UNIVERSITI TEKNOLOGI MARA
Cawangan Kedah
Kampus Sungai Petani

2021 iCMS
INTERNATIONAL CONFERENCE ON COMPUTING, MATHEMATICS AND STATISTICS

e-PROCEEDINGS

of The 5th International Conference on Computing, Mathematics and Statistics (iCMS2021)

4-5 August 2021

Driving Research Towards Excellence

e-Proceedings of the 5th International Conference on Computing, Mathematics and Statistics (iCMS 2021)

Driving Research Towards Excellence

e-ISBN: 978-967-2948-12-4
DOI

Library of Congress Control Number:

Publication by
Department of Mathematical Sciences
Faculty of Computer & Mathematical Sciences
UiTM Kedah

TABLE OF CONTENT

PART 1: MATHEMATICS

PART 2: STATISTICS

PART 3: COMPUTER SCIENCE & INFORMATION TECHNOLOGY

PART 4: OTHERS

HARMONY SEARCH HYPER-HEURISTIC WITH DIFFERENT PITCH ADJUSTMENT OPERATOR FOR SCHEDULING PROBLEMS

Khairul Anwar[1] , Mohammed A.Awadallah [2], Mohammed Azmi Al-Betar [3]
[1] Universiti Teknologi MARA (UiTM) Cawangan Sabah Kampus Kota Kinabalu Beg Berkunci 71, 88997 Kota Kinabalu, Sabah, MALAYSIA.
[2] Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, United Arab Emirates.
[3] Department of Electrical and Computer Engineering College of Engineering and Information Technology Ajman University Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Al-Huson, Irbid, Jordan.
([1]khair274@uitm.edu.my [2]ma.awadallah@alaqsa.edu.ps [3]mohbetar@bau.edu.jo)

The Harmony Search Algorithm (HSA) has been adapted in the hyper-heuristics framework as the high-level heuristic named *Harmony Search-based Hyper Heuristic* (HSHH). Two operators in HSA are used to select and generate a heuristics vector. In this paper the pitch adjustment operator was used in order to enhance the original HSHH. The purpose of applying the pitch adjustment operator was to insert a different way of heuristic selection instead of randomness in memory consideration operator. The effectiveness of the proposed methods was tested with two distinct timetabling and rostering problems. Experimentally, the new HSHH method had improved the original HSHH.

Keywords: Hyperheuristic, Harmony search, Scheduling, Timetabling

1. Introduction

Scheduling problems occur in almost all fields or domains related to services especially in health, transportation and, even educational institutions. Generally, the main purpose of producing the scheduling is to organize people's lives, activities, and works. Numerous metaheuristic approaches have been produced in order to solve the scheduling problems. The most studied approaches in the past twenty years include tabu search, simulated annealing, scatter search, ant algorithms, and genetic algorithms (Bianchi et al., 2009; Gendreau and Potvin, 2005). However, there are also researches in optimization toward general approaches that can solve not only as single problem but a host of problems. This can be observed when these general approaches i.e. *Hyperheuristic* have been successfully used in various scheduling problems such as personnel scheduling problem (Cowling et al., 2001), course timetabling problems (Burke et al., 2003), examination timetabling problem (Burke et al., 2012; Kendall and Hussin, 2005; Pillay and Banzhaf, 2009; Qu and Burke, 2009; Sabar and Ayob, 2009; Sabar et al., 2012), nurse rostering problem (Bilgin et al., 2012, 2010; Burke and Soubeiga, 2003), vehicle routing (Garrido and Castro, 2009; Mlejnek and Kubalik, 2013) and others (Burke et al., 2013; Pillay, 2016; Drake et al., 2020). These approaches aim to produce a method that "*good enough-soon enough-cheap enough*" to solve optimization problems in hand (Burke et al., 2013).

Generally, hyper-heuristics also referred to as a heuristic or meta-heuristic, select one or more low-level heuristics for the particular problem in the hope of solving the problem in hand. The original definition of *Hyperheuristic* can be described as *"heuristics to choose heuristics"* in the context of combinatorial optimisation problems(Cowling et al., 2001). The main motivation of hyper-heuristic approaches is to produce automated heuristic design methods to solve the hard computational optimization problems (Burke et al., 2013). The major defining property of hyper-heuristics is that, this method (Hyperheuristics) operates on the search space of heuristics (or heuristic component) rather than directly on the search space of solution (Burke et al., 2013). A hyper-heuristic then can be regarded as an 'off-the-peg' method as opposed to a 'made-to-measure' bespoke metaheuristic (Burke et al., 2009). In a traditional framework of hyperheuristic, it consists of two fundamental mechanisms: a heuristic selection mechanism and move acceptance mechanism. Heuristic selection mechanism choose the best low-level heuristic for generating a new (partial or complete) solution in the current optimization step and move acceptance mechanism to decide on the acceptability of the new solution.

This paper presents, the enhancement of HSHH method Anwar et al. (2013) called as HSHH-MPA. Previous work the HSHH only applied two HSA operators namely *memory consideration* and *random consideration*. The main contribution of this paper is to introduce three differences way of selection mechanism in HSHH through the pitch adjustment operator.

The outline of the paper is as follows. Section 2 introduce the harmony search hyper-heuristic approach and discusses in detail the high level selection strategy, and the move acceptance mechanism in harmony search hyper-heuristic. Section 3 deliberate the scheduling problems and, also on the low-level heuristics. Section 4 discusses the experimental setup and the results. The results are compared against results of previous works for the same benchmark problems. Section 5 concludes this study and recommends directions for future research.

2. Harmony search-based hyper-heuristic (HSHH)

The original idea of HSHH is to apply a sequence of low-level heuristics to a selected solution in order to produce good quality solutions to given problems. HSHH consists of two different search spaces: heuristic search space and solution search space. Heuristic search space contains sets of heuristics vectors (or individual), whereby every vector is a heuristics sequence. The representation of heuristic vector is a sequence of integers, each of which represents one low-level heuristic. This sequence of heuristics, tells which low level heuristic to use and in what order to apply it. Solution search space consists of sets of complete feasible solutions. Basically, HSHH has five main steps, as follows:

Step 1: **Initialization.** The HSHH begins by setting the harmony search parameters such as harmony memory size (HMS), harmony memory consideration rate (HMCR), number of iterations (NI) and Harmony Memory Length (HML). Pitch Adjustment Rate (PAR) parameter will also be set during this step.

Step 2: **Initializing of Harmony Memory.** In initializing Heuristic Harmony Memory (HHM) and Solution Harmony Memory (SHM), firstly, the HSHH, constructs the feasible solution by using constructive heuristic. In order to fill the SHM, this process will be repeated until HMS is completed (see equation 2). On the other hand, to construct the HHM, different heuristic vectors are initialized by using the random technique. In this process, a low-level heuristic index number is randomly selected from the existing set of low-level heuristic provided by the domain problem and put in a sequence (e.g., $1, 3, 2, 1 \ldots h_{HML}$). This process will be continued until the HMS is completed (see equation 1).

$$\textbf{HHM} = \begin{bmatrix} h_1^1 & h_2^1 & \cdots & h_{HML}^1 \\ h_1^2 & h_2^2 & \cdots & h_{HML}^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{HMS} & h_2^{HMS} & \cdots & h_{HML}^{HMS} \end{bmatrix}. \tag{1}$$

$$\textbf{SHM} = \left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & f(x^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} & f(x^{HMS}) \end{array} \right]. \tag{2}$$

Step 3: **Improvise a new heuristic vector.** In this step, a new heuristic vector ($\boldsymbol{h}'$) is generated from scratch, based on three HSA operators: memory consideration, random consideration and pitch adjustment operator.

In *Memory Consideration (MC) operator*, the new heuristic index of h_i' in the new heuristic vector ($\boldsymbol{h}'$) is randomly selected from the historical indexes (e.g. h_i^1,h_i^2,. . .,h_i^{HMS}), stored in the heuristic harmony memory with probability of HMCR, where HMCR $\in$ [0, 1].

For *Random Consideration (RC) operator*, the new heuristic index of h_i' is randomly assigned from the set of heuristics H where H = $\{h_1$,h_2,. . .,$h_{LLH}\}$ with probability of (1-HMCR) as in equation 3. Note that the LLH is the number of heuristics provided by problem domains.

$$h_i^{'}\epsilon \begin{cases} h_i^1, h_i^2, \ldots, h_i^{HMS}, & w.p \quad HMCR \\ \{h_1, h_2, \ldots, h_{LLH}\}, & w.p \quad (1 - HMCR) \end{cases} \tag{3}$$

In *Pitch Adjustment (PA) operator*, three different selection mechanisms are proposed. All these mechanisms will be executed with a probability of PAR where $0 \leq PAR \leq 1$ as in equation 4.

$$h_i^{'} = \begin{cases} Yes, & w.p \quad PAR \\ No, & w.p \quad (1 - PAR) \end{cases} \tag{4}$$

First selection mechanism (*PAType1*) in pitch adjustment operator is a simple adjustment, for which the new index of h_i' will be added/substracted by 1. In second selection mechanism (*PAType2*), the new index of h_i' will be selected from the best heuristic sequence in the HHM. For the third selection mechanism (*PAType3*) the new index of h_i' will be selected based on the lowest heuristics usage in the HHM.

Then the new harmony of heuristic vector $\boldsymbol{h}'$ will be applied to a selected solution (x^{rand}) where x^{rand} is randomly selected from SHM. The HSHH uses random selection to select the solution from the SHM to avoid the local optima. In this process, the heuristic in $\boldsymbol{h}'$ will be executed sequentially to the selected solution x^{rand}. The process will continue until all the heuristics in $\boldsymbol{h}'$ have been applied, and a new solution $\boldsymbol{x}'$ will be produced. Algorithm 1 shows the pseudo-code for step 3 in HSHH.

Step 4: **Update HHM and SHM**. In hyper-heuristic framework, this step is called move acceptance mechanism. HSHH will decide either to accept or neglect the new heuristic vector ($\boldsymbol{h}'$). In this process, the new solution ($\boldsymbol{x}'$) will be evaluated using the objective function. The new solution must be a complete and feasible solution. If the new solution is better than the selected solution (x^{rand}) in SHM, the new $\boldsymbol{h}'$ and $\boldsymbol{x}'$ will be accepted and saved in the memory ($\boldsymbol{h}'$ in HHM and $\boldsymbol{x}'$ in SHM) and the heuristic vector h^{rand} and solution x^{rand} will be excluded from the memory (i.e., HHM and SHM).

Step 5: **Check the stop criterion**. Step 3 and step 4 in this approach are repeated until the stop criterion (i.e., NI) is met.

Algorithm 1 Pseudo-code for selecting heuristic and generating new heuristic vector during improvisation step in HSHH-MPA

$\boldsymbol{h'}$ = 0; *vector of heuristic sequences*
for $i = 0$ to HML-1 **do**
  if $U(0,1) \leq$ HMCR **then**
    Memory consideration
    if $U(0,1) \leq$ PAR **then**
      Pitch Adjustment for h'_i
    end if
  else
    Random consideration
  end if
end for
$x^{rand} \in \{x^1,x^2,\ldots,x^{HMS}\}$ Select random solution from SHM
$\boldsymbol{x'}$ = apply $\boldsymbol{h'}$ to x^{rand}

3. The Scheduling Problems

The proposed HSHH-MPA is evaluated using two different scheduling problems i.e., examination timetabling and nurse rostering. For examination timetabling, the Carter dataset (Carter et al., 1996) will be used. This dataset provides a common framework for meaningful evaluations of the proposed methods because this dataset varies in terms of size (exams, and timeslots) and complexity. Futhermore, Carter dataset is a real-world examination timetabling problem that comprises 12 datasets. Meanwhile, the second scheduling problem used in this study is the first standard dataset for nurse rostering problem established by the organizers of International Nurse Rostering Competition 2010 (INRC2010) [1]. This dataset can be divided into three tracks, namely *sprint*, *medium*, and *long* instances. These tracks are different in terms of size and complexity. Each track is further classified into four categories which are *early*, *hidden*, *late*, and *hint*. Note that in this study, only the *early* tracks will be used. The characteristics of the datasets used in this study are provided in Table 1 and Table 2 respectively.

In order to construct a feasible solution for Carter dataset during the initialization steps, the Saturation Degree (SD) algorithm has been adopted for ordering the exams such as in Al-Betar et al. (2010). Same goes to INRC2010 dataset, in finding a feasible solution, heuristic ordering as implemented by Awadallah et al. (2011) will be used. Several heuristics have been utilized as low-level heuristics (LLH) depending on the problem domain. In this study, three neighbourhood structures have been utilized as low level heuristics for Carter dataset and three improvement heuristics are employed for INRC2010 dataset. They can be summarized as follows:

1. Examination Timetabling Problem (Carter Dataset):
   - **h1: Move Exam**. Select one exam and move the selected exam to a new feasible timeslot randomly. For instance, replace the timeslot x'_i of exam i with another feasible timeslot.
   - **h2: Swap-Timeslot**. Select two exams at random and swap their timeslots. For instance, randomly select exam i and exam j and swap their timeslots x'_i,x'_j) while the feasibility is preserved.
   - **h3: Kempe Chain**. Select two timeslots randomly and exchange the conflicting exams.
2. Nurse rostering (INRC Dataset):

[1]Detail about INRC can found in https://www.kuleuven-kulak.be/nrpcompetition

Table 1: The characteristics of Carter's dataset (Carter et al., 1996)

Datasets	Institution	Exams	Students	Timeslots	Density
CAR-S-91-I	Carleton University	682	16,925	35	0.13
CAR-F-92-I	Carleton University	543	18,419	32	0.14
EAR-F-83-I	Earl Haig Collegiate Institute	190	1125	24	0.27
HEC-S-92-I	Ecole des Hautes Etudes Commercials	81	2823	18	0.42
KFU-S-93	King Fahd University	461	5349	20	0.06
LSE-F-91	London School of Economics	381	2726	18	0.06
RYE-S-93	Ryeson University	481	11,483	23	40.07
STA-F-83-I	St. Andrew's Junior High School	139	611	13	0.14
TRE-S-92	Trent University Peterborough	261	4360	23	0.18
UTA-S-92-I	University of Toronto	622	21,266	35	0.13
UTE-S-92	University of Toronto	184	2750	10	0.08
YOR-F-83	York Mills Collegiate Institute	181	941	21	0.29

- **h1: Move one shift**. This heuristic is used to move the shift of current nurse to another nurse who is selected randomly while maintaining feasibility.
- **h2: Swap one shift**. In this heuristic, the shift allocated to a specific nurse is exchanged with the shift of another nurse while maintaining feasibility. Note that both nurses are assigned to two different shifts on the same day.
- **h3: Shuffle moves**. This heuristic swaps patterns between the worst nurse schedules based on their penalty values from the roster with a random nurse. The length of the patterns to be exchanged is specified by day, beginning with one day and increased by a day each time until the scheduling period is completed.

4. Experimental and results

A series of experiments were carried out to evaluate the proposed HSHH-MPA. The proposed method is coded in C++ and the experiments were conducted under windows platforms on Intel processor with 8G RAM. Each case has different selection mechanism in pitch adjustment operator as shown in Table 3. Experiment *case1*, *case2* and *case3* were used in order to study the performances of HSHH-MPA with different heuristic selection (i.e., *PAType1, PAType2, and PAType3*). Each experimental case was replicated 10 times for each benchmark instance with the most suitable iteration numbers fixed to 50,000 for all runs. Note that all the parameters value (i.e., HML, HMS, HMCR, and PAR) were set after running several experiments to make sure the proposed methods achieve the best results when applied with different selection mechanism. In addition, as aforementioned, the value of HMCR determines the rate of using memory consideration instead of random consideration operator in the HSHH-MPA. It must be stated that the pitch adjustment rate in this experiment was set as 10% of the usage (PAR=0.1).

The results for the experiments are presented in Table 4 and Table 5. Note that the numbers in these tables refer to the lowest of the penalty value of the soft constraints violations over 10 runs for all experimental cases on Carter and INRC2010 datasets. The lowest penalty value is the best among the experimental cases (e.g., case1, case2 and case3) on each problem instance (highlighted in bold). The performance of HSHH-MPA with three different selection mechanisms are presented in Table 6. This table shows the percentages of HSHH-MPA for each type of selection mechanism that was able to achieve the lowest results. Note that the high percentage rate indicates the best performance. Referring to this table, HSHH-MPA with *PAType2* showed the best performance in the Carter dataset and INRC2010 dataset which were 66.67% and 70% respectively. Clearly, the HSHH-MPA with *PAType2* showed better performance among the three selection mechanisms used in this study. On the other hand, when comparing HSHH-MPA with the original HSHH (i.e.,B-HSHH), the experimental results showed that the HSHH-MPA were able to produce a slightly better result.

Table 2: The characteristics of the *early track* in INRC2010

Type	Index	Shifts	Skills	Contracts	UnWanted	Weekend	Day Off	Shift Off	Period
Sprint	01-10	4	1	4	3	2	√	√	1-28/01/2010
Medium	01-05	4	1	4	0	2	√	√	1-28/01/2010
Long	01-05	5	2	3	3	2	√	√	1-28/01/2010

Table 3: Different cases to study the effectiveness of the proposed method

Cases	HML	HMS	HMCR	PAR	PATypes
case1	10	10	0.99	0.1	*PAType1*
case2	10	10	0.99	0.1	*PAType2*
case3	10	10	0.99	0.1	*PAType3*

Table 4: Results of HSHH-MPA with different selection mechanism (i.e., *PAType1*, *PAType2* and *PAType3*) for Carter dataset.

Instance	B-HSHH	case1 (PAType1)	case2 (PAType2)	case3 (PAType3)
CAR-S-91-1	6.09	6.13	**6.00**	6.04
CAR-F-92-1	**4.88**	4.92	**4.88**	4.97
EAR-F-83-1	38.71	**37.22**	38.22	37.26
HEC-S-92-1	11.16	10.75	**10.74**	11.05
KFU-S-93	14.71	14.82	**14.78**	14.87
LSE-F-91	**11.97**	12.37	12.02	12.09
RYE-S-93	10.00	10.04	**9.83**	9.84
STA-F-83-1	**157.31**	157.38	157.40	157.32
TRE-S-92	9.25	9.10	9.20	**9.05**
UTA-S-92-1	3.84	3.82	**3.76**	3.82
UTE-S-92	27.07	26.89	**26.58**	27.14
YOR-F-83	39.53	38.92	**38.90**	39.85

Table 5: Results of HSHH-MPA with different selection mechanism (i.e., *PAType1*, *PAType2* and *PAType3*) for INRC2010 dataset.

Instance	B-HSHH	case1 (PAType1)	case2 (PAType2)	case3 (PAType3)
sprint early01	58	**57**	**57**	58
sprint early02	60	**59**	**59**	60
sprint early03	53	53	**52**	53
sprint early04	62	62	**61**	**61**
sprint early05	**58**	59	59	**58**
sprint early06	**55**	**55**	**55**	**55**
sprint early07	**58**	**58**	**58**	**58**
sprint early08	**56**	57	**56**	57
sprint early09	57	57	**56**	57
sprint early10	54	54	**53**	**53**
medium early01	**249**	251	250	**249**
medium early02	251	**250**	**250**	251
medium early03	247	**246**	**246**	249
medium early04	248	247	**246**	248
medium early05	315	**312**	313	314
long early01	214	213	214	**212**
long early02	245	242	**241**	243
long early03	248	**245**	246	**245**
long early04	317	**316**	**316**	317
long early05	298	**292**	295	296

Table 6: The performance of original HSHH and HSHH-MPA with different selection mechanism (i.e., *PAType1*, *PAType2* and *PAType3*) for two datasets (i.e., Carter and INRC2010).

	Percentage % of lowest results achieved			
Dataset	B-HSHH	*PAType1*	*PAType2*	*PAType3*
Carter's dataset	25	8.33	**66.67**	8.33
INRC2010	20	50	**70**	40

5. Conclusions

In this study, the Harmony search-based hyper-heuristic with pitch adjustment operator (i.e., HSHH-MPA) is presented with three types of heuristic selection mechanism (*PAType1*, *PAType2*, and *PAType3*). The purpose of applying the pitch adjustment operator is to insert a different way of heuristic selection (e.g., using learning mechanism) instead of randomness in the memory consideration operator. Apparently, by combining the HSHH with pitch adjustment operators (HSHH-MPA), it was slightly better compared to the original HSHH approach (B-HSHH) in most of problem instances in both datasets (Carter and INRC2010).

The performance of HSHH-MPA with three different selection mechanism (*PAType1*, *PAType2* and *PAType3*) were studied through a series of experiments. The purpose of these experiments is to observe the effects of selection mechanism to the heuristic usage in HHM. Based on the experimental performances, HSHH-MPA with *PAType2* appeared to be the best selection mechanism. The computational results indicated that selecting the heuristic from best heuristics vector inside HHM, had led the algorithm to select the most appropriate heuristics. Thus, using the most appropriates heuristic at the right time could lead to better results.

The focus of this study is the selection mechanism for choosing low-level heuristics. It can be in the best interest of future work in the same field, to test the effectiveness of different ways of solution selection from solution harmony memory (SHM) and different move acceptance criteria with other methods.

References

Al-Betar, M., Khader, A., and Nadi, F. (2010). Selection mechanisms in memory consideration for examination timetabling with harmony search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO '10)*, pages 1203–1210. ACM.

Anwar, K., Khader, A. T., Al-Betar, M. A., and Awadallah, M. A. (2013). Harmony search-based hyper-heuristic for examination timetabling. In *9th International Colloquium on Signal Processing and its Applications (CSPA)*, pages 176–181. IEEE.

Awadallah, M. A., Khader, A. T., Al-Betar, M. A., and Bolaji, A. L. (2011). Nurse scheduling using harmony search. In *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on*, pages 58–63. IEEE.

Bianchi, L., Dorigo, M., Gambardella, L. M., and Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, 8(2):239–287.

Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., and Berghe, G. V. (2012). One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics*, 18(3):401–434.

Bilgin, B., Demeester, P., Mısır, M., Vancroonenburg, W., Berghe, G. V., and Wauters, T. (2010). A hyper-heuristic combined with a greedy shuffle approach to the nurse rostering competition. In *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT'10)*.

Burke, E. and Soubeiga, E. (2003). Scheduling nurses using a tabu-search hyperheuristic. In *Proceedings of the 1st multidisciplinary international conference on scheduling: Theory and applications (MISTA 2003), Nottingham, UK*, pages 180–197.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724.

Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., and Qu, R. (2009). A survey of hyper-heuristics. *Computer Science Technical Report No. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham.*

Burke, E. K., Kendall, G., Mısır, M., and Özcan, E. (2012). Monte carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*, 196(1):73–90.

Burke, E. K., MacCarthy, B. L., Petrovic, S., and Qu, R. (2003). Knowledge discovery in a hyper-heuristic for course timetabling using case-based reasoning. In *Practice and Theory of Automated Timetabling IV*, pages 276–287. Springer.

Carter, M. W., Laporte, G., and Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, pages 373–383.

Cowling, P., Kendall, G., and Soubeiga, E. (2001). A hyperheuristic approach to scheduling a sales summit. In *Practice and Theory of Automated Timetabling III*, pages 176–190. Springer.

Drake, J. H., Kheiri, A., Özcan, E., and Burke, E. K. (2020). Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, 285(2):405–428.

Garrido, P. and Castro, C. (2009). Stable solving of cvrps using hyperheuristics. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 255–262. ACM.

Gendreau, M. and Potvin, J.-Y. (2005). Metaheuristics in combinatorial optimization. *Annals of Operations Research*, 140(1):189–213.

Kendall, G. and Hussin, N. M. (2005). A tabu search hyper-heuristic approach to the examination timetabling problem at the mara university of technology. In *Practice and Theory of Automated Timetabling V*, pages 270–293. Springer.

Mlejnek, J. and Kubalik, J. (2013). Evolutionary hyperheuristic for capacitated vehicle routing problem. In *Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion*, pages 219–220. ACM.

Pillay, N. (2016). A review of hyper-heuristics for educational timetabling. *Annals of Operations Research*, 239(1):3–38.

Pillay, N. and Banzhaf, W. (2009). A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem. *European Journal of Operational Research*, 197(2):482–491.

Qu, R. and Burke, E. K. (2009). Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems. *Journal of the Operational Research Society*, 60(9):1273–1285.

Sabar, N. R. and Ayob, M. (2009). Examination timetabling using scatter search hyper-heuristic. In *Data Mining and Optimization, 2009. DMO'09. 2nd Conference on*, pages 127–131. IEEE.

Sabar, N. R., Ayob, M., Qu, R., and Kendall, G. (2012). A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence*, 37(1):1–11.

2021 ICMS

INTERNATIONAL CONFERENCE ON COMPUTING, MATHEMATICS AND STATISTICS

e ISBN 978-967-2948-12-4

9 7 8 9 6 7 2 9 4 8 1 2 4