IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 17, 2022, accepted May 19, 2022, date of publication May 23, 2022, date of current version June 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3177280

A Survey of the Nurse Rostering Solution Methodologies: The State-of-the-Art and Emerging Trends

CHONG MAN NGOO[1], SAY LENG GOH[1], SAN NAH SZE[2], NASSER R. SABAR[3], SALWANI ABDULLAH[4], AND GRAHAM KENDALL[5], (Senior Member, IEEE)

[1]Optimisation Research Group, Faculty of Computing and Informatics, Universiti Malaysia Sabah Kampus Antarabangsa Labuan, Labuan 87000, Malaysia
[2]Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak 94300, Malaysia
[3]Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC 3086, Australia
[4]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, Selangor 43600, Malaysia
[5]School of Computer Science, University of Nottingham Malaysia Campus, Semenyih, Selangor 43500, Malaysia

Corresponding authors: Say Leng Goh (slgoh@ums.edu.my) and San Nah Sze (snsze@unimas.my)

This work was supported by the Universiti Malaysia Sarawak under Grant F08/PARTNERS/2109/2021.

ABSTRACT This paper presents an overview of recent advances for the Nurse Rostering Problem (NRP) based on methodological papers published between 2012 to 2021. It provides a comprehensive review of the latest solution methodologies, particularly computational intelligence (CI) approaches, utilized in benchmark and real-world nurse rostering. The methodologies are systematically categorised (Heuristics, Meta-heuristics, Hyper-heuristics, Mathematical Optimisation, Matheuristics and Hybrid Approaches). The NRP benchmark repositories and the respective state-of-the-art methods are also presented. A distinctive feature of this survey is its focus on the emerging trends in terms of solution methodologies and benchmark datasets. Meta-heuristics are the most popular choices in addressing NRP. Matheuristics, one of most popular methodologies in addressing the NRP, has been an emerging trend in recent years (2018 onwards). The INRC-I dataset is the most popular benchmark currently in use by researchers to test their algorithms. An in-depth discussion on the challenges and research opportunities is provided. The summary and analysis of the recently published NRP methodological papers in this survey is valuable for the CI and Operational Research (OR) communities especially early career researchers seeking to find gaps and identify emerging trends in this fast-developing, important research area.

INDEX TERMS Combinatorial optimization, nurse rostering, nurse scheduling, computational intelligence, operational research.

I. INTRODUCTION

The Nurse Rostering Problem (NRP) is one of many challenging combinatorial optimisation problems (COP) [5], [6], [27], [42], [47], [67], [75]. The NRP was introduced by Miller et. al. [58] and Warner [81] in 1976 and proven to be NP-hard by Osogami and Imai [61], in 2000. The NRP is a specific type of personnel scheduling problem and plays an important role in healthcare management. It involves the assignment of shifts to nurses, with different skills, over a given planning horizon (e.g. 1 month) and requires satisfying a set of hard and soft constraints. The goal is to improve the operational efficiency of hospital wards by aiming to have an optimal utilization of the limited resources, with a focus on the well-being of patients and the job satisfaction of nurses.

The NRP is widely researched due to its practical relevance and combinatorial complexity, making it a challenging and important problem. The importance of NRP stems from its direct application in healthcare organisations. An optimal nurse roster improves the efficiency of hospitals and addresses issues such as under & over staffing, skill matching and job satisfaction. A nurse-centred roster improves the morale of nurses, positively impacting the quality of service provided and the well-being of patients. Nurse rosters can be created manually but generating high quality schedules is challenging and time consuming [26]. Automating this

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

process provides the opportunity to generate better quality schedules in shorter time. The NRP has recently gathered increasing attention from governments and researchers amid the COVID-19 pandemic, which has greatly impacted healthcare front liners in terms of psychological well-being, morale and work performance due to long working hours under stressful conditions.

There are a number of survey papers that address the NRP. For example, [26] presented a bibliographic survey providing an overview of NRP models and solution approaches such as mathematical programming, constraint programming, artificial intelligence (AI) and meta-heuristics. Burke *et al.* [14] provided a comprehensive review on nurse rostering. They described, and critically evaluated, the state-of-the-art solution approaches, including OR and AI methodologies. The paper presented the strengths, weaknesses and scientific achievements of these approaches over the last forty years. In addition, key issues that should be addressed in the future were highlighted. Causmaecker and Berghe [22], rather than focusing on solution approaches, considered problem-related features and abstract models for the NRP. A framework for categorizing NRP was introduced by using the notations $\alpha \mid \beta \mid \gamma$ that were used to classify scheduling problems. α referred to the personnel environment (number of nurses, skills and availability), β referred to the work characteristics (time structure and services to be delivered) and γ referred to the optimization objectives.

These survey papers were published several years ago and many new approaches have appeared since. Therefore, it is timely for this paper, which provides a comprehensive review of the latest works in nurse rostering. The analysis of the recently published NRP papers included in this survey enables the Operational Research (OR) and Computational Intelligence (CI) communities, especially early career researchers, to understand the research problem and identify emerging trends in this fast-developing, important research area.

The contributions of this paper are:

- We provide the problem definition, terminology, constraints and variants of the NRP. Recent solution methodologies for the benchmark and real-world NRP are presented in chronological order where the mechanisms and performance of each methodology are analysed. In addition, the methodologies are categorised (Heuristics, Meta-heuristics, Hyper-heuristics, Mathematical Optimisation, Matheuristics, Hybrid Approaches). The benchmark repositories are shown and the respective state-of-the-art methods are identified and presented. Furthermore, readers are referred to the existing work the NRP from a real-world perspective.
- We present the primary methodologies for the NRP, on a timeline (by year) (see Table 2). In addition, we group the methodologies by datasets in an effort to determine the most popular benchmark datasets.
- We discuss the advantages and drawbacks of the solution methodologies and suggest future directions.

TABLE 1. List of journals considered in this survey.

Journal	Paper Count
Annals of Operations Research	9
European Journal of Operational Research	8
Computers & Operations Research	7
Computers & Industrial Engineering	4
Applied Soft Computing	2
International Journal of Computational Intelligence Systems	2
Neural Computing and Applications	2
Others	16
Total	50

The remainder of the paper is organised as follows: Section II describes the scope of the paper and the methodology adopted. Section III discusses the problem definition, terminologies, and problem constraints. Section IV provides an overview and classification of the methodologies applied to NRP. The advantages and challenges for the method categories are presented in Section V. Section VI highlights the popular NRP benchmark datasets and their respective state-of-the-art methodologies. In addition, the real-world nurse rostering problems are presented. Finally, sections VII and VIII present potential future directions and concluding remarks.

II. SURVEY SCOPE AND METHODOLOGY

The aim of this survey is to record the evolution of NRP and its solution methodologies on a timeline. About 100 papers, related to NRP, published between 2012 and 2021 were collected from a range of bibliographic databases. The keywords used for the search included "*nurse rostering problem*", "*nurse scheduling problem*", and "*optimization*". Most of the papers were drawn from operational research, computer science, management science and healthcare. From the 100 papers, 50 were selected for further analysis. We favoured the more established journals/conferences, in an effort to avoid predatory journals [49]. Table 1 shows the journals of the selected methodological papers. Majority of the methodological papers are published in journals such as Annals of Operations Research (9), European Journal of Operational Research (8), Computers & Operations Research (7), Computers & Industrial Engineering (4) etc.

We approach the review in four stages. Stage 1 reviews the scientific literature. Stage 2 collates the recent methodological papers from various sources, favouring the more established journals/conferences, in an effort to avoid predatory journals [49]. In stage 3, we extract the information such as methodologies, benchmark dataset, results, findings, limitations, research opportunities from various sections of the papers such abstract, proposed methodology, experimental results, discussions and future work. In stage 4, we analyse the information and develop a perspective on the area, evaluate trends, identify the research challenges and discover the research opportunities. It is worth noting that the focus on emerging trends should not be interpreted in any way as a lack of respect towards other traditional NRP solution methodologies.

III. NURSE ROSTERING PROBLEM

A. PROBLEM DEFINITION

The nurse rostering problem can be defined as a staff scheduling problem that assigns a set of nurses with different skills to work shifts, subject to a variety of constraints [26], [64]. The aim is to improve operational efficiency of hospitals through effective utilisation of limited resources.

B. TERMINOLOGY

The following terms are used in this paper:

- *Planning Period*: The scheduling time interval, for example, 4 weeks, 3 months, 6 months, and 1 year.
- *Skill Category*: The skill, qualification, or the responsibility of nurses.
- *Shift Type*: A shift with a specific start and end time. For example, early shift (e.g. 7:00am-3:00pm), late shift (e.g. 3:00pm-10:00pm), and night shift (e.g. 10:00pm-7:00am).
- *Work Regulations*: The contract that a nurse has with a hospital. For example, nurse A works for 5 days a week while nurse B works for 6 days a week.
- *Hard Constraints*: The rules and conditions that must be satisfied for a solution to be valid.
- *Soft Constraints*: The conditions that can be violated but the solution will be penalised accordingly.
- *Coverage*: The number of nurses needed for every shift or skill category.
- *Time Restriction*: The time restrictions on a nurse's schedule. This is used to balance the workload among nurses to avoid overwork and/or unfair treatment.
- *Request*: The requests from nurses that will be incorporated into the schedule, if possible.

C. PROBLEM CONSTRAINTS

There are two types of constraints; hard and soft. Hard constraints must always be respected and, if violated the solution is not considered feasible. The violation of soft constraints is acceptable but the solution will be penalized according to a cost function [64], [83]. A solution with a lower cost function is considered superior to a solution with a higher cost function [7].

In NRP, coverage requirements (e.g. nurse demand per day, per skill, or per shift) are normally considered as hard constraints, while constraints that involve time requirements are usually regarded as soft constraints. There are three generic types soft constraints [73]:

- *Series*: These constraints limit the number of consecutive occurrences of a specific subject such as consecutive idle days, consecutive shifts of the same skill category, etc.
- *Successive Series*: The succession of two series will be restricted by these constraints. For example, days worked → days off.
- *Counters*: For a specific subject, the number of instances will be restricted by these constraints over a specified period. For example, working hours, days of work, number of off-day, etc.

Burke and Cutois [15] modelled many constraints found in staff scheduling problems, noting that these constraints vary from one hospital to another, and depends on the hospital's rules, regulations and waiting practices.

We present some commonly occurring constraints so that researchers can better understand the problem. The constraints can be either hard or soft depending on users' requirements:

- C1: Minimum/maximum workload.
- C2: Minimum/maximum consecutive shift/working days.
- C3: Minimum/maximum consecutive non-shift/working days (free days).
- C4: Minimum/maximum/exact number consecutive same working shift.
- C5: Minimum/maximum number of weekend shifts.
- C6: Minimum/maximum number of consecutive weekend shifts.
- C7: Minimum/maximum shift rotations.
- C8: Minimum/maximum free time between working shifts.
- C9: Minimum/maximum/exact number of working hours (e.g. 6-8 hours a day).
- C10: Nurses' skill level categories.
- C11: Nurses' requirements and preferences.
- C12: Nurses' historical record (e.g. previous shift record).
- C13: Nurses' shift types assignments.
- C14: Nurses' request for a day on/off.
- C15: Nurses' request for a shift on/off.
- C16: Vacations or public holidays (e.g. annual leave, medical leave).
- C17: Constraints among types/groups of nurses (e.g. nurses must work together or cannot work together).
- C18: Constraints among shifts (e.g. same shift cannot be assigned to the same nurse twice; one nurse cannot be assigned to two shifts at the same time).

D. PROBLEM VARIANTS

Two variants of NRP are identified in the scientific literature namely single-stage and multi-stage.

1) SINGLE-STAGE NRP

In the academic environment, it is common for approaches or problems to consider a single and restricted planning horizon where complete information is available. The single-stage NRP focuses on assigning nurses to shifts in a fixed planning horizon (one week or longer) without considering future or past information [23], [60]. The problem instances from the First International Nurse Rostering Competition (INRC-I) belong to this category.

TABLE 2. Solution methodologies for the nurse rostering problems (sorted according to year). H = Heuristic, MH-S = Meta-heuristic Singe Solution-based, MH-P = Meta-heuristic Population-based, HH = Hyper-heuristic, MO = Mathematical Optimization, MAH = Matheuristic and HA = Hybrid approach.

Year	Methodology	Category	Reference
2012	Adaptive Neighborhood Search (ANS)	MH-S	[52]
	Particle Swarm Optimization (PSO)	MH-P	[4]
	Constraint Programming (CP) based Column Generation (CG)	MO	[44]
	Variable Neighborhood Search (VNS) and Mixed Integer Linear Programming (MILP)	MAH	[34]
	Integer Programming (IP) and Hill Climbing Optimization (HCO)	MAH	[79]
2013	Multi-Assignment Procedures Algorithm (MAPA)	H	[31]
	Harmony Search Algorithm (HSA)	MH-P	[40]
	Harmony Search Algorithm	MH-P	[9]
	Artificial Bee Colony (ABC)	MH-P	[21]
	Cyber Swarm Algorithm (CSA)	MH-P	[85]
	Time Predefined Variable Depth Search (TP-VDS)	HA	[17]
2014	Particle Swarm Optimization	MH-P	[84]
	Single-point Selection Hyper-Heuristic (SPSHH)	HH	[71]
	Branch-and-Price (BAP)	MO	[15]
	Evolutionary Algorithm (EA) and IP	MAH	[45]
	Mixed Integer Programming (MIP) and Variable Neighborhood Descent (VND)	MAH	[69]
	Iterative Local Search (ILS) and CNR	HA	[57]
2015	Variable Neighborhood Search	MH-S	[77]
	Fuzzy Mathematical Programming (FMP)	MO	[46]
	Harmony Search Algorithm, Hill Climbing Optimization and Particle Swarm Optimization	HA	[7]
	Artificial Bee Colony and Hill Climbing Optimization	HA	[8]
2016	Two Stage Stochastic Programming (TSSP)	MO	[11]
	Mixed Integer Linear Programming	MO	[66]
	Minimum Cost Network Flow (MCNF)	MO	[72]
	Integer Programming and Variable Neighborhood Search	MAH	[62]
2017	Variable Neighborhood Search	MH-S	[87]
	Simulated Annealing (SA)	MH-S	[51]
	Harmony Search Algorithm	MH-P	[25]
	Directed Bee Colony Optimization (DBCO)	MH-P	[64]
	Integer Proramming and Constraint Programming	MO	[63]
	Integer Programming	MO	[60]
	Harmony Search Algorithm and Artificial Immune System (AIS)	HA	[48]
2018	Variable Neighborhood Search	MH-S	[83]
	Linear Integer Programming (LIP)	MO	[86]
	Online Stochastic Algorithm (OSA)	MO	[53]
	Variable Fixing Heuristic (VFH), Iterative Local Search and Mixed Integer Programming	MAH	[39]
2019	Iterative Local Search	MH-S	[56]
	Variable Fixing Heuristic and Mixed Integer Programming	MAH	[38]
	Branch-and-Price and Adaptive Large Neighborhood Search (ALNS)	MAH	[54]
2020	Simulated Annealing	MH-S	[24]
	Differential Evolution Algorithm (DEA)	MH-P	[43]
	Lexicographic Goal Programming (LGP)	MO	[13]
	First Order Linear Programming (FOLP) and CG	MO	[74]
	Column Generation based Diving Heuristic (CG-DH)	MAH	[3]
	Integer Programming and Fix-and-Optimize (FO) Matheuristic	MAH	[82]
	Fix-and-Relax (FaR), Fix-and-Optimize and Simulated Annealing	MAH	[78]
	Heuristic Algorithms (HAs), Bat Algorithm (BA) and Particle Swarm Optimization	HA	[28]
2021	Population-based Local Search (PB-LS)	MH-P	[2]
	Sequence-based Selection Hyper-Heuristic (SSHH)	HH	[50]
	Dynamic Programming (DP) and Variable Neighborhood Search	MAH	[1]

2) MULTI-STAGE NRP

In a real-world environment, roster quality within the current planning horizon is strongly influenced by the outcome from the previous planning horizon and future data (e.g. days off requests) [68]. As a result, optimising each planning horizon separately might induce a bad (quality) overall roster. The problem instances from the Second International Nurse Rostering Competition (INRC-II) were created based on a multi-stage formulation [23]. In this setting, a search method needs to solve a single stage of the problem corresponding to one week. In addition, history information (such as last worked shift of each nurse and total worked night shifts) carried over from the previous week must be taken into account. Furthermore, a roster is evaluated over multiple planning horizons.

IV. SOLUTION METHODOLOGIES (COMPUTATIONAL INTELLIGENCE APPROACHES)

In this section, we analyse the mechanisms and performance of the existing solution methodologies applied to the benchmark and real-world NRP. These methodologies can be classified into six categories namely heuristics, meta-heuristics, hyper-heuristics, mathematical optimization, matheuristics and hybrid approaches. Table 2 shows the solution methodologies for the nurse rostering problems

TABLE 3. **Categorization of the solution methodologies for the nurse rostering problems. H = Heuristic, MH-S = Meta-heuristic Singe Solution-based, MH-P = Meta-heuristic Population-based, HH = Hyper-heuristic, MO = Mathematical Optimization, MAH = Matheuristic and HA = Hybrid approach.**

Category	Methodology	Paper
H	MAPA	A heuristic algorithm based on multi-assignment procedures for nurse scheduling [31]
MH-S	ANS	Adaptive neighborhood search for nurse rostering. [52]
	VNS	Two-phase adaptive variable neighborhood approach for nurse rostering. [77]
		A simple randomized variable neighbourhood search for nurse rostering. [87]
		The nurse rerostering problem: Strategies for reconstructing disrupted schedules. [83]
	ILS	A neutrality-based iterated local search for shift scheduling optimization and interactive reoptimization. [56]
	SA	Simulated annealing approach to nurse rostering benchmark and real-world instances. [51]
		Solving the static inrc-ii nurse rostering problem by simulated annealing based on large neighborhoods. [24]
MH-P	PSO	Anaesthesiology nurse scheduling using particle swarm optimization. [4]
		A particle swarm optimization approach with refinement procedure for nurse rostering problem. [84]
	HSA	Harmony search with novel selection methods in memory consideration for nurse rostering problem. [9]
		A harmony search algorithm for nurse rostering problems. [40]
		Nurse scheduling with opposition-based parallel harmony search algorithm. [25]
	ABC	Applicability of artificial bee colony algorithm for nurse scheduling problems. [21]
	CSA	Cyber swarm algorithms for multi-objective nurse rostering problem. [85]
	DBCO	Directed bee colony optimization algorithm to solve the nurse rostering problem. [64]
	DEA	A scenario-based robust optimization with a pessimistic approach for nurse rostering problem. [43]
	PB-LS	A novel population-based local search for nurse rostering problem. [2]
HH	SPSHH	Modelling and evaluation issues in nurse rostering. [71]
	SSHH	A hyper-heuristic approach based upon a hidden markov model for the multi-stage nurse rostering problem. [50]
MO	CP-CG	A constraint programming based column generation approach to nurse rostering problems. [44]
	BAP	New approaches to nurse rostering benchmark instances. [15]
	FMP	Fuzzy mathematical modelling approach for the nurse scheduling problem: A case study. [46]
	TSSP	A two-stage stochastic programming approach for nurse scheduling problemin a real word hospital. [11]
	MILP	A direct milp approach based on state-expanded network flows andanticipation for multi-stage nurse rostering under uncertainty. [66]
	MILP	A direct milp approach based on state-expanded network flows andanticipation for multi-stage nurse rostering under uncertainty. [66]
	MCNF	Polynomially solvable personnel rostering problems. [72]
	IP	Integer programming model extensions for a multi-stage nurse rostering problem. [60]
	IP and CG	A hybrid integer and constraint programming approach to solve nurse rostering problems. [63]
	OSA	An online stochastic algorithm for a dynamic nurse scheduling problem. [53]
	LIP	Long term nurse scheduling via a decision support system based on LIP: A case study at the university hospital in cagliari. [86]
	FOLP and CG	First-order linear programming in a column generation-based heuristic approach to the nurse rostering problem. [74]
	LGP	Achieving compromise solutions in nurse rostering by using automatically estimated acceptance thresholds. [13]
MAH	IP and HCO	A systematic two phase approach for the nurse rostering problem. [79]
	VNS and MILP	A variable neighborhood search based matheuristic for nurse rostering problems. [34]
	MIP and VND	Integer programming techniques for the nurse rostering problem. [69]
	EA and IP	An evolutionary algorithm based on constraint set partitioning for nurse rostering problems. [45]
	IP and VNS	A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. [62]
	VFH, ILS and MIP	Three-phase method for nurse rostering. [39]
	VFH and MIP	Simple, yet fast and effective two-phase method for nurse rostering. [38]
	BAP and ALNS	A rotation-based branch-and-price approach for the nurse scheduling problem. [54]
	CG-DH	A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering. [3]
	IP and FO	An integer programming approach for the physician rostering problem. [82]
	FaR, FO and SA	A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem. [78]
	DP and VNS	A hybrid variable neighbourhood search and dynamic programming approach for the nurse rostering problem. [1]
HA	TP-VDS	A time predefined variable depth search for nurse rostering. [17]
	ILS and CNR	Nurse preference rostering using agents and iterated local search. [57]
	ABC and HCO	A hybrid artificial bee colony for a nurse rostering problem. [8]
	HSA, HCO and PSO	Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem. [7]
	HSA and AIS	Hybrid and cooperative strategies using harmony search and artificial immune systems for solving the nurse rostering problem. [48]
	HAs, BA and PSO	Developing two heuristic algorithms with metaheuristic algorithms to improve solutions of optimization problems with soft and hard constraints: An application to nurse rostering problems. [28]

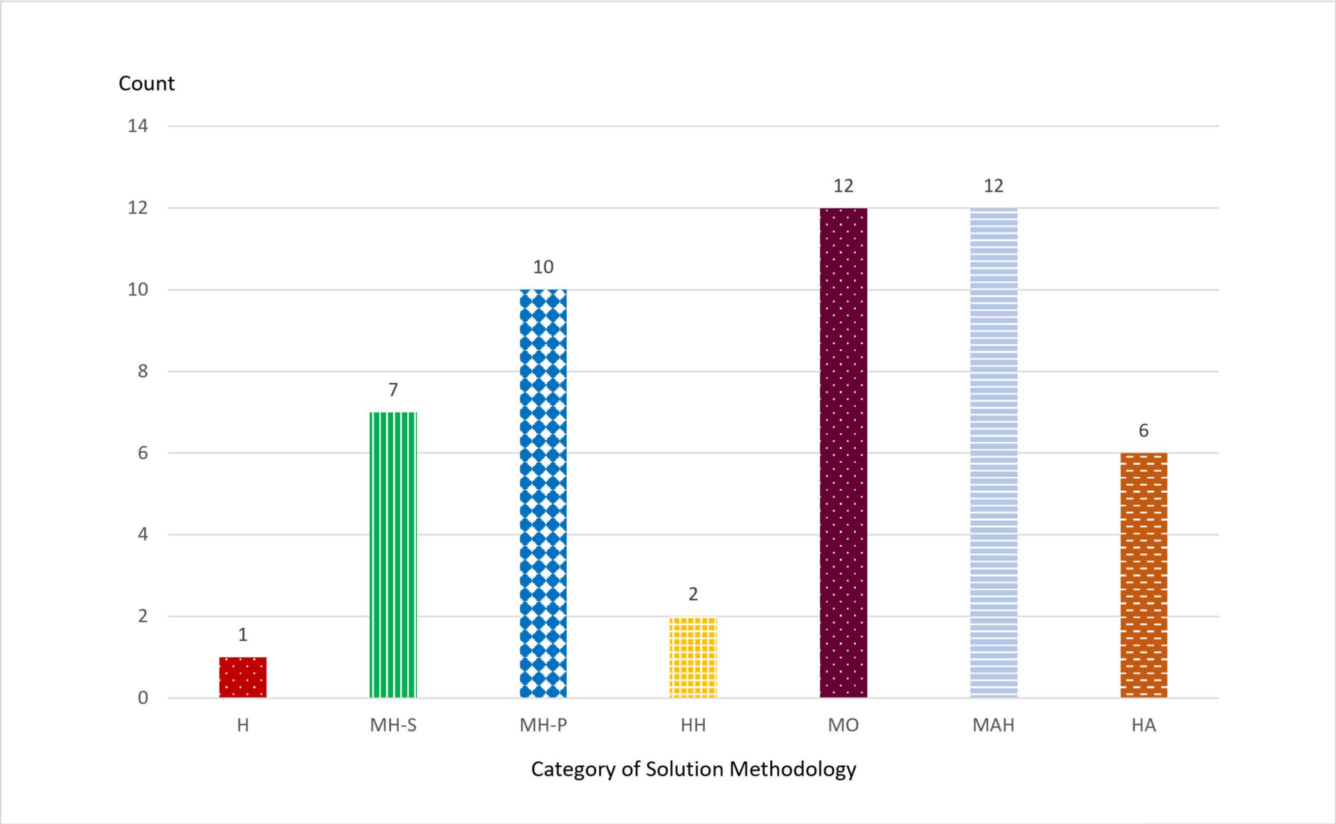

FIGURE 1. Categories of solution methodology. H = Heuristic, MH-S = Meta-heuristic Singe Solution-based, MH-P = Meta-heuristic Population-based, HH = Hyper-heuristic, MO = Mathematical Optimization, MAH = Matheuristic and HA = Hybrid approach.

TABLE 4. Categories of solution methodology per year. H = Heuristic, MH-S = Meta-heuristic Singe Solution-based, MH-P = Meta-heuristic Population-based, HH = Hyper-heuristic, MO = Mathematical Optimization, MAH = Matheuristic and HA = Hybrid approach.

	Category							
Year	H	MH-S	MH-P	HH	MO	MAH	HA	Total
2012	-	1	1	-	1	2	-	5
2013	1	-	4	-	-	-	1	6
2014	-	-	1	1	1	2	1	6
2015	-	1	-	-	1	-	2	4
2016	-	-	-	-	3	1	-	4
2017	-	2	2	-	2	-	1	7
2018	-	1	-	-	2	1	-	4
2019	-	1	-	-	-	2	-	3
2020	-	1	1	-	2	3	1	8
2021	-	-	1	1	-	1	-	3
Total	1	7	10	2	12	12	6	50

(sorted according to year). Table 3 shows the categorization of the solution methodologies for the nurse rostering problems. The most recent applications are Sequence-based Selection Hyper-heuristic [50], a hybrid of Dynamic Programming and Variable Neighbourhood Search [1] and Population-based Local Search [2].

Figure 1 shows the count of each solution methodology. There are 1 heuristic, 17 meta-heuristics (10 Population-based and 7 Single solution-based), 2 hyper-heuristics, 12 mathematical optimizations, 12 matheuristics and 6 hybrid approaches. Meta-heuristics are the most popular choice in addressing NRP. The number of population-based meta-heuristics utilized in NRP is more than the single solution-based variant. This is followed closely by matheuristic, mathematical optimisation and hybrid. There are a total of 24 mathematical based approaches (mathematical optimisations + matheuristics), implying a strong dominance of mathematical methodologies being applied to the NRP. Heuristics and hyper-heuristics are the least popular choices for researchers. Table 4 shows the breakdown of solution methodologies into categories by year. Matheuristics appear to be an emerging trend in recent years (2018 onwards).

A. HEURISTIC ALGORITHMS

Heuristics are rules of thumb based on domain knowledge. A heuristic methodology seeks good quality solutions, in a reasonable computation time, but comes with no guarantee of optimality [65].

1) MULTI-ASSIGNMENT PROBLEM-BASED ALGORITHM (MAPA)

Constantino *et al.* [31] proposed a deterministic heuristic algorithm called Multi-Assignment Problem-based Algorithm (MAPA) to address the NSPLib dataset. This algorithm is comprised of two phases; a constructive phase and an improvement phase. In the constructive phase, they first addressed the successive assignment problems before constructing a full schedule for each day in the planning period. In the improvement phase, they resolved the assignment problems to improve the schedule. The same schedule was obtained each time MAPA was applied to the same problem instances due to the deterministic nature of MAPA. Computational results showed that MAPA can produce more feasible solutions than the other algorithms in the literature. The only drawback of this algorithm was that it did not have the learning capability to improve, or adapt, its performance over time.

B. META-HEURISTICS

Unlike heuristic algorithms, meta-heuristics can be problem-independent and can be used to solve a range of problems within a reasonable amount of computational time. There are two types of meta-heuristic algorithms; single solution-based and population-based.

1) SINGLE SOLUTION-BASED ALGORITHMS

These approaches focus on maintaining, improving, and modifying a single candidate solution by exploring the surrounding area of the current solution utilising a neighbourhood(s) operator.

a: ADAPTIVE NEIGHBORHOOD SEARCH

Lu and Hao [52] applied Adaptive Neighborhood Search to the INRC-I dataset. This algorithm adaptively switches among three search strategies (intensive, intermediate, and diversification searches). In addition, two distinct joint neighborhood moves are used according to search history. The approach was able to match the previous best-known results for 39 instances and improve the best-known results for 12 instances. They claimed that their solver was robust and had reached a balance between diversification and intensification.

b: VARIABLE NEIGHBORHOOD SEARCH

Tassopoulos *et al.* [77] proposed a two-phase Variable Neighborhood Search algorithm to the INRC-I dataset. The algorithm consisted of two-phases. In phase 1, it assigned nurses to working days. In phase 2, it assigned nurses to specific shift types. Only a single candidate solution was used and different swap procedures were applied to improve solution quality. The algorithm searched in a different neighborhood of the search space after each swap procedure. The algorithm found new best solutions for two instances and achieved best-known results for 48 other instances. The algorithm was stochastic in nature where different results may occur each time it is execute.

Zeng *et al.* [87] proposed a randomized Variable Neighborhood Search algorithm for the INRC-I dataset. The algorithm was much simpler when compared to other, similar approaches. When stuck in a local optima, the algorithm used a cycle shift operator to diversify the search and randomly combined group operators to iteratively look for better solutions. The performance of the algorithm was comparable to other approaches but a deeper study was required to enhance the algorithm.

Wickert *et al.* [83] compared an exact method and Variable Neighborhood Descent (VND) in addressing the INRC-II dataset. The nurse re-rostering problem (NRRP) was also considered in this work. The problems were formulated as a general integer programming formulation and solved by using two commercial solvers (CPLEX and CBC). Results showed that CBC performed slightly worse than CPLEX. CBC could not find a feasible solution for instances containing 70 and 110 nurses. VND performed better than CPLEX as CPLEX could not find feasible solutions for larger instances (500 nurses) within the allowed time limit. VND produced near-optimum feasible solutions for all the instances, in reasonable computational time.

c: ITERATIVE LOCAL SEARCH

Meignan and Knust [56] proposed a neutrality-based Iterative Local Search (NILS) approach to the INRC-I dataset. When a local optimum was reached, the NILS approach, unlike normal ILS, implemented a plateau exploration stage. In the case of ILS, the first step consisted of a local search until it finds a local optimum. A plateau exploration process then begins with this local optimum solution to further improve the solution. Several neutral moves were used in the second step in finding better solutions. Computational results showed NILS outperformed other methodologies on small instances and achieved similar performance to the best-known methodologies on larger instances. NILS was effective in reoptimizing solutions quickly.

d: SIMULATED ANNEALING

Knust and Xie [51] tested two solution approaches in addressing the ORTEC dataset and other benchmark instances. The first approach was an exact method. A Mixed Integer Programming (MIP) model was formulated to compute the bounds for the instances. CPLEX and Gurobi were used as solvers. In the beginning, Gurobi was better than CPLEX but was outperformed by the end of the search. They showed that exact methods could find good solutions for the instances. The second approach utilised Simulated Annealing (SA). SA produced competitive results for the instances quickly in a time-restricted environment. An exact methodology was not practical due to the amount of time needed to generate a good quality solution. Both Gurobi and CPLEX required more than five hours to generate a better solution than the one produced by SA.

Ceschia *et al.* [24] utilised simulated annealing (SA), with large neighborhoods to tackle the INRC-II dataset. SA acted as a local search, returning an initial solution. Two neighborhood structures were proposed to improve the initial solution; MultiSwap (swapping the shifts of two nurses for several consecutive days) and MultiChange (changing the skill or shift assigned to a nurse for several consecutive days). The algorithm improved many best-known results for the 4-week horizon instances but produced worse results compared to [54] for the 8-week horizon instances.

2) POPULATION-BASED ALGORITHMS

These approaches improve and maintain multiple candidate solutions where each solution corresponds to a unique point in the problem search space.

a: PARTICLE SWARM OPTIMIZATION

A basic Particle Swarm Optimization algorithm works by generating a population (known as a swarm) of candidate solutions (known as particles). Each particle has a position and a velocity, and it moves around in the search space. The local best position, as well as the global best position, will guide these movements. When better locations are found, they will be used to guide the swarm's movements.

Altamirano *et al.* [4] proposed a Particle Swarm Optimization (PSO) algorithm to address a real-world problem for a French hospital. It was a simplified variant of PSO which only used the best-known position of the swarm to guide the particle's movements. The drawback for this approach is the difficulty in modifying the particles' position in the desired direction. The best and average results of the proposed algorithm improved the previous results of integer programming and constraint programming in shorter computational times.

Wu *et al.* [84] proposed a refined Particle Swarm Optimization (PSO) algorithm to address the ORTEC01 benchmark problem. The initial solution generated by a standard PSO might not be feasible. The shift patterns might be destroyed and cause infeasibility to the schedule if the position and velocity of each particle were forcibly updated. Therefore, they refined the algorithm by adding mutation functions to repair the schedule in order to satisfy all the hard constraints. The use of the mutation operator allowed the algorithm to explore previously unexplored solution regions and produce new solutions that were better than the current best solution. The best solution obtained by the proposed algorithm matched the optimal solutions reported in the benchmark problem.

b: CYBER SWARM ALGORITHM

Yin and Chiang [85] proposed a multiobjective Cyber Swarm Algorithm (CSA) to address the benchmark ZDT dataset. CSA is a variation of particle swarm optimization. The proposed algorithm implemented some major features from PSO namely scattered search, path relinking, and adaptive memory programming. There were four memory components in the algorithm; swarm memory, individual memory, global memory and reference memory. The algorithm outperformed several multiobjective evolutionary algorithms, with better diversity and convergence performances.

c: HARMONY SEARCH ALGORITHM

Harmony Search Algorithm (HSA) imitates the process of musical improvisation, in which a group of musicians plays the pitches of their instruments together to achieve a pleasing harmony based on audio-aesthetic criteria. HSA begins with a population of solutions stored in a harmony memory (HM). It iteratively improvises the new harmony using three operators: *memory consideration, random consideration,* and *pitch adjustment*. At each iteration, a new harmony is created and, if it is better, it replaces the worst solution in the HM. This procedure is repeated until convergence is achieved.

Awadallah *et al.* [9] proposed a Harmony Search Algorithm with different memory selection methods to address the INRC-I dataset. These selection methods include linear rank, proportional, tournament, and global-best. The tournament selection method gained the highest rate of convergence and achieved the best performance. The algorithm obtained best-published results for four instances and comparable results for the other instances.

Hadwan *et al.* [40] proposed a Harmony Search Algorithm to address a real-world NRP for the Universiti Kebangsaan Malaysia Medical Centre (UKMMC). They tested the efficiency of their algorithm with various parameter settings through a series of simulations. They found that higher values of harmony memory size (number of solutions stored inside the harmony memory) led to better quality solutions. The proposed algorithm outperformed a basic genetic algorithm in terms of best and average results. They claimed that it was due to well-balanced intensification and diversification characteristics.

A parallel Harmony Search Algorithm (HSA) is slightly different from a classical HSA. In the parallel version the whole HM (population) is equally divided into various sub-HMs and each uses their members to reach better solutions. The best remaining harmonies are combined again to form a whole HM for a re-grouping process before the search process is restarted.

A parallel HSA in [25] added a new feature that was based on opposition-based learning to tackle the NSPLib dataset. In the proposed parallel HSA, the re-grouping process was omitted. Instead, each sub-HM was developed individually and a new harmony vector was generated for each subgroup. This enabled them to drastically reduce the computational time. The proposed parallel HSA was superior to the parallel and classical HSA. The algorithm was able to find the best-known solutions in most of the problems, in short computational times.

d: BEE COLONY OPTIMIZATION ALGORITHM

Buyukozkan and Sarucan [21] were the first to propose an Artificial Bee Colony algorithm for a real-world NRP for a hospital in Karadeniz Technical University. Initial solutions were generated (randomly assigning shifts to nurses) for the bees (scout bees), then every bee's suitability value (punishment value) was calculated and the bees were sorted in ascending order. A neighborhood search was applied to the scout bees to define its follower bees. The suitability values of the follower bees were then calculated and compared to the scout bees. If a follower bee was better than its scout bee, the follower bee would replace the scout bee. These processes were iterated until a stopping condition was met. Finally, there was one scout bee left which was the best solution. The algorithm managed to return better schedules than the existing method and was faster.

Rajeswari *et al.* [64] proposed a Multiobjective Directed Bee Colony Optimization algorithm to address the INRC-I dataset. The algorithm was divided into two steps; forward and backward passes. During the forward pass, bees explored the area around their current solution, seeking out all possible solutions. The backward pass occurred when the bees returned to the hive and shared the values of their current solution's objective function. The search area of a bee was divided into multiple fragments. A local search algorithm called Modified Nelder-Mead Method was used to find the best solution in each fragment. The algorithm obtained 34 best solutions out of 69 instances.

e: EVOLUTION ALGORITHM

[43] proposed a robust scenario-based optimization method utilising a Differential Evolution (DE) algorithm to address the self-written five simple instances. The solutions (individuals) were represented as matrices. The locations and distances of these matrices were the input required by the algorithm's operators. The difference (distance) of the solutions was used by the algorithm to identify a direction in which to move. Three operators (Mutation, Crossover and Selection) were used in DE in generating the initial feasible solution, developing candidate solutions and selecting the best solution respectively. The algorithm obtained better results (cost function) than a conventional genetic algorithm in shorter computational times.

f: POPULATION-BASED LOCAL SEARCH

A population-based local search (PB-LS) was presented in [2], which addressed the INRC-I dataset. An integration of multi-neighborhood particle collision algorithm and adaptive randomized descent algorithm (MPCA-ARDA) was used as the local search technique in PB-LS. Initial solutions with different direction value (i.e., velocity) were generated by using a shaking neighborhood structure. At each iteration, the direction values were updated. The solutions were then compared with each other based on their direction values. Solutions with a higher direction value were prioritized for further improvement. The algorithm obtained 55 optimal solutions over 69 instances and outperformed all other methodologies previously reported.

C. HYPER-HEURISTICS

Hyper-heuristics are high-level approaches that generate and/or select low-level heuristics, often without domain at the higher level of the search [32]. A hyper-heuristic is defined as an automated methodology for generating or selecting heuristics to solve hard computational problems [18].

Smet *et al.* [71] applied a Single-point Selection Hyper-Heuristic approach to address their proposed dataset. The authors addressed modelling and evaluation issues in nurse rostering. A rich and generic model was built based on the real-world problem characteristics. A solution evaluation procedure, based on realistic quality measurement, was introduced. In addition, a novel benchmark dataset based on the model was proposed for researchers to test and compare their algorithms in a complex real-world setting. The hyper-heuristic approach outperformed adaptive large neighborhood search on 16 out of 18 instances.

A Sequence-based Selection Hyper-Heuristic (SSHH) was proposed in [50] to address the INRC-II dataset. SSHH used a hidden Markov model as a selection method, to learn successful transitions between low-level heuristics to find an effective sequence of heuristics. A heuristic was randomly chosen as a starting position. In each iteration, a heuristic was selected by using roulette wheel selection and added to the sequence until a complete sequence was achieved. The complete sequence was applied to the current solution to generate a new solution. If the quality of the new solution was better than the best solution, the sequence (new solution) was awarded by updating its two score matrices. The higher the score, the higher the chance of the sequence getting selected again. The iterative process stopped when a time limit was exceeded. The algorithm ranked first among general purpose hyper-heuristic and meta-heuristic approaches and was the best-ranked method for achieving feasibility across all problem instances.

D. MATHEMATICAL OPTIMIZATIONS

Mathematical optimization or mathematical programming (MP) involves the use of a mathematical model, which is optimised [70]. There are two categories of MP; Linear Programming (LP) and Nonlinear Programming (NLP). LP incorporates linear functions where all variables must be integer values. NLP incorporates general functions where the variables can be both discrete and continuous values.

1) INTEGER PROGRAMMING

Integer Programming is linear programming where both the constraints and objective functions must be linear. An IP starts with a linear program. Requirements are added where some or all variables take on integer values.

Zanda *et al.* [86] proposed a long-term nurse rostering approach based on Linear Integer Programming to address a real-world NRP for the surgery department of a university hospital in Cagliari, Italy. The problem was challenging due to sudden variation in nurse availability and lengthy time horizon. The proposed model was included in a Decision Support System allowing long intervals to be divided into several shorter intervals and enabling schedules to be constantly updated. The approach was implemented using a CPLEX solver and generated solutions that were not previously considered by the hospital. It could handle a sudden injection of new (unexpected) requirements and re-optimize a schedule quickly.

An extension to the basic Integer Programming (IP) model was proposed in [60], which addressed the INRC-II dataset. INRC-II is challenging because the previous week's roster affects the following week's roster. A basic IP model was used to solve the problem. It was able to obtain optimal solutions for each week provided enough time was given. However, the connection between weeks were ignored causing overall larger penalties. An extension to their model included six additional (soft) constraints to establish the connection between weeks. Three of them were based on existing constraints while the remaining three were newly created. A CPLEX solver was used to solve the IP model. The extended IP model was competitive with the competition results although no best solutions were found.

Rahimian *et al.* [63] proposed an integration of Integer Programming (IP) and Constraint Programming (CP) to address the INRC-I dataset. Both the IP and CP solvers are efficient in finding optimal solutions and feasible solutions respectively, for small to medium sized problems. However, their performance drops when addressing large scale or highly constrained problems. The authors hybridised IP and CP in a novel way to achieve better overall performance. An IP solver (Gurobi) was used to pre-solve the problem in order to obtain information that can be used to adjust the parameters of other components. A CP solver (IBM ILOG CP) was applied to solve different constraint satisfaction problem (CSP) models in order to get good quality solution and to identify difficult constraints. Then, the best solution provided within the CSP generation step was further improved by the IP solver in the remaining time. The algorithm was comparable to other state-of-the-art methodologies for a range of instances. It obtained best known solutions for 6 instances.

Romer and Mellouli [66] proposed a Mixed Integer Linear Programming (MILP) approach to address the INRC-II dataset. The problem was reformulated as a network-flow-based MILP. Coin CBC was used to solve the MILP. A directed acyclic network layer was associated with each nurse in which arcs correspond to days and shift off assignments; a flow from the source to the sink can be interpreted as a roster. The solver implemented a deterministic look-ahead method, that extended the planning period with an anticipation period and relaxed the integrality constraint by using artificially generated demand data. The demand data was generated based on the demand information of the previous and current weeks. The proposed algorithm was ranked first in the second competition by finding the best solution in 394 out of 600 runs.

2) BRANCH-AND-PRICE

Burke and Curtois [15] proposed a Branch-and-Price algorithm to address the INRC-I dataset. Branch and price was solved using column generation. Column generation consists of a master problem (formulated as a set covering problem) and a pricing problem (formulated as a resource constrained shortest path problem) which involves finding the optimal work schedule for an individual nurse. The master problem was solved using the Coin-OR linear programming (CLP) solver and the pricing problem was solved using dynamic programming. Constraint branching schemes (where branching is performed on shift assignments) were implemented as the branch and bound tree was too large to complete a full search. The proposed algorithm was ranked first or first-equal for all the instances except the hidden ones.

Strandmark *et al.* [74] proposed first-order linear programming in a column-generation-based heuristic approach to address the shift scheduling benchmark dataset. The algorithm was inspired by Branch-and-Price but it differed in two ways. Firstly, the proposed algorithm produced approximate solutions instead of solving the column generation exactly. Secondly, instead of exploring the whole tree, they employed a diving heuristic to find the best solution. The drawback of this algorithm was low accuracy since it performed an incomplete search. COIN-OR Linear Programming solver (CLP) was used to solve the master problem. The proposed algorithm produced the best-known results for some large instances, while other commercial solvers were unable to find any solutions for the same instances.

3) CONSTRAINT PROGRAMMING

A hybrid Constraint Programming based Column Generation (CP-CG) approach to address the ORTEC dataset was proposed in [44]. A feasible initial solution was generated and fed into the restricted master problem. Each column held a cost in the initial solution. The highest cost was used as the threshold in the CP pricing subproblem. An ILOG solver was used to solve the CP pricing subproblem. Candidate columns which had a cost below the threshold were generated by using Depth Bounded Discrepancy Search (DDS). Branch-and-Bound was applied to the generated columns to obtain an integer solution. A CPLEX solver was used in the branching strategy. The solution produced served as the upper bound of the master problem. DDS was then restarted to generate columns with tighter bounds. The process stopped when no more improvement could be made after a certain number of iterations. Based on the same time limit, the algorithm outperformed 4 other algorithms in six out of 12 instances.

4) LEXICOGRAPHIC GOAL PROGRAMMING

A multiobjective approach by using Lexicographic Goal Programming (LGP) to address a real-world problem for two Danish hospitals was proposed in [13]. There were two phases in the proposed methodology. Firstly, the acceptance thresholds were estimated by drawing upon instance-specific information. Secondly, before moving on to lower priority goals, lexicographic goal programming was used to resolve the targets in a prioritized sequence, checking the feasibility of the associated acceptance thresholds and applying them as hard constraints. A lexicographic framework had two main benefits; i) it prevented direct comparisons between goals of different measurement units and ii) it provided more control on searching the search space. Gurobi was used in each step of the LGP. The approach produced rosters with no or little deviations from the acceptance thresholds in a very short time.

5) FUZZY MATHEMATICAL APPROACH

Jafari *et al.* [46] proposed a Fuzzy Mathematical Programming approach to solve a real-world NRP for Milad Hospital in Iran. Four different types of fuzzy solution approaches were applied to form the proposed fuzzy mathematical model: min operator, weighted averaging operator, fuzzy-and operator and two-phase approach. The fuzzy mathematical model was solved using CPLEX. Results showed that the min operator performed the worst, while the weighted averaging operator performed the best.

6) STOCHASTIC PROGRAMMING

Unlike the other approaches (NRP as a deterministic problem), some authors have tackled the NRP as a stochastic problem.

Bagheri, Devin and Izanloo [11] proposed a Two-Stage Stochastic Programming approach to address a real-world NRP for the Department of Heart Surgery, Razavi Hospital. The problem was formulated as a mathematical model called Stochastic NRP (SNRP). Sample Average Approximation (SAA) was used in minimizing the overtime and regular assignment costs. Numerical experiments demonstrated the convergence of the statistical bounds, and moderate sample sizes that suit the model.

Legrain, Omer and Rosat [53] proposed an online stochastic algorithm that participated in the Second International Nurse Rostering Competition (INRC-II). The algorithm embedded a primal-dual algorithm within a sample average approximation (SAA). The algorithm was built upon existing nurse scheduling software that was based on the COIN-OR Linear Programming (CLP) and branch-and-cut-and-price (BCP) framework. The primal-dual algorithm generated candidate solutions for the current week. The candidate solutions were then evaluated by using SAA, with the best solution being retained. The proposed algorithm won the second place in the competition. Although it could not find the best schedules for most of the instances, it found high-quality feasible solutions for most of the instances.

7) MINUMUM COST NETWORK FLOW

Smet *et al.* [72] presented a minimum cost network flow formulation for several personnel rostering problems (PRP) (including NRP) to disprove the assumption that they are all NP-hard. New cases that can be solved in polynomial time after transforming them to minimum-cost network flow problems, were identified. The complexity of three academic nurse rostering benchmark datasets was discussed. They showed that a feasible solution can be guaranteed in polynomial time for two of the datasets (Nottingham dataset and INRC-I). The new solution approach could not be extended to the third dataset (KaHo dataset) due to a hard constraint forbidding overlapping assignments.

E. MATHEURISTICS

Matheuristic algorithms are the combination of meta-heuristic algorithms and mathematical optimization.

Valouxis *et al.* [79] proposed a two-phase approach that combined Integer Programming (IP) and Hill Climbing to address the INRC-I dataset. In the first phase, an IP formulation was applied to decide the workload for each nurse. Then, a randomized hill-climbing heuristic was applied to improve the current solution by attempting moves in various parts of the problem search space. In the second phase, another IP formulation was applied to assign nurses to specific shift types based on shift type demand for each day. GNU Linear Programming Kit (GLPK) was used to solve all the IP problems. The proposed algorithm won the INRC-I competition.

A hybrid of Variable Neighborhood Search (VNS) and Mixed Integer Linear Programming (MILP) to solve a real-world NRP for a ward of a private hospital in Turin was presented in [34]. The problem was formulated as an integer linear programming (ILP) model. A starting solution was generated using commercial MIP-solvers (CPLEX and XPRESS). Four extra constraints were added to the original ILP model. VNS was implemented to search for the best solution relying on the starting solution within a global time limit. The algorithm outperformed commercial MIP-solvers in terms of solution quality.

Santos *et al.* [69] proposed an integration of Mixed Integer Programming (MIP) and Variable Neighborhood Descent (VND) to solve the INRC-I dataset. The algorithm consists of two phases which are the constructive phase and the local search phase. In the constructive phase, a simple heuristic rule (greedy constructive algorithm) was used to create a feasible initial solution outside the MIP framework. In the local search phase, the MIP search using VND was applied utilising all the remaining time to find the local minimum by searching the search space through several large neighborhoods. During the process, MIP solvers (COIN-OR CBC and CPLEX) were also applied to produce better quality solutions in a limited time. The computational results show the algorithm was able to improve several best-known solutions even for the competition's hidden instances. They still need to rely on CPLEX if they want to develop a competitive MIP heuristic.

Huang *et al.* [45] proposed an Evolutionary Algorithm (EA) based on constraint set partitioning with Integer Programming (IP) to address a real-world problem called Chinese-NRP (CNRP). The proposed algorithm consisted of three phases. In phase one, they used constraint set partitioning to separate the constraints into hard and soft. In phase two, an IP approach (Lingo 11.0) was used to produce a high-quality initial solution to assist the

EA that followed in searching the promising regions of solution space. In phase three, a mutator operator was used to search for a better solution in the restricted feasible solution space. The algorithm outperformed four other methodologies on 201 instances in terms of average penalty values.

A hybrid approach combining Integer Programming (IP) and Variable Neighborhood Search (VNS) to tackle the 24 publicly test instances (Shift Scheduling Benchmark dataset) and ORTEC benchmark dataset was proposed in [62]. An initial solution was generated using a greedy heuristic. Variable neighborhood descent (VND) was then used to improve the initial solution, cycling through all the neighborhoods until no more improvement could be made. The best solution obtained from the VND was then passed to an IP solver (Gurobi) to fix the low-penalty parts, generating a different structured solution with better quality. The algorithm produced better solutions than the Gurobi IP solver and state-of-the-art methodologies for most of the instances in shorter computational times.

Haddadi [39] proposed a three-phase matheuristic to address the NSPLib dataset. Firstly, a variable fixing heuristic was used to reduce the original NRP instance size, producing a reduced problem called RNRP. Secondly, an iterative local search (ILS) was applied to solve the RNRP. Thirdly, a very small and sparse NRP was defined from the elite solution generated by the ILS and was solved by using a MIP-solver (CPLEX). The algorithm outperformed four recent methodologies and was much faster than a general-purpose commercial solver.

In [38] a two-phase method to address the NSPLib dataset was presented. In the first phase, a generic Variable Fixing Heuristic (VFH) was utilized to reduce the size of the original problem. Consequently, a reduced NRP (RNRP) was obtained. In the second phase, the RNRP was solved by using a commercial solver (CPLEX). The algorithm provided competitive results and was faster than other approaches in the literature.

Legrain *et al.* [54] proposed a rotation-based Branch-and-Price (BAP) procedure, embedding BAP within an Adaptive Large Neighborhood Search (ALNS), to address the INRC-II dataset. The rotation-based approach decreased the complexity of the pricing problems and reduced the number of generated feasible columns. A rolling horizon method was developed based on the BAP procedure to find the initial solution for ALNS. At each iteration, ALNS destroyed a part of the current solution and repaired it for improvement. A subset of variables was freed while others were fixed to their current value in the destruction phase. In the repair phase, a new solution was generated by finding a feasible solution from the freed variables by using BAP. COIN-OR Linear Programming (CLP) was used in the BAP approach. The algorithm achieved good results by beating several best-known solutions.

Turhan and Bilgen [78] proposed a hybrid of Mixed Integer Programming (MIP) based Heuristics and Simulated Annealing to address 24 publicly test instances (Shift Scheduling Benchmark dataset). The MIP-based heuristics were Fix-and-Relax (F&R) and Fix-and-Optimize (F&O). F&R was used as a starting point. The problem was decomposed into a set of smaller problems. Each of the smaller problems were iteratively solved using an IP solver (CPLEX). High quality initial solutions were generated and fed to the SA algorithm. The F&O was inserted into the SA when solutions could no longer be improved. This allowed the proposed algorithm to diversify the search space and generate better solutions. A final best solution was reported when a termination criterion was reached in SA. The proposed algorithm produced seven new best results.

In [3] a Column Generation based Diving Heuristic to address a real-world NRP for Sina Hospital in Iran was proposed. The algorithm consisted of two steps. In the first step, an optimal linear programming (LP) solution was found using column generation. In the second step, the fractional solution produced in the first step was converted into a feasible integer solution by using a greedy diving heuristic. Fractional assignments were fixed by the diving heuristic in a depth-first search without backtracking. The algorithm obtained near-optimal solutions and outperformed the general MIP-solver for larger instances in terms of solution quality. In addition, it could perform re-rostering efficiently.

Wickert, Smet and Vanden Berghe [82] proposed a matheuristic, which was an integration of Integer Programming and Fix-and-Optimize Matheuristic to address a real-world problem in Hospital de Clínicas de Porto Alegre (HCPA), Brazil. A feasible solution was generated using a Mixed Integer Programming (MIP) solver (CPLEX and Coin-OR CBC). The algorithm iteratively fixed the current values of a subset of variables, decomposing the problem into subproblems. The subproblems were then solved (considering hard and soft constraints) using the solvers. The MIP-solver could not find feasible solutions for larger instances. The proposed algorithm generated good results for smaller instances in acceptable computational times.

Abdelghany *et al.* [1] proposed a new hybrid of Variable Neighbourhood Search (VNS) and Dynamic Programming (DP) to tackle the INRC-I dataset. An initial feasible solution was generated by using a greedy constructive heuristic. A VNS was applied to improve the solution locally using different sets of neighborhood structures iteratively until no improvement could be made. Two perturbation mechanism (classical and destroy-and-recreate) were randomly selected for employment after a certain number of iterations to help the search escape from local optimums. Classical perturbation was used to diversify the search process. In destroy-and-recreate perturbation, schedules of random selected nurses were destroyed and recreated by using DP based heuristic algorithm. The algorithm achieved best known results for 44 out of the 60 instances.

F. HYBRID APPROACHES

Hybrid approaches provide a greater degree of integration in which, typically, a classical approach is embedded into a modern search tool or vice versa [19]. Some hybrid methodologies found in the scientific literature are tightly integrated while others are loosely integrated [37], [76]. We categorise a methodology as a hybrid if it integrates two or more methods in a single stage/phase. For example, a methodology that integrates an SA and a TS in one stage/phase is considered to be hybrid. In addition, a methodology is considered to be hybrid if it integrates two or more methodologies in separate stages/phases, if and only if they are from different categories, e.g. a methodology that employs a meta-heuristic in one stage and a hyper-heuristic in another is categorized as a hybrid.

Burke *et al.* [17] presented a Time Predefined Variable Depth Search (VDS) to address the ORTEC01 instance (ORTEC dataset). An initial solution was generated using a randomized greedy assignment method. VDS was employed to improve the initial solution. A single neighborhood swap was carried out by swapping two nurses. If the quality of the candidate solution was worse than the best solution's penalty, all the changes were reversed and a different swap was chosen, otherwise the candidate solution was accepted. During penalty recalculations (after swapping), days that required changing were flagged. The swaps were tested only if they involved at least one of these days. The algorithm outperformed the hybrid method proposed by [16] in 7 out of 10 instances.

In [57], the authors applied a hybrid of Iterative Local Search (ILS) and competitive nurse rostering (CNR) in addressing a real-world problem for the Mike O'Callaghan Federal Hospital's Air Force Medical Surgical Unit. CNR is an agent-based nurse rostering system [30]. A broker agent in the ILS (BA-ILS) was used to improve a feasible schedule. When no further improvement could be made, convergence to a local optimum was carried out by an auction control agent. This methodology treated each nurse as a distinct individual with distinct priorities, giving them an equal chance to strengthen their schedules while accounting for staffing slack to reduce schedule disruptions. The method performed well over the testing period in the hospital and consistently outperformed those solutions from the hospital scheduling practice and common integer programs. CNR-ILS was not a completely refined algorithm. It could be modified or expanded in a variety of ways, such as combining ILS auction mechanisms and applying shift swapping.

The authors of [7] proposed a Hybrid Harmony Search Algorithm (HHSA), a hybridization of HSA, Hill-Climbing Optimization (HCO) and Particle Swarm Optimization (PSO). The algorithm addressed the INRC-I dataset. Two modifications were applied to the conventional HSA. They replaced the random selection scheme with the global-best selection scheme, drawn from PSO, to improve the convergence rate of the algorithm. In addition, HCO was employed to enhance its exploitation capability. In HHSA, a solution was generated using memory consideration and random consideration (from HSA). The roster was then exploited locally using HCO. HM was then updated by replacing the worst roster with a better roster. The algorithm achieved 38 best results out of 69 instances.

Awadallah *et al.* [8] proposed a hybrid Artificial Bee Colony (HABC) algorithm, which is an integration of Artificial Bee Colony (ABC) and Hill-Climbing Optimization (HCO), addressing the INRC-I dataset. In ABC, there are three group of bees; employed, onlookers and scouts. Employed bees share the solutions that it had searched before. Onlooker bees choose a desired solution among the solutions shared by the employed bees. Scout bees carry out random search in order to discover new solutions. In the proposed methodology, the operator of the employed bee was replaced with HCO to enhance its exploitation capability. Initially, a set of feasible solutions were saved in a Food Source Memory (FSM). The employed bees applied HCO to the solutions in the FSM. The exploited solutions were shared with onlooker bees. The onlooker bees decided which employed bee to follow and exploited its corresponding solution randomly with the aid of some neighbourhood structures. The solution with better quality replaced the old one in FSM. The scout bees searched for possible solutions that were abandoned by the employed bee. The quality of the best solution was memorized. The process was repeated until a stopping condition was met. The algorithm achieved 37 new best results out of 69 instances. It obtained comparable results for the remaining instances.

Jin *et al.* [48] proposed two hybrid approaches, combining Harmony Search Algorithm (HSA) and Artificial Immune Systems (AIS) when addressing the INRC-I dataset. The first approach hybridized HSA with AIS (HHSAIS). When a new harmony (solution) was not better than the worst harmony, it was assumed that the current harmony memory (population) consisted of similar solutions. The harmony memory was renewed with the help of AIS, to ease the algorithm in searching the solution space. The second approach was cooperating HSA with AIS (CHSAIS). The population of each algorithm was maintained separately. Good solutions generated by HSA and AIS were exchanged to update their existing harmony memory (population). The drawback was they required a longer computational time than a normal HSA. CHSAIS was better than HHSAIS as it achieved better average results and matched the best average results acquired by other methods.

Chen and Zeng [28] proposed a hybrid of decision trees, greedy search, Bat Algorithm (BA) and Particle Swarm Optimization (PSO) to address a real-world problem for the Emergency Diagnostic Radiology Unit in Taiwan. They tested BA and PSO separately, with a decision tree and a greedy search. A decision tree was used to generate a set of initial solutions. A greedy search was applied at the end of each generation to improve the quality of the solutions. With the presence of both heuristics, BA and PSO could generate better quality solutions, with reduced computational time.

V. ADVANTAGES AND DRAWBACKS OF THE SOLUTION METHODOLOGIES (CHALLENGES)

Heuristic approaches provide an alternative to exhaustive search for large combinatorial optimisation problems. They cannot guarantee optimality, but can often produce acceptable solutions is reasonable computational times. These approaches are often problem dependent, relying heavily on domain knowledge and requires substantial development to be utilised on another problem domain.

Meta-heuristics often require parameter tuning in order to perform at their best. They can return good quality solutions, especially for large optimisation problems, in reasonable computation times. However, they do not guarantee an optimal solution even for small problems. Population-based meta-heuristics have been well received and known for their ability in exploring search spaces as they deal with multiple candidate solutions, increasing their chances of finding better quality solutions. The limitations of population-based meta-heuristics are long computational times and slow convergence [2], [12]. They are often hybridised with other methods to improve their exploitation capabilities [7], [8]. Single solution-based meta-heuristics are relatively faster than population-based variants as they explore the vicinity of the current solution and deal with a single candidate solution. Single solution-based meta-heuristics are mediocre in terms of exploration capability. Therefore, improvements are often proposed in this regard.

Hyper-heuristics were proposed to overcome the parameter tuning requirement of meta-heuristics. Rather than solving a problem directly, hyper-heuristics have a high level search strategy, which does not require domain knowledge. This knowledge is delegated to low level heuristics which make it relatively simple to apply the same algorithm to a different domain by simply switching these low level heuristics. Solutions generated are often acceptable and even comparable to the state-of-the-art methods in some cases. The challenge in hyper-heuristic approaches is balancing trade-off between greater level of information exchange and maintaining a clear split between problem domain and high-level solution methodology [35]. A greater level of information exchange has a direct impact on performance while a clear split between problem domain and high-level solution methodology ensures applicability (generality) of the methodology across problem domains.

Mathematical optimisation (exact methods) is guaranteed to find an optimal solution, assuming that there is enough time and memory for the algorithm to complete executing. Given this constraint, exact methods are often only suitable for solving small and medium sized problems to optimality, especially for NP-Hard problems [20], [44]. The implementation of mathematical optimisation is made simpler with the use of off-the-shelf mathematical solvers. The user is only required to define the mathematical model and key features (objective functions, decision variables and constraints) of the problem and feed them into the solver. The limitation of this approach is that parameters are assumed to be constant when they are neither constant nor known in real-life situations. Table 5 and Figure 2 summarise the mathematical solvers used in solving the NRP. CPLEX (13) seems to be the most popular solver followed by Gurobi (4), CLP (4), CBC (4) and ILOG (2). The rest of the solvers were used once each.

TABLE 5. Mathematical solvers used in solving the NRP; A = CPLEX, B = Gurobi, C = ILOG CP optimizer, D = COIN-OR LP solver (CLP), E = COIN-OR Branch and Cut (CBC), F = Lingo, G = GNU Linear programming Kit, H = XPRESS solver.

	Mathematical Solver							
Reference	A	B	C	D	E	F	G	H
[44]	✓	-	✓	-	-	-	-	-
[79]	-	-	-	-	-	-	✓	-
[34]	✓	-	-	-	-	-	-	✓
[15]	-	-	-	✓	-	-	-	-
[69]	✓	-	-	-	✓	-	-	-
[45]	-	-	-	-	-	✓	-	-
[46]	✓	-	-	-	-	-	-	-
[66]	-	-	-	-	✓	-	-	-
[62]	-	✓	-	-	-	-	-	-
[63]	-	✓	✓	-	-	-	-	-
[60]	✓	-	-	-	-	-	-	-
[51]	✓	✓	-	-	-	-	-	-
[53]	-	-	-	✓	-	-	-	-
[86]	✓	-	-	-	-	-	-	-
[39]	✓	-	-	-	-	-	-	-
[83]	✓	-	-	-	✓	-	-	-
[38]	✓	-	-	-	-	-	-	-
[54]	-	-	-	✓	-	-	-	-
[74]	-	-	-	✓	-	-	-	-
[13]	-	✓	-	-	-	-	-	-
[3]	✓	-	-	-	-	-	-	-
[82]	✓	-	-	-	✓	-	-	-
[78]	✓	-	-	-	-	-	-	-
Total	13	4	2	4	4	1	1	1

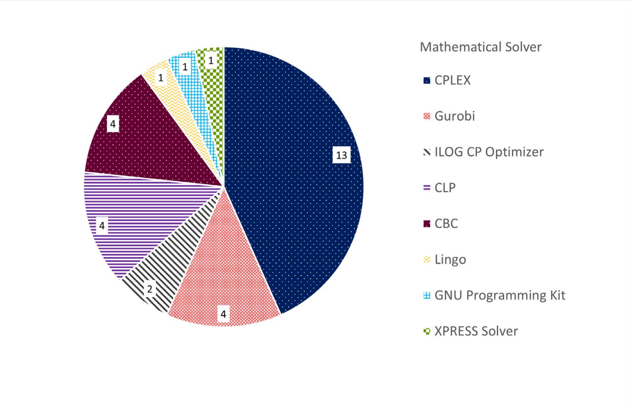

FIGURE 2. The count of mathematical solvers used in solving the NRP.

Matheuristics (integration of mathematical optimisation and meta-heuristic) can be considered a special case of hybridsation, which are getting popular among NRP researchers. This hybridisation exploits the strength of

TABLE 6. Solution methodologies for the NRP benchmark datasets; H = Heuristic, MH-S = Meta-heuristic Singe Solution-based, MH-P = Meta-heuristic Population-based, HH = Hyper-heuristic, MO = Mathematical Optimization, MAH = Matheuristic and HA = Hybrid approach.

Benchmark NRP (Count)	Year	Methodology	Category	Reference
INRC-I (16)	2012	ANS	MH-S	[52]
		IP and HCO	MAH	[79]
	2013	HSA	MH-P	[9]
	2014	MIP and VND	MAH	[69]
		BAP	MO	[15]
	2015	ABC and HCO	HA	[8]
		HSA, HCO and PSO	HA	[7]
		VNS	MH-S	[77]
	2016	MCNF	MO	[72]
	2017	DBCO	MH-P	[64]
		VNS	MH-S	[87]
		IP and CP	MO	[63]
		HSA and AIS	HA	[48]
	2019	ILS	MH-S	[56]
	2021	PB-LS	MH-P	[2]
		DP and VNS	MAH	[1]
INRC-II (7)	2016	MILP	MO	[66]
	2017	IP	MO	[60]
	2018	OSA	MO	[53]
		VNS	MH-S	[83]
	2019	BAP and ALNS	MAH	[54]
	2020	SA	MH-S	[24]
	2021	SSHH	HH	[50]
ORTEC (5)	2012	CP-CG	MO	[44]
	2013	TP-VDS	HA	[17]
	2014	PSO	MH-P	[84]
	2016	IP and VNS	MAH	[62]
	2017	SA	MH-S	[51]
NSPLib (4)	2013	MAPA	H	[31]
	2017	HSA	MH-P	[25]
	2018	VFH, ILS and MIP	MAH	[39]
	2019	VFH and MIP	MAH	[38]
Shift Scheduling (3)	2016	IP and VNS	MAH	[62]
	2020	FOLP and CG	MO	[74]
		FaR, FO and SA	MAH	[78]
Others (5)	2013	CSA	MH-P	[85]
	2014	EA and IP	MAH	[45]
		SPSHH	HH	[71]
	2016	MCNF	MO	[72]
	2020	DEA	MH-P	[43]

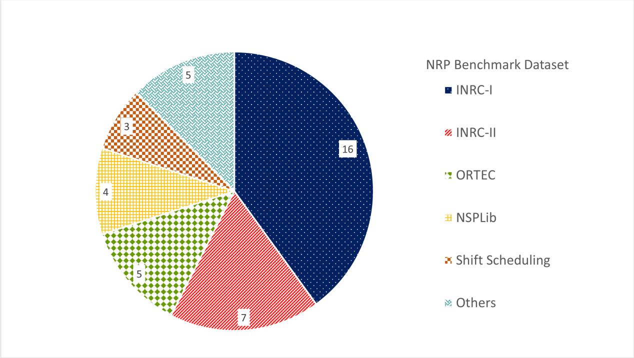

FIGURE 3. The count of solution methodologies for each NRP benchmark dataset.

mathematical optimisation (convergence) and that of meta-heuristic (exploration) with the aim of generating high quality solutions in short computation times.

Hybrid is another popular approach among researchers. As with matheuristics, the aim is to achieve a synergy between various methods, combining strengths and overcoming weaknesses of each method. However, hybrid methods (especially tight integration hybrids) are hard to implement as it is not easy to achieve the desired synergy.

VI. NRP BENCHMARK DATASETS AND REAL-WORLD NRP

In this section, we provide an overview of the NRP benchmark datasets and the real-world NRP. Table 6 presents the NRP benchmark datasets and the respective solution methodologies. Figure 3 shows the count of solution methodologies for each NRP benchmark dataset. We can see that INRC-I is the most popular among all the benchmark datasets, followed by the INRC-II, ORTEC, NSPLib, Shift Scheduling and others. Table 7 shows the solution methodologies for the real-world NRP.

A. FIRST INTERNATIONAL NURSE ROSTERING COMPETITION (INRC-I) DATASET

The INRC-I was organised in 2010 [41]. The dataset used in this competition can be downloaded from the website.[1] It

[1] https://nrpcompetition.kuleuven-kulak.be/

TABLE 7. Solution methodologies for real-world NRP; H = Heuristic, MH-S = Meta-heuristic Singe Solution-based, MH-P = Meta-heuristic Population-based, HH = Hyper-heuristic, MO = Mathematical Optimization, MAH = Matheuristic and HA = Hybrid approach.

Year	Real-world NRP	Country	Methodology	Category	Reference
2012	Turin Private Hospital	Italy	VNS and MILP	MAH	[34]
	French Public Hospital	French	PSO	MH-P	[4]
2013	Karadeniz Technical University	Turkey	ABC	MH-P	[21]
	UKMMC Hospital	Malaysia	HSA	MH-P	[9]
2014	Mike O'Callaghan Federal Hospital	U.S.A	ILS and CNR	HA	[57]
2015	Milad Hospital	Iran	FMP	MO	[46]
2016	Razavi Hospital	Iran	TSSP	MO	[11]
2018	Cagliari University Hospital	Italy	LIP	MO	[86]
2020	Research Hospital	Taiwan	HAs, BA and PSO	HA	[28]
	Hospital de Clínicas de Porto Alegre	Brazil	IP and FO	MAH	[82]
	Sina Hospital	Iran	CG-DH	MAH	[3]
	Danish Hospital	Denmark	LGP	MO	[13]

TABLE 8. Characteristics of the INRC-I instances.

Instances	Shifts	Nurses	Contracts	Skills	Patterns	Weekends
sprint (early,late,hidden)	3 to 4	10	3 to 4	1	0 to 8	2 to 3
medium (early,late,hidden)	4 to 5	30	3 to 4	1 to 2	0 to 9	2
long (early,late,hidden)	5	50	3 to 4	2	3 to 9	2 to 3

consists of three tracks namely Sprint, Medium and Long. Each track consists of three sets of instances which are *Early*, *Late* and *Hidden* instances. The characteristics of the INRC-I instances are given in Table 8. The winner of the competition [79] proposed a matheuristic approach, integrating Integer Programming and Hill-Climbing. Meta-heuristics seem to be a common approach for this dataset where the population based approach outnumbered the single solution-based variant. The state-of-the-art methods for this dataset are a hybrid of Integer Programming and Hill Climbing [79], Adaptive Neighborhood Search (ANS) [52], Randomized Variable Neighborhood Search (RVNS) [87], a hybrid of Dynamic Programming and Variable Neighbourhood Search [1] and Population-based Local Search (PB-LS) [2].

B. SECOND INTERNATIONAL NURSE ROSTERING COMPETITION (INRC-II) DATASET

The INRC-II competition was run in 2014 [23]. The dataset can be obtained from the link.[2] While INRC-I focuses on assigning shifts to nurses in a fixed planning horizon with many hard and soft constraints, INRC-II deals with multi-objective formulation problem with a smaller set of constraints. In addition, the planning horizon (4 or 8 weeks) used in the INRC-II is longer than that used in INRC-I. Table 9 shows the characteristics of the INRC-II instances. There are a total of 28 late instances and a total of 60 hidden instances. The winner of the competition implemented a Mixed Integer Linear Programming [66]. The first runner up applied a mathematical optimization approach called online stochastic algorithm [53]. The third placing was won by a Sequence-based Selection Hyper-Heuristic approach [50]. The state-of-the-art methods for this dataset are Mixed Integer Linear Programming (MILP) [66], Online Stochastic Algorithm (OSA) [53], Simulated Annealing (SA) [24] and Sequence-based Selection Hyper-Heuristic (SSHH) [50].

TABLE 9. Characteristics of the INRC-II instances.

Instance	Nurses	Weeks	Skills	Shifts	Contracts
Late	30,40,50,60,80,100,120	4,8	4	4	3,4
Hidden	35,70,110	4,8	4	4	3,4

TABLE 10. Characteristics of the ORTEC instances.

Instances	Shift Types	Nurses	Period (days)	Skill Levels
ORTEC01-04	4	16	35	1
ORTECJan-Dec	4	16	35	1

C. OAK RIDGE TECHNICAL ENTERPRISES CORP (ORTEC) DATASET

This dataset was provided by ORTEC, a supplier of planning, optimisation and scheduling software products. It can be downloaded from their website.[3] Table 10 shows the characteristics of the ORTEC instances. The state-of-the-art methods for this dataset are Constraint Programming based Column Generation [44] and a hybrid of Integer Programming and Variable Neighborhood Search [62]. Interestingly, [62] showed that their hybrid algorithm was superior than the one proposed by [20].

D. NURSE SCHEDULING PROBLEM LIBRARY DATASET (NSPLib)

NSPLib was proposed based on different complexity indicators for the NRP [80]. This dataset is available on the Operation Research and Scheduling research group website.[4] There are two sets of instances, diverse and realistic, which have 29,160 and 1,920 instances respectively. There are

[2] http://mobiz.vives.be/inrc2/

[3] http://www.schedulingbenchmarks.org/index.html

[4] https://www.projectmanagement.ugent.be/nsp.php

TABLE 11. Characteristics of NSPLib.

	Diverse Set	Realistic Set
Nurses	25,50,75 or 100	30 or 60
Shifts	4 (including free shift)	4 (including free shift)
Days	7	28

TABLE 12. Characteristics of shift scheduling instances.

Instance	Weeks	Nurses	Shift Types
1 to 24	2 to 52	8 to 150	1 to 18

8 different cases with different coverage constraints and preferences for each instance. In total, there are 248,640 (8 × (29,160+1,920)) problem instances. Table 11 shows the characteristics of the dataset.

The state-of-the-art methods for this dataset are Multi-Assignment Procedures Algorithm (MAPA) [31] and a hybrid of Variable Fixing Heuristic, Iterated Local Search and Mixed Integer Programming [39]. Other approaches that utilized this dataset are Harmony Search Algorithm [25] and a hybrid of Variable Fixing Heuristic and Mixed Integer Programming [38].

E. SHIFT SCHEDULING (NURSE ROSTERING) DATASET

This dataset was introduced by [33] and can be downloaded from the public benchmark nurse rostering instances website.[5] It consists of 24 instances with different planning periods (weeks), number of nurses and shift types as presented in Table 12. The state-of-the-art methods for this dataset are a hybrid of Integer Programming and Variable Neighborhood Search [62], First Order Linear Programming and Column Generation [74] and a hybrid of Fix-and-Relax, Fix-and-Optimize and Simulated Annealing [78].

F. OTHER DATASETS

The other NRP datasets that can be found in the scientific literature are listed below. A Minimum Cost Network Flow formulation [72] was proposed to address the KaHo and Nottingham Benchmark datasets. [85] proposed a Cyber Swarm Algorithm (population-based meta-heuristic) to address the ZDT benchmark dataset. [45] applied a matheuristic approach by combining Integer Programming with an Evolutionary Algorithm to solve the Chinese NRP dataset. [71] and [43] proposed Single-point Selection Hyper-Heuristic and Differential Evolution Algorithm respectively, to address their own datasets.

G. REAL-WORLD NRP

Addressing real-world NRP is often harder and more challenging than solving benchmark datasets due to uncertainties (sudden changes) in nurses' personal preferences and unexpected events (medical leave, natural disaster, holidays, emergency leave) as hospitals operate 24 hours a day, 7 days a week [11], [29], [46]. As each hospital has its own rules and regulations, the constraints involved in a real-world NRP may vary from each other [15]. Table 7 shows the related work on the real-world NRP.

VII. FUTURE DIRECTIONS (RESEARCH OPPORTUNITIES)

As given in Table 6, matheuristics were utilized in INRC-I, INRC-II, ORTEC, NSPLib and Shift Scheduling datasets. In fact, matheuristic appeared as one of the state-of-the-art methods for each dataset. Matheuristics were competitive if not better than other approaches. It is a promising approach for NRP.

As shown in Table 4, there are only two application of hyper-heuristics to the NRP. This suggests that there is scope for more study on hyper-heuristic in NRP [2].

It could be interesting to test approaches shown to be effective on other optimization areas and apply them to NRP. Relatively new meta-heuristics such as grey wolf optimizer (GWO) [59] and elitist self-adaptive step-size search (ESASS) [10], methods based on Monte Carlo simulations [36], algorithms rooted in game theory and bi-level optimization, cognitively motivated methods, etc. are among the viable alternatives to more traditional techniques.

There are many benchmark datasets publicly available for researchers to compare the performance of their algorithms objectively. Established datasets such as INRC-I provide researchers with many references on existing work while new datasets such as INRC-II present researchers with plenty of opportunities.

In this survey, we note that there is a lack of work on re-optimization in NRP [3], [56], [86]. For benchmark NRP, requirements are fixed and schedule re-optimization is not required. In a real-world situation, uncertainties (emergency leaves, staff dismissal, natural disasters, and accidents) exist that require nurse roster to be constantly changed and worse, in a sudden manner. Therefore, schedule re-optimization is an interesting topic and warrants more research.

Despite the increasing number of publications, research in this domain is relatively low compared to other combinatorial optimisation problems, such as the vehicle routing problem (VRP) which is the most widely researched problem in the domain of operational research [55].

In addition, there exists a gap between the benchmark and real-world NRP in terms of requirements and implementations. Perhaps, more benchmark datasets that mimic real-world NRP can be introduced in future, in order to close the gap. The benchmark datasets should cover real-world constraints. Case studies on real-world NRP has a role to play in this regard. As real-world NRP varies greatly, research on real-world NRP is highly encouraged.

Some authors have expressed their hope to test their proposed algorithms on real-world environment; Particle Swarm Optimization [4], a hybrid of Integer Programming and Constraint Programming [63], a hybrid of Harmony Search Algorithm and Artificial Immune System [48],

[5] http://www.schedulingbenchmarks.org/nrp/

Differential Evolution Algorithm [43] and a hybrid of Fix-and-Relax, Fix-and-Optimize and Simulated Annealing [78].

Several researchers hope to hybridize their proposed methodology by replacing certain components or integrating it with other methods. Some want to integrate a local search into their proposed methodology; Harmony Search Algorithm [9], [25]. [79] would like to implement a tabu search mechanism or simulated annealing in their proposed algorithm; a hybrid of Integer Programming and Hill Climbing. Some hope to hybridize their proposed methodology with meta-heuristics; Constraint Programming based Column Generation [44], Multi-Assignment Procedures Algorithm [31], a hybrid of Integer Programming and Variable Neighborhood Search [62] and First Order Linear Programming and Column Generation [74]. [34] hopes to combine their mathematical solver implementation with other approaches. The others seeking to hybridize their algorithms are; Artificial Bee Colony [21], Branch-and-Price [15], a hybrid of Iterative Local Search and CNR [57], and Simulated Annealing [24].

Some researchers hope to test their proposed methods/formulations on other approaches. [17] was looking forward to utilize variable depth search on other population-based approaches (scatter search or memetic algorithm).

Various researchers have provided suggestions to improve their proposed algorithms. [1] hopes to improve the exploration capability of their methodology by suggesting another destroy-and-recreate vertical mechanism for individual days instead of individual nurses. [39] would like to improve the ILS (second phase) by diversifying the search instead of searching around the best solution. [53] suggests refining the primal-dual algorithm and enhance it with non-linear updates.

Others hope to further extend their proposed model to include more constraints [11], [54], [60], [84] and resource types such as operating rooms and surgical nurses [3]. [28] proposes developing a universal algorithm that can be utilized in more hospitals. [86] planned to tackle the NRP involving several departments that share the same resources.

Some researchers are looking at parameter tuning or conducting parameter sensitivity on their proposed methodologies; Artificial Bee Colony [21], a hybrid of Artificial Bee Colony and Hill Climbing Optimization [8] and Directed Bee Colony Optimization [64].

Some authors plan to apply their proposed methodologies on other benchmark NRP datasets or instances; Adaptive Neighborhood Search [52], Multi-Assignment Procedures Algorithm [31], a hybrid of Evolutionary Algorithm and Integer Programming [45], a hybrid of Artificial Bee Colony and Hill Climbing Optimization [8], Variable Neighborhood Search [77], Directed Bee Colony Optimization [64], Harmony Search Algorithm [25] and Lexicographic Goal Programming [13].

A number of researchers intend to employ their proposed methodologies (Variable Neighborhood Search [87] and Harmony Search Algorithm [25]) on other combinatorial optimization problems. Rajeswari *et al.* and Rahimian *et al.* plan to apply Directed Bee Colony Optimization [64] and a hybrid of Integer Programming and Constraint Programming [63] respectively, to other scheduling problems. Turhan and Bilgen plan to utilize a hybrid of Fix-and-Relax, Fix-and-Optimize and Simulated Annealing [78] on high school timetabling problems.

VIII. CONCLUSION

The field of the NRP is developing rapidly and many new papers are published each year. The existing survey papers, particularly on solution methodologies are becoming obsolete. For this reason, we have reviewed the recent articles on NRP between 2012 and 2021, with the aim to extend the taxonomy of the NRP in order to keep it up-to-date. We present the perspectives: a brief description of the mechanisms and performance of the solution methodologies (CI approaches) for both the benchmark and real-world NRP, a categorisation of the solution methodologies, the benchmark repositories and the state-of-the-art methods. Notably, we suggest the emerging trends. In addition, we discuss the advantages and drawbacks of the solution methodologies (challenges) as well as the future directions (research opportunities) in NRP. Among the findings are; meta-heuristics are the most popular choices in addressing NRP, matheuristics appear to be an emerging trend in recent years (2018 onwards) and the INRC-I dataset is the most popular benchmark currently in use by researchers to test their algorithms. We believe that this survey paper will be valuable to OR and CI communities, especially young researchers in planning their research in this domain.

REFERENCES

[1] M. Abdelghany, A. B. Eltawil, Z. Yahia, and K. Nakata, "A hybrid variable neighbourhood search and dynamic programming approach for the nurse rostering problem," *J. Ind. Manage. Optim.*, vol. 17, no. 4, p. 2051, 2021.

[2] A. Abuhamdah, W. Boulila, G. M. Jaradat, A. M. Quteishat, M. K. Alsmadi, and I. A. Almarashdeh, "A novel population-based local search for nurse rostering problem," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, p. 471, Feb. 2021.

[3] B. Akbarzadeh, G. Moslehi, M. Reisi-Nafchi, and B. Maenhout, "A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering," *J. Scheduling*, vol. 23, no. 2, pp. 265–288, Apr. 2020.

[4] L. Altamirano, M. C. Riff, I. Araya, I. Araya, and L. Trilling, "Anesthesiology nurse scheduling using particle swarm optimization," *Int. J. Comput. Intell. Syst.*, vol. 5, no. 1, pp. 111–125, 2012.

[5] A. Arram and M. Ayob, "A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems," *Comput. Ind. Eng.*, vol. 133, pp. 267–274, Jul. 2019.

[6] A. Arram, M. Ayob, G. Kendall, and A. Sulaiman, "Bird mating optimizer for combinatorial optimization problems," *IEEE Access*, vol. 8, pp. 96845–96858, 2020.

[7] M. A. Awadallah, M. A. Al-Betar, A. T. Khader, A. L. Bolaji, and M. Alkoffash, "Hybridization of harmony search with Hill climbing for highly constrained nurse rostering problem," *Neural Comput. Appl.*, vol. 28, no. 3, pp. 463–482, Mar. 2017.

[8] M. A. Awadallah, A. L. Bolaji, and M. A. Al-Betar, "A hybrid artificial bee colony for a nurse rostering problem," *Appl. Soft Comput.*, vol. 35, pp. 726–739, Oct. 2015.

[9] M. A. Awadallah, A. T. Khader, M. A. AL-Betar, and A. L. Bolaji, "Harmony search with novel selection methods in memory consideration for nurse rostering problem," *Asia–Pacific J. Oper. Res.*, vol. 31, no. 3, Jun. 2014, Art. no. 1450014.

[10] S. K. Azad and O. Hasançebi, "An elitist self-adaptive step-size search for structural design optimization," *Appl. Soft Comput.*, vol. 19, pp. 226–235, Jun. 2014.

[11] M. Bagheri, A. G. Devin, and A. Izanloo, "An application of stochastic programming method for nurse scheduling problem in real word hospital," *Comput. Ind. Eng.*, vol. 96, pp. 192–200, Jun. 2016.

[12] Z. Beheshti and S. M. Shamsuddin, "A review of population-based meta-heuristic algorithm," *Int. J. Adv. Soft Comput. Appl.*, vol. 5, no. 1, pp. 1–35, Mar. 2013.

[13] E. B. Böóvarsdóttir, P. Smet, G. V. Berghe, and T. J. R. Stidsen, "Achieving compromise solutions in nurse rostering by using automatically estimated acceptance thresholds," *Eur. J. Oper. Res.*, vol. 292, no. 3, pp. 980–995, Aug. 2021.

[14] E. K. Burke, P. De Causmaecker, G. V. Berghe, and H. Van Landeghem, "The state of the art of nurse rostering," *J. Scheduling*, vol. 7, no. 6, pp. 441–499, Nov. 2004.

[15] E. K. Burke and T. Curtois, "New approaches to nurse rostering benchmark instances," *Eur. J. Oper. Res.*, vol. 237, no. 1, pp. 71–81, Aug. 2014.

[16] E. K. Burke, T. Curtois, G. Post, R. Qu, and B. Veltman, "A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem," *Eur. J. Oper. Res.*, vol. 188, no. 2, pp. 330–341, 2008.

[17] E. K. Burke, T. Curtois, R. Qu, and G. V. Berghe, "A time predefined variable depth search for nurse rostering," *INFORMS J. Comput.*, vol. 25, no. 3, pp. 411–419, Aug. 2013.

[18] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), vol. 146. Boston, MA, USA: Springer, 2010.

[19] K. E. Burke and G. Kendall, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. New York, NY, USA: Springer, 2014.

[20] E. K. Burke, J. Li, and R. Qu, "A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems," *Eur. J. Oper. Res.*, vol. 203, no. 2, pp. 484–493, Jun. 2010.

[21] K. Buyukozkan and A. Sarucan, "Applicability of artificial bee colony algorithm for nurse scheduling problems," *Int. J. Comput. Intell. Syst.*, vol. 7, no. 1, p. 121, 2014.

[22] P. De Causmaecker and G. V. Berghe, "A categorisation of nurse rostering problems," *J. Scheduling*, vol. 14, no. 1, pp. 3–16, Feb. 2011.

[23] S. Ceschia, N. T. T. Dang, P. De Causmaecker, S. Haspeslagh, and A. Schaerf, "Second international nurse rostering competition (INRC-II)—Problem description and rules—," 2015, *arXiv:1501.04177*.

[24] S. Ceschia, R. Guido, and A. Schaerf, "Solving the static INRC-II nurse rostering problem by simulated annealing based on large neighborhoods," *Ann. Oper. Res.*, vol. 288, no. 1, pp. 95–113, May 2020.

[25] E. C. Yagmur and A. Sarucan, "Nurse scheduling with opposition-based parallel harmony search algorithm," *J. Intell. Syst.*, vol. 28, no. 4, pp. 633–647, Sep. 2019.

[26] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems—A bibliographic survey," *Eur. J. Oper. Res.*, vol. 151, no. 3, pp. 447–460, Dec. 2003.

[27] M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall, "A survey of university course timetabling problem: Perspectives, trends and opportunities," *IEEE Access*, vol. 9, pp. 106515–106529, 2021.

[28] P.-S. Chen and Z.-Y. Zeng, "Developing two heuristic algorithms with metaheuristic algorithms to improve solutions of optimization problems with soft and hard constraints: An application to nurse rostering problems," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106336.

[29] Y. Chen, A. Liu, E. Sciannella, and A. Zhang, "A comparison of approaches to the nurse scheduling problem," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., Dec. 2016.

[30] M. V. Chiaramonte, "Competitive nurse rostering and rerostering," Ph.D. dissertation, Dept. Ind. Eng., Arizona State Univ., Tempe, AZ, USA, May 2008, p. 154.

[31] A. A. Constantino, D. Landa-Silva, E. L. de Melo, C. F. X. de Mendonça, D. B. Rizzato, and W. Romão, "A heuristic algorithm based on multi-assignment procedures for nurse scheduling," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 165–183, Apr. 2013.

[32] P. Cowling, G. Kendall, and E. Soubeiga, "A hyperheuristic approach to scheduling a sales summit," in *Practice and Theory of Automated Timetabling III*. Berlin, Germany: Springer, 2001, pp. 176–190.

[33] T. Curtois and R. Qu, "Computational results on new staff scheduling benchmark instances," ASAP Res. Group, School Comput. Sci., Univ. Nottingham, Nottingham, U.K., Tech. Rep., 2014.

[34] F. D. Croce and F. Salassa, "A variable neighborhood search based matheuristic for nurse rostering problems," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 185–199, Jul. 2014.

[35] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, "Recent advances in selection hyper-heuristics," *Eur. J. Oper. Res.*, vol. 285, no. 2, pp. 405–428, Sep. 2020.

[36] S. L. Goh, G. Kendall, and N. R. Sabar, "Monte Carlo tree search in finding feasible solutions for course timetabling problem," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 6, p. 1936, Dec. 2019.

[37] S. L. Goh, G. Kendall, R. N. Sabar, and S. Abdullah, "An effective hybrid local search approach for the post enrolment course timetabling problem," *Opsearch*, vol. 57, no. 4, pp. 1131–1163, Jun. 2020.

[38] F. Guessoum, S. Haddadi, and E. Gattal, "Simple, yet fast and effective two-phase method for nurse rostering," *Amer. J. Math. Manage. Sci.*, vol. 39, no. 1, pp. 1–19, Jan. 2020.

[39] S. Haddadi, "Three-phase method for nurse rostering," *Int. J. Manage. Sci. Eng. Manage.*, vol. 14, no. 3, pp. 193–205, Jul. 2019.

[40] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Inf. Sci.*, vol. 233, pp. 126–140, Jun. 2013.

[41] S. Haspeslagh, P. De Causmaecker, A. Schaerf, and M. Stølevik, "The first international nurse rostering competition 2010," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 221–236, Jul. 2014.

[42] A. A. Hassan, S. Abdullah, K. Z. Zamli, and R. Razali, "Combinatorial test suites generation strategy utilizing the whale optimization algorithm," *IEEE Access*, vol. 8, pp. 192288–192303, 2020.

[43] M. R. Hassani and J. Behnamian, "A scenario-based robust optimization with a pessimistic approach for nurse rostering problem," *J. Combinat. Optim.*, vol. 41, no. 1, pp. 143–169, Jan. 2021.

[44] F. He and R. Qu, "A constraint programming based column generation approach to nurse rostering problems," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3331–3343, Dec. 2012.

[45] H. Huang, W. Lin, Z. Lin, Z. Hao, and A. Lim, "An evolutionary algorithm based on constraint set partitioning for nurse rostering problems," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 703–715, Sep. 2014.

[46] H. Jafari, S. Bateni, P. Daneshvar, S. Bateni, and H. Mahdioun, "Fuzzy mathematical modeling approach for the nurse scheduling problem: A case study," *Int. J. Fuzzy Syst.*, vol. 18, no. 2, pp. 320–332, Apr. 2016.

[47] G. Jaradat, M. Ayob, and I. Almarashdeh, "The effect of elite pool in hybrid population-based meta-heuristics for solving combinatorial optimization problems," *Appl. Soft Comput.*, vol. 44, pp. 45–56, Jul. 2016.

[48] S. H. Jin, H. Y. Yun, S. J. Jeong, and K. S. Kim, "Hybrid and cooperative strategies using harmony search and artificial immune systems for solving the nurse rostering problem," *Sustainability*, vol. 9, no. 7, p. 1090, Jun. 2017.

[49] G. Kendall, "Beall's legacy in the battle against predatory publishers," *Learned Publishing*, vol. 34, no. 3, pp. 379–388, Feb. 2021.

[50] A. Kheiri, A. Gretsista, E. Keedwell, G. Lulli, M. G. Epitropakis, and E. K. Burke, "A hyper-heuristic approach based upon a hidden Markov model for the multi-stage nurse rostering problem," *Comput. Oper. Res.*, vol. 130, Jun. 2021, Art. no. 105221.

[51] F. Knust and L. Xie, "Simulated annealing approach to nurse rostering benchmark and real-world instances," *Ann. Oper. Res.*, vol. 272, nos. 1–2, pp. 187–216, Jan. 2019.

[52] Z. Lü and J.-K. Hao, "Adaptive neighborhood search for nurse rostering," *Eur. J. Oper. Res.*, vol. 218, no. 3, pp. 865–876, May 2012.

[53] A. Legrain, J. Omer, and S. Rosat, "An online stochastic algorithm for a dynamic nurse scheduling problem," *Eur. J. Oper. Res.*, vol. 285, no. 1, pp. 196–210, Aug. 2020.

[54] A. Legrain, J. Omer, and S. Rosat, "A rotation-based branch-and-price approach for the nurse scheduling problem," *Math. Program. Comput.*, vol. 12, no. 3, pp. 417–450, Sep. 2020.

[55] J. Mandziuk, "New shades of the vehicle routing problem: Emerging problem formulations and computational intelligence solution methods," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 3, pp. 230–244, Jun. 2019.

[56] D. Meignan and S. Knust, "A neutrality-based iterated local search for shift scheduling optimization and interactive reoptimization," *Eur. J. Oper. Res.*, vol. 279, no. 2, pp. 320–334, Dec. 2019.

[57] C. Michael, C. Jeffery, and C. David, "Nurse preference rostering using agents and iterated local search," *Ann. Oper. Res.*, vol. 226, no. 1, pp. 443–461, Mar. 2015.

[58] H. E. Miller, W. P. Pierskalla, and G. J. Rath, "Nurse scheduling using mathematical programming," *Oper. Res.*, vol. 24, no. 5, pp. 857–870, Oct. 1976.

[59] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[60] F. Mischek and N. Musliu, "Integer programming model extensions for a multi-stage nurse rostering problem," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 123–143, Sep. 2017.

[61] T. Osogami and H. Imai, "Classification of various neighborhood operations for the nurse scheduling problem," in *Algorithms and Computation*. Berlin, Germany: Springer, 2000, pp. 72–83.

[62] E. Rahimian, K. Akartunalı, and J. Levine, "A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems," *Eur. J. Oper. Res.*, vol. 258, no. 2, pp. 411–423, Apr. 2017.

[63] E. Rahimian, K. Akartunalı, and J. Levine, "A hybrid integer and constraint programming approach to solve nurse rostering problems," *Comput. Oper. Res.*, vol. 82, pp. 83–94, Jun. 2017.

[64] M. Rajeswari, J. Amudhavel, S. Pothula, and P. Dhavachelvan, "Directed bee colony optimization algorithm to solve the nurse rostering problem," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–26, Apr. 2017.

[65] C. R. Reeves, *Modern Heuristic Search Methods, Chapter Modern Heuristic Techniques*. Hoboken, NJ, USA: Wiley, 1996, pp. 1–25.

[66] M. Romer and T. Mellouli, "A direct MILP approach based on state-expanded network flows and anticipation for multi-stage nurse rostering under uncertainty," in *Proc. 11th Int. Conf. Pract. Theory Automated Timetabling (PATAT)*, 2016, pp. 549–551.

[67] N. R. Sabar, S. L. Goh, A. Turky, and G. Kendall, "Population-based iterated local search approach for dynamic vehicle routing problems," *IEEE Trans. Autom. Sci. Eng.*, early access, Aug. 13, 2021, doi: 10.1109/TASE.2021.3097778.

[68] F. Salassa and G. V. Berghe, "A stepping horizon view on nurse rostering," in *Proc. 9th Int. Conf. Pract. Theory Automated Timetabling*, 2012, pp. 161–173.

[69] H. G. Santos, T. A. M. Toffolo, R. A. M. Gomes, and S. Ribas, "Integer programming techniques for the nurse rostering problem," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 225–251, Apr. 2016.

[70] B. Singh and M. Eisner, "Optimization/mathematical programming," Inst. Oper. Res. Manage. Sci., Catonsville, MD, USA, Tech. Rep., 2014.

[71] P. Smet, B. Bilgin, P. De Causmaecker, and G. V. Berghe, "Modelling and evaluation issues in nurse rostering," *Ann. Oper. Res.*, vol. 218, no. 1, pp. 303–326, Jul. 2014.

[72] P. Smet, P. Brucker, P. De Causmaecker, and G. V. Berghe, "Polynomially solvable personnel rostering problems," *Eur. J. Oper. Res.*, vol. 249, no. 1, pp. 67–75, Feb. 2016.

[73] P. Smet, P. D. Causmaecker, B. Bilgin, and G. V. Berghe, "Nurse rostering: A complex example of personnel scheduling with perspectives," in *Automated Scheduling and Planning*. Berlin, Germany: Springer, 2013.

[74] P. Strandmark, Y. Qu, and T. Curtois, "First-order linear programming in a column generation-based heuristic approach to the nurse rostering problem," *Comput. Oper. Res.*, vol. 120, Aug. 2020, Art. no. 104945.

[75] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113943.

[76] J. S. Tan, S. L. Goh, S. Sura, G. Kendall, and N. R. Sabar, "Hybrid particle swarm optimization with particle elimination for the high school timetabling problem," *Evol. Intell.*, vol. 14, no. 4, pp. 1915–1930, Dec. 2021.

[77] I. X. Tassopoulos, I. P. Solos, and G. N. Beligiannis, "A two-phase adaptive variable neighborhood approach for nurse rostering," *Comput. Oper. Res.*, vol. 60, pp. 150–169, Aug. 2015.

[78] A. M. Turhan and B. Bilgen, "A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem," *Comput. Ind. Eng.*, vol. 145, Jul. 2020, Art. no. 106531.

[79] C. Valouxis, C. Gogos, G. Goulas, P. Alefragis, and E. Housos, "A systematic two phase approach for the nurse rostering problem," *Eur. J. Oper. Res.*, vol. 219, no. 2, pp. 425–433, Jun. 2012.

[80] M. Vanhoucke and B. Maenhout, "NSPLib—A nurse scheduling problem library: A tool to evaluate (meta-) heuristic procedures," in *Proc. 31st Annu. Meeting Work. Group Oper. Res. Appl. Health Services*, Jan. 2007, pp. 151–165.

[81] D. M. Warner, "Scheduling nursing personnel according to nursing preference: A mathematical programming approach," *Oper. Res.*, vol. 24, no. 5, pp. 842–856, Oct. 1976.

[82] T. I. Wickert, A. F. K. Neto, M. M. Boniatti, and L. S. Buriol, "An integer programming approach for the physician rostering problem," *Ann. Oper. Res.*, vol. 302, no. 2, pp. 363–390, Jul. 2021.

[83] T. I. Wickert, P. Smet, and G. V. Berghe, "The nurse rerostering problem: Strategies for reconstructing disrupted schedules," *Comput. Oper. Res.*, vol. 104, pp. 319–337, Apr. 2019.

[84] T.-H. Wu, J.-Y. Yeh, and Y.-M. Lee, "A particle swarm optimization approach with refinement procedure for nurse rostering problem," *Comput. Oper. Res.*, vol. 54, pp. 52–63, Feb. 2015.

[85] P.-Y. Yin and Y.-T. Chiang, "Cyber swarm algorithms for multi-objective nurse rostering problem," *Int. J. Innov. Comput., Inf. Control*, vol. 9, no. 5, pp. 2043–2063, May 2013.

[86] S. Zanda, P. Zuddas, and C. Seatzu, "Long term nurse scheduling via a decision support system based on linear integer programming: A case study at the university hospital in Cagliari," *Comput. Ind. Eng.*, vol. 126, pp. 337–347, Dec. 2018.

[87] Z. Zheng, X. Liu, and X. Gong, "A simple randomized variable neighbourhood search for nurse rostering," *Comput. Ind. Eng.*, vol. 110, pp. 165–174, Aug. 2017.

CHONG MAN NGOO received the bachelor's degree (Hons.) in business computing (IT) from the Universiti Malaysia Sabah, Malaysia, where he is currently pursuing the master's degree with the Faculty of Computing and Informatics. His current research interest includes meta-heuristic algorithms with specific interest in scheduling problems.

SAY LENG GOH received the B.I.T. degree (Hons.) from the Universiti Malaysia Sabah, Malaysia, the M.Sc. degree from Imperial College London, and the Ph.D. degree from the University of Nottingham. He is currently an Academician with the Faculty of Computing and Informatics, Universiti Malaysia Sabah. He has published in various top-tier journals such as *European Journal of Operational Research*, *Journal of the Operational Research Society*, and *Expert Systems with Applications*. His current research interests include artificial intelligence, operational research, discrete optimization, and meta-heuristics, timetabling, and scheduling.

SAN NAH SZE received the Doctor of Philosophy (Ph.D.) degree from The University of Sydney, in 2011. She is currently working as a Senior Lecturer at the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS). She has more than ten years of experience in research and development (R&D). She had presented at numerous conferences and published her work in ISI and Scopus publications. Her research interests include educational timetabling, vehicle routing, heuristics, and meta-heuristics.

NASSER R. SABAR is currently a Lecturer with the Computer Science and IT Department, La Trobe University, Australia. He has published over 76 papers in international journals and peer-reviewed conferences. His current research interests include hyper-heuristic frameworks, evolutionary computation, and hybrid algorithms with a specific interest in big data optimization problems, cloud computing, dynamic optimization, and scheduling problems.

SALWANI ABDULLAH received the B.Sc. degree in computer science from the University of Technology Malaysia, the master's degree in computer science from the National University of Malaysia, and the Ph.D. degree in computer science from the University of Nottingham, U.K. Now, she is a Professor in computational optimization at the Faculty of Information Science and Technology, National University of Malaysia. Her research interests include artificial intelligence and operation research, particularly computational optimization algorithms (heuristic and meta-heuristic, evolutionary algorithms, and local search) that involve different real-world applications for single and multiobjective continuous and combinatorial optimization problems such as timetabling, scheduling, routing, nurse rostering, dynamic optimization, data mining problems (feature selection, clustering, classification, and time-series prediction), intrusion detection, and search-based software testing.

GRAHAM KENDALL (Senior Member, IEEE) is currently the Senior Vice President of the PETRA Group and the Chief Executive of the Good Capitalism Forum (GCF). Prior to this, he worked at the University of Nottingham for 21 years and, prior to this, he worked in the IT sector, holding several senior positions. Prior to joining the PETRA Group, he worked in Malaysia for ten years, being-based at the University of Nottingham's campus, Semenyih, where he was the CEO and Provost, and also the Pro-Vice Chancellor of Global University. He holds honorary and distinguished professorial positions at universities in Hong Kong and India. He is also an Emeritus Professor at the University of Nottingham. He is a world recognized academic, with his research focusing on solving real world problems through the use of operations research, utilizing artificial intelligence methods which are based on Charles Darwin's principles of natural evolution. He has written over 250 peer-reviewed papers.

Dr. Kendall is (or has been, prior to leaving academia) a fellow of both the British Computer Society and the Operational Research Society. He has served on several editorial boards including being the Editor-in-Chief for the IEEE Transactions on Computational Intelligence and AIin Games and an Associate Editor for the *Journal of the Operational Research Society*.