HARMONY SEARCH-BASED HYPER-HEURISTIC FOR SCHEDULING PROBLEMS

KHAIRUL BIN ANWAR

UNIVERSITI SAINS MALAYSIA

2018

HARMONY SEARCH-BASED HYPER-HEURISTIC FOR SCHEDULING PROBLEMS

by

KHAIRUL BIN ANWAR

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

April 2018

ACKNOWLEDGEMENT

In the Name of Allah, the Most Compassionate, the Most Merciful. First and foremost, all praises are for Allah, the Almighty, on whom ultimately depend for sustenance and guidance.

My sincere thanks go to my supervisor Prof. Ahamad Tajudin Khader, who consistently steered me in the right direction whenever he thought I needed it or whenever I had a question about my thesis. His commitment and determination were truly exemplary. For all these, I am and always be grateful and indebted to him.

I would also like to thank my co-supervisor, Dr. Mohammed Azmi Al-Betar, for his continuous and unwavering support for me. His passionate motivation, enthusiasm and immense knowledge to help me writing of this thesis.

I also wish to thank the School of Computer Science, its leadership and the staff for providing me with an academic base that enabled me to take up and complete my thesis. Special thanks, tribute and appreciation to all my friends Mohammed A. M. Awadallah, Farhad Nadi, Asaju La'aro Bolaji, Basem O. Alijla, Lim T.Y. and many others whose names do not appear here who had contributed to the successful completion of this thesis. Finally, I must express my profound gratitude to my parents and to my family for their prayers, continuous encouragement, and unfailing support throughout the long, arduous years of my study. Without a doubt, the accomplishment of this thesis would have not been possible without them. Thank you.

May Allah bless us all

TABLE OF CONTENTS

CHAPTER 3 –METHODOLOGY

CHAPTER 4 –BASIC HARMONY SEARCH-BASED HYPER-HEURISTIC FOR SOLVING SCHEDULING PROBLEMS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABC	Artificial Bee Colony
B-HSHH	Basic Harmony Search-Based Hyper-Heuristic
COP	Combinatorial Optimization Problems
C-ETP	Capacitated Examination Timetabling Problem
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GD	Great Deluge
GP	Genetic Programming
HH	Hyper-heuristic
HMCR	Harmony Memory Consideration Rate
HML	Harmony Memory Length
HMS	Harmony Memory Sizes
HSA	Harmony Search Algorithm
HSHH	Harmony Search-Based Hyper-Heuristic
HSHH-MPA	Harmony Search-Based Hyper-Heuristic with Modified Pitch Adjustment
ITC2007	Second International Timetabling Competition 2007

INRC2010	First International Nurse Rostering Competition 2010
LD	Largest Degree
LE	Largest Enrolment
LLH	Low-level Heuristic
LWD	Largest Weighted Degree
NRP	Nurse Rostering Problem
PAR	Pitch Adjustment Rate
SD	Saturation Degree
SA	Simulated Annealing
TB	Tabu Search
UC-ETP	Uncapacitated Examination Timetabling Problem
VNS	Variable Neighbourhood Search

HIPER-HEURISTIK BERASASKAN GELINTARAN HARMONI UNTUK MASALAH PENJADUALAN

ABSTRAK

Masalah penjadualan ditemui dalam setiap bidang yang biasanya berkaitan dengan perkhidmatan seperti hospital, pengangkutan, dan juga institusi pendidikan. Masalah ini boleh menjadi sangat sukar apabila ia melibatkan sebilangan besar acara atau sumber yang perlu dijadualkan dan pelbagai kekangan yang perlu dipenuhi atau dipertimbangkan. Oleh itu, pelbagai pendekatan heuristik dan meta-heuristik telah diperkenalkan untuk menyelesaikan masalah ini. Walau bagaimanapun, sesetengah institusi mencari kaedah yang boleh menangani pelbagai masalah penjadualan tanpa perlu menggunakan banyak wang dan masa asalkan penyelesaian tersebut boleh diterima pakai. Atas sebab ini, pendekatan hiper-heuristik telah diperkenalkan. Motivasi menggunakan hiper-heuristik adalah untuk menghasilkan kaedah umum yang boleh digunakan untuk menyelesaikan masalah penjadualan yang berbeza. Oleh itu, kajian ini mencadangkan satu rangka kerja hiper-heuristik baru yang dinamakan sebagai Hiper-heuristik berasaskan Gelintaran Harmoni (HSHH). Idea asal adalah untuk melaksanakan satu urutan heuristik peringkat rendah pada sesuatu penyelesaian bagi menghasilkan penyelesaian yang lebih baik dan berkualiti. Oleh itu, tiga operator utama dalam algoritma gelintaran harmoni (HSA) iaitu *pertimbangan memori*, *pertimbangan rawak* dan *pelarasan nada* digabung dan digunakan sebagai heuristik peringkat tinggi untuk memilih dan menjana urutan heuristik peringkat rendah. Prestasi HSHH yang dicadangkan, diuji dengan tiga masalah penjadualan dimana dua masalah adalah dari penjadualan peperiksaan dan satu masalah dari penjadualan jururawat. Keputusan yang dihasilkan oleh HSHH dibandingkan dengan kaedah

pilihan perturbatif hyper-heuristik yang lain. Hasil ujikaji mendapati bahawa HSHH boleh mencapai keputusan yang setanding dan dalam beberapa kes HSHH dapat menghasilkan keputusan yang kompetitif dengan kaedah hyper-heuristik yang lain.

HARMONY SEARCH-BASED HYPER-HEURISTIC FOR SCHEDULING PROBLEMS

ABSTRACT

Scheduling problems are encountered in every field which are typically related to services such as in hospital, transportation, and also educational institutions. These problems can be extremely difficult when its involves a large number of events or resources to be scheduled and a wide variety of constraints which need to be satisfied or taken into consideration. Hence, many heuristic and meta-heuristic approaches have been introduced to solve these problems. However, some institutions look for a method that can deal with a wide range of scheduling problems without spending a lot of money and times as long the solutions produced are acceptable. For this reason, the hyper-heuristic approach have been introduced. The motivation of using hyper-heuristics is to produce a general method that can be used to solve different scheduling problems. Therefore, this research proposes a new alternative heuristic selection mechanism in a hyper-heuristic framework named Harmony Search-based Hyper-heuristic (HSHH). The original idea was to apply a sequence of low-level heuristics to a selected solution in order to produce good quality solutions to a given problem. Three main op-erators in harmony search algorithm: memory consideration, random consideration, and pitch adjustment were combined as high-level heuristics in order to select and generate a sequence of improvements low-level heuristics. To demonstrate the effectiveness of the method, the proposed method was tested with three different scheduling problems (two examination timetabling problems and one nurse rostering problem) taken from the real world. The results produced by the proposed methods were compared with those

of other selection perturbative hyper-heuristic methods working on the same datasets. Experimentally, the HSHH approach had achieved comparable re-sults and in several instances was able to produce competitive results.

CHAPTER 1

INTRODUCTION

1.1 Background

Scheduling is the arrangement of a set of objects (e.g., events or resources) to a set of time or spaces subject to some goal and constraints (Wren, 1996). The main purpose of producing a schedule is to organize people's lives, activities, and work without being interrupted while still being able to perform them timely. Scheduling problems are encountered in every field or domain which are typically related to services such as in hospital or clinics, transportation, and also educational institutions. These problems can be extremely difficult when its involves a large number of events or resources (can be hundreds or thousands) to be scheduled and a wide variety of constraints which need to be satisfied or taken into consideration. Therefore, scheduling problems can be classified as hard and complex problems (Lewis, 2008; Noronha and Sarma, 1991).

Over the years, various techniques and approaches have been studied and introduced in order to solve scheduling problems. All the techniques and approaches can be classified into several categories which are heuristics, meta-heuristics, and hyper-heuristics. The classification of scheduling techniques and approaches are shown in Figure 1.1.

Heuristic is a *'rule of thumbs'* algorithm on finding a good quality solution in a reasonable time-frame for the given problem. Although they cannot guarantee to

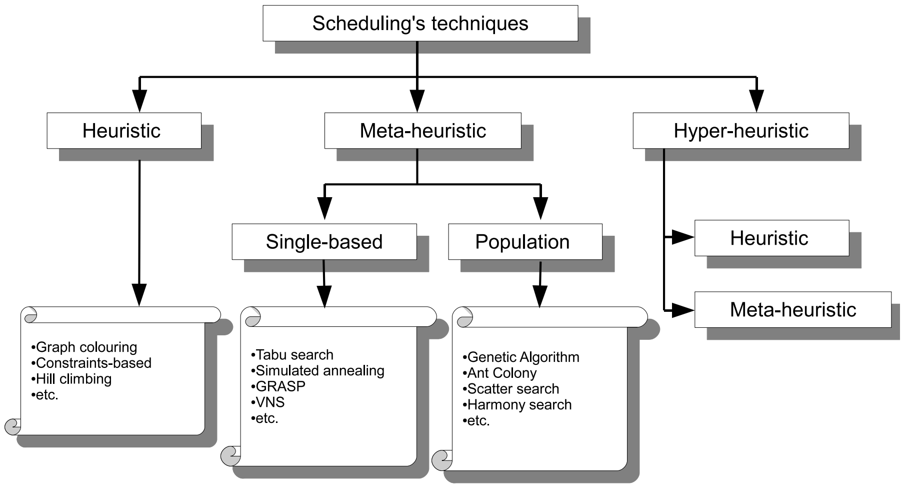

Figure 1.1: Classification of scheduling problems techniques

provide an optimal solution, a number of the studies and researches have shown that these algorithms are able to find near-optimal solution in a fast manner.

Meta-heuristic is a new type of heuristic used to explore the search space more efficiently and effectively. Most meta-heuristic algorithms are inspired from existing processes of nature's behaviour. For example, an artificial bee colony (Bolaji et al., 2012; Alzaqebah and Abdullah, 2011) had been designed based on the bees' behaviour in finding food. Generally, the strategy of meta-heuristic techniques is to balance the intensification and diversification process during the search.

Hyper-heuristic (HH) was introduced by Cowling et al. (2001). The term *'hyper-heuristic'*, basically describes the *heuristics to choose heuristics* in the context of combinatorial optimisation problems. Even though the term *'hyper-heuristic'* is new, the idea has been applied since the 1960s (Ross et al., 2002). Generally, hyper-heuristic is known as a heuristic(s) or meta-heuristic(s) which selects an appropriate low-level heuristics for a particular problem in order to solve it. In hyper-heuristic framework the low-level heuristics are referred as an algorithm that will be used to operates in the problem domain in order to construct or improve the complete solution.

Hyper-heuristic approach is more towards a general method that can solve not only a single problem but many problems. According to the initial definitions, hyper-heuristic is a method that operates on the search space of heuristics or heuristic component rather than directly on the search space of the solution (Burke et al., 2003). This is the main difference between hyper-heuristic and meta-heuristic. Figure 1.2 illustrates the aforementioned general concept of hyper-heuristic.

In addition, hyper-heuristic could be regarded as an *'off-the-peg'* method compared to a *'made-to-measure'* bespoke meta-heuristic (Burke et al., 2009). In other words, hyper-heuristic is a standard method that can be used to solve problems and not a specific method used to solve a specific problem domain (e.g., meta-heuristic). Therefore, the aim of hyper-heuristic method is not necessarily to *'beat'* the best methods that have been introduced before but to produce a method that is *"cheap enough - good enough - soon-enough"* across a wide-range of problems and domains (Burke et al.,2003).

1.2 Problem Statement

As mentioned before, hyper-heuristic method is more towards generality of which focus is to resolve not only single combinatorial optimization problem (COP) but many. Since its introduction in 2000, innumerable studies have been conducted in order to investigate the potential of this approach. Based on the extant literature, the hyper-heuristic approaches are able to generate a good and even better solution in solving COPs including the scheduling problems.

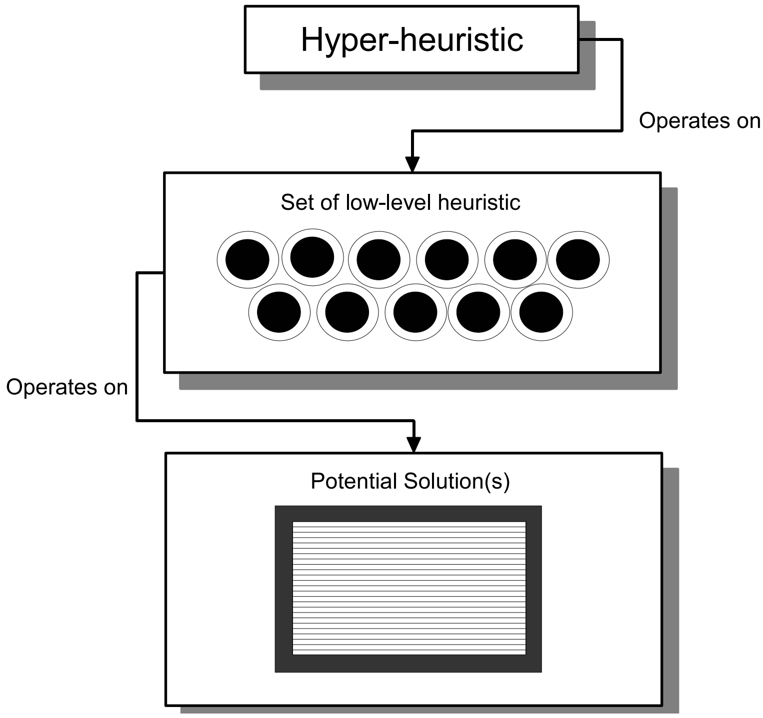

Figure 1.2: Hyper-heuristic general concept

Various types of hyper-heuristics have been introduced such as random, greedy, tabu search, choice function, reinforcement learning, scatter-search, and many others as discussed in Burke et al. 2003, Burke et al. 2009, Pillay (2014). Some of these hyper-heuristic methods are based on random selection and some are based on memory. The hyper-heuristic based on random selection (e.g., simple random) will choose the heuristic from the heuristic set randomly. On the other hand, hyper-heuristics which are based on the memory (e.g., tabu search, genetic algorithm, scatter search and etc.) will choose the heuristic based on their previous performances. In addition, some of these hyper-heuristic methods are combine with learning mechanism (e.g., random decent, reinforcement learning, and etc.) in order to increase the capability of the methods during the search process.

The combination of different ways of heuristic selection can produce different effects in a hyper-heuristic development. For example, in the basic hyper-heuristic framework, a simple random selection is used in order to choose a heuristic from the heuristics set. Even though random can be an effective method to avoid a local optima, in certain conditions it can also lead to the worst outcome. Therefore, some other methods can be inserted in order to guide this randomness. By combining different ways of heuristic selection, it might increase the capability of the hyper-heuristic approach to select the most appropriate set of low-level heuristics.

Besides, studies on the hyper-heuristic approach are done towards applying a heuristics combination instead of using a single heuristic at a time. This is because applying different combinations of heuristics (i.e., heuristic sequence) to different solutions can generate different effects to the solutions. This has been proven by several works related to this idea such as in Cowling et al. (2002); Han et al. (2002); Han and Kendall (2003); Pillay and Banzhaf (2009); Pillay (2010a).

The Harmony Search algorithm (HSA), basically permits the combination of different operators for which in this algorithm three operators are commonly used namely *memory consideration*, *random consideration* and *pitch adjustment*. All of these operators in HSA will be triggered based on parameter settings (rate of probability). Moreover, HSA also allows the implementation of heuristics combination in it framework. Even though HSA has been tailored successfully for numerous optimization problems including scheduling problems (Mohd Alia and Mandava, 2011; Manjarres et al., 2013), not much research have been done to investigate the uses of HSA for selecting and generating the heuristic combination within hyper-heuristic framework.

Therefore, the motivation of this thesis is to further investigate whether the use of HSA as a high level heuristic in a hyper-heuristic framework could be beneficial in solving scheduling problems. Pursuant to that, the main reason of choosing the scheduling problems such as examination timetabling problems and nurse rostering problem lies in the fact that although these problems are not critical (in most situation) but it is important to produce a good schedule in order to organize people's lives, activities, and work without being interrupted. Furthermore, for some organizations, it is imperative that they look for a method that can be used for solving various problems without having to spend a lot of money and times as long as the solution produced are acceptable quality (Burke and Soubeiga, 2003).

1.3 Research Questions

In order to clarify the purpose of the research, several research questions are highlighted based on the problem statement such as:

- How is the performance of harmony search algorithm when it is applied as a high-level heuristic in hyper-heuristic framework in term of: i) How harmony search algorithm select and generate a heuristic sequences by using its operators? ii) What is the suitable length for heuristic sequence in the suggested hyper-heuristic algorithm?
- Original harmony search algorithm use random selection when it select value in its memory. How the performance of harmony search algorithm as heuristic selection if different selection mechanisms such as learning mechanisms are used?
- The aim of hyper-heuristic is to produce with a general method. How the suggested HH algorithm performance when it is tested with different scheduling problems.

1.4 Research Objectives

The aim of this research is to present an alternative hyper-heuristic approach that combines different selection mechanism as high level heuristic in order to select and generate a heuristics combination for solving scheduling problems. The main objectives of the research are:

1. To introduce an alternative hyper-heuristic method by adapting the harmony search algorithm within a hyper-heuristics framework.
2. To enhance the performance of harmony search hyper-heuristic by employing a different selection mechanism to control the selection of low-level heuristic.
3. To test the generality of the proposed hyper-heuristic framework using three different scheduling problems.

1.5 Research Scope

This research aims to produce a general framework called hyper-heuristic for solving scheduling problems. This framework is about building a method that is able to select and apply an appropriate low-level heuristic at each decision point. Through this ability, hyper-heuristic is easily reused to different scheduling problems by having only to change the set of heuristic and the evaluation function (depending on the problem domains). Hyper-heuristic can be classified under many classifications and categories such as selection constructive, selection purturbative, generation constructive and generation (Burke et al., 2010). In this research, the selection perturbative hyper-heuristic will be studied.

In order to test the general applicability of the proposed hyper-heuristic, three different datasets will be considered in this research i.e., Carter dataset Carter et al. (1996), the Second International Timetabling Competition in 2007 (or ITC2007) dataset (McCollum et al., 2007), and the International Nurse Rostering Competition 2010 (or INRC2010) dataset (Haspeslagh et al., 2014). The Carter's dataset (Carter et al., 1996) is an Examination Timetabling Problem (ETP) which can be categorized as Un-Capacitated Examination Timetabling Problem (Uc-ETP). In this problem, the capacity of the room is not considered. However, the ITC2007 dataset for the examination track is used in order to test the proposed method with different category of ETP for which is, in this category the capacity of rooms will be putted into consideration during the construction of the solution. The third dataset is from a different problem domain which is a nurse rostering problem (NRP). The NRP involved producing a periodic duty roster for nursing staff subject to a set of constraints. For this research, the first standard dataset for NRP established by the organizers of the International Nurse Rostering Competition in 2010 will be used.

1.6 Research Methodology

This section provides a brief discussion on the methodology used for this research. It involves three stages. The methodology employed is modelled as in Figure 1.3.

The first stage involves studying and exploring the hyper-heuristic framework. During this research a simple hyper-heuristic framework was developed. Apart from that, the problem domains (for scheduling problem) that would be considered was also determined in this stage. In this thesis, three datasets from two different problem domains

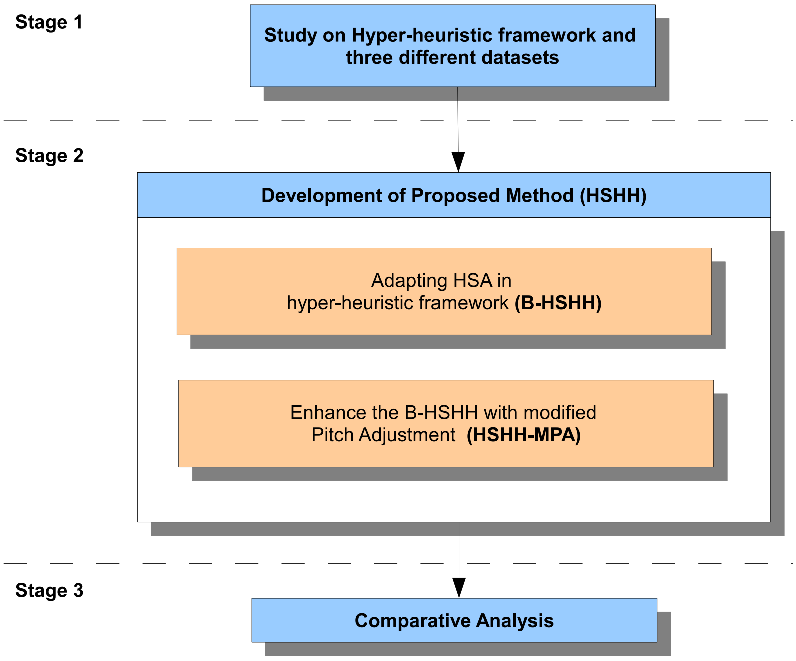

Figure 1.3: Research methodology

were used in order to test the efficiency of the proposed method. As aforementioned, the first dataset is a Carter dataset which is an examination timetabling problem. In the same problem domain, the proposed method were also tested with the ITC2007 dataset (for examination timetabling track). The third dataset is from a different problem domain which is a nurse rostering problem. A detail discussion for each dataset used in this research will be presented in Chapter 3. Subsequently, in this step, potential low-level heuristics for each problem domain were determined and implemented.

In the next stage (second stage), the proposed method known as the Harmony Search based hyper-heuristic (HSHH) will be developed. At the first step, only two main operators of HSA i.e., memory consideration and random consideration were used to select an appropriate low-level heuristics. The basic adaptation of HSA which is named Basic-HSHH (or B-HSHH) were tested with the three scheduling datasets as mentioned before, in order to evaluate the performance of the proposed method. The next step includes enhancing the B-HSHH by applying different selection methods through the pitch adjustment operator in HSHH. This enhanced method known as HSHH with Modified Pitch Adjustment (or HSHH-MPA) were also tested with the same datasets in order to evaluate the performance of the method.

In the last stage, the results of the proposed HSHH methods (i.e., B-SHAH and HSHH-MPA) were compared with each other. Furthermore, the best results achieved by the HSHH were compared with results obtained by other hyper-heuristic methods that are used on the same datasets.

1.7 Research Contributions

The contributions of this research are:

1. Adaptation of the Harmony Search algorithm (HSA) in a hyper-heuristic framework named as Basic Harmony Search Hyper-Heuristic (B-HSHH):
   i. Using two main operators: memory consideration and random consideration in HSA as a heuristic selection mechanism in order to select and generate a sequence of low-level heuristics.
   ii. Combining two different types of harmony memory: heuristics harmony memory and solutions harmony memory. In the original HSA, only one harmony memory will be constructed and applied, meanwhile in the B-HSHH, two different harmony memory (i.e., HHM and SHM) will be applied during the search process.

2. Modification of the Harmony Search Hyper-heuristic which incorporates the B-HSHH with other heuristic selection techniques (HSHH-MPA) in selecting a low-level heuristics.

    i. Adding pitch adjustment operator.

    ii. Implementing three different types of heuristic selection.

3. Adaptation of the Harmony Search Hyper-Heuristic approach for solving two types of examinations timetabling problem (i.e., Carter and ITC2007 dataset) and nurse rostering problem (i.e., INRC2010).

1.8 Overview of Thesis

This thesis includes seven chapters organized as follow:

Chapter 2 presents a detailed discussion of the main components for this thesis. It divided into two parts. In the first part of the chapter, a hyper-heuristic framework and it classification will be explained. Subsequently, in the second part of the chapter, the harmony search algorithm and the steps in this algorithm are described.

Chapter 3 is the introduction to the proposed methods. The hyper-heuristic framework proposed in this study will be elaborated and discussed in detail together with the problem domain used in this thesis.

Chapter 4, and 5 present all the steps in the proposed methods that will be elaborated and discussed in detail. Furthermore, all experiments and results with detailed analysis done in order to study the performance of the proposed hyper-heuristic methods are also reported in these chapters.

Chapter 6 provides a comparison analysis among the proposed methods together with other hyper-heuristic methods using the same datasets studied in this thesis.

Chapter 7 presents the summary of the research as well as some directions for future research that will be mentioned in this last chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides a detailed explanation on the hyper-heuristic framework. The structure of this chapter is based on the classification of the hyper-heuristic (i.e., selection and generation). Subsequently, the chapter continues with a brief discussion on the selection mechanisms that will be used for this thesis (i.e., harmony search algorithm).

2.2 Hyper-Heuristic

The initial hyper-heuristic definitions describe it as a method which is performed at a higher level to select an appropriate heuristics for solving a particular problem. The definition of hyper-heuristic has been extended to be more sophisticated and more general where the method is not only used for selecting appropriate low-level heuristics but also for automatically generating low-level heuristics (Burke et al., 2010). Below is the new definition for hyper-heuristic defined by Edmund Burke.

'A hyper-heuristic is an automated methodology for selecting or generating heuristics to solve hard computational search problems.'

Burke et al. (2010)

Based on these definitions, hyper-heuristic can be divided into two categories called *Selection hyper-heuristic* and *Generation hyper-heuristic*. Selection hyper-heuristic can be defined as a method for choosing or selecting existing low-level heuristics while generation hyper-heuristic refers to a method for generating a new low level-heuristic from the component of existing heuristics (Burke et al., 2009; 2010; 2013).

Furthermore, these two categories of hyper-heuristics (i.e. selection and generation) are classified based on the nature of the low-level heuristics. As mentioned before, the low-level heuristics are referred as a simple algorithm that will be used to operates in the problem domain in order to construct or improve the complete solution. These low-level heuristics natures can be classified into two different classes namely *constructive* and *perturbative*. *Constructive low-level heuristics* are used to construct a complete solution gradually from an empty solution. Basically, these heuristics will determine the next events to be scheduled according to the consideration factor (or difficulty). Constructive heuristics are commonly used in the initial phase of sequential methods during the construction of an initial solution. In most cases, the solution produced by these heuristics are feasible. Nevertheless, *perturbative low-level heuristics* are also used to improve the initial solutions. It can be a simple heuristic strategy such as move and swap or shift. Each low-level heuristic can be considered as improvement heuristics that return a move and a change in the penalty function. The categorisations of these hyper-heuristics are shown in Figure 2.1.

Apart from the nature of the low-level heuristics search space, hyper-heuristic has also been classified according to the feedback mechanisms (see Figure 2.2). The three categories of feedback mechanisms in hyper-heuristic are *online learning*, *of-*

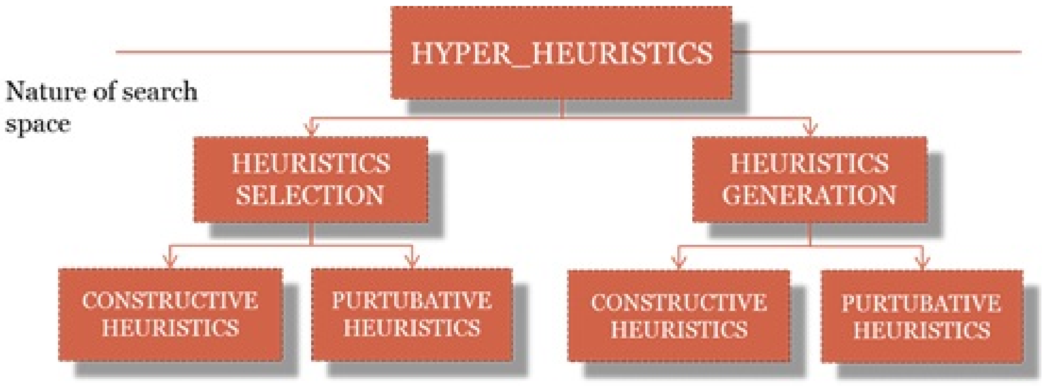

Figure 2.1: Hyper-heuristic classification according to the nature of the heuristic search space

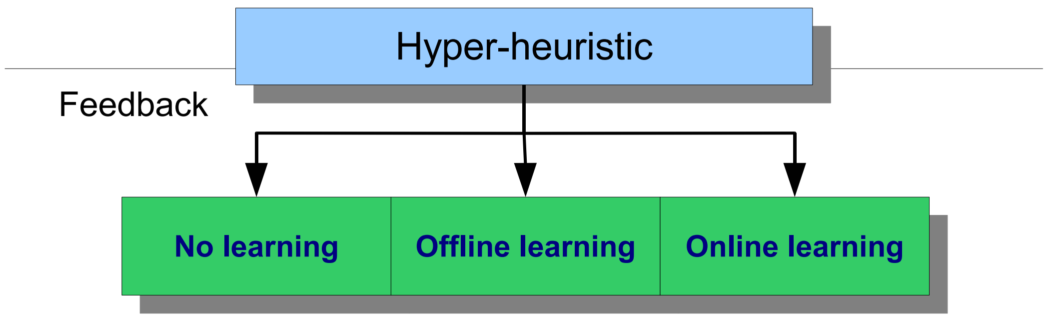

Figure 2.2: The two dimensions of hyper-heuristic classifications i.e. source of feedback and nature of the heuristics search space (Burke et al., 2010)

fline learning and *no learning*. The hyper-heuristics are categorized as online learning, when the learning process takes place during the search process. Tabu search (Burke and Soubeiga, 2003), reinforcement learning (Özcan et al., 2012) and choice function (Cowling et al., 2001) can be considered as online learning. However, when the learning process occurs before the actual search process start, the hyper-heuristics are categorized as offline learning. Case based reasoning (Burke et al., 2006) and learning classifier systems (Ross et al., 2002) are works in an offline manner. In addition, some hyper-heuristic components without any learning devices are also available such as the simple random selection mechanism (Misir et al., 2010). Even though this method (i.e., simple random) is a simple and naive heuristic, in certain cases it can be an effective selection method. Table 2.1 summarizes the hyper-heuristic methods that have been developed before. This table provides the classification of each hyper-heuristic based on the types and feedback mechanism.

Hyper-heuristic has been used in many different scheduling problems such as personnel scheduling problem (Cowling et al., 2001), course timetabling problems (Burke et al., 2003a), examination timetabling problem (Burke et al., 2012, Kendall and Hussin, 2005, Pillay, 2010, Qu and Burke, 2005, Sabar and Ayob, 2009), nurse rostering problem (Bilgin et al., 2012, Burke et al., 2003b). Besides scheduling problems, hyper-heuristics have also been used in other optimisation problems such as bin packing (Burke et al., 2006; 2007, Ross et al., 2002), vehicle routing (Garrido and Castro, 2009; Mlejnek and Kubalik, 2013), sequencing problems (Ayob and Kendall, 2003), satisfiability (SAT) problems (Barder-El-Den et al., 2007; 2008) and many others.

Table 2.1: The classification of Hyper-Heuristic methods based on types and feedback mechanism

Hyper-heuristic Methods	**Type**	**References**
Simple Random	Selection	Cowling et al. (2001), Misir et al. (2010), Bilgin et al. (2012), Ayob and Kendall (2003), Kendall and Mohamad (2004)
Greedy	Selection	Bilgin et al. (2010), Cowling et al. (2001)
Choice Function	Selection	Cowling et al. (2001), Bilgin et al. (2012)
Case-based reasoning	Selection	Burke, MacCarthy, Petrovic and Qu (2003), Burke, Petrovic and Qu (2006)
Reinforcement Learning	Selection	Özcan et al. (2008)
Genetic Algorithm	Selection	Cowling et al. (2002),Han and Kendall (2003)
Tabu Search	Selection	Burke and Soubeiga (2003), Burke, McCollum, Meisels, Petrovic and Qu (2007), Kendall and Hussin (2005a),Kendall and Hussin (2005b)
Memetic Hybridisation	Selection	Bilgin et al. (2007), Sabar et al. (2012), Qu, Burke and McCollum (2009), Qu and Burke (2009)
Genetic Programming	Selection Generation	Pillay (2008) Pillay and Banzhaf (2009), Nguyen et al. (2011), Bader-El-Den et al. (2009)
Scatter Search	Selection	Sabar and Ayob (2009)
Adaptive Heuristic	Generation	Burke and Newall (2004), Rahman et al. (2009)
Grammatical Evolution	Generation	Sabar et al. (2013)

2.2.1 Selection hyper-heuristic framework

A traditional framework of selection hyper-heuristic consists of two main mechanisms namely *heuristic selection mechanism* and *move acceptance mechanism*. The hyper-heuristic search process starts with the selection mechanism for choosing the appropriate low-level heuristic for generating a new (partial or complete) solution in the current optimization step. Afterwards, move acceptance mechanism will decide on the acceptability of the new solution based on some problem-independent information (e.g., fitness function, etc.). Figure 2.3 illustrates the traditional framework of selection hyper-heuristic.

There are many works that are related to heuristic selection compared to heuristic generation. Heuristic selection hyper-heuristic can be single-based (e.g., tabu search, choice function, reinforcement learning, etc.) or population-based (e.g., genetic algorithm, ant colony algorithm, etc.).

As classified by Burke et al. (2010), the heuristic selection hyper-heuristic can be based on constructive low-level heuristics or perturbative low-level heuristics. Constructive hyper-heuristic will build a complete solution gradually from scratch by using constructive heuristic such as Largest Degree (LD), Largest Enrolment (LE), Largest Weighted Degree (LWD), Saturation Degree (SD) and others. On the other hand, perturbative hyper-heuristic will start with an initial solution, and then local search or neighbourhood search will be used to improve the solution.

Selection constructive hyper-heuristic is an approach that usually employs constructive heuristics (e.g., Largest Degree, Saturation Degree, Large Enrolment, etc.)

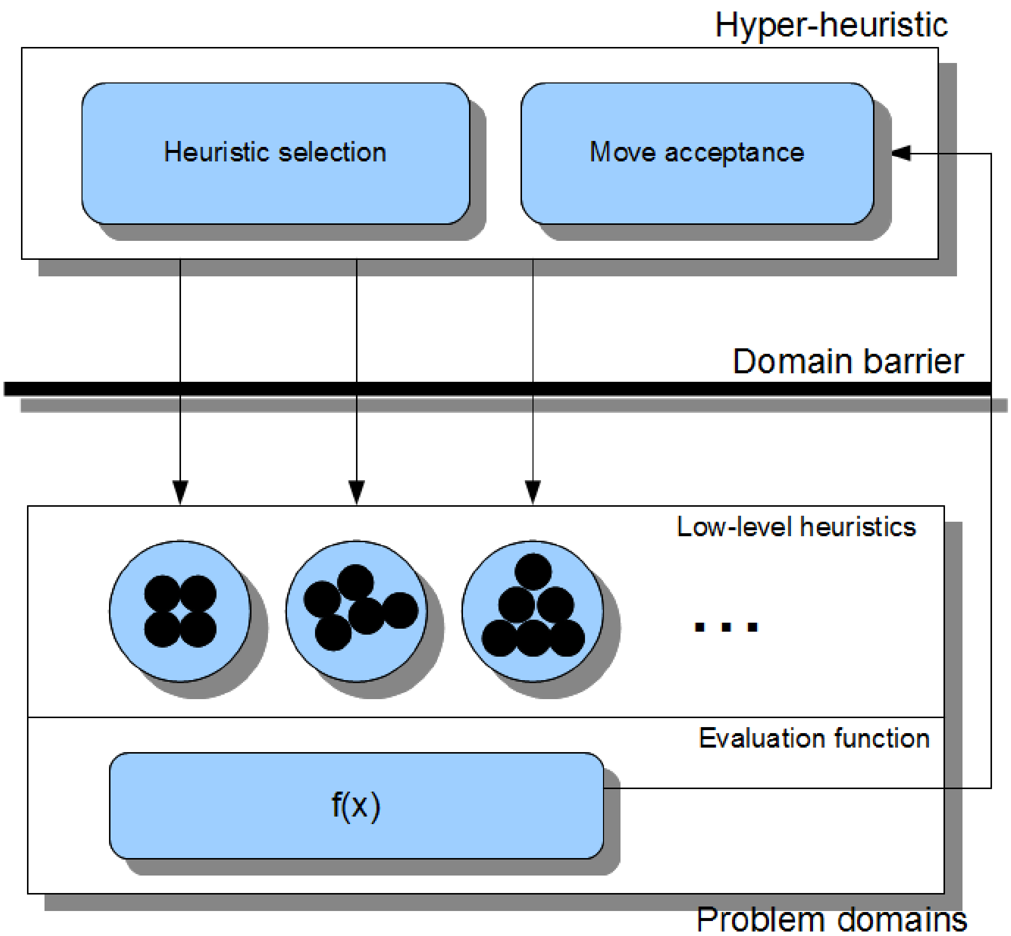

Figure 2.3: Traditional framework of selection hyper-heuristic Burke, Kendall, Newall, Hart, Ross and Schulenburg (2003)

as a low-level heuristic. In order to select or explore a search space of heuristic, several heuristics (or meta-heuristics) have been used. For instance, Pillay (2008) used genetic programming (GP) for the evolution on hyper-heuristics framework to solve un-capacitated examination timetabling problems. In their study, the GP framework has a sequence of constructive low-level heuristics and specifies their order in genetic solution (e.g., *lssll* where *l* and *s* represent the largest degree and saturation degree construction heuristics respectively). Note that each genetic solution contains a sequence of low-level heuristic ordered to construct a single timetable.

Moreover, Sabar et al. (2012), investigated a new graph colouring constructive hyper-heuristic for solving examination timetabling problems. This approach utilized the hierarchical hybridizations of four low-level graph colouring heuristics such as largest degree, saturation degree, largest coloured degree and largest enrolment. These were hybridized to produce four ordered lists. In each list, the difficulty index of scheduling the first exam was calculated by considering its order in all lists to obtain a combined evaluation of its difficulty. The most difficult exam to be scheduled was scheduled first (minimum difficulty index will be scheduled first). To improve the effectiveness of timeslot selection, a roulette wheel selection mechanism was included in the algorithm to probabilistically select an appropriate timeslot for the chosen exam.

On the other hand, selection perturbative hyper-heuristics is an approach that selects a low-level perturbative heuristics. Perturbative refers to a local search heuristic or/and neighbourhood heuristic (e.g., move, swap, etc.). Basically, these approaches try to select the most appropriate low-level heuristic (e.g., perturbative heuristics) in order to improve a current complete solution. As in the case with selection constructive hyper-heuristics described above, these hyper-heuristics also employ heuristic techniques (or meta-heuristics) to explore the search space of low-level perturbation heuristics. Burke et al., 2003 used the tabu search as a high-level heuristic selection mechanism in a hyper-heuristic framework in order to rank the low-level heuristic. In their approach, the ranking of the low-level heuristics was based on a reinforcement learning mechanism that considers the change in the candidate solution. The proposed method had been successfully applied to course timetabling and nurse rostering problems with a comparative result.

Kendall and Hussin (2005a), developed a tabu search-based hyper-heuristic. The process begins by generating an initial solution based on the saturation degree or largest degree. The neighbourhood of the initial solution is then examined in order to improve the generated timetable. Four low-level heuristics (move, swap, select and schedule, and remove) were utilized within a tabu search framework. The heuristics that produces the best improvement in the current iteration is to be used in the following iterations. The process is repeated until a time limit is reached or no improvements are achieved for a predefined number of iterations. The same approach is also applied in the different examination timetabling problem as discussed in Kendall and Hussin (2005b).

Cowling et al. (2002) investigated the implementation of genetic algorithm as a high level heuristic called hyper-GA for scheduling geographically distributed training staff and courses. The goal of hyper-GA was to evolve a good-quality heuristic to find good-quality solution by applying a suitable ordering from a set of low-level heuristic. The hyper-GA approach was extended in Han et al. (2002), where in this study, adaptive length chromosomes was investigated. This approach was named ALChyper-GA. The aim of this study was to provide a more effective heuristic list. In addition, the most optimal length for chromosomes was to be determined. The results demonstrated that ALChyper-GA was better than hyper-GA for most of the problem instances. A year later, Han and Kendall (2003) enhanced the work (i.e.,ALChyper-GA) by adding the tabu method. The aim of adding a tabu list to the ALChyper-GA is to indicate the efficiency of each gene (i.e.,low-level heuristic) within the chromosome. The experimental results showed that hyper-TGA produces better solution than the previous two versions of hyper-GA (i.e.,hyper-GA and ALChyper-GA).

Qu and Burke (2005a), implemented a hybrid variable neighbourhood search (VNS) within the graph based hyper-heuristic framework. In their work, two sets of variable neighbourhood structures were designed namely single flipped and block flipped structures. In the single flipped structure, the algorithm randomly changes two, three, four or five heuristics in the heuristic list. For block flipped, the algorithm randomly changes two, three, four or five consecutive heuristics in the heuristic list. These methods have been evaluated using the exam timetabling problem benchmark dataset. Through the experiment, the author concludes that the method employed for the high level search is not crucial within the graph based hyper-heuristic approach.

In Ersoy et al. (2007) a hyper-heuristic is embedded in a memetic algorithm used to solve the examination timetabling problem. The hyper-heuristic is used to select one of three hill-climbers to be used by the memetic algorithm. In their study, the hyper-heuristics using either a choice function and great deluge (CF-GD) or simple random combined with improving and equal (SR-IE), were found to perform well in Carter benchmark problem.

Sabar and Ayob (2009), proposed a scatter search based hyper-heuristic (SSHH) approach for solving examination timetabling problems. The scatter search is used at a high level of abstraction which intelligently evolves a sequence of low level heuristics to use for a given problem. Each low level heuristic represents a single neighbourhood structure.

Swan et al. (2012), use an evolutionary-based hyper-heuristic to solve examination timetabling problem. In this hyper-heuristic framework, a sequence of add and delete operations were used in order to improve the existing timetable. These operators are represented by using binary numbers which zero for delete and one for add. The hyper-heuristic performed well on the ITC2007 benchmark dataset.

Burke et al. (2014) employ a hyper-heuristic to improve the quality of an initial feasible solution created using the largest degree construction heuristic. The hyper-heuristic uses four low-level perturbative heuristics, namely, move exam, swap exam, Kempe chain move and swap timeslot. All four heuristics aim at producing the least penalty timetable. An adaptive component is built into the hyper-heuristic to perform the hybridization of these two heuristics. The hyper-heuristic was used to find solutions to problems from the Carter benchmark set and the benchmark set for the second international timetabling competition. Preliminary studies indicated that Kempe chain in combination with swap timeslot performed the best over problems of differing characteristics.

There are also other work that have used harmony search algorithm as hyper-heuristic as discussed in Dempster and Drake (2016). In their work, the harmony search was used to select a mixture of continuous and discrete variables forming the components of a Memetic Algorithm. Two hyper-heuristic framework using harmony search were investigated: i) *single-point based search* which maintains a single solution, and ii) *population-based search* which co-evolves a set of solutions to a problem alongside a set of harmony memory. In this method, the harmony memory consisted of three heuristics such as crossover, mutation, and hill climbing, together with the parameters for the heuristics. The parameters were used to indicate the values of the *intensity of mutation* and *depth of search*. They tested the performance of their frameworks on HyFLex and the results show that population-based approach is able to achieve significantly better results than the single-point approach.

Another major influence in a hyper-heuristic's performance is the move acceptance criterion. The purpose of the acceptance mechanism in the hyper-heuristic framework is to determines the diversification characteristics. There are several mechanisms which have been used as move acceptance in the hyper-heuristic framework. For example a basic and simple mechanism such as Only Improve (OI). Besides that there are other intelligent and complicated mechanisms which have been applied such as simulated annealing, great deluge, monte carlo and others.

Bilgin et al. (2012) presents one general hyper-heuristic approach for addressing two scheduling problems in the health care domain: the patient admission scheduling problem and the nurse rostering problem. In this study, they tested thirty six hyper-heuristic variations, a combination of three types of selection mechanism such as simple random, choice function and dynamic heuristic set strategy and four different move