Samarra Journal of Pure and Applied Science

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

Learn heuristics in routing and scheduling problems: A review

Ahmed Abdulmunem Hussein[1]*, Esam Taha Yaseen[2] and Ahmed Noori Rashid[2]
1 College of Islamic Sciences, University of Samarra, Iraq
2 College of Computer Science and Information Technology, University of Anbar, Iraq
https://doi.org/10.54153/sjpas.2023.v5i1.445

Article Information
Received: 09/01/2023
Accepted: 22/02/2023

Keywords:
Metaheuristics, Machine learning, Learnheuristics, Routing problems, Combinatorial optimization problems, and Scheduling problems.

Corresponding Author

Email: ahmed.abd@uosamarra.edu.iq
Mobile: 07710344767

Abstract

Combinatorial optimization problems (COPs) are the most important class of optimization problems, with great practical significance. This class is concerned with identifying the best solution from a discrete set of all available options. The transportation (routing) and distribution (scheduling) systems are considered the most challenging optimization examples of the COPs. Given the importance of routing and scheduling problems, many methods have been proposed to address them. These methods can be categorized into traditional (exact and metaheuristics (MHs) methods) and machine learning (ML) methods. ML methods have been proposed to overcome the problems that traditional methods suffer from, especially high computational time and dependence on the knowledge of experts. Recently, ML methods and MHs have been combined to tackle the COPs, and then the learn heuristics term emerged. This combination aims to guide the MHs toward an efficient, effective, and robust search and improve their performance in terms of solution quality. This work reviews the publications in which the collaboration between MHs and ML has been utilized to propose a guideline for the researchers to put forward new algorithms that have a good ability to solve routing and scheduling problems.

1. Introduction

Achieving the balance between resource reduction and profit increase represents the most important factor in real-life applications. All companies work to achieve that balance, which guarantees their success and continuity. However, this balance is the main challenge that faces the managers of these companies due to the fact that all these applications are complex and highly constrained, and their resources are limited. Consequently, such a balance is very difficult to achieve. COPs are the name given to this type of problem [1]. COPs are the most important class of optimization problems, which are concerned with finding the best solution among a discrete group of all available ones for a specified problem instance. Routing and scheduling problems represent the most challenging optimization problems in the COP, belonging to the NP-hard class of optimization problems that require exponential time to be solved to optimality [2]. So, finding the global optimum for these problems is quite a daunting process. Given the importance of these problems, researchers from various fields have been attracted to solving them, and many algorithms have been proposed to address them [3].

These algorithms can be categorized into exact (exhaustive), heuristic, and MHs algorithms [1]. MHs are computational intelligence paradigms widely used for solving complex optimization problems, particularly COPs. Despite the positive characteristics of MHs and their successful application to various problems, they still suffer from different problems when addressing constraint problems, especially COPs. These problems prevent them from obtaining the best solutions. Several researchers have attempted to address these issues by modifying MHs and combining them with other optimization algorithms. But this hybridization involves adding extra components as well as the COP's components. The appropriate selection of these components has a significant impact on the balance between exploration and exploitation, which is essential for helping the MHs work well for different COPs and at different periods of the search [4].

ML can assist the MHs by extracting useful knowledge from the generated data during the search. Adopting such knowledge during the search makes MHs more intelligent and enhances their performance in terms of solution quality. Recently, there has been a growing research interest in hybridizing and integrating ML with MHs for tackling and solving COPs [4]. This integration aims to guide the MHs toward an efficient, effective, and robust search and improve their performance in terms of solution quality. Since many integration and hybridization methods have been proposed for different purposes, the need to review the recent advances in using ML to improve MHs is still present and urgent [5].

Apart from the literature, this paper provides a comprehensive review of the integration of ML algorithms into MHs algorithms in the following components of MHs: Algorithm selection problem (ASP), Objective function (OF), Initial solution (IS), Operator selection (OS), and Parameter Setting (PS). We considered these components for both routing and scheduling in the literature in recent years. This work provides researchers and students with the most recent published works in this area by providing a collective analysis of both problems and solutions in each publication considered based on the classification offered in the advanced sections of this paper.

This paper is organized as follows: The concepts of the routing problem and the scheduling problem, MHs, and ML are covered in Section 2. Section 3 offers a taxonomy and a search methodology. Algorithm selection is covered in Section 4. Section 5 explains the objective function, while Section 6 gives the details of the initial solution. Operator selection is the main topic of Section 7. The final section presents parameter settings.

2. Preliminaries

2.1. Routing problems

Routing problems are a class of optimization problems with finite search spaces and discrete variables. The main objective of solving routing problems is to design a set of routes that serve a set of geographically distributed points (cities, customers, etc.) at the minimal cost (minimum traveling distance or time) while respecting all the imposed constraints [5]. Routing problems have numerous applications that are applicable in real life. The vehicle routing problem (VRP) and the traveling salesman problem (TSP) are popular problems that have attracted significant attention through the decades [5,6]. Table 1 provides a list of the routing problems that are referred to in this work.

Table 1: A list of routing problems

NO.	Abb.	Problem name	Description
1	LRP [7]	Location-Routing Problem	Sets the locations of facilities and the routes of vehicles to provide a service; some of its constraints are route length and capacity of vehicles.
2	TSP [8]	Traveling Salesman Problem	Finding minimum routes without reaching the visited cities
3	VRP [9]	Vehicle Routing Problem	Cost minimization of the tour to provide demands to set of customers and start with a depot and ends with the same depot
4	CVRP [10]	Capacitated Vehicle Routing Problem	A variant of VRP that deals with the capacity of the vehicles
5	EMVRP [11]	Energy minimizing Vehicle Routing Problem	A variant of GVRP that aims to minimize the energy consumption
6	GVRP [12]	Green Vehicle Routing Problem	A version of VRP that interested in climate and sustainability by reducing CO2 emissions
7	CEVRP [13]	Capacitated Electric Vehicle Routing Problem	A special variant of GVRP that focuses on the capacity of electric vehicles in transportation services

2.2. Scheduling problems

Scheduling problems are another class of optimization problems that focus on reducing the time required to accomplish a certain task in a given time [14]. The researchers paid significant attention to these problems due to their applications in manufacturing, cloud computing, etc. For instance, the Job Shop Problem (JSP) and its variants have been well studied and have been applied to large-scale applications. In contrast, the machine is given a set of tasks that need to be executed in a certain order and within a given time. The objective is to minimize the makespan and obtain the best possible and valid schedule [15]. Table 2 shows the scheduling problems referred to in this work.

Table 2: A list of scheduling problems

NO.	Abb.	Problem name	Description
1	FSP [16]	Flowshop Scheduling Problem	A version of JSP a set of N jobs in a certain sequence with the aim of minimizing the makespan or tardiness
2	JSP [17]	Job-Shop Scheduling	Find a set of N jobs in a certain order with the aim of minimizing the makespan or tardiness
3	MSP [18]	Multiprocessor Scheduling Problem	Minimize the makespan of schedules for network of processors

4	NRP [19]	Nurse Rostering Problem	Finding the best schedules for nurses with the goal of minimize the cost for hospitals and maximize the preferences for nurses
5	PMSP [20]	Parallel Machine Scheduling Problem	Minimize makespan for N job on M parallel machines
6	SMSP [21]	Single-Machine Scheduling Problem	A variant of FSP that deal with only one machine
7	TTP [22]	Timetabling Problem	Finding the best schedule for teacher and rooms to classes to prevent disorder in the classes
8	WSRP [23]	Workforce Scheduling and Routing Problem	Interested in scheduling the workforce to locations and the goal is to minimize the travelling time among locations
9	PLO [24]	Product-line Optimization	Maximize the profit and minimize the cost of productions
10	UPMS [25]	Unrelated parallel machine scheduling problem	A variant of PMSP minimize makespan for N job on M parallel machines each job J available at time zero on M unrelated parallel machine
11	EDSSP [26]	electromagnetic detection satellite scheduling problem	Mapping satellite resources to execute a certain task
12	SBS [27]	School bus scheduling	Find the best schedule for school buses to serve a set of schools

2.3. Metaheuristics

MHs are viewed as an iterative, generating process that drives a subordinate heuristic by mixing intelligently distinct concepts for exploring and exploiting the search space. MHs are characterized as being able to provide near-optimal solutions for a given optimization problem in a reasonable amount of computation time [1]. MHs can be classified in several ways. One of the classifications is **(i) "nature-inspired" and "non-nature-inspired."** Nature-inspired algorithms based on evolutionary algorithms (EA) such as the genetic algorithm (GA) [28], memetic algorithm (MA) [29], and differential evolution (DE) [30] swarm intelligence-based algorithms such as particle swarm optimization (PSO) [31], ant colony optimization (ACO) [32], artificial bee colony (ABC) [33], and bat algorithm (BA) [34]. Non-nature inspired societal algorithms, such as the Imperialist Competitive Algorithm (ICA) [35], music-based algorithms, such as Harmony Search (HS) [36], and physics-based algorithms, such as Simulated Annealing (SA) [37]. The second classification is **(ii) "memory" and "memoryless"** which refer to whether or not memory is used during the search process. GA, SA, etc. are examples of memoryless algorithms. Tabu search (TS) [38], and harmony search (HS) are algorithms that use memory to improve the solution found during the search while avoiding revisiting previously visited solutions. The third MHs classification is **(iii) "stochastic" and "deterministic".** Deterministic means obtaining the same final solution when beginning with the same initial solution, such as TS, while stochastic means the final solution is always different, such as GA and SA [1]. The fourth category is **(iv) "population-based" and "single-solution"**. Single solution algorithms such as iterated local search (ILS) [39], breakout local search (BLS) [40], variable neighbourhood search (VNS) [41], hill climbing (HC) [42], large neighbourhood search (LNS) [43], TS, SA, and etc. As for the population-based algorithms, Water Wave Optimization (WWO) [44], Cuckoo Search (CS) [45], GA, PSO, and etc. are some examples. **(v) "iterative" and "greedy"** are the final classifications. Iterative algorithms that use search operators in each iteration, such as ILS and GA, Greedy algorithms are also called constructive heuristics that generate an initial solution from an empty solution, such as nearest neighbour (NN) [46], greedy heuristics (GH) [47], iterated greedy (IG) [47], and the greedy randomized adaptive search procedure (GRASP) [48].

2.4. Machine learning

ML is a subfield of artificial intelligence (AI) that has received a great deal of attention in recent years and has evolved into powerful techniques for a wide range of applications and problem-solving [49]. The basic objective of ML is to automate systems and extract hidden knowledge from data in order to train the system to carry out tasks in an intelligent and autonomous manner [50]. ML algorithms can be categorized into three categories: supervised learning, unsupervised learning, and reinforcement learning [51].

i) Supervised learning

The first class of ML, supervised learning (Sup.), receives input variable values and produces output variables whose values are already known. Finding the connection between the input and the output and using it to predict the output of a new input is the basic objective of supervised learning. Methods for supervised learning are divided into classification and regression tasks. Artificial Neural Network (ANN) [52], Linear Regression (LR) [53], Support Vector Machine (SVM) [54], Gradient Boosting (GB) [55], Decision Tree (DT) [56], Random Forest (RF) [57], k-Nearest Neighbour (KNN) [58], Naive Bayes (NB) [59], and Convolution Neural Network (CNN) [60] are the most popular algorithms of supervised learning.

ii) Unsupervised learning

Unsupervised learning (Unsup.) is different from supervised learning in that the data is not labelled and the input variables' values are known, but the output variables' values are not. Based on this, the primary objective of unsupervised learning is to apply two well-known approaches, clustering, and association rules (ARs), to discover the hidden patterns in the data. The most popular algorithms of unsupervised learning are K-means [61], C-means [62], Shared Nearest Neighbour Clustering (SNNC) [63], the Apriori algorithm [64], and the Frequent Pattern (FP) algorithm [65].

iii) Reinforcement learning

The third class of ML (ML) is reinforcement learning (RL), which bases its training process on rewarding desired action and punishing undesirable behaviour. The agents repeatedly take actions that decrease or increase the reward. Through a process of trial and error, RL carries out its actions and the learning process. RL algorithms are SARSA [66], Q-Learning [67], Monte Carlo (MC) [68], Proximal Policy Optimization (PPO) [69], Epsilon Greedy (ϵ -G) [70], Thompson Sampling Algorithm (TSA) [71], Upper Confidence Bound 1 (UCB1) [72], and Deep Reinforcement Learning (DRL) [73].

4. Algorithm Selection Problem

Low-level operators make up MHs, which are regarded as high-level schemes [4]. The algorithm selection process involves selecting an algorithm from a provided portfolio of algorithms based on its performance and the problem instances identified after evaluation [80]. According to the no free lunch theorem, no single metaheuristic algorithm surpasses every other algorithm in terms of performance and problem instances; hence, we must agree that there is no such thing as the global best [81]. Which method is the best to solve those COPs? is the question that still needs to be answered [82]. The use of meta-learning, a sub-field of ML, in the algorithm selection problem (ASP) was advantageous [83]. ASP enabled researchers to examine a wide range of MHs and select the best approach from a set of algorithms to use on specific problem situations, resulting in significantly improved performance [84]. In the literature, ML efficiency is frequently mentioned in relation to ASP [85, 86]. The ASP can be divided into offline and online categories [79]. By using occurrence includes, offline learning may be viewed as a model for accumulating information from numerous preparation instances, which will ideally sum up to resolve new instances [87]. On the other hand, the dynamic creation of the online learning model occurred during the optimization process [88].

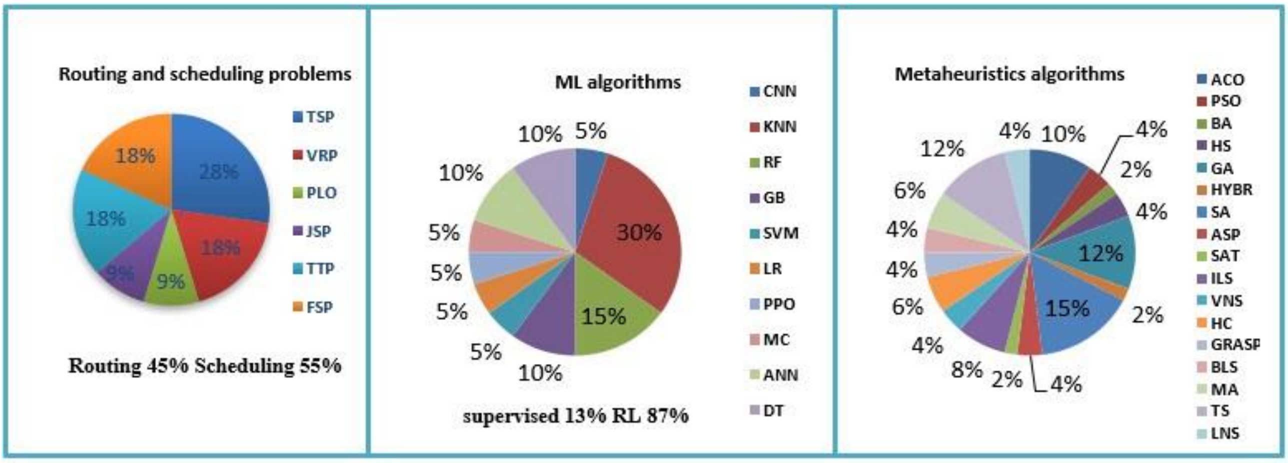
Table 3: Algorithm selection problems in literature.

Ref.	MHs.	ML	Learning	Class	Problem	Rout./sched.
[89] 2022	ACO, PSO, BA,HS,GA	CNN	Offline	Supervised	TSP	Routing
[90] 2021	HYBR, SA, ASP, SAT	KNN, RF,GB,SVM	Offline	Supervised	TTP	Scheduling
[91] 2021	ILS, SA, VNS	LR	Offline	Supervised	TTP	Scheduling
[92] 2021	GA, ACO	PPO	Offline	RL	JSP	Scheduling
[93] 2020	GA, SA, MA	MC	Offline	RL	PLO	Scheduling
[94] 2019	GA, PSO	ANN	Offline	Supervised	VRP	Routing
[95] 2018	TS	RF	Offline	Supervised	AP	Routing
[96] 2018	HC, SA, TS, ILS	DT	Offline	Supervised	FSP	Scheduling
[97] 2018	HC, SA, TS, ILS	GB	Offline	Supervised	FSP	Scheduling
[98] 2017	LNS	KNN	Offline	Supervised	VRP, TSP	Routing
[99] 2016	TS, SA, GA, ACO	ANN, KNN	Offline	Supervised	TSP	Routing

In Table 3, the studied papers are categorized by the various types of MHs that have been used in the literature, the learning environment (offline vs. online), the class of ML (supervised vs. unsupervised vs. reinforcement learning), and the routing and scheduling problems that have been addressed. (Refer to Tables 1 and 2 in sections (2.1) and (2.2), respectively.) There are five papers for routing problems and six papers for scheduling problems pertaining to the problems in Table 3 and Fig. 1a, therefore both of them are fairly discussed in the literature. From Table 3, we deduced two important findings: 1) It appeared that offline learning was more considered than online learning, and 2) The majority of ASP literature uses supervised learning techniques (Fig. 1b). Most of the MHs algorithms studied are used to solve routing and scheduling problems and achieve significant performance (Fig. 1c); Isaas I. Huerta et al. 2022, provide an automatic method selection based on the anytime algorithm for TSP, which is used to handle routing and scheduling problems and obtain substantial performance. By using an input matrix grid and an output staggered representation, CCN's job is to categorize the best MH solver [89]. The researchers' proposed algorithm selection method for the curriculum-based course timetabling problem (CB-CTT) was presented in Arnaud De Coster et al., 2021. The right approach is chosen using four classifiers: support vector machines (SVM), random forest (RF), gradient-boosted trees (GB), and k-nearest neighbour classification (KNN) [90]. Algorithms with a single solution are used in [91]. Felipe de la Rosa-Rivera et al.'s description of the algorithm selection of MHs using LR to solve TTB in 2021 gave rise to this concept. To do this, the authors employed the meta-learning concept. In order to choose heuristics to solve JSP, Tilo van Ekeris et al. presented Proximal Policy Optimization (PPO), a DRL based on reinforcement learning. The DRL agents discover MHs automatically based on several features after learning problem classes through trial and error [92].

4.1. Discussion

Although ASP may offer the best MHs to resolve COPs, it is not always the best option. In this section, we first outline the circumstances in which ASP is helpful before delving into the prerequisites for doing so. The purpose of offering a study guideline for ASP is to aid researchers in better comprehending MHs. When there are only a few computing resources (i.e., time and cores) available to solve a particular problem, learning about ASP is helpful [93]. This is true for optimization problems at the operational level where there is a time limitation. Running all algorithms and selecting the best one is the best option for solving strategic optimization problems [94]. When there are many possible algorithms and little prior understanding of the problem, the typical trial-and-error optimization effort can take the place of ASP. Additionally, ASP can assist non-experts in choosing the best method or algorithm to solve optimization challenges. An adequate training instance pool that accurately represents brand-new examples must be offered for building the meta-model [95]. Instance dissimilarity and algorithmic discrimination are two crucial ASP criteria. The algorithms require a variety of instances in order to provide such a rich collection of data [96]. If there are many instances produced, it could take a long time to execute all candidate algorithms on them. Characterizing the problem instances using a set of criteria, known as meta-features, is a significant challenge in the development of a meta-model. Instance characteristics that have an impact on the algorithm's performance must be revealed by meta-features. Feature extraction [98] can be a computationally inexpensive process for simple features and an expensive one for more complicated ones, depending on the type of features. By switching from problem-specific characteristics to more broad, straightforward features that cost less to extract computationally, ASP can be enhanced [97]. Another interesting area for future investigation is creating online ASPs. The goal is to assure instance dissimilarity by using EAs to evolve COP instances that are as different from one another as possible [99]. As alterations (if necessary) can be made to the algorithm selection mechanism during the resolution of COP instances, the online ASP imposes an additional computational burden. Scheduling numerous suggested algorithms to address a problem could be one future research area. The main emphasis has been on COPs with discrete value-based and permutation-based representations. Fig. 6 refers to the percentage of papers considered in the algorithm selection problem, where the papers were 11 out of 56.



5. Objective Function Evaluation

A frequent MHs search component known as fitness (in the evolutionary community) is called objective function [1]. By modifying the objective function, ML can help MHs by guiding them to better regions of the search space and improving convergence [3, 100]. It may also aid MHs in the approximation of the objective function, which is assessed more quickly than the initial fitness due to the high computational cost of the optimization problem [101, 102]. Surrogate models can be utilized in the optimization process [103, 104, 105]. Surrogate models have been used to evaluate the objective function and explore more promising areas of the search space, which leads to improved convergence [106, 107]. An example of a special case of approximation is evolutionary approximation, which is used in evolutionary algorithms (EAs) to approximate the components of EAs to lower computation costs [108]. It can be divided into two types: fitness inheritance, in which the calculation of the population depends on the similarity with the parents of the individual [109, 110]. The second type is fitness imitation, in which clustering algorithms are included to help calculate the fitness value of the individual from the siblings [111, 112].

Table 4: Objective function Evaluation in the literatures.

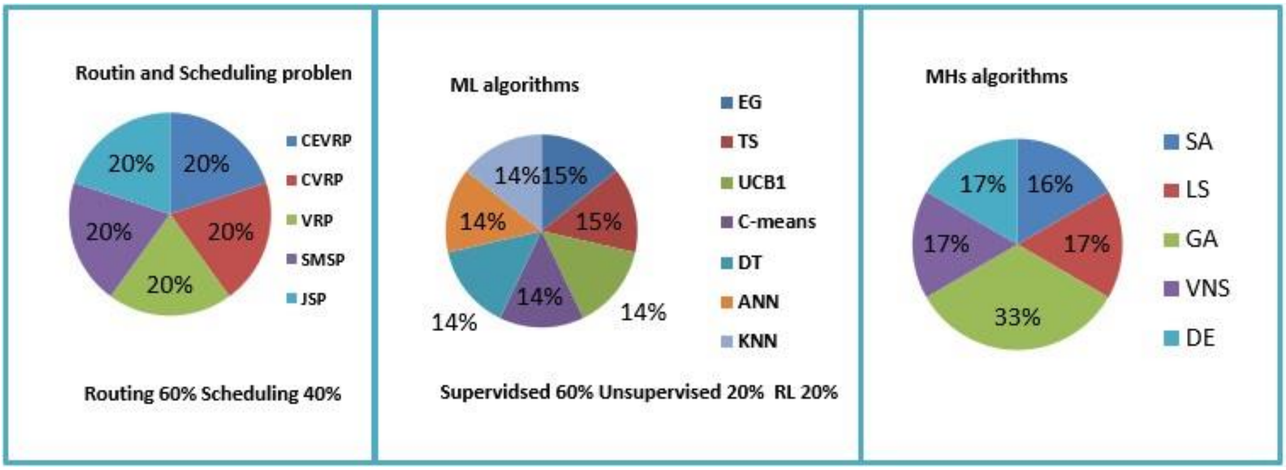
Ref.	MHs.	ML	Single/Multi. Objective	Class	Problem	Rout./Sched.
[113] 2022	SA, HC	ϵ G, TSA,UCB1	Single	RL	CEVRP	Routing
[114]	GA	C-means	Single	Unsupervised	CVRP	Routing

2022						
[115] 2020	VNS	DT	Single	Supervised	VRP	Routing
[116] 2016	DE	ANN	Single	Supervised	SMSP	Scheduling
[117] 2014	GA	KNN	Single	Supervised	JSP	Scheduling

In the objective function evaluation of MHs applied to the routing and scheduling problem, Table 4 reveals and categorizes certain information about ML (Fig. 2b). In the literature that has investigated these two problems, single objectives have been considered over multi-objectives. Both problems are fairly well discussed in the literature; two and three publications, respectively, are devoted to scheduling and routing (Fig. 2a). Three classes of ML (ML) are used for supervised learning, three papers founded in the literature, and one paper each for RL and unsupervised learning (Fig. 2b). To the best of our knowledge, the only routing problems examined in the papers are VRP and its variants. MHs significantly contribute to the resolution of these problems (Fig. 3c). To address the CEVRP, Erick Rodriguez-Esparza et al. proposed RL with SA in 2022. The use of hyper-heuristics (HH) has been shown to improve fitness. The acquired solution then enters a phase of perturbation until it reaches the maximum amount of fitness permitted [113]. Ji Zhu presented enhanced GA using C-means clustering in 2022 to break down CVRP into smaller-scale sub-problems. To improve fitness and reduce the search space for the best solutions, this involves modifying the crossover and mutation rates [114]. New MHs were proposed by Flavien Lucas et al. in 2020, namely feature-guided MNS (FGMNS), which combines LS and ML. The purpose of DT was to find promising regions in the search space and enhance fitness [115]. A hybrid strategy was proposed by Jing-hua Hao et al. in 2016 to address the SMSP problem, utilizing a surrogate model of the ANN and DE algorithms. To evaluate the fitness value, they used DE to enhance ANN performance [116]. Su Nguyen and co. In 2014, researchers looked at how surrogate KNN and GA models could help tackle JSP. Combining two strategies can increase the quality of the solution and provide a better balance between exploration and exploitation [117].

5.1. Discussion

It is not as simple to use objective function approximation in MHs as one might think, and there are difficulties involved. To ensure that MHs with an approximate objective function converge to the near optimum solution of the original function, the original objective function of the MH must be replaced. The optimum method for estimating the fitness function of COPs is still up for debate in the field of fitness approximation. It is possible to utilize more complex methods such as polynomial regression, Kriging, RBF, and clustering/classification [113]. Usually, more complicated systems take longer to construct but offer better fitting accuracy. In MHs, objective approximation can be expanded to allow for the creation of fresh objective functions depending on details about the current optimization challenge and features gleaned from areas visited during the search process. Real-time COPs are ones that must be regularly solved within a time constraint. For these MHs, even one MH iteration's worth of calculation time might be too much for real-time applications. To reduce the computational work required for objective function evaluation, one potential path is to use objective approximation. Fig. 6 refers to the percentage of papers considered in objective function evaluation, where the papers were 5 out of 56.



a b c

Fig. 2 *Objective function evaluation; (a) Routing and Scheduling problems in the Literature; (b) ML class and techniques in Literature; (c) MHs in the literature.*

6. Initial solution

Initialization of several MHs' solutions occasionally happens at random [118]. The crucial issue the researchers are dealing with is the first solution's inadequate initialization, which may have an adverse effect on the solution's diversification [119]. Three initialization types are: hybrid initialization, greedy initialization, and random initialization [1]. In the previous two decades, ML has proven to be a useful initialization strategy for scholars. In offline learning approaches, ML can help MHs by extracting knowledge to produce solutions for valid patterns [120]. Sometimes, that knowledge can be used to help solve problems that are similar to ones that have already been solved, which is known as transfer learning [121]. The decomposition of the search space or data is another application for applying a variety of applications [122]. By separating the data into smaller sub-spaces, decomposition has a nontrivial role in reducing the computing cost of the process and facilitating the generation of better first solutions [123]. Decomposition is used in search space to increase the variety of the initial solution, which will result in diverse solutions from various search space regions and prevent the search space from becoming stuck in local optima [124].

Table 5: Initial solution in the literature

Ref.	MHs.	ML	Single/Multi. Obj.	Class	Problem	Rout./Sched.
[125] 2022	ACO	SARSA	Multi.	RL	TSP	Routing
[126] 2022	GRASP	RF	Single	Supervised	CVRP	Routing
[127] 2021	ABC	K-means	single	Unsupervised	UPMSP	Scheduling
[128] 2020	DE	K-means	Single	Unsupervised	TSP	Routing
[129] 2019	GA,PSO	Apriori	Single	Unsupervised	JSP	Scheduling
[130] 2019	TS	K-means	Single	Unsupervised	VRP	Routing

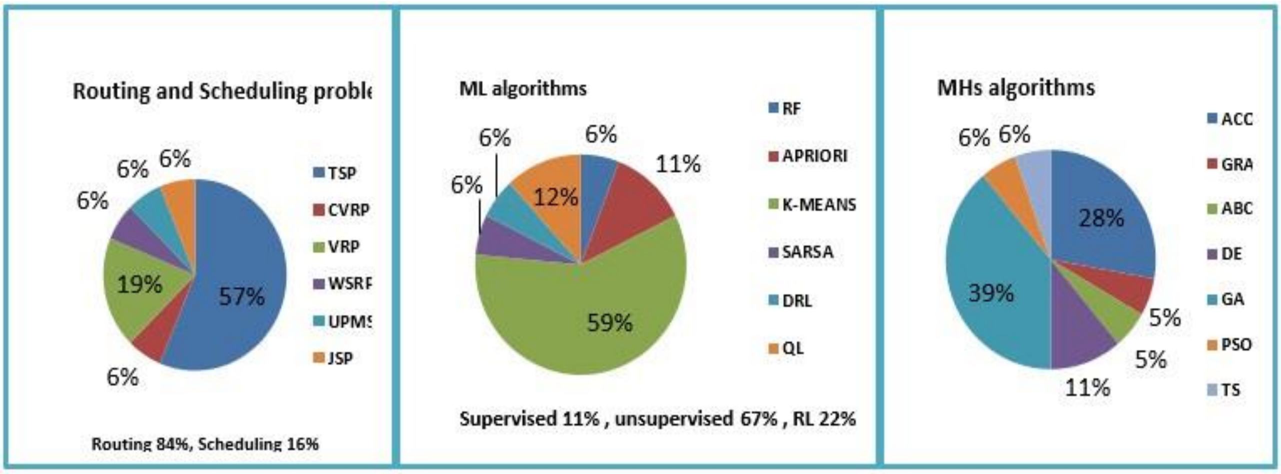
[131]	2019	GA	K-means	Single	Unsupervised	VRP	Routing
[132]	2019	GA,DE	K-means	Single	Unsupervised	TSP	Routing
[133]	2018	MHs	DRL	Single	RL	TSP	Routing
[134]	2018	GA	QL	Single	RL	TSP	Routing
[135]	2018	ACO	K-means	Single	Unsupervised	WSRP	Scheduling
[136]	2018	GA	LR	Single	Supervised	TSP	Routing
[137]	2017	ACO	K-means	Single	Unsupervised	TSP	Routing
[138]	2017	MHs	QL	Single	RL	TSP	Routing
[139]	2017	ACO	K-means	Single	Unsupervised	VRP	Routing
[140]	2016	ACO	K-means	Single	Unsupervised	LRP	Routing
[141]	2016	GA	Apriori	Single	Unsupervised	TSP	Routing
[142]	2015	GA	K-means	Single	Unsupervised	TSP	Routing

The studies on this topic are summarized in Table 5. ML was helpful in the initialization of the MHs. Except for one paper that looked at a multi-objective problem, the majority of publications only considered single objective when solving the problem at hand. There are only three types of scheduling problems, and many researchers are more interested in routing problems than scheduling problems (Fig. 3a). Unsupervised learning algorithms are used more frequently than other classes in ML algorithms, as shown by Table 5 and Fig. 3b, and the majority of these algorithms employ clustering techniques (Fig. 3d), supporting the claim that the initialization of MHs significantly depends on the problem instances. With the help of ML, MH's algorithms (Fig. 3c) produced effective initialization techniques that outperformed the traditional ones. In order to solve TSP, Haitong Zhao et al., 2022, presented ACO with the SARSA algorithm. Their work is based on the theory that by initializing the pheromone using a Q-table and choosing the largest reward received, the convergence performance can be improved [125]. To tackle CVRP, Juan Pablo Mesa et al. proposed RF and GRASP in 2022. In order to simplify the setup of LS and enhance the multi-start MHs (MSM) used to solve the problem, RF's role is to extract features from the domain of the problem [126]. In their investigation, Chen-Yang Cheng et al. (2021) demonstrate the viability of employing the unsupervised K-means clustering technique on UPMSP using the ABC algorithm. This prevents local optimal solutions and improves the population's initialization method [127]. Ismail M. Ali et al.'s discrete DE with K-means clustering for solving TSP is presented in their 2020 paper. In order to improve the population's initialization, this work tries to map continuous variables to discrete variables and vice versa [128]. In their 2019 paper, Mohammad Mahdi Nasiri et al. established association rules to extract JSP scheduling rules and then applied these rules to heuristically solve the problem using GA and PSO. For population-based approaches, the resulting solution may be regarded as the initial solution [129].

6.1. Discussion

The way an MHs is initialized significantly affects its capacity for exploration and exploitation. The diversity of solutions can be preserved while producing initial solutions of high-quality using ML approaches [130]. A complete solution can be created from an incomplete one using ML techniques. Additionally, they are utilized to create partial initial solutions and divide the data space into smaller portions. In search space decomposition, ML approaches are used to diversify the initial solutions for each sub-space. Online and offline learning are also possible. Knowledge is collected during offline learning from the initial solutions produced for a set of training instances [131,132]. Online learning entails the extraction and application of dynamic knowledge while producing the initial solution(s) to a problem instance [133]. Even though the retrieved knowledge may not be particularly rich, it perfectly fits the current case.

In order to improve performance in terms of the trade-off between exploration and exploitation, ML approaches are being incorporated into MHs. Because of this phenomenon, less computing time is spent throughout the search process looking for intial (local) solutions and more time is spent exploring and exploiting more promising regions in the solution space. Instead of expending considerable work on locating the best local solutions, this saves computing effort and directs it toward exploring or exploiting more promising regions in the solution space. The initial population [134] of COPs can be produced using effective opposition-based learning tactics. This is a novel idea in ML that was motivated by the opposing interactions between things. An initial population can also be created using an interpolation technique. Interpolation and opposition-based learning have not yet been used in COPs, but this could be a future research area. Fig. 6 refers to the percentage of papers considered in the initial solution, where the papers were 18 out of 56.



a b c

Fig. 3 *Initial Solution; (a) Routing and Scheduling problems in the Literature; (b) ML class and techniques in Literature; (c) MHs in the literature.*

7- Operator selection

Low-level MH components can be thought of as operator selection [143]. Operators' primary characteristic is that they can be effective for one search method or problem but ineffective for another [144,1]. Search operators are divided into the following categories: mutation, binary operators (such as crossover), perturbation, local search (hill climbing), and ruin recreate. By properly implementing search operators, it may be possible to achieve a balance between exploitation and exploration during the search process [145]. While a search is being conducted in an online learning environment, the performance of MHs can be assessed using various search operators [146]. When ML is included in MHs, for example, RL is involved in operator selection by treating the operators as a problem space and taking various actions to select better operators [147]. The best neighbourhood at a certain iteration can be chosen using ML [148].

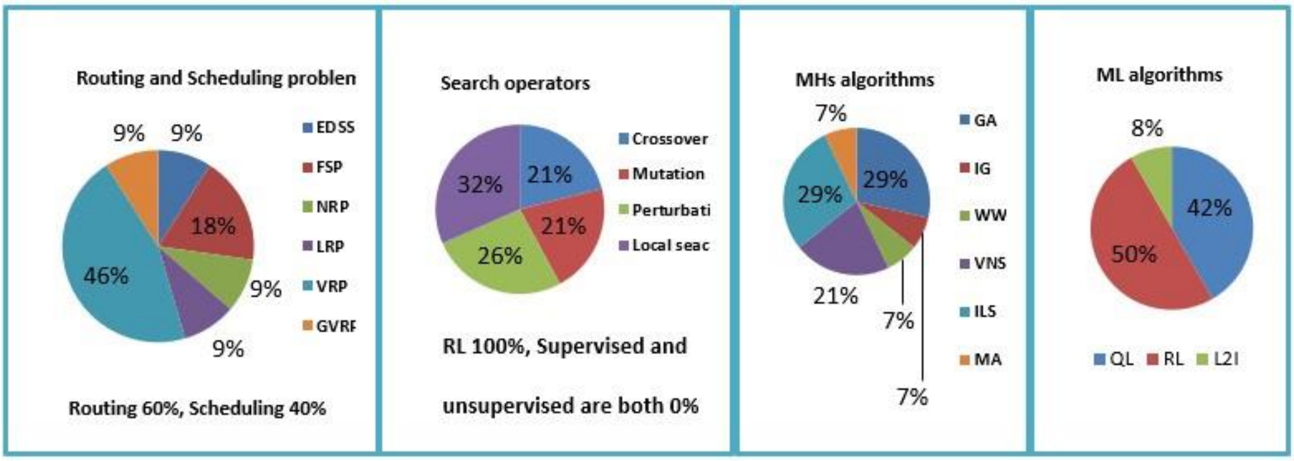
Table 6: Operator selection in the literatures

Ref.	MHs.	ML	Search operator	Class	Problem	Rout./Sched.
[149]2022	GA	QL	Crossover, mutation	RL	EDSSP	Scheduling
[150]2022	IG	QL	Perturbation	RL	FSP	Scheduling
[151]2021	WWO, VNS	QL	Perturbation	RL	FSP	Scheduling
[152]2019	ILS	L2I	Local search	RL	VRP	Routing
[153]2019	MA, ILS	QL	Local search	RL	GVRP	Routing
[154]2017	ILS	RL	Local search, Perturbation	RL	NRP	Scheduling
[155]2016	GA	RL	Perturbation, Crossover, mutation	RL	LRP	Routing
[156]2016	VNS	RL	Local search	RL	VRP	Routing
[157]2016	VNS	RL	Local search	RL	VRP	Routing
[158]2015	ILS	RL	Local search, Perturbation	RL	VRP	Routing

Operator selection is classified in Table 6 based on MHs and ML methods that have been studied in the literature, search operators employed throughout the search process, ML classification (see Section 2.4), and problem type. Table 6 demonstrates that all of the studies under consideration used the RL class of ML. The variance in search operators that can be attributed to the variations in the several MHs under study can also be observed in Table 6 and Fig. 4b. To the best of our knowledge, Table 6 and Fig 4a show that scheduling problems receive more attention in 2022 than routing problems, whereas the converse was apparent in the last decade. The majority of ML (Fig. 4d) algorithms use QL techniques because of their <action, reward> capabilities. Selecting the right operators is critical for directing the search space to the best possible solutions. Several MHs (Fig. 4c) have a variety of operators. To address EDSSP, Yanjie Song et al. (2022) presented a QL and GA framework. They added an operator selection process in their study that updated the Q value in the Q-table based on the cumulative reward attribute of QL [149]. The (PFSP) was addressed by Maryam Karimi-Mamaghan et al. in 2022 utilizing a novel perturbation mechanism and a Q-learning (QL) technique (JSP). The purpose of integrating (QL) into (IG) is to offer adaptive operator selection, which can define which operator should be used for each iteration of the search stages. The selection depends on the status of the search and the operator's performance. In other words, the integration enabled the algorithm to avoid becoming trapped in a local optimum and provided various strengths for perturbation operators [150]. A model to solve the FSP using WWO and QL was presented by Fuqing Zhao et al. in 2021. To improve the balance between exploration and exploitation and prevent being stranded in local optima, the researchers used VNS as a breaker operator when choosing operators [151]. The L2I technique with ILS was suggested by Hao Lu et al. in 2019 to solve CVRP. Using RL-based improvement operators, L2I learns to iteratively enhance the solution. A pool of operators is used to choose the operators [152]. In their study, Bo Peng et al. 2019 used MA population based with QL to address GVRP. To provide workable systems with a good balance between exploration and exploitation, they combined ILS with RL to manage neighbourhood moves and crossover operators [153].

7.1. Discussion

Selecting and using these operators within an MHs requires a high level of domain understanding. However, in order to choose the operators effectively for COPs outside of this common framework, one must be familiar with the operators that are unique to the given situation. The most popular RL-based approach utilized in operator selection is QL. One must first specify the range of potential states and actions before applying them. The states can be defined in three different ways, including search-dependent, problem-dependent, and instance-dependent. When compared to the non-learning version of MHs, ML approaches in operator selection have significantly improved performance. Both the quality of the solutions and the computational efficiency have improved as a result of the incorporation of ML approaches into operator selection. New best-known solutions for a few COPs have also been found. The purpose of operator selection is to offer an optimal design for MHs such that the MH converges more quickly to the near optimal solution and, as a result, the user's time is preserved and the additional computational burden of learning in MHs minimized. Before using operator selection, the majority of credit assignment and selection methods add new factors that need to be modified. Tuning the discount factor and learning rate responsible for creating a balance between MHs' exploration and exploitation capabilities is a significant problem in operator selection. The performance of operator selection may decline as the number of operators rises. Fig. 6 refers to the percentage of papers considered in operator selection, where the papers were 10 out of 56.



a **b** **c** **d**

Fig. 4 *Operators Selection; (a) Routing and Scheduling problems in the Literature; (b) Search operators in Literature; (c) MHs in the literature; (d) ML techniques in the literature.*

8- Parameter setting

The settings of MHs parameters, which govern how the algorithm behaves, have a significant impact on it [159]. Any parameter's value has an impact on the search process, so setting them correctly will result in the best performance [160, 161]. Manual parameter setting by MHs experts is difficult and time-consuming, but it is done in accordance with their level of knowledge [162]. Parameter setting categorized into parameter control and parameter tuning

[1]. Setting the parameters prior to the start of the optimization process is known in the literature as offline parameter tuning. Online parameter setting, sometimes referred to as parameter control, modifies the value of parameters throughout the optimization process in accordance with certain criteria [163, 164]. To assist MHs in choosing the appropriate values for the parameters, ML is used in parameter setting. Based on the action reward functionality and addressing the problem space of MH's algorithms, RL was used to set the parameter values [165]. In fact, parameter setting is comparable to AOS (section 5) when feedback data is taken into account when choosing parameter values.

Table 7: Parameter setting in the literature

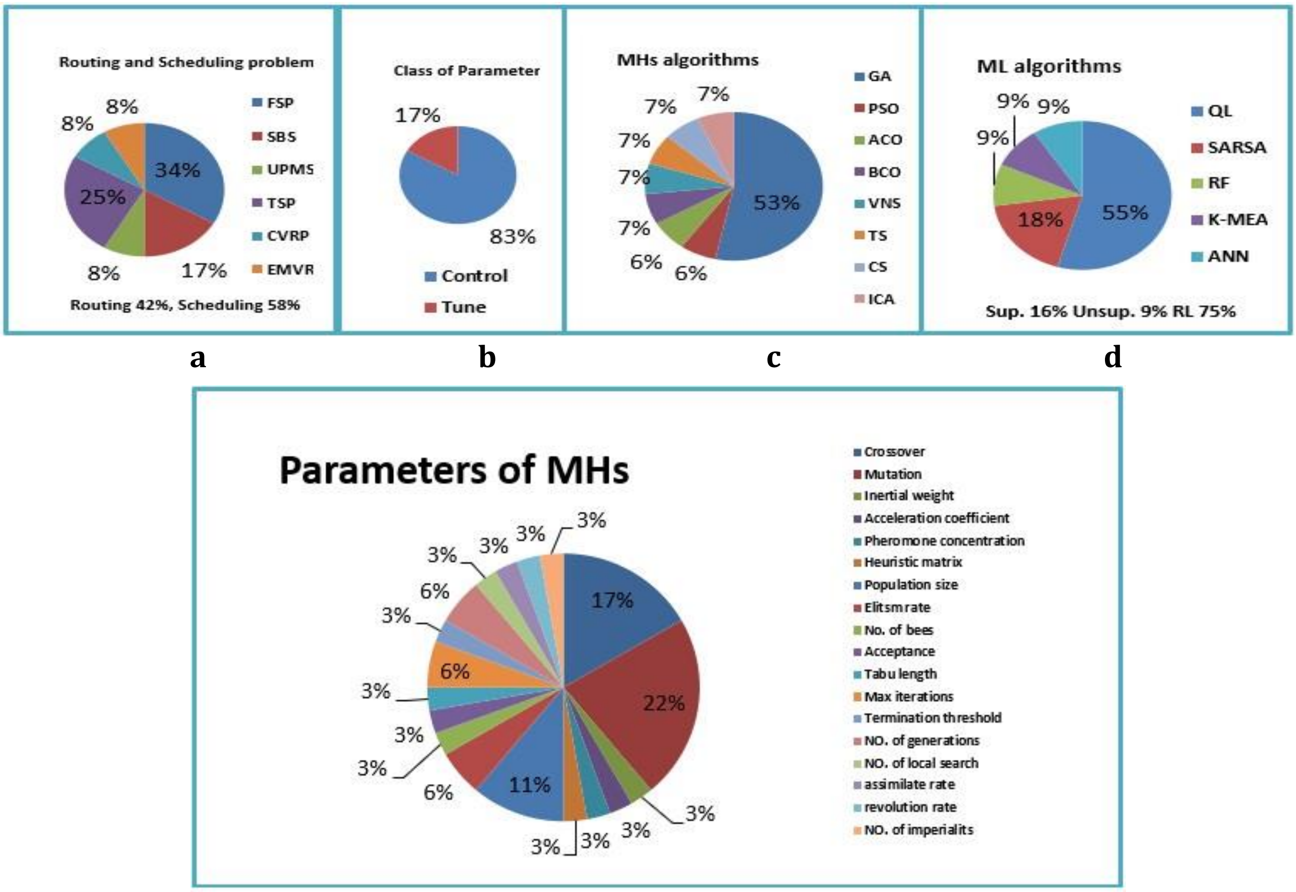
Ref.	MHs	ML	Tuning /Control	Parameter	Class	Problem	Rout./Sched.
[166] 2022	GA	QL	Control	Crossover, Mutation	RL	FSP	Scheduling
[167] 2022	GA	SARSA	Control	Crossover, Mutation	RL	SBS	Scheduling
[168] 2021	PSO, ACO ,GA	QL	Control	Crossover, Mutation, inertia weight, acceleration coefficient, pheromone concentration, heuristic matrix	RL	SBSR	Scheduling, Routing
[169] 2021	GA	QL	Control	population size, mutant rate, elitism rate,	RL	TSP	Routing
[170] 2021	GA	QL	Tune	Crossover, Mutation	RL	CVRP	Routing
[171] 2021	ABC	RF	Control	Number of bee	Supervised	TSP	Routing
[172] 2020	VNS	QL	Control	Acceptance	RL	FSP	Scheduling
[173] 2020	GA	QL	Control	Crossover, Mutation	RL	FSP	Scheduling
[174] 2020	TS	DRL	Tune	Tabu length, Max Iterations, Termination threshold	RL	TSP	Routing
[175] 2020	CS	SARSA	Control	Population size, Max Iterations	RL	FSP	Scheduling
[176] 2017	GA	K-means	Control	Number of generation, Mutation, Population size	Unsupervised	EMVRP	Routing

[177] 2017	GA, ICA	ANN	Control	population size, number of generations, crossover rate, mutation rate, elitism rate, number of local search, assimilate rate, revolution rate, number of imperialists	Supervised	UPMS	Scheduling

Due to its feedback information, RL (Fig. 5d), the class of ML that is most frequently utilized in the literature, is listed in Table 7. In comparison to offline parameter setting, online parameter setting (control) appears more in Table 7 and Fig. 5b in the majority of the publications under consideration. Routing and scheduling problems have each been considered separately, and the research direction has taken both concerns into account (Fig. 5a). Table 7 and Fig. 5e show that there are several parameters that can be adjusted depending on the many MHs that have been identified in the literature. In terms of MHs algorithms (Fig. 5c), GA is the algorithm that has been employed to solve the issues the most frequently. In order to tackle the school bus routing and scheduling problem (SBRS), Eda Koksal and colleagues published a performance characterization for RL-enabled evolutionary algorithms (EAs) dubbed GA, Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). Because EAs are sensitive to the values of the parameters, RL is employed to select the parameters' values. RL provided new means for exploring and utilizing the search space, allowing the EAs to manage hyper-parameters. Crossover and mutation rates are the parameters that are regulated in GA; inertia weight and acceleration coefficient are controlled in PSO; and pheromone concentration and heuristic matrix are controlled by RL in ACO [168]. In order to solve TSP, Antonio Augusto Chaves and Luiz Henrique Nogueira Lorena (2021) proposed combining Q-Learning with The Biased Random-key genetic algorithm (BRKGA). The parameter control of BRKGA based on QL is the main topic of this work. For BRKGA, the following variables are controlled: population size, elites, mutation rate, and probability of inheriting a key from an elite parent. Each parameter has a single state, and the operation is predetermined by the Q-table [169]. GA and RL were suggested by Jose Quevedo et al. in 2021 to address the problem of capacitated vehicle routing (CVRP). Their method is based on the idea of solving the CVRP by adaptively adjusting the GA and RL parameters. To execute actions that influence how GA evolves, RL agents engage with GA as its environment. The agents of RL were used to create a new population with the associated reward using mutation and crossover probabilities that were tuned based on feedback from the evolutionary process that is the GA environment [170].

8.1. Discussion

The capacity of an MHs to explore and utilize promising search space regions has a substantial impact on the MH's performance. The crossover and mutation rates, using the GA as an example, should vary according to the algorithm's effectiveness and the characteristics of the search space. Despite the computational overhead put on the search process, parameter management is advised [178]. The trade-off between choosing the existing best configuration or looking for good new ones is the choice between exploration and exploitation. Another characteristic that needs to be adjusted or managed during the search process is the rate of configuration change itself. As one approach to solving this, consider parameter setting as a separate optimization problem. Creating a self-adaptive parameter control mechanism is an additional option. The precision of the intervals would be higher if the levels of a parameter were split into numerous small intervals but choosing between the intervals would require substantially more calculation [179]. To fully comprehend how minor variations in the values of continuous parameters affect MHs performance, more research is necessary. Future studies should concentrate on creating a method to regulate SA using variables like the cooling temperature and others that haven't gotten much attention up to now. It should be mentioned that defining the feedback as a single value is one of the difficulties with adaptive parameter management in multi-objective optimization problems. Fig. 6 refer to the percentage of papers considered in parameter setting, where the papers were 12 out of 56.



e

Fig. 5 *Parameter Setting; (a) Routing and Scheduling problems in the Literature; (b) ML class in Literature; (c) MHs in the literature; (d) ML techniques in the literature; (e) Parameter of MHs*

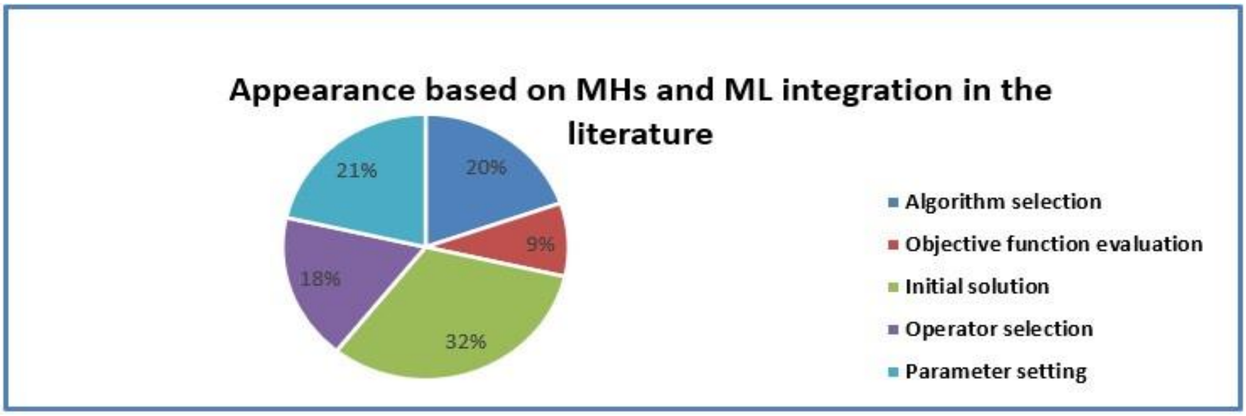


Fig. 6 *Learnheuristics in the literature*

Conclusion

Learnheuristics have gained substantial attention in place of more conventional approaches since they enhance the performance of MHs. We have provided a thorough taxonomy for several MHs components managed via ML, including operator selection, parameter control and tuning, initial solution, algorithm selection, and objective function evaluation to solve COPs which is a complex class of optimization problems. In fact, the integration of ML algorithms into MHs algorithms has extra effects when solving different COPs rather than solving the same COPs in traditional ways, where the components of MHs managed based on the knowledge of the experts, which is time consuming. When an ML technique is integrated into an MH, a new set of parameters is added that must be properly tuned or controlled to achieve the best results. To overcome this issue, dynamically adjusting the additional parameters based on the integration's features. In large scale problems (e.g., routing, and scheduling problems), the training of ML on problem instances corresponds to a specific size, which requires larger instances during the training, but this approach may be time consuming. Most of the studies examined in this paper only focus on the integration of ML approaches into MHs for a singular objective, although improved MH performance is expected when ML techniques serve MHs for several objectives. As a result, incorporating ML approaches with MHs for multiple objectives at the same time could be an attractive future study direction. The algorithm selection problem depends heavily on supervised learning in deciding which algorithms perform well in particular routing and scheduling problems, while unsupervised learning and RL need to be investigated more. The majority of studies that investigated routing and scheduling problems took a single objective into account for objective function evaluation and initial solution, while multi-objective optimization (MOP), as previously mentioned, was poorly considered by the researchers. Operator selection offered an optimal design for MHs such that the MHs converged more quickly to the near optimal solution and, as a result, the time preservation and additional computational burden of learning in MHs relieved. The parameter variables are compiled when ML algorithms are integrated with MH algorithms, so those parameters must independently determine their values. Most of the examined articles concentrate on using well-known ML methods like K-means, KNN, SVM, etc., which makes it possible to look into additional algorithms. In terms of most widespread use, MHs algorithms were similar to ML algorithms; GA, PSO, and ACO, among others, regularly appear in the literature, albeit there are alternative options to take into account.

References

1. Talbi, E. G. (2009). Metaheuristics: from design to implementation. John Wiley & Sons.
2. Michalewicz, Z., & Fogel, D. B. (2013). How to solve it: modern heuristics. Springer Science & Business Media.
3. Calvet, L., de Armas, J., Masip, D., & Juan, A. A. (2017). Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. Open Mathematics, 15(1), 261-280.
4. Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E. G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. European Journal of Operational Research, 296(2), 393-422.

5. Castaneda, J., Ghorbani, E., Ammouriova, M., Panadero, J., & Juan, A. A. (2022). Optimizing Transport Logistics under Uncertainty with Simheuristics: Concepts, Review and Trends. Logistics, 6(3), 42.
6. Shi, Y., & Zhang, Y. (2022). The neural network methods for solving Traveling Salesman Problem. Procedia Computer Science, 199, 681-686.
7. Laporte, G. (1987). Location-routing problems.
8. Flood, M. M. (1956). The traveling-salesman problem. Operations research, 4(1), 61-75.
9. Toth, P., & Vigo, D. (Eds.). (2002). The vehicle routing problem. Society for Industrial and Applied Mathematics.
10. Ralphs, T. K., Kopman, L., Pulleyblank, W. R., & Trotter Jr, L. E. (2001). On the capacitated vehicle routing problem.
11. Kara, I., Kara, B. Y., & Yetis, M. K. (2007, August). Energy minimizing vehicle routing problem. In International Conference on Combinatorial Optimization and Applications (pp. 62-71). Springer, Berlin, Heidelberg.
12. Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. Transportation research part E: logistics and transportation review, 48(1), 100-114.
13. Jia, Y. H., Mei, Y., & Zhang, M. (2021). A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem. IEEE Transactions on Cybernetics.
14. Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The non-permutation flow-shop scheduling problem: a literature review. Omega, 77, 143-153.
15. Gür, Ş., & Eren, T. (2018). Scheduling and planning in service systems with goal programming: Literature review. Mathematics, 6(11), 265.
16. Gupta, J. N. (1988). Two-stage, hybrid flowshop scheduling problem. Journal of the operational Research Society, 39(4), 359-364.
17. Manne, A. S. (1960). On the job-shop scheduling problem. Operations research, 8(2), 219-223.
18. Hou, E. S., Ansari, N., & Ren, H. (1994). A genetic algorithm for multiprocessor scheduling. IEEE Transactions on Parallel and Distributed systems, 5(2), 113-120.
19. Burke, E., De Causmaecker, P., & Vanden Berghe, G. (1999). A hybrid tabu search algorithm for the nurse rostering problem. In Asia-Pacific Conference on Simulated Evolution and Learning (pp. 187-194). Springer, Berlin, Heidelberg.
20. Cheng, T. C. E., & Sin, C. C. S. (1990). A state-of-the-art review of parallel-machine scheduling research. European Journal of Operational Research, 47(3), 271-292.
21. Biskup, D. (1999). Single-machine scheduling with learning considerations. European Journal of Operational Research, 115(1), 173-178.
22. Welsh, D. J., & Powell, M. B. (1967). An upper bound for the chromatic number of a graph and its application to timetabling problems. The Computer Journal, 10(1), 85-86.
23. Castillo-Salazar, J. A., Landa-Silva, D., & Qu, R. (2014). Computational study for workforce scheduling and routing problems.
24. Yano, C., & Dobson, G. (1998). Profit-optimizing product line design, selection and pricing with manufacturing cost consideration. In Product Variety Management (pp. 145-175). Springer, Boston, MA.

25. Li, Z., Yang, H., Zhang, S., & Liu, G. (2016). Unrelated parallel machine scheduling problem with energy and tardiness cost. The International Journal of Advanced Manufacturing Technology, 84(1), 213-226.
26. Chen, H., Li, J., Jing, N., & Tang, Y. (2008, November). Hybrid algorithms for electromagnetic detection satellites scheduling. In 2008 20th IEEE International Conference on Tools with Artificial Intelligence (Vol. 1, pp. 411-418). IEEE.
27. Kim, B. I., Kim, S., & Park, J. (2012). A school bus scheduling problem. European Journal of Operational Research, 218(2), 577-585.
28. Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.
29. Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826, 1989.
30. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.
31. Zomaya, A. Y. (Ed.). (2006). Handbook of nature-inspired and innovative computing: integrating classical models with emerging technologies. Springer Science & Business Media.
32. Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical computer science, 344(2-3), 243-278.
33. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
34. Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.
35. Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661-4667). Ieee.
36. Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. simulation, 76(2), 60-68.
37. Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.
38. Glover, Fred, and Manuel Laguna. "Tabu search." In Handbook of combinatorial optimization, pp. 2093-2229. Springer, Boston, MA, 1998.
39. Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search. In Handbook of metaheuristics (pp. 320-353). Springer, Boston, MA.
40. Benlic, U., Epitropakis, M. G., & Burke, E. K. (2017). A hybrid breakout local search and reinforcement learning approach to the vertex separator problem. European Journal of Operational Research, 261(3), 803-818.
41. Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. Computers & operations research, 24(11), 1097-1100.
42. Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (1988). How easy is local search?. Journal of computer and system sciences, 37(1), 79-100.

43. Shaw, P. (1998, October). Using constraint programming and local search methods to solve vehicle routing problems. In International conference on principles and practice of constraint programming (pp. 417-431). Springer, Berlin, Heidelberg.
44. Zheng, Y. J. (2015). Water wave optimization: a new nature-inspired metaheuristic. Computers & Operations Research, 55, 1-11.
45. Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). Ieee.
46. Gutin, G., Yeo, A., & Zverovich, A. (2002). Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discrete Applied Mathematics, 117(1-3), 81-86.
47. Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European journal of operational research, 177(3), 2033-2049.
48. Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. Journal of global optimization, 6(2), 109-133.
49. Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2(3), 1-21.
50. Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. Neurocomputing, 237, 350-361.
51. Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
52. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), e00938.
53. Weisberg, S. (2005). Applied linear regression (Vol. 528). John Wiley & Sons.
54. Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.
55. Friedman, J. H. (2002). Stochastic gradient boosting. Computational statistics & data analysis, 38(4), 367-378.
56. Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.
57. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
58. Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.
59. Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).
60. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.
61. Krishna, K., & Murty, M. N. (1999). Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 29(3), 433-439.
62. Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & geosciences, 10(2-3), 191-203.
63. Ertoz, L., Steinbach, M., & Kumar, V. (2002, April). A new shared nearest neighbor clustering algorithm and its applications. In Workshop on clustering high dimensional data and its applications at 2nd SIAM international conference on data mining (Vol. 8).

64. Hegland, M. (2007). The apriori algorithm–a tutorial. Mathematics and computation in imaging science and information processing, 209-262.
65. Borgelt, C. (2005, August). An Implementation of the FP-growth Algorithm. In Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations (pp. 1-5).
66. Sprague, N., & Ballard, D. (2003). Multiple-goal reinforcement learning with modular sarsa (0).
67. Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3), 279-292.
68. Bouzy, B., & Chaslot, G. (2006, May). Monte-Carlo Go reinforcement learning experiments. In 2006 IEEE symposium on computational intelligence and games (pp. 187-194). IEEE.
69. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
70. Stadie, B. C., Levine, S., & Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. arXiv preprint arXiv:1507.00814.
71. Gopalan, A., Mannor, S., & Mansour, Y. (2014, January). Thompson sampling for complex online problems. In International conference on machine learning (pp. 100-108). PMLR.
72. Auer, P. (2000, November). Using upper confidence bounds for online learning. In Proceedings 41st Annual Symposium on Foundations of Computer Science (pp. 270-279). IEEE.
73. Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.
74. Kiranyaz, S., Ince, T., & Gabbouj, M. (2014). Multidimensional particle swarm optimization for machine learning and pattern recognition (p. 383). Berlin: Springer.
75. Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM computing surveys (CSUR), 35(3), 268-308.
76. Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. IEEE Transactions on systems, man, and cybernetics, 16(1), 122-128.
77. Aine, S., Kumar, R., & Chakrabarti, P. P. (2009). Adaptive parameter control of evolutionary algorithms to improve quality-time trade-off. Applied Soft Computing, 9(2), 527-540.
78. Gambella, C., Ghaddar, B., & Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. European Journal of Operational Research, 290(3), 807-828.
79. Talbi, E. G. (2021). Machine learning into metaheuristics: A survey and taxonomy. ACM Computing Surveys (CSUR), 54(6), 1-32.
80. Sabar, N. R., Ayob, M., Kendall, G., & Qu, R. (2014). Automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems. IEEE Transactions on Evolutionary Computation, 19(3), 309-325.

81. Joyce, T., & Herrmann, J. M. (2018). A review of no free lunch theorems, and their implications for metaheuristic optimisation. Nature-inspired algorithms and applied optimization, 27-51.
82. Rice, J. R. (1976). The algorithm selection problem. In Advances in computers (Vol. 15, pp. 65-118). Elsevier.
83. Kotthoff, L. (2014). Algorithm Selection for Combinatorial Search Problems: A Survey.
84. Kotthoff, L. (2016). Algorithm selection for combinatorial search problems: A survey. In Data mining and constraint programming (pp. 149-190). Springer, Cham.
85. Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Fréchette, A., ... & Vanschoren, J. (2016). Aslib: A benchmark library for algorithm selection. Artificial Intelligence, 237, 41-58.
86. Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2008). Metalearning: Applications to data mining. Springer Science & Business Media.
87. Miranda, E. S., Fabris, F., Nascimento, C. G., Freitas, A. A., & Oliveira, A. C. (2018, October). Meta-learning for recommending metaheuristics for the maxsat problem. In 2018 7th Brazilian conference on intelligent systems (BRACIS) (pp. 169-174). IEEE.
88. Gagliolo, M., & Schmidhuber, J. (2010, January). Algorithm selection as a bandit problem with unbounded losses. In International conference on learning and intelligent optimization (pp. 82-96). Springer, Berlin, Heidelberg.
89. Huerta, I. I., Neira, D. A., Ortega, D. A., Varas, V., Godoy, J., & Asín-Achá, R. (2022). Improving the state-of-the-art in the Traveling Salesman Problem: An Anytime Automatic Algorithm Selection. Expert Systems with Applications, 187, 115948.
90. De Coster, A., Musliu, N., Schaerf, A., Schoisswohl, J., & Smith-Miles, K. (2022). Algorithm selection and instance space analysis for curriculum-based course timetabling. Journal of Scheduling, 25(1), 35-58.
91. de la Rosa-Rivera, F., Nunez-Varela, J. I., Ortiz-Bayliss, J. C., & Terashima-Marín, H. (2021). Algorithm selection for solving educational timetabling problems. Expert Systems with Applications, 174, 114694.
92. Tilo, R., & Meyes, T. (2021). Discovering Heuristics And Metaheuristics For Job Shop Scheduling From Scratch Via Deep Reinforcement Learning. 2701–6277.
93. Vökler, S., & Baier, D. (2020). Investigating Machine Learning Techniques for Solving Product-line Optimization Problems. Archives of Data Science, 6, 1–11.
94. Gutierrez-Rodríguez, A. E., Conant-Pablos, S. E., Ortiz-Bayliss, J. C., & Terashima-Marín, H. (2019). Selecting meta-heuristics for solving vehicle routing problems with time windows via meta-learning. Expert Systems with Applications, 118, 470–481. doi:10.1016/j.eswa.2018.10.036
95. Degroote, H., Luis González-Velarde, J., & Causmaecker, P. D. (2018). Applying algorithm selection-a case study for the generalised assignment problem. Electronic Notes in Discrete Mathematics, 69, 205–212.
96. Pavelski, L., Delgado, M., & Kessaci, M.-E. (2018, July). Meta-learning for optimization: A case study on the flowshop problem using decision trees. 2018 IEEE Congress on Evolutionary Computation (CEC). Presented at the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro. doi:10.1109/cec.2018.8477664
97. Pavelski, L. M., Kessaci, M.-E., & Delgado, M. R. (2018, October). Recommending meta-heuristics and configurations for the flowshop problem via meta-learning: Analysis and design. 2018 7th Brazilian Conference on Intelligent Systems (BRACIS). Presented at

the 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), Sao Paulo. doi:10.1109/bracis.2018.00036
98. León, A., Dávila De, E., Lalla-Ruiz, B., & Melián-Batista, J. M. (2017). Meta-learning-based system for solving logistic optimization problems. In International conference on computer aided systems theory (pp. 339–346). Cham: Springer.
99. Kanda, J., Carvalho, A. de, Hruschka, E., Soares, C., & Brazdil, P. (2016). Meta-learning to select the best meta-heuristic for the Traveling Salesman Problem: A comparison of meta-features. Neurocomputing, 205, 393–406. doi:10.1016/j.neucom.2016.04.027
100. de León, A. D., Lalla-Ruiz, E., Melián-Batista, B., & Marcos Moreno-Vega, J. (2017). A Machine Learning-based system for berth scheduling at bulk terminals. Expert Systems with Applications, 87, 170–182. doi:10.1016/j.eswa.2017.06.010
101. Song, H., Triguero, I., & Özcan, E. (2019). A review on the self and dual interactions between machine learning and optimisation. Progress in Artificial Intelligence, 8(2), 143–165. doi:10.1007/s13748-019-00185-z
102. Beham, A., Affenzeller, M., & Wagner, S. (2017, July 15). Instance-based algorithm selection on quadratic assignment problem landscapes. Proceedings of the Genetic and Evolutionary Computation Conference Companion. Presented at the GECCO ’17: Genetic and Evolutionary Computation Conference, Berlin Germany. doi:10.1145/3067695.3082513
103. Wawrzyniak, J., Drozdowski, M., & Sanlaville, É. (2020). Selecting algorithms for large berth allocation problems. European Journal of Operational Research, 283(3), 844–862. doi:10.1016/j.ejor.2019.11.055
104. Dantas, A., & Pozo, A. (2020). On the use of fitness landscape features in meta-learning based algorithm selection for the quadratic assignment problem. Theoretical Computer Science, 805, 62–75. doi:10.1016/j.tcs.2019.10.033
105. Tenne, Y. (2012). A computational intelligence algorithm for expensive engineering optimization problems. Engineering Applications of Artificial Intelligence, 25(5), 1009–1021. doi:10.1016/j.engappai.2012.03.009
106. Cai, X., Gao, L., Li, X., & Qiu, H. (2019). Surrogate-guided differential evolution algorithm for high dimensional expensive problems. Swarm and Evolutionary Computation, 48, 288–311. doi:10.1016/j.swevo.2019.04.009
107. Fan, C., Hou, B., Zheng, J., Xiao, L., & Yi, L. (2020). A surrogate-assisted particle swarm optimization using ensemble learning for expensive problems with small sample datasets. Applied Soft Computing, 91(106242), 106242. doi:10.1016/j.asoc.2020.106242
108. Xiang, X., Tian, Y., Xiao, J., & Zhang, X. (2020). A clustering-based surrogate-assisted multiobjective evolutionary algorithm for shelter location problem under uncertainty of road networks. IEEE Transactions on Industrial Informatics, 16(12), 7544–7555. doi:10.1109/tii.2019.2962137
109. Smith, R. E., Dike, B. A., & Stegmann, S. A. (1995). Fitness inheritance in genetic algorithms. Proceedings of the 1995 ACM Symposium on Applied Computing - SAC ’95. Presented at the the 1995 ACM symposium, Nashville, Tennessee, United States. doi:10.1145/315891.316014
110. Dantas, A. L., & Pozo, A. T. R. (2018, July). A meta-learning algorithm selection approach for the quadratic assignment problem. 2018 IEEE Congress on Evolutionary

Computation (CEC). Presented at the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro. doi:10.1109/cec.2018.8477989
111. Pelikan, M., & Sastry, K. (2004). Fitness inheritance in the Bayesian optimization algorithm. Retrieved from http://arxiv.org/abs/cs/0402032
112. Zhang, J., Yim, Y.-S., & Yang, J. (1997). Intelligent selection of instances for prediction functions in lazy learning algorithms. In Lazy Learning (pp. 175–191). doi:10.1007/978-94-017-2053-3_7
113. Erick, A. D., Masegosa, D., & Oliva, E. (2022). A new Hyper-heuristic based on Adaptive Simulated Annealing and Reinforcement Learning for the Capacitated Electric Vehicle Routing Problem.
114. Zhu, J. (2022). Solving capacitated vehicle routing problem by an improved genetic algorithm with fuzzy C-means clustering. Scientific Programming, 2022, 1–8. doi:10.1155/2022/8514660
115. Lucas, F., Billot, R., Sevaux, M., & Sörensen, K. (2020). Reducing space search in combinatorial optimization using machine learning tools. In Lecture Notes in Computer Science. Lecture Notes in Computer Science (pp. 143–150). doi:10.1007/978-3-030-53552-0_15
116. Hao, J.-H., Liu, M., Lin, J.-H., & Wu, C. (2016). A hybrid differential evolution approach based on surrogate modelling for scheduling bottleneck stages. Computers & Operations Research, 66, 215–224. doi:10.1016/j.cor.2015.08.005
117. Nguyen, S., Zhang, M., Johnston, M., & Tan, K. C. (2014). Selection schemes in surrogate-assisted genetic programming for job shop scheduling. In Lecture Notes in Computer Science. Lecture Notes in Computer Science (pp. 656–667). doi:10.1007/978-3-319-13563-2_55
118. Knowles, J. (2006). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation : A Publication of the IEEE Neural Networks Council, 10(1), 50–66. doi:10.1109/tevc.2005.851274
119. Porumbel, D. C., Hao, J.-K., & Kuntz, P. (2010). A search space “cartography” for guiding graph coloring heuristics. Computers & Operations Research, 37(4), 769–778. doi:10.1016/j.cor.2009.06.024
120. Oliveira, J., Almeida, M., Santos, R., De Gusmão, R., & Britto, A. (2020). New surrogate approaches applied to metaheuristic algorithms. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing. 400.
121. Sun, T., Yuan, J.-M., School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China, & Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan 411105, China. (2023). Prediction of band gap of transition metal sulfide with Janus structure by deep learning atomic feature representation method. Wuli Xuebao, 72(2), 028901. doi:10.7498/aps.72.20221374
122. Hussein, A. A. (2017). A Comparison between Harris and FAST-Corner Detection of Noisy Images Using Adaptive Non-Local Means. Diyala Journal For Pure Science, 13(4-part 1).
123. Hussein, A. A., & Ahmed, M. K. (2018). Image De-Noising Based On Wavelet Transform and Block Matching. Tikrit Journal of Pure Science, 23(2), 129-136.

124. Chen, B., Qu, R., Bai, R., & Laesanklang, W. (2020). A variable neighbourhood search algorithm with reinforcement learning for a real-life periodic vehicle routing problem with time windows and open routes. RAIRO Operations Research, 54(5), 1467–1494. doi:10.1051/ro/2019080
125. Zhao, H., Zhang, C., Zheng, X., Zhang, C., & Zhang, B. (2022). A decomposition-based many-objective ant colony optimization algorithm with adaptive solution construction and selection approaches. Swarm and Evolutionary Computation, 68(100977), 100977. doi:10.1016/j.swevo.2021.100977
126. Mesa, J. P., Montoya, A., Toro, M., & Ramos, R. (2022). Machine-learning component for multi-start metaheuristics to solve the capacitated vehicle routing problem. SSRN Electronic Journal. doi:10.2139/ssrn.4120102
127. Cheng, C.-Y., Pourhejazy, P., Ying, K.-C., & Lin, C.-F. (2021). Unsupervised Learning-based Artificial Bee Colony for minimizing non-value-adding operations. Applied Soft Computing, 105(107280), 107280. doi:10.1016/j.asoc.2021.107280
128. Ali, I. M., Essam, D., & Kasmarik, K. (2020). A novel design of differential evolution for solving discrete traveling salesman problems. Swarm and Evolutionary Computation, 52(100607), 100607. doi:10.1016/j.swevo.2019.100607
129. Nasiri, M. M., Salesi, S., Rahbari, A., Salmanzadeh Meydani, N., & Abdollai, M. (2019). A data mining approach for population-based methods to solve the JSSP. Soft Computing, 23(21), 11107–11122. doi:10.1007/s00500-018-3663-2
130. Min, J. N., Jin, C., & Lu, L. J. (2019). Maximum-minimum distance clustering method for split-delivery vehicle-routing problem: Case studies and performance comparisons. Advances in Production Engineering & Management, 14(1), 125–135. doi:10.14743/apem2019.1.316
131. Ali, I., Essam, D., & Kasmarik, K. (2019). New designs of k-means clustering and crossover operator for solving traveling salesman problems using evolutionary algorithms. Proceedings of the 11th International Joint Conference on Computational Intelligence. Presented at the 11th International Conference on Evolutionary Computation Theory and Applications, Vienna, Austria. doi:10.5220/0007940001230130
132. Miki, S., Yamamoto, D., & Ebara, H. (2018, August). Applying deep learning and reinforcement learning to traveling salesman problem. 2018 International Conference on Computing, Electronics & Communications Engineering (ICCECE). Presented at the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, United Kingdom. doi:10.1109/iccecome.2018.8659266
133. Alipour, M. M., Razavi, S. N., Feizi Derakhshi, M. R., & Balafar, M. A. (2018). A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. Neural Computing & Applications, 30(9), 2935–2951. doi:10.1007/s00521-017-2880-4
134. Eduyn, W., & Rodríguez-Vásquez, G. (2018). A hybrid expert system, clustering and ant colony optimization approach for scheduling and routing problem in courier services. International Journal of Industrial Engineering Computations, 9(3), 369–396.
135. Hassanat, A., Prasath, V., Abbadi, M., Abu-Qdari, S., & Faris, H. (2018). An improved genetic algorithm with a new initialization mechanism based on regression techniques. Information (Basel), 9(7), 167. doi:10.3390/info9070167

136. Chang, Y.-C. (2017, October). Using k-means clustering to improve the efficiency of ant colony optimization for the traveling salesman problem. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2017 IEEE International Conference on Systems, Man and Cybernetics (SMC), Banff, AB. doi:10.1109/smc.2017.8122633
137. Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. Retrieved from http://arxiv.org/abs/1704.01665
138. Zhang, J. (2017, September). An efficient density-based clustering algorithm for the capacitated vehicle routing problem. 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA). Presented at the 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an. doi:10.1109/iccnea.2017.96
139. Gao, S., Wang, Y., Cheng, J., Inazumi, Y., & Tang, Z. (2016). Ant colony optimization with clustering for solving the dynamic location routing problem. Applied Mathematics and Computation, 285, 149–173. doi:10.1016/j.amc.2016.03.035
140. Li, C., Chu, X., Chen, Y., & Xing, L. (2016). A knowledge-based technique for initializing a genetic algorithm. Journal of Intelligent & Fuzzy Systems, 31(2), 1145–1152. doi:10.3233/jifs-169043
141. Deng, Y., Liu, Y., & Zhou, D. (2015). An improved genetic algorithm with initial population strategy for symmetric TSP. Mathematical Problems in Engineering, 2015, 1–6. doi:10.1155/2015/212794
142. Drugan, M. M. (2019). Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms. Swarm and Evolutionary Computation, 44, 228–246. doi:10.1016/j.swevo.2018.03.011
143. Samorani, M., Wang, Y., Wang, Y., Lv, Z., & Glover, F. (2019). Clustering-driven evolutionary algorithms: an application of path relinking to the quadratic unconstrained binary optimization problem. Journal of Heuristics, 25(4–5), 629–642. doi:10.1007/s10732-018-9403-z
144. Potvin, J.-Y., & Thangiah, S. R. (2020). Vehicle routing through simulation of natural processes. In Fusion of Neural Networks, Fuzzy Systems, and Genetic Algorithms (pp. 141–168). doi:10.1201/9780367811464-6
145. Mallipeddi, R., Das, S., & Suganthan, P. N. (2015). Ensemble of constraint handling techniques for single objective constrained optimization. In Infosys Science Foundation Series (pp. 231–248). doi:10.1007/978-81-322-2184-5_9
146. Drugan, M. M., & Talbi, E.-G. (2014). Adaptive Multi-operator MetaHeuristics for Quadratic Assignment Problems. In Advances in Intelligent Systems and Computing. Advances in Intelligent Systems and Computing (pp. 149–163). doi:10.1007/978-3-319-07494-8_11
147. Chen, Y., Cowling, P. I., Fiona, A. C., & Polack, P. J. (2016). A multi-arm bandit neighbourhood search for routing and scheduling problems.
148. Song, Y., Wei, L., Yang, Q., Wu, J., Xing, L., & Chen, Y. (2022). RL-EA: A Reinforcement Learning-Based Evolutionary Algorithm Framework for Electromagnetic Detection Satellite Scheduling Problem.
149. Karimi-Mamaghan, M., Mohammadi, M., Pasdeloup, B., & Meyer, P. (2023). Learning to select operators in meta-heuristics: An integration of Q-learning into the iterated

greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research, 304(3), 1296–1330. doi:10.1016/j.ejor.2022.03.054
150. Zhao, F., Zhang, L., Cao, J., & Tang, J. (2021). A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem. Computers & Industrial Engineering, 153(107082), 107082. doi:10.1016/j.cie.2020.107082
151. Lu, H., Zhang, X., & Yang, S. (2019). A learning-based iterative method for solving vehicle routing problems. In International conference on learning representations.
152. Peng, B., Zhang, Y., Gajpal, Y., & Chen, X. (2019). A memetic algorithm for the Green vehicle routing problem. Sustainability, 11(21), 6055. doi:10.3390/su11216055
153. Gretsista, A., & Burke, E. K. (2017). An iterated local search framework with adaptive operator selection for nurse rostering. In Lecture Notes in Computer Science. Lecture Notes in Computer Science (pp. 93–108). doi:10.1007/978-3-319-69404-7_7
154. Zhalechian, M., Tavakkoli-Moghaddam, R., Zahiri, B., & Mohammadi, M. (2016). Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. Transportation Research Part E: Logistics and Transportation Review, 89, 182–214. doi:10.1016/j.tre.2016.02.011
155. Li, K., & Tian, H. (2016). A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. Applied Soft Computing, 43, 469–479. doi:10.1016/j.asoc.2016.02.040
156. Jian, P. M., Pardalos, H., Sun, J., & Pei, Y. (2015). Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. Expert Systems with Applications, 42(7), 3551–3561.
157. Jong, K. (2007). Parameter setting in EAs: a 30 year perspective. In Parameter setting in evolutionary algorithms (pp. 1–18). Berlin, Heidelberg: Springer.
158. Eiben, A. E., & Smit, S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation, 1(1), 19–31. doi:10.1016/j.swevo.2011.02.001
159. Mohammadi, M., Jula, P., & Tavakkoli-Moghaddam, R. (2019). Reliable single-allocation hub location problem with disruptions. Transportation Research Part E: Logistics and Transportation Review, 123, 90–120. doi:10.1016/j.tre.2019.01.008
160. Huang, C., Li, Y., & Yao, X. (2020). A survey of automatic parameter tuning methods for metaheuristics. IEEE Transactions on Evolutionary Computation : A Publication of the IEEE Neural Networks Council, 24(2), 201–216. doi:10.1109/tevc.2019.2921598
161. Lee, H. M., Jung, D., Sadollah, A., & Kim, J. H. (2020). Performance comparison of metaheuristic algorithms using a modified Gaussian fitness landscape generator. Soft Computing, 24(10), 7383–7393. doi:10.1007/s00500-019-04363-y
162. Ahmadi, E., Goldengorin, B., Süer, G. A., & Mosadegh, H. (2018). A hybrid method of 2-TSP and novel learning-based GA for job sequencing and tool switching problem. Applied Soft Computing, 65, 214–229. doi:10.1016/j.asoc.2017.12.045
163. Shankar, G., & Mukherjee, V. (2016). Quasi oppositional harmony search algorithm based controller tuning for load frequency control of multi-source multi-area power system. International Journal of Electrical Power & Energy Systems, 75, 289–302. doi:10.1016/j.ijepes.2015.09.011

164. Gao, X., Yang, S., & Li, L. (n.d.). Optimization of flow shop scheduling based on genetic algorithm with reinforcement learning. In Journal of Physics: Conference Series (Vol. 2258). IOP Publishing.
165. Ahmed, K., Eda, Z., Li, B., & Veeravalli, S. (2022). Reinforcement learning-enabled genetic algorithm for school bus scheduling. Journal of Intelligent Transportation Systems, 26(3), 269–283.
166. Koksal, E., Hegde, A. R., Pandiarajan, H. P., & Veeravalli, B. (2021). Performance characterization of reinforcement learning-enabled evolutionary algorithms for integrated school bus routing and scheduling problem. International Journal of Cognitive Computing in Engineering, 2, 47–56. doi:10.1016/j.ijcce.2021.02.001
167. Chaves, A. A., & Lorena, L. H. N. (2021, June 28). An adaptive and near parameter-free BRKGA using Q-learning method. 2021 IEEE Congress on Evolutionary Computation (CEC). Presented at the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland. doi:10.1109/cec45853.2021.9504766
168. Quevedo, J., Abdelatti, M., Imani, F., & Sodhi, M. (2021, July 7). Using reinforcement learning for tuning genetic algorithms. Proceedings of the Genetic and Evolutionary Computation Conference Companion. Presented at the GECCO ’21: Genetic and Evolutionary Computation Conference, Lille France. doi:10.1145/3449726.3463203
169. Tan, C. G., Siang Choong, S., & Wong, L.-P. (2021, November). A machine-learning-based approach for parameter control in bee colony optimization for traveling salesman problem. 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI). Presented at the 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taichung, Taiwan. doi:10.1109/taai54685.2021.00019
170. Oztop, H., Tasgetiren, M. F., Kandiller, L., & Pan, Q.-K. (2020, July). A novel general variable neighborhood search through Q-learning for no-idle flowshop scheduling. 2020 IEEE Congress on Evolutionary Computation (CEC). Presented at the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, United Kingdom. doi:10.1109/cec48606.2020.9185556
171. Chen, R., Yang, B., Li, S., & Wang, S. (2020). A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. Computers & Industrial Engineering, 149(106778), 106778. doi:10.1016/j.cie.2020.106778
172. Zhang, R., Prokhorchuk, A., & Dauwels, J. (2020, July). Deep reinforcement learning for traveling salesman problem with time windows and rejections. 2020 International Joint Conference on Neural Networks (IJCNN). Presented at the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom. doi:10.1109/ijcnn48605.2020.9207026
173. Cao, Z., Lin, C., & Zhou, M. (2021). A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. IEEE Transactions on Automation Science and Engineering: A Publication of the IEEE Robotics and Automation Society, 18(1), 56–69. doi:10.1109/tase.2019.2945717
174. Cooray, P. L. N. U., & Rupasinghe, T. D. (2017). Machine learning-based parameter tuned genetic algorithm for energy minimizing vehicle Routing Problem. Journal of Industrial Engineering, 2017, 1–13. doi:10.1155/2017/3019523
175. Abedi, M., Seidgar, H., & Fazlollahtabar, H. (2017). Hybrid scheduling and maintenance problem using artificial neural network based meta-heuristics. Journal of Modelling in Management, 12(3), 525–550. doi:10.1108/jm2-02-2016-0011
176. Beham, A., Affenzeller, M., & Wagner, S. (2017, July 15). Instance-based algorithm selection on quadratic assignment problem landscapes. Proceedings of the Genetic and Evolutionary Computation Conference Companion. Presented at the GECCO ’17: Genetic and Evolutionary Computation Conference, Berlin Germany. doi:10.1145/3067695.3082513

177. León, A., Dávila De, E., Lalla-Ruiz, B., & Melián-Batista, J. M. (2017). Meta-learning-based system for solving logistic optimization problems. In International conference on computer aided systems theory (pp. 339–346). Cham: Springer.
178. Salih, A. A., & Alrawy, S. S. (2020). Shifted modified chebyshev direct method for solving quadratic optimal control problem. Samarra Journal of Pure and Applied Science, 2(1), 67-75.
179. Abdulrahman, A. A., & Ibrahem, M. K. (2020). Toward constructing a balanced intrusion detection dataset based on CICIDS2017. Samarra Journal of Pure and Applied Science, 2(3), 132-142.

IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS
مجلة سامراء للعلوم الصرفة والتطبيقية

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

التعلم الارشادي في مشاكل التوجيه والجدولة: مراجعة

احمد عبد المنعم حسين [1]*، عصام طه ياسين [2]، احمد نوري رشيد[2]

1 كلية العلوم الاسلامية، جامعة سامراء، العراق

2 كلية علوم الحاسوب وتكنلوجيا المعلومات، جامعة الانبار، العراق

معلومات البحث:

تأريخ الاستلام: 2023/01/09
تأريخ القبـــول: 2023/02/22

الكلمات المفتاحية:

الادلة الارشادية العليا، التعلم الالي ، مشاكل التوجيه ، مشاكل الجدولة ، التعلم الارشادي، مشاكل التحسين الاندماجي

معلومات المؤلف

الايميل:
ahmed.abd@uosamrra.edu.iq
الموبايل:

الخلاصة:

تعد مشاكل التحسين الاندماجية أهم فئة من مشاكل التحسين نظراً لتطبيقاتها العملية الكثيرة. تهتم هذه المشاكل بتحديد أفضل حل من مجموعة متقطعة من جميع الحلول المتاحة. تعتبر أنظمة النقل (التوجيه) والتوزيع (الجدولة) من أكثر أمثلة التحسين الاندماجي صعوبةً. نظرًا لأهمية مشاكل التوجيه والجدولة، فقد تم اقتراح العديد من الطرق لمعالجتها. يمكن تصنيف هذه الأساليب إلى طرق تقليدية (طرق دقيقة ومتعددة المسارات) وطرق التعلم الآلي. تم اقتراح طرق التعلم الآلي للتغلب على المشكلات التي تعاني منها الطرق التقليدية، وخاصة الوقت الحسابي العالي والاعتماد على معرفة الخبراء. في الآونة الأخيرة، تم الجمع بين أساليب التعلم الآلي والأدلة الارشادية العليا للتعامل مع مشاكل التحسين الاندماجية، ثم ظهر مصطلح التعلم الارشادي. يهدف هذا المزيج إلى توجيه الادلة الارشادية العليا نحو بحث فعال وكفء وقوي وتحسين أداء الادلة الارشادية العليا من حيث جودة الحل. يستعرض هذا العمل الادبيات والمنشورات التي تم فيها استخدام التعاون بين الادلة الارشادية العليا والتعلم الآلي لاقتراح مبادئ توجيهية للباحثين لطرح خوارزميات جديدة لديها قدرة جيدة على حل مشاكل التوجيه والجدولة