The 6th International Seminar on Science and Technology (ISST) 2020
July 25th 2020, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Evaluation of Hyper-Heuristic Method Using Simple Random-Step Counting Hill Climbing in the Examination Timetabling Problem

Rusdi Hamidan and Ahmad Mukhlason
Department of Information System, Institut Teknologi Sepuluh Nopember, Surabaya
e-mail: mukhlason@is.its.ac.id

***Abstrak*—Exam Timetabling Problem (ETP) is a problem that occurs at the university. Solution to the ETP problem involves computational search methods to get results. In the process, if done manually it will require lot of time to achieve the optimal solution. ETP is basically allocating a schedule into room at particular time. Several previous researchers developed a hyper-heuristic method to obtain solutions that are expected to provide result that are close to optimal. In this study, ITC 2007 dataset will be used to find generic solutions that are near optimal. Simple Random (SR) was chosen as strategy to choose Low Level Heuristic (LLH) and Step Counting Hill Climbing (SCHC) was chosen as move-acceptance strategy for ETP. The results obtained show that one pair of algorithms proposed in this study is better than the literature while other algorithms also provide significant results.**

***Kata Kunci*—Examination Timetabling Problem, ITC 2007, Hyper-Heuristic, Hill Climbing, Simple Random.**

I. INTRODUCTION

CURRENTLY the use of operations research has been widely used in various field such as transportation, supply chain management, sport, government, manufacturing, and education. Example of implementation in the field of transportation are flight scheduling, train departures, while in the education sector are scheduling lessons and examinations. Variations in educational scheduling are school scheduling, university course scheduling, and exam scheduling [1]. This research will focus on exam timetabling.

Scheduling has rules (constraint) that must not be violated, usually categorized into two, namely hard constraint and soft constraint. Hard constraint should not be violated in any situation that if hard constraint violated it will make solution infeasible, for example exams that collide with each other or student cannot carry out more than one exam at the same time. Soft constraint is a desirable but does not cause problems if violated. Violations of soft constraint will cause penalty in the scheduling. Example of soft constraints in scheduling an exam are giving sufficient time between exams so that students can make time to do the learning or review [1].

This scheduling problem is formulated in a datasets. Carter and ITC 2007 dataset is one of the benchmarks used to represent exam scheduling problems. The difference between two dataset is that the ITC 2007 dataset offers a far more complex representation of scheduling problems in the real world [3]. In this research will focus on ITC 2007 dataset.

Examination timetabling is a NP-complete problem where there is no algorithm that is really able to solve this problem in non-polynomial timeframe [2]. As a result of this complexity, the solution will be more appropriate when using the heuristic method rather than using the exact method. Hyper-heuristic and meta-heuristic is an example of this methodology.

Hyper-heuristic is a general algorithm that can solve many problems because this method works in determining the low-level-heuristic for problem, not directly to the solution. This research will focus on the methodology for choosing a heuristic strategy with two step, heuristic selection and move acceptance.

This study evaluates the Simple Random - Step Counting Hill Climbing – Hyper Heuristic (SR-SCHC-HH). Therefore this study evaluates Simple Random algorithm as a heuristic selection while Hill Climbing and Step Count Hill Climbing as move acceptance for examination timetabling problem and testing it on the ITC 2007 dataset. List of low-level-heuristic that used in this study will be explained further.

II. METHOD

A. Problem Domain

Over the last few decades, there is dataset that has caught the attention of researchers especially scheduling dataset in education. Scientific research that first discussed scheduling problems began in the 1960 to become one of the most popular studies today [4]. The ITC 2007 dataset was introduced as a form of Toronto development which is one of the benchmark dataset. The difference lies in the addition of new hard constraint and soft constraint that are more complex [5]. Instances in the ITC 2007 dataset can be seen in Table 1. The following hard constraint are found in the ITC 2007 dataset:

1. No student sits more than one examination at the same time.
2. Every exam room and time has no additional use of tables or chairs. Use table or chair available in the room.
3. Examination shall not exceed time limit.
4. Period lengths are not violated. Example satisfaction of period related hard constrain (Exam A must be carried out after exam B).
5. Room related, example exam C must use room 12.

When the hard constraint is fulfilled then an objective function is used to minimize the total penalty. Penalty is obtained depending on how big the violation of soft

Table 1.
Dataset ITC 2007

Instances	Time Slot	Exams	Students	Rooms	Conflict Density
EXAM 1	54	607	7981	7	0.05
EXAM 2	40	870	12743	49	0.01
EXAM 3	36	934	16439	48	0.03
EXAM 4	21	273	5045	1	0.15
EXAM 5	42	1018	9253	3	0.01
EXAM 6	16	242	7909	8	0.06
EXAM 7	80	1096	14676	15	0.02
EXAM 8	80	598	7718	8	0.05

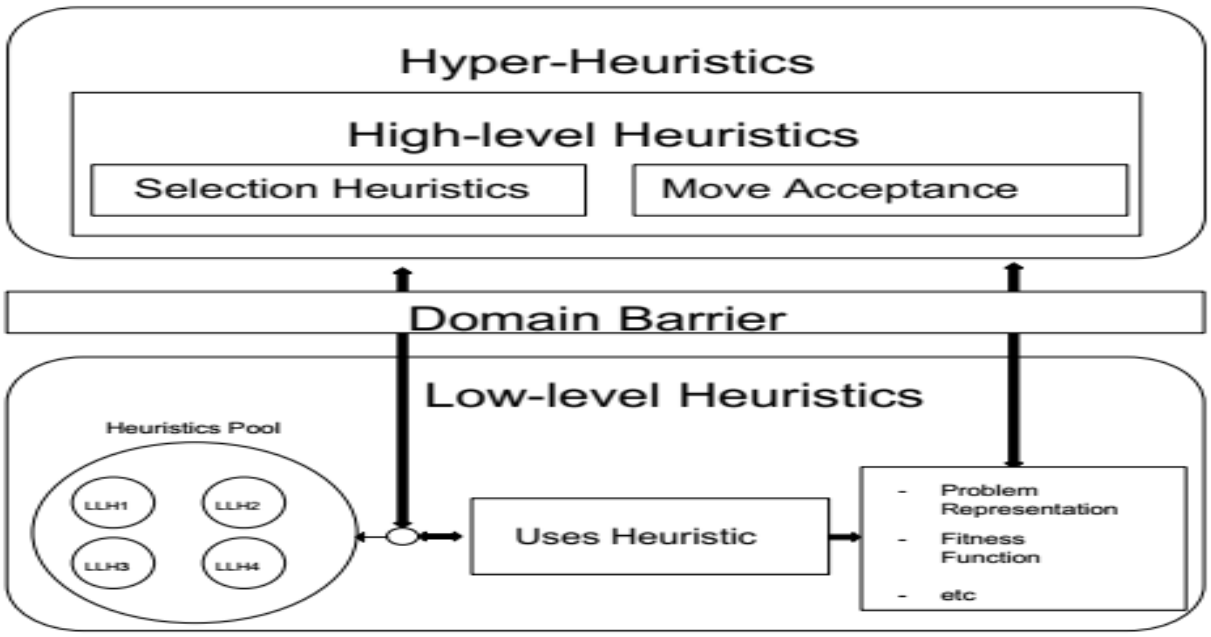

Figure 1. Framework Hyper-heuristic.

constraints. Each dataset has its own penalty weight [6]. The following are objective functions for calculating penalties that can be seen in the equation (1):

$$\sum_{s\in S}(w^{2R}C_s^{2R} + w^{2D}C_s^{2D} + w^{PS}C_s^{PS}) + w^{NMD}C^{NMD} + w^{FL}C^{FL} + C^P + C^R \quad (1)$$

Where:
C_s^{2R} : two exam in a row penalty for student s
C_s^{2D} : two exam in a day penalty for student s
C_s^{PS} : period spread penalty for student s
C^{NMD} : no mixed duration penalty
C^{FL}: front load penalty
C^P : soft period penalty
C^R: soft room penalty

B. Hyper Heuristik

Hyper-heuristic is an important research field in the area of optimization, it can be defined as search method that aims to solve optimization problems by selecting or producing heuristics [7]. There are no lunch free theorem specifies that for any algorithm, any increase in performance over one class of problems is offset by reduced performance in another class [8]. This theorem applies to metaheuristics, because to find a good solution, these method often need to be designed and turned to several problem domains or even just a single problem. Hyper-heuristic overcome this by increasing generality.

Hyper-heuristics divides heuristics into two types, high-level and low-level. Basically a high-level heuristic is responsible for choosing the low-level heuristic to be applied and which solution will be accepted to replace the less optimal solution. Low-level heuristics are responsible for solving problems or finding solutions space. High-level heuristics are independent, while low-level heuristics still depend on the problem.

The basic framework of hyper-heuristics can be seen in figure 1 [9]. The core part of hyper-heuristics framework is domain barrier. The section separates hyper-heuristics from the problem domain, so that hyper-heuristics can directly select information and which low-level heuristics can be chosen to solve what problem. Unlike the low-level algorithm, this approach focuses on searching for heuristics spaces rather than finding solutions. In addition, hyper-heuristic uses low-level to provide solutions to various problem rather than providing specific solutions to certain problem, can handle various examples of problems with different characteristics without requiring expert intervention.

C. Low Level Heuristics

One of the main components of the problem domain in

```
1.  Input: NLLH = number of low level heuristic, TL = Time Limit
2.  Construct initial solution S
3.  Calculate an initial cost C(S)
4.  Initial Counter i =0
5.  While not exceed TL do
6.      choose low level heuristic (llh) randomly
7.      apply llh on S to generate new solution S*
8.      calculate candidate cost function C(S*)
9.      if C(S*) > C(S)
10.         Then i = i +1
11.             if C(S*) < C(S²)
12.                 Then S² = S*
13.     if C(S*) < C(S)
14.         Then accept new solution S = S*
15.             i = 0
16.     if i > max counter
17.         Then accept new solution S = S²
18.             reset counter
19. End while
```

Figure 2. Pseudocode of extending Step Counting Hill Climbing.

Table 2.
Results Experiments

NO	INSTANCE	SR-SCHC-HH	SR-HC-HH	MUKLASON 2017		
				SR-GD-HH	RL-GD-HH	SA-GD-HH
1	EXAM 1	29889	31154	6579	7019	5809
2	EXAM 2	38900	39322	584	535	490
3	EXAM 3	100598	102826	11153	11592	10819
4	EXAM 4	NA	NA	13233	21992	14100
5	EXAM 5	128059	129837	3658	4610	3596
6	EXAM 6	54009	55106	26515	28130	26075
7	EXAM 7	63425	67890	5145	5151	5185
8	EXAM 8	118987	129388	9348	11405	9180

hyper-heuristics is the low-level heuristics. Hyper-heuristics control low-level heuristics in order to provide a solution rather than providing specific solution for a particular problem domain. In this research used six low-level for exam scheduling.

1. Swap exam, two exams are chosen randomly, then swap their time slot and room.
2. Change period, choose an exam randomly, assign to randomly chosen new time slot.
3. Change room, choose an exam randomly, assign to randomly chosen new room.
4. Swap room, two exams are chosen randomly, then swap their room.
5. Change period, choose an exam randomly, assign to randomly chosen new time slot and room.
6. Swap period, two exams are chosen randomly, then swap their time slot.

For each iteration, the low-level heuristics above will be chosen randomly using the Simple Random algorithm, which is known simple algorithm.

D. Move Acceptance

After the chosen low-level heuristics builds a new solution, then the solution will be decided whether or not it is accepted by the move acceptance. In this study Hill Climbing and Step Counting Hill Climbing chosen as move acceptance to decided solution is accepted or not.

The basics algorithm of Hill Climbing is to always accept a solution that is better than previous solution. In this study, researcher proposes to add iteration each time move acceptance rejects new solutions. The purpose of the iteration is to calculate the solution rejected by move acceptance, if iteration has reached the specified number, then the move acceptance must accept a worse solution in the next iteration. However, so that solution does not accept a solution that has vast differences, therefore second solution is made. Algorithm can be seen in Figure 2.

III. RESULT AND DISCUSSIONS

The results can be seen in Table 2. The experimental results

show that Simple Random – Step Count Hill Climbing is able to solve scheduling problems in the Examination Timetabling Problems domain in ITC 2007 dataset, but still requires adjustments so that results are more optimum and able to outperform all algorithm by previous researchers. Result from Simple Random – Step Counting Hill Climbing can exceed Simple Random – Hill Climbing because the ability of Simple Random – Step Counting Hill Climbing not easily stuck at local optima. There are parameter that can be studied further in this algorithm that is max iteration for algorithm to receive more trial from low-level heuristics to new solutions better than old solutions.

IV. CONCLUSION

Simple Random – Hill Climbing shows that algorithm can solve Examination Timetabling Problem in ITC 2007 dataset, but still not optimal .This study tests two algorithm on one of dataset commonly used by researchers, ITC 2007 dataset through hyper-heuristic approach. The researcher chose time limit based on the minimum time from ITC 2007 website, that is 300 seconds. Although the test is carried out with the same time limit of 300 seconds or five minutes and executed 10 times. The algorithm Simple Random – Step Counting Hill Climbing outperform Simple Random – Hill Climbing, although algorithm used by previous researchers have better results. Even the previous researchers also have not found optimal results for all instances in the dataset.

This research is to see how the performance of algorithm through hyper-heuristic approach and only tested on 8 instance in ITC 2007 datasets. For further research will be tested on Carter dataset (Toronto) and the results will be compared with some various Hill Climbing such as Late Acceptance Hill Climbing.

REFERENCES

[1] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, "A survey of search methodologies and automated system development for examination timetabling," J Sched, vol. 12, no. 1, pp. 55–89, Feb. 2009, doi: 10.1007/s10951-008-0077-5.

[2] E. Burke and W. Erben, Eds., Practice and Theory of Automated Timetabling III: Third International Conference, PATAT 2000 Konstanz, Germany, August 16-18, 2000 Selected Papers. Berlin Heidelberg: Springer-Verlag, 2001.

[3] A. Mukhlason, "Hyper-heuristics and fairness in examination timetabling problems," Jul. 2017.

[4] A. P. Dornelles, "A matheuristic approach for solving the high school timetabling problem," 2015.

[5] B. Mccollum, P. McMullan, A. Parkes, E. K. Burke, and R. Qu, "A New Model for Automated Examination Timetabling," Annals of Operations Research, vol. 194, pp. 291–315, Jan. 2012.

[6] S. Abdullah and M. Alzaqebah, "A hybrid self-adaptive bees algorithm for examination timetabling problems," Applied Soft Computing, vol. 13, no. 8, pp. 3608–3620, Aug. 2013, doi: 10.1016/j.asoc.2013.04.010.

[7]] E. Burke, M. R. Hyde, G. Kendall, G. Ochoa, and E. Özcan, "A classification of hyper-heuristic approaches," in International Series in Operations Research and Management Science, 2010, pp. 449–468.

[8] P. Cowling, G. Kendall, and E. Soubeiga, "A Hyperheuristic Approach to Scheduling a Sales Summit," in Practice and Theory of Automated Timetabling III, Berlin, Heidelberg, 2001, pp. 176–190, doi: 10.1007/3-540-44629-X_11.

[9] A. Ferreira, R. Gonçalves, and A. Pozo, "A Multi-Armed Bandit selection strategy for Hyper-heuristics," Jun. 2017, pp. 525–532, doi: 10.1109/CEC.2017.7969356.