Applied Soft Computing Journal 164 (2024) 111947

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A survey of mat-heuristics for combinatorial optimisation problems: Variants, trends and opportunities

Chong Man Ngoo [a], Say Leng Goh [a,*], San Nah Sze [b], Nasser R. Sabar [c], Mohd Hanafi Ahmad Hijazi [d], Graham Kendall [e,f]

[a] Optimization and Visual Analytics Research Group, Faculty of Computing & Informatics, Universiti Malaysia Sabah-Labuan International Campus, Labuan, 87000, Labuan F.T., Malaysia

[b] Faculty of Computer Science & Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, 94300, Sarawak, Malaysia

[c] Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC 3086, Australia

[d] Faculty of Computing & Informatics, Universiti Malaysia Sabah, Kota Kinabalu, 88400, Sabah, Malaysia

[e] School of Engineering and Computing, MILA University, Putra Nilai, 71800, Negeri Sembilan, Malaysia

[f] School of Computer Science, University of Nottingham, Nottingham, NG7 2RD, United Kingdom

ARTICLE INFO

Keywords:
Combinatorial optimisation
Mat-heuristic
Matheuristic
Meta-heuristic
Mathematical optimisation

ABSTRACT

This survey paper presents an overview of recent application of mat-heuristics on combinatorial optimisation problems (COPs) from 2018 to 2024. In this review, we categorise the mat-heuristics into six categories based on three integration types (loose, tight and multi) and two approaches (direct and decomposition). Descriptive statistics reveal that tight integration mat-heuristics are widely favoured. It is also observed that direct approaches are more commonly employed compared to decomposition approaches, perhaps due to the complexity involved in the latter. Next, we briefly present the mechanism of each mat-heuristic and its performance in a comparison to other state-of-the-art solution methodologies. CPLEX emerges as the predominant solver. Mat-heuristics have demonstrated their versatility across COPs, consistently achieving or setting new best-known solutions (BKS). We analyse highly effective mat-heuristics and outline the implementation strategies employed by those that managed to set new BKS. In addition, we discuss the advantages and challenges of utilising mat-heuristics as a solution methodology, as well as future research opportunities in this domain.

Contents

* Correspondence to: Optimization Research Group, Faculty of Computing and Informatics, Universiti Malaysia Sabah Kampus Antarabangsa, Labuan, 87000, Malaysia

E-mail address: slgoh@ums.edu.my (S.L. Goh).

https://doi.org/10.1016/j.asoc.2024.111947
Received 20 February 2024; Received in revised form 3 June 2024; Accepted 3 July 2024
Available online 8 July 2024

1. Introduction

Combinatorial optimisation problems (COPs) are ubiquitous in many fields, including logistics to scheduling. They are challenging to solve due to their combinatorial nature, and traditional optimisation techniques often fail to produce satisfactory results. To address these challenges, researchers have developed a new class of algorithms called *matheuristic*, which combines mathematical programming (exact methods) and heuristic methods. The concept is not new and can be traced back to as early as 1964, in the work of Bellman who applied dynamic programming with heuristic techniques in addressing map colouring problems [1]. The terminology 'matheuristic' was established at the *First Workshop on Mathematical Contributions to Metaheuristics — Matheuristics 2006* which was held in Italy [2]. The work of Vansteenwegen et al. (2009) [3] is among the earliest to use the term 'matheuristic', where they combined linear programming with an iterated local search framework in addressing a capacitated arc routing problem. Matheuristic has recently emerged as a promising approach in tackling COPs such as vehicle routing [4], time-tabling [5], nurse rostering [6] and portfolio optimisation [7]. To draw a distinction between metaheuristic and matheuristic, the term 'mat-heuristic' will be used throughout this paper for clarity purpose.

Mat-heuristics were defined by Manniezzo et al. [8] as "*problem agnostic optimisation algorithms that make use of mathematical programming (MP) techniques in order to obtain heuristic solutions.*" Boschetti et al. [9] and Archetti and Speranza [10] defined mat-heuristics as "*heuristic algorithms created by the interoperation of mathematical programming and metaheuristics*". Another definition is provided by Doerner and Schmid [11] as "*An algorithm combining the strengths of metaheuristics and exact search components*". We define mat-heuristic as "*a hybrid algorithm that combines exact and (meta-) heuristic methodologies*".

Exact methods can guarantee an optimal solution but are not efficient in dealing with large and complex COPs [12]. Heuristics are advantageous for COPs because they can efficiently search large solution space without having to deal with explicit problem models thus are flexible in handling complex problem structures and objectives. However, heuristics cannot guarantee optimality, as they are designed to find good-quality solutions in reasonable computational time [13].

Puchinger and Raidl [14] conducted an early survey in 2005 on solutions approaches that combined exact methods and metaheuristics in addressing COPs. They divided the approaches into two combinations:

- Collaborative: Exact and heuristic algorithms exchange information but are not part of each other. They may be executed sequentially, intertwined or in parallel.
- Integrative: One method is embedded into another method. Either an exact method or a metaheuristic can be the master or the integrated slave.

Ball [15] proposed four categories of mat-heuristics namely:

- Decomposition: A problem is decomposed into smaller sub-problems. Exact methods are used to address some or all these sub-problems optimally or sub-optimally.
- Improvement heuristics: Exact methods are applied to improve the solution found by a heuristic approach.
- Approximation based approaches: Exact methods, particularly branch-and-bound, are used to derive an approximate solution.
- Relaxation based approaches: Approaches that begin by solving a relaxation of the original problem to produce a high-quality feasible solution.

The survey of Archetti and Speranza [10] covers a variety of mat-heuristic approaches, specifically in routing problems. The following categories of mat-heuristic were proposed:

- Decomposition approaches
- Improvement heuristics
- Branch-and-price or column generation-based approaches: An exact method is modified in a heuristic manner to improve the convergence rate. Consequently, the algorithm may lose its guaranteed optimality attributes.

The aim of this survey paper is to enable the Operational Research (OR) and Computational Intelligence (CI) communities to develop a perspective on mat-heuristics, find gaps and identify emerging trends in this important research area. The contributions of this survey paper are:

- We systematically categorise mat-heuristics into six categories based on the combinations of integration types (loose, tight, and multi) and approaches (direct and decomposition).
- We analyse solution methodologies employing mat-heuristics for various COPs in terms of mechanism and their relative performance to existing approaches.
- We analyse and compare the performance of different categories of mat-heuristic. In addition, we study and present the implementation strategies of highly effective mat-heuristics.
- We discuss the advantages and challenges of utilising mat-heuristics and offer valuable insights into future research directions.

This paper is structured as follows: Section 2 outlines the scope of this survey. In Section 3, mat-heuristics are categorised based on integration types and approaches. Section 4 presents the application of mat-heuristics on various COPs. The mathematical solvers utilised by the mat-heuristics are presented in Section 5. Section 6 presents the performance analysis of the mat-heuristics. The advantages and challenges associated with mat-heuristics are discussed in Section 7. Section 8 presents the potential research opportunities. Finally, the concluding remarks are given in Section 9.

Table 1
Collected papers by journal.

Journal	Number of papers
Computers and Operations Research	11
European Journal of Operational Research	10
Computers & Industrial Engineering	4
Expert Systems With Applications	3
Journal of Scheduling	3
Annals of Operation Research	2
Applied Soft Computing	2
EURO Journal on Transportation & Logistics	2
International Transactions in Operational Research	2
Transportation Research Part E	2
Transportation Research Part B	1
European Journal of Industrial Engineering	1
Mathematical Programming Computation	1
Transportation Science	1
Total	**45**

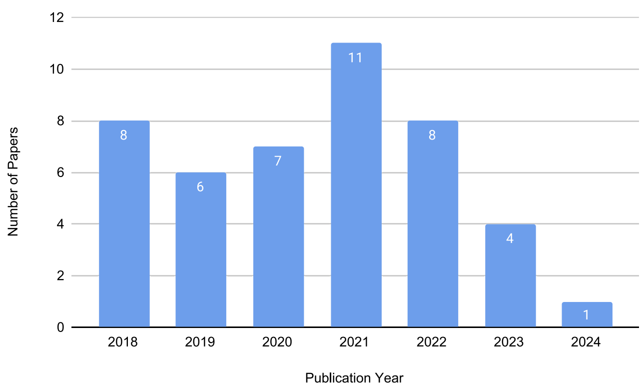

Fig. 1. Collected papers by publication year.

2. Survey scope and methodology

This survey aims to record, analyse and categorise the recent application of mat-heuristics in addressing COPs. First, we collected and read the latest survey papers on mat-heuristic to gain an overview of the topic. Next, we collected around 70 methodological papers published between 2018 and 2024 from a range of bibliographic databases such as Scopus and Web of Science. The keywords used in our search included "*mat-heuristic*", "*matheuristic*", "*combinatorial optimisation*" and "*operations research*".

The collected papers underwent a screening process of inclusion and exclusion to filter these papers via different stages. During the first stage, the papers retrieved underwent filtering based on their title and authors, with the removal of duplicate items. In the second stage, papers that were irrelevant to combinatorial optimisation were removed. In the third stage, the remaining papers were filtered based on their proposed methodology. Papers were excluded if the proposed methodology did not involve a mat-heuristic approach. Finally, papers published in less established journals were excluded to avoid predatory journals [16]. We refined our selection to 45 papers.

Next, we extracted and tabulated pertinent information from the papers such as publication year, problem addressed, dataset, methodology, results, findings, limitations and future work. Finally, we analysed the information and visualised it through figures and tables. We developed perspectives in the field, categorise the methodologies, assessed the performance of the methodologies, identified emerging trends and discovered research challenges and opportunities.

The count of papers by journal, publication year and COP are shown in

Table 1, Figs. 1 and 2 respectively.

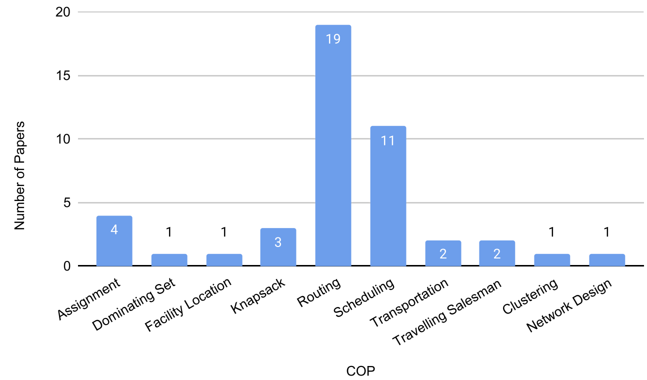

Fig. 2. Collected papers by COP.

3. Classification of mat-heuristics

Table 2 exhibits the application of mat-heuristics on COPs (sorted by year). We systematically categorise these mat-heuristics based on integration types and approaches. Fig. 3 illustrates the categorisation of the mat-heuristics.

3.1. Types of integration

After analysing the collected papers, we notice that mat-heuristics can be categorised based on integration types; loose, tight and multi. Each type offers different ways to combine mathematical optimisation methods (exact methods) and (meta-) heuristics, providing unique advantages and opportunities for tackling complex optimisation problems.

3.1.1. Loose integration

In a loose integration mat-heuristic, an exact method and a (meta-) heuristic are executed sequentially right after one another. Both methods are distinct from one another and exist as separate solution frameworks. They are not contained or nested within each other. This type of integration (similar to improvement heuristics [10,15]) is widely applied due to its relative ease of use. Implementation wise, an initial solution is generated using an exact method (typically by solving a mixed integer linear programming (MILP) model). The solution is further optimised utilising a heuristic method. Alternatively, MILP models can be employed (as local optimisation) on solutions generated by heuristic methods. Fig. 4 depicts the loose integration mat-heuristic.

3.1.2. Tight integration

A tight integration mat-heuristic involves the incorporation of an exact method and a (meta-) heuristic in a closely intertwined manner. It is similar to the integrative combination proposed by Puchinger and Raidl [14] and branch-and-price/column generation-based mat-heuristic proposed by Archetti and Speranza [10]. In this type of integration, an exact method is embedded within a heuristic (to serve as a local search component) or a heuristic method can be incorporated within a mathematical solution process, to enhance solution quality. Implementing this integration is more challenging compared to implementing a loose integration mat-heuristic, as it requires merging two distinct methods together within a single solution framework and fine-tuning the algorithm. Fig. 5 provides a visual representation of the tight integration mat-heuristic.

3.1.3. Multi integration

Multi integration encompasses the utilisation of multiple methodologies in addressing a COP where at least one base mat-heuristic must be present. For example:

- *Base Mat-heuristic + Heuristic Method + ...*

Table 2
The application of mat-heuristics on COPs (sorted by year).

Year	Mat-heuristic	COP	Ref.
2018	FiNeMath	Assignment	[17]
	Fix-and-Optimise based mat-heuristic (FnOMH)	Routing	[18]
	Adaptive Large Neighborhood Search with MIP (ALNS-MIP)	Routing	[19]
	Iterative Local Search with MILP (ILS-MILP)	Routing	[20]
	POPMUSIC mat-heuristic with Iterative Greedy Algorithm (POPIGA)	Scheduling	[21]
	Two-phase mat-heuristic (TPM)	Scheduling	[22]
	Perturbation mat-heuristic (PM)	Scheduling	[23]
	Evolutionary Algorithm Framework mat-heuristic (EAMH)	Transportation	[24]
2019	Tabu Search mat-heuristic (MTS)	Knapsack	[25]
	Mat-heuristic in Large Neighborhood Search (MH-LNS)	Routing	[26]
	ILS with Randomised Variable Neighborhood Descent (ILS-RVND)	Routing	[27]
	Iterative Local Search with Column Generation (ILS-CG)	Routing	[28]
	Rotation-based Branch-and-Price (R-BnP)	Scheduling	[29]
	Comprehensive mat-heuristic (CMH)	Travelling salesman	[30]
2020	Variable Neighborhood Descent with Integer Programming (VND-IP)	Knapsack	[31]
	Feasibility Pump with Large Neighborhood Search (FP-LNS)	Routing	[32]
	Column Generation with Diving Heuristic (CG-DH)	Scheduling	[33]
	Two mat-heuristics (2-MH)	Scheduling	[34]
	Fix-and-Optimise and Simulated Annealing (FnOSA)	Scheduling	[35]
	Fix-and-Optimise (FnO)	Scheduling	[36]
	Genetic Algorithm based mat-heuristic (GAMH)	Transportation	[37]
2021	Multi-start Iterative Local Search (MS-ILS)	Dominating set	[38]
	CS-ALNS-LB	Facility location	[39]
	Simulated Annealing and Local Search-based mat-heuristic (SALSBM)	Routing	[40]
	LNS with Set Partitioning Problem (LNS-SPP)	Routing	[41]
	GRASP-based mat-heuristic (GBM)	Routing	[42]
	LNS-based mat-heuristic v2 (MathHeu2)	Routing	[43]
	POPMUSIC-based mat-heuristic (POPMH)	Routing	[44]
	Iterative mat-heuristic (IM)	Routing	[45]
	Iterative Local Search-based mat-heuristic (ILS-MH)	Routing	[46]
	MILP-GA-ILS	Travelling salesman	[47]
	2-Phase Mat-heuristic (2PM)	Clustering	[48]
2022	Two-phase Iterative Kernel Search (TIKS)	Knapsack	[49]
	Hybrid Infeasible Space mat-heuristic (HISM)	Routing	[50]
	Branch-and-Cut with Iterative Local Search (BnC-ILS)	Routing	[51]
	Iterated Greedy mat-heuristic (IGM)	Scheduling	[52]
	Robust Two-Phase mat-heuristic (RTPM)	Routing	[53]
	TILS, OILS, and IILS	Routing	[54]
	Adaptive Large Neighborhood Search with Kernel Search (ALNS-KS)	Routing	[55]
	Two-stage mat-heuristic (TSM)	Scheduling	[56]
2023	Four-steps Decomposition Strategy mat-heuristic (4SD-MHA)	Assignment	[57]
	Three-steps mat-heuristic (3S-MHA)	Scheduling	[58]
	Restricted Formulation-based Heuristic (RFH)	Assignment	[59]
	Three-phase mat-heuristic (TPMH)	Assignment	[60]
2024	Relax-and-Restrict Mat-heuristic (RARM)	Network design	[61]

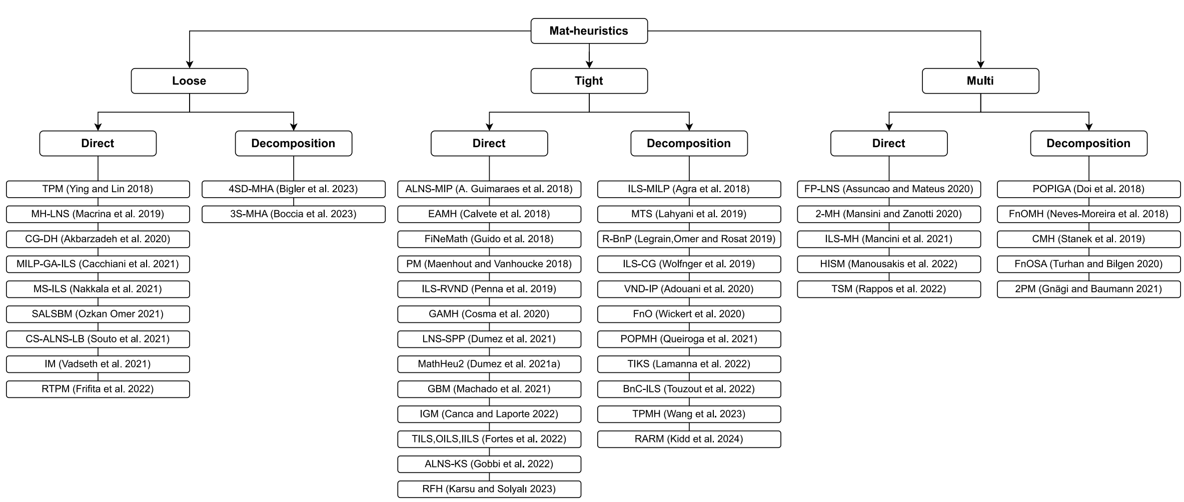

Fig. 3. Categorisation of mat-heuristics based on integration types and approaches.

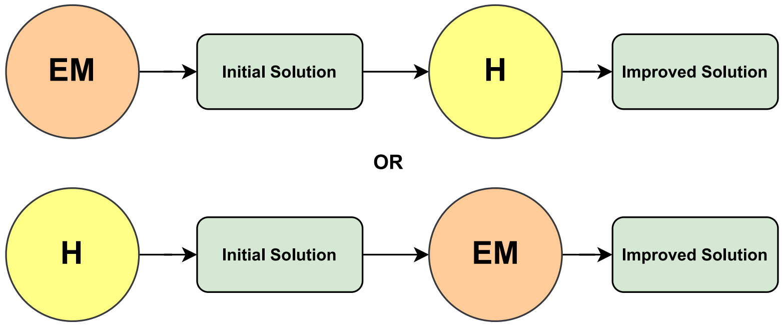

Fig. 4. Loose Integration. EM = Exact Method, H = Heuristic.

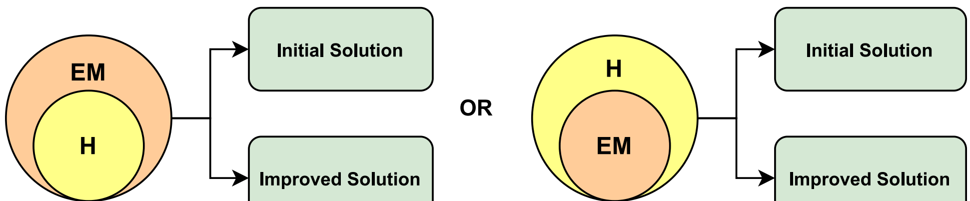

Fig. 5. Tight Integration. EM = Exact Method, H = Heuristic.

- *Base Mat-heuristic + Exact Method + ...*
- *Base Mat-heuristic + Mat-heuristic + ...*

Multi integration is typically applied to complicated multi-objective problems where each methodology is employed to address each objective or sub-problem separately. The integration of various methodologies requires careful coordination and synchronisation. It is not frequently utilised due to its complexity and computational overheads. Fig. 6 presents an example of multi integration mat-heuristic.

3.2. Approaches

From observation, researchers utilise a mat-heuristic to address a COP either directly or by using a decomposition approach.

3.2.1. Direct approach

This is a common and straightforward approach to implement a mat-heuristic where a COP is addressed directly without modifying the underlying problem structure.

3.2.2. Decomposition approach

In cases where the COP is highly complex, researchers may choose to decompose the problem into smaller and more manageable sub-problems to facilitate solution. Certain methodologies (such as branch-and-price and column generation) handle problem decomposition naturally where manual decomposition is not required. These methodologies are considered as decomposition approaches [10]. The visualisation of the decomposition approach is depicted in Fig. 7.

3.3. Analysis of the categorisation

Table 3 illustrates the distribution of mat-heuristics based on integration types and approaches. Among the analysed mat-heuristics, there are 11 loose integration (9 direct and 2 decomposition approaches), 24 tight integration (13 direct and 11 decomposition approaches), and 10 multi integration (5 direct and 5 decomposition approaches). It is observed that tight integration mat-heuristics are the most popular. In addition, direct approaches outnumbered decomposition approaches for each integration type.

Table 3
Distribution of mat-heuristics based on integration types and approaches.

Integration	Approach		Total
	Direct	Decomposition	
Loose	9	2	11
Tight	13	11	24
Multi	5	5	10
Total	27	18	45

Table 4
Application of mat-heuristics (category) per year. LD = loose-direct, LDC = loose-decomposition. TD = tight-direct, TDC = tight-decomposition, MD = multi-direct, MDC = multi-decomposition.

Year	Category of mat-heuristic						Total
	LD	LDC	TD	TDC	MD	MDC	
2018	1	–	4	1	–	2	8
2019	1	–	1	3	–	1	6
2020	1	–	1	2	2	1	7
2021	5	–	3	1	1	1	10
2022	1	–	3	2	2	–	8
2023	–	2	1	1	–	–	4
2024	–	–	–	1	–	–	4
Total	9	2	13	11	5	5	45

Direct approaches are favoured among researchers as they involve fewer complexities in addressing a problem. On the other hand, decomposition approaches required additional steps in decomposing a problem into sub-problems. Table 4 shows the application of mat-heuristics (category) per year.

Table 5 summarises the general properties, strengths and weaknesses of mat-heuristics based on categories.

4. Solution methodologies

In this section, we analyse the mechanisms and performances of the existing mat-heuristic applied to COPs.

4.1. Loose integration mat-heuristics

Table 6 shows the a list of loose integration mat-heuristics.

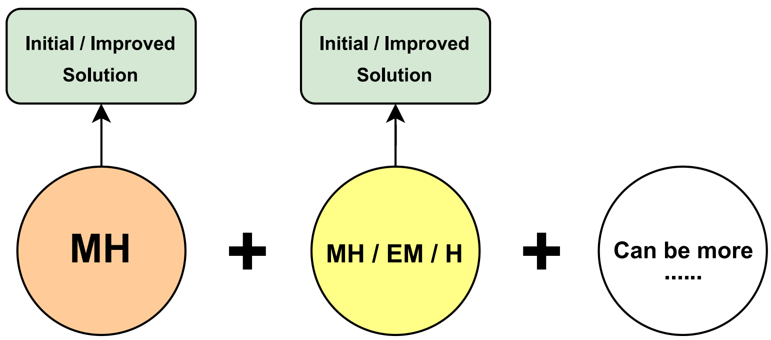

Fig. 6. Multi Integration. MH = Mat-heuristic, EM = Exact Method, H = Heuristic Method.

Table 5
Properties, strengths and weaknesses of mat-heuristic (category).

Mat-heuristic	Properties	Strengths	Weaknesses
Loose-Direct (LD)	Distinct methods are executed sequentially	Easy to implement, modify and tune	Limited synergy and interaction between methods
	Addresses a problem directly (without modifying problem structure)	Suitable for small-scale problems	May struggle with large-scale and complex problems
Loose-Decomposition (LDC)	Distinct methods are executed sequentially	Easy to implement, modify and tune	Limited synergy and interaction between methods
	Decomposes a problem into sub-problems	Produces manageable sub-problems	Requires careful coordination of sub-problems
		Suitable for large-scale and complex problems	Computationally expensive Solution combination issues
Tight-Direct (TD)	Methods are closely intertwined/embedded	High synergy and interaction between methods	Hard to implement, modify and tune
	Addresses a problem directly (without modifying problem structure)	May struggle with large-scale and complex problems	
Tight-Decomposition (TDC)	Methods are closely intertwined/embedded	High synergy and interaction between methods	Hard to implement, modify and tune
	Decomposes a problem into sub-problems	Produces manageable sub-problems	Requires careful coordination of sub-problems
		Suitable for large-scale and complex problems	Computationally expensive Solution combination issues
Multi-Direct (MD)	Multiple methodologies (at least one base mat-heuristic)	Suitable for multi-objective problems	High complexity
	Addresses a problem directly (without modifying problem structure)	Suitable for small-scale problems	Requires careful coordination of methodologies
			May struggle with large-scale and complex problems
Multi-Decomposition (MDC)	Multiple methodologies (at least one base mat-heuristic) Decomposes a problem into sub-problems	Suitable for multi-objective problems	High complexity
		Produces manageable sub-problems	Requires careful coordination of methodologies and sub-problems
		Suitable for large-scale and complex problems	Computationally expensive Solution combination issues

Table 6
List of loose integration mat-heuristics.

Mat-heuristic	Approach	COP	Heuristic method	Exact method	Ref.
TPM	Direct	Scheduling	NEH, Lin–Kernighan–Helsgaun (LKH)	Binary Integer Programming (BIP)	[22]
MH-LNS		Routing	Large Neighborhood Search (LNS)	Mixed-Integer Linear Programming (MILP)	[26]
CG-DH		Scheduling	Diving Heuristic	Column-generation, Linear Programming (LP)	[33]
MILP-GA-ILS		Travelling salesman	Genetic Algorithm (GA), ILS	MILP	[47]
MS-ILS		Dominating set	Iterative Local Search (ILS), H-MECU	Integer Linear Programming (ILP)	[38]
SALSBM		Routing	Simulated Annealing (SA), Local Search (LS)	ILP	[40]
CS-ALNS-LB		Facility location	Clustering Search (CS), ALNS, LB	MILP	[39]
IM		Routing	Split Algorithm, GA	Path-Flow-Model (PFM)	[45]
RTPM		Routing	Variable Neighborhood Search (VNS)	MILP	[53]
4SD-MHA	Decomposition	Assignment	Clustering Algorithm, Iterative Algorithm	LP	[57]
3S-MHA		Scheduling	ILS	MILP	[58]

4.1.1. Loose-Direct (LD)

Ying and Lin [22] proposed a two-phase mat-heuristic (TPM) to address a no-wait flow-shop scheduling problem with setup times. The TPM algorithm entailed the use of the Nawaz–Enscore–Ham (NEH) heuristic and the Lin–Kernighan–Helsgaun (LKH) algorithm (both are local search algorithms) in the first phase to rapidly generate a good

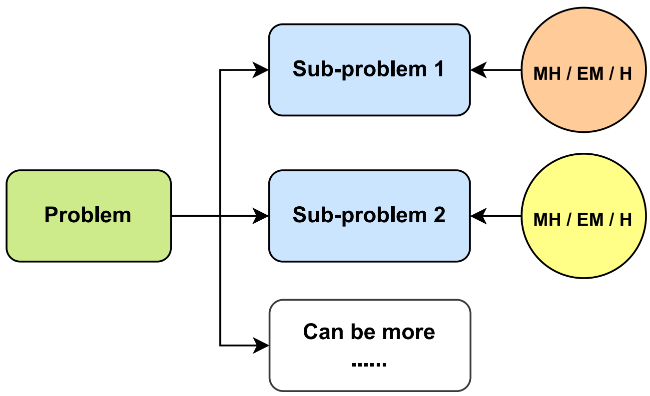

Fig. 7. Decomposition Approach. MH = Mat-heuristic, EM = Exact Method, H = Heuristic Method.

quality initial solution. In the second phase, the near-optimal solution obtained from the first phase acted as an upper bound for candidate solutions. A relaxed binary integer programming (BIP) model was repeatedly solved using the GUROBI solver until an optimal solution was found. The proposed mat-heuristic delivered optimal solutions for all test instances including extremely large-scale instances within a reasonable computational time.

Macrina et al. [26] introduced a mat-heuristic in large neighborhood search (MH-LNS) to tackle the energy-efficient green-vehicle routing problem. An initial feasible solution was obtained by solving the MILP using the CPLEX solver. The initial solution was improved iteratively using a large neighborhood search algorithm, utilising four destroy operators and four insertion operators. The proposed mat-heuristic demonstrated superior computational efficiency compared to the general-purpose CPLEX solver.

Akbarzadeh et al. [33] proposed a column generation with a diving heuristic (CG-DH) to address a real-world nurse rostering problem. An optimal fractional solution was derived using linear programming (LP) by employing column generation. The fractional solution was transformed into a feasible integer solution. If the integer solution could not be generated, the fractional solution was then passed to a greedy diving heuristic that operated in a depth-first search manner without backtracking. All models were solved using a CPLEX solver. The methodology generated near-optimal solutions.

Cacchiani et al. [47] introduced a mat-heuristic to address two travelling salesman problems: the Pollution Travelling Salesman Problem (PTSP) and the Energy Minimisation Travelling Salesman Problem (EMTSP). An initial feasible solution was generated by solving a linear programming relaxation of a MILP model using the CPLEX solver. A multi-operator genetic algorithm was then utilised to enhance the initial solution, followed by an iterated local search process to further refine the solution. Best-known solutions were found for all PTSP and most EMTSP instances. The limitation of this study was time window and capacity constraints were not considered in the PTSP.

Nakkala et al. [38] introduced a multi-start iterative local search (MS-ILS) to tackle the minimum capacitated dominating set (CAPMDS) problem. The ILS algorithm were restarted multiple times and the overall best solution was selected. If the MS-ILS failed to find an optimal solution, an integer linear programming model was then solved using the CPLEX solver to obtain the final solution. The proposed mat-heuristic generated better performance compared to state-of-the-art approaches in terms of solution quality and execution time.

Ozkan Omer [40] proposed a simulated annealing local search-based mat-heuristic (SALSBM) to address a real-world distance-constrained multi-based multi-UAV routing problem. The SALSBM integrated simulated annealing (SA), local search (LS), and an integer linear programming (ILP) model. An SA was utilised to generate constant parameter values for the ILP model. The ILP model was then solved using the CPLEX solver. If the ILP model could not yield an optimal solution, the LS algorithm was used to enhance the current best solution. The same neighborhood structure (swap move) was utilised in both SA and LS. The proposed mat-heuristic achieved higher-quality solutions compared to a basic genetic algorithm.

Souto et al. [39] integrated clustering search (CS), adaptive large neighborhood search (ALNS), and local branching (LB) in addressing the two-stage capacitated facility location (TSCFL) problem. The ALNS acted as the local search method within the CS, efficiently improving the solution quality during each iteration. The best solution obtained from CS-ALNS was further enhanced using LB (CPLEX solver). The proposed methodology outperformed the current state-of-the-art methods in terms of solution quality on 40 out of 50 instances.

Vadseth et al. [45] introduced an iterative mat-heuristic (IM) to tackle an inventory routing problem. Initially, a giant tour was generated, which was then split into a set of routes through a split algorithm. A modified version of a path-flow model was solved utilising the GUROBI solver. The obtained solution was then iteratively enhanced using various operators and a genetic algorithm. The proposed mat-heuristic achieved best-known solutions for 179 instances. The limitation of the proposed mat-heuristic was the frequent use of random operators which led to unnecessary removal of routes.

Frifita et al. [53] proposed a robust two-phase mat-heuristic (RTPM) to address a disassembly assembly routing problem. In the first phase, the problem was formulated as a MILP model (solved using the CPLEX solver). Values for the routing and purchasing decision variables were fixed. In the second phase, a variable neighborhood search (VNS) algorithm was employed. Four neighborhood structures (based on remove and swap operators) were tested. The methodology generated high-quality solutions for small and medium-sized instances (with fast computational time compared to a general MILP) and acceptable results for larger instances.

4.1.2. Loose-Decomposition (LDC)

Bigler et al. [57] proposed a four-step decomposition strategy mat-heuristic (4SD-MHA) to address a real-world customer assignment problem in direct marketing. First, customers who shared the same activities were grouped. Second, each group was further divided using a clustering algorithm. Third, a linear programming (LP) model was solved using the GUROBI solver to determine customer subgroup-to-activity assignments. Based on LP solution, an iterative algorithm was used to assign individual customers to activities. The mat-heuristic was successfully implemented, leading to a substantial increase in sales (up to 90%).

Boccia et al. [58] proposed a three-step mat-heuristic (3S-MHA) to address an automated guided vehicles (AGVs) scheduling problem with battery constraints (ASP-BC). In the first and second steps, the problem was decomposed into two sub-problems and sequentially solved using the GUROBI solver. In the last step, the obtained solution was refined through a local search phase utilising add, remove, and swap operations. The results showed the potency and scalability of the 3S-MHA, adeptly handling instances with up to 200 jobs and 10 AGVs in mere minutes, while maintaining an average gap of less than 1%.

4.2. Tight integration mat-heuristics

Table 7 shows a list of tight integration mat-heuristics.

4.2.1. Tight-Direct (TD)

A. Guimarães et al. [19] proposed an adaptive large neighborhood search with MIP (ALNS-MIP) to tackle the two-echelon multi-depot inventory-routing problem (2E-MDIRP). Vehicle routes were handled by ALNS while input pickups, product deliveries and routing were improved by solving a MIP formulation using the GUROBI solver. The search procedure was divided into segments. In each iteration, a subset

Table 7
List of tight integration mat-heuristics.

Mat-heuristic	Approach	COP	Heuristic method	Exact method	Ref.
ALNS-MIP	Direct	Routing	Adaptive Large Neighborhood Search (ALNS)	Mixed-Integer Programming (MIP)	[19]
EAMH		Transportation	Evolutionary Algorithm (EA)	Linear Programming (LP)	[24]
FiNeMath		Assignment	Relax-and-Fix (RnF), LNS	ILP	[17]
PM		Scheduling	VNS, Local Branching (LB)	Integer Programming (IP)	[23]
ILS-RVND		Routing	ILS, Variable Neighborhood Descent (VND)	Set Partitioning (SP) Formulation	[27]
GAMH		Transportation	GA	Network Simplex Algorithm (NSA)	[37]
LNS-SPP		Routing	LNS	SP	[41]
MathHeu2		Routing	LNS, Balas–Simonetti Neighborhood	SP	[43]
GBM		Routing	GRASP, Constructive Heuristics, VNS	Set-Covering (SC), SP	[42]
IGM		Scheduling	Iterated Greedy Algorithm (IGA), SA	LIP	[52]
TILS,OILS,IILS		Routing	ILS, Neighborhood Routing Search (NRS)	Vehicle-Indexed Formulation	[54]
ALNS-KS		Routing	ALNS, Kernel Search (KS)	MILP	[55]
RFH		Assignment	Iterative Framework	MILP	[59]
ILS-MILP	Decomposition	Routing	ILS, LB	MILP	[20]
MTS		Knapsack	Tabu Search (TS)	MILP	[25]
R-BnP		Scheduling	ALNS	Bracnh-and-Price (BnP)	[29]
ILS-CG		Routing	ILS	Column-generation	[28]
VND-IP		Knapsack	VND	IP	[31]
FnO		Scheduling	VNS	MIP	[36]
POPMH		Routing	POPMUSIC	MIP, LP	[44]
TIKS		Knapsack	Iterative Kernel Search (IKS)	LP	[49]
BnC-ILS		Routing	ILS	Branch-and-Cut	[51]
TPMH		Assignment	Variable Fixing Heuristic	IP	[60]
RARM		Network design	Relax-and-Restrict	MILP	[61]

of customers was either removed or inserted into their current route using a set of fifteen destroy and repair operators. The proposed mat-heuristic outperformed a branch-and-cut algorithm in terms of solution quality and computational time.

Calvete et al. [24] proposed a novel evolutionary algorithm framework mat-heuristic (EAMH) to address the two-stage fixed-charge transportation problem (TS-FCTP). An initial population of chromosomes was randomly generated. The chromosomes were represented as binary vectors. The chromosomes were iteratively improved through crossover, mutation, and survivor selection operators. An assigning and repairing procedure (ARP) was devised to check the feasibility of the chromosomes and repair the chromosomes (if required). A transshipment problem was solved using the LEMON solver in the ARP. The proposed mat-heuristic achieved optimal results while requiring less computational time.

Guido et al. [17] proposed a mat-heuristic named FiNeMath to address an offline patient-to-bed assignment problem. FiNeMath integrated a relax-and-fix heuristic (RnF), a large neighborhood search (LNS) heuristic, and an integer linear programming (ILP) model. An initial feasible solution was generated by the RnF. The initial solution was then iteratively improved by the LNS heuristic and the CPLEX solver. Two key components of the LNS were destroy and repair operations. The FiNeMath improved all the best-known bounds of the state-of-the-art.

Maenhout and Vanhoucke [23] proposed a perturbation mat-heuristic (PM) to address personnel shift and task re-scheduling problems. The two key components in the proposed mat-heuristic were a perturbation mechanism (diversification) and a local search mechanism (intensification). The perturbation mechanism was based on local branching principles, utilising the GUROBI solver to generate a new solution guided by specific equality and diversity thresholds. The new solution was then further enhanced using a variable neighborhood search (VNS). The VNS implemented three different neighborhood structures: a worker-based local search, a day-based local search, and a novel bucket list local search. The experimental results demonstrated that the bucket list local search yielded the most significant improvement in solution quality when compared to other local search methods.

Penna et al. [27] proposed a mat-heuristic to address a diverse range of vehicle routing problems involving a heterogeneous fleet. The mat-heuristic integrated a multi-start iterated local search with randomised variable neighborhood descent (ILS-RVND) and a set partitioning (SP) formulation. An initial solution was generated using a simple insertion heuristic. The initial solution was further improved by the ILS-RVND where a pool of routes was constructed. The pool was then used to build a restricted SP model (solved by a MIP solver). The proposed mat-heuristic consistently produced high-quality solutions by equalling or improving 71.70% of the best-known solutions.

Cosma et al. [37] proposed a genetic algorithm-based mat-heuristic (GAMH) to address a two-stage transportation problem. An initial population of chromosomes was randomly generated. An algorithm called chromosome optimisation (CO) was developed to improve the estimates (fitness values) of the chromosomes. CO involved solving a classical minimum cost flow problem using the Network Simplex Algorithm. The chromosomes then underwent a local search phase utilising genetic operators such as selection, crossover, and mutation. The proposed mat-heuristic was capable of generating high-quality solutions within a reasonable computational time.

Dumez et al. [41] introduced a large neighborhood search (LNS) with a set partitioning problem (LNS-SPP) formulation to tackle a vehicle routing problem with delivery options. In each iteration, a pool of routes was generated using various ruin and recreate operators. These routes were recombined by solving an SPP model (using CPLEX solver) to identify an optimal combination. The proposed mat-heuristic generated solutions comparable to the ones produced by existing algorithms, on specific instances.

Dumez et al. [43] introduced a new exact method into their LNS framework [41] in addressing a vehicle routing problem with time windows. In addition, a new neighborhood structure (Balas–Simonetti) was proposed. As the newly added component was time-consuming, an adaptive layer was integrated to carefully control the frequency and timing of component invocation. The enhanced mat-heuristic yielded 81 new best-known solutions.

Machado et al. [42] proposed a greedy randomised adaptive search procedure (GRASP) based mat-heuristic (GBM) to address a capacitated vehicle routing problem. A partial tour was created by solving a set-covering problem (SCP) (employing local optimums attained from a Variable Neighborhood Search (VNS)). It was filled by a constructive heuristic if required. The solution underwent a local search phase (VNS). Lastly, the solution was further improved by solving a set-partitioning problem (SPP). The SPP and SCP were solved using the CPLEX solver. The solutions produced were comparable to the ones reported in recent studies. The proposed mat-heuristic faced difficulty in generating good solutions in a timely manner for larger datasets.

Canca and Laporte [52] proposed an iterated greedy mat-heuristic (IGM) to address a real-size stochastic railway rapid transit network construction scheduling problem. The IGM generated and enhanced solutions by rearranging the construction order of segments. To ensure compliance with resource constraints, a simplified linear integer model was rapidly solved at each iteration. A simulated annealing framework was implemented to guide the search process, enabling the consideration of suboptimal solutions and facilitating a comprehensive exploration of the solution space. The proposed mat-heuristic consistently delivered solutions of high quality, achieving optimality for numerous instances within an acceptable computational time.

Fortes et al. [54] introduced a mat-heuristic in addressing a rich production routing problem. An initial solution was generated by solving a vehicle-indexed formulation, using the CPLEX solver. The solution was subsequently improved by a neighborhood routing search (NRS). Three perturbation strategies were proposed to aid in escaping local optimums. The mat-heuristic achieved best-known solutions for most instances, with reduced computational effort. Meanwhile, standalone CPLEX solver was unable to find feasible solution for 31% of the instances.

Gobbi et al. [55] proposed an adaptive large neighborhood search with kernel search (ALNS-KS) to tackle a nurse routing problem. The problem was formulated as a MILP model. An initial feasible solution was generated using a greedy procedure, which was then enhanced through destroy and repair operators. The KS component was employed to enhance the current best solution when further improvement was not possible after an epoch. The KS solved a sequence of restricted integer problems utilising the GUROBI solver. The proposed mat-heuristic produced high-quality solutions within a short computational time, even for challenging scenarios.

Karsu and Solyalı [59] introduced a novel mat-heuristic known as restricted formulation-based heuristic (RFH) to address an airport gate assignment problem. A newly devised MILP formulation utilising three-indexed continuous variables was proposed. An initial feasible solution was generated by solving an approximate formulation termed as Initial Feasible Solution (IFS). The IFS formulation contributed a swift yet high-quality upper bound for the problem. The restricted version of the MILP was then solved utilising the initial solution. All the formulations were solved using the CPLEX solver. The proposed RFH produced new best-known results for all instances, surpassing other existing methods in the literature.

4.2.2. Tight-Decomposition (TDC)

Agra et al. [20] proposed a mat-heuristic called iterative local search with MILP (ILS-MILP) to tackle a maritime inventory routing problem. A robust MILP model was formulated, decomposing the problem into two components: a master problem (MP) and an adversarial separation problem (ASP). A ILS-MILP mat-heuristic was implemented to address the MP (NP-hard). The algorithm utilised the concept of local branching to enhance the solution in each iteration. The Xpress Optimiser was used to solve the MILP models in both the MP and ASP. The proposed mat-heuristic was faster than a general ILS heuristic.

Lahyani et al. [25] proposed a tabu search mat-heuristic (MTS) to tackle the multiple knapsack problem with setup (MKPS). An initial feasible solution was generated by applying linear relaxation to a MILP model. In each iteration, the problem was decomposed into several sub-problems of reduced complexity, which were then solved using the CPLEX solver. The tabu search procedure was subsequently applied to refine the best solution, utilising a class-exchange neighborhood structure and a perturbation function. The MTS mat-heuristic achieved 93 optimal solutions and established new best-known results for 185 problem instances.

Legrain et al. [29] proposed a rotation-based branch-and-price (R-BnP) to address a nurse rostering problem. The BnP method was embedded within an adaptive large neighborhood search (ALNS) framework. The problem was decomposed into a master problem and a pricing problem based on constraints. A rotation-based approach was used to reduce the complexity of the pricing problem and the number of generated feasible columns. An initial solution was found by using a rolling horizon method based on a BnP procedure. In each iteration, a portion of the solution was destroyed and repaired using ALNS. During a destruction phase, the values of a subset of variables were freed. In the repair phase, a new solution was generated from the freed variables using the BnP approach. In a repair phase, a new solution was generated by manipulating the freed variables using the BnP approach. The BnP procedure was implemented using COIN-OR Linear Programming (CLP). The proposed mat-heuristic outperformed several best-known solutions.

Wolfinger et al. [28] proposed an iterative local search with column generation (ILS-CG) to address the multimodal long haul routing problem (MMLHRP). They introduced a set-covering formulation (SCF) with additional notation. A column generation method was proposed to solve the linear relaxation of the SCF to obtain the lower bounds. Sub-problems were generated and addressed using an ILS and a label setting algorithm. The ILS algorithm comprised two phases: *Perturbation* and *LocalSearch*, which implemented two neighborhood structures, namely *Insertion* and *Removal*. The MIP formulations were solved utilising the CPLEX solver. ILS performed better than label-setting algorithm in addressing the sub-problems.

Adouani et al. [31] introduced a novel variable neighborhood descent with integer programming (VND-IP) to tackle the multiple knapsack problem with setup (MKPS). An initial solution was generated using a constructive heuristic called reduction-based heuristic (RBH). Three local search procedures were then implemented to improve the solution. These local search procedures were the combination of VND operators such as *SWAP*, *DROP/ADD*, and *INSERT* with IP. The problem was decomposed into several independent Knapsack Problems (KPs). Each independent KP was subsequently solved using the CPLEX solver. The proposed mat-heuristic outperformed IP alone in terms of computational time and solution quality.

In a general Fix-and-Optimise (FnO) mat-heuristic, the values of a subset of variables are fixed. A feasible solution is obtained using an exact method. The feasible solution is improved by manipulating the remaining variables using a heuristic approach. Wickert et al. [36] proposed a FnO mat-heuristic with a decomposition approach to address a real-world physician scheduling problem (PSP). A feasible solution was generated using the CPLEX solver. In each iteration, the values of a subset of variables were fixed, effectively decomposing the problem into sub-problems. Each sub-problem was then addressed using a variable neighborhood search (VNS). Despite that, the proposed mat-heuristic still produced near-optimal results for large instances in acceptable computational times.

Queiroga et al. [44] introduced a mat-heuristic based on the POPMUSIC framework (POPMH) to tackle a capacitated vehicle routing problem (CVRP). The problem was decomposed into a series of sub-problems. A modified version of branch-cut-and-price (BCP) was implemented as a heuristic method to address these sub-problems. The CPLEX solver was applied within the POPMUSIC framework to solve LP relaxation and MIPs. The proposed mat-heuristic produced new best solutions for several instances. However, the proposed mat-heuristic is impractical as it took several hours to generate high-quality solutions.

The iterative kernel search (IKS) is a popular heuristic framework for MILP problems [62]. Lamanna et al. [49] further improved the framework by introducing a two-phase iterative kernel search (TIKS) method to address the multiple-choice knapsack problem (MMKP). Subproblems were formulated by constraining the original problem to a selected subset of variables based on the linear programming (LP) relaxation. The TIKS algorithm consisted of two distinct phases. In the first phase, the algorithm focused on finding feasible solutions and gathering important information about the sub-problems. In the second

Table 8
List of multi integration mat-heuristics.

Mat-heuristic	Approach	COP	Base mat-heuristic	Heuristic method	Exact method	Ref.
FP-LNS	Direct	Routing	Feasibility Pump (FP)	LNS	MILP	[32]
2-MH		Scheduling	Probabilistic LP Round (PLPR)	ALNS	MILP	[34]
ILS-MH		Routing	MIP-based Mat-heuristic	ILS	MIP	[46]
HISM		Routing	ILS-MIP	GRASP	MIP	[50]
TSM		Scheduling	ILS-MIP	ILS	MIP	[56]
POPIGA	Decomposition	Scheduling	POPMUSIC-based mat-heuristic	IGA	IP, SP	[21]
FnOMH		Routing	Fix-and-Optimise (FnO)	Nearest Neighbour Heuristic, ALNS	MIP	[18]
CMH		Travelling salesman	Magnifying Glass mat-heuristic	2-opt,3-opt, lens-metaheuristic	ILP	[30]
FnOSA		Scheduling	FnO	Fix-and-Relax (FnR), SA	IP	[35]
2PM		Clustering	ILS-BIP	ILS, K-means++	BIP	[48]

phase, focus was shifted towards generating high-quality solutions. GUROBI was utilised as the mathematical solver. The effectiveness of the Kernel Search was dependent on how variables were selected and sorted, the size of sub-problems and the type of buckets used. The TIKS mat-heuristic achieved new best-known results for 185 out of 276 open benchmark instances.

Touzout et al. [51] proposed a branch-and-cut with iterative local search (BnC-ILS) to tackle the time-dependent inventory routing problem. The problem was decomposed into inventory management and routing problems. Both problems were addressed using the same mathematical formulations and algorithms. An ILS algorithm was utilised to generate travel-time constraint values, while a branch-and-cut procedure was applied to achieve optimal solutions using the GUROBI solver. The proposed mat-heuristic generated solutions with small gaps compared to the best lower bounds available, outperforming a general BnC method.

Wang et al. [60] proposed a three-phase mat-heuristic (TPHM) to address a multi-day task assignment problem. The three phases were construction, intensification and diversification. In the construction phase, the problem was decomposed into sub-problems. Solutions (sub-problems) were aggregated to produce an overall feasible solution. In the intensification phase, a variable fixing heuristic was iteratively executed and a reduced model was iteratively solved. In the diversification phase, a modified model (a distance component was added into the original objective function) was solved. The GUROBI solver was utilised. The proposed mat-heuristic was superior to (solution quality and computational time) GUROBI, LocalSolver, and a tabu search algorithm.

Kidd et al. [61] proposed an iterative relax-and-restrict mat-heuristic (RARM) to tackle a supply chain network design problem (SCND) that integrated production, facility location, inventory management, and distribution with due delivery dates. An initial feasible solution is obtained by solving a model that only assigned zeros to variables. In each iteration, the main model was relaxed by fixing location and assignment variables. The restricted model was solved using CPLEX in finding an improved solution. RARM was competitive on smaller instances and outperformed CPLEX on larger instances.

4.3. Multi integration mat-heuristic

Table 8 shows a list of multi integration mat-heuristics.

4.3.1. Multi-Direct (MD)

Assunção and Mateus [32] proposed a feasibility pump with large neighborhood search (FP-LNS) to tackle the Steiner team orienteering problem (STOP). A feasibility pump (FP) mat-heuristic was used to obtain an initial feasible solution by solving a MILP model using the CPLEX solver. The initial solution underwent further improvement through a large neighborhood search (LNS). The LNS utilised four different neighborhood structures to explore the solution space. The proposed mat-heuristic achieved exceptional results, reaching 382 out of 387 best-known bounds and obtaining new best-known results for 21 instances of the STOP problem.

Mansini and Zanotti [34] introduced two mat-heuristics (2-MH) to address a physician scheduling problem (PSP) in a large hospital ward. They proposed two distinct compact formulations based on MILP for the problem. A mat-heuristic named Probabilistic LP Round (PLPR) was developed to obtain an initial feasible solution. The initial solution obtained from PLPR was then fed into an adaptive large neighborhood search (ALNS) framework. The ALNS made use of a CPLEX solver in its destroy and repair procedures, exploiting both compact formulations. The proposed mat-heuristic demonstrated superior performance compared to the GUROBI solver in terms of solution quality and running time.

Mancini et al. [46] introduced an iterative local search-based mat-heuristic (ILS-MH) to tackle the collaborative consistent vehicle routing problem (CCVRP). Initially, a mat-heuristic called MH was proposed to explore very large neighborhoods utilising MIP. The same MH was then incorporated as a local search operator within an iterative local search (ILS) framework. Whenever a local minimum was encountered, a perturbation function was invoked, and the mat-heuristic was restarted. The MIP models were solved using the Xpress solver. The proposed mat-heuristic achieved near-optimal solutions within short computational times.

Manousakis et al. [50] introduced a two-phase mat-heuristic called hybrid infeasible space mat-heuristic (HISM) to address a production routing problem. In the first phase, a partial solution was obtained by solving a novel two-commodity flow formulation (constraint relaxation). In the second phase, the partial solution was completed using a greedy randomised adaptive search procedure (GRASP). The solution was further refined by a mat-heuristic local search algorithm equipped with exact components to diversify the search process and handle infeasibility. All mathematical formulations were solved using the GUROBI solver. The proposed mat-heuristic outperformed other existing methods, achieving new best-known results for both small-medium (999 instances) and large-scale (55 instances) problem cases.

Rappos et al. [56] proposed a two-stage mat-heuristic (TSM) to address the university course timetabling problem (UCTP). A MIP model for this problem was formulated. In the first stage, two MIP solvers (CPLEX and GUROBI) were used to address the model (only hard constraints) and derive a feasible solution. In the second stage, the feasible solution was iteratively improved in a local search heuristic framework (mat-heuristic) utilising the same solvers. The proposed mat-heuristic achieved second place in the International Timetabling Competition 2019. Finding a feasible solution was challenging due to high room and time slot occupancy when dealing with large number of students.

4.3.2. Multi-Decomposition (MDC)

Doi et al. [21] proposed a POPMUSIC mat-heuristic with an iterative greedy algorithm (POPIGA) to address the airline crew rostering problem with fair working time. An initial solution was generated using a greedy algorithm and a general-purpose solver. This initial solution served as the starting point for the POPMUSIC mat-heuristic. Sub-problems were then formulated and solved using the CPLEX solver. The proposed mat-heuristic outperformed CPLEX and CP Optimiser in terms of solution quality and running time. Finding feasible solutions for the sub-problems was challenging.

Neves-Moreira et al. [18] proposed a three-phase Fix-and-Optimise-based mat-heuristic (FnOMH) to tackle the time window assignment vehicle routing problem. In the first phase, a set of routes was generated using the nearest neighbour heuristic and adaptive large neighborhood search (ALNS). The problem was then decomposed into smaller sub-problems. In the second phase, these routes were utilised to solve a series of sub-MIPs, resulting in an initial solution. In the third phase, the initial solution was iteratively improved using a Fix-and-Optimise-based mat-heuristic with the CPLEX solver. The proposed mat-heuristic generated better solutions than general-purpose solvers.

Stanek et al. [30] addressed the Angular-Distance-Metric travelling salesman problem using a comprehensive mat-heuristic (CMH). The proposed mat-heuristic involved a diverse range of heuristic methods. Initially, lens-shaped neighborhood (a subset of the 3-opt-neighborhood) was utilised to decompose the graph into layers of convex hulls, which were then merged to form a tour. An integer linear programming (ILP) model based on the more general quadratic travelling salesman problem (QTSP) was then solved using the GUROBI solver. Several improvement heuristics were proposed such as 2-opt-heuristic, lens-metaheuristic and the "*magnifying glass mat-heuristic*", which locally re-optimised the solution for rectangular sectors of the given point set using an ILP approach. The proposed mat-heuristic demonstrated superior performance in terms of computational time and objective value compared to previously published heuristics.

Turhan and Bilgen [35] introduced a Fix-and-Optimise and simulated annealing (FnOSA) to address a nurse rostering problem. The MIP-based heuristics utilised in the approach were Fix-and-Relax (FnR) and Fix-and-Optimise (FnO). FnR served as the starting point where the problem was decomposed into smaller sub-problems. Each sub-problem was iteratively solved using the CPLEX solver to generate high-quality initial solutions. These initial solutions were then fed into a simulated annealing (SA) algorithm. When the SA algorithm could no longer improve the solutions, the FnO method was integrated, enabling the algorithm to explore a diverse search space. The proposed mat-heuristic produced seven new best results.

Gnägi and Baumann [48] proposed a two-phase mat-heuristic (2PM) for the capacitated *p*-median problem (CPMP). In a global optimisation phase, the CPMP was decomposed into a series of generalised assignment problems, solved as binary linear programming (BIP). In a local optimisation phase, the problem was decomposed into sub-problems (cluster groups) and solved as BIP. Both phases were implemented in an iterated local search framework. A K-d tree data structure was used to reduce distance computations. The formulations were solved using GUROBI. The proposed 2PM found new best-known solutions for larger instances.

5. Mathematical solvers

Mat-heuristics often leverage the power of off-the-shelf mathematical solvers to facilitate implementation and solution process. By defining the mathematical formulation or model and specifying the problem's key features such as constraints, decision variables, and objective functions, these solvers can be employed effectively.

Table 9 shows the mathematical solvers utilised by the mat-heuristics reviewed in this survey. Dash (–) symbols indicate that no mathematical solver is mentioned in the paper. Fig. 8 shows the count of mathematical solvers. CPLEX (26) seems to be the most popular solver, followed by GUROBI (14), XPRESS(2), LEMON (1) and COIN-OR (1).

Table 9
Mathematical solvers utilised by the mat-heuristics. LD = Loose-Direct, LDC = Loose-Decomposition, TD = Tight-Direct, TDC = Tight-Decomposition, MD = Multi-Direct, MDC = Multi-Decomposition.

Mat-heuristic	Category	COP	Solver	Ref.
TPM	LD	Scheduling	GUROBI	[22]
MH-LNS	LD	Routing	CPLEX	[26]
CG-DH	LD	Scheduling	CPLEX	[33]
MILP-GA-ILS	LD	Travelling salesman	CPLEX	[47]
MS-ILS	LD	Dominating set	CPLEX	[38]
SALSBM	LD	Routing	CPLEX	[40]
CS-ALNS-LB	LD	Facility location	CPLEX	[39]
IM	LD	Routing	GUROBI	[45]
RTPM	LD	Routing	CPLEX	[53]
4SD-MHA	LDC	Assignment	GUROBI	[57]
3S-MHA	LDC	Scheduling	GUROBI	[58]
ALNS-MIP	TD	Routing	GUROBI	[19]
EAMH	TD	Transportation	LEMON	[24]
FiNeMath	TD	Assignment	CPLEX	[17]
PM	TD	Scheduling	GUROBI	[23]
ILS-RVND	TD	Routing	CPLEX	[27]
GAMH	TD	Transportation	–	[37]
LNS-SPP	TD	Routing	CPLEX	[41]
MathHeu2	TD	Routing	CPLEX	[43]
GBM	TD	Routing	CPLEX	[42]
IGM	TD	Scheduling	–	[52]
TILS,OILS,IILS	TD	Routing	CPLEX	[54]
ALNS-KS	TD	Routing	GUROBI	[55]
RFH	TD	Assignment	CPLEX	[59]
ILS-MILP	TDC	Routing	Xpress	[20]
MTS	TDC	Knapsack	CPLEX	[25]
R-BnP	TDC	Scheduling	CLP	[29]
ILS-CG	TDC	Routing	CPLEX	[28]
VND-IP	TDC	Knapsack	CPLEX	[31]
FnO	TDC	Scheduling	CPLEX	[36]
POPMH	TDC	Routing	CPLEX	[44]
TIKS	TDC	Knapsack	GUROBI	[49]
BnC-ILS	TDC	Routing	GUROBI	[51]
TPMH	TDC	Assignment	GUROBI	[60]
RARM	TDC	Network design	CPLEX	[61]
FP-LNS	MD	Routing	CPLEX	[32]
2-MH	MD	Scheduling	CPLEX	[34]
ILS-MH	MD	Routing	Xpress	[46]
HISM	MD	Routing	GUROBI	[50]
TSM	MD	Scheduling	CPLEX, GUROBI	[56]
POPIGA	MDC	Scheduling	CPLEX	[21]
FnOMH	MDC	Routing	CPLEX	[18]
CMH	MDC	Travelling salesman	GUROBI	[30]
FnOSA	MDC	Scheduling	CPLEX	[35]
2PM	MDC	Clustering	GUROBI	[48]

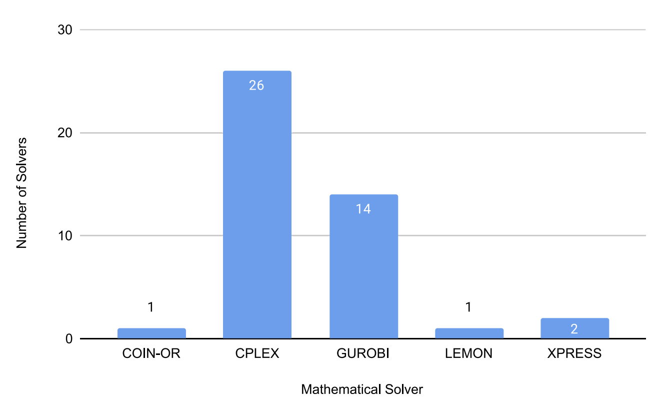

Fig. 8. Mathematical solvers utilised by the mat-heuristics.

6. Performance of mat-heuristics

Mat-heuristics have been employed to address a wide range of COPs. Their performances vary significantly as shown in Table 10. The notations <BKS, =BKS, and >BKS indicate whether the performance of a mat-heuristic is inferior, comparable, or superior to the best

Table 10
Performance of mat-heuristics. BKS = Best Known Solution, LD = Loose-Direct, LDC = Loose-Decomposition, TD = Tight-Direct, TDC = Tight-Decomposition, MD = Multi-Direct, MDC = Multi-Decomposition.

Mat-heuristic	Category	Problem	Performance	Remarks	Ref.
TPM	LD	Scheduling	**=BKS**		[22]
MH-LNS	LD	Routing	–	Results were superior to CPLEX.	[26]
CG-DH	LD	Scheduling	**<BKS**		[33]
MILP-GA-ILS	LD	Travelling salesman	**=BKS**		[47]
MS-ILS	LD	Dominating set	**>BKS**		[38]
SALSBM	LD	Routing	–	Results were superior to general GA	[40]
CS-ALNS-LB	LD	Facility location	**>BKS**		[39]
IM	LD	Routing	**=BKS**		[45]
RTPM	LD	Routing	–	Results were superior to MILP	[53]
4SD-MHA	LDC	Assignment	–	Increased sales by 90%.	[57]
3S-MHA	LDC	Scheduling	–	Proven to be a robust algorithm.	[58]
ALNS-MIP	TD	Routing	–	Results were superior to BnC.	[19]
EAMH	TD	Transportation	**=BKS**		[24]
FiNeMath	TD	Assignment	**>BKS**		[17]
PM	TD	Scheduling	–	Bucket list LS was better than other LS methods.	[23]
ILS-RVND	TD	Routing	**>BKS**		[27]
GAMH	TD	Transportation	**=BKS**		[37]
LNS-SPP	TD	Routing	**=BKS**		[41]
MathHeu2	TD	Routing	**>BKS**		[43]
GBM	TD	Routing	**<BKS**		[42]
IGM	TD	Scheduling	**=BKS**		[52]
TILS,OILS,IILS	TD	Routing	**=BKS**		[54]
ALNS-KS	TD	Routing	–	Results were superior to general EM.	[55]
RFH	TD	Assignment	**>BKS**		[59]
ILS-MILP	TDC	Routing	–	Results were superior to general ILS.	[20]
MTS	TDC	Knapsack	**>BKS**		[25]
R-BnP	TDC	Scheduling	**>BKS**		[29]
ILS-CG	TDC	Routing	–	Results were superior to label setting algorithm.	[28]
VND-IP	TDC	Knapsack	–	Results were superior to IP	[31]
FnO	TDC	Scheduling	**<BKS**		[36]
POPMH	TDC	Routing	**>BKS**		[44]
TIKS	TDC	Knapsack	**>BKS**		[49]
BnC-ILS	TDC	Routing	–	Results were superior to BnC	[51]
TPMH	TDC	Assignment	–	Results were superior to GUROBI and TS	[60]
RARM	TDC	Network design	–	Results were superior to CPLEX	[61]
FP-LNS	MD	Routing	**>BKS**		[32]
2-MH	MD	Scheduling	–	Results were superior to GUROBI.	[34]
ILS-MH	MD	Routing	**<BKS**		[46]
HISM	MD	Routing	**>BKS**		[50]
TSM	MD	Scheduling	–	Achieved second place in the competition.	[56]
POPIGA	MDC	Scheduling	–	Results were superior to CPLEX and CPO.	[21]
FnOMH	MDC	Routing	–	Results were superior to general solvers	[18]
CMH	MDC	Travelling salesman	**>BKS**		[30]
FnOSA	MDC	Scheduling	**>BKS**		[35]
2PM	MDC	Clustering	**>BKS**		[48]

known solution (BKS) in terms of hard/soft constraint violations. A mat-heuristic is notated as; >BKS (set new BKS), =BKS (achieved BKS) and <BKS (did not achieve BKS). Dash (–) symbols indicate that the performance of the mat-heuristic was not compared to BKS.

Fig. 9 shows the performance of the mat-heuristics reviewed in this survey. Based on the chart, majority of the mat-heuristics (15 out of 45) managed to set new BKS regardless of problem instances. Eight (8) of the mat-heuristics achieved BKS. Meanwhile, four (4) mat-heuristics did not achieve BKS. Eighteen (18) mat-heuristics were not benchmarked against the state-of-the-art methodologies.

Fig. 10 shows the performance of the mat-heuristics based on category. Eight (8) tight integration mat-heuristics (4 direct, 4 decomposition), five (5) multi integration mat-heuristics (2 direct, 3 decomposition) and two (2) loose integration mat-heuristics (direct) set new BKS. Meanwhile, five (5) tight integration mat-heuristics (direct) and three (3) loose integration mat-heuristics (direct) achieved BKS.

Table 11 shows the performance of the mat-heuristics based on integration type. 18.2% of loose integration mat-heuristics, 33.3% of tight integration mat-heuristics and 50% of multi integration mat-heuristics set new BKS. 27.3% of loose integration mat-heuristics and 20.8% of tight integration mat-heuristics achieved BKS. Note that a portion of the mat-heuristics were not compared to BKS (45.5% of loose integration, 37.5% of tight integration, and 40% of multi integration).

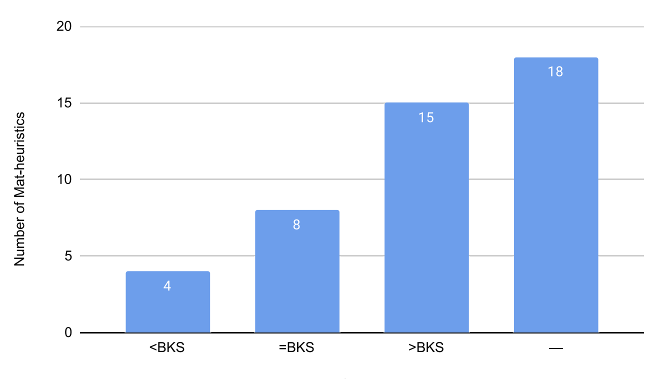

Fig. 9. Performance of mat-heuristics. BKS = Best Known Solution.

Fig. 11 shows the performance of the mat-heuristics based on COP. Mat-heuristics performed relatively well for the COPs. They managed to set new BKS for all the COPs except Transportation. In fact, the performance of the mat-heuristic utilised in the transportation problem is comparable to BKS.

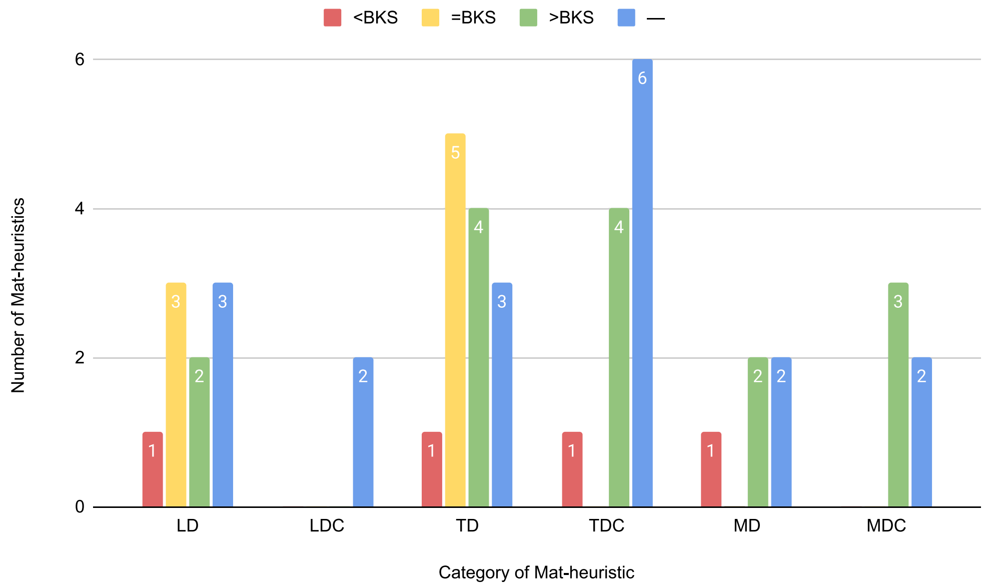

Fig. 10. Performance of mat-heuristics based on category. BKS = Best Known Solution, LD = Loose-Direct, LDC = Loose-Decomposition, TD = Tight-Direct, TDC = Tight-Decomposition, MD = Multi-Direct, MDC = Multi-Decomposition.

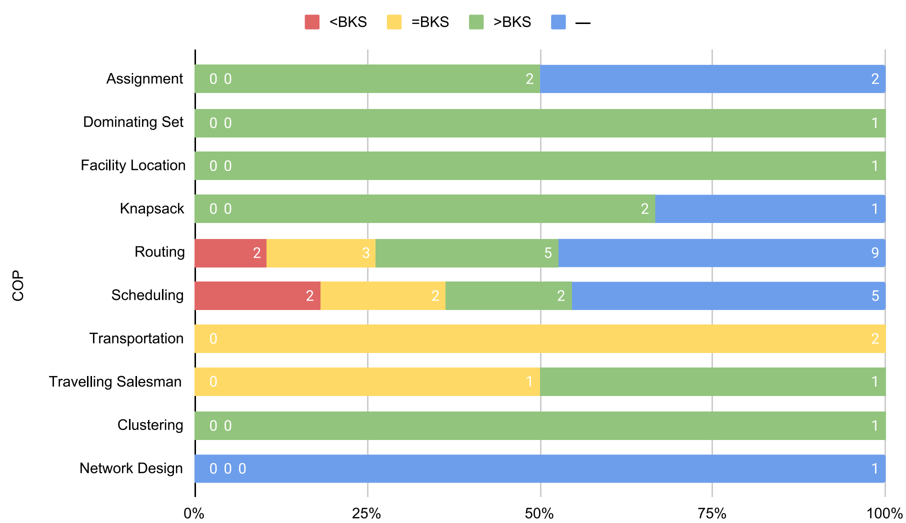

Fig. 11. Performance of mat-heuristics based on COP. BKS = Best Known Solution.

Table 11
Performance of the mat-heuristics based on integration. BKS = Best Known Solution.

Integration	Performance			
	<BKS	=BKS	>BKS	–
Loose	9.1%	27.3%	18.2%	45.5%
Tight	8.3%	20.8%	33.3%	37.5%
Multi	10.0%	0.0%	50.0%	40.0%

6.1. *Effective implementation of mat-heuristics*

Table 12 shows the implementation strategies employed in mat-heuristics that managed to set new BKS. Successful implementation of mat-heuristics usually involves the enhancement of problem formulation. For instance, Nakkala et al. [38] enhanced their ILP model with new variables and capacity constraints. Assuncao and Mateus [32] reinforced their formulation with the addition of three classes of valid inequalities. Guido et al. [17] and Penna et al. [27] introduced new constraints into their formulation. Karsu and Solyali [59] replaced old parameters with new ones, leading to optimal algorithm performance. For Lamanna et al. the success of their mat-heuristic relied on the meticulous sorting and selection of variables for inclusion in the kernel set and buckets [49].

Some researchers embraced infeasible solutions by integrating destroy and repair operators into their mat-heuristics. Candidate solutions were engaged in destruction and semi-greedy repair operations when further improvement was not achieved [39]. Legrain et al. [29] implemented multiple destruction operators with diverse strategies for selecting nurses' schedules and the number of weeks to be affected. Penna et al. [27] incorporated and penalised infeasible solutions according to the characteristics of the problem such as fixed fleet, unlimited fleet, multiple depots, backhauls, site-dependencies, split deliveries and time windows. Manousakis et al. [50] proposed a mat-heuristic that oscillated between feasible and infeasible solution spaces, significantly improving the final solution quality. Guido et al. [17] presented destroy and repair operations in their mat-heuristic.

In addition, effective mat-heuristic implementations usually involve utilising efficient neighborhood structures. Dumez et al. [43] tested combinations of neighborhood operators in addressing a vehicle routing problem with time constraints. Stanek et al. [30] implemented lens

Table 12
Implementation strategies employed in mat-heuristics that managed to set new BKS. LD = Loose-Direct, LDC = Loose-Decomposition, TD = Tight-Direct, TDC = Tight-Decomposition, MD = Multi-Direct, MDC = Multi-Decomposition.

Mat-heuristic	Category	Problem	Implementation strategy	Ref.
MS-ILS	LD	Dominating set	Formulation enhancement	[38]
CS-ALNS-LB	LD	Facility location	Embracing infeasible solution	[39]
FiNeMath	TD	Assignment	Formulation enhancement, Embracing infeasible solution	[17]
ILS-RVND	TD	Routing	Formulation enhancement, Embracing infeasible solution	[27]
MathHeu2	TD	Routing	Efficient neighborhood structures	[43]
RFH	TD	Assignment	Formulation enhancement	[59]
MTS	TDC	Knapsack	High-quality initial solution	[25]
R-BnP	TDC	Scheduling	Embracing infeasible solution	[29]
POPMH	TDC	Routing	High-quality initial solution	[44]
TIKS	TDC	Knapsack	Formulation enhancement	[49]
FP-LNS	MD	Routing	Formulation enhancement	[32]
HISM	MD	Routing	Embracing infeasible solution	[50]
CMH	MDC	Travelling salesman	Efficient neighborhood structures	[30]
FnOSA	MDC	Scheduling	Diversification	[35]
2PM	MDC	Clustering	Efficient neighborhood structures	[48]

neighborhood (a subset of the 3-opt-neighborhood) that can be evaluated quickly and allowed substantial diversification of the current solution. Gnägi and Baumann [48] implemented an efficient k-d tree structures (binary tree) to eliminate large portions of the search space to reduce computational effort when dealing with large instances.

Furthermore, initial solution quality seems to affect the performance of a mat-heuristic. Lahyani et al. [25] and Queiroga et al. [44] leveraged on good initial solutions in their mat-heuristics to obtain high quality solutions.

Search space diversification plays a pivotal role in the design of an effective mat-heuristic that involves heuristic methodologies that are susceptible to local optima. Turhan and Bilgen [35] proposed a component (Fix-and-Optimise) that effectively diversified the search space of their simulated annealing algorithm.

7. Advantages and challenges

Mat-heuristic is a hybrid approach that combines the strengths of exact methods and heuristics. It utilises the precision (exploitation) of exact methods and the robustness (exploration and exploitation) of metaheuristics in tackling various complex and large-scale COPs effectively (high quality solutions). It benefits from the synergy between exploration (searching unexplored areas of the solution space) and exploitation (intensively searching near promising regions) [63,64]. Exploring the interplay of the integration (exact methods and heuristics) may lead to further advancements, unlocking new possibilities for optimisation in diverse domains.

The suitability of direct and decomposition approaches depends on the characteristics of COP such as problem size and constraints, and the desired level of solution quality and computational efficiency. Mat-heuristics with direct approach are often preferred for problems that require a holistic and integrated search strategy for significant performance gains. On the other hand, mat-heuristics with decomposition approach are well-suited for large-scale and complex COPs, where dividing the problem into smaller sub-problems allows for efficient parallelisation and optimisation. Careful planning is required while breaking down a problem into sub-problems. Ensuring the synchronisation and coherence among the sub-problems is essential, as misalignment could hinder the overall effectiveness of a mat-heuristic. Despite the challenges, sub-problems are more manageable, offering the potential for improved efficiency and solution quality.

Based on observation in [19,21,26,31,34,53,55,60,61], mat-heuristics were generally faster than mathematical solvers in solving COPs. In addition, mat-heuristics reportedly outperformed general solvers (in terms of solution quality), based on the same time limit. Furthermore, general solvers often face difficulty in finding a feasible solution for larger problem instances.

With regard to challenges, mat-heuristics require fine-tuning to perform optimally. The effectiveness of a mat-heuristic depends on heuristics, integration strategies, and parameter values (component algorithms) [35,49,56]. Finding the right balance can be time-consuming and demands extensive experimentation.

Moreover, mat-heuristics may not guarantee optimality. The quality of solutions can vary, depending on the problem instance. While they may produce excellent solutions for some instances, they might struggle to handle others effectively [53,56]. The performance of a mat-heuristic may be influenced by the initial solutions [25,44].

Mat-heuristics, particularly multi integration, are highly complex and may exhibit high computational overhead. Combining multiple algorithms and techniques may lead to increased processing times, making them very challenging for real-time or time-critical applications. Mat-heuristics may face challenges when dealing with large-scale or highly constrained COPs [21,41–44,49], where finding high-quality solutions within reasonable times is difficult.

Mat-heuristic is a promising approach to tackle a wide range of COPs. However, careful consideration (design, implementation and fine-tuning) is essential to harness their full potential and achieve superior results.

8. Research opportunities

One promising direction is the integration of mat-heuristics with machine learning techniques [65]. Machine learning algorithms can be used to learn from past solutions and guide the search process in mat-heuristics. Reinforcement learning, neural networks, and other machine learning approaches can help mat-heuristics adapt and improve their performance over time, leading to more efficient and effective optimisation strategies.

Extending mat-heuristics to handle multi-objective optimisation problems is another exciting opportunity. Many real-world problems involve multiple conflicting objectives, and mat-heuristics can be extended to find a set of solutions that represent trade-offs between these objectives. Techniques such as Pareto optimisation [66] and evolutionary algorithms [67] can be integrated into mat-heuristics to explore the multi-objective solution space efficiently. Many real-world COPs involve uncertainty and stochasticity, such as uncertain demand or travel times. Future research can focus on developing mat-heuristics that can handle such uncertainties and provide solutions that are resilient to variations in input parameters.

Some authors hope to apply their proposed mat-heuristic to different COPs, exploring its applicability in real-world problems, or evaluating its performance on various problem instances. For instance, Ying and Lin [22] hope to generalise their proposed two-phase mat-heuristic (TPM) algorithm to other scheduling problems for further performance measures. Guido et al. [17] hope to test their FineMath algorithm on benchmarks of the red-blue transportation problems. Cosma et al. [37] suggest testing their genetic algorithm-based mat-heuristic

(GAMH) algorithm on larger instances. Lahyani et al. [25] hope to test the proposed tabu search mat-heuristic (MTS) on the real-world multiple knapsack problem with setup, while Adouani et al. [31] hope to test their variable neighborhood descent with integer programming (VND-IP) on other variants of knapsack problems. Assuncao and Mateus [32] hope to extend the proposed feasibility pump with large neighborhood search (FP-LNS) to other routing problems. Stanek et al. [30] hope to apply the proposed comprehensive mat-heuristic (CMH) to other COPs. Turhan and Bilgen [35] plan to test the proposed Fix-and-Optimise and simulated annealing (FnOSA) on high school timetabling problems. Gnägi and Baumann [48] believe that their decomposition strategies can be applied to other related problems such as the capacitated *p*-centre problem.

Several researchers hope to integrate their proposed mat-heuristic with other exact methods, heuristic approaches and methodologies to enhance algorithm efficiency and solution quality. For example, Nakkala et al. [38] hope to explore different ways of combining integer linear programming with other (meta-) heuristics. Machado et al. [42] hope to integrate constructive heuristics and set-covering problem (SCP) formulation into their proposed GRASP-based mat-heuristic (GBH). Dumez et al. [41] and Touzout et al. [51] plan to incorporate dynamic programming into their proposed mat-heuristic.

Some researchers hope to enhance their proposed mat-heuristic by introducing sophisticated local search procedures and novel neighborhood structures. E.g. Calvete et al. [24] plan to add new local search components into their proposed evolutionary algorithm framework mat-heuristic (EAMH). Manousakis et al. [50] suggest to introduce novel neighborhood structures for their proposed hybrid infeasible space mat-heuristic (HISM) to explore disconnected search space. Souto et al. [39] intend to add repairer and destructive procedures into their proposed CS-ALNS-LB. Rappos et al. [56] plan to develop an improved local search method to aid their two-stage mat-heuristic (TSM) in escaping local optimums. Penna et al. [27] suggest to employ additional neighborhood structures into their proposed ILS with randomised variable neighborhood descent (ILS-RVND).

Various researchers hope to incorporate more constraints into their problem formulation to test the robustness and applicability of their mat-heuristics to real-world scenarios. For example, Cacchiani et al. [47] hope to extend the MILP-GA-ILS to deal with time-window constraints in addressing travelling salesman problems. Mansini and Zanotti [34] plan to identify new and stronger valid inequalities for the physician scheduling problem. Bigler et al. [57] hope to incorporate the conflict constraints with branching rules into the proposed four-step decomposition strategy mat-heuristic (4SD-MHA) in addressing the customer assignment problem. Neves-Moreira et al. [18] plan to add realism constraints into their proposed formulation in addressing the time window assignment vehicle routing problem. Gobbi et al. [55] suggest that uncertainties and possible disruptions should be considered when addressing nurse routing problems. Legrain et al. [29] plan to incorporate other constraints into their proposed rotation model for the nurse rostering problem. Wolfinger et al. [28] suggested adding a "green" objective to the problem formulation such as minimising carbon dioxide emissions in addressing multimodal long-haul routing problems. Wickert et al. [36] hope to formulate a general model that can cover the constraints present in both physician and nurse rostering problems. Doi et al. [21] hope to include real-world constraints into their proposed model in addressing the airline crew rostering problem. Akbarzadeh et al. [33] plan to incorporate uncertainty and to include more resource types in addressing real-world nurse rostering problems. Karsu and Solyali [59] hope to utilise newer technologies such as the Internet of Things (IoT) or Video Analytics to determine which resources to be used in problem formulation, in addressing airport gate assignment problems. Others hope to fine-tune the parameters of their mat-heuristic to improve its performance and achieve better results on various problem instances such as two-phase mat-heuristic (TPM) [22], column generation with diving heuristic (CG-DH) [33], robust two-phase mat-heuristic (RTPM) [53], two-phase iterative kernel search (TIKS) [49], three-steps mat-heuristic (3S-MHA) [58].

9. Conclusion

This review (spanning the period from 2018 to 2024) provides a thorough analysis of mat-heuristics in addressing a diverse range of Combinatorial Optimisation Problems (COPs). The mat-heuristics are categorised into six categories based on the combinations of three integration types (loose, tight and multi) and two approaches (direct and decomposition). Next, the mechanism of each mat-heuristic and its performance in a comparison to other state-of-the-art solution methodologies are presented. Highly effective mat-heuristics are analysed. The implementation strategies employed by these methodologies are presented. An in-depth discussion on the advantages, challenges and future research opportunities is provided. We believe that this survey paper will be valuable to researchers in planning their research as well as practitioners in this domain.

CRediT authorship contribution statement

Chong Man Ngoo: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Say Leng Goh:** Writing – review & editing, Supervision. **San Nah Sze:** Writing – review & editing, Validation, Methodology, Investigation. **Nasser R. Sabar:** Writing – review & editing, Validation, Methodology, Investigation. **Mohd Hanafi Ahmad Hijazi:** Writing – review & editing, Validation. **Graham Kendall:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work was supported by Universiti Malaysia Sabah under Grant SBK0463-2021.

References

[1] R.E. Bellman, An Application of Dynamic Programming to the Coloring of Maps, RAND Corporation, Santa Monica, CA, 1964.

[2] V. Maniezzo, P. Hansen, S. Voss, Special issue on mathematical contributions to metaheuristics editorial, J. Heuristics 15 (3) (2008) 197–199, http://dx.doi.org/10.1007/s10732-008-9093-z.

[3] P. Vansteenwegen, W. Souffriau, K. Sörensen, The Mobile Mapping Van Problem: a matheuristic for capacitated arc routing with soft time windows and depot selection, IFAC Proc. Vol. 42 (4) (2009) 1114–1119, http://dx.doi.org/10.3182/20090603-3-RU-2001.0297, 13th IFAC Symposium on Information Control Problems in Manufacturing.

[4] J. Mandziuk, New shades of the vehicle routing problem: Emerging problem formulations and computational intelligence solution methods, IEEE Trans. Emerg. Top. Comput. Intell. 3 (3) (2019) 230–244, http://dx.doi.org/10.1109/TETCI.2018.2886585.

[5] J.S. Tan, S.L. Goh, G. Kendall, N.R. Sabar, A survey of the state-of-the-art of optimisation methodologies in school timetabling problems, Expert Syst. Appl. 165 (2021) 113943, http://dx.doi.org/10.1016/j.eswa.2020.113943.

[6] C.M. Ngoo, S.L. Goh, S.N. Sze, N.R. Sabar, S. Abdullah, G. Kendall, A survey of the nurse rostering solution methodologies: The state-of-the-art and emerging trends, IEEE Access 10 (2022) 56504–56524, http://dx.doi.org/10.1109/ACCESS.2022.3177280.

[7] Z.X. Loke, S.L. Goh, G. Kendall, S. Abdullah, N.R. Sabar, Portfolio optimization problem: A taxonomic review of solution methodologies, IEEE Access 11 (2023) 33100–33120, http://dx.doi.org/10.1109/ACCESS.2023.3263198.

[8] V. Maniezzo, M.A. Boschetti, T. Stutzle, Matheuristics: Algorithms and Implementations, first ed., Springer Cham, 2021, http://dx.doi.org/10.1007/978-3-030-70277-9, 2364-687X.

[9] M.A. Boschetti, V. Maniezzo, M. Roffilli, A. Bolufé Röhler, Matheuristics: Optimization, simulation and control, in: M.J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, A. Schaerf (Eds.), Hybrid Metaheuristics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 171–177.

[10] C. Archetti, M. Speranza, A survey on Matheuristics for routing problems, EURO J. Comput. Optim. 2 (4) (2014) 223–246, http://dx.doi.org/10.1007/s13675-014-0030-7.

[11] K.F. Doerner, V. Schmid, Survey matheuristics for rich vehicle routing problems, Hybrid Metaheuristics (2010) 206–221, http://dx.doi.org/10.1007/978-3-642-16054-7_15.

[12] E.K. Burke, G. Kendall, Search methodologies introductory tutorials in optimization and decision support techniques, 2014.

[13] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Comput. Surv. 35 (3) (2003) 268–308, http://dx.doi.org/10.1145/937503.937505.

[14] J. Puchinger, G.R. Raidl, Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification, Artif. Intell. Knowl. Eng. Appl.: Bioinspired Appr. (2005) 41–53, http://dx.doi.org/10.1007/11499305_5.

[15] M.O. Ball, Heuristics based on Mathematical Programming, Surv. Oper. Res. Manag. Sci. 16 (1) (2011) 21–38, http://dx.doi.org/10.1016/j.sorms.2010.07.001.

[16] G. Kendall, Beall's legacy in the battle against predatory publishers, Learn. Publ. 34 (3) (2021) 379–388, http://dx.doi.org/10.1002/leap.1374.

[17] R. Guido, M.C. Groccia, D. Conforti, An efficient matheuristic for offline patient-to-bed assignment problems, European J. Oper. Res. 268 (2) (2018) 486–503, http://dx.doi.org/10.1016/j.ejor.2018.02.007.

[18] F. Neves-Moreira, D. Pereira da Silva, L. Guimarães, P. Amorim, B. Almada-Lobo, The time window assignment vehicle routing problem with product dependent deliveries, Transp. Res. E 116 (2018) 163–183, http://dx.doi.org/10.1016/j.tre.2018.03.004.

[19] T. A. Guimarães, L. C. Coelho, C. M. Schenekemberg, C. T. Scarpin, The two-echelon multi-depot inventory-routing problem, Comput. Oper. Res. 101 (2018) 220–233, http://dx.doi.org/10.1016/j.cor.2018.07.024.

[20] A. Agra, M. Christiansen, L.M. Hvattum, F. Rodrigues, Robust optimization for a maritime inventory routing problem, Transp. Sci. 52 (3) (2018) 509–525, http://dx.doi.org/10.1287/trsc.2017.0814.

[21] T. Doi, T. Nishi, S. Voß, Two-level decomposition-based matheuristic for airline crew rostering problems with Fair Working Time, European J. Oper. Res. 267 (2) (2017) 428–438, http://dx.doi.org/10.1016/j.ejor.2017.11.046.

[22] K.-C. Ying, S.-W. Lin, Minimizing makespan for no-wait flowshop scheduling problems with Setup Times, Comput. Ind. Eng. 121 (2018) 73–81, http://dx.doi.org/10.1016/j.cie.2018.05.030.

[23] B. Maenhout, M. Vanhoucke, A perturbation matheuristic for the Integrated Personnel Shift and task re-scheduling problem, European J. Oper. Res. 269 (3) (2018) 806–823, http://dx.doi.org/10.1016/j.ejor.2018.03.005.

[24] H.I. Calvete, C. Galé, J.A. Iranzo, P. Toth, A matheuristic for the two-stage fixed-charge transportation problem, Comput. Oper. Res. 95 (2018) 113–122, http://dx.doi.org/10.1016/j.cor.2018.03.007.

[25] R. Lahyani, K. Chebil, M. Khemakhem, L.C. Coelho, Matheuristics for solving the multiple knapsack problem with setup, Comput. Ind. Eng. 129 (2019) 76–89, http://dx.doi.org/10.1016/j.cie.2019.01.010.

[26] G. Macrina, G. Laporte, F. Guerriero, L. Di Puglia Pugliese, An energy-efficient green-vehicle routing problem with mixed vehicle fleet, partial battery recharging and time windows, European J. Oper. Res. 276 (3) (2019) 971–982, http://dx.doi.org/10.1016/j.ejor.2019.01.067.

[27] P.H. Penna, A. Subramanian, L.S. Ochi, T. Vidal, C. Prins, A hybrid heuristic for a broad class of vehicle routing problems with Heterogeneous Fleet, Ann. Oper. Res. 273 (1–2) (2019) 5–74, http://dx.doi.org/10.1007/s10479-017-2642-9.

[28] D. Wolfinger, F. Tricoire, K.F. Doerner, A matheuristic for a multimodal long haul routing problem, EURO J. Transp. Logist. 8 (4) (2019) 397–433, http://dx.doi.org/10.1007/s13676-018-0131-1.

[29] A. Legrain, J. Omer, S. Rosat, A rotation-based branch-and-price approach for the nurse scheduling problem, Math. Program. Comput. 12 (3) (2019) 417–450, http://dx.doi.org/10.1007/s12532-019-00172-4.

[30] R. Staněk, P. Greistorfer, K. Ladner, U. Pferschy, Geometric and LP-based heuristics for angular travelling salesman problems in the plane, Comput. Oper. Res. 108 (2019) 97–111, http://dx.doi.org/10.1016/j.cor.2019.01.016.

[31] Y. Adouani, B. Jarboui, M. Masmoudi, Efficient matheuristic for the generalised multiple knapsack problem with setup, Eur. J. Ind. Eng. 14 (5) (2020) 715, http://dx.doi.org/10.1504/ejie.2020.109906.

[32] L. Assunção, G.R. Mateus, Coupling feasibility pump and large neighborhood search to solve the steiner team orienteering problem, Comput. Oper. Res. 128 (2020) 105175, http://dx.doi.org/10.1016/j.cor.2020.105175.

[33] B. Akbarzadeh, G. Moslehi, M. Reisi-Nafchi, B. Maenhout, A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering, J. Sched. 23 (2) (2020) 265–288, http://dx.doi.org/10.1007/s10951-020-00639-6.

[34] R. Mansini, R. Zanotti, Optimizing the physician scheduling problem in a large hospital ward, J. Sched. 23 (3) (2020) 337–361, http://dx.doi.org/10.1007/s10951-019-00614-w.

[35] A.M. Turhan, B. Bilgen, A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem, Comput. Ind. Eng. 145 (2020) 106531, http://dx.doi.org/10.1016/j.cie.2020.106531.

[36] T.I. Wickert, A.F. Kummer Neto, M.M. Boniatti, L.S. Buriol, An integer programming approach for the physician rostering problem, Ann. Oper. Res. 302 (2) (2020) 363–390, http://dx.doi.org/10.1007/s10479-020-03552-5.

[37] O. Cosma, P.C. Pop, D. Danciulescu, A novel matheuristic approach for a two-stage transportation problem with fixed costs associated to the routes, Comput. Oper. Res. 118 (2020) 104906, http://dx.doi.org/10.1016/j.cor.2020.104906.

[38] M.R. Nakkala, A. Singh, A. Rossi, Multi-start iterated local search, exact and matheuristic approaches for minimum capacitated dominating set problem, Appl. Soft Comput. 108 (2021) 107437, http://dx.doi.org/10.1016/j.asoc.2021.107437.

[39] G. Souto, I. Morais, G.R. Mauri, G.M. Ribeiro, P.H. Gonzalez, A hybrid matheuristic for the two-stage capacitated facility location problem, Expert Syst. Appl. 185 (2021) 115501, http://dx.doi.org/10.1016/j.eswa.2021.115501.

[40] O. Ozkan, Optimization of the distance-constrained multi-based multi-uav routing problem with simulated annealing and local search-based matheuristic to detect forest fires: The case of Turkey, Appl. Soft Comput. 113 (2021) 108015, http://dx.doi.org/10.1016/j.asoc.2021.108015.

[41] D. Dumez, F. Lehuédé, O. Péton, A large neighborhood search approach to the vehicle routing problem with delivery options, Transp. Res. B 144 (2021) 103–132, http://dx.doi.org/10.1016/j.trb.2020.11.012.

[42] A.M. Machado, G.R. Mauri, M.C. Boeres, R.d. Rosa, A new hybrid matheuristic of grasp and VNS based on constructive heuristics, set-covering and set-partitioning formulations applied to the capacitated vehicle routing problem, Expert Syst. Appl. 184 (2021) 115556, http://dx.doi.org/10.1016/j.eswa.2021.115556.

[43] D. Dumez, C. Tilk, S. Irnich, F. Lehuédé, O. Péton, Hybridizing large neighborhood search and exact methods for generalized vehicle routing problems with time windows, EURO J. Transp. Logist. 10 (2021) 100040, http://dx.doi.org/10.1016/j.ejtl.2021.100040.

[44] E. Queiroga, R. Sadykov, E. Uchoa, A POPMUSIC matheuristic for the capacitated vehicle routing problem, Comput. Oper. Res. 136 (2021) 105475, http://dx.doi.org/10.1016/j.cor.2021.105475.

[45] S.T. Vadseth, H. Andersson, M. Stålhane, An iterative matheuristic for the inventory routing problem, Comput. Oper. Res. 131 (2021) 105262, http://dx.doi.org/10.1016/j.cor.2021.105262.

[46] S. Mancini, M. Gansterer, R.F. Hartl, The collaborative consistent vehicle routing problem with workload balance, European J. Oper. Res. 293 (3) (2021) 955–965, http://dx.doi.org/10.1016/j.ejor.2020.12.064.

[47] V. Cacchiani, C. Contreras-Bolton, L.M. Escobar-Falcón, P. Toth, A matheuristic algorithm for the pollution and energy minimization traveling salesman problems, Int. Trans. Oper. Res. 30 (2) (2021) 655–687, http://dx.doi.org/10.1111/itor.12991.

[48] M. Gnägi, P. Baumann, A matheuristic for large-scale capacitated clustering, Comput. Oper. Res. 132 (2021) 105304, http://dx.doi.org/10.1016/j.cor.2021.105304.

[49] L. Lamanna, R. Mansini, R. Zanotti, A two-phase kernel search variant for the multidimensional multiple-choice Knapsack problem, European J. Oper. Res. 297 (1) (2022) 53–65, http://dx.doi.org/10.1016/j.ejor.2021.05.007.

[50] E.G. Manousakis, G.A. Kasapidis, C.T. Kiranoudis, E.E. Zachariadis, An infeasible space exploring matheuristic for the production routing problem, European J. Oper. Res. 298 (2) (2022) 478–495, http://dx.doi.org/10.1016/j.ejor.2021.05.037.

[51] F.A. Touzout, A.-L. Ladier, K. Hadj-Hamou, An assign-and-route matheuristic for the time-dependent inventory routing problem, European J. Oper. Res. 300 (3) (2022) 1081–1097, http://dx.doi.org/10.1016/j.ejor.2021.09.025.

[52] D. Canca, G. Laporte, Solving real-size Stochastic Railway Rapid Transit network construction scheduling problems, Comput. Oper. Res. 138 (2022) 105600, http://dx.doi.org/10.1016/j.cor.2021.105600.

[53] S. Frifita, H.M. Afsar, F. Hnaien, A robust optimization approach for disassembly assembly routing problem under uncertain yields, Expert Syst. Appl. 202 (2022) 117304, http://dx.doi.org/10.1016/j.eswa.2022.117304.

[54] A. Fortes, R. Camargo, L.R. Muniz, F.M. Lima, F.d. Cota, Efficient matheuristics to solve a rich production-routing problem, Comput. Ind. Eng. 171 (2022) 108369, http://dx.doi.org/10.1016/j.cie.2022.108369.

[55] A. Gobbi, D. Manerba, R. Mansini, R. Zanotti, Hybridizing adaptive large neighborhood search with kernel search: A new solution approach for the nurse routing problem with incompatible services and minimum demand, Int. Trans. Oper. Res. 30 (1) (2022) 8–38, http://dx.doi.org/10.1111/itor.13127.

[56] E. Rappos, E. Thiemard, S. Robert, J.-F. Heche, A mixed-integer programming approach for solving university course timetabling problems, J. Sched. 25 (4) (2022) 391–404, http://dx.doi.org/10.1007/s10951-021-00715-5.

[57] T. Bigler, M. Kammermann, P. Baumann, A matheuristic for a customer assignment problem in direct marketing, European J. Oper. Res. 304 (2) (2023) 689–708, http://dx.doi.org/10.1016/j.ejor.2022.04.009.

[58] M. Boccia, A. Masone, C. Sterle, T. Murino, The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach, European J. Oper. Res. 307 (2) (2023) 590–603, http://dx.doi.org/10.1016/j.ejor.2022.10.023.

[59] O. Karsu, O. Solyalı, A new formulation and an effective matheuristic for the airport gate assignment problem, Comput. Oper. Res. (2023) http://dx.doi.org/10.2139/ssrn.4055150.

[60] Y. Wang, H. Liu, B. Peng, H. Wang, A.P. Punnen, A three-phase matheuristic algorithm for the multi-day task assignment problem, Comput. Oper. Res. 159 (2023) 106313, http://dx.doi.org/10.1016/j.cor.2023.106313.

[61] M.P. Kidd, M. Darvish, L.C. Coelho, B. Gendron, A relax-and-restrict matheuristic for supply chain network design with facility location and customer due date flexibility, Transp. Res. E 182 (2024) 103370, http://dx.doi.org/10.1016/j.tre.2023.103370.

[62] E. Angelelli, R. Mansini, M.G. Speranza, Kernel search: A new heuristic framework for portfolio selection, Comput. Optim. Appl. 51 (1) (2010) 345–361, http://dx.doi.org/10.1007/s10589-010-9326-6.

[63] S.L. Tilahun, Balancing the degree of exploration and exploitation of swarm intelligence using parallel computing, Int. J. Artif. Intell. Tools 28 (03) (2019) 1950014, http://dx.doi.org/10.1142/S0218213019500143.

[64] B. Saib, M.-R. Abdessemed, R. Hocin, K. Khoualdi, Study of exploration and exploitation mechanisms in nature inspired metaheuristics for global optimization, in: M.R. Laouar, V.E. Balas, B. Lejdel, S. Eom, M.A. Boudia (Eds.), 12th International Conference on Information Systems and Advanced Technologies "ICISAT 2022", Springer International Publishing, Cham, 2023, pp. 442–453.

[65] Y. Bengio, A. Lodi, A. Prouvost, Machine learning for combinatorial optimization: A methodological tour d'horizon, European J. Oper. Res. 290 (2) (2021) 405–421, http://dx.doi.org/10.1016/j.ejor.2020.07.063.

[66] N. Gunantara, A review of multi-objective optimization: Methods and its applications, Cogent Eng. 5 (1) (2018) 1502242, http://dx.doi.org/10.1080/23311916.2018.1502242.

[67] P.A. Vikhar, Evolutionary algorithms: A critical review and its future prospects, in: 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication, ICGTSPICC, 2016, pp. 261–265, http://dx.doi.org/10.1109/ICGTSPICC.2016.7955308.