the search using a scoring mechanism. An example of good selection method is the choice function, which could provide a balance between intensification and diversification.

4.3. Learning and feedback mechanism

Not all hyper-heuristic approaches incorporate a learning mechanism. However, a learning mechanism is strongly linked to the selection method. An example of this is a random hyper-heuristic classified as an offline-learning approach [1], because the random selection does not provide any kind of learning. The learning mechanism guides the selection method to which best heuristic should be chosen at each decision point. A best heuristic refers to the heuristic that produces solutions with good quality based on some criteria using performance measurements. It is good to note that the measurement of the quality of the solution for multi-objective problems requires assessing different aspects of the non-dominated set in the objective space. In the scientific literature, many performance metrics have been proposed to measure different aspects of the quality and quantity of the resulting non-dominated set [4, 29, 57].

4.4. Move-acceptance method

The selection hyper-heuristic framework comprises two main stages: selection and move-acceptance methods. A move-acceptance criterion can be deterministic or non-deterministic. A deterministic move-acceptance criterion produces the same result, given the configuration

(e.g., proposed new solution, etc.). Non-deterministic move-acceptance criteria may generate a different result even when the same solutions are used for the decision at a same given time. This could be because the move-acceptance criterion depends on time or it might have a stochastic component while making the accept/reject decision. Examples of deterministic move-acceptance criteria are all-moves, only-improving, and improving & equal. For non-deterministic move-acceptance criteria, the candidate solution is always accepted if it improves the solution quality, while worsening solutions can be accepted based on an acceptance function, including GDA [27], LA [28], Monte Carlo [58], and SA [17]. Selection of the move-acceptance criteria has to be compatible with the selection methods. In the scientific literature, many combinations of the selection method and move-acceptance criterion have been successfully applied to single-objective optimization [59]. It could be worth employing them in the context of hyper-heuristics for multi-objective optimization.

5. Case study: multi-objective choice-function-based hyper-heuristics

5.1. The proposed multi-objective hyper-heuristics methodologies

This study [26] highlights the lack of scientific study that has been conducted in hyper-heuristics and multi-objective optimization, investigates the design of a hyper-heuristic framework for multi-objective optimization, and develops hyper-heuristic approaches for multi-objective optimization (HHMOs) to solve continuous multi-objective problems. Hyper-heuristic frameworks generally impose a domain barrier that separates the hyper-heuristic from the domain implementation along with low-level heuristics to provide a higher level of abstraction. The domain barrier does not allow any problem-specific information to be passed to the hyper-heuristic itself during the search process. The multi-objective choice-function-based hyper-heuristic framework is designed in this same modular manner (see **Figure 1**).

One of the advantages of the multi-objective choice-function-based hyper-heuristic framework is its simplicity. It is also highly flexible, and its components are reusable. Moreover, it is built on an interface that allows other researchers to write their own multi-objective hyper-heuristic components easily. Even the low-level heuristics can be easily changed if required. If new and better-performing components are found in the future, they can be incorporated. Based on the multi-objective selection hyper-heuristic framework, three online-learning-selection, choice-function-based hyper-heuristics are proposed: HHMO_CF_AM, HHMO_CF_GDA, and HHMO_CF_LA [24, 25]. The multi-objective choice-function-based hyper-heuristics control and combine the strengths of three well-known multi-objective evolutionary algorithms (NSGAII, SPEA2, and MOGA), which are utilized as the low-level heuristics. The choice-function-selection heuristic acts as a high-level strategy that adaptively ranks the performance of those low-level heuristics according to feedback received during the search process, determining which one to call at each decision point. Four performance measurements (algorithm effort (AE), ratio of non-dominated individuals (RNI) [5], size of space covered (SSC) [60], and uniform distribution of a non-dominated population (UD) [61]) are integrated into a ranking scheme that acts as a feedback-learning mechanism

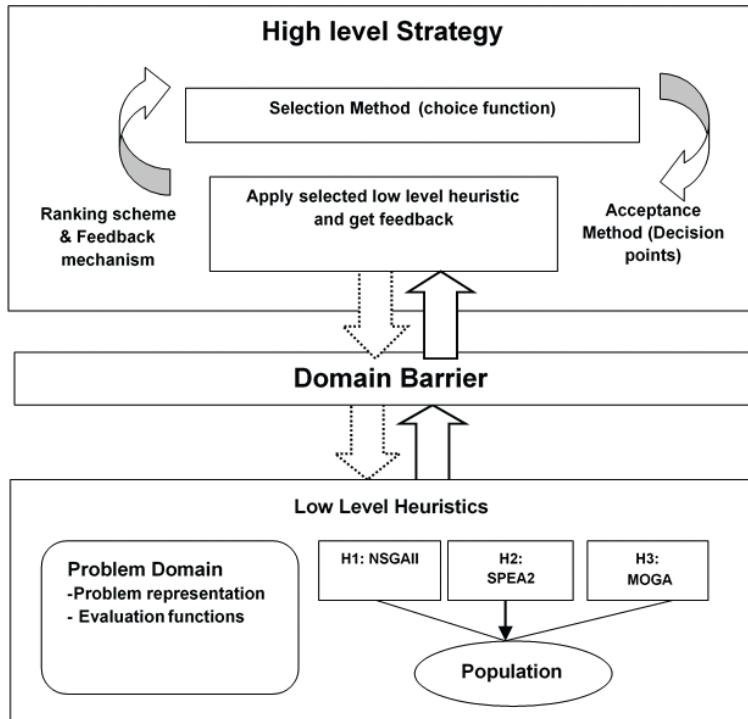


Figure 1. Proposed framework of the hyper-heuristic choice function based on multi-objective optimization problems.

to provide knowledge of the problem domain to the high-level strategy. The multi-objective choice-function-based hyper-heuristic is combined with different move-acceptance strategies, including all-moves (AM) as a deterministic move acceptance and GDA [27] and LA [28] as a non-deterministic move acceptance. GDA and LA require a change in the value of a single objective at each step, and hence a well-known hypervolume metric, referred to as D metric, is proposed for their applicability to the multi-objective optimization problems. The D metric is integrated into the non-deterministic move-acceptance criterion in order to convert the multi-objective optimization to the single-objective optimization without having to define value weights for the various objectives. For more details about the HHMOs algorithm, see Refs. [24, 25].

5.2. Problem description

The multi-objective vehicle crashworthiness design problem has only five decision variables and no constraints [31]. The output of the problem provides a wider choice for engineers to make their final design decision based on the Pareto solution space. The decision variables of the problem represent the thickness of five reinforced members around the front as they could have a significant effect on the crash safety. The mass of the vehicle is tackled as the

first design objective, while the integration of collision acceleration between $t_1 = 0.05$ s and $t_2 = 0.07$ s in the full-frontal crash is considered as the second objective function. The toe-board intrusion in the 40% offset-frontal crash is tackled as the third objective as it is the most severe mechanical injury. The three objectives are formulated as follows:

$$\text{Mass} = 1640.2823 + 2.3573285 t_1 - 2.3220035 t_2 + 4.5688768 t_3 + 7.7213633 t_4 + 4.4559504 t_5 \quad (1)$$

$$\begin{aligned} \text{Ain} = & 6.5856 + 1.15 t_1 - 1.0427 t_2 + 0.9738 t_3 + 0.8364 t_4 - 0.3695 t_1 t_4 + 0.0861 t_1 t_5 \\ & + 0.3628 t_2 t_4 - 0.1106 t_1^2 - 0.3437 t_3^2 + 0.1764 t_4^2 \end{aligned} \quad (2)$$

$$\begin{aligned} \text{Intrusion} = & -0.0551 + 0.0181 t_1 + 0.1024 t_2 + 0.0421 t_3 - 0.0073 t_1 t_2 + 0.024 t_2 t_3 \\ & - 0.0118 t_2 t_4 - 0.0204 t_3 t_4 - 0.008 t_3 t_5 - 0.0241 t_2^2 + 0.0109 t_4^2 \end{aligned} \quad (3)$$

Thus, the multi-objective design of vehicle crashworthiness problem in T decision variable space is formulated as

$$\begin{aligned} m \in F(x) = & [\text{Mass}, \text{Ain}, \text{Intrusion}] \\ s.t. & 1 \text{ mm} \leq x \leq 3 \text{ mm} \\ \text{where } & x = (t_1, t_2, t_3, t_4, t_5)^T \end{aligned} \quad (4)$$

5.3. Performance evaluation criteria and experimental settings

A set of experiments are conducted over a multi-objective vehicle crashworthiness design problem as a real-world problem to evaluate the performance of the multi-objective choice-function-based hyper-heuristics: HHMO_CF_AM, HHMO_CF_GDA, and HHMO_CF_LA. The performance of three multi-objective hyper-heuristics is compared to the well-known multi-objective evolutionary algorithm, NSGAII [20]. Five performance metrics are used to measure the quality of the approximation sets from different aspects: (i) RNI [5], (ii) SSC [60], (iii) UD [61], (iv) generational distance (GD) [62], and (v) inverted generational distance (IGD) [63]. In addition, the *t*-test is used as a statistical test for the average performance comparison of selection hyper-heuristics and the results. Thirty independent runs are performed for each comparison method using the same parameter settings as those provided in Ref. [31] with a population size equal to 30. All multi-objective hyper-heuristics methodologies run for a total of 75 iterations (stages). In each iteration, a low-level heuristic is selected and applied to execute 50 generations. Thus, all methods are terminated after 3750 generations. Other experimental settings are provided in Refs. [24, 25].

5.4. Results

The mean performance comparison of AM, GDA, LA, and NSGAII based on the performance metrics (RNI, SSC, UD, GD, and IGD) for solving the vehicle crashworthiness problems is provided in **Tables 2** and **3**.

Method	RNI				SSC				UD			
	Avg	Min	Max	Std	Avg	Min	Max	Std	Avg	Min	Max	Std
NSGAII	1.00	1.00	1.00	0.00	7.936E+07	4.168E+07	9.587E+07	1.595E+07	0.592	0.532	0.670	0.045
AM	1.00	1.00	1.00	0.00	7.381E+07	5.315E+07	9.577E+07	1.463E+07	0.585	0.516	0.707	0.050
GDA	1.00	1.00	1.00	0.00	8.289E+07	6.294E+07	9.580E+07	1.954E+07	0.613	0.555	0.692	0.034
LA	1.00	1.00	1.00	0.00	7.538E+07	4.512E+07	9.550E+07	1.474E+07	0.582	0.302	0.641	0.062

Table 2. The performance NSGAII and the selection choice-function-based hyper-heuristics using different move-acceptance strategies on the vehicle crashworthiness problems with respect to the metrics: the ratio of non-dominated individuals (RNI), the hypervolume (SSC), and the uniform distribution (UD).

Method	GD				IGD			
	Avg	Min	Max	Std	Avg	Min	Max	Std
NSGAII	2.48E-03	1.46E-03	4.21E-03	9.10E-04	4.156E-03	1.543E-03	1.289E-02	3.859E-03
AM	2.71E-03	1.59E-03	4.06E-03	7.90E-04	4.376E-03	1.738E-03	1.288E-02	4.168E-03
GDA	2.11E-03	1.10E-03	4.28E-03	7.10E-04	3.552E-03	1.661E-03	1.230E-02	3.075E-03
LA	3.32E-03	1.70E-03	6.76E-03	1.33E-03	3.604E-03	1.525E-03	1.238E-02	2.582E-03

Table 3. The performance NSGAII and the selection choice-function-based hyper-heuristics using different move-acceptance strategies on the vehicle crashworthiness problems with respect to the metrics: the generational distance (GD) and the inverted generational distance (IGD).

For each performance metric, the average, minimum, maximum, and standard deviation values are computed. For all metrics, a higher value indicates better performance, except in GD and IGD, where a lower value indicates better performance. The statistical *t*-test results of NSGAII and the three multi-objective choice-function-based hyper-heuristics (AM, GDA, and LA) are given in **Table 4**.

The distribution of the simulation data of the 30 independent runs for the comparison methods with respect to these performance metrics is visualized as box plots, shown in **Figure 2**. The results indicate that all methods perform similar to each other with respect to the metric of RNI over. GDA exhibits the best performance in the metrics of SSC, GD, and IGD, and it converges better toward the POF than the other methods. GDA also exhibits the best performance in the metric of UD and distributes more uniformly than other methods.

The 50% attainment surface for each method, from the 30 fronts after 3750 generations, is computed and illustrated in **Figure 3**. GDA appears to generate a good convergence. It can be clearly observed that GDA converges to the best POF with a well-spread Pareto front as compared to the other approaches. By contrast, AM generates the poorest solutions. NSGAII and LA have similar convergence. In general, the hyper-heuristics for real-world multi-objective problems benefits from the use of a learning heuristic selection method as well as GDA. The results demonstrate the effectiveness of our selection hyper-heuristics particularly when combined with great deluge algorithm as a move-acceptance criterion. HHMO_CF_GDA

Methods	Metrics				
	RNI	SSC	UD	GD	IGD
NSGAII:AM	n/a	+	±	±	±
NSGAII:GDA	n/a	-	+	±	-
NSGAII:LA	n/a	+	+	+	-
AM:GDA	n/a	-	-	±	-
AM:LA	n/a	-	±	+	-
GDA:LA	n/a	+	+	+	±

Table 4. The *t*-test results of NSGAII and the three multi-objective choice-function-based hyper-heuristics methodologies on the multi-objective vehicle crashworthiness design problems with respect to the metrics: the ratio of non-dominated individuals (RNI), the hypervolume (SSC), the uniform distribution (UD), the generational distance (GD), and the inverted generational distance (IGD).

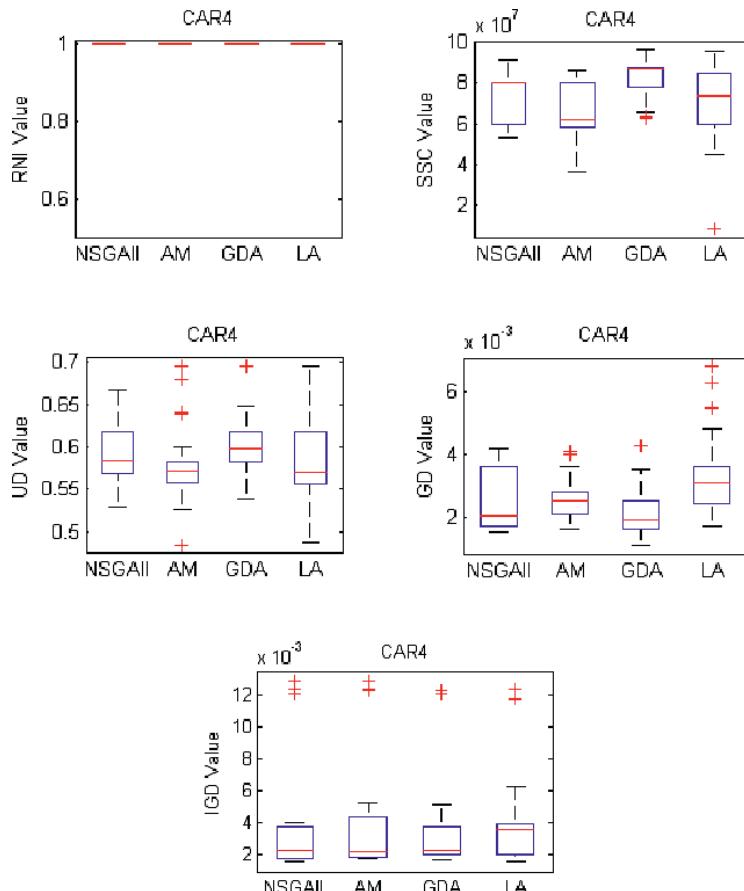


Figure 2. Box plots of multi-objective choice-function-based hyper-heuristics methodologies and NSGAII on the multi-objective vehicle crashworthiness design problems.

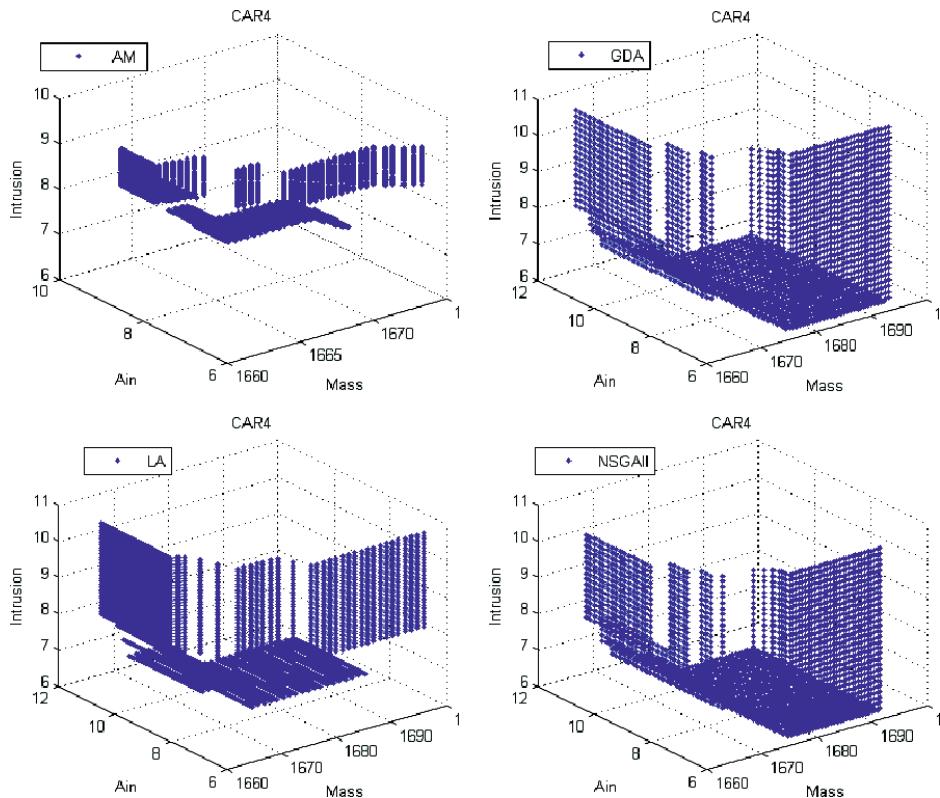


Figure 3. The 50% attainment surfaces for NSGAII and the three multi-objective choice-function-based hyper-heuristics (AM, GDA, and LA) after 3750 generations on the multi-objective design of vehicle crashworthiness problem.

turns out to be the best choice for solving this problem. Although other multi-objective hyper-heuristics still produce solutions with acceptable quality in some cases, they could not perform as well as NSGAII. In summary, the results of the real-world problem demonstrate the capability and potential of the multi-objective hyper-heuristic approaches in solving continuous multi-objective optimization problems.

6. Some promising research area

Multi-objective hyper-heuristic offers interesting potential research directions in multi-objective optimization. Some of these promising research areas are recommended as follows:

- Many heuristic selection methods can be adapted from previous research in single-objective optimization and can be used for multi-objective optimization within multi-objective selection hyper-heuristic. This process is not a trivial process, requiring elaboration of existing methods

and their usefulness in a multi-objective setting. Furthermore, other acceptance criteria such as simulated annealing (SA) and tabu search (TS) [58] could be employed as a move-acceptance component within the selection hyper-heuristic framework for multi-objective optimization. As those criteria involve many parameters, this methodology would require initial experiments to tune the parameters for multi-objective settings, such as defining a cooling schedule and an initial temperature for SA and aspiration criterion and tabu tenure for TS.

- There are many low-level heuristics choices possible and, therefore, great scope for research in this area. It would be interesting to employ *state-of-the-art* multi-objective optimizers and other population-based methods that obtain promising results to act as low-level heuristics within the multi-objective combination/selection hyper-heuristic framework.
- It would be interesting to test the level of generality of existing multi-objective hyper-heuristic frameworks further on some other problems and domains, including the continuous real-valued constrained, combinatorial, discrete, and dynamic problems.
- Since no studies are found in the scientific literature that address multi-objective generation hyper-heuristic, it would be interesting to propose a multi-objective generation hyper-heuristic framework. This indicates great scope for research. Many-generation hyper-heuristic methodologies have been successfully applied to single-objective optimization; it would also be interesting to modify them to deal with multi-objective optimization.

Author details

Mashael Suliaman Maashi

Address all correspondence to: m.maashi@gmail.com

Tabuk University, King Saud University, Saudi Arabia

References

- [1] Burke E, Hyde M, Kendall G, Ochoa G, Özcan E, Woodward JR. A classification of hyper-heuristic approaches. In: Handbook of Meta-Hruristics. USA: Kluwer Academic Publisher; 2010. pp. 449-468
- [2] Özcan E, Bilgin B, Korkmaz, E. A comprehensive analysis of hyper-heuristics. Intelligent Data Analysis. 2008;12(1):3-23
- [3] Van Veldhuizen DV, Lamont G. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. Evolutionary Computation. 2000;8(2):125-147
- [4] Coello CC, Van Veldhuizen DV, Lamont G, editors. Evolutionary Algorithms for Solving Multi-Objective Problems. USA: Kluwer Academic Publishers; 2007

- [5] Tan KC, Lee TH, Khor EF. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial Intelligence Review*. 2002;17:253-290
- [6] Landa-Silva D, Burke E, Petrovic S. An introduction to multiobjective metaheuristics for scheduling and timetabling. In: *Lecture Notes in Economics and Mathematical Systems*. Berlin, Heidelberg: Springer; 2004. pp. 91-129
- [7] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*. 2000;8(2):173-195
- [8] Burke EK, Gendreau M, Hyde M, Kendall G, Ochoa G, Özcan E, Qu R. Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*. 2013;64:1695-1724
- [9] Ross P. Hyper-heuristics. In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Methodologies*. Bücher: Springer; 2005. pp. 529-556
- [10] Burke E, Kendall G, Newall J, Hart E, Ross P, Schulenburg S. Hyper-heuristics: An emerging direction in modern search technology. In: *Handbook of Meta-Heuristics*. USA: Kluwer Academic Publishers; 2003. pp. 457-474
- [11] Burke E, Landa-Silva D, Soubeiga E. Multi-objective hyper-heuristic approaches for space allocation and timetabling. In: *MIC 2003-Meta-Heuristics: Progress as Real Problem Solvers*. USA: Springer; 2003. pp. 129-158
- [12] McClymont K, Keedwell EC. Markov chain hyperheuristic (mchh): An online selective hyper-heuristic for multiobjective continuous problems. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO11)*; 2011. pp. 2003-2010
- [13] Deb K, Jain S. Running performance metrics for evolutionary multiobjective optimization. Technical Report KanGAL Report No. 2002004. India: Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur; 2002
- [14] Huband S, Hingston P, Barone L, While, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*. 2006;10:477-506
- [15] McClymont K, Keedwell E, Savic, D. A general multi-objective hyper-heuristic for water distribution network design with discolouration risk. *Journal of Hydroinformatics*. 2013;15(3):700-716
- [16] Rafique AF. Multiobjective hyper-heuristic scheme for system design and optimization. In: *Proceedings of 9th International Conference on Mathematical Problems in Engineering, Aerospace and Science, AIP Conference 1493*; 2012
- [17] Kirkpatrick S, Gelatt C, Vecchi M. Optimization by simulated annealing. *Science*. 1983;22:671-680
- [18] Kennedy J, Eberhart R, Shi Y. *Swarm Intelligence*. USA: Morgan Kaufmann; 2001

- [19] Vázquez-Rodríguez J, Petrovic S. Calibrating continuous multi-objective heuristics using mixture experiments. *The Journal of Heuristics*. 2012;**18**:699-726
- [20] Deb K, Controlled elitist nondominated sorting genetic algorithms for better convergence. In: Proceedings of Evolution Multi Criterion Optimization Conference; 2001. pp. 67-81
- [21] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Lecture Notes in Computer Science, Parallel Problem Solving from Nature (PPSN VIII); USA: Springer; 2004. pp. 832-842
- [22] Zitzler E, Laumanns M, Thiele L. Spea2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: EUROGEN 2001—Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problem. USA: Springer; 2001. pp. 95-100
- [23] Kumari A, Srinivas K, Gupta M. Software module clustering using a hyperheuristic based multi-objective genetic algorithm. In: Advance Computing Conference (IACC), 2013 IEEE 3rd International; 2013. pp. 813-818
- [24] Maashi, MS, Özcan E, Kendall G. A multi-objective hyper-heuristic based on choice function. *Expert Systems with Applications*. 2014;**41**(9):4475-4493
- [25] Maashi MS, Kendall G, Özcan E. Choice function based hyper-heuristics for multi-objective optimization. *Applied Soft Computing*. 2015;**28**:312-326
- [26] Maashi MS. An investigation of multi-objective hyper-heuristics for multi-objective optimisation [thesis]. Nottingham, UK: University of Nottingham; 2014
- [27] Dueck G. New optimization heuristics: The great deluge algorithm and the record to record travel. *Journal of Computational Physics*. 1993;**104**:86-92
- [28] Burke EK, Bykov Y. A late acceptance strategy in hill-climbing for exam timetabling problems. In: International Conference on the Practice and Theory of Automated Timetabling; 2008
- [29] Fonseca C, Fleming P. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*. 1998;**28**(1):26-37
- [30] Glover F. Future paths for integer programming and links to artificial intelligence. *Computer Operations Research*. 1986;**13**(5):533-549
- [31] Liao X, Li Q, Yang X, Zhang W, Li W. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*. 2008;**35**:561-569
- [32] Li W, Özcan E, John R. Multi-objective evolutionary algorithms and hyper-heuristics for wind farm layout optimization. *Renewable Energy*. 2016;**105**:473-482
- [33] Walker D, Keedwell, editors. Multi-objective optimisation with a sequence-based selection hyper-heuristic GECCO '16 companion. In: Proceedings of the 2016 on Genetic and

- Evolutionary Computation Conference Companion; July 20-24, 2; Denver, Colorado, USA. New York, NY, USA: ACM; 2016. pp. 81-82
- [34] Bai R, Woensel T, Kendall G, Burke EK. A new model and a hyper-heuristic approach for two-dimensional shelf space allocation. *Journal Operation Research*. 2013;**11**:31-55
 - [35] Veerapen N, Landa-Silva D, Gandibleux X. Hyperheuristic as component of a multi-objective metaheuristic. In: Proceedings of the Doctoral Symposium on Engineering Stochastic Local Search Algorithms (SLS-DS 2009); 2009
 - [36] Len C, Miranda G, Segura C. Hyperheuristics for a dynamic-mapped multiobjective island-based model. In: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. In: Lecture Notes in Computer Science; Berlin Heidelberg: Springer; 2009. pp. 41-49
 - [37] Gomez J, Terashima-Marín H. Approximating multi-objective hyper-heuristics for solving 2D irregular cutting stock problems. In: Lecture Notes in Computer Science. In Advances in Soft Computing. USA: Springer; 2010. pp. 349-360
 - [38] Miranda G, Armas J, Segura C, León, C. Hyperheuristic codification for the multi-objective 2d guillotine strip packing problem. In: In Proceedings of IEEE Congress on Evolutionary Computation; 2010. pp. 1-8
 - [39] Armas J, Miranda G, and Leòn, C. Hyperheuristic encoding scheme for multiobjective guillotine cutting problems. In: In Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference; 2011
 - [40] Furtuna R, Curteanu S, Leon F. Multi-objective optimization of a stacked neural network using an evolutionary hyper-heuristic. *Applied Soft Computing*. 2012;**12**(1): 133-144
 - [41] Vrugt J, Robinson B. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences*. 2007;**104**(3):708-711
 - [42] Haario H, Saksman E, Tamminen, J. An adaptive metropolis algorithm. *Bernoulli*. 2001;**7**:223-242
 - [43] Storn R, Price K. Differential evolution: A simple and efficient heuristic for global optimization over continuous. *Journal of Global Optimization*. 1997;(11):341-359
 - [44] Raad D, Sinkse A, Vuuren J. Multiobjective optimization for water distribution systemdesign using a hyperheuristic. *Journal of Water Resources Management*. 2010;**136**(5):592-596
 - [45] Zhang X, Srinivasan R, Liew MV. On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the swat model. *Hydrological Processes*. 2010;**24**(8):955-1094
 - [46] Wang Y, Li B. Multi-strategy ensemble evolutionary optimization for dynamic multiobjective optimization. *Memetic Computing*. 2010;**2**:3-24
 - [47] Deb K, Goldberg D. An investigation on niche and species formation in genetic function optimization. In: Proceedings of 3rd International Conference on Genetic Algorithms; 1989. pp. 42-50

- [48] Bäck T. *Evolutionary Algorithms in Theory and Practice*. UK: Oxford University Press; 1996
- [49] Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. US: Wiley; 2001
- [50] Anderson JM, Sayers TM, Bell MGH. Optimisation of a fuzzy logic traffic signal controller by a multiobjective genetic algorithm. *IEEE Road Transport Information and Control*. 2007;454:186-190
- [51] Zhang Q, Li H. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *IEEE Transactions on Evolutionary Computation*. 2007;11(6):712-731
- [52] Li H, Zhang Q. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*. 2009;13(2):284-302
- [53] Li H, Landa-Silva D. An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolutionary Computation*. 2011;19(4):561-595
- [54] Auger A, Bader J, Brockhoff D, Zitzler E. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*. 2012;425:75-103
- [55] Bader J, Zitzler E. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*. 2008;19(1):45-76
- [56] Sutton R, Barto A. *Reinforcement Learning: An Introduction*. USA: MIT Press; 1998
- [57] Van Veldhuizen D. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations [thesis]. Wright-Patterson AFB. Ohio: Air Force Institute of Technology; 1999
- [58] Glover F, Laguna M. Tabu search. In: *Modern Heuristic Techniques for Combinatorial Problems*. USA: John Wiley & Sons; 1995. pp. 70-150
- [59] Kendall G, Mohamad M. Channel assignment in cellular communication using a great deluge hyperheuristic. In: *IEEE International Conference on Network*; 2004. pp. 769-773
- [60] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*. 1999;3(4):253-290
- [61] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*. 1994;2(3):221-248
- [62] Van Veldhuizen DA, Lamont G. Evolutionary computation and convergence to a pareto front. In: *Proceedings of Late Breaking Papers at the Genetic Programming 1998 Conference*; 1998. pp. 221-228
- [63] Coello Coello CA, Cruz Cortés N. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*. 2005;6(2):163-190

