GROUP DECISION MAKING FOR MOVE ACCEPTANCE IN HYPERHEURISTICS

by

Mustafa Mısır

Submitted to the Institute of Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Computer Engineering

Yeditepe University
2008

GROUP DECISION MAKING FOR MOVE ACCEPTANCE IN HYPERHEURISTICS

APPROVED BY:

Assist.Prof.Dr. Ender Özcan ……………………
(Thesis Supervisor)

Assist.Prof.Dr. Emin Erkan Korkmaz ....…………………

Assoc.Prof.Dr. Kudret Yurtseven ……………………

DATE OF APPROVAL: ……………………

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Assist.Prof.Dr. Ender Özcan for his valuable guidance, encouragement and undivided attention during my study. I was allowed to use many facilities in the Computer Engineering Department, so, I would like to present my special gratitude to the administration, secretariat and technical staff for their support.

During my stay at Yeditepe University, I had an opportunity to interact closely with the members of Artificial Intelligence (ART+I) Research Laboratory and among them, I would like to particularly thank Assist.Prof.Dr. Emin Erkan Korkmaz for the discussions that we had. Many thanks go to a previous ART+I member, Burak Bilgin for his contributions on hyperheuristics.

Lastly, thanks to Dr. David H. Wolpert and Dr. William G. MacReady for their simple but amazing "*No Free Lunch*" theorem concerning search and optimization which made all this study possible.

ABSTRACT

GROUP DECISION MAKING FOR MOVE ACCEPTANCE IN HYPERHEURISTICS

A hyperheuristic is a heuristic that performs a search over a set of low-level heuristics for solving difficult problems. A perturbative hyperheuristic consists of two successive stages. In the first stage, the most appropriate perturbative low-level heuristic is selected and applied to a candidate solution, then, a decision is made whether to accept or reject the new solution. In this study, seven heuristic selection mechanisms are combined with four group decision making strategies for move acceptance to investigate twenty-eight hyperheuristics over well-known benchmark function optimization and examination timetabling problems. Experimental results on these problems show that the group decision making move acceptance strategies might improve the performance of hyperheuristics significantly.

ÖZET

ÜSTBULUŞSALLARDA HAREKET KABULÜ İÇİN GRUP KARAR VERME

Bir üstbuluşsal, zor problemleri çözmek için bir düşük-seviyeli buluşsallar kümesi üzerinde arama yapan bir buluşsaldır. Geliştirici bir üstbuluşsal ikiardışık aşama içermektedir. İlk aşamada, en uygun, geliştirici düşük seviyeli buluşsal seçilir ve bir aday çözüme uygulanır, daha sonra, yeni çözümü kabul etmek ya da reddetmek için bir karar verilir. Bu çalışmada, yedi buluşsal seçim mekanizması, dört hareket kabul için grup karar verme stratejisi ile, iyi bilinen matematiksel denektaşı fonksiyonları ve sınav zaman çizelgeleme problemleri üzerinde yirmi sekiz üstbuluşsalı incelemek için birleştirilmiştir. Bu problemler üzerindeki eneysel sonuçlar, grup karar verme hareket kabul stratejilerinin, üstbuluşsalların başarımını önemli ölçüde geliştirebileceğini göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

ACO	Ant Colony Optimization
ALChyper-GA	Adaptive Length Chromosome Genetic Algorithm based Hyperheuristic
AM	All moves accepted
BPP	Bin Packing Problem
CBR	Case-Based Reasoning
CF	Choice Function Heuristic Selection
CSP	Constraint Satisfaction Problem
DBHC	Davis' Bit Hill-Climbing Operator
DIMM	Dimensional Mutation Operator
EA	Evolutionary Algorithm
EMC	Exponential Monte Carlo
EMCQ	Exponential Monte Carlo with Counter
FD	Flex Deluge
G-AND	Group Decision Making Move Acceptance with ANDing
G-OR	Group Decision Making Move Acceptance with ORing
G-PVO	Group Decision Making Move Acceptance with Probabilistic VOting
G-VOT	Group Decision Making Move Acceptance with VOTing
GA	Genetic Algorithm
GCP	Graph Coloring Problem
GD	Great Deluge Acceptance Criterion
GP	Genetic Programming
GR	Greedy Heuristic Selection
HFS	Hybrid Flow Shop
Hyper-GA	Genetic Algorithm based Hyperheuristic
Hyper-TGA	Tabu assisted Genetic Algorithm based Hyperheuristic
IE	Improving or Equal Moves Accepted
JSSP	Job-Shop Scheduling Problem
LCS	Learning Classifier System

MC	Monte Carlo Acceptance Criterion
MHSS	Meta-Hyper-Heuristic Scheduler
MOEA	Multi-objective Evolutionary Algorithm
MOP	Multi-Objective Optimization Problem
MUTN	Mutation
NGHC	Next Gradient Hill Climber
NP	Non-deterministic Polynomial time
NRP	Nurse Rostering Problem
OI	Only Improving Moves Accepted
OSSP	Open-Shop Scheduling Problem
RMHC	Random Mutation Hill Climber
SA	Simulated Annealing
SAT	Satisfiability Problem
SPP	Strip Packing Problem
SR	Simple Random Heuristic Selection
SWPD	Swap Dimension Operator
PPSP	Project Presentation Scheduling Problem
RBHC	Random Bit Mutation Hill-Climbing Operator
RD	Random Descent Heuristic Selection
RP	Random Permutation Heuristic Selection
RPD	Random Permutation Descent Heuristic Selection
SSSP	Sales Summit Scheduling Problem
TABU	Tabu Search Heuristic Selection
TSHH	Tabu Search Hyperheuristic
TTML	Timetabling Markup Language
UCTT	University Course Timetabling
VNS	Variable Neighborhood Search

1. INTRODUCTION

1.1. Motivation

Cowling, Kendall and Soubeiga (2000) and Burke et al. (2003) describe hyperheuristics as easy to implement high level heuristics that manage a set of low level heuristics. It is also stated as "*heuristics to choose heuristics*" (Burke et al. (2003)), since the method performs search within search space of heuristics, instead of problem space (Ross (2005)). In a hyperheuristic approach, a single (or a set of) low level heuristic is selected based on some problem independent measures and applied to a candidate solution. Bilgin, Ozcan and Korkmaz (2006) identify a *simple hyperheuristic* as a two-stage approach that performs a search using a single candidate solution in an iterative cycle. A simple hyperheuristic combines *heuristic selection* and *move acceptance* strategies. Simple hyperheuristics are also referred to as *perturbative* (or improvement) hyperheuristics as well, since they utilize a set of perturbative (improvement) low level heuristics. In this thesis, these terminologies can be used interchangeably. In Bilgin, Ozcan and Korkmaz (2006), combinations of seven heuristic selection mechanisms and five move acceptance strategies are tested over a set of benchmark problems. The empirical results indicate that the move acceptance strategy plays an important role in the overall performance of a hyperheuristic. Additionally, it is observed that different acceptance mechanisms might yield different performances for different problem instances. This observation is vital, since it implies that another level can be introduced on top of the hyperheuristics that can be used for managing them. Then the question arises: "How are we going to end this hierarchical growth in the levels?"

Burke, Kendall and Soubeiga (2003) present a hyperheuristic framework (F_A) without differentiating the type of low level heuristics. On the other hand, Ozcan, Bilgin and Korkmaz (2006) separate *mutational heuristics* and *hill climbers* and propose three additional hyperheuristic frameworks (F_B, F_C, F_D) that utilize such low level heuristics in a different way. An improved or equal quality solution is expected from a hill climber as a local search component, while a mutational heuristic is a methodological random perturbation. The empirical results indicate the significant success of the framework F_C

that utilizes mutational heuristics only as low level heuristics and employs a single predetermined hill climber at each step. Ozcan, Bilgin and Korkmaz (2008) verify the same results in a different experimental setting having the number of heuristics reduced.

There are many different approaches used in search and optimization. Meta-heuristics are commonly preferred methodologies for solving complex problems. Genetic algorithms (GAs) are population based metaheuristics that simulate Darwinian evolution and biological processes at a genetic level (Holland (1975), Goldberg (1989a, 1989b)). *Meme* as a terminology is invented by Dawkins (1976). A meme denotes a “contagious” piece of information that can be processed, digested, adapted and transmitted by each infected member in a population. This overall course carries some similarities with local improvement. Hence, GAs hybridized with hill climbing are referred to as memetic algorithms, in which a meme denotes a hill climber (Moscato and Norman (1992), Radcliffe and Surry (1994)). Multimeme memetic algorithms extend the definition of a meme from hill climbing to other operators (Krasnogor (2002), Krasnogor and Smith (2000-2002)). A meme (-plex) is allowed to encode all relevant features and properties of a set of operators in a single structure. The memes are co-evolved with the genes. Ozcan, Bilgin and Korkmaz (2008) analyze all these algorithms and compare their performance to a hyperheuristic using the F_C framework on the same set of problem instances. The results show that a hyperheuristic can generate a matching performance to a meta-heuristic.

As a disadvantage, the components of a meta-heuristic designed for solving a problem might require modifications while solving another problem in another domain. Although, Cowling, Kendall and Soubeiga (2000) imply that hyperheuristics are problem independent, Ozcan, Bilgin and Korkmaz (2008) show that they still can not get away from the “no free lunch” theorem (Wolpert and MacReady (1997)). The set of low level heuristics, heuristics selection method, move acceptance strategy and/or the framework used in the methodology might become problem dependent. Therefore, the properties of the problem at hand should still be considered while using a hyperheuristic for solving it.

1.2. Methodology

In this study, move acceptance stage within the simple hyperheuristics is focused and four different move acceptance methods that are derived from well known group decision making models which involve different characteristics are investigated. As it is mentioned before, hyperheuristics have a heuristic selection mechanism to choose the best heuristic for the current step to get a better performance among a set of low-level heuristics. However, for move acceptance mechanism, researchers used just one of them in their hyperheuristics until now. What about using a bunch of them and giving more healthy decisions by combining their strength.

The use of a group decision making strategy allows all mechanisms to operate in the same level. Hyperheuristics that combine these move acceptance strategies with seven heuristic selection methods are tested within the traditional framework over fourteen benchmark functions and twenty-one examination timetabling problem instances. The experiments are repeated using the F_C hyperheuristic framework for the benchmark functions. Moreover, the performances of the group decision making hyperheuristics are compared to the other approaches from previous studies.

1.3. Objective of the Research

Objective of the research is to find a way to end the hierarchical growth of hyperheuristics by employing group decision making strategies during move acceptance. Actually, it is obviously hard to state such a certain expression about the level of hyperheuristics. But, here, the thing that we tried to say is, there will be no need to bother about the move acceptance part of hyperheuristics by using group decision making strategies. Since, we anticipate that this group decision making idea will get rid of weaknesses of them during benefiting from the power of their combination.

The other aim of this study is to provide new research directions for hyperheuristics. It is stated that there will be no need to work on move acceptance, anymore, because, it will already meet the requirements of today’s studies. Nevertheless, there can be additional studies to improve the proposed idea by answering some questions and getting some desired conclusions such as “Which set of move acceptance is the best”, “What about other group decision making strategies”, “Performance of them on different problems”.

1.4. Organization of the Thesis

This thesis is divided into seven chapters. This chapter is about motivation behind the research and objective that directs us to work on it. The remaining chapters are constructed in the following way:

In Chapter 2, the idea of hyperheuristics is introduced, then, the intellectual roots of hyperheuristics for being aware of the starting point and a detailed literature survey that consists of previous academic and practical works is presented. Some hyperheuristic approaches are explained. Current application areas, problems, are listed with references.

In Chapter 3, overview belongs to the main subject of our research which is group decision making is provided and question of “How can group decision making be applied onto Hyperheuristics” is answered. The related hyperheuristic frameworks are presented.

In Chapter 4, first experimental phase of this research is introduced and some mathematical benchmark functions for the experiments are provided. Our heuristic set, experimental settings and results of these experiments with a comprehensive performance analysis is given.

In Chapter 5, another problem domain, examination timetabling is introduced and literature survey is presented. Also, the mathematical formulation of this problem is provided. In Chapter 6, group decision making hyperheuristics for examination timetabling are discussed along with the experimental data set. Additionally, low level heuristics to solve the problem, experimental settings and experimental results are provided.

The last chapter discusses conclusions and remarks with possible research directions.

Additionally, full list of references and some appendixes about experimental results are provided at the end of the thesis.

2. HYPERHEURISTICS

2.1. Introduction

Many researchers have been progressively involved in hyperheuristics as an emerging approach in search and optimization (Cowling, Kendall and Soubeiga (2000), Burke, Kendall and Soubeiga (2003)). A hyperheuristic can be considered as a heuristic scheduler. An appropriate heuristic or a set of heuristics is selected and applied to a candidate solution. As a layered approach, and *hyperheuristic* layer interact with *problem* and *heuristic* layers through problem independent measures, such as the quality change in a candidate solution when the selected heuristic is employed as illustrated in Figure 2.1. A *hyperheuristic pattern* denotes a triplet; the hyperheuristic instance, the hyperheuristic framework and the set of low level heuristics used for solving a problem. Initially hyperheuristics are suggested as an alternative to meta-heuristics. However, meta-heuristics can be used as a hyperheuristic or a hyperheuristic can be used within a meta-heuristic. Moreover, hyperheuristics can be hybridized with any other approach. A problem can be encoded using a direct representation or an indirect representation. For example, assuming that there is a timetabling problem for which the aim is assigning a set of events to a given set of time periods, then a candidate solution can be implemented using an array. The encoding where each entry is an assignment of an event is a direct representation. For example, Bilgin, Ozcan and Korkmaz (2006) investigate the performance of hyperheuristics over examination timetabling using direct representation. Each entry in a candidate solution encodes the period when a corresponding examination will be held. On the other hand, if each entry of the array encodes a heuristic that will construct the schedule for the corresponding event, this scheme is an indirect representation. As an example, Burke et al. (2007b) solve examination and course timetabling problems by using such a representation. Their hyperheuristic is based on a tabu search mechanism that assigns proper graph colouring heuristics for constructing an examination timetable.

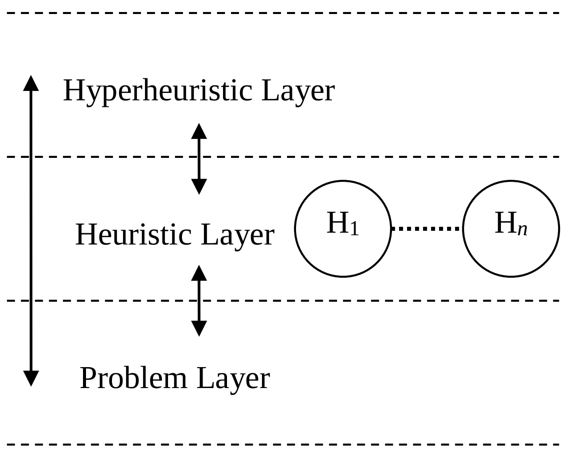


Figure 2.1. Layers in a generic hyperheuristic framework

Bilgin, Ozcan and Korkmaz (2006) distinguish the heuristic selection and move acceptance processes within the hyperheuristic layer. Simple hyperheuristics select a heuristic from a set of low level heuristics, apply the chosen heuristic to the candidate solution and finally decides whether to accept or reject the new solution at each step as presented in Figure 2.2. An initially generated solution goes through this process repetitively until a set of termination criteria is satisfied. Hopefully, the final solution is the optimal solution for the problem at hand. The best performing hyperheuristic framework F_C allows better use of hill climbers in combination with mutational heuristics by embedding a hill climbing component to the generic framework, right after the step number 5 in Figure 2.2. (Bilgin, Ozcan and Korkmaz (2006)). The move acceptance criteria in this framework evaluate the combined performance of the selected mutational heuristic and the hill climber.

```
1.  start from an initial candidate solution c
2.  while (termination criteria not met){
3.     select a heuristic a= Hi|{H1, ..., Hi, ..., Hn}
4.     make a move to a new solution c′ =a(c)
5.          by applying chosen heuristic to c
6.     decide accept_reject( c′ )
7.     if (c′ is accepted) then
8.          c= c′
9.  }
```

Figure 2.2. Generic simple hyperheuristic framework

2.2. Literature Survey

Fisher and Thompson (1961, 1963) and Crowston et al. (1963) generated initial studies on hyperheuristics by employing their approaches to the Job-Shop Scheduling Problem (JSSP). In their hyperheuristic, a probabilistic learning strategy is employed that assigns and sets weights of heuristics for adaptation. Fang, Ross and Corne (1994) used genetic algorithm based on a hyperheuristic for solving an Open-Shop Scheduling Problem (OSSP). Although the approach was not referred to as hyperheuristic, Gratch, Chein and Jong (1993) utilized multiple heuristics for planning communication schedules to satisfy available constraints for earth-orbiting satellites and ground stations selecting the best one a not the term. They called their approach as COMPOSER. Hart, Ross and Nelson (1998) utilized a genetic algorithm for managing a set of heuristics to solve chicken catching and transportation problem. Cowling, Kendall and Soubeiga (2000) tested most of the simple hyperheuristic components on a sales summit scheduling problem (SSSP). *Simple Random* (SR) heuristic selection mechanism randomly chooses a low level heuristic based on a uniform probability distribution at each step. *Random Descent* (RD) selects the heuristic in the same manner as SR, but applies it repeatedly until no improvement is achieved. *Random Permutation* (RP) generates a random initial permutation of the low level heuristics and at each step applies a low level heuristic in the provided order sequentially. *Random Permutation Descent* (RPD) processes the low level heuristics in the same manner as RP, but proceeds in the same manner as RD without changing the order of heuristics. The *Greedy* (GR) method applies all heuristics to a given candidate solution and selects the one that generates the most improved solution. *Choice Function* (CF) uses a learning mechanism that scores low level heuristics based on their individual and pair-wise performances. The heuristic having the best score is selected at each step and applied to the candidate solution. Cowling, Kendall and Soubeiga (2000) used only two simple acceptance criteria in their study. AM accepts all moves and OI, that accepts only improving moves. According to the experimental results, the CF_AM hyperheuristic shows potential. Again, Cowling, Kendall and Soubeiga (2001) used their background from their previous study (Cowling, Kendall and Soubeiga (2000)) and applied hyperheuristics onto Project Presentation Scheduling Problem (PPSP). In addition, Cowling, Kendall and Soubeiga (2001) used choice function which ranks the low-level heuristics based hyperheuristic to solve SSSP.

Burke et al. (2002a) proposed a new hyperheuristic utilizing case based reasoning (CBR) approach that attempts to make reasonable predictions during heuristic selection process by the help of previous knowledge. Burke, Petrovic and Qu (2006) extended this study and tested such a system on a set of timetabling problems. Cowling, Kendall and Han (2002a) named genetic algorithm based hyperheuristics as hyper-GA and investigated its performance on a trainer scheduling problem. After that, Cowling, Kendall and Han (2002b) modified this approach that allows variable length chromosomes, named as adaptive length chromosome hyper-GA (ALChyper-GA) and tested it on the same problem. Ross et al. (2002) proposed a hyperheuristic learning classifier system (LCS) for solving bin-packing problem. Han and Kendall (2003a) extended their hyper-GA approach using tabu search for preventing invocation of inefficient low-level heuristics and named it as hyper-TGA. Han and Kendall (2003b) attempted to improve hyper-GA using somewhat guided genetic operators to support more efficient removal and insertion processes of heuristics and heuristic sequencing. Cowling and Chakhlevitch (2003) applied eleven hyperheuristics including greedy, simple random, peckish and variants of a tabu-search using a large set of low level heuristics to two personnel scheduling problems. Rossi-Doria and Paechter (2003) used an evolutionary algorithm based hyperheuristic to solve course timetabling problem.

Ayob and Kendall (2003) tested *Monte Carlo* acceptance mechanisms that accept non-improving moves based on a probabilistic framework along with the improving moves. *Exponential Monte Carlo* (EMC) accepts a worsening move with a probability of p_t as presented in Equation (2.1). *Exponential Monte Carlo with Counter* (EMCQ) extends EMC by utilizing a counter that resets and increments for each consecutive non-improving move and causes the probability to increase. The authors compared different hyperheuristics and the results yielded with the success of the SR_MC hyperheuristic.

$$p_t = e^{-\frac{\Delta f \times \Delta t}{\Delta F \times t \times Q}} \tag{2.1}$$

where Δf is the fitness change, ΔF is an expected range for the maximum fitness change, Δt is the time change, t is the possible time interval between two moves, Q is a counter

Burke, Kendall and Soubeiga (2003) proposed a hyperheuristic that combines *tabu-search* and *ranking* as a heuristic selection mechanism for timetabling (TABU_IE). The ranks of heuristics determine which heuristic will be applied to the candidate solution, while the tabu list holds the heuristics that should be avoided. A reinforcement learning mechanism updates the rank of a low level heuristic based on the change in the quality of a candidate solution after the selected heuristic is employed. Ross et al. (2003) used a messy-GA based hyperheuristic that decides on which heuristic combination is the best to reach a feasible solution for one-dimensional bin packing problems. Burke, Silva and Soubeiga (2003) investigated multi-objective hyperheuristics for generating a uniform pareto front. Ross, Marin-Blazquez and Hart (2004) experimented with the same hyperheuristic on a set of timetabling problem instances using three different fitness measures. Nareyek (2004) compared two hyperheuristics using a variety of reinforcement learning mechanisms based on different weight adaptation strategies on two constraint optimization problems.

Kendall and Mohamad (2004) experimented with a hyperheuristic that used SR heuristic selection method and *Great Deluge* (GD) acceptance criterion stochastic acceptance mechanism on a set of channel assignment problems. GD is based on a stochastic framework which allows improving moves by default and non-improving moves if the objective value of the candidate solution is better or equal to an expected objective value, named as *level* at each step. The objective value of the first generated candidate solution is used as the initial level and the level is updated at a linear rate towards a final objective value as shown in Equation (2.2).

$$\tau_t = f_o + \Delta F \times \left(1 - \frac{t}{T}\right) \tag{2.2}$$

where τ_t is the threshold level at step t in a minimization problem, T is the maximum number of steps, ΔF is an expected range for the maximum fitness change and f_o is the final objective value. Burke and Bykov (2006) proposed a modified version of GD, referred to as *Flex Deluge* (FD). This new hyperheuristic introduced a *flexibility* factor that provides a search characteristic in between GD and hill climbing. The experimental results over a subset of examination timetabling benchmarks showed that the approach is promising.

Bai and Kendall (2003) employed a simulated annealing based hyperheuristic to solve different types of shelf space allocation problems, while Dowsland, Soubeiga and Burke (2005) used the similar strategy for providing optimal space allocation. Kendall and Hussin (2004) applied the tabu-search hyperheuristic to solve the examination timetabling problem of University Technology MARA, the largest university in Malaysia. Two different tabu duration management strategies and three different move acceptance mechanisms were tested. Gaw, Rattadilok and Kwan (2004) proposed a distributed choice function for solving timetabling and scheduling problems. Burke et al. (2005a) developed an ant algorithm based hyperheuristic that determines an effective sequence of heuristic moves for solving project presentation scheduling problem. Burke, Silva and Soubeiga (2005) modelled a tabu-search based hyperheuristic as a multi-objective approach for selecting the best heuristic considering the objectives. This multi-objective hyperheuristic is tested on two different real-world optimization problems, namely; space allocation and timetabling. Burke et al. (2005b) studied a different approach that hybridized two graph colouring heuristics (Saturation Degree and Largest Degree) with a tabu search hyperheuristic and the related experiments were performed on a set of examination timetabling data. Cuesta-Cañada, Garrido and Terashima-Marín (2005) successfully combined hyperheuristics with ant-colony optimization algorithm for solving 2D bin packing problems. Qu and Burke (2005) proposed a hybrid approach by using VNS in hyperheuristics as a mechanism which provide efficient usage of search spaces belonging low-level heuristics. Burke, Hyde and Kendall (2006) used genetic programming as a hyper-heuristic that provided an efficient decision mechanism for a set of low-level building blocks during construction of a successful heuristic in the study on bin packing problem. Ersoy, Ozcan and Uyar (2007) compared the performance of different simple hyperheuristics that were utilized to manage multiple hill climbers within the memetic algorithms on a set of examination timetabling benchmark problem instances. Marin et al. (2007) presented two different evolutionary computation based models which includes Learning Classifier System (LCS) and Genetic Algorithm (GA) to produce efficient hyperheuristics for solving 2D Bin Packing Problem (BPP).

Rodriguez, Petrovic and Salhi (2007a) proposed a number of meta-hyperheuristic approaches with GA and applied them Hybrid Flow Shop (HFS) problem. Again, Rodriguez, Petrovic and Salhi (2007b) worked another study which is about associated problem that is related finding the best sequence of heuristics to construct the desired solution and defined the associated problem for constructive hyperheuristics. Rodriguez and Salhi (2007c) introduced a hybrid approach called Meta-Hyper-Heuristic Scheduler (MHHS) by combining meta-heuristic and hyper-heuristic based on an evolutionary method. Burke et al. (2007a) studied an automatic heuristic generation technique using GP. Garrido and Riff (2007) presented an evolutionary hyperheuristic to solve 2D Strip Packing Problem (SPP), also, Araya, Neveu and Riff (2008) designed a hyperheuristic framework to solve 2D SPP. Thabtah and Cowling (2008) used some associative classification techniques as supervised learning mechanisms to perform data mining for predicting the most appropriate low-level heuristic to apply during further steps.

An acceptance criterion used within the simple hyperheuristics can be labeled as *parametric* if the acceptance and rejection of a move is decided by a rule depending on a set of parameters. Otherwise, the move acceptance method is called *non-parametric*. Additionally, an acceptance mechanism can be characterized as *stochastic* (non-stochastic) if a probabilistic framework is (not) utilized while accepting or rejecting a move. Existing move acceptance methods falls in one of three categories as presented in Table 2.1.

Table 2.1. Categorization of existing move acceptance methods used within simple hyperheuristics.

	non-parametric	*parametric*
stochastic	–	MC, SA
non-stochastic	AM, IE, OI	GD, FD

In the following subsections, some promising hyperheuristics are focused. Most of these techniques attempt to embed some form of intelligent mechanism in the approaches to perform a better search. In the previous studies, learning is achieved mainly based a machine learning technique or through Darwinian evolution. The low level heuristics can be perturbative of constructive heuristics.

2.2.1. Reinforcement Learning Hyperheuristics

Reinforcement Learning is a widely studied research area under machine learning. It provides a learning mechanism which helps an agent to learn how to behave an action comes during any state through "*trial-and-error*" interactions (Kaelbling, Littman and Moore (1996)). It can also be stated as learning "*how to map situations to actions*" to maximize reward of the agent (Sutton and Barto (1998)). The characteristic of the environment is important to apply reinforcement learning. It can be either stationary or non-stationary. Stationary environment means that any action taken during any state will result with the same state for all the time. For instance, a robot that must get out from a maze will go in the same direction when an action comes at the same coordinate. On the other hand, non-stationary environment has a dynamic structure, so, it is the opposite of the stationary. The most famous work about reinforcement learning and its application as a hyperheuristic is presented in Nareyek (2004). He used reinforcement learning to provide an adaptive system by using weights for each heuristics based on their performances. He proposed two different heuristic selection functions. First one is about probability based selection by looking theirs utility values, weights (2.3) and it is called as fair random choice. The other one is available for heuristics which have maximum utility values. A heuristic with maximum utility is chosen to be applied onto the current candidate, if there are more than one maximum weighted heuristic, a random choice applied to choose one of them.

$$P_a = w_a / \sum_i w_i \tag{2.3}$$

Different weighting strategies applied to make adaptive performance measurements and they are listed in the following table. If newly created solution is better than the current one, then positive reinforcement is applied. On the other hand, if the new solution is worse than or equal to the current, than negative reinforcement is applied on the chosen heuristic.

Table 2.2. Reinforcement learning weighting strategies with *positive* and *negative reinforcement* (Additive: +, Subtractive: -, Multiplicative: x, Divisive: /).

$\{+;-\}$ Adaptation	$w_a \longleftarrow w_a + 1$ $w_a \longleftarrow w_a - 1$
Escalating $\{+;-\}$ Adaptation	$w_a \longleftarrow w_a + m_{promotion}$ $w_a \longleftarrow w_a - m_{promotion}$
$\{x;/\}$ Adaptation	$w_a \longleftarrow w_a \times 2$ $w_a \longleftarrow w_a / 2$
Escalating $\{x;/\}$ Adaptation	$w_a \longleftarrow w_a \times m_{promotion}$ $w_a \longleftarrow w_a \times m_{promotion}$
Power Adaptation Root Adaptation	$w_a \longleftarrow \begin{cases} w_a \times w_a \Leftarrow w_a > 1 \\ 2 \Leftarrow w_a = 1 \end{cases}$ $w_a \longleftarrow \sqrt{w_a}$

2.2.2. Choice Function Hyperheuristics

A Choice Function Hyperheuristic (Cowling, Kendall and Soubeiga (2000)) is a learning based adaptive system that tries to rank low-level heuristic by looking their previous performances during optimization process. This performance measurement phase is handled by three distinct strategies: *own performance, pair-wise performance, elapsed time for a heuristic last called*. So, it gives a chance to make a deeper judgment regarding low-level heuristics.

The first used performance measurement criteria is straightforward, measure the performance of each heuristic separately. Evaluation of a heuristic is achieved by getting information about improvement that is provided by a heuristic in a unit time. To make this phase more plausible and reach a better conclusion, a constant integer term, α [0, 1], is used to increase the degree of importance of the recent successes. It is mathematically defined as in Formula 2.4. N_j is the low-level heuristic that we want to measure its performance, I_n (N_j) is value of improvement and T_n (N_j) is the elapsed time during the n^{th} iteration.

$$f_1(N_j) = \sum_i \alpha^{i-1} \frac{I_i(N_j)}{T_i(N_j)} \tag{2.4}$$

For the further performance calculations of each heuristic, previous values that come from (2.4) can be used as it is in (2.5).

$$f_1^{current}(N_j) = \frac{I(N_j)}{T(N_j)} + \alpha \cdot f_1^{previous}(N_j) \tag{2.5}$$

The second one is related to pair-wise performance. That is, this approach measures performances of consecutively applied heuristics. Mathematical definition is given in (2.6). According to the formula, heuristic N_j is applied just after heuristic N_k and the other functions, I_n (N_j, N_k) and T_n (N_j, N_k) do the same job in (2.4), but now for two low-level heuristics instead of one. Differently, α is replaced with β, but the possible values are the same.

$$f_2(N_k, N_j) = \sum_n \beta^{n-1} \left(\frac{I_n(N_k, N_j)}{T_n(N_{k,} N_j)}\right) \tag{2.6}$$

Again, this calculation can be done for only once, then, for the further evaluations (2.7) can be used.

$$f_2^{current}(N_k, N_j) = \frac{I(N_k, N_j)}{T(N_k, N_j)} + \beta \cdot f_2^{previous}(N_k, N_j) \tag{2.7}$$

As it is stated in the beginning, there is one more function (2.8), for the performance measurement and it is simply elapsed time since the last called for a heuristic. It is used for diversification process, but the first two are used for intensification. So, choice function hyperheuristic provides a system which improves the solution and decreases the possibility of being stuck at local optima.

$$f_3(N_j) = \tau(N_j) \tag{2.8}$$

2.2.3. Simulated Annealing Hyperheuristics

Simulated Annealing (SA) (Kirkpatrick, Gelatt and Vechhi (1983)) is a non-deterministic optimization technique that was born from annealing metals and it is imitated version of real annealing operation. The process of giving form to a metal includes two phases; first, solid metal must be heated to a high temperature which causes the atoms move freely to change its state into a soft structure (*high energy state*), and, it must be cooled down to change its state into a crystallized structure (*low energy state*) with a rigid shape. SA is used solving NP-hard (Garey and Johnson (1979)) combinatorial optimization problems in a problem independent manner by using a stochastic decision system. It includes a diversification mechanism for escaping from local optima and intensification mechanism, naturally.

SA approach is used as a hyperheuristic mechanism in Bai and Kendall (2003), Downsland, Soubeiga and Burke (2005). Hyperheuristics have a heuristic selection and a move acceptance methods and SA is used as a move acceptance decision strategy in the first mentioned study. The pseudo code of SA hyperheuristic for a maximization problem is provided in Figure 2.3. First of all, an initial solution is selected, as it happens in each perturbative or improvement hyperheuristics, then, optimization process of hyperheuristic starts by selecting a heuristic, randomly. So, in this method, heuristic selection is coded as *SR*.

```
Select an initial solution s0 ;
Repeat
        Randomly select a heuristic h∈H ;
        iteration_count = 0;
        Repeat
                iteration_count ++;
                Applying h to s0, get a new solution s1 ;
                δ = f(s1) − f(s0)
                if (δ ≥ 0) then s0 = s1 ;
                else
                        Generate a random x uniformly in the range (0,1);
                        if (x < exp(δ / t)) then s0 = s1 ;
        Until iteration_count = nrep;
        t = t /(1 + β * t);
Until the stopping criteria = true.
```

Figure 2.3. SA Hyperheuristic Pseudo Code (Bai and Kendall (2003))

After the heuristic selection process, the chosen heuristic applied on the solution at hand and new solution is generated. It is evaluated by a fitness function and IE acceptance mechanism looks at the new solution to give its decision whether to swap the solutions. If the solution is not good enough at least as the previous one, then a worsening moves acceptance system gives its decision about the solution based on Metropolis probability (Metropolis et al. (1953)). Then, the key point about annealing comes out by the temperature. Each time, the temperature is decreased as cooling process by a constant value, β , which is calculated as in (2.9) (t_s = *Starting Temperature* , t_f = *Ending Temperature, K = Total # of Evaluations*). In conclusion, this cooling means that, probability concerning acceptance of bad moves decreases in time, too. In addition, there exists a learning based selection for heuristics by assigning some weights to heuristics and updating them by looking at their performances.

$$\beta = (t_s - t_f) / K \times t_s \times t_f \tag{2.9}$$

$$K = T_{allowed} / T_{avg} \tag{2.10}$$

In addition, to calculate the value of *K*, (2.10) is used with two parameters called $T_{allowed}$ and T_{avg}. To reach the value, we take the ratio of maximum time for the search and optimization process and average spent unit time for each iteration. That is, this gives us information about possible number of iterations to reach a desired solution within pre-defined limited time. Finally, there is also one other parameter, initial temperature (t_s) to be calculated carefully. Some researchers (Dowsland (1995), Johnson et al. (1989, 1991), Ben-Ameur (2004)) worked on this issue, especially and proposed different approaches to calculate it.

2.2.4. Tabu Search Hyperheuristics

Tabu Search ((Glover (1989, 1990))) is an optimization algorithm and a local search technique that is used to solve combinatorial optimization problems. It is *based on introducing flexible memory structures in conjunction with strategic restrictions and aspiration levels as a mean for exploiting search spaces* (Ganapathy, Marimuthu and Ponnambalam (2004)). The main structure of this meta-heuristic is about mentioned flexible memory structure. It is a short-term memory to prevent making cyclic moves by using a *tabu list* which holds recent history belonging forbidden moves. Here, it is an important issue to determine the tabu list size to reach an efficient search mechanism that does not be stuck in any local optima.

Tabu Search Hyperheuristic (TSHH) (Burke, Kendall and Soubeiga (2003)) is a methodology to embed and adapt tabu search idea into a hyperheuristic to provide a generic problem solving strategy via getting rid of tabu search’s problem specific structure.

In Burke, Kendall and Soubeiga (2003), heuristics are thought as attributes that are used for tabu list as forbidden moves to exclude some heuristics from the selection pool of low-level heuristics. Also, reinforcement learning is used to present a ranking mechanism within a score range, changes between *0* and *number of low-level heuristics*, because of the competition between heuristics and to differentiate them based on their performances. In Figure 2.4, pseudo code of TSHH is given and it simple states that, if there is an improvement, then increment ranking of currently applied heuristic, otherwise, decrement it and add it to a variable length dynamic tabu list.

<u>Do:</u>   *Select heuristic k with highest rank and apply it once*   *If Δ > 0 then rk = rk + α*   *Else rk = rk - α, Include heuristic k inTABULIST*   ***<u>Until Stopping condition is met</u>***

Figure 2.4. THH Framework; r_k denotes rank of the heuristic k, $\alpha = 1$, Δ is change in the objective function.

2.2.5. Genetic Algorithm based Hyperheuristics

A Genetic Algorithm based Hyperheuristic (hyper-GA) (Cowling, Kendall and Han (2002)) is a hyperheuristic which uses GA as a heuristic selection mechanism to solve wide range of problems via indirect representation. In this representation, each gene is coded with a number showing a heuristic. In Han and Kendall (2003), an example of hyper-GA is presented. 14 low-level heuristic provided to solve geographically distributed training staff and course scheduling problem and they are encoded with numbers from 0 to 13.

2	3	1	5	0	7	9	8	11	1	10	9	12	13

Figure 2.5. An example of a hyper-GA (Han and Kendall (2003))

In Figure 2.5, an example belongs to hyper-GA is given and it shows the order of heuristics which will be applied onto the current solution. That is to say, this representation, chromosome, is available for answering the questions of which heuristic will be applied and in which order? Because of GA is a population based search and optimization problem solving methodology, hyper-GA also performs its search with a population and each chromosome is an individual in this population. Mutation and crossover operations also exist to evolve the population for generating a better generation in an adaptive way.

1. *Generate an initial solution (S) randomly*
2. *Generate 30 initial chromosomes (length of 14, # of low-level heuristics), put them into a pool (population)*
3. *For each chromosome k (0 ≤ k <30),*
  a. *Apply low-level heuristics in the order given in the chromosome to S*
  b. *Record the solution S_k*
  c. *Record the change each single gene makes to the objective function*
4. *Compare each S_k to S: if $S_k > S$, then $S = S_k$*
5. *Select parents. For each pair of parents: decide which crossover operator to us and apply it based on p_x (Crossover Rate)*
6. *Select chromosomes for mutation, for each: decide which mutation and apply it based on p_m (Mutation Rate)*
7. *Add all new chromosomes and 10 best chromosomes in current pool to a new pool. If the stopping criteria is met, stop the evolution, else, go to 3.*

Pseudo code of hyper-GA is available in Figure 2.6. It simply works in a way that; take one chromosome which includes heuristic order and apply all the heuristics in the given order, repeat this process for each chromosome. As a result, get the best produced solution from the pool. Then, perform genetic operations for each individual based on *crossover* and *mutation* rates.

2.2.6. Multi-Objective Hyperheuristics

Multi-Objective Optimization (Sawaragi, Nakayama and Tanin (1985)) is a parallel optimization process that tries to satisfy the available objectives concerning a set of constraints by preserving some kind of trade off about the objectives. This trade off issue provides a balanced solution for any multi-objective optimization problem (MOP) such as graph coloring problem (GCP). At this point, famous “no free lunch” theorem comes out and it states that if one objective or constraint is solved in the most optimized way, then for the rest, the situation will not be as bright as the solved one. So, it is an absolute necessity to solve all the objectives according to their importance by arranging a settlement between them.

Differently from general characteristics of single objective optimization, there can be more than one global optimal point which satisfies the requirements of a problem. These points form a set of non-dominated solutions which provides the same benefits from the general quality of the solutions' perspective and it is called *Pareto Optimality Set*. The pareto concept is provided in Figure 2.7 and the figure shows two objectives (y_1, y_2) and decision points for them.

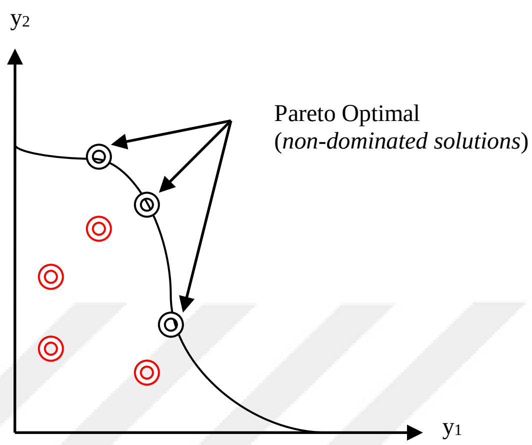


Figure 2.7. Pareto Optimality (Zitzler (2002)). The line denotes the *pareto front* that takes shape from all the optimum points or solutions. The red donuts denote dominated solutions.

When a MOP is solved, then we have a list of optimal solutions as a pareto front. Then, we must give a decision concerning necessities of the problem to choose one among them.

Multi-objective optimization is used with TSHH in Burke, Silva and Soubeiga (2005). In this study, reward & punishment strategy belonging reinforcement learning is used and aim of the study is to push the solution to the pareto front to reach optimal solution. So, the learning mechanism helps to determine which objective can be solved with which heuristic at which time. They proposed three distinct TSHH frameworks and tested them with learning and without learning. One of the presented algorithms is provided in Figure 2.8.

Randomly generate an initial population of P solutions.
<u>For each solution in the initial population, Do</u>

a. *Select an individual objective u uniformly at random.*

Figure 2.8. Single Tabu Random Uniform (TSRandUnif)

In the given algorithm, first of all, an initial population is generated for the problem. For each possible solution in the problem, an objective is selected to be optimized and to optimize it a heuristic which is the highest ranked one via reinforcement learning scoring mechanism is selected and applied onto it. Based on the performances of heuristics, they are added into the tabu list or the tabu list is made empty as it performs in TSHH. These operations are performed for the other objectives, too.

2.2.7. Ant Algorithm based Hyperheuristics

Ant algorithm or Ant Colony Optimization (ACO) (Dorigo (1992)) is a nature-inspired algorithm which simulates real ants or their colonies who try to find a path for food to solve combinatorial optimization problems. ACO is used within hyperheuristics as population which involves ants as hyperheuristic agents to construct good sequences of heuristics in a stochastic way (Burke et al. (2005), Cuesta-Cañada, Garrido and Terashima-Marín (2005)). This is a population based technique and the ants move together through the best moving place or vertex or heuristic. That is to say, if ant produces the best move for the current stage at a vertex, then all the others go the vertex where the applied heuristic located in and make their moves at there. In addition, successive moves, which heuristic will be applied then, are determined based on a probability value called *pheromone* that gives some information about how well to apply a certain heuristic afterone other.

2.2.8. Case-Based Reasoning Hyperheuristics

Case-based reasoning is a learning approach which tries to solve any problem by using some information that comes from the past; that is, it is an offline, experience oriented problem solving methodology. Another definition of CBR in different words can be stated as "to solve a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation" (Aamodt and Plaza (1994)). The terminology of *case-base* refers to *memory of stored cases* (Leake (1996)) which denotes previous data: problems and solutions.

This idea is embedded in a hyperheuristic framework as a heuristic selection criterion (Burke et al. (2002), Burke, Petrovic and Qu (2006)) and it is experimented on a set of timetabling problem data. In these studies, all the main steps to create a CBR system are listed. It starts with an important step for CBR design that is knowledge discovery process which recognizes similarities between sources cases and target cases. It is implemented in two different ways (Burke, Petrovic and Qu);

- Choose proper features (Kira and Rendell (1992)) which will be used as cases to make good predictions about heuristics to be applied.
- Choose proper source cases.

Then, to reach an efficient feature list, features are trained to adjust their weights, remove irrelevant features and introduce new features, if it is necessary. After that, case bases are constructed.

2.2.9. Learning Classifier System Hyperheuristics

Learning Classifier System (LCS) (Bull (2003)) is an adaptive, rule-based learning mechanism which consists of reinforcement learning and genetic algorithms (Holland (1975), Booker et al. (1989), Goldberg (1989)). Classifiers are a set of rules that construct the system and they can also be defined as set of state to action mappings. So, LCS provides interaction between environment and actions and applies reinforcement learning to give score to each classifier by using a reward-punishment mechanism. In Ross et al. (2002), a widely used version of LCS called XCS (Stewart (1995)) is used as a hyperheuristic to solve bin-packing problem. The proposed system tries to determine the best combination of low-level heuristics to solve a given problem for the current state belongs to the solution. That is, this system finds the best pairs of *state* and *action* by observing which heuristic must be applied at which state. With this approach, it will be possible to get better results by using a combination of heuristics instead of just one. LCS based hyperheuristic framework is provided in Figure 2.9.

Do

Select a heuristic and apply it to pack a bin
If it fills the bin, then reward
Else punish
(Do this for each altered states for each heuristic)

Until All Items has been Packed

Figure 2.9. LCS Hyperheuristic for bin packing problem. It finds which heuristic is better to put an item into the bin. For instance, if the item is over 1/2 of bin capacity, then use Largest-Fit Decreasing (LFD) or if the item's size is from 1/3 upto 1/2 of bin capacity, then apply Next-Fit Decreasing (NFD) etc.

2.2.10. Variable Neighborhood Search Hyperheuristics

Variable Neighborhood Search (VNS) (Hansen and Mladenovic (1997, 2001, 2005)) is a recent metaheuristic which iteratively explores the search space by growing neighborhood size in order not to get stuck at local optima. So, differently from other single neighborhood search techniques such as SA, it does not accept worsening moves to handle this local optima issue. A basic version of VNS algorithm for a minimization problem is provided in Figure 2.10. VNS attempts to escape from a local optimum by performing other local searches from starting points sampled from a neighborhood of the current optimum, which grows its size iteratively until a local minimum is better than the current one is found. These steps are repeated until a given termination condition is met. As long as the candidate solution improves, the same neighborhood operator is used. In case of a worsening move, the next operator which has a larger step size than the current one is invoked.

Choose the neighborhood, N_k, for k = 1
Generate an initial solution (S) randomly from N_k
<u>Do</u>

- ***g.*** *Perform a local search and find a new solution (S')*
- ***h.*** *If f(S') < f(S) then S = S' and k = 1*
- ***i.*** *Else*
    - *i. If $k \neq k_{max}$ then k = k + 1 (go to another neighborhood)*

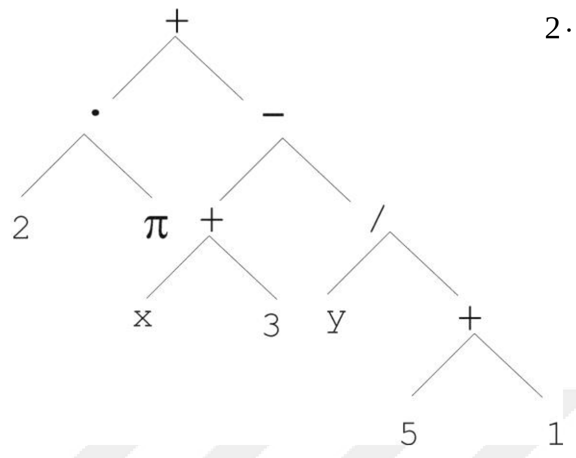
<u>Until Stopping condition is met</u>

Figure 2. 10. Basic VNS algorithm (k_{max} is the index of last neighborhood).

In Qu and Burke (2005) a hybrid VNS hyperheuristic approach is proposed. Neighborhoods are thought as heuristics within low-level heuristic set. VNS hyperheuristic manages a set of low level constructive graph heuristics by employing two different high level VNS neighbourhoods. The low level heuristics include Color Degree, Largest Degree, Largest Enrollment, Largest Weight Degree, Saturation Degree and a Random Ordering method. The first high level VNS randomly updates N_1=2, N_2=3, N_3=4, or N_4=5 low level heuristics in a given candidate solution. On the other hand, the second one randomly updates N_1=2, N_2=3, N_3=4, or N_4=5 *consecutive* heuristics as a block in a given candidate solution. The former VNS generates better results. Comparison with other methods show that iterated local search is the better than VNS, tabu search and steepest descent approaches for solving exam timetabling problems.

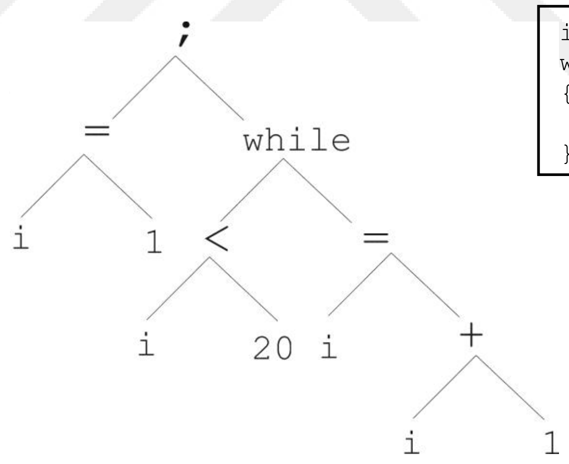
2.2.11. Genetic Programming Hyperheuristics

Genetic Programming (GP) (Koza (1992)) is a sub-category of evolutionary algorithms (EAs) to create or construct computer programs which perform a given task by a building blocks technique. It works like GA does in population that consists of some individuals refer to computer programs (mathematical formula, logical formula etc.) in tree structure. For instance, this tree structure can be simply a binary expression tree that presents a mathematical formula like in Figure 2.11, also directly a program structure is provided in tree in Figure 2.12.



$$2 \cdot \pi + \left((x+3) - \frac{y}{5+1} \right)$$

Figure 2.11. Simple Binary Expression Tree (Eiben and Smith (2003))




```
i =1;
while (i < 20)
{
        i = i +1
}
```

Figure 2.12. A simple computer program (Eiben and Smith (2003))

There is also a ranking or scoring mechanisms (fitness measurement) for each individual to observe their performances and is used for generally for parent selection process. In addition, regular genetic operators that are used in GA are available in GP, too. *Recombination* is for making an exchange between sub-trees and *Mutation* changes a tree in a random way based on related probabilistic constants.

GP based hyperheuristics (Burke, Hyde and Kendall (2006)) construct or evolve heuristics by building it. It works in a population based strategy that decodes individuals as trees. So, it evolves trees by the mentioned GA operations. In Burke, Hyde and Kendall (2006)), GP hyperheuristic is applied onto bin-packing problem. Based on an evaluation algorithm, each piece that must be placed into a bin is tried to put into a bin just by checking the suitable one. If it fits, then a new piece is taken and the algorithm performed for the same procedure. Best of run individuals from this study is provided in Figure 2.13.

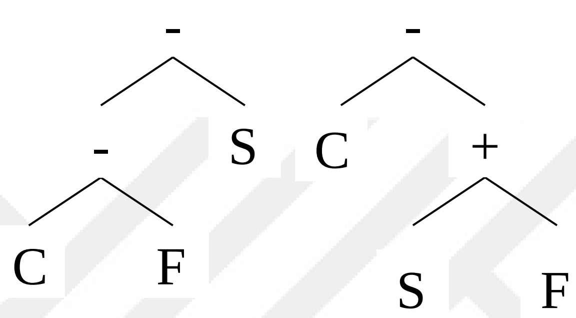


Figure 2.13. Best of Run individuals for Bin Packing Problem (BPP)

C = Bin Capacity, F = Bin Fullness, S = Piece Size

From the figures, following two mathematical equations come out. According to the evaluation algorithm, if they returns a value bigger than zero, then the current piece will be put into a currently checked bin, if not, the next bin will be checked for its availability.

$$(C-F)-S \tag{2.11}$$

$$C-(S+F) \tag{2.12}$$

The important idea here is a human product heuristic can be evolved by a GP hyperheuristic. In the result of the given study, a widely used BPP solver heuristic called first-fit is generated by GP hyperheuristic.

2.3. Application Areas

Applications of hyperheuristics concentrate on combinatorial optimization and operations research fields. Problems tackled using hyperheuristics vary from theoretical to real world problems, such as, bin packing and timetabling. In the following table some problems that are tried to be solved by hyperheuristics are provided.

Table 2.3. Sample application areas of hyperheuristics

Problem Domain	**Reference(s)**
(University) Course Timetabling	[24, 25, 26, 31, 77, 130]
(University) Exam Timetabling	[17, 18, 26, 31, 59, 86, 129]
Staff Scheduling	[77]
Nurse Rostering/Scheduling	[24]
Bin Packing	[21, 22, 47, 100, 127, 128]
Job-Shop Scheduling	[46, 62, 63]
Open Shop Scheduling	[61]
Project Presentation Scheduling	[23, 45, 78]
Sales Summit Scheduling	[43, 44]
Trainer/Training Scheduling	[40,41, 42, 136]
Channel Assignment	[87, 88]
(Shelf) Space Allocation	[11, 32, 54]
Hybrid Flow Shop	[123, 125]
Strip Packing	[6, 67]
Communication Scheduling	[75]
Component Placement Sequencing	[9]
Orc Quest Problem	[106]
Logistic Domain Problem	[106]
Class Timetabling	[126]
Production Scheduling	[124]
Satisfiability Problem	[10]

3. GROUP DECISION MAKING

3.1. Introduction

3.1.1. Definition

Group decision-making is defined as “the process by which a collective of individuals attempt to reach a required level of consensus on a given issue” in Eliaz, Ray and Razin (2007). This process contains two main phases; discussion between group members and reaching a single group decision. The final outcome requires an agreement based on a specified strategy which is available as decision criteria such as voting via synergy that comes from each individual’s opinion.

Robbins and DeCenzo (2003) describe decision-making process consists of a set of steps to reach a choice. At first, the problem is identified. Then, the factors that are expected to be influential on the decision are listed. Each member of this list should be associated with a specific weight according to its importance, that is, some kind of priority should be established. After that, the alternatives that can meet the requirements are considered. The effect and performance of each alternative strategy is analyzed. Among all the alternatives, the best one is chosen and performed on the given issue. During this process, three main circumstances can be encountered; *certainty*, *uncertainty* and *risk*. From the certainty perspective, all the possible effects of the decisions are known. For uncertainty, there is not enough information about the results of alternatives, then; a risk must be taken to get rid of this uncertainty by associating some probabilistic values.

During the group decision making process, one of four main decision making strategies as classified by Schwartz and Andrew (1994) should be chosen and applied depending on the characteristics of a problem. One of them is the *plop* method. It works by providing different ideas about a subject and arguing them, then accepting one of them. It is very simple and commonly used approach, but it is not appropriate for all types of group decisions. The other one is group decision making under an *authority rule*. It is an obvious strategy and directly related to the power. For instance, in a company, everyone provides some ideas about a subject and discusses their ideas to reach a decision. However, in this strategy the final decision is made by an authorized person, such as, a chairman. Another model for group decision is the *minority rule*. It is similar to the previous case, but here, there is no deep discussion. An authorized person asks whether the idea is accepted or not and the silence of group members is considered to be the acceptance of the proposed idea. In some case, everyone can be allowed to state an opposing idea, but the final decision can be given by a small group of people, such as, the shareholders of a company without other board of members. The last and the most known one is *majority rule* and it can be exemplified with two different approaches. One of them is *voting* and it is a well known system. Everyone votes for a decision, and then the decision that receives the majority of the votes is the final decision. The other majority rule is called *polling*. Voting is performed twice. A discussion session is arranged in between them. If the general opinion is the same as before the discussions, then the idea is accepted.

3.2. Group Decision Making Hyperheuristics

Four different group decision making strategies are proposed as a hyperheuristic move acceptance mechanism: G-AND, G-OR, G-VOT, G-PVO. Each one of these move acceptance mechanisms provides a decision whether the new candidate solution formed after employing the selected heuristic is accepted or not by evaluating the decisions of member move acceptance mechanisms as presented in Figure 3. G-AND and G-OR are biased strategies. G-OR makes an acceptance oriented decision. If the members willing to admit the new solution are in the minority, still, it is accepted. Even if there is a single member that admits the new solution, that member acts as an authority and makes the final decision. On the other hand, G-AND makes a rejection oriented decision. All the member move acceptance mechanisms must be in agreement so that the new solution gets accepted. Even if the members that reject the new solution are in the minority, it is rejected. G-VOT and G-PVO are based on the majority rule. G-VOT is based on the traditional voting scheme. If the number of members that vote for acceptance of the new solution, it is accepted, otherwise it is rejected. G-AND, G-OR and G-VOT act under certainty, whereas G-PVO is modeled favoring uncertainty to a degree using a probabilistic framework while making the final decision. The probability of acceptance of a new solution dynamically changes proportional to the number of members that vote for acceptance within the group at each step in G-PVO. For example, assuming that there are ten members in the group and six of them accept the new solution at a step, then this solution is accepted by G-PVO with a probability of 0.6. None of the group decision making move acceptance criteria requires odd number of members, but it is preferable by G-VOT.

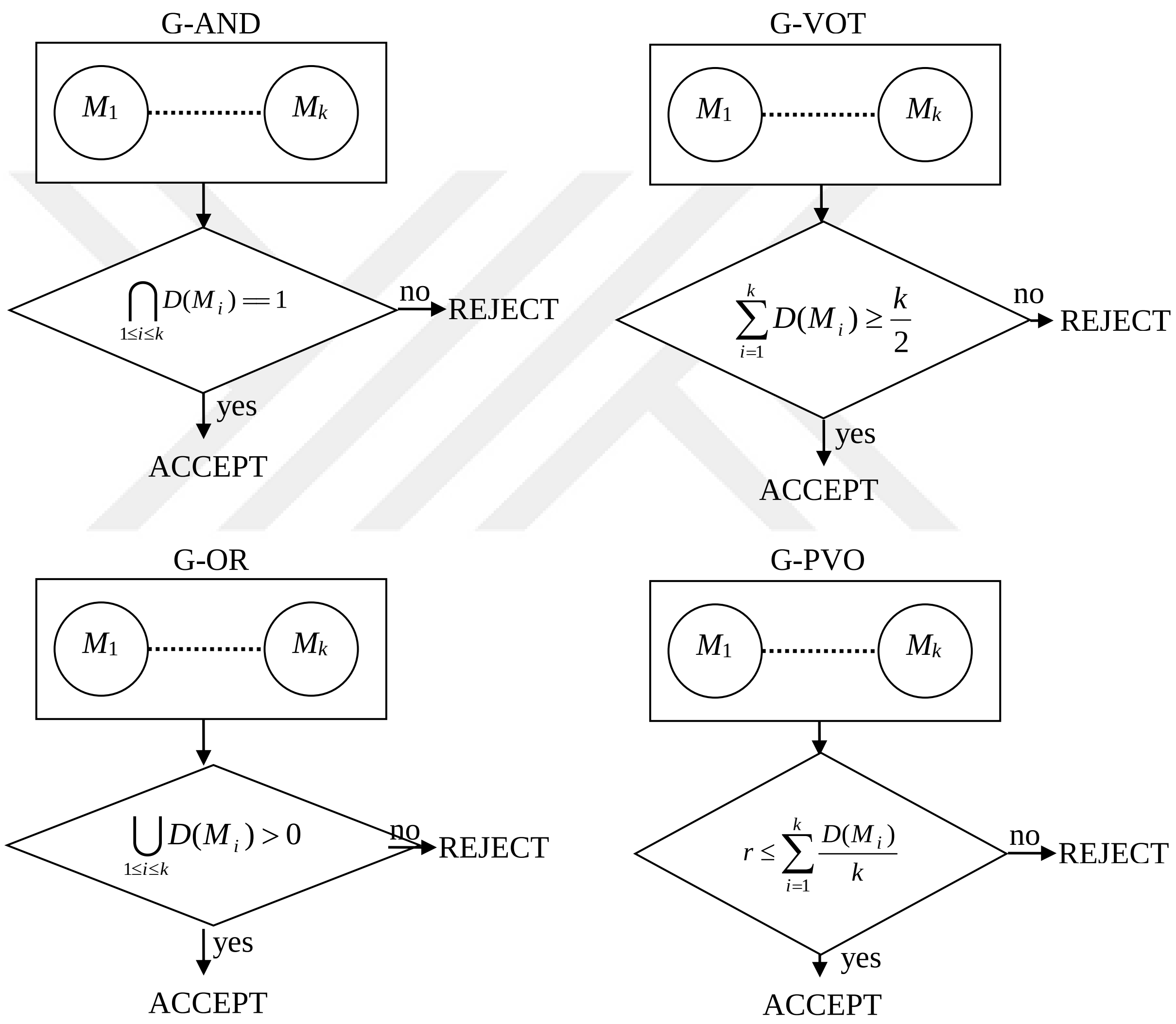


Figure 3.1. Group decision making strategies proposed as single move acceptance mechanisms composed of k members, where M_i denotes the i^{th} member move acceptance mechanism, $D(x)$ returns 1, if the strategy x accepts the new solution and 0, otherwise and r is a uniform random number in [0,1].

4. GROUP DECISION MAKING HYPERHEURISTICS FOR BECHMARK FUNCTION OPTIMIZATION

4.1. Experimental Data and Settings

Fourteen well-known benchmark functions provided in Table 4.1 are used during the initial set of experiments. Using benchmark functions with known characteristics allow researchers to evaluate and compare the performance of their algorithms. The characteristics of each function are summarized in Table 4.2. Binary representation is used for the discrete functions, gray encoding is preferred for the continuous functions. *Royal Road* (F12), *Goldberg's 3 bit Deceptive Function* (F13) and *Whitley's 4 bit Deceptive Function* (F14) are the discrete functions, whereas the rest of the functions are continuous. The deceptiveness of Goldberg and Whitley functions arise due to the large hamming distance between the global optimum and the local optima. Being separable indicates that the overall fitness of a candidate solution can be decomposed into dimensional contributions. This feature is important for an efficient execution, since it allows fast computation of fitness by delta evaluation if a bit (or a set of bits) is flipped in a single dimension. Modality denotes the number of global optima in a given function. Unimodal functions have only a single optimum, while multimodal functions might have multiple global and local optima. It is highly likely that an algorithm gets stuck at a local optimum during the search.

Table 4.1. Benchmark functions used during the experiments

Label	*Formula*	*Source*
F1	$f(\vec{x}) = \sum_{i=1}^{n} x_i^2$	De Jong (1975)
F2	$f(\vec{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	De Jong (1975)
F3	$f(\vec{x}) = 6 \cdot n + \sum_{i=1}^{n} \lfloor x_i \rfloor$	De Jong (1975)
F4	$f(\vec{x}) = \sum_{i=1}^{n} (i \cdot x_i^4 + U(0,1))$	De Jong (1975)

F5

$$f(\vec{x}) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}}$$

$$a_{1j} = \begin{cases} -32 \rightarrow \text{mod}(j,25) = 1 \\ -16 \rightarrow \text{mod}(j,25) = 2 \\ 0 \rightarrow \text{mod}(j,25) = 3 \\ 16 \rightarrow \text{mod}(j,25) = 4 \\ 32 \rightarrow \text{mod}(j,25) = 0 \end{cases}$$

$$a_{2j} = \begin{cases} -32 \rightarrow j > 0 \wedge j \le 5 \\ -16 \rightarrow j > 5 \wedge j \le 10 \\ 0 \rightarrow j > 10 \wedge j \le 15 \\ 16 \rightarrow j > 15 \wedge j \le 20 \\ 32 \rightarrow j > 20 \wedge j \le 5 \end{cases}$$

De Jong (1975)

F6 $$f(\vec{x}) = 10 \cdot n + \sum_{i=1}^{n} (x_i^2 - 10 \cdot \cos(2\pi x_i))$$ Rastrigin (1974)

F7 $$f(\vec{x}) = 418.9829 \cdot n + \sum_{i=1}^{n} x_i \cdot \sin(\sqrt{|x_i|})$$ Schwefel (1981)

F8 $$f(\vec{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod \cos(\frac{x_i}{\sqrt{i}}) + 1$$ Griewangk (1981)

F9 $$f(\vec{x}) = 20 + e - 20 \cdot e^{-0.2 \cdot \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)}$$ Ackley (1987)

F10 $$f(\vec{x}) = -(\prod_{i=1}^{n} \cos(x_i)) \cdot (e^{-\sum_{i=1}^{n} (x_i - \pi)^2})$$ Easom (1990)

F11 $$f(\vec{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j^2)$$ Schwefel (1981)

F12

$$f(\vec{x}) = \sum_{s \in S} order(s) \sigma_s(\vec{x}),$$

where $\sigma_s(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \text{ is an instance of } s \\ 0 & \text{otherwise} \end{cases}$

and s is a schema

Mitchell (1997)

F13

String	000	001	010	011
Value	1	3	3	8
String	100	101	110	111
Value	5	8	8	0

$$f(\vec{x}) = \sum_{i=1}^{n} Value(x_i),$$

where x_i is the i^{th} 3-bit string

Goldberg (1989a, 1989b)

F14

String	0000	0001	0010	0011
Value	2	4	6	12
String	0100	0101	0110	0111
Value	8	14	16	30

Whitley (1991)

String	1000	1001	1010	1011
Value	10	18	20	28
String	1100	1101	1110	1111
Value	22	26	24	0

$$f(\vec{x}) = \sum_{i=1}^{n} Value(x_i) \,,$$

where x_i is the i^{th} 4-bit string

Table 4.2. Characteristics of the benchmark functions used during the experiments

label	*range of x_i*	*dimension*	*optimum*	*isContinuous*	*isMultimodal*	*isSeparable*
F1	[-5.12,5.12]	10	0	✓	×	✓
F2	[-2.048,2.048]	10	0	✓	×	✓
F3	[-5.12,5.12]	10	0	✓	×	✓
F4	[-1.28,1.28]	10	1	✓	✓	✓
F5	[-65.536,65.536]	2	1	✓	✓	×
F6	[-5.12,5.12]	10	0	✓	✓	✓
F7	[-500,500]	10	0	✓	✓	✓
F8	[-600,600]	10	0	✓	✓	×
F9	[-32.768,32.768]	10	0	✓	✓	×
F10	[-100,100]	6	-1	✓	×	×
F11	[-65.536,65.536]	10	0	✓	×	×
F12	n/a	8	0	×	n/a	✓
F13	n/a	30	0	×	n/a	✓
F14	n/a	6	0	×	n/a	✓

Two sets of experiments are performed. During the initial experiments, twenty eight hyperheuristic patterns are evaluated using all six heuristics under each group decision making hyperheuristics within the traditional framework. During the second set of experiments, all the experiments are repeated while the traditional framework is replaced by the F_C framework. Within this framework, a hyperheuristic manages three mutational heuristics and DBHC is employed after each application of a mutational heuristic.

During the experiments, Pentium IV 3 GHz LINUX machines having 2 Gb memories are used. Fifty runs are performed during each test on a benchmark function. For a fair comparison between all algorithms, the experiments are terminated if the execution time exceeds 600 CPU seconds or the expected global optimum is achieved. *Success rate, s.r.*, denotes the ratio of successful runs in which the expected fitness is achieved to the total number of runs.

4.2. Hyperheuristic Patterns used During Benchmark Function Experiments

Six different heuristics are realized for the experiments. Seven heuristic selection methods {SR, RD, RP, RPD, CF, GR, TABU} are combined with four group decision making move acceptance mechanisms {G-AND, G-OR, G-VOT, G-PVO}, generating twenty eight hyperheuristics. These move acceptance mechanisms embed IE, SA and GD as group members. All three methods are the top methods obtained as a result of the experiments performed in Ozcan, Bilgin and Korkmaz (2008). Furthermore, each member move acceptance mechanism is an instance from a different category as previously presented in Table 2.1.

4.3. Heuristics for Benchmark Function Optimization

Half of the heuristics are mutational heuristics, namely; *mutation* (MUTN), *dimensional mutation* (DIMM) and *swap dimension* (SWPD). MUTN is the traditional mutation used in the genetic algorithms. A bit is flipped with probability of 1/*len*, where *len* is the length of a configuration representing candidate solutions. DIMM perturbs all the bits along a randomly selected dimension. SWPD selects two different dimensions in a candidate solution randomly and then swaps their contents. The rest of the heuristics are hill climbers: *random mutation hill climber* (RMHC), *next gradient hill climber* (NGHC), *Davis's bit hill climber* (DBHC). RMHC flips a randomly selected bit at each step and repeats this process until a maximum number of steps is exceeded. NGHC processes each bit in a given candidate solution at each step, consecutively, starting from the most significant bit going towards the least. If there is an improvement in the quality of the candidate solution when the bit in question is inverted, then the move is accepted, otherwise it is rejected. The hill climbing process continues from the next bit. DBHC adapts the same process as in NGHC at each step. The only difference is that the bits representing a candidate solution are inverted in DBHC, successively with respect to a sequence that is a random permutation of the bit locations.

4.4. Experimental Results and Comparisons

Table 4.3 presents the success rate of each hyperheuristic for all benchmark functions within the traditional hyperheuristic framework. As a group decision making move acceptance mechanism, G-VOT performs the best considering the average success rate over all test cases. G- PVO, G-AND and G-OR follows G-VOT performance-wise in that order as illustrated in Figure 4. The performance variance between the majority rule move acceptance mechanisms and G-OR is significant based on the student's two-tailed paired t-test within a confidence interval of %97. CF as a heuristic selection mechanism performs slightly better than the rest of the heuristic selection mechanism with an average success rate of 0.78 over all experiments. The rest of the heuristic selection methods have comparable performances. The CF_G-VOT hyperheuristic performs the best with an average success rate of 0.92 over all benchmark functions, beating the performance of each member hyperheuristic when used as a single approach. CF_IE, CF_GD and CF_MC hyperheuristics generate an average success rate of 0.69, 0.88 and 0.91, respectively. CF_G-VOT achieves a success rate that is greater or equal to 0.96 for F4, F6 and F10 functions. Full success is obtained in locating the global optimum for all functions, excluding F13 during the runs. This hyperheuristic is obviously susceptible to deception. The global optimum is not found for Goldberg's deceptive function (F13) in none of the runs. On the other hand, G-AND locates the global optimum for F13 at least for once during the runs when combined with any heuristic selection method.

Table 4.3. Performance of each group decision making hyperheuristic over benchmark functions based on success rate. "G-" prefix is omitted from the names of the acceptance criteria.

label	SR_AND	SR_OR	SR_PVO	SR_VOT	RD_AND	RD_OR	RD_PVO
F1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F3	1.00	0.84	1.00	1.00	1.00	0.84	1.00
F4	0.24	0.52	0.80	0.92	0.14	0.26	0.54
F5	0.90	1.00	1.00	1.00	0.02	1.00	1.00
F6	1.00	0.00	0.96	1.00	1.00	0.02	1.00
F7	1.00	0.74	1.00	1.00	1.00	0.92	1.00

F8	0.06	1.00	1.00	1.00	0.04	1.00	1.00
F9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F10	1.00	0.88	0.96	1.00	1.00	0.88	0.94
F11	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F12	1.00	0.00	1.00	1.00	1.00	0.00	1.00
F13	0.02	0.00	0.00	0.00	0.00	0.00	0.00
F14	1.00	0.46	1.00	1.00	1.00	1.00	1.00
avr.	0.66	0.53	0.77	0.78	0.59	0.57	0.75
std.	0.46	0.44	0.42	0.42	0.50	0.47	0.42
label	RD_VOT	RP_AND	RP_OR	RP_PVO	RP_VOT	RPD_AND	RPD_OR
F1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F3	1.00	1.00	0.76	1.00	1.00	1.00	0.56
F4	0.52	0.20	0.50	0.84	0.84	0.18	0.68
F5	0.94	0.98	1.00	1.00	1.00	0.96	1.00
F6	1.00	1.00	0.00	0.96	1.00	1.00	0.00
F7	1.00	1.00	0.70	1.00	1.00	1.00	0.52
F8	1.00	0.08	1.00	1.00	1.00	0.02	1.00
F9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F10	1.00	1.00	0.94	0.96	1.00	1.00	0.96
F11	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F12	1.00	1.00	0.00	1.00	1.00	1.00	0.00
F13	0.00	0.04	0.00	0.00	0.00	0.04	0.00
F14	1.00	1.00	0.10	1.00	1.00	1.00	0.08
avr.	0.75	0.66	0.50	0.77	0.77	0.66	0.49
std.	0.42	0.47	0.46	0.42	0.42	0.47	0.45
label	RPD_PVO	RPD_VOT	CF_AND	CF_OR	CF_PVO	CF_VOT	GR_AND
F1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F2	0.00	0.04	0.00	1.00	1.00	1.00	0.00
F3	1.00	1.00	1.00	0.78	1.00	1.00	1.00
F4	0.84	0.84	0.54	0.64	0.92	0.96	0.16
F5	1.00	1.00	1.00	1.00	1.00	1.00	0.90
F6	0.84	1.00	1.00	0.04	0.48	0.96	1.00
F7	1.00	1.00	1.00	0.50	1.00	1.00	1.00
F8	1.00	1.00	0.04	1.00	1.00	1.00	0.12
F9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F10	0.92	1.00	1.00	0.96	0.92	0.98	1.00
F11	0.00	0.00	0.02	1.00	1.00	1.00	0.00
F12	1.00	1.00	1.00	0.00	1.00	1.00	1.00

F13	0.00	0.00	0.00	0.00	0.00	0.00	0.02
F14	1.00	1.00	1.00	0.00	1.00	1.00	1.00
avr.	0.76	0.78	0.69	0.64	0.88	0.92	0.66
std.	0.41	0.42	0.46	0.44	0.29	0.27	0.46
label	GR_OR	GR_PVO	GR_VOT	TABU_AND	TABU_OR	TABU_PVO	TABU_VOT
F1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F2	0.00	0.00	0.00	0.00	0.02	0.06	0.12
F3	1.00	1.00	1.00	1.00	0.54	1.00	1.00
F4	0.36	0.26	0.26	0.14	0.78	0.86	0.86
F5	0.88	0.88	0.90	0.98	1.00	1.00	1.00
F6	1.00	1.00	1.00	1.00	0.00	0.70	0.98
F7	1.00	1.00	1.00	1.00	0.18	1.00	1.00
F8	0.14	0.12	0.12	0.06	1.00	1.00	1.00
F9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F10	1.00	1.00	1.00	1.00	0.90	0.88	1.00
F11	0.04	0.00	0.00	0.00	0.00	0.06	0.00
F12	1.00	1.00	1.00	1.00	0.00	1.00	1.00
F13	0.04	0.02	0.00	0.08	0.00	0.00	0.00
F14	1.00	1.00	1.00	1.00	0.08	1.00	1.00
avr.	0.68	0.66	0.66	0.66	0.46	0.75	0.78
std.	0.44	0.46	0.46	0.47	0.46	0.40	0.41

The same trials with all the decision making hyperheuristics are repeated using the most successful framework F_C (Ozcan, Bilgin and Korkmaz (2007)) instead of the traditional one in the second set of experiments. The results show that almost in all cases, group decision making hyperheuristics when used in the F_C framework generates a better performance. Figure 4.1 illustrates an overall evaluation. Although G-AND turns out to be the best, its performance variation is not significant as compared to G-PVO and G-VOT. G-OR in the F_C framework worsens. As a result, it is observed that majority towards an agreement of acceptance is more valuable among the group members. The heuristic selection methods starting from the one having the best performance to the worst is GR, CF, TABU, RPD, SR, RP and RD, respectively. The best performing hyperheuristic from the previous set of experiments CF-G_VOT improved its success rate from 0.92 to 0.99 by this framework modification. The hyperheuristics GR_G-PVO and GR_G-VOT when used in the F_C framework generate full success (1.00) over all benchmark functions. It is reported in Ozcan, Bilgin and Korkmaz (2008) that the best performing memetic algorithm has generated full success in locating the global optimum in each benchmark function. CF_IE in F_C has generated a slightly worse performance as compared to the MA, yet the difference is reported to be statistically insignificant. It is observed that the performances of GR_G-PVO and GR_G-VOT turn out to be entirely comparable to the memetic algorithms.

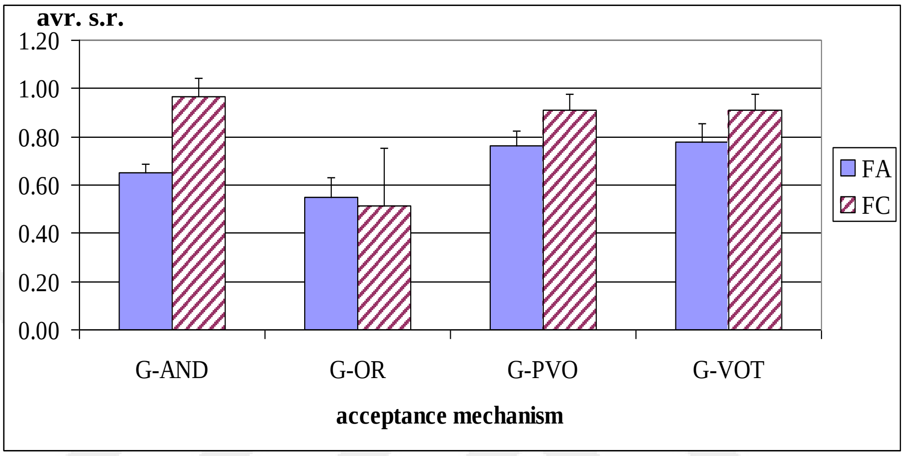


Figure 4.1. Average success rate of each group decision making acceptance mechanism over all benchmark function experiments when used within the traditional hyperheuristic framework (F_A) and the F_C framework.

5. EXAMINATION TIMETABLING

5.1. Introduction

In timetabling problems, an optimal schedule is searched for a given set of events and resources subject to a set of constraints. Two different types of constraints are identified: *hard* and *soft constraints*. Hard constraints are required to be satisfied, while soft constraints represent the preferences that should be resolved as many as possible. The solutions in which no hard constraints are violated are called *feasible*. The size of the search space for a timetabling problem might increase exponentially as the number of items to be scheduled increases. Moreover, the search landscape might contain many *infeasible* regions due to the constraints. Timetabling problems are known to be NP-complete constraint optimization problems (Even, Itai and Shamir (1976)). Hence, an optimal solution might not be obtained by a traditional approach. Many researchers have been employing many different non-standard methods to solve many different type of timetabling problems (Burke and Petrovic (2002)). In this paper, an examination timetabling problem is used as a case study for testing the performance of group decision making hyperheuristics.

5.2. Literature Survey

The first studies started with some computer based strategies for examination timetabling, developed by Cole (1964) and Broder (1964). Then, Wood (1968) designed a large university examination timetabling system. Foxley and Lockyer (1968) provided a strategy to solve examination timetabling problems of some specific universities. In 1967, Welsh and Powell gave a new direction to timetabling research by using graph colouring methods as a solution method. Wood (1969) provided a comprehensive paper that pointed out the similarities of scheduling and graph colouring, then applied a graph colouring based approach for solving large timetabling problems. Carter (1986) provided a survey on real-world applications of timetabling in different universities and described how to design specific timetabling algorithms for each institution separately. In Arani and Lotfi (1989) proposed a three-phase approach that included assigning final exams into separate blocks, assigning the blocks into days and optimizing the relationship between the blocks and the days. Then, Lotfi and Cerveny (1991) modified this approach by adding fourth phase of assigning final exams into classrooms in an efficient way.

Carter, Laporte and Lee (1996) applied different heuristic orderings based on graph colouring, since, Leighton (1979) showed that the timetabling problem can be reduced to a graph colouring problem. Moreover, Carter provided some widely used benchmark data to analyze any timetabling optimization algorithms. Burke, Newall, and Weare (1995) and Burke et al. (1996) applied a light or a heavy mutation, randomly selecting one mutation operator which is followed by a hill-climbing among them. Marin (1998) provided constraint satisfaction strategies combined with genetic algorithms for solving examination timetabling problems. Burke and Newall (1999), proposed an effective use of evolutionary algorithms by dividing a large-scale problem into smaller instances and solving these instances, separately. Gaspero and Schaerf (2000) tested some tabu search based algorithms with graph colouring heuristics and provided some experimental results in a comprehensive manner considering the previous works (Burke, Newall and Weare (1995), Carter, Laporte and Lee (1996), Burke and Newall (1999)) on the same set of examination timetabling data. Paquete and Fonseca (2001) designed a multi-objective evolutionary algorithm (MOEA) based on a direct encoding of the mapping between exams and time slots. The approach attempted to minimize the number of violations of each type of constraints as separate objectives. Wong, Côté and Gely (2002) used a genetic algorithm with a non-elitist replacement strategy to solve a single examination timetabling problem at École de Technologie Supérieure. In their algorithm, genetic operators were applied to an individual first, and then the violations were fixed using a hill-climbing approach. Merlot et al. (2002) proposed a new hybrid algorithm which involved three phases; programming, simulated annealing and hill-climbing. Petrovic, Yang and Dror (2003, 2007) introduced a case based reasoning system to generate initial solutions to be used by a great deluge algorithm. Burke and Newall (2004) proposed a general and fast adaptive method that arranges the heuristic to be used for ordering exams to be scheduled next. Their algorithm produced comparable results on a benchmark of problems with the current state of the art. Asmuni, Burke and Garibaldi (2004) used a fuzzy expert system with some combinations of ordering criteria based on fuzzy weights of exams. Abdullah et al. (2004) proposed a solution improvement technique to make an efficient search over a large set of neighbourhood solutions. Ozcan (2005) proposed a new XML data format which is based on MathML for representing timetabling problems and their solutions. Ozcan and Ersoy (2005) generalized their previous study in Alkan and Ozcan (2003) and proposed an extended framework for designing violation directed adaptive operators. These operators perform a search over the constraint oriented neighbourhoods. A memetic algorithm utilizing such a hill-climber is implemented as a problem solver in a tool called FES. FES is the first tool that supports timetabling markup language (TTML) and accepts input in that format. Petrovic, Patel and Yang (2005) and Asmuni et al. (2006) used fuzzy reasoning for examination timetabling. In Bilgin, Ozcan and Korkmaz (2006) tested a set of hyperheuristics that combine heuristic selection and move acceptance mechanisms over a set of examination timetabling benchmark problems. Eley (2006) provided a detailed performance comparison of two ant colony based approaches called Max-Min and ANTCOL for solving examination timetabling problems. Cheong, Tan and Veeralli (2007) presented a multi-objective evolutionary algorithm (MOEA) that aims to generate feasible exam timetables without any prior knowledge of timetable length. Qu and Burke (2007) worked on an adaptive decomposition approach to divide any given examination timetabling problem into two different sets called *difficult set* and *easy set*, under an ordering mechanism and a construction strategy to combine small parts of the solutions into one. Tounsi and Ouis (2008) proposed a mechanism which is available for diversification using local search algorithms for constraint satisfaction and optimization problems. The operation of escaping from local optima applied onto a real world examination timetabling data set belongs to a French engineering school, Ecole des Mines de Nantes.

5.3. Examination Timetabling Problem

In the real-world examination timetabling problems, the constraints might change from one institution to another. More on exam examination timetabling, such as, their formulations and approaches can be found in the survey provided by Qu et al. (2006). In this study, the formulation from the examination timetabling problem at Yeditepe University Faculty of Architecture and Engineering is used. Hard constraints can be listed as:

- each examination should be scheduled just for once (Equation (5.1)),

- if a student takes more than one exam, then these exams must be assigned to different time periods (Equation (5.2)),
- total number of students taking exams at a time period is not allowed to exceed a seating capacity (Equation (5.3)).

The only soft constraint in this problem is leaving at least one empty slot for the students who have more than one examination in the same day (Equation (5.4)).

$$\forall j, \sum_{i=1}^{M} a_{ij} = 1 \tag{5.1}$$

where M is the number of periods and $a_{ij} = \begin{cases} 1 & \text{if } j^{\text{th}} \text{ exam is in } i^{\text{th}} \text{ period} \\ 0 & else \end{cases}$

$$\forall i, \sum_{j=2}^{N} a_{ij} \sum_{k=1}^{j-1} a_{ik} c_{jk} = 0 \tag{5.2}$$

where N is the number of exams and c_{jk} is the number of students taking both exams j and k

$$\forall i, \sum_{j=1}^{N} a_{ij} b_j \leq C \tag{5.3}$$

where C is the seating capacity and b_j is the number of students taking examination j

$$\forall i \text{, if } i \text{ is not the last period in the day, } \sum_{j=1}^{N} a_{ij} \sum_{k=1}^{N} a_{i+1,k} c_{jk} = 0 \tag{5.4}$$

6. GROUP DECISION MAKING HYPERHEURISTICS FOR EXAMINATION TIMETABLING

6.1. Experimental Data and Settings

Direct encoding is used that represents the mapping of each examination to a period. Hence, the constraint denoted by Equation (5.1) is explicitly satisfied. During the optimization process, candidate solutions are evaluated using Equation (6.1). The evaluation function computes the weighted average of constraint violations. A value calculated using the evaluation function will be referred to as *fitness value* in the rest of the paper. The evaluation function is multiplied by -1 to convert the problem into a minimization problem. In the evaluation function**,** 0.4 is used as the weight for the constraints denoted by Equations 4 and 5 and 0.2 for the constraint denoted by Equation (6.1).

$$eval(T) = \frac{-1}{1 + \sum_{\forall i} w_i v_i(T)} \tag{6.1}$$

where T is a candidate solution, w_i indicates the weight associated with the i^{th} constraint, v_i indicates the number of constraint violations in T due to the i^{th} constraint.

Experiments are performed on Carter (Carter (1996)) and Yeditepe University, Faculty of Architecture and Engineering data sets (Ozcan and Ersoy (2005)) (Table 6.1). During the experiments, Pentium IV 3 GHz LINUX machines having 2 Gb memories are used. Fifty runs are performed during each test on a benchmark function.

Table 6.1. Properties and parameters of the examination timetabling problem instances used in the experiments.

Instance	*Exams*	*Students*	*Enrollment*	*Density*	*Days*	*Capacity*
Carf92	543	18419	54062	0.14	12	2000

Cars91	682	16925	59022	0.13	17	1550
Earf83	190	941	6029	0.27	8	350
Hecs92	81	2823	10634	0.20	6	650
Kfus93	486	5349	25118	0.06	7	1955
Lsef91	381	2726	10919	0.06	6	635
Purs93	2419	30032	120690	0.03	10	5000
Ryes93	486	11483	45051	0.07	8	2055
Staf83	139	611	5539	0.14	4	3024
Tres92	261	4360	14901	0.18	10	655
Utas92	622	21267	58981	0.13	12	2800
Utes92	184	2749	11796	0.08	3	1240
Yorf83	181	1125	8108	0.29	7	300
Yue20011	140	559	3488	0.14	6	450
Yue20012	158	591	3706	0.14	6	450
Yue20013	30	234	447	0.19	2	150
Yue20021	168	826	5757	0.16	7	550
Yue20022	187	896	5860	0.16	7	550
Yue20023	40	420	790	0.19	2	150
Yue20031	177	1125	6716	0.15	6	550
Yue20032	210	1185	6837	0.14	6	550

6.2. Hyperheuristic Patterns used During Examination Timetabling Experiments

The same experimental settings are used from the benchmark function experiments during the evaluation of twenty eight decision making hyperheuristics that is a combination of heuristic selection methods {SR, RD, RP, RPD, CF, GR, TABU} by move acceptance strategies {G-AND, G-OR, G-VOT, G-PVO}. Similarly, IE, SA and GD are the members in all groups. The traditional hyperheuristic framework is used.

6.3. Heuristics for Examination Timetabling Problem

Four different mutational heuristics are implemented; RANDSC, TOURC1, TOURC2 and TOURC3. The last three heuristics aim to prevent each specific constraint type conflict by searching constraint oriented neighbourhoods, while the former one employs random perturbation(s). RANDSC scans a candidate solution and might reassign an examination to a randomly chosen time slot with a probability of (1/*number of exams*). TOURC1-3 employ a tournament based strategy while deciding the examination to reschedule at each step. TOURC1, TOURC2 and TOURC3 heuristics attempt to repair the violations of constraints denoted by Equations 5.1, 5.2 and 5.3, respectively. Each one of these heuristics performs a directed search aiming a possible improvement for a specific constraint type, yet they are not hill climbers. Improving a constraint type does not guarantee an overall improvement, since other violations might arise due to the other constraint types. TOURC1 and TOURC3 select a number of exams and count the number of conflicts due to the corresponding constraint type. A tournament is arranged based on the number of conflicts and the examination with the highest number of conflicts is selected for rescheduling. The examination is assigned to a period from a randomly selected subset of periods that produces the minimum number of conflicts due to the constraint type in question. TOURC2 employs two tournament stages successively. At first, a subset of periods is selected and the capacity violations at each period are measured. After the tournament, the set of exams at the period that causes the maximum number of conflicts is processed. A subset of these exams with a predetermined size is passed through the second tournament process. At the end of this process, the examination with the maximum number of students is rescheduled. This examination is assigned to a period from a randomly selected subset of periods that contains the minimum number of seated students.

6.4. Experimental Results and Comparisons

Proposed group decision making based hyperheuristics are tested over all examination timetabling benchmark problem instances. To make a fair performance comparison and determine significant performance variance, t-test with the conficence interval of 95% is applied. As another methodology to compare the hyberheuristics based on their experimental results, ranking is used for 1 through 4 for each problem instances. Here, 1 indicates that the corresponding hyperheuristic generates the best average fitness over fifty runs as provided in Table 6.2. The move acceptance methods that do not generate significant performance variances over fifty runs are grouped together and the same rank that takes ties into account is assigned to them. Remembering that the traditional framework is used during the experiments, a similar result is obtained for the online performance of the group decision making strategies as in the benchmark functions. G-VOT becomes the best acceptance mechanism considering the average rank over all problems, while G-PVO, G-AND and G-OR follows it in that order, respectively as illustrated in Figure 6.1.

Table 6.2. Performance comparison of the group decision making hyperheuristics over benchmark functions based on rankings. "G-" prefix is omitted from the names of the acceptance criteria.

label	SR_AND	SR_OR	SR_PVO	SR_VOT	RD_AND	RD_OR	RD_PVO
Carf92	2.50	4.00	2.50	1.00	2.00	4.00	2.00
Cars91	2.00	4.00	3.00	1.00	2.00	4.00	3.00
Earf83	3.00	4.00	1.50	1.50	3.00	4.00	1.00
Hecs92	3.00	4.00	1.50	1.50	3.00	4.00	1.00
Kfus93	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Lsef91	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Purs93	2.00	4.00	3.00	2.00	2.00	4.00	3.00
Ryes93	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Staf83	3.00	4.00	1.00	2.00	3.00	4.00	1.50
Tres92	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Utas92	2.00	4.00	3.00	1.00	2.00	4.00	3.00
Utes92	3.00	4.00	1.00	2.00	3.00	4.00	1.00
Yorf83	3.00	4.00	2.00	1.00	3.00	4.00	1.00
Yue20011	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Yue20012	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Yue20013	3.00	3.00	1.00	1.00	4.00	3.00	1.00
Yue20021	3.00	4.00	2.00	1.00	3.00	4.00	2.00
Yue20022	3.00	4.00	1.50	1.50	3.00	4.00	1.50
Yue20023	4.00	3.00	1.00	2.00	4.00	2.00	1.00
Yue20031	3.00	4.00	1.50	1.50	3.00	4.00	1.50
Yue20032	3.00	4.00	1.00	1.00	2.50	4.00	1.00
avr.	2.88	3.90	1.83	1.29	2.88	3.86	1.74
std.	0.44	0.30	0.66	0.41	0.55	0.48	0.68
label	RD_VOT	RP_AND	RP_OR	RP_PVO	RP_VOT	RPD_AND	RPD_OR
Carf92	1.00	3.00	4.00	2.00	1.00	2.50	4.00
Cars91	1.00	2.00	4.00	3.00	1.00	2.00	4.00
Earf83	2.00	3.00	4.00	1.50	1.50	3.00	4.00

Hecs92	3.00	3.00	4.00	1.00	2.00	3.00	4.00
Kfus93	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Lsef91	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Purs93	1.00	1.50	4.00	3.00	1.50	2.00	4.00
Ryes93	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Staf83	1.50	3.00	4.00	1.00	2.00	3.00	4.00
Tres92	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Utas92	1.00	2.00	4.00	3.00	1.00	2.00	4.00
Utes92	2.00	3.00	4.00	1.00	2.00	3.00	4.00
Yorf83	2.00	3.00	4.00	2.00	1.00	3.00	4.00
Yue20011	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Yue20012	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Yue20013	4.00	3.00	4.00	1.00	2.00	3.00	4.00
Yue20021	1.00	3.00	4.00	2.00	1.00	3.00	4.00
Yue20022	1.50	3.00	4.00	1.50	1.50	3.00	4.00
Yue20023	3.00	4.00	3.00	1.00	2.00	4.00	3.00
Yue20031	1.50	3.00	4.00	1.50	1.50	3.00	4.00
Yue20032	2.50	3.00	4.00	1.00	2.00	3.00	4.00
avr.	1.62	2.88	3.95	1.79	1.38	2.88	3.95
std.	0.86	0.50	0.22	0.66	0.44	0.44	0.22
label	RPD_PVO	RPD_VOT	CF_AND	CF_OR	CF_PVO	CF_VOT	GR_AND
Carf92	2.50	1.00	2.50	4.00	2.50	1.00	3.00
Cars91	3.00	1.00	2.00	4.00	3.00	1.00	2.00
Earf83	1.50	1.50	3.00	4.00	1.50	1.50	3.00
Hecs92	1.00	2.00	3.00	4.00	1.00	2.00	3.50
Kfus93	2.00	1.00	3.00	4.00	2.00	1.00	3.00
Lsef91	2.00	1.00	3.00	4.00	2.00	2.00	3.00
Purs93	3.00	1.00	2.00	4.00	3.00	2.00	3.00
Ryes93	2.00	1.00	2.50	4.00	2.50	1.00	3.00
Staf83	1.00	2.00	3.00	4.00	1.00	2.00	3.00
Tres92	2.00	1.00	3.00	4.00	2.00	1.00	3.00
Utas92	3.00	1.00	2.00	4.00	3.00	1.00	3.00
Utes92	1.00	2.00	3.00	4.00	1.00	2.00	4.00
Yorf83	2.00	1.00	3.00	4.00	2.00	2.00	3.00
Yue20011	2.00	1.00	3.00	4.00	2.00	1.00	3.00
Yue20012	2.00	1.00	3.00	4.00	2.00	1.00	3.00
Yue20013	1.00	2.00	3.00	4.00	1.00	2.00	4.00
Yue20021	2.00	1.00	3.00	4.00	2.00	2.00	3.00
Yue20022	1.50	1.50	3.00	4.00	1.50	1.50	3.00

Yue20023	1.00	2.00	4.00	3.00	1.00	3.00	4.00
Yue20031	1.50	1.50	3.00	4.00	1.50	1.50	3.00
Yue20032	1.00	2.00	3.00	4.00	1.00	2.00	4.00
avr.	1.81	1.36	2.86	3.95	1.83	1.60	3.17
std.	0.68	0.45	0.45	0.22	0.70	0.56	0.48
label	GR_OR	GR_PVO	GR_VOT	TABU_AND	TABU_OR	TABU_PVO	TABU_VOT
Carf92	4.00	2.00	1.00	3.50	4.00	3.50	2.00
Cars91	4.00	3.00	1.00	2.00	4.00	4.00	2.00
Earf83	4.00	1.00	1.00	3.00	4.00	1.50	1.50
Hecs92	3.50	1.00	1.00	3.00	4.00	2.00	3.00
Kfus93	4.00	2.00	1.00	3.00	4.00	3.00	2.00
Lsef91	4.00	1.00	1.00	3.00	4.00	3.00	2.00
Purs93	4.00	1.00	1.00	2.00	4.00	4.00	2.00
Ryes93	4.00	1.50	1.50	3.00	4.00	4.00	2.00
Staf83	4.00	1.00	1.00	3.00	4.00	1.50	1.50
Tres92	4.00	2.00	1.00	3.00	4.00	4.00	2.00
Utas92	4.00	2.00	1.00	2.00	4.00	4.00	2.00
Utes92	3.00	1.00	2.00	3.00	4.00	2.00	2.00
Yorf83	4.00	1.50	1.50	3.00	4.00	4.00	2.00
Yue20011	4.00	2.00	1.00	3.00	4.00	3.00	2.00
Yue20012	4.00	2.00	1.00	3.00	4.00	4.00	2.00
Yue20013	3.00	1.00	2.00	3.00	4.00	1.00	2.00
Yue20021	4.00	1.50	1.50	3.00	4.00	3.00	1.00
Yue20022	4.00	1.00	1.00	3.00	4.00	1.50	1.50
Yue20023	2.00	1.00	2.00	4.00	4.00	2.00	3.00
Yue20031	4.00	1.50	1.50	3.00	4.00	3.00	2.00
Yue20032	3.00	1.00	2.00	3.00	4.00	2.00	2.00
avr.	3.74	1.48	1.29	2.93	4.00	2.86	1.98
std.	0.54	0.56	0.41	0.46	0.00	1.04	0.43

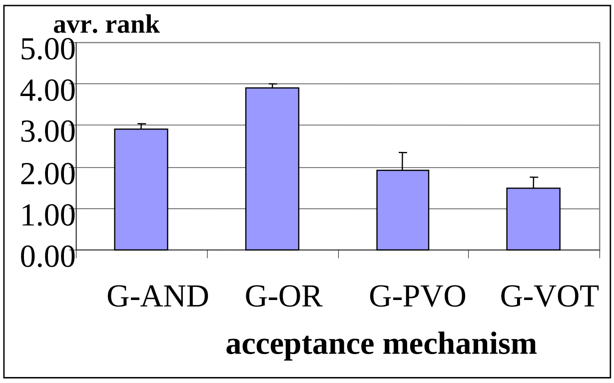


Figure 6.1. Average rank of each group decision making move acceptance mechanism over all examination timetabling experiments provided in Table 6.2.

In order to evaluate the offline performance of the approaches, twenty eight hyperheuristics are ranked from 1 to 28 considering the best fitness produced in fifty runs for each problem, where 1 indicates that the corresponding hyperheuristic provides the best value. Figure 6.2 illustrates six hyperheuristics that deliver a better average performance that are statistically significant considering the ranks as compared to the rest, namely; GR_G-VOT, TABU_G-VOT, RP_G-VOT, GR_G-PVO, SR_G-VOT and CF_G-VOT. GR_G-VOT has the best performance.

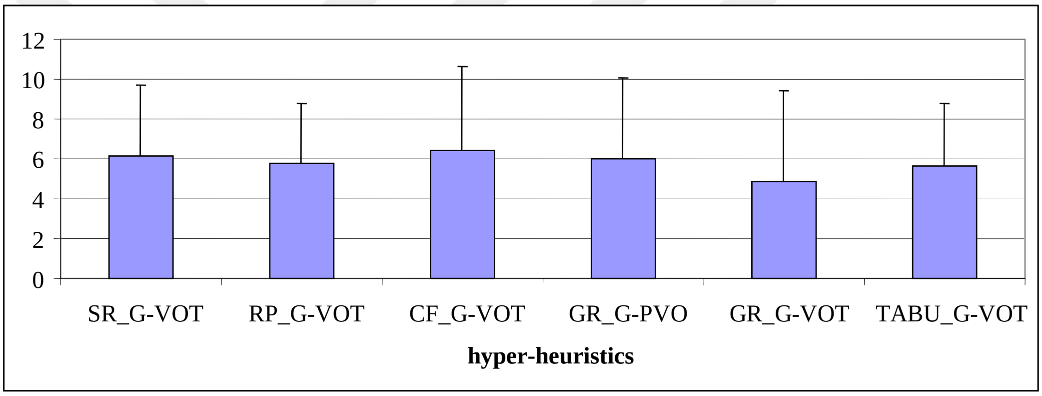


Figure 6.2. Average rank of group decision making hyperheuristics that generate statistically significant performance variance from the rest over all examination timetabling problems.

The average best fitness value and its standard deviation of the best performing heuristic selection-acceptance criterion combination(s) among the top six group decision making hyperheuristics (see Figure 6.2) for each benchmark problem instance are provided in Table 6.3. Moreover, a comparison to a previous study (Bilgin, Ozcan and Korkmaz (2006)) is presented over the same data set. Hyperheuristics utilizing multiple move acceptance criteria under decision making models generated a superior performance as compared to the hyperheuristics where each utilizes a single move acceptance method within. This performance variation is statistically significant. In eleven out of the twenty one problems, hyperheuristics utilizing VOT and PVO delivers the best performances.

Moreover, at least one of the top six group decision making hyperheuristic provide a matching performance to the previous best hyperheuristics for the rest of the problems.

Table 6.3. Comparison of the previous results obtained in Bilgin, Ozcan and Korkmaz (2006) and the current results obtained during this study. Bold entries mark the best performing hyperheuristic. If a group decision making hyperheuristic delivers a statistically significant performance, it appears in the "*Current*" column. "+" indicates that all hyperheuristics in {GR_G-VOT, TABU_G-VOT, RP_G-VOT, GR_G-PVO, SR_G-VOT, CF_G-VOT} has similar performance. "\" excludes the hyperheuristic from this set that is displayed afterwards. "-" shows that there is at least one group decision making hyperheuristic that has a matching performance to the one that appears in the "*Previous*" column. The hyperheuristics that have a similar performance to the bold entry are displayed within parentheses.

Instance	(*Av. B. Fit., Std. Dev.*)	*Current*	*Previous*
Carf92	(-1.85E-02, 1.54E-03)	**GR_G-VOT**+	TABU_IE
Cars91	(-5.73E-01, 2.02E-01)	**GR_G-VOT**+\ GR_G-PVO	TABU_IE
Earf83	(-7.35E-03, 4.38E-04)	**GR_G-PVO** (GR_G-VOT)	CF_MC
Hecs92	(-2.66E-02, 4.97E-03)	**GR_G-PVO** (GR_G-VOT, SR_G-VOT, TB_G-VOT)	CF_MC
Kfus93	(-4.45E-02, 3.26E-03)	-	**SR_GD**
Lsef91	(-1.61E-02, 1.88E-03)	**GR_G-PVO**+	CF_MC
Purs93	(-1.63E-03, 9.71E-05)	**GR_G-PVO** (SR_G-VOT)	SR_IE
Ryes93	(-1.53E-02, 2.25E-03)	**TABU_G-VOT**+	CF_MC
Staf83	(-2.68E-03, 1.45E-05)	-	**SR_MC**
Tres92	(-1.31E-01, 2.49E-02)	-	SR_GD
Utas92	(-2.55E-02, 1.75E-03)	**GR_G-VOT**+\ GR_G-PVO	TABU_IE
Utes92	(-2.27E-03, 7.63E-05)	**GR_G-PVO**	**CF_MC**
Yorf83	(-9.07E-03, 5.84E-04)	**GR_G-PVO**+	CF_MC
Yue20011	(-1.09E-01, 1.19E-02)	-	**SR_GD**
Yue20012	(-9.42E-02, 9.33E-03)	-	**SR_GD**
Yue20013	(-2.50E-01, 0.00E+00)	-	**SR_MC**
Yue20021	(-4.07E-02, 6.02E-03)	-	**SR_GD**
Yue20022	(-1.31E-02, 1.11E-03)	**GR_G-PVO**	CF_MC
Yue20023	(-1.55E-02, 1.34E-04)	**GR_G-PVO**	CF_MC
Yue20031	(-1.66E-02, 1.99E-03)	**GR_G-PVO** (GR_G-VOT, SR_G-VOT)	CF_MC
Yue20032	(-5.02E-03, 4.13E-04)	-	**CF_MC**

In the following table, minimum number of conflicts belongs to hard and soft constraints that could not be solved for Carter's benchmark data for the best hyperheuristics with group decision making strategies based on their best fitness values are provided. Our approaches solved all the constraints only for Cars91 data, the rest of them have some conflicts that were not handled, yet. On the other hand, when we look at the constraints separately, we can see that all the constraints except the hard constraint of *Exam Conflict* are satisfied at least for one time for the examination timetabling data excluding Cars91.

Our aim of this study is to see the effect of group decision making strategies in hyperheuristics and compare their performances. We are not trying to beat the state of the art approaches. In addition, our problem formulation is based on the examination timetabling problem at Yeditepe University; hence the quality of resulting timetables can not be compared to the previous studies.

Table 6.4. Number of conflicts for hard and soft constraints. Numbers in paranthesis are the best (minimum) values among all experimented hyperheuristics (Hard constraint about number of occurrences for each exam is not provided in the table, since, it is solved directly because of the representation that we used)

Instance	*Hyperheuristic*	*Capacity*	*Exam Conflict*	*Empty Slot*
Carf92	**GR_G-VOT**	0	67	56 (50)
Cars91	**GR_G-VOT**	0	0	0
Earf83	**GR_G-VOT**	0	96	354
Hecs92	**GR_G-VOT**	0	37	39
Kfus93	**GR_G-VOT**	0	44	0
Lsef91	**TABU_G-VOT**	0	34	148
Purs93	**GR_G-PVO**	0	834	861
Ryes93	**TABU_G-VOT**	0	61 (47)	50
Staf83	**GR_G-PVO**	0	429	314 (228)
Tres92	**GR_G-VOT**	0	1	9
Utas92	**GR_G-VOT**	0	79	0
Utes92	**GR_G-PVO**	633	53	503
Yorf83	**GR_G-PVO**	0	62	246

7. CONCLUSION AND REMARKS

Simple hyperheuristics combine a heuristic selection method that manages a set of low level heuristics and a move acceptance mechanism in an iterative cycle for search and optimization. Bilgin, Ozcan, Korkmaz (2006) and Ozcan, Bilgin, Korkmaz (2008) show that the move acceptance mechanism might be more influential over the performance compared to the heuristic selection mechanism in a simple hyperheuristic. This phenomenon might be due to the use of a small set of low level heuristics. As the number of low level heuristics used by a simple hyperheuristic increases, it is expected that the heuristic selection component will become more imperative. Focusing back to the move acceptance stage, it is also observed that different move acceptance methods might perform better on different problems. In this study, group decision making acceptance methods that utilize multiple move acceptance strategies are proposed to relieve the difficulty of choosing a move acceptance method to be used within a hyperheuristic for solving a problem. The experimental results show that the majority rule based acceptance methods can improve the performance significantly in some problems. Voting and the probabilistic voting scheme that dynamically computes the acceptance probability based on the votes of group members generate the most successful acceptance mechanisms to be used within the hyperheuristics. It is still observed that, if the mutational and hill climbing heuristics can be distinguished and implemented separately for solving a problem, then an additional improvement can be obtained by using the memetic hyperheuristic framework as proposed in Ozcan, Bilgin and Korkmaz (2006). For some problems, this improvement is comparable to the meta-heuristics, such as, memetic algorithms. Group decision making methods have the potential to generate a synergy in between member acceptance mechanisms yielding a better performance. Proposed group decision making mechanisms can be extended to combine different acceptance mechanisms as group members, hence new group decision making hyperheuristics can be generated.

Considering the performance of heuristic selection mechanisms over the benchmark problems, GR seems to perform the best. GR does not utilize any learning mechanism. In different regions of the search space, a different heuristic might operate the best. A *good* hyperheuristic is expected to recover the most appropriate heuristic to utilize in a given region as rapid as possible. GR employs all heuristics to the same candidate solution simultaneously and selects the best one. It seems that this mechanism allows GR to react faster to the transitions from one region to another. As the number of low level heuristics increase for solving a problem, it is expected that the learning mechanisms exploited by the heuristic selection methods will become more useful. Moreover, RP, RPD and GR heuristic selection methods might become impractical to be used in case of large set of low level heuristics. Ersoy, Ozcan and Uyar (2007) proposed a simple modification of GR that randomly selects a subset of low level heuristics and chooses the one from this subset which generates the best result. This hyperheuristic did not perform well, possibly, because it was used as a mechanism for managing hill climbers only. Moreover, instead of a random heuristic selection, a more informed decision could have been made. Hence, there is still a potential to incorporate a learning mechanism within GR to improve its performance further.

APPENDIX A: EXPERIMENTAL RESULTS TABLES OF HYPERHEURISTICS PATTERNS ON BENCHMARK FUNCTIONS

For 14 mathematical benchmark functions, on following 14 tables, *Best Fitness, Average Best Fitness* and *Average Number of Evaluations per Execution* values are provided for experiments that are performed on FA framework with 28 different hyperheuristics. The hyperheuristics comes from 7 heuristic selection mechanisms which are SR, RD, RPER, RPD, CF, GR, TABU and 4 move acceptance strategies that I proposed as group decision making methods, G-AND,G- OR, G-PVO, G-VOT. Also, related standard deviation values added to the tables and they are added as second column for Avg. Best Fit. And Avg. Num. of Eval. sub-sections.

Table A.1. Results of performance evaluations of hyperheuristic patterns on Sphere Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	0,00E+00	0,00E+00	0,00E+00	1,34E+03	5,92E+02	**100.00%**
SR_OR	0,00E+00	0,00E+00	0,00E+00	1,54E+03	7,51E+02	**100.00%**
SR_PVO	0,00E+00	0,00E+00	0,00E+00	1,42E+03	7,03E+02	**100.00%**
SR_VOT	0,00E+00	0,00E+00	0,00E+00	1,59E+03	8,06E+02	**100.00%**
RD_AND	0,00E+00	0,00E+00	0,00E+00	1,39E+03	6,49E+02	**100.00%**
RD_OR	0,00E+00	0,00E+00	0,00E+00	1,66E+03	7,23E+02	**100.00%**
RD_PVO	0,00E+00	0,00E+00	0,00E+00	1,51E+03	1,06E+03	**100.00%**
RD_VOT	0,00E+00	0,00E+00	0,00E+00	1,58E+03	8,38E+02	**100.00%**
RP_AND	0,00E+00	0,00E+00	0,00E+00	1,18E+03	1,96E+02	**100.00%**
RP_OR	0,00E+00	0,00E+00	0,00E+00	1,43E+03	3,83E+02	**100.00%**
RP_PVO	0,00E+00	0,00E+00	0,00E+00	1,33E+03	3,21E+02	**100.00%**
RP_VOT	0,00E+00	0,00E+00	0,00E+00	1,38E+03	3,43E+02	**100.00%**
RPD_AND	0,00E+00	0,00E+00	0,00E+00	4,29E+03	3,26E+03	**100.00%**
RPD_OR	0,00E+00	0,00E+00	0,00E+00	4,30E+03	3,26E+03	**100.00%**
RPD_PVO	0,00E+00	0,00E+00	0,00E+00	4,38E+03	2,83E+03	**100.00%**

RPD_VOT	0,00E+00	0,00E+00	0,00E+00	4,30E+03	3,26E+03	**100.00%**
CF_AND	0,00E+00	0,00E+00	0,00E+00	3,27E+03	2,61E+03	**100.00%**
CF_OR	0,00E+00	0,00E+00	0,00E+00	3,35E+03	2,66E+03	**100.00%**
CF_PVO	0,00E+00	0,00E+00	0,00E+00	2,90E+03	2,56E+03	**100.00%**
CF_VOT	0,00E+00	0,00E+00	0,00E+00	3,70E+03	2,75E+03	**100.00%**
GR_AND	0,00E+00	0,00E+00	0,00E+00	1,81E+03	0,00E+00	**100.00%**
GR_OR	0,00E+00	0,00E+00	0,00E+00	1,81E+03	0,00E+00	**100.00%**
GR_PVO	0,00E+00	0,00E+00	0,00E+00	1,81E+03	0,00E+00	**100.00%**
GR_VOT	0,00E+00	0,00E+00	0,00E+00	1,81E+03	0,00E+00	**100.00%**
TABU_AND	0,00E+00	0,00E+00	0,00E+00	5,08E+03	2,87E+03	**100.00%**
TABU_OR	0,00E+00	0,00E+00	0,00E+00	5,03E+03	2,70E+03	**100.00%**
TABU_PVO	0,00E+00	0,00E+00	0,00E+00	5,08E+03	2,97E+03	**100.00%**
TABU_VOT	0,00E+00	0,00E+00	0,00E+00	6,29E+03	2,92E+03	**100.00%**

Table A.2. Results of performance evaluations of hyperheuristic patterns on Rosenbrock Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	8,59E-13	1,03E-12	8,96E-14	6,53E+08	1,28E+07	**0.00%**
SR_OR	3,21E-11	1,78E-09	2,32E-09	6,50E+08	5,04E+06	**0.00%**
SR_PVO	2,69E-15	8,76E-12	1,34E-11	6,38E+08	2,30E+07	**0.00%**
SR_VOT	3,64E-15	4,52E-13	7,64E-13	6,56E+08	4,68E+06	**0.00%**
RD_AND	5,74E-03	7,18E+00	1,31E+01	1,77E+08	1,54E+05	**0.00%**
RD_OR	7,28E-11	9,64E-09	1,06E-08	6,34E+08	2,13E+05	**0.00%**
RD_PVO	1,31E-12	1,93E-10	2,02E-10	6,35E+08	4,08E+05	**0.00%**
RD_VOT	4,14E-14	3,19E-11	9,07E-11	6,37E+08	4,68E+05	**0.00%**
RP_AND	8,63E-13	1,04E-12	9,28E-14	6,29E+08	2,06E+07	**0.00%**
RP_OR	6,49E-10	1,17E-08	9,45E-09	6,53E+08	1,49E+05	**0.00%**
RP_PVO	4,50E-14	2,43E-11	5,66E-11	6,55E+08	4,36E+06	**0.00%**
RP_VOT	1,76E-15	1,03E-12	3,06E-12	6,61E+08	3,26E+05	**0.00%**
RPD_AND	8,63E-13	1,04E-12	7,65E-14	6,58E+08	1,33E+05	**0.00%**
RPD_OR	4,03E-28	3,02E-14	3,97E-14	6,60E+08	4,76E+07	**0.00%**

RPD_PVO	1,47E-15	1,26E-14	1,73E-14	6,83E+08	3,21E+07	**0.00%**
RPD_VOT	4,03E-28	8,94E-15	1,03E-14	6,93E+08	3,87E+07	**0.00%**
CF_AND	7,42E-15	8,66E-13	3,70E-13	6,53E+08	1,21E+07	**0.00%**
CF_OR	4,03E-28	4,03E-28	2,72E-43	6,63E+08	1,50E+07	**0.00%**
CF_PVO	4,03E-28	4,03E-28	2,72E-43	6,74E+08	4,92E+06	**0.00%**
CF_VOT	4,03E-28	4,03E-28	2,72E-43	6,70E+08	1,56E+07	**0.00%**
GR_AND	8,29E-13	1,03E-12	7,82E-14	6,59E+08	1,64E+07	**0.00%**
GR_OR	8,29E-13	1,02E-12	9,35E-14	6,70E+08	1,93E+05	**0.00%**
GR_PVO	8,29E-13	1,02E-12	8,02E-14	6,64E+08	4,34E+06	**0.00%**
GR_VOT	7,48E-13	1,04E-12	1,01E-13	6,64E+08	1,31E+07	**0.00%**
TABU_AND	8,42E-13	1,03E-12	8,63E-14	6,57E+08	9,74E+04	**0.00%**
TABU_OR	4,03E-28	5,88E-15	4,85E-15	6,87E+08	2,44E+06	**0.00%**
TABU_PVO	4,03E-28	3,80E-15	3,43E-15	6,86E+08	5,23E+06	**0.00%**
TABU_VOT	4,03E-28	3,16E-15	2,20E-15	6,88E+08	2,25E+06	**0.00%**

Table A.3. Results of performance evaluations of hyperheuristic patterns on Step Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	0,00E+00	0,00E+00	0,00E+00	7,05E+05	3,78E+05	**100,00 %**
SR_OR	0,00E+00	1,60E-01	3,70E-01	2,23E+08	1,69E+08	**84,00 %**
SR_PVO	0,00E+00	0,00E+00	0,00E+00	1,59E+08	1,11E+08	**100,00 %**
SR_VOT	0,00E+00	0,00E+00	0,00E+00	1,85E+08	1,20E+08	**100,00 %**
RD_AND	0,00E+00	0,00E+00	0,00E+00	4,94E+05	2,27E+05	**100,00 %**
RD_OR	0,00E+00	1,60E-01	3,70E-01	2,43E+08	1,72E+08	**84,00 %**
RD_PVO	0,00E+00	0,00E+00	0,00E+00	1,55E+08	1,06E+08	**100,00 %**
RD_VOT	0,00E+00	0,00E+00	0,00E+00	1,88E+08	1,32E+08	**100,00 %**
RP_AND	0,00E+00	0,00E+00	0,00E+00	6,30E+05	3,28E+05	**100,00 %**
RP_OR	0,00E+00	2,40E-01	4,31E-01	2,03E+08	1,91E+08	**76,00 %**
RP_PVO	0,00E+00	0,00E+00	0,00E+00	1,53E+08	1,45E+08	**100,00 %**
RP_VOT	0,00E+00	0,00E+00	0,00E+00	1,58E+08	1,48E+08	**100,00 %**
RPD_AND	0,00E+00	0,00E+00	0,00E+00	6,14E+05	3,07E+05	**100,00 %**

RPD_OR	0,00E+00	4,40E-01	5,01E-01	2,86E+08	1,85E+08	**56,00 %**
RPD_PVO	0,00E+00	0,00E+00	0,00E+00	1,60E+08	1,60E+08	**100,00 %**
RPD_VOT	0,00E+00	0,00E+00	0,00E+00	2,48E+08	1,43E+08	**100,00 %**
CF_AND	0,00E+00	0,00E+00	0,00E+00	9,71E+05	4,69E+05	**100,00 %**
CF_OR	0,00E+00	2,20E-01	4,18E-01	2,66E+08	1,71E+08	**78,00 %**
CF_PVO	0,00E+00	0,00E+00	0,00E+00	1,56E+08	1,27E+08	**100,00 %**
CF_VOT	0,00E+00	0,00E+00	0,00E+00	1,95E+08	1,18E+08	**100,00 %**
GR_AND	0,00E+00	0,00E+00	0,00E+00	2,39E+06	8,46E+05	**100,00 %**
GR_OR	0,00E+00	0,00E+00	0,00E+00	2,39E+06	8,46E+05	**100,00 %**
GR_PVO	0,00E+00	0,00E+00	0,00E+00	2,36E+06	1,19E+06	**100,00 %**
GR_VOT	0,00E+00	0,00E+00	0,00E+00	2,39E+06	8,46E+05	**100,00 %**
TABU_AND	0,00E+00	0,00E+00	0,00E+00	7,07E+05	3,19E+05	**100,00 %**
TABU_OR	0,00E+00	4,60E-01	5,03E-01	3,26E+08	1,86E+08	**54,00 %**
TABU_PVO	0,00E+00	0,00E+00	0,00E+00	2,23E+08	1,28E+08	**100,00 %**
TABU_VOT	0,00E+00	0,00E+00	0,00E+00	2,33E+08	1,24E+08	**100,00 %**

Table A.4. Results of performance evaluations of hyperheuristic patterns on Quartic Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	8,86E-01	1,33E+00	3,88E-01	3,14E+08	9,52E+07	**24,00 %**
SR_OR	7,19E-01	9,86E-01	1,06E-01	2,39E+08	1,33E+08	**52,00 %**
SR_PVO	8,09E-01	9,57E-01	6,11E-02	2,13E+08	1,22E+08	**80,00 %**
SR_VOT	7,59E-01	9,48E-01	6,13E-02	2,20E+08	1,03E+08	**92,00 %**
RD_AND	8,36E-01	1,28E+00	2,77E-01	7,52E+07	3,82E+07	**14,00 %**
RD_OR	7,40E-01	1,05E+00	1,08E-01	3,12E+08	8,84E+07	**26,00 %**
RD_PVO	8,31E-01	9,98E-01	6,77E-02	2,45E+08	1,04E+08	**54,00 %**
RD_VOT	7,08E-01	9,96E-01	1,05E-01	2,52E+08	8,90E+07	**52,00 %**
RP_AND	6,29E-01	1,38E+00	4,17E-01	2,99E+08	8,87E+07	**20,00 %**
RP_OR	8,08E-01	9,96E-01	7,91E-02	2,64E+08	1,16E+08	**50,00 %**
RP_PVO	7,34E-01	9,41E-01	7,65E-02	1,94E+08	1,20E+08	**84,00 %**

RP_VOT	6,70E-01	9,32E-01	8,89E-02	2,20E+08	1,13E+08	**84,00 %**
RPD_AND	8,52E-01	1,38E+00	3,95E-01	3,13E+08	1,11E+08	**18,00 %**
RPD_OR	7,60E-01	9,54E-01	8,97E-02	1,85E+08	1,30E+08	**68,00 %**
RPD_PVO	7,62E-01	9,51E-01	6,89E-02	2,12E+08	1,25E+08	**84,00 %**
RPD_VOT	8,20E-01	9,49E-01	5,61E-02	2,02E+08	1,19E+08	**84,00 %**
CF_AND	7,02E-01	1,04E+00	1,74E-01	2,43E+08	1,21E+08	**54,00 %**
CF_OR	7,10E-01	9,52E-01	9,42E-02	2,40E+08	1,19E+08	**64,00 %**
CF_PVO	6,87E-01	9,40E-01	6,95E-02	1,58E+08	1,11E+08	**92,00 %**
CF_VOT	6,80E-01	9,35E-01	6,17E-02	1,57E+08	1,05E+08	**96,00 %**
GR_AND	8,96E-01	1,30E+00	3,62E-01	3,04E+08	1,12E+08	**16,00 %**
GR_OR	7,81E-01	1,29E+00	4,15E-01	2,74E+08	1,27E+08	**36,00 %**
GR_PVO	8,52E-01	1,33E+00	4,01E-01	3,05E+08	1,14E+08	**26,00 %**
GR_VOT	7,09E-01	1,31E+00	4,93E-01	3,00E+08	1,18E+08	**26,00 %**
TABU_AND	8,84E-01	1,46E+00	4,76E-01	3,28E+08	9,16E+07	**14,00 %**
TABU_OR	5,62E-01	9,44E-01	9,73E-02	1,89E+08	1,35E+08	**78,00 %**
TABU_PVO	8,13E-01	9,44E-01	5,89E-02	1,82E+08	1,26E+08	**86,00 %**
TABU_VOT	6,84E-01	9,35E-01	8,25E-02	2,10E+08	1,14E+08	**86,00 %**

Table A.5. Results of performance evaluations of hyperheuristic patterns on Foxhole Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	9,98E-01	9,98E-01	1,12E-15	1,93E+07	4,66E+07	**90,00 %**
SR_OR	9,98E-01	9,98E-01	1,12E-15	2,70E+04	2,59E+04	**100,00 %**
SR_PVO	9,98E-01	9,98E-01	1,12E-15	3,03E+04	2,86E+04	**100,00 %**
SR_VOT	9,98E-01	9,98E-01	1,12E-15	2,92E+04	3,25E+04	**100,00 %**
RD_AND	9,98E-01	1,41E+02	2,19E+02	9,08E+07	1,32E+07	**2,00 %**
RD_OR	9,98E-01	9,98E-01	1,12E-15	2,01E+04	2,69E+04	**100,00 %**
RD_PVO	9,98E-01	9,98E-01	1,12E-15	2,16E+04	1,66E+04	**100,00 %**
RD_VOT	9,98E-01	3,08E+01	1,19E+02	5,53E+06	2,20E+07	**94,00 %**
RP_AND	9,98E-01	9,98E-01	1,12E-15	1,14E+07	3,04E+07	**98,00 %**

RP_OR	9,98E-01	9,98E-01	1,12E-15	2,91E+04	3,32E+04	**100,00 %**
RP_PVO	9,98E-01	9,98E-01	1,12E-15	2,46E+04	2,73E+04	**100,00 %**
RP_VOT	9,98E-01	9,98E-01	1,12E-15	2,99E+04	3,37E+04	**100,00 %**
RPD_AND	9,98E-01	9,98E-01	1,12E-15	9,70E+06	3,03E+07	**96,00 %**
RPD_OR	9,98E-01	9,98E-01	1,12E-15	2,86E+04	2,39E+04	**100,00 %**
RPD_PVO	9,98E-01	9,98E-01	1,12E-15	3,81E+04	4,13E+04	**100,00 %**
RPD_VOT	9,98E-01	9,98E-01	1,12E-15	3,38E+04	4,09E+04	**100,00 %**
CF_AND	9,98E-01	9,98E-01	1,12E-15	8,71E+06	1,96E+07	**100,00 %**
CF_OR	9,98E-01	9,98E-01	1,12E-15	6,04E+04	5,78E+04	**100,00 %**
CF_PVO	9,98E-01	9,98E-01	1,12E-15	5,43E+04	6,24E+04	**100,00 %**
CF_VOT	9,98E-01	9,98E-01	1,12E-15	5,33E+04	6,13E+04	**100,00 %**
GR_AND	9,98E-01	9,98E-01	1,12E-15	1,63E+07	4,43E+07	**90,00 %**
GR_OR	9,98E-01	9,98E-01	1,12E-15	2,00E+07	4,92E+07	**88,00 %**
GR_PVO	9,98E-01	9,98E-01	1,12E-15	2,00E+07	4,93E+07	**88,00 %**
GR_VOT	9,98E-01	9,98E-01	1,12E-15	1,81E+07	4,56E+07	**90,00 %**
TABU_AND	9,98E-01	9,98E-01	1,12E-15	1,36E+07	3,30E+07	**98,00 %**
TABU_OR	9,98E-01	9,98E-01	1,12E-15	6,00E+04	5,31E+04	**100,00 %**
TABU_PVO	9,98E-01	9,98E-01	1,12E-15	4,98E+04	5,63E+04	**100,00 %**
TABU_VOT	9,98E-01	9,98E-01	1,12E-15	4,93E+04	5,88E+04	**100,00 %**

Table A.6. Results of performance evaluations of hyperheuristic patterns on Rastrigin Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	0,00E+00	0,00E+00	0,00E+00	4,72E+06	1,95E+06	**100,00 %**
SR_OR	9,95E-01	2,41E+00	6,39E-01	3,68E+08	6,80E+04	**0,00 %**
SR_PVO	0,00E+00	3,98E-02	1,97E-01	3,62E+08	1,08E+07	**96,00 %**
SR_VOT	0,00E+00	0,00E+00	0,00E+00	3,47E+08	1,71E+07	**100,00 %**
RD_AND	0,00E+00	0,00E+00	0,00E+00	2,96E+04	7,49E+03	**100,00 %**
RD_OR	0,00E+00	1,88E+00	5,85E-01	3,52E+08	4,07E+07	**2,00 %**
RD_PVO	0,00E+00	0,00E+00	0,00E+00	3,42E+08	9,40E+06	**100,00 %**

RD_VOT	0,00E+00	0,00E+00	0,00E+00	3,24E+08	1,14E+07	**100,00 %**
RP_AND	0,00E+00	0,00E+00	0,00E+00	4,53E+06	1,73E+06	**100,00 %**
RP_OR	9,95E-01	2,31E+00	8,39E-01	3,69E+08	8,99E+05	**0,00 %**
RP_PVO	0,00E+00	3,98E-02	1,97E-01	3,64E+08	9,46E+06	**96,00 %**
RP_VOT	0,00E+00	0,00E+00	0,00E+00	3,50E+08	6,58E+06	**100,00 %**
RPD_AND	0,00E+00	0,00E+00	0,00E+00	4,48E+06	2,30E+06	**100,00 %**
RPD_OR	9,95E-01	3,34E+00	9,80E-01	3,67E+08	6,91E+06	**0,00 %**
RPD_PVO	0,00E+00	1,59E-01	3,68E-01	3,68E+08	5,77E+06	**84,00 %**
RPD_VOT	0,00E+00	0,00E+00	0,00E+00	3,55E+08	8,08E+06	**100,00 %**
CF_AND	0,00E+00	0,00E+00	0,00E+00	9,92E+06	3,23E+06	**100,00 %**
CF_OR	0,00E+00	2,91E+00	9,39E-01	3,64E+08	1,91E+07	**4,00 %**
CF_PVO	0,00E+00	6,37E-01	6,89E-01	3,68E+08	1,90E+07	**48,00 %**
CF_VOT	0,00E+00	3,98E-02	1,97E-01	3,53E+08	1,37E+07	**96,00 %**
GR_AND	0,00E+00	0,00E+00	0,00E+00	4,43E+06	2,05E+06	**100,00 %**
GR_OR	0,00E+00	0,00E+00	0,00E+00	4,43E+06	2,05E+06	**100,00 %**
GR_PVO	0,00E+00	0,00E+00	0,00E+00	5,42E+06	2,80E+06	**100,00 %**
GR_VOT	0,00E+00	0,00E+00	0,00E+00	4,43E+06	2,05E+06	**100,00 %**
TABU_AND	0,00E+00	0,00E+00	0,00E+00	5,18E+06	2,63E+06	**100,00 %**
TABU_OR	9,95E-01	3,13E+00	8,06E-01	3,66E+08	2,00E+06	**0,00 %**
TABU_PVO	0,00E+00	2,98E-01	4,61E-01	3,43E+08	3,51E+06	**70,00 %**
TABU_VOT	0,00E+00	1,99E-02	1,41E-01	3,54E+08	1,00E+07	**98,00 %**

Table A. 7. Results of performance evaluations of hyperheuristic patterns on Schwefel Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	1,27E-04	1,27E-04	2,74E-20	1,01E+06	4,44E+05	**100,00 %**
SR_OR	1,27E-04	1,97E+01	4,37E+01	1,69E+08	1,20E+08	**74,00 %**
SR_PVO	1,27E-04	1,27E-04	2,74E-20	9,92E+07	6,80E+07	**100,00 %**
SR_VOT	1,27E-04	1,27E-04	2,74E-20	1,17E+08	6,69E+07	**100,00 %**
RD_AND	1,27E-04	1,27E-04	2,74E-20	1,67E+04	5,49E+03	**100,00 %**

RD_OR	1,27E-04	2,38E+00	1,67E+01	1,11E+08	9,11E+07	**92,00 %**
RD_PVO	1,27E-04	1,27E-04	2,74E-20	6,33E+07	5,40E+07	**100,00 %**
RD_VOT	1,27E-04	1,27E-04	2,74E-20	8,43E+07	6,15E+07	**100,00 %**
RP_AND	1,27E-04	1,27E-04	2,74E-20	9,63E+05	4,13E+05	**100,00 %**
RP_OR	1,27E-04	3,55E+01	5,48E+01	1,51E+08	1,24E+08	**70,00 %**
RP_PVO	1,27E-04	1,27E-04	2,74E-20	1,16E+08	8,79E+07	**100,00 %**
RP_VOT	1,27E-04	1,27E-04	2,74E-20	1,03E+08	8,30E+07	**100,00 %**
RPD_AND	1,27E-04	1,27E-04	2,74E-20	9,72E+05	4,39E+05	**100,00 %**
RPD_OR	1,27E-04	8,24E+01	9,70E+01	1,98E+08	1,15E+08	**52,00 %**
RPD_PVO	1,27E-04	1,27E-04	2,74E-20	1,77E+08	9,55E+07	**100,00 %**
RPD_VOT	1,27E-04	1,27E-04	2,74E-20	1,56E+08	9,05E+07	**100,00 %**
CF_AND	1,27E-04	1,27E-04	2,74E-20	3,47E+06	2,67E+06	**100,00 %**
CF_OR	1,27E-04	5,92E+01	5,98E+01	2,36E+08	1,07E+08	**50,00 %**
CF_PVO	1,27E-04	1,27E-04	2,74E-20	1,57E+08	9,16E+07	**100,00 %**
CF_VOT	1,27E-04	1,27E-04	2,74E-20	1,49E+08	7,31E+07	**100,00 %**
GR_AND	1,27E-04	1,27E-04	2,74E-20	8,62E+05	4,90E+05	**100,00 %**
GR_OR	1,27E-04	1,27E-04	2,74E-20	8,62E+05	4,90E+05	**100,00 %**
GR_PVO	1,27E-04	1,27E-04	2,74E-20	1,04E+06	4,17E+05	**100,00 %**
GR_VOT	1,27E-04	1,27E-04	2,74E-20	8,62E+05	4,90E+05	**100,00 %**
TABU_AND	1,27E-04	1,27E-04	2,74E-20	1,11E+06	6,57E+05	**100,00 %**
TABU_OR	1,27E-04	1,11E+02	6,25E+01	2,88E+08	6,99E+07	**18,00 %**
TABU_PVO	1,27E-04	1,27E-04	2,74E-20	2,08E+08	6,93E+07	**100,00 %**
TABU_VOT	1,27E-04	1,27E-04	2,74E-20	2,04E+08	6,09E+07	**100,00 %**

Table A. 8. Results of performance evaluations of hyperheuristic patterns on Griewangk Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	0,00E+00	7,45E-02	4,84E-02	1,84E+08	4,65E+07	**6,00 %**
SR_OR	0,00E+00	0,00E+00	0,00E+00	6,37E+05	6,90E+05	**100,00 %**
SR_PVO	0,00E+00	0,00E+00	0,00E+00	7,22E+05	7,44E+05	**100,00 %**

SR_VOT	0,00E+00	0,00E+00	0,00E+00	6,07E+05	6,68E+05	**100,00 %**
RD_AND	0,00E+00	2,16E-01	5,49E-01	1,17E+08	2,48E+07	**4,00 %**
RD_OR	0,00E+00	0,00E+00	0,00E+00	3,36E+05	2,94E+05	**100,00 %**
RD_PVO	0,00E+00	0,00E+00	0,00E+00	6,08E+05	5,94E+05	**100,00 %**
RD_VOT	0,00E+00	0,00E+00	0,00E+00	3,90E+05	3,43E+05	**100,00 %**
RP_AND	0,00E+00	7,29E-02	4,80E-02	1,76E+08	5,24E+07	**8,00 %**
RP_OR	0,00E+00	0,00E+00	0,00E+00	8,38E+05	1,15E+06	**100,00 %**
RP_PVO	0,00E+00	0,00E+00	0,00E+00	4,67E+05	7,46E+05	**100,00 %**
RP_VOT	0,00E+00	0,00E+00	0,00E+00	5,08E+05	6,16E+05	**100,00 %**
RPD_AND	0,00E+00	7,99E-02	4,08E-02	1,90E+08	2,75E+07	**2,00 %**
RPD_OR	0,00E+00	0,00E+00	0,00E+00	3,79E+06	4,77E+06	**100,00 %**
RPD_PVO	0,00E+00	0,00E+00	0,00E+00	4,51E+06	5,16E+06	**100,00 %**
RPD_VOT	0,00E+00	0,00E+00	0,00E+00	3,70E+06	3,95E+06	**100,00 %**
CF_AND	0,00E+00	4,72E-02	3,21E-02	1,86E+08	3,85E+07	**4,00 %**
CF_OR	0,00E+00	0,00E+00	0,00E+00	3,52E+06	4,37E+06	**100,00 %**
CF_PVO	0,00E+00	0,00E+00	0,00E+00	2,45E+06	2,65E+06	**100,00 %**
CF_VOT	0,00E+00	0,00E+00	0,00E+00	5,36E+06	6,59E+06	**100,00 %**
GR_AND	0,00E+00	7,45E-02	4,78E-02	1,63E+08	5,89E+07	**12,00 %**
GR_OR	0,00E+00	7,12E-02	5,21E-02	1,69E+08	6,55E+07	**14,00 %**
GR_PVO	0,00E+00	7,16E-02	5,01E-02	1,71E+08	6,23E+07	**12,00 %**
GR_VOT	0,00E+00	7,12E-02	5,03E-02	1,73E+08	6,19E+07	**12,00 %**
TABU_AND	0,00E+00	7,85E-02	4,74E-02	1,83E+08	4,34E+07	**6,00 %**
TABU_OR	0,00E+00	0,00E+00	0,00E+00	3,48E+06	2,83E+06	**100,00 %**
TABU_PVO	0,00E+00	0,00E+00	0,00E+00	4,80E+06	3,26E+06	**100,00 %**
TABU_VOT	0,00E+00	0,00E+00	0,00E+00	3,46E+06	3,14E+06	**100,00 %**

Table A. 9. Results of performance evaluations of hyperheuristic patterns on Ackley Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	2,84E-14	2,84E-14	1,91E-29	9,94E+03	2,38E+04	**100.00 %**

SR_OR	2,84E-14	2,84E-14	1,91E-29	5,18E+03	7,06E+03	**100.00 %**
SR_PVO	2,84E-14	2,84E-14	1,91E-29	4,23E+03	4,75E+03	**100.00 %**
SR_VOT	2,84E-14	2,84E-14	1,91E-29	5,36E+03	5,75E+03	**100.00 %**
RD_AND	2,84E-14	2,84E-14	1,91E-29	3,04E+03	2,53E+03	**100.00 %**
RD_OR	2,84E-14	2,84E-14	1,91E-29	3,28E+03	2,04E+03	**100.00 %**
RD_PVO	2,84E-14	2,84E-14	1,91E-29	3,24E+03	2,51E+03	**100.00 %**
RD_VOT	2,84E-14	2,84E-14	1,91E-29	3,17E+03	3,11E+03	**100.00 %**
RP_AND	2,84E-14	2,84E-14	1,91E-29	1,44E+04	2,68E+04	**100.00 %**
RP_OR	2,84E-14	2,84E-14	1,91E-29	4,56E+03	8,97E+03	**100.00 %**
RP_PVO	2,84E-14	2,84E-14	1,91E-29	5,33E+03	8,65E+03	**100.00 %**
RP_VOT	2,84E-14	2,84E-14	1,91E-29	4,15E+03	6,56E+03	**100.00 %**
RPD_AND	2,84E-14	2,84E-14	1,91E-29	1,21E+04	1,74E+04	**100.00 %**
RPD_OR	2,84E-14	2,84E-14	1,91E-29	9,67E+03	1,70E+04	**100.00 %**
RPD_PVO	2,84E-14	2,84E-14	1,91E-29	1,61E+04	3,04E+04	**100.00 %**
RPD_VOT	2,84E-14	2,84E-14	1,91E-29	9,67E+03	1,70E+04	**100.00 %**
CF_AND	2,84E-14	2,84E-14	1,91E-29	1,67E+04	3,84E+04	**100.00 %**
CF_OR	2,84E-14	2,84E-14	1,91E-29	7,67E+03	1,61E+04	**100.00 %**
CF_PVO	2,84E-14	2,84E-14	1,91E-29	6,37E+03	8,43E+03	**100.00 %**
CF_VOT	2,84E-14	2,84E-14	1,91E-29	1,34E+04	2,31E+04	**100.00 %**
GR_AND	2,84E-14	2,84E-14	1,91E-29	4,57E+03	6,51E+03	**100.00 %**
GR_OR	2,84E-14	2,84E-14	1,91E-29	4,57E+03	6,51E+03	**100.00 %**
GR_PVO	2,84E-14	2,84E-14	1,91E-29	3,88E+03	4,85E+03	**100.00 %**
GR_VOT	2,84E-14	2,84E-14	1,91E-29	4,57E+03	6,51E+03	**100.00 %**
TABU_AND	2,84E-14	2,84E-14	1,91E-29	1,98E+04	3,19E+04	**100.00 %**
TABU_OR	2,84E-14	2,84E-14	1,91E-29	1,51E+04	2,67E+04	**100.00 %**
TABU_PVO	2,84E-14	2,84E-14	1,91E-29	1,63E+04	2,45E+04	**100.00 %**
TABU_VOT	2,84E-14	2,84E-14	1,91E-29	2,43E+04	5,56E+04	**100.00 %**

Table A.10. Results of performance evaluations of hyperheuristic patterns on Easom Function

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.		S. R.
SR_AND	-1,00E+00	-1,00E+00	0,00E+00	3,62E+07	2,34E+07	**100,00 %**
SR_OR	-1,00E+00	-8,80E-01	3,28E-01	1,57E+08	1,36E+08	**88,00 %**
SR_PVO	-1,00E+00	-9,60E-01	1,98E-01	1,33E+08	1,08E+08	**96,00 %**
SR_VOT	-1,00E+00	-1,00E+00	0,00E+00	3,50E+07	2,37E+07	**100,00 %**
RD_AND	-1,00E+00	-1,00E+00	0,00E+00	1,76E+07	8,59E+06	**100,00 %**
RD_OR	-1,00E+00	-9,00E-01	3,03E-01	1,72E+08	1,30E+08	**88,00 %**
RD_PVO	-1,00E+00	-9,40E-01	2,40E-01	1,32E+08	1,15E+08	**94,00 %**
RD_VOT	-1,00E+00	-1,00E+00	0,00E+00	1,57E+07	9,38E+06	**100,00 %**
RP_AND	-1,00E+00	-1,00E+00	0,00E+00	2,66E+07	1,56E+07	**100,00 %**
RP_OR	-1,00E+00	-9,59E-01	1,98E-01	1,21E+08	1,14E+08	**94,00 %**
RP_PVO	-1,00E+00	-9,60E-01	1,98E-01	1,31E+08	1,08E+08	**96,00 %**
RP_VOT	-1,00E+00	-1,00E+00	0,00E+00	2,66E+07	1,56E+07	**100,00 %**
RPD_AND	-1,00E+00	-1,00E+00	0,00E+00	2,96E+07	2,04E+07	**100,00 %**
RPD_OR	-1,00E+00	-9,60E-01	1,98E-01	1,39E+08	1,15E+08	**96,00 %**
RPD_PVO	-1,00E+00	-9,20E-01	2,74E-01	1,24E+08	1,13E+08	**92,00 %**
RPD_VOT	-1,00E+00	-1,00E+00	0,00E+00	3,58E+07	2,78E+07	**100,00 %**
CF_AND	-1,00E+00	-1,00E+00	0,00E+00	2,70E+07	1,96E+07	**100,00 %**
CF_OR	-1,00E+00	-9,60E-01	1,98E-01	1,25E+08	1,14E+08	**96,00 %**
CF_PVO	-1,00E+00	-9,20E-01	2,74E-01	1,46E+08	1,12E+08	**92,00 %**
CF_VOT	-1,00E+00	-9,80E-01	1,41E-01	3,96E+07	6,26E+07	**98,00 %**
GR_AND	-1,00E+00	-1,00E+00	0,00E+00	6,92E+07	4,54E+07	**100,00 %**
GR_OR	-1,00E+00	-1,00E+00	0,00E+00	6,92E+07	4,54E+07	**100,00 %**
GR_PVO	-1,00E+00	-1,00E+00	0,00E+00	5,71E+07	4,34E+07	**100,00 %**
GR_VOT	-1,00E+00	-1,00E+00	0,00E+00	6,92E+07	4,54E+07	**100,00 %**
TABU_AND	-1,00E+00	-1,00E+00	0,00E+00	3,25E+07	2,65E+07	**100,00 %**
TABU_OR	-1,00E+00	-9,00E-01	3,03E-01	1,55E+08	1,23E+08	**90,00 %**
TABU_PVO	-1,00E+00	-8,80E-01	3,28E-01	1,69E+08	1,37E+08	**88,00 %**
TABU_VOT	-1,00E+00	-1,00E+00	0,00E+00	3,25E+07	2,65E+07	**100,00 %**

Table A.11. Results of performance evaluations of hyperheuristic patterns on Rotated Function

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.		S. R.
SR_AND	2,98E-14	1,51E-13	9,84E-14	1,70E+08	1,17E+06	**0,00 %**
SR_OR	3,16E-07	7,42E-05	7,21E-05	1,69E+08	1,27E+06	**0,00 %**
SR_PVO	2,53E-13	1,04E-12	5,15E-13	1,65E+08	5,04E+06	**0,00 %**
SR_VOT	3,58E-13	1,08E-12	4,94E-13	1,70E+08	1,28E+06	**0,00 %**
RD_AND	2,75E-01	3,87E+02	6,97E+02	9,95E+07	1,18E+06	**0,00 %**
RD_OR	2,17E-05	1,21E-03	1,17E-03	1,68E+08	1,71E+06	**0,00 %**
RD_PVO	1,43E-12	2,25E-06	4,30E-06	1,69E+08	9,66E+04	**0,00 %**
RD_VOT	1,34E-13	1,39E-08	4,43E-08	1,69E+08	6,31E+04	**0,00 %**
RP_AND	2,98E-14	1,73E-13	1,13E-13	1,59E+08	1,26E+04	**0,00 %**
RP_OR	5,51E-06	3,17E-04	3,07E-04	1,70E+08	7,13E+04	**0,00 %**
RP_PVO	1,19E-13	1,35E-12	5,15E-13	1,70E+08	1,26E+06	**0,00 %**
RP_VOT	2,24E-13	1,15E-12	5,70E-13	1,71E+08	3,82E+04	**0,00 %**
RPD_AND	1,49E-14	1,80E-13	1,23E-13	1,70E+08	2,57E+04	**0,00 %**
RPD_OR	2,98E-14	7,96E-14	2,82E-14	1,62E+08	5,99E+06	**0,00 %**
RPD_PVO	1,49E-14	4,65E-14	1,72E-14	1,71E+08	2,17E+06	**0,00 %**
RPD_VOT	1,49E-14	5,42E-14	2,51E-14	1,71E+08	2,57E+06	**0,00 %**
CF_AND	7,78E-26	2,32E-13	3,35E-13	1,67E+08	1,48E+07	**2,00 %**
CF_OR	7,78E-26	7,78E-26	2,32E-41	8,51E+06	7,48E+06	**100,00 %**
CF_PVO	7,78E-26	7,78E-26	2,32E-41	1,65E+07	1,39E+07	**100,00 %**
CF_VOT	7,78E-26	7,78E-26	2,32E-41	6,12E+06	3,99E+06	**100,00 %**
GR_AND	1,49E-14	1,21E-13	8,31E-14	1,68E+08	4,73E+06	**0,00 %**
GR_OR	7,78E-26	1,31E-13	1,14E-13	1,70E+08	5,78E+06	**4,00 %**
GR_PVO	1,49E-14	1,18E-13	9,19E-14	1,70E+08	1,25E+06	**0,00 %**
GR_VOT	1,49E-14	1,25E-13	8,20E-14	1,70E+08	2,09E+06	**0,00 %**
TABU_AND	1,49E-14	1,88E-13	1,47E-13	1,70E+08	3,76E+05	**0,00 %**
TABU_OR	1,49E-14	3,55E-14	1,27E-14	1,71E+08	9,33E+04	**0,00 %**
TABU_PVO	7,78E-26	2,89E-14	1,29E-14	1,63E+08	2,85E+07	**6,00 %**
TABU_VOT	1,49E-14	3,40E-14	1,31E-14	1,71E+08	7,98E+04	**0,00 %**

Table A.12. Results of performance evaluations of hyperheuristic patterns on Royal Road Function

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.		S. R.
SR_AND	0,00E+00	0,00E+00	0,00E+00	8,82E+04	4,64E+04	**100,00 %**
SR_OR	1,00E+00	2,06E+00	3,73E-01	1,52E+09	2,36E+05	**0,00 %**
SR_PVO	0,00E+00	0,00E+00	0,00E+00	1,01E+09	1,06E+08	**100,00 %**
SR_VOT	0,00E+00	0,00E+00	0,00E+00	8,18E+08	1,08E+08	**100,00 %**
RD_AND	0,00E+00	0,00E+00	0,00E+00	3,67E+04	1,29E+04	**100,00 %**
RD_OR	2,00E+00	2,18E+00	3,88E-01	1,51E+09	6,72E+05	**0,00 %**
RD_PVO	0,00E+00	0,00E+00	0,00E+00	1,06E+09	5,91E+07	**100,00 %**
RD_VOT	0,00E+00	0,00E+00	0,00E+00	7,63E+08	9,19E+07	**100,00 %**
RP_AND	0,00E+00	0,00E+00	0,00E+00	9,03E+04	3,61E+04	**100,00 %**
RP_OR	1,00E+00	1,98E+00	3,77E-01	1,53E+09	2,66E+05	**0,00 %**
RP_PVO	0,00E+00	0,00E+00	0,00E+00	1,02E+09	1,53E+08	**100,00 %**
RP_VOT	0,00E+00	0,00E+00	0,00E+00	7,93E+08	1,18E+08	**100,00 %**
RPD_AND	0,00E+00	0,00E+00	0,00E+00	8,89E+04	4,54E+04	**100,00 %**
RPD_OR	1,00E+00	2,04E+00	3,48E-01	1,48E+09	5,09E+07	**0,00 %**
RPD_PVO	0,00E+00	0,00E+00	0,00E+00	1,03E+09	1,91E+08	**100,00 %**
RPD_VOT	0,00E+00	0,00E+00	0,00E+00	8,31E+08	8,49E+07	**100,00 %**
CF_AND	0,00E+00	0,00E+00	0,00E+00	1,20E+05	5,41E+04	**100,00 %**
CF_OR	1,00E+00	2,06E+00	4,70E-01	1,35E+09	1,20E+07	**0,00 %**
CF_PVO	0,00E+00	0,00E+00	0,00E+00	9,41E+08	8,54E+07	**100,00 %**
CF_VOT	0,00E+00	0,00E+00	0,00E+00	7,52E+08	7,17E+07	**100,00 %**
GR_AND	0,00E+00	0,00E+00	0,00E+00	3,10E+05	1,25E+05	**100,00 %**
GR_OR	0,00E+00	0,00E+00	0,00E+00	3,10E+05	1,25E+05	**100,00 %**
GR_PVO	0,00E+00	0,00E+00	0,00E+00	3,42E+05	1,49E+05	**100,00 %**
GR_VOT	0,00E+00	0,00E+00	0,00E+00	3,10E+05	1,25E+05	**100,00 %**
TABU_AND	0,00E+00	0,00E+00	0,00E+00	8,85E+04	4,05E+04	**100,00 %**
TABU_OR	1,00E+00	1,92E+00	3,96E-01	1,50E+09	1,95E+07	**0,00 %**

TABU_PVO	0,00E+00	0,00E+00	0,00E+00	9,52E+08	8,91E+07	**100,00 %**
TABU_VOT	0,00E+00	0,00E+00	0,00E+00	8,05E+08	9,03E+07	**100,00 %**

Table A.13. Results of performance evaluations of hyperheuristic patterns on Goldberg Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	0,00E+00	6,52E+00	3,20E+00	7,16E+08	2,77E+07	**2,00 %**
SR_OR	1,60E+01	2,00E+01	1,54E+00	6,92E+08	4,55E+05	**0,00 %**
SR_PVO	1,60E+01	2,05E+01	1,69E+00	7,00E+08	2,28E+07	**0,00 %**
SR_VOT	1,80E+01	2,10E+01	1,29E+00	7,26E+08	2,01E+06	**0,00 %**
RD_AND	2,00E+00	8,40E+00	2,06E+00	9,81E+07	2,26E+05	**0,00 %**
RD_OR	1,60E+01	1,99E+01	1,10E+00	6,18E+08	2,93E+05	**0,00 %**
RD_PVO	1,00E+01	1,64E+01	2,09E+00	4,68E+08	1,67E+07	**0,00 %**
RD_VOT	8,00E+00	1,51E+01	2,29E+00	4,42E+08	2,11E+07	**0,00 %**
RP_AND	0,00E+00	6,08E+00	3,05E+00	7,16E+08	2,62E+07	**4,00 %**
RP_OR	1,60E+01	1,95E+01	1,69E+00	6,94E+08	4,35E+06	**0,00 %**
RP_PVO	1,60E+01	2,06E+01	1,51E+00	7,25E+08	2,83E+06	**0,00 %**
RP_VOT	1,60E+01	2,08E+01	1,56E+00	7,35E+08	2,91E+06	**0,00 %**
RPD_AND	0,00E+00	5,76E+00	3,09E+00	7,35E+08	3,23E+07	**4,00 %**
RPD_OR	1,40E+01	2,00E+01	1,67E+00	6,99E+08	2,75E+07	**0,00 %**
RPD_PVO	1,40E+01	2,06E+01	1,87E+00	7,37E+08	8,69E+06	**0,00 %**
RPD_VOT	1,80E+01	2,11E+01	1,52E+00	7,37E+08	7,95E+06	**0,00 %**
CF_AND	6,00E+00	2,01E+01	9,61E+00	6,85E+08	4,03E+07	**0,00 %**
CF_OR	2,20E+01	2,70E+01	1,47E+00	7,51E+08	1,89E+07	**0,00 %**
CF_PVO	2,40E+01	2,85E+01	1,55E+00	8,02E+08	6,54E+06	**0,00 %**
CF_VOT	1,60E+01	2,62E+01	3,89E+00	7,88E+08	2,33E+07	**0,00 %**
GR_AND	0,00E+00	6,72E+00	3,74E+00	7,57E+08	3,37E+07	**2,00 %**
GR_OR	0,00E+00	6,80E+00	3,38E+00	7,64E+08	2,68E+07	**4,00 %**
GR_PVO	0,00E+00	7,44E+00	3,57E+00	7,60E+08	2,00E+07	**2,00 %**
GR_VOT	2,00E+00	5,84E+00	3,40E+00	7,59E+08	3,21E+07	**0,00 %**

TABU_AND	0,00E+00	6,16E+00	3,07E+00	7,18E+08	4,30E+07	**8,00 %**
TABU_OR	1,40E+01	2,03E+01	1,71E+00	7,06E+08	1,29E+05	**0,00 %**
TABU_PVO	1,80E+01	2,13E+01	1,25E+00	7,15E+08	2,27E+07	**0,00 %**
TABU_VOT	1,60E+01	2,16E+01	1,62E+00	7,34E+08	1,74E+06	**0,00 %**

Table A.14. Results of performance evaluations of hyperheuristic patterns on Whitley Function

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**		**S. R.**
SR_AND	0,00E+00	0,00E+00	0,00E+00	6,44E+04	3,89E+04	**100,00 %**
SR_OR	0,00E+00	1,08E+00	1,01E+00	8,08E+08	3,23E+08	**46,00 %**
SR_PVO	0,00E+00	0,00E+00	0,00E+00	6,37E+08	2,14E+08	**100,00 %**
SR_VOT	0,00E+00	0,00E+00	0,00E+00	5,74E+08	2,48E+08	**100,00 %**
RD_AND	0,00E+00	0,00E+00	0,00E+00	3,10E+03	1,59E+03	**100,00 %**
RD_OR	0,00E+00	0,00E+00	0,00E+00	6,46E+07	5,87E+07	**100,00 %**
RD_PVO	0,00E+00	0,00E+00	0,00E+00	5,85E+07	6,01E+07	**100,00 %**
RD_VOT	0,00E+00	0,00E+00	0,00E+00	6,36E+07	5,63E+07	**100,00 %**
RP_AND	0,00E+00	0,00E+00	0,00E+00	5,29E+04	2,49E+04	**100,00 %**
RP_OR	0,00E+00	1,80E+00	6,06E-01	1,02E+09	1,45E+08	**10,00 %**
RP_PVO	0,00E+00	0,00E+00	0,00E+00	7,75E+08	1,39E+08	**100,00 %**
RP_VOT	0,00E+00	0,00E+00	0,00E+00	6,82E+08	1,73E+08	**100,00 %**
RPD_AND	0,00E+00	0,00E+00	0,00E+00	5,59E+04	2,95E+04	**100,00 %**
RPD_OR	0,00E+00	1,84E+00	5,48E-01	1,03E+09	2,04E+08	**8,00 %**
RPD_PVO	0,00E+00	0,00E+00	0,00E+00	8,07E+08	1,24E+08	**100,00 %**
RPD_VOT	0,00E+00	0,00E+00	0,00E+00	7,25E+08	1,53E+08	**100,00 %**
CF_AND	0,00E+00	0,00E+00	0,00E+00	2,88E+06	1,84E+06	**100,00 %**
CF_OR	2,00E+00	3,44E+00	9,07E-01	1,07E+09	2,32E+07	**0,00 %**
CF_PVO	0,00E+00	0,00E+00	0,00E+00	9,23E+08	6,87E+07	**100,00 %**
CF_VOT	0,00E+00	0,00E+00	0,00E+00	8,27E+08	1,35E+08	**100,00 %**
GR_AND	0,00E+00	0,00E+00	0,00E+00	7,27E+04	4,28E+04	**100,00 %**
GR_OR	0,00E+00	0,00E+00	0,00E+00	7,27E+04	4,28E+04	**100,00 %**

GR_PVO	0,00E+00	0,00E+00	0,00E+00	6,21E+04	3,48E+04	**100,00 %**
GR_VOT	0,00E+00	0,00E+00	0,00E+00	7,27E+04	4,28E+04	**100,00 %**
TABU_AND	0,00E+00	0,00E+00	0,00E+00	5,29E+04	2,84E+04	**100,00 %**
TABU_OR	0,00E+00	1,84E+00	5,48E-01	1,04E+09	2,26E+08	**8,00 %**
TABU_PVO	0,00E+00	0,00E+00	0,00E+00	7,10E+08	1,72E+08	**100,00 %**
TABU_VOT	0,00E+00	0,00E+00	0,00E+00	7,15E+08	1,21E+08	**100,00 %**

APPENDIX B: SUCCESS RATE BELONGS TO BENCHMARK FUNCTIONS FOR F_C FRAMEWORK

Success rate of each F_C based hyperheuristic for benchmark function optimization is provided in the following tables (Table B.1- B.7). Success rate values are presented as the ratio of finding optimal solution with their standard deviations.

Table B.1. Success rate of SR based group decision making hyperheuristic patterns on Benchmark Functions

	SR_AND	**SR_OR**	**SR_PVO**	**SR_VOT**
F1	1,00	1,00	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	0,14	1,00	1,00
F4	0,92	0,32	0,74	0,76
F5	1,00	1,00	1,00	1,00
F6	1,00	0,00	1,00	1,00
F7	1,00	0,10	1,00	1,00
F8	1,00	0,46	0,98	0,88
F9	1,00	1,00	1,00	1,00
F10	1,00	0,06	1,00	1,00
F11	1,00	0,00	0,76	0,74
F12	1,00	0,00	1,00	1,00
F13	1,00	0,00	1,00	1,00
F14	1,00	1,00	1,00	1,00
avr.	***0,92***	***0,36***	***0,89***	***0,88***
std.	***0,27***	***0,44***	***0,27***	***0,27***

Table B.2. Success rate of RD based group decision making hyperheuristic patterns on Benchmark Functions

	RD_AND	RD_OR	RD_PVO	RD_VOT
F1	1,00	1,00	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	0,24	1,00	1,00
F4	0,88	0,42	0,84	0,74
F5	0,42	1,00	1,00	1,00
F6	1,00	0,00	0,98	1,00
F7	1,00	0,04	1,00	1,00
F8	0,84	0,40	0,62	0,44
F9	0,98	1,00	1,00	1,00
F10	1,00	0,06	0,92	1,00
F11	0,00	0,00	0,00	0,00
F12	1,00	0,00	1,00	1,00
F13	1,00	0,00	1,00	1,00
F14	1,00	1,00	1,00	1,00
avr.	***0,79***	***0,37***	***0,81***	***0,80***
std.	***0,37***	***0,44***	***0,36***	***0,37***

Table B. 3. Success rate of RP based group decision making hyperheuristic patterns on Benchmark Functions

	RP_AND	RP_OR	RP_PVO	RP_VOT
F1	1,00	0,98	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	0,22	1,00	1,00
F4	0,88	0,34	0,66	0,76
F5	1,00	1,00	1,00	1,00
F6	1,00	0,00	1,00	1,00
F7	1,00	0,00	1,00	1,00
F8	1,00	0,00	0,66	0,72
F9	1,00	0,58	1,00	1,00
F10	1,00	0,00	1,00	1,00

F11	1,00	0,00	0,72	0,86
F12	1,00	0,00	1,00	1,00
F13	1,00	0,00	1,00	1,00
F14	1,00	0,90	1,00	1,00
avr.	***0,92***	***0,29***	***0,86***	***0,88***
std.	***0,27***	***0,40***	***0,28***	***0,27***

Table B.4. Success rate of RPD based group decision making hyperheuristic patterns on Benchmark Functions

	RPD_AND	**RPD_OR**	**RPD_PVO**	**RPD_VOT**
F1	1,00	1,00	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	0,20	1,00	1,00
F4	0,98	0,30	0,84	0,84
F5	1,00	1,00	1,00	1,00
F6	1,00	0,00	1,00	1,00
F7	1,00	0,12	1,00	1,00
F8	1,00	1,00	1,00	1,00
F9	1,00	1,00	1,00	1,00
F10	1,00	0,84	1,00	1,00
F11	1,00	0,00	0,90	0,74
F12	1,00	0,00	1,00	1,00
F13	1,00	0,00	1,00	1,00
F14	1,00	1,00	1,00	1,00
avr.	***0,93***	***0,46***	***0,91***	***0,90***
std.	***0,27***	***0,47***	***0,27***	***0,27***

Table B.5. Success rate of CF based group decision making hyperheuristic patterns on Benchmark Functions

	CF_AND	CF_OR	CF_PVO	CF_VOT
F1	1,00	1,00	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	0,18	1,00	1,00
F4	0,88	0,88	0,78	0,88
F5	1,00	1,00	1,00	1,00
F6	1,00	0,00	1,00	1,00
F7	1,00	0,06	1,00	1,00
F8	1,00	1,00	1,00	1,00
F9	1,00	1,00	1,00	1,00
F10	1,00	0,78	1,00	1,00
F11	1,00	1,00	1,00	1,00
F12	1,00	0,00	1,00	1,00
F13	1,00	0,00	1,00	0,98
F14	1,00	0,98	1,00	1,00
avr.	***0,92***	***0,56***	***0,91***	***0,92***
std.	***0,27***	***0,48***	***0,27***	***0,27***

Table B.6. Success rate of GR based group decision making hyperheuristic patterns on Benchmark Functions

	GR_AND	GR_OR	GR_PVO	GR_VOT
F1	1,00	1,00	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	1,00	1,00	1,00
F4	0,86	0,98	1,00	1,00
F5	1,00	1,00	1,00	1,00
F6	1,00	1,00	1,00	1,00
F7	1,00	1,00	1,00	1,00
F8	1,00	1,00	1,00	1,00
F9	1,00	1,00	1,00	1,00

F10	1,00	1,00	1,00	1,00
F11	1,00	1,00	1,00	1,00
F12	1,00	1,00	1,00	1,00
F13	1,00	1,00	1,00	1,00
F14	1,00	1,00	1,00	1,00
avr.	***0,92***	***0,93***	***0,93***	***0,93***
std.	***0,27***	***0,27***	***0,27***	***0,27***

Table B.7. Success rate of TABU based group decision making hyperheuristic patterns on Benchmark Functions

	TABU_AND	**TABU_OR**	**TABU_PVO**	**TABU_VOT**
F1	1,00	1,00	1,00	1,00
F2	0,00	0,00	0,00	0,00
F3	1,00	0,26	1,00	1,00
F4	0,96	0,50	0,82	0,86
F5	1,00	1,00	1,00	1,00
F6	1,00	0,00	1,00	1,00
F7	1,00	0,10	1,00	1,00
F8	1,00	1,00	1,00	1,00
F9	1,00	1,00	1,00	1,00
F10	1,00	0,84	0,98	1,00
F11	1,00	0,00	0,92	0,84
F12	1,00	0,00	1,00	1,00
F13	1,00	0,00	1,00	1,00
F14	1,00	1,00	1,00	1,00
avr.	***0,93***	***0,48***	***0,91***	***0,91***
std.	***0,27***	***0,47***	***0,27***	***0,27***

APPENDIX C: EXPERIMENTAL RESULTS TABLES OF HYPERHEURISTICS PATTERNS ON EXAMINATION TIMETABLING DATA

For 21 university examination timetabling data, on following 14 tables, *Average Best Fitness* and *Average Fitness Evaluation per Execution* values are provided for experiments that are performed on FA framework with 28 different hyperheuristics. The hyperheuristics comes from 7 heuristic selection mechanisms which are SR, RD, RP, RPD, CF, GR, TABU and 4 move acceptance strategies that I proposed as group decision making methods, G-AND,G- OR, G-PVO, G-VOT. Also, related standard deviation values added to the tables as second columns under each table title.

Table C.1. Results of performance evaluations of hyperheuristic patterns on car-f-92

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-1,19E-02	-1,01E-02	9,83E-04	1,04E+08	2,07E+05
SR_OR	-1,22E-03	-1,12E-03	2,62E-05	8,68E+07	3,79E+04
SR_PVO	-1,21E-02	-1,00E-02	9,03E-04	9,73E+07	2,82E+06
SR_VOT	-2,07E-02	-1,73E-02	1,72E-03	1,03E+08	1,58E+06
RD_AND	-1,44E-02	-1,07E-02	1,31E-03	1,01E+08	2,25E+05
RD_OR	-1,17E-03	-1,12E-03	1,99E-05	8,85E+07	3,06E+04
RD_PVO	-1,24E-02	-1,00E-02	8,16E-04	1,01E+08	3,60E+04
RD_VOT	-2,06E-02	-1,76E-02	1,62E-03	1,02E+08	3,59E+05
RP_AND	-1,25E-02	-9,89E-03	1,01E-03	1,05E+08	2,64E+05
RP_OR	-1,19E-03	-1,12E-03	2,03E-05	8,79E+07	6,79E+05
RP_PVO	-1,19E-02	-1,01E-02	7,56E-04	1,01E+08	7,72E+04
RP_VOT	-2,04E-02	-1,71E-02	1,69E-03	1,05E+08	2,49E+05
RPD_AND	-1,29E-02	-1,01E-02	1,23E-03	1,05E+08	2,30E+05
RPD_OR	-1,19E-03	-1,12E-03	1,96E-05	8,79E+07	7,03E+04
RPD_PVO	-1,17E-02	-1,02E-02	6,68E-04	9,98E+07	1,68E+06

RPD_VOT	-2,11E-02	-1,73E-02	1,52E-03	1,05E+08	2,24E+05
CF_AND	-1,45E-02	-1,03E-02	1,35E-03	8,34E+07	1,35E+05
CF_OR	-1,16E-03	-1,12E-03	1,82E-05	6,99E+07	4,82E+04
CF_PVO	-1,15E-02	-9,95E-03	8,12E-04	7,78E+07	2,66E+06
CF_VOT	-2,07E-02	-1,64E-02	1,79E-03	8,33E+07	2,11E+05
GR_AND	-1,29E-02	-1,03E-02	1,10E-03	1,54E+08	4,38E+05
GR_OR	-3,52E-03	-3,32E-03	1,11E-04	1,41E+08	1,10E+05
GR_PVO	-1,72E-02	-1,49E-02	9,70E-04	1,45E+08	5,01E+06
GR_VOT	-2,17E-02	-1,85E-02	1,54E-03	1,54E+08	5,74E+05
TABU_AND	-1,21E-02	-9,86E-03	1,02E-03	1,01E+08	1,97E+05
TABU_OR	-1,17E-03	-1,13E-03	1,88E-05	8,53E+07	2,63E+04
TABU_PVO	-1,26E-02	-1,03E-02	7,58E-04	9,70E+07	3,03E+04
TABU_VOT	-2,15E-02	-1,76E-02	1,89E-03	1,00E+08	1,86E+05

Table C.2. Results of performance evaluations of hyperheuristic patterns on car-s-91

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-3,57E-01	-1,66E-01	5,82E-02	1,01E+08	8,18E+05
SR_OR	-1,37E-03	-1,33E-03	1,43E-05	8,37E+07	1,04E+05
SR_PVO	-4,55E-02	-3,54E-02	5,84E-03	9,08E+07	3,27E+06
SR_VOT	-1,00E+00	-5,55E-01	1,87E-01	1,00E+08	7,06E+06
RD_AND	-5,00E-01	-1,98E-01	8,40E-02	9,88E+07	1,50E+06
RD_OR	-1,39E-03	-1,33E-03	2,15E-05	8,48E+07	1,32E+06
RD_PVO	-4,63E-02	-3,36E-02	4,57E-03	9,65E+07	1,47E+06
RD_VOT	-7,14E-01	-5,07E-01	1,50E-01	9,98E+07	1,23E+06
RP_AND	-5,00E-01	-1,87E-01	7,93E-02	1,02E+08	1,47E+06
RP_OR	-1,41E-03	-1,33E-03	2,30E-05	8,40E+07	1,19E+06
RP_PVO	-5,49E-02	-3,83E-02	5,97E-03	9,58E+07	1,50E+06
RP_VOT	-1,00E+00	-5,27E-01	1,63E-01	1,01E+08	6,89E+06
RPD_AND	-4,55E-01	-1,86E-01	9,35E-02	1,02E+08	1,54E+06

RPD_OR	-1,40E-03	-1,34E-03	1,63E-05	8,45E+07	1,33E+06
RPD_PVO	-4,85E-02	-3,72E-02	5,16E-03	9,07E+07	2,48E+06
RPD_VOT	-1,00E+00	-5,29E-01	1,62E-01	1,02E+08	5,10E+06
CF_AND	-3,13E-01	-1,47E-01	5,11E-02	8,03E+07	2,24E+06
CF_OR	-1,48E-03	-1,33E-03	2,76E-05	6,97E+07	3,96E+04
CF_PVO	-4,50E-02	-3,45E-02	4,82E-03	7,79E+07	4,71E+04
CF_VOT	-1,00E+00	-4,74E-01	1,76E-01	8,06E+07	4,21E+06
GR_AND	-5,00E-01	-2,03E-01	7,51E-02	1,48E+08	1,60E+06
GR_OR	-4,85E-03	-4,45E-03	1,43E-04	1,36E+08	1,40E+06
GR_PVO	-1,35E-01	-9,20E-02	1,65E-02	1,39E+08	5,52E+06
GR_VOT	-1,00E+00	-5,73E-01	2,02E-01	1,45E+08	1,02E+07
TABU_AND	-3,57E-01	-1,80E-01	6,79E-02	9,61E+07	2,96E+06
TABU_OR	-1,52E-03	-1,36E-03	3,42E-05	8,28E+07	2,98E+04
TABU_PVO	-5,56E-02	-3,73E-02	5,57E-03	9,32E+07	1,43E+06
TABU_VOT	-1,00E+00	-5,11E-01	1,79E-01	9,59E+07	7,53E+06

Table C.3. Results of performance evaluations of hyperheuristic patterns on ear-f-83

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-5,73E-03	-4,61E-03	3,93E-04	1,29E+08	1,58E+06
SR_OR	-1,96E-03	-1,84E-03	3,71E-05	1,12E+08	1,39E+06
SR_PVO	-7,79E-03	-6,74E-03	4,15E-04	1,21E+08	2,44E+06
SR_VOT	-8,12E-03	-6,69E-03	5,52E-04	1,29E+08	6,52E+05
RD_AND	-5,13E-03	-4,18E-03	3,92E-04	1,22E+08	1,31E+06
RD_OR	-1,95E-03	-1,82E-03	4,68E-05	1,13E+08	3,77E+04
RD_PVO	-6,69E-03	-5,79E-03	3,40E-04	1,31E+08	1,01E+05
RD_VOT	-6,91E-03	-5,51E-03	4,03E-04	1,22E+08	1,57E+06
RP_AND	-5,32E-03	-4,63E-03	3,40E-04	1,31E+08	4,27E+05
RP_OR	-1,98E-03	-1,82E-03	4,48E-05	1,13E+08	5,60E+04
RP_PVO	-7,52E-03	-6,61E-03	3,33E-04	1,31E+08	1,16E+05

RP_VOT	-8,50E-03	-6,67E-03	5,18E-04	1,30E+08	1,55E+06
RPD_AND	-5,54E-03	-4,63E-03	4,61E-04	1,31E+08	4,60E+05
RPD_OR	-2,00E-03	-1,88E-03	4,08E-05	1,12E+08	5,74E+04
RPD_PVO	-8,01E-03	-6,76E-03	4,34E-04	1,25E+08	4,14E+06
RPD_VOT	-7,49E-03	-6,62E-03	4,75E-04	1,30E+08	4,13E+05
CF_AND	-5,21E-03	-4,55E-03	3,88E-04	9,32E+07	2,55E+06
CF_OR	-2,09E-03	-1,92E-03	4,45E-05	8,68E+07	1,09E+06
CF_PVO	-7,30E-03	-6,56E-03	3,49E-04	9,83E+07	1,14E+06
CF_VOT	-7,91E-03	-6,56E-03	4,67E-04	9,91E+07	2,08E+05
GR_AND	-5,93E-03	-4,56E-03	4,17E-04	2,12E+08	2,60E+06
GR_OR	-3,63E-03	-3,34E-03	8,61E-05	1,98E+08	2,26E+06
GR_PVO	-8,61E-03	-7,35E-03	4,38E-04	1,95E+08	1,98E+06
GR_VOT	-8,18E-03	-7,10E-03	5,35E-04	2,12E+08	1,05E+06
TABU_AND	-5,85E-03	-4,71E-03	4,57E-04	1,22E+08	3,53E+06
TABU_OR	-2,01E-03	-1,91E-03	4,60E-05	1,08E+08	1,20E+06
TABU_PVO	-7,59E-03	-6,71E-03	4,45E-04	1,24E+08	1,70E+05
TABU_VOT	-7,73E-03	-6,61E-03	4,41E-04	1,24E+08	5,95E+05

Table C.4. Results of performance evaluations of hyperheuristic patterns on hec-s-92

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-1,69E-02	-8,44E-03	2,13E-03	1,47E+08	6,26E+05
SR_OR	-3,14E-03	-2,70E-03	1,67E-04	1,29E+08	1,83E+05
SR_PVO	-2,62E-02	-2,21E-02	2,20E-03	1,42E+08	3,72E+06
SR_VOT	-4,03E-02	-2,24E-02	6,15E-03	1,47E+08	2,09E+06
RD_AND	-1,33E-02	-6,44E-03	1,77E-03	1,41E+08	2,40E+06
RD_OR	-3,27E-03	-2,71E-03	1,65E-04	1,32E+08	1,16E+05
RD_PVO	-2,16E-02	-1,54E-02	2,43E-03	1,53E+08	8,01E+04
RD_VOT	-1,14E-02	-7,84E-03	1,37E-03	1,43E+08	2,54E+06
RP_AND	-1,26E-02	-8,34E-03	1,69E-03	1,50E+08	7,10E+05
RP_OR	-3,15E-03	-2,71E-03	1,54E-04	1,31E+08	1,37E+06

RP_PVO	-3,09E-02	-2,27E-02	2,66E-03	1,53E+08	1,41E+05
RP_VOT	-3,62E-02	-1,99E-02	4,68E-03	1,51E+08	6,99E+05
RPD_AND	-1,27E-02	-8,03E-03	1,70E-03	1,50E+08	7,73E+05
RPD_OR	-3,06E-03	-2,67E-03	1,18E-04	1,30E+08	1,83E+05
RPD_PVO	-2,75E-02	-2,33E-02	2,17E-03	1,47E+08	5,04E+06
RPD_VOT	-3,65E-02	-2,17E-02	5,19E-03	1,50E+08	7,36E+05
CF_AND	-1,62E-02	-8,08E-03	1,97E-03	1,10E+08	5,04E+05
CF_OR	-3,08E-03	-2,63E-03	1,31E-04	9,40E+07	3,25E+05
CF_PVO	-2,76E-02	-2,19E-02	2,54E-03	1,09E+08	3,00E+06
CF_VOT	-2,99E-02	-1,97E-02	3,94E-03	1,10E+08	6,17E+05
GR_AND	-1,29E-02	-8,32E-03	1,71E-03	2,67E+08	2,17E+06
GR_OR	-9,26E-03	-7,90E-03	5,13E-04	2,48E+08	3,03E+06
GR_PVO	-3,73E-02	-2,66E-02	4,97E-03	2,57E+08	8,58E+06
GR_VOT	-4,00E-02	-2,56E-02	6,27E-03	2,68E+08	2,09E+06
TABU_AND	-1,39E-02	-8,43E-03	2,22E-03	1,41E+08	7,83E+05
TABU_OR	-3,23E-03	-2,71E-03	1,44E-04	1,24E+08	8,65E+05
TABU_PVO	-2,84E-02	-2,26E-02	2,53E-03	1,44E+08	7,58E+04
TABU_VOT	-3,47E-02	-2,07E-02	4,97E-03	1,41E+08	6,64E+05

Table C.5. Results of performance evaluations of hyperheuristic patterns on kfu-s-93

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-3,45E-02	-2,43E-02	4,48E-03	9,61E+07	1,35E+06
SR_OR	-1,56E-03	-1,44E-03	4,00E-05	7,67E+07	9,60E+05
SR_PVO	-3,65E-02	-2,86E-02	3,15E-03	8,85E+07	3,45E+04
SR_VOT	-5,10E-02	-4,35E-02	2,61E-03	9,69E+07	1,19E+05
RD_AND	-3,65E-02	-2,53E-02	4,85E-03	9,49E+07	1,75E+06
RD_OR	-1,51E-03	-1,43E-03	3,67E-05	7,71E+07	1,15E+06
RD_PVO	-3,40E-02	-2,74E-02	2,95E-03	8,78E+07	1,34E+06
RD_VOT	-5,21E-02	-4,26E-02	3,74E-03	9,55E+07	1,51E+06

RP_AND	-3,62E-02	-2,48E-02	4,58E-03	9,67E+07	1,39E+06
RP_OR	-1,50E-03	-1,43E-03	2,73E-05	7,71E+07	4,13E+05
RP_PVO	-3,70E-02	-2,82E-02	3,35E-03	8,90E+07	1,33E+06
RP_VOT	-5,38E-02	-4,44E-02	3,91E-03	9,74E+07	1,20E+06
RPD_AND	-3,50E-02	-2,54E-02	4,17E-03	9,65E+07	1,27E+06
RPD_OR	-1,52E-03	-1,43E-03	3,57E-05	7,70E+07	8,54E+04
RPD_PVO	-3,60E-02	-2,83E-02	3,37E-03	8,80E+07	1,83E+06
RPD_VOT	-5,21E-02	-4,42E-02	3,67E-03	9,73E+07	1,71E+05
CF_AND	-3,62E-02	-2,32E-02	5,28E-03	7,52E+07	2,32E+06
CF_OR	-1,56E-03	-1,44E-03	4,27E-05	6,44E+07	3,58E+04
CF_PVO	-3,50E-02	-2,77E-02	3,77E-03	7,26E+07	4,67E+05
CF_VOT	-5,32E-02	-4,37E-02	4,23E-03	7,80E+07	1,04E+06
GR_AND	-3,23E-02	-2,40E-02	4,59E-03	1,37E+08	2,06E+06
GR_OR	-1,40E-02	-1,09E-02	9,89E-04	1,29E+08	2,73E+05
GR_PVO	-4,95E-02	-4,16E-02	3,11E-03	1,27E+08	1,47E+06
GR_VOT	-5,21E-02	-4,44E-02	3,64E-03	1,38E+08	2,15E+05
TABU_AND	-3,88E-02	-2,52E-02	4,61E-03	9,31E+07	1,67E+06
TABU_OR	-1,54E-03	-1,44E-03	3,45E-05	7,46E+07	8,88E+04
TABU_PVO	-3,55E-02	-2,87E-02	3,37E-03	8,59E+07	1,33E+06
TABU_VOT	-5,26E-02	-4,40E-02	3,72E-03	9,30E+07	1,31E+06

Table C.6. Results of performance evaluations of hyperheuristic patterns on lse-f-91

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-1,31E-02	-1,03E-02	1,51E-03	9,99E+07	3,04E+05
SR_OR	-2,59E-03	-2,42E-03	5,80E-05	8,67E+07	7,80E+05
SR_PVO	-1,67E-02	-1,37E-02	1,30E-03	9,40E+07	3,10E+06
SR_VOT	-1,92E-02	-1,52E-02	2,02E-03	9,91E+07	1,21E+06
RD_AND	-1,24E-02	-9,65E-03	1,64E-03	9,79E+07	3,46E+05
RD_OR	-2,67E-03	-2,39E-03	8,05E-05	8,79E+07	4,95E+04
RD_PVO	-1,48E-02	-1,27E-02	1,25E-03	9,95E+07	3,64E+04

RD_VOT	-1,84E-02	-1,44E-02	1,94E-03	9,77E+07	3,38E+05
RP_AND	-1,28E-02	-1,04E-02	1,30E-03	1,01E+08	3,93E+05
RP_OR	-2,65E-03	-2,40E-03	7,77E-05	8,78E+07	4,63E+04
RP_PVO	-1,75E-02	-1,33E-02	1,44E-03	9,95E+07	5,66E+04
RP_VOT	-1,86E-02	-1,53E-02	1,58E-03	1,00E+08	1,13E+06
RPD_AND	-1,47E-02	-1,01E-02	1,59E-03	1,00E+08	3,47E+05
RPD_OR	-2,68E-03	-2,50E-03	6,39E-05	8,77E+07	5,83E+04
RPD_PVO	-1,59E-02	-1,34E-02	1,23E-03	9,53E+07	3,32E+06
RPD_VOT	-2,04E-02	-1,56E-02	2,10E-03	1,00E+08	3,30E+05
CF_AND	-1,45E-02	-1,04E-02	1,51E-03	8,05E+07	1,16E+06
CF_OR	-2,78E-03	-2,52E-03	7,27E-05	7,19E+07	1,25E+05
CF_PVO	-1,56E-02	-1,28E-02	1,36E-03	7,92E+07	5,04E+04
CF_VOT	-1,97E-02	-1,44E-02	2,40E-03	8,07E+07	2,37E+05
GR_AND	-1,37E-02	-1,04E-02	1,54E-03	1,44E+08	7,95E+05
GR_OR	-7,73E-03	-6,85E-03	3,55E-04	1,35E+08	1,98E+06
GR_PVO	-1,97E-02	-1,61E-02	1,88E-03	1,42E+08	3,29E+05
GR_VOT	-1,74E-02	-1,44E-02	1,74E-03	1,43E+08	7,19E+05
TABU_AND	-1,35E-02	-1,00E-02	1,54E-03	9,59E+07	1,75E+06
TABU_OR	-2,68E-03	-2,53E-03	5,57E-05	8,50E+07	1,20E+06
TABU_PVO	-1,53E-02	-1,33E-02	1,12E-03	9,55E+07	3,81E+04
TABU_VOT	-1,95E-02	-1,52E-02	2,16E-03	9,65E+07	3,70E+05

Table C.7. Results of performance evaluations of hyperheuristic patterns on pur-s-93

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-1,47E-03	-1,37E-03	6,82E-05	3,65E+07	9,81E+04
SR_OR	-2,89E-04	-2,80E-04	3,36E-06	3,00E+07	5,95E+03
SR_PVO	-1,04E-03	-9,60E-04	3,18E-05	3,15E+07	1,16E+06
SR_VOT	-1,63E-03	-1,41E-03	7,44E-05	3,66E+07	3,23E+05
RD_AND	-1,69E-03	-1,53E-03	8,72E-05	3,81E+07	1,39E+05
RD_OR	-2,83E-04	-2,76E-04	2,68E-06	3,01E+07	5,95E+03

RD_PVO	-9,45E-04	-8,92E-04	2,45E-05	3,29E+07	3,42E+05
RD_VOT	-1,79E-03	-1,56E-03	9,36E-05	3,80E+07	1,55E+05
RP_AND	-1,63E-03	-1,39E-03	8,16E-05	3,64E+07	5,57E+05
RP_OR	-2,90E-04	-2,80E-04	3,98E-06	2,97E+07	4,26E+05
RP_PVO	-1,03E-03	-9,70E-04	3,06E-05	3,30E+07	1,40E+04
RP_VOT	-1,57E-03	-1,39E-03	6,83E-05	3,69E+07	1,09E+05
RPD_AND	-1,53E-03	-1,37E-03	6,48E-05	3,67E+07	3,00E+05
RPD_OR	-3,03E-04	-2,88E-04	4,26E-06	2,99E+07	4,35E+05
RPD_PVO	-1,03E-03	-9,65E-04	3,25E-05	3,12E+07	4,02E+05
RPD_VOT	-1,57E-03	-1,40E-03	7,55E-05	3,66E+07	1,41E+05
CF_AND	-1,50E-03	-1,37E-03	6,64E-05	3,35E+07	4,81E+05
CF_OR	-3,03E-04	-2,84E-04	6,11E-06	2,77E+07	6,76E+04
CF_PVO	-9,99E-04	-9,46E-04	2,95E-05	2,92E+07	7,15E+05
CF_VOT	-1,61E-03	-1,39E-03	7,34E-05	3,36E+07	4,26E+05
GR_AND	-1,52E-03	-1,37E-03	6,91E-05	4,13E+07	1,36E+05
GR_OR	-9,23E-04	-8,78E-04	2,20E-05	3,88E+07	3,75E+04
GR_PVO	-1,89E-03	-1,63E-03	9,71E-05	3,93E+07	1,04E+06
GR_VOT	-1,54E-03	-1,40E-03	6,71E-05	4,12E+07	1,35E+05
TABU_AND	-1,49E-03	-1,37E-03	5,75E-05	3,47E+07	9,64E+05
TABU_OR	-3,03E-04	-2,93E-04	4,56E-06	2,97E+07	6,80E+03
TABU_PVO	-1,06E-03	-9,77E-04	3,51E-05	3,26E+07	1,08E+04
TABU_VOT	-1,55E-03	-1,40E-03	6,64E-05	3,62E+07	1,13E+05

Table C.8. Results of performance evaluations of hyperheuristic patterns on rye-s-93

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-1,49E-02	-9,59E-03	2,04E-03	9,70E+07	1,34E+06
SR_OR	-7,88E-04	-7,24E-04	2,41E-05	8,36E+07	1,24E+06
SR_PVO	-1,45E-02	-1,03E-02	1,44E-03	9,11E+07	3,06E+06
SR_VOT	-2,16E-02	-1,47E-02	2,75E-03	9,74E+07	3,05E+05

RD_AND	-1,27E-02	-9,04E-03	1,67E-03	9,51E+07	1,47E+06
RD_OR	-7,58E-04	-7,13E-04	1,72E-05	8,44E+07	1,26E+06
RD_PVO	-1,22E-02	-1,02E-02	1,04E-03	9,61E+07	1,08E+06
RD_VOT	-2,02E-02	-1,47E-02	2,47E-03	9,52E+07	1,54E+06
RP_AND	-1,54E-02	-9,30E-03	2,06E-03	9,84E+07	3,19E+05
RP_OR	-7,59E-04	-7,22E-04	1,98E-05	8,48E+07	6,87E+04
RP_PVO	-1,32E-02	-1,03E-02	1,29E-03	9,54E+07	1,45E+06
RP_VOT	-2,14E-02	-1,43E-02	2,83E-03	9,75E+07	1,54E+06
RPD_AND	-1,46E-02	-9,19E-03	2,35E-03	9,76E+07	1,35E+06
RPD_OR	-8,19E-04	-7,34E-04	1,99E-05	8,31E+07	1,30E+06
RPD_PVO	-1,37E-02	-1,02E-02	1,47E-03	8,85E+07	1,68E+06
RPD_VOT	-2,01E-02	-1,54E-02	2,01E-03	9,79E+07	3,19E+05
CF_AND	-1,58E-02	-9,57E-03	1,96E-03	7,40E+07	2,05E+06
CF_OR	-8,31E-04	-7,45E-04	1,94E-05	6,86E+07	9,81E+05
CF_PVO	-1,32E-02	-1,01E-02	1,19E-03	7,61E+07	1,08E+06
CF_VOT	-2,33E-02	-1,50E-02	2,33E-03	7,87E+07	4,87E+05
GR_AND	-1,30E-02	-9,06E-03	1,85E-03	1,39E+08	2,12E+06
GR_OR	-3,83E-03	-3,61E-03	1,01E-04	1,30E+08	2,00E+06
GR_PVO	-1,99E-02	-1,41E-02	2,20E-03	1,35E+08	3,56E+06
GR_VOT	-2,39E-02	-1,44E-02	2,88E-03	1,39E+08	6,37E+05
TABU_AND	-1,63E-02	-9,81E-03	2,27E-03	9,26E+07	2,60E+06
TABU_OR	-7,82E-04	-7,40E-04	1,63E-05	8,21E+07	5,04E+04
TABU_PVO	-1,40E-02	-1,01E-02	1,26E-03	9,14E+07	1,34E+06
TABU_VOT	-2,22E-02	-1,53E-02	2,25E-03	9,37E+07	1,42E+06

Table C.9. Results of performance evaluations of hyperheuristic patterns on sta-f-83

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-2,70E-03	-2,64E-03	5,70E-05	1,27E+08	2,34E+05
SR_OR	-1,29E-03	-1,25E-03	1,74E-05	1,11E+08	5,86E+04

SR_PVO	-2,69E-03	-2,68E-03	1,01E-05	1,21E+08	4,07E+06
SR_VOT	-2,69E-03	-2,68E-03	1,22E-05	1,26E+08	1,61E+06
RD_AND	-2,70E-03	-2,64E-03	5,47E-05	1,22E+08	3,71E+05
RD_OR	-1,31E-03	-1,25E-03	2,13E-05	1,12E+08	4,40E+04
RD_PVO	-2,70E-03	-2,68E-03	1,24E-05	1,27E+08	8,54E+05
RD_VOT	-2,70E-03	-2,67E-03	1,43E-05	1,22E+08	3,53E+05
RP_AND	-2,69E-03	-2,63E-03	5,72E-05	1,29E+08	2,37E+05
RP_OR	-1,34E-03	-1,25E-03	1,99E-05	1,12E+08	6,87E+04
RP_PVO	-2,70E-03	-2,68E-03	1,03E-05	1,28E+08	7,67E+04
RP_VOT	-2,70E-03	-2,67E-03	1,95E-05	1,29E+08	2,53E+05
RPD_AND	-2,69E-03	-2,63E-03	6,03E-05	1,28E+08	2,50E+05
RPD_OR	-1,33E-03	-1,26E-03	2,34E-05	1,11E+08	1,33E+06
RPD_PVO	-2,70E-03	-2,68E-03	1,04E-05	1,28E+08	8,00E+04
RPD_VOT	-2,69E-03	-2,67E-03	1,28E-05	1,28E+08	2,19E+05
CF_AND	-2,69E-03	-2,64E-03	5,51E-05	9,37E+07	3,39E+06
CF_OR	-1,31E-03	-1,25E-03	2,01E-05	8,73E+07	3,79E+04
CF_PVO	-2,70E-03	-2,68E-03	9,26E-06	9,64E+07	3,73E+04
CF_VOT	-2,69E-03	-2,67E-03	1,55E-05	9,74E+07	1,06E+06
GR_AND	-2,70E-03	-2,62E-03	6,39E-05	2,06E+08	6,03E+05
GR_OR	-2,31E-03	-2,23E-03	3,60E-05	1,91E+08	1,24E+05
GR_PVO	-2,69E-03	-2,68E-03	9,11E-06	1,89E+08	1,50E+05
GR_VOT	-2,69E-03	-2,68E-03	1,26E-05	2,06E+08	4,33E+05
TABU_AND	-2,70E-03	-2,64E-03	5,36E-05	1,18E+08	3,81E+06
TABU_OR	-1,31E-03	-1,26E-03	2,09E-05	1,08E+08	1,38E+06
TABU_PVO	-2,69E-03	-2,68E-03	1,03E-05	1,22E+08	5,12E+04
TABU_VOT	-2,70E-03	-2,68E-03	1,59E-05	1,22E+08	1,49E+05

Table C.10. Results of performance evaluations of hyperheuristic patterns on tre-s-92

	Best Fit.	**Avg. Best Fit.**	**Avg. Num. of Eval.**

SR_AND	-5,68E-02	-4,12E-02	6,00E-03	1,24E+08	2,26E+05
SR_OR	-3,99E-03	-3,74E-03	7,39E-05	1,07E+08	3,67E+04
SR_PVO	-5,95E-02	-4,46E-02	5,26E-03	1,18E+08	3,54E+06
SR_VOT	-1,85E-01	-1,19E-01	2,56E-02	1,23E+08	2,51E+05
RD_AND	-5,68E-02	-4,06E-02	6,37E-03	1,19E+08	1,59E+06
RD_OR	-4,16E-03	-3,76E-03	1,16E-04	1,07E+08	1,40E+06
RD_PVO	-5,05E-02	-4,37E-02	3,99E-03	1,23E+08	1,46E+06
RD_VOT	-1,92E-01	-1,05E-01	2,47E-02	1,17E+08	1,65E+06
RP_AND	-6,41E-02	-4,19E-02	8,14E-03	1,25E+08	1,60E+06
RP_OR	-3,98E-03	-3,75E-03	9,60E-05	1,08E+08	1,43E+06
RP_PVO	-5,88E-02	-4,69E-02	5,77E-03	1,24E+08	7,35E+04
RP_VOT	-1,72E-01	-1,14E-01	2,02E-02	1,26E+08	2,92E+05
RPD_AND	-6,58E-02	-4,08E-02	7,44E-03	1,24E+08	1,45E+06
RPD_OR	-4,03E-03	-3,77E-03	8,58E-05	1,08E+08	6,80E+04
RPD_PVO	-6,02E-02	-4,56E-02	5,92E-03	1,20E+08	4,18E+06
RPD_VOT	-2,17E-01	-1,23E-01	2,96E-02	1,25E+08	2,70E+05
CF_AND	-5,68E-02	-4,17E-02	6,51E-03	9,56E+07	1,65E+05
CF_OR	-3,89E-03	-3,72E-03	7,49E-05	7,78E+07	7,27E+06
CF_PVO	-5,75E-02	-4,62E-02	5,85E-03	9,46E+07	3,95E+04
CF_VOT	-1,56E-01	-1,11E-01	2,12E-02	9,57E+07	6,59E+05
GR_AND	-5,81E-02	-4,20E-02	7,60E-03	1,97E+08	2,76E+06
GR_OR	-1,19E-02	-1,06E-02	4,52E-04	1,85E+08	5,46E+04
GR_PVO	-8,77E-02	-7,00E-02	7,51E-03	1,90E+08	8,20E+06
GR_VOT	-2,08E-01	-1,31E-01	2,49E-02	1,98E+08	6,45E+05
TABU_AND	-5,56E-02	-4,07E-02	6,65E-03	1,15E+08	3,83E+06
TABU_OR	-4,18E-03	-3,82E-03	8,62E-05	1,05E+08	3,15E+04
TABU_PVO	-5,68E-02	-4,59E-02	5,56E-03	1,19E+08	4,16E+04
TABU_VOT	-2,00E-01	-1,14E-01	2,39E-02	1,19E+08	2,04E+05

Table C.11. Results of performance evaluations of hyperheuristic patterns on uta-s-92

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-2,27E-02	-1,89E-02	1,91E-03	9,74E+07	1,48E+06
SR_OR	-1,24E-03	-1,19E-03	1,55E-05	8,04E+07	1,25E+06
SR_PVO	-1,57E-02	-1,34E-02	8,47E-04	8,92E+07	2,93E+06
SR_VOT	-2,89E-02	-2,47E-02	1,71E-03	9,86E+07	9,83E+04
RD_AND	-2,28E-02	-1,79E-02	1,88E-03	9,71E+07	2,55E+05
RD_OR	-1,26E-03	-1,19E-03	1,94E-05	8,20E+07	2,70E+04
RD_PVO	-1,21E-02	-1,06E-02	6,77E-04	9,35E+07	3,88E+04
RD_VOT	-2,75E-02	-2,26E-02	2,00E-03	9,79E+07	2,49E+05
RP_AND	-2,42E-02	-1,88E-02	1,76E-03	9,89E+07	8,77E+04
RP_OR	-1,25E-03	-1,19E-03	1,63E-05	8,17E+07	7,21E+04
RP_PVO	-1,55E-02	-1,34E-02	8,90E-04	9,21E+07	1,51E+06
RP_VOT	-2,79E-02	-2,42E-02	1,74E-03	9,89E+07	1,53E+06
RPD_AND	-2,35E-02	-1,86E-02	2,02E-03	9,89E+07	7,38E+05
RPD_OR	-1,23E-03	-1,19E-03	1,44E-05	8,20E+07	5,91E+04
RPD_PVO	-1,48E-02	-1,34E-02	7,27E-04	8,74E+07	1,26E+06
RPD_VOT	-2,89E-02	-2,53E-02	1,81E-03	9,90E+07	1,02E+05
CF_AND	-2,35E-02	-1,87E-02	1,94E-03	7,86E+07	1,95E+06
CF_OR	-1,29E-03	-1,21E-03	3,15E-05	6,72E+07	8,39E+05
CF_PVO	-1,46E-02	-1,30E-02	6,71E-04	7,52E+07	4,17E+04
CF_VOT	-2,89E-02	-2,39E-02	1,99E-03	7,95E+07	6,79E+04
GR_AND	-2,20E-02	-1,88E-02	1,47E-03	1,42E+08	1,58E+05
GR_OR	-3,68E-03	-3,50E-03	8,20E-05	1,27E+08	1,99E+06
GR_PVO	-2,28E-02	-1,94E-02	1,21E-03	1,34E+08	3,37E+06
GR_VOT	-2,91E-02	-2,55E-02	1,75E-03	1,42E+08	1,84E+05
TABU_AND	-2,42E-02	-1,85E-02	2,00E-03	9,44E+07	1,32E+06
TABU_OR	-1,25E-03	-1,21E-03	1,72E-05	7,98E+07	3,24E+04
TABU_PVO	-1,53E-02	-1,35E-02	8,57E-04	8,94E+07	1,43E+06
TABU_VOT	-2,86E-02	-2,44E-02	1,71E-03	9,44E+07	1,40E+06

Table C.12. Results of performance evaluations of hyperheuristic patterns on ute-s-92

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-1,79E-03	-1,57E-03	1,09E-04	1,06E+08	6,45E+05
SR_OR	-1,03E-03	-9,73E-04	2,09E-05	9,31E+07	6,47E+04
SR_PVO	-2,38E-03	-2,19E-03	8,24E-05	1,04E+08	2,13E+06
SR_VOT	-2,25E-03	-2,05E-03	1,03E-04	1,05E+08	1,40E+06
RD_AND	-1,86E-03	-1,58E-03	8,60E-05	1,02E+08	1,63E+06
RD_OR	-1,03E-03	-9,70E-04	2,00E-05	9,42E+07	5,98E+04
RD_PVO	-2,36E-03	-2,13E-03	8,72E-05	1,06E+08	1,36E+06
RD_VOT	-2,27E-03	-2,01E-03	1,19E-04	1,00E+08	1,01E+06
RP_AND	-1,86E-03	-1,55E-03	1,14E-04	1,07E+08	5,97E+05
RP_OR	-1,04E-03	-9,69E-04	2,05E-05	9,38E+07	6,24E+04
RP_PVO	-2,38E-03	-2,17E-03	9,55E-05	1,07E+08	6,13E+04
RP_VOT	-2,37E-03	-2,05E-03	1,13E-04	1,07E+08	4,80E+05
RPD_AND	-1,77E-03	-1,56E-03	7,95E-05	1,05E+08	1,56E+06
RPD_OR	-1,01E-03	-9,73E-04	1,74E-05	9,29E+07	1,15E+06
RPD_PVO	-2,40E-03	-2,14E-03	1,08E-04	1,00E+08	2,09E+06
RPD_VOT	-2,29E-03	-2,05E-03	1,02E-04	1,05E+08	1,21E+06
CF_AND	-1,78E-03	-1,54E-03	9,88E-05	8,18E+07	2,53E+06
CF_OR	-1,01E-03	-9,71E-04	1,90E-05	7,54E+07	6,61E+05
CF_PVO	-2,35E-03	-2,17E-03	8,50E-05	8,33E+07	1,08E+06
CF_VOT	-2,26E-03	-2,01E-03	1,02E-04	8,38E+07	1,11E+06
GR_AND	-1,80E-03	-1,56E-03	1,03E-04	1,56E+08	1,42E+06
GR_OR	-1,95E-03	-1,75E-03	4,80E-05	1,48E+08	8,84E+04
GR_PVO	-2,47E-03	-2,27E-03	7,63E-05	1,55E+08	1,51E+05
GR_VOT	-2,31E-03	-2,10E-03	9,37E-05	1,55E+08	9,27E+05
TABU_AND	-1,86E-03	-1,56E-03	1,24E-04	1,01E+08	2,39E+06
TABU_OR	-1,07E-03	-9,78E-04	2,03E-05	8,98E+07	1,22E+06
TABU_PVO	-2,33E-03	-2,17E-03	9,43E-05	1,01E+08	1,36E+06
TABU_VOT	-2,31E-03	-2,04E-03	1,05E-04	1,01E+08	1,44E+06

Table C.13. Results of performance evaluations of hyperheuristic patterns on yor-f-83

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-6,05E-03	-5,21E-03	3,85E-04	1,30E+08	4,28E+05
SR_OR	-2,48E-03	-2,33E-03	4,24E-05	1,14E+08	1,45E+06
SR_PVO	-8,85E-03	-7,86E-03	4,99E-04	1,29E+08	2,46E+06
SR_VOT	-1,01E-02	-8,31E-03	6,89E-04	1,30E+08	3,76E+05
RD_AND	-5,94E-03	-4,89E-03	4,68E-04	1,20E+08	1,65E+06
RD_OR	-2,45E-03	-2,33E-03	4,79E-05	1,15E+08	1,40E+06
RD_PVO	-8,13E-03	-6,96E-03	3,85E-04	1,33E+08	6,86E+05
RD_VOT	-7,84E-03	-6,69E-03	5,73E-04	1,21E+08	1,87E+06
RP_AND	-6,16E-03	-5,25E-03	3,53E-04	1,31E+08	1,51E+06
RP_OR	-2,43E-03	-2,33E-03	3,93E-05	1,14E+08	1,47E+06
RP_PVO	-8,71E-03	-7,81E-03	4,56E-04	1,32E+08	1,07E+05
RP_VOT	-9,90E-03	-8,40E-03	7,57E-04	1,31E+08	1,66E+06
RPD_AND	-6,24E-03	-5,15E-03	3,17E-04	1,31E+08	4,31E+05
RPD_OR	-2,49E-03	-2,38E-03	3,20E-05	1,14E+08	8,59E+04
RPD_PVO	-8,76E-03	-7,69E-03	3,89E-04	1,31E+08	8,36E+04
RPD_VOT	-1,09E-02	-8,52E-03	7,07E-04	1,31E+08	4,35E+05
CF_AND	-6,28E-03	-5,22E-03	3,29E-04	9,92E+07	4,90E+05
CF_OR	-2,55E-03	-2,41E-03	4,21E-05	7,83E+07	6,85E+06
CF_PVO	-9,01E-03	-7,67E-03	5,93E-04	9,93E+07	4,22E+04
CF_VOT	-9,58E-03	-8,13E-03	7,16E-04	9,94E+07	2,55E+05
GR_AND	-5,90E-03	-5,20E-03	3,29E-04	2,12E+08	2,79E+06
GR_OR	-3,65E-03	-3,55E-03	4,86E-05	2,00E+08	4,40E+04
GR_PVO	-1,01E-02	-9,07E-03	5,84E-04	1,99E+08	2,80E+06
GR_VOT	-1,10E-02	-8,93E-03	6,94E-04	2,14E+08	8,57E+05
TABU_AND	-5,87E-03	-5,17E-03	3,73E-04	1,21E+08	4,07E+06
TABU_OR	-2,51E-03	-2,41E-03	3,90E-05	1,09E+08	2,68E+06
TABU_PVO	-8,99E-03	-7,73E-03	5,01E-04	1,25E+08	1,83E+06
TABU_VOT	-1,05E-02	-8,54E-03	8,02E-04	1,24E+08	1,31E+06

Table C.14. Results of performance evaluations of hyperheuristic patterns on yue20011

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-8,47E-02	-5,88E-02	1,23E-02	1,29E+08	2,07E+06
SR_OR	-9,31E-03	-8,55E-03	2,58E-04	1,11E+08	5,22E+04
SR_PVO	-1,00E-01	-7,75E-02	8,30E-03	1,24E+08	4,00E+06
SR_VOT	-1,28E-01	-1,07E-01	1,03E-02	1,30E+08	2,64E+05
RD_AND	-9,43E-02	-5,73E-02	1,21E-02	1,26E+08	6,78E+05
RD_OR	-9,75E-03	-8,62E-03	3,64E-04	1,12E+08	5,28E+04
RD_PVO	-9,62E-02	-7,64E-02	8,57E-03	1,28E+08	3,86E+04
RD_VOT	-1,28E-01	-1,04E-01	9,62E-03	1,27E+08	5,63E+05
RP_AND	-8,93E-02	-5,82E-02	1,27E-02	1,30E+08	1,84E+06
RP_OR	-1,01E-02	-8,62E-03	3,59E-04	1,12E+08	7,12E+04
RP_PVO	-1,09E-01	-7,92E-02	9,36E-03	1,28E+08	1,71E+06
RP_VOT	-1,32E-01	-1,09E-01	1,07E-02	1,32E+08	2,51E+05
RPD_AND	-8,33E-02	-5,80E-02	1,23E-02	1,31E+08	2,69E+05
RPD_OR	-9,28E-03	-8,59E-03	2,74E-04	1,11E+08	1,46E+06
RPD_PVO	-1,02E-01	-7,80E-02	1,04E-02	1,23E+08	5,35E+06
RPD_VOT	-1,28E-01	-1,08E-01	1,05E-02	1,31E+08	2,85E+05
CF_AND	-1,11E-01	-5,70E-02	1,38E-02	9,58E+07	3,19E+06
CF_OR	-9,43E-03	-8,56E-03	3,22E-04	8,52E+07	9,39E+05
CF_PVO	-9,62E-02	-7,66E-02	8,67E-03	9,76E+07	6,93E+04
CF_VOT	-1,39E-01	-1,08E-01	9,25E-03	9,93E+07	1,53E+05
GR_AND	-9,80E-02	-6,02E-02	1,32E-02	2,12E+08	5,66E+05
GR_OR	-4,13E-02	-3,16E-02	3,05E-03	1,96E+08	2,57E+06
GR_PVO	-1,25E-01	-1,01E-01	8,70E-03	2,09E+08	1,57E+05
GR_VOT	-1,35E-01	-1,09E-01	1,19E-02	2,13E+08	6,79E+05
TABU_AND	-8,20E-02	-5,51E-02	1,01E-02	1,20E+08	3,61E+06
TABU_OR	-9,28E-03	-8,60E-03	2,74E-04	1,07E+08	6,21E+04
TABU_PVO	-1,00E-01	-7,84E-02	8,58E-03	1,22E+08	1,01E+06
TABU_VOT	-1,28E-01	-1,05E-01	1,00E-02	1,24E+08	1,03E+06

Table C.15. Results of performance evaluations of hyperheuristic patterns on yue20012

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-6,33E-02	-4,76E-02	8,44E-03	1,31E+08	1,61E+05
SR_OR	-6,36E-03	-6,05E-03	1,43E-04	1,12E+08	1,09E+06
SR_PVO	-7,94E-02	-5,85E-02	6,90E-03	1,20E+08	7,43E+04
SR_VOT	-1,14E-01	-9,33E-02	1,09E-02	1,30E+08	1,64E+06
RD_AND	-6,17E-02	-4,98E-02	6,13E-03	1,22E+08	2,94E+05
RD_OR	-6,44E-03	-6,07E-03	1,52E-04	1,14E+08	7,11E+04
RD_PVO	-8,33E-02	-6,34E-02	6,11E-03	1,29E+08	1,64E+06
RD_VOT	-1,06E-01	-9,10E-02	8,10E-03	1,23E+08	2,81E+05
RP_AND	-7,04E-02	-5,03E-02	8,93E-03	1,32E+08	1,36E+06
RP_OR	-6,66E-03	-6,06E-03	1,62E-04	1,13E+08	1,52E+06
RP_PVO	-7,69E-02	-6,04E-02	6,02E-03	1,29E+08	1,79E+06
RP_VOT	-1,14E-01	-9,42E-02	9,33E-03	1,32E+08	1,33E+06
RPD_AND	-8,20E-02	-4,91E-02	9,59E-03	1,31E+08	1,54E+06
RPD_OR	-6,61E-03	-6,07E-03	1,56E-04	1,14E+08	1,14E+05
RPD_PVO	-7,94E-02	-5,93E-02	7,65E-03	1,24E+08	4,11E+06
RPD_VOT	-1,14E-01	-9,28E-02	1,01E-02	1,31E+08	2,02E+05
CF_AND	-6,85E-02	-4,75E-02	8,87E-03	9,54E+07	2,77E+06
CF_OR	-6,67E-03	-6,08E-03	1,86E-04	8,88E+07	5,40E+05
CF_PVO	-7,25E-02	-5,76E-02	5,53E-03	9,80E+07	1,19E+06
CF_VOT	-1,11E-01	-8,85E-02	8,68E-03	9,91E+07	1,16E+06
GR_AND	-6,67E-02	-4,81E-02	8,09E-03	2,15E+08	3,82E+05
GR_OR	-2,50E-02	-2,08E-02	1,43E-03	2,01E+08	8,79E+04
GR_PVO	-1,09E-01	-8,59E-02	1,08E-02	1,98E+08	7,66E+04
GR_VOT	-1,11E-01	-9,18E-02	9,90E-03	2,16E+08	4,75E+05
TABU_AND	-7,04E-02	-4,77E-02	8,27E-03	1,25E+08	1,15E+05
TABU_OR	-6,72E-03	-6,08E-03	1,78E-04	1,08E+08	2,75E+06
TABU_PVO	-7,94E-02	-5,89E-02	6,19E-03	1,24E+08	1,59E+06
TABU_VOT	-1,06E-01	-9,22E-02	8,75E-03	1,24E+08	1,47E+06

Table C.16. Results of performance evaluations of hyperheuristic patterns on yue20013

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-2,50E-01	-1,64E-01	3,49E-02	1,43E+08	8,21E+05
SR_OR	-1,67E-01	-1,41E-01	1,01E-02	1,24E+08	9,68E+05
SR_PVO	-2,50E-01	-2,50E-01	0,00E+00	1,40E+08	5,30E+06
SR_VOT	-2,50E-01	-2,39E-01	1,71E-02	1,43E+08	5,18E+05
RD_AND	-1,52E-01	-8,33E-02	3,12E-02	1,26E+08	6,31E+06
RD_OR	-1,85E-01	-1,41E-01	1,20E-02	1,25E+08	1,42E+06
RD_PVO	-2,50E-01	-2,23E-01	1,29E-02	1,44E+08	2,14E+05
RD_VOT	-1,32E-01	-1,04E-01	1,17E-02	1,39E+08	2,11E+06
RP_AND	-2,50E-01	-1,63E-01	3,46E-02	1,44E+08	1,97E+06
RP_OR	-1,61E-01	-1,39E-01	9,81E-03	1,25E+08	1,38E+06
RP_PVO	-2,50E-01	-2,50E-01	0,00E+00	1,45E+08	1,07E+05
RP_VOT	-2,50E-01	-2,37E-01	1,80E-02	1,45E+08	1,35E+06
RPD_AND	-2,50E-01	-1,57E-01	3,89E-02	1,45E+08	1,00E+06
RPD_OR	-1,92E-01	-1,40E-01	1,44E-02	1,24E+08	9,84E+05
RPD_PVO	-2,50E-01	-2,50E-01	0,00E+00	1,42E+08	4,37E+06
RPD_VOT	-2,50E-01	-2,40E-01	1,59E-02	1,44E+08	4,45E+05
CF_AND	-2,27E-01	-1,61E-01	3,02E-02	1,06E+08	1,26E+06
CF_OR	-1,72E-01	-1,36E-01	1,10E-02	8,41E+07	7,75E+06
CF_PVO	-2,50E-01	-2,50E-01	0,00E+00	9,87E+07	2,50E+06
CF_VOT	-2,50E-01	-2,41E-01	1,41E-02	1,06E+08	2,51E+05
GR_AND	-2,50E-01	-1,66E-01	3,52E-02	2,49E+08	3,95E+06
GR_OR	-2,50E-01	-2,32E-01	1,05E-02	2,33E+08	1,30E+05
GR_PVO	-2,50E-01	-2,50E-01	0,00E+00	2,39E+08	1,00E+07
GR_VOT	-2,50E-01	-2,46E-01	8,62E-03	2,51E+08	1,52E+06
TABU_AND	-2,27E-01	-1,71E-01	2,56E-02	1,33E+08	4,40E+06
TABU_OR	-1,67E-01	-1,38E-01	1,03E-02	1,17E+08	1,53E+05
TABU_PVO	-2,50E-01	-2,50E-01	0,00E+00	1,35E+08	1,21E+06
TABU_VOT	-2,50E-01	-2,42E-01	1,47E-02	1,36E+08	1,23E+06

Table C.17. Results of performance evaluations of hyperheuristic patterns on yue20021

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-2,76E-02	-1,85E-02	4,01E-03	1,32E+08	1,59E+06
SR_OR	-3,69E-03	-3,25E-03	1,33E-04	1,12E+08	4,78E+04
SR_PVO	-4,03E-02	-3,18E-02	3,76E-03	1,31E+08	5,41E+04
SR_VOT	-5,38E-02	-4,00E-02	6,28E-03	1,33E+08	3,62E+05
RD_AND	-2,82E-02	-1,78E-02	4,23E-03	1,27E+08	7,97E+05
RD_OR	-3,62E-03	-3,24E-03	1,48E-04	1,14E+08	5,02E+04
RD_PVO	-3,50E-02	-2,84E-02	3,09E-03	1,33E+08	5,45E+04
RD_VOT	-4,67E-02	-3,30E-02	4,92E-03	1,28E+08	8,41E+05
RP_AND	-2,50E-02	-1,83E-02	3,12E-03	1,34E+08	1,28E+06
RP_OR	-3,54E-03	-3,20E-03	1,35E-04	1,14E+08	5,07E+04
RP_PVO	-4,03E-02	-3,24E-02	3,65E-03	1,31E+08	1,74E+06
RP_VOT	-4,95E-02	-3,99E-02	5,13E-03	1,35E+08	3,46E+05
RPD_AND	-2,84E-02	-1,82E-02	4,02E-03	1,34E+08	3,53E+05
RPD_OR	-3,74E-03	-3,34E-03	1,28E-04	1,13E+08	1,20E+06
RPD_PVO	-4,00E-02	-3,12E-02	3,12E-03	1,26E+08	4,73E+06
RPD_VOT	-4,85E-02	-3,91E-02	4,80E-03	1,35E+08	3,03E+05
CF_AND	-2,94E-02	-1,83E-02	3,90E-03	9,80E+07	3,30E+06
CF_OR	-3,85E-03	-3,38E-03	1,42E-04	8,81E+07	9,05E+04
CF_PVO	-3,97E-02	-3,08E-02	3,32E-03	9,97E+07	5,14E+04
CF_VOT	-4,95E-02	-3,70E-02	5,20E-03	1,01E+08	1,85E+05
GR_AND	-3,03E-02	-1,74E-02	3,55E-03	2,22E+08	1,11E+06
GR_OR	-1,47E-02	-1,22E-02	1,01E-03	2,03E+08	3,38E+06
GR_PVO	-4,90E-02	-3,84E-02	5,57E-03	2,14E+08	6,62E+06
GR_VOT	-5,68E-02	-3,97E-02	7,34E-03	2,23E+08	7,89E+05
TABU_AND	-3,13E-02	-1,80E-02	4,71E-03	1,23E+08	3,71E+06
TABU_OR	-3,75E-03	-3,39E-03	1,32E-04	1,09E+08	8,76E+05
TABU_PVO	-4,24E-02	-3,20E-02	4,29E-03	1,26E+08	1,21E+06
TABU_VOT	-6,10E-02	-4,07E-02	6,02E-03	1,27E+08	1,13E+06

Table C.18. Results of performance evaluations of hyperheuristic patterns on yue20022

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-1,00E-02	-8,22E-03	1,03E-03	1,30E+08	4,01E+05
SR_OR	-2,51E-03	-2,38E-03	6,64E-05	1,11E+08	1,38E+06
SR_PVO	-1,38E-02	-1,19E-02	8,58E-04	1,28E+08	2,58E+06
SR_VOT	-1,40E-02	-1,17E-02	1,04E-03	1,28E+08	1,68E+06
RD_AND	-1,01E-02	-8,08E-03	9,35E-04	1,23E+08	9,98E+05
RD_OR	-2,58E-03	-2,34E-03	7,65E-05	1,13E+08	4,27E+04
RD_PVO	-1,29E-02	-1,08E-02	9,94E-04	1,30E+08	2,13E+06
RD_VOT	-1,35E-02	-1,09E-02	1,17E-03	1,25E+08	1,05E+06
RP_AND	-9,96E-03	-8,17E-03	8,86E-04	1,31E+08	1,36E+06
RP_OR	-2,54E-03	-2,37E-03	8,36E-05	1,12E+08	7,96E+05
RP_PVO	-1,42E-02	-1,18E-02	9,19E-04	1,31E+08	8,44E+04
RP_VOT	-1,36E-02	-1,16E-02	9,66E-04	1,31E+08	1,75E+06
RPD_AND	-1,04E-02	-8,27E-03	9,44E-04	1,30E+08	1,67E+06
RPD_OR	-2,70E-03	-2,48E-03	8,81E-05	1,13E+08	4,94E+04
RPD_PVO	-1,46E-02	-1,19E-02	9,10E-04	1,28E+08	3,72E+06
RPD_VOT	-1,38E-02	-1,19E-02	9,01E-04	1,30E+08	4,41E+05
CF_AND	-1,00E-02	-8,23E-03	9,30E-04	9,54E+07	3,24E+06
CF_OR	-2,90E-03	-2,54E-03	9,22E-05	8,57E+07	1,16E+06
CF_PVO	-1,40E-02	-1,18E-02	9,94E-04	9,72E+07	4,15E+04
CF_VOT	-1,45E-02	-1,16E-02	9,20E-04	9,81E+07	1,55E+06
GR_AND	-1,03E-02	-8,15E-03	8,79E-04	2,12E+08	1,32E+06
GR_OR	-7,41E-03	-6,72E-03	2,81E-04	1,98E+08	1,37E+05
GR_PVO	-1,63E-02	-1,31E-02	1,11E-03	2,01E+08	6,54E+06
GR_VOT	-1,49E-02	-1,20E-02	1,06E-03	2,13E+08	1,25E+06
TABU_AND	-1,10E-02	-8,37E-03	1,15E-03	1,24E+08	4,53E+05
TABU_OR	-2,75E-03	-2,52E-03	7,91E-05	1,07E+08	2,15E+06
TABU_PVO	-1,47E-02	-1,17E-02	1,06E-03	1,24E+08	4,32E+04
TABU_VOT	-1,43E-02	-1,17E-02	9,34E-04	1,24E+08	3,20E+05

Table C.19. Results of performance evaluations of hyperheuristic patterns on yue20023

	Best Fit.	**Avg. Best Fit.**		**Avg. Num. of Eval.**	
SR_AND	-1,40E-02	-1,23E-02	7,45E-04	1,38E+08	1,31E+06
SR_OR	-1,37E-02	-1,33E-02	1,84E-04	1,21E+08	9,60E+04
SR_PVO	-1,57E-02	-1,51E-02	2,91E-04	1,32E+08	3,17E+06
SR_VOT	-1,56E-02	-1,41E-02	5,63E-04	1,38E+08	1,08E+06
RD_AND	-1,30E-02	-1,16E-02	7,62E-04	1,30E+08	4,49E+06
RD_OR	-1,38E-02	-1,34E-02	1,87E-04	1,23E+08	1,74E+06
RD_PVO	-1,50E-02	-1,44E-02	2,43E-04	1,41E+08	8,37E+04
RD_VOT	-1,37E-02	-1,29E-02	3,27E-04	1,33E+08	3,25E+06
RP_AND	-1,40E-02	-1,23E-02	7,20E-04	1,41E+08	1,14E+06
RP_OR	-1,40E-02	-1,34E-02	1,82E-04	1,23E+08	7,95E+05
RP_PVO	-1,57E-02	-1,51E-02	2,99E-04	1,41E+08	1,28E+06
RP_VOT	-1,53E-02	-1,41E-02	5,38E-04	1,37E+08	4,51E+06
RPD_AND	-1,42E-02	-1,25E-02	8,20E-04	1,40E+08	1,30E+06
RPD_OR	-1,40E-02	-1,34E-02	2,33E-04	1,21E+08	1,44E+06
RPD_PVO	-1,55E-02	-1,50E-02	2,57E-04	1,40E+08	1,62E+06
RPD_VOT	-1,56E-02	-1,41E-02	4,77E-04	1,40E+08	1,45E+06
CF_AND	-1,45E-02	-1,26E-02	8,45E-04	1,04E+08	9,49E+05
CF_OR	-1,39E-02	-1,33E-02	1,85E-04	8,21E+07	7,88E+06
CF_PVO	-1,57E-02	-1,51E-02	3,23E-04	1,04E+08	5,32E+04
CF_VOT	-1,54E-02	-1,41E-02	5,22E-04	1,04E+08	6,80E+05
GR_AND	-1,39E-02	-1,24E-02	7,99E-04	2,36E+08	4,74E+06
GR_OR	-1,53E-02	-1,50E-02	1,60E-04	2,25E+08	2,28E+05
GR_PVO	-1,57E-02	-1,55E-02	1,34E-04	2,29E+08	9,79E+06
GR_VOT	-1,55E-02	-1,46E-02	3,20E-04	2,37E+08	4,93E+06
TABU_AND	-1,40E-02	-1,25E-02	6,92E-04	1,32E+08	2,33E+06
TABU_OR	-1,37E-02	-1,34E-02	1,45E-04	1,15E+08	2,08E+06
TABU_PVO	-1,57E-02	-1,52E-02	2,96E-04	1,32E+08	1,56E+06
TABU_VOT	-1,52E-02	-1,41E-02	5,25E-04	1,32E+08	1,91E+06

Table C.20. Results of performance evaluations of hyperheuristic patterns on yue20031

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-1,29E-02	-8,49E-03	1,55E-03	1,28E+08	1,24E+08
SR_OR	-3,08E-03	-2,62E-03	1,18E-04	1,14E+08	1,10E+08
SR_PVO	-1,80E-02	-1,50E-02	1,46E-03	1,29E+08	1,21E+08
SR_VOT	-2,01E-02	-1,54E-02	2,34E-03	1,29E+08	1,23E+08
RD_AND	-1,08E-02	-8,25E-03	1,37E-03	1,22E+08	1,20E+08
RD_OR	-2,83E-03	-2,56E-03	8,92E-05	1,15E+08	1,12E+08
RD_PVO	-1,75E-02	-1,35E-02	1,30E-03	1,32E+08	1,27E+08
RD_VOT	-1,81E-02	-1,34E-02	1,70E-03	1,21E+08	1,20E+08
RP_AND	-1,18E-02	-8,26E-03	1,56E-03	1,30E+08	1,25E+08
RP_OR	-3,20E-03	-2,60E-03	1,17E-04	1,15E+08	1,12E+08
RP_PVO	-1,90E-02	-1,51E-02	1,46E-03	1,32E+08	1,28E+08
RP_VOT	-2,05E-02	-1,51E-02	1,93E-03	1,31E+08	1,21E+08
RPD_AND	-1,23E-02	-8,23E-03	1,71E-03	1,30E+08	1,24E+08
RPD_OR	-3,13E-03	-2,70E-03	1,02E-04	1,14E+08	1,11E+08
RPD_PVO	-1,77E-02	-1,50E-02	1,37E-03	1,31E+08	1,27E+08
RPD_VOT	-2,11E-02	-1,56E-02	2,54E-03	1,30E+08	1,25E+08
CF_AND	-1,16E-02	-8,12E-03	1,67E-03	9,30E+07	9,25E+07
CF_OR	-2,90E-03	-2,77E-03	8,36E-05	8,74E+07	8,66E+07
CF_PVO	-1,82E-02	-1,45E-02	1,43E-03	9,86E+07	9,72E+07
CF_VOT	-2,01E-02	-1,49E-02	1,87E-03	9,87E+07	9,52E+07
GR_AND	-1,35E-02	-8,61E-03	1,61E-03	2,12E+08	1,97E+08
GR_OR	-7,62E-03	-7,02E-03	2,37E-04	1,97E+08	1,87E+08
GR_PVO	-2,02E-02	-1,66E-02	1,99E-03	1,99E+08	1,85E+08
GR_VOT	-2,14E-02	-1,62E-02	2,27E-03	2,13E+08	1,99E+08
TABU_AND	-1,20E-02	-8,13E-03	1,73E-03	1,24E+08	1,19E+08
TABU_OR	-2,94E-03	-2,73E-03	8,73E-05	1,09E+08	1,07E+08
TABU_PVO	-1,81E-02	-1,44E-02	1,48E-03	1,25E+08	1,22E+08
TABU_VOT	-2,12E-02	-1,52E-02	2,08E-03	1,24E+08	1,19E+08

Table C.21. Results of performance evaluations of hyperheuristic patterns on yue20032

	Best Fit.	Avg. Best Fit.		Avg. Num. of Eval.	
SR_AND	-3,98E-03	-3,31E-03	2,83E-04	1,24E+08	7,51E+05
SR_OR	-1,82E-03	-1,70E-03	4,24E-05	1,10E+08	1,41E+06
SR_PVO	-6,00E-03	-4,92E-03	4,28E-04	1,21E+08	4,05E+06
SR_VOT	-5,47E-03	-4,55E-03	4,36E-04	1,23E+08	1,69E+06
RD_AND	-3,56E-03	-3,11E-03	2,11E-04	1,20E+08	2,00E+06
RD_OR	-1,94E-03	-1,69E-03	5,47E-05	1,12E+08	4,40E+04
RD_PVO	-5,33E-03	-4,51E-03	3,66E-04	1,27E+08	1,69E+06
RD_VOT	-4,95E-03	-4,20E-03	3,60E-04	1,20E+08	1,67E+06
RP_AND	-3,98E-03	-3,32E-03	2,85E-04	1,25E+08	8,15E+05
RP_OR	-1,83E-03	-1,70E-03	4,32E-05	1,12E+08	5,86E+04
RP_PVO	-5,73E-03	-4,95E-03	3,00E-04	1,28E+08	1,69E+05
RP_VOT	-5,68E-03	-4,44E-03	4,45E-04	1,21E+08	5,12E+06
RPD_AND	-3,80E-03	-3,27E-03	2,61E-04	1,24E+08	1,73E+06
RPD_OR	-1,92E-03	-1,79E-03	4,64E-05	1,11E+08	9,16E+04
RPD_PVO	-6,15E-03	-5,07E-03	4,62E-04	1,27E+08	2,09E+06
RPD_VOT	-5,11E-03	-4,47E-03	3,09E-04	1,25E+08	7,78E+05
CF_AND	-3,95E-03	-3,29E-03	2,97E-04	9,25E+07	3,25E+06
CF_OR	-2,10E-03	-1,85E-03	5,89E-05	8,66E+07	9,50E+04
CF_PVO	-6,08E-03	-4,83E-03	4,32E-04	9,72E+07	4,61E+04
CF_VOT	-5,61E-03	-4,44E-03	3,77E-04	9,52E+07	1,27E+06
GR_AND	-3,90E-03	-3,22E-03	2,90E-04	1,97E+08	1,94E+06
GR_OR	-4,08E-03	-3,73E-03	1,05E-04	1,87E+08	2,15E+05
GR_PVO	-6,13E-03	-4,92E-03	4,14E-04	1,85E+08	1,86E+06
GR_VOT	-5,79E-03	-4,67E-03	4,26E-04	1,99E+08	1,92E+06
TABU_AND	-4,13E-03	-3,25E-03	3,36E-04	1,19E+08	1,03E+06
TABU_OR	-1,96E-03	-1,81E-03	4,76E-05	1,07E+08	3,13E+06
TABU_PVO	-6,59E-03	-4,93E-03	4,58E-04	1,22E+08	4,09E+04
TABU_VOT	-5,69E-03	-4,49E-03	4,24E-04	1,19E+08	5,29E+05

REFERENCES

[1] Aamodt, A. and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches." AI Communications 7(1):39-52, 1994.

[2] Abdullah, S., S. Ahmadi, E. K. Burke and M. Dror, "Applying Ahuja-Orlin's Large Neighbourhood for Constructing Examination Timetabling Solution." In *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT'04)*, E. K. Burke and M. Trick (eds.), 413–419, 2004.

[3] Ackley, D. H., "An Empirical Study of Bit Vector Function Optimization." L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, 170–204, Pitman Publishing, London, 1987.

[4] Alkan, A. and E. Ozcan, "Memetic Algorithms for Timetabling." In *Proceedings of IEEE Congress on Evolutionary Computation*, 1796–1802, 2003.

[5] Arani, T. and V. Lotfi, "A Three Phased Approach to Final Exam Scheduling." *IEEE Trans.*, vol 21, iss. 1, 86–96, 1989.

[6] Araya, I., B. Neveu and M-C. Riff, "An efficient hyperheuristic for strip packing problems." Adaptive and Multilevel Metaheuristics, Studies on Computational Intelligence, Springer, 2008.

[7] Asmuni, H., E. K. Burke and J. Garibaldi, "Fuzzy Multiple Ordering Criteria for Examination Timetabling." In *Proceedings of the 5th international conference on the Practice and Theory of Automated Timetabling (PATAT)*, 51-66, Pittsburgh, USA, August 18th-20th, 2004.

[8] Asmuni H, Burke E. K., Garibaldi J M and B. McCollum, "A Novel Fuzzy Approach to Evaluate the Quality of Examination Timetabling." *The 6th International Conference on*

the Practice and Theory of Automated Timetabling (PATAT'06), 2006.

[9] Ayob, M. and G. Kendall, “A Monte Carlo Hyper-Heuristic To Optimise Component Placement Sequencing For Multi Head Placement Machine.” In *Proceedings of the International Conference on Intelligent Technologies (InTech'03)*, December 17-19, Chiang Mai, Thailand, 132–141, 2003.

[10] Bader-El-Den, M. and R. Poli, "Generating SAT Local-Search Heuristics using a GP Hyper-Heuristic Framework." *8th International Conference, Evolution Artificielle*, 2007.

[11] Bai, R. and G. Kendall, “An Investigation of Automated Planograms Using a Simulated Annealing Based Hyper-heuristics.” *The 5th Metaheuristics International Conference (MIC 2003)*, 23-25 August 2003, Kyoto International Conference Hall, Kyoto, Japan, 2003.

[12] Ben-Ameur, W., "Computing the Initial Temperature of Simulated Annealing." Computational Optimization and Applications. Vol. 29, 369-385, 2004.

[13] Bilgin, B., E. Ozcan, E. E. Korkmaz, “An Experimental Study on Hyperheuristics and Exam Scheduling.” In *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT'06)*, 123–140, 2006.

[14] Booker, L., D. E. Goldberg and J. H. Holland, “Classifier Systems and Genetic Algorithms.” Artificial Intelligence, 40(1-3):235–282, 1989.

[15] Broder, S., “Final examination scheduling.” Communications of the *ACM* 7(8), 494–498, 1964.

[16] Bull, L., “Learning Classifier Systems: A Brief Introduction.” Technical Report, Faculty of Computing Engineering & Mathematical Sciences, University of the West of England, U.K, 2003.

[17] Burke, E. K. and Y. Bykov, “Solving Exam Timetabling Problems with the Flex-

Deluge Algorithm.” In *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT'06)*, Brno, August 2006, 351–358, 2006.

[18] Burke, E. K., M. Dror, S. Petrovic and R. Qu, “Hybrid Graph Heuristics within a Hyper-heuristic Approach to Exam Timetabling Problems.” The Next Wave in Computing, Optimization, and Decision Technologies. B.L. Golden, S. Raghavan and E. A. Wasil (eds.), Conference Volume of *the 9th informs Computing Society Conference*. Springer, Jan, 79–91, 2005.

[19] Burke, E. K., D. Elliman, P. Ford and B. Weare, “Examination Timetabling in British Universities - A Survey.” LNCS, Springer-Verlag, Vol. 1153, 76–90, 1996.

[20] Burke E. K., E. Hart, G. Kendall, J. Newall, P. Ross, S. Schulenburg, Hyper-heuristics: an Emerging Direction in Modern Search Technology, In: Glover F.W., Kochenberger G.A. (eds.), Handbook of Metaheuristics, Kluwer Academic Publishers, 2003.

[21] Burke, E. K., M. R. Hyde and G. Kendall, “Evolving Bin Packing Heuristics with Genetic Programming.” *Parallel Problem Solving from Nature - PPSN IX*, LNCS, Vol. 4193, 860–869, Springer-Verlag, 2006.

[22] Burke, E. K., M. R. Hyde, G. Kendall and J. R. Woodward, "Automatic heuristic generation with genetic programming: Evolving a jack-of-alltrades or a master of one." In Lipson, H., Thierens, D., eds.: *Genetic and Evolutionary Computation Conference, GECCO 2007*, London, UK, July 7-11, 2007.

[23] Burke, E. K., G. Kendall, D. L. Silva, R. O'Brien and E. Soubeiga, “An Ant Algorithm Hyperheuristic for the Project Presentation Scheduling Problem.” In Proc. of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, September 2-5, 2263–2270, 2005.

[24] Burke, E. K., G. Kendall and E. Soubeiga, “A Tabu-Search Hyper-Heuristic for Timetabling and Rostering.” *Journal of Heuristics*, 9(6), 451–470, 2003.

[25] Burke, E. K., B. MacCarthy, S. Petrovic and R. Qu, “Knowledge Discovery in a Hyperheuristic Using Case-Based Reasoning on Course Timetabling.” In Proc. of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT'02), Gent, Belgium, Springer LNCS Volume 2740, August 21–23, 276–286, 2002.

[26] Burke, E. K., B. McCollum, A. Meisels, S. Petrovic and R. Qu, “A Graph-Based Hyper Heuristic for Educational Timetabling Problems.” *European Journal of Operational Research*, 176: 177-192, 2007.

[27] Burke, E. K. and J. P. Newall, “A multi-stage evolutionary algorithm for the timetable problem.” *IEEE Transactions on Evolutionary Computation*, 3(1):63–74, 1999.

[28] Burke, E. K., J. P. Newall and R. F. Weare, “A Memetic Algorithm for University Exam Timetabling.” *1st International Conference on the Practice and Theory of Automated Timetabling (PATAT'95)*, Edinburgh, 30 August - 1 September 1995, 241–250, 1995.

[29] Burke, E. K. and J. P. Newall, “Solving Examination Timetabling Problems through Adaption of Heuristic Orderings: Models and Algorithms for Planning and Scheduling Problems.” *Annals of Operations Research*, Vol. 129, 107–134, 2004.

[30] Burke, E. K. and S. Petrovic, “Recent Research Directions in Automated Timetabling.” *European Journal of Operational Research*, 140(2), 266–280, 2002.

[31] Burke, E. K., S. Petrovic and R. Qu, “Case Based Heuristic Selection for Timetabling Problems.” *Journal of Scheduling*, Vol. 9, No. 2, 115–132, 2006.

[32] Burke E. K., J. D. L. Silva and E. Soubeiga, "Hyperheuristic Approaches for Multiobjective Optimisation." In: *Proceedings of the 5th Metaheuristics International Conference (MIC 2003)*, Kyoto Japan, pp. 11.1-11.6, 2003.

[33] Burke, E. K., D. L. Silva and E. Soubeiga, “Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling.” *Meta-heuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nonobe, M. Yagiura (eds.), Springer, 2005.

[34] Carter, M. W., “A Survey of Practical Applications of Examination Timetabling Algorithms.” *Operations Research* Vol 34, 193–202, 1986.

[35] Carter, M. W., G. Laporte and S. T. Lee, “Examination timetabling: algorithmic strategies and applications.” *Journal of the Operational Research Society*, 47:373–383, 1996.

[36] Carter, M. W., G. Laporte and S. Y. Lee, “Recent Developments in Practical Exam Timetabling”. Selected papers from the *1st International Conference PATAT’95*. LNCS 1153, 3-21, 1996.

[37] Carter, M. W., G. Laporte and S. Y. Lee, “Recent Developments in Practical Course Timetabling”. Selected papers from the *2nd International Conference PATAT’97*. LNCS 1408, 3-19, 1998.

[38] Cheong, C. Y., K. C. Tan and B. Veeravalli, “Solving the Exam Timetabling Problem via a Multi-Objective Evolutionary Algorithm – A More General Approach”, *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched),* 2007.

[39] Cole, A. J., “The preparation of examination timetables using a small-store computer.” *Computer Journal* 7, 117–121, 1964.

[40] Cowling, P. and K. Chakhlevitch, “Hyperheuristics for Managing a Large Collection of Low Level Heuristics to Schedule Personnel.” In Proc. of the 2003 IEEE Congress on Evolutionary Computation (CEC'2003), Canberra, Australia, 8-12 December, 1214–1221, 2003.

[41] Cowling, P., G. Kendall and L. Han, “An Investigation of a Hyperheuristic Genetic

Algorithm Applied to a Trainer Scheduling Problem." *In Congress on Evolutionary Computation, CEC'02*, May 12–17, 1185–1190, 2002.

[42] Cowling, P., G. Kendall and L. Han, "An Adaptive Length Chromosome Hyperheuristic Genetic Algorithm for a Trainer Scheduling Problem." In Proc. of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL'02), November 18-22, Orchid Country Club, Singapore, 267–271, 2002.

[43] Cowling, P., G. Kendall and E. Soubeiga, "A Hyper-heuristic Approach to Scheduling a Sales Summit." LNCS 2079, *the 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT'00)*, E. K. Burke and W. Erben (eds.), Konstanz, Germany, August 16-18, 176–190, 2000.

[44] Cowling, P., G. Kendall and E. Soubeiga, "Adaptively Parameterised Hyperheuristics for Sales Summit Scheduling." 4th Metahuristics International Conference (MIC 2001), 2001.

[45] Cowling, P., G. Kendall, and E. Soubeiga, "Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation." In *Second European Conference on Evolutionary Computing for Combinatorial Optimisation, EvoCop 2002*, Lecture Notes in Computer Science, pages 1–10, Kinsale, Ireland, April, 3-51, Springer, 2001.

[46] Crowston, W. B., F. Glover, G. L. Thompson, and J. D. Trawick, "Probabilistic and parametric learning combinations of local job shop scheduling rules." *ONR Research memorandum*, GSIA, Carnegie Mellon University, Pittsburgh, (117), 1963.

[47] Cuesta-Canada, A., L. Garrido and H. Terashima-Marin, "Building Hyper-heuristics through Ant Colony Optimization for the 2D Bin Packing Problem." KES'05, LNCS 3684, 654–660, 2005.

[48] Dawkins, R., "*The Selfish Genes*." Oxford University Press, 1976.

[49] De Jong, K. A., "An analysis of the behavior of a class of genetic adaptive systems."

PhD thesis, University of Michigan, 1975.

[50] De Werra, D., “An introduction to timetabling.” European Journal of Operational Research, 19:151–162, 1985.

[51] Di Gaspero, L. and A. Schaerf., “Tabu Search Techniques for Examination Timetabling.” LNCS, *the 3rd International Conference on Practice and Theory of Automated Timetabling (PATAT'00)*, Konstanz, 16-18 August 2000, 104–117, 2000.

[52] Di Gaspero, L. and A. Schaerf, “Multi-neighbourhood local search with application to course timetabling.” In Edmund Burke and Patrick De Causmaecker, editors, *The Practice and Theory of Automated Timetabling IV*: Selected Papers from the 4th International Conference, Springer Lecture Notes in Computer Science Vol 2740, pp 262-275, 2002.

[53] Dorigo, M., Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Italy, 1992.

[54] Dowsland, K., E. Soubeiga and E. K. Burke, “A simulated annealing hyper-heuristic for determining shipper sizes for storage and transportation.” European Journal of Operational Research, 2005.

[55] Easom, E. E., “A Survey of Global Optimization Techniques.” M. Eng. thesis, University of Louisville, 1990.

[56] Eiben, A. E. and J. E. Smith, Introduction to Evolutionary Computing. Springer, Natural Computing Series, 2003.

[57] Eley, M., “Ant Algorithms for the Exam Timetabling Problem.” *The 6th International Conference on Practice and Theory of Automated Timetabling (PATAT'06)*, 2006.

[58] Eliaz, K., D. Ray and R. Razin, “Group decision-making in the shadow of disagreement.” Journal of Economic Theory 132, 236–273, 2007.

[59] Ersoy, E, E. Ozcan and S. Uyar, “Memetic Algorithms and Hyperhill-climbers.” In *Proceedings of the 3rd Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2007)*, 2007.

[60] Even, S., A. Itai and A. Shamir, “On the Complexity of Timetable and Multicommodity Flow Problems.” *SIAM J. Computing*, 5(4), 691–703, 1976.

[61] Fang, H.-L., P. M. Ross and D. Corne, “A Promising Hybrid GA/Heuristic Approach for Open-Shop Scheduling Problems.” In Proc. of ECAI 94: 11th European Conference on Artificial Intelligence, A. Cohn (ed), pages 590–594, John Wiley and Sons Ltd., 1994.

[62] Fisher, H. and G. L. Thompson, “Probabilistic learning combinations of local job-shop scheduling rules.” *In Factory Scheduling Conference, Carnegie Institute of Technology*, May 10-12, 1961.

[63] Fisher, H. and G. L. Thompson, “Probabilistic learning combinations of local job-shop scheduling rules.” Industrial Scheduling (New Jersey) (J. F. Muth and G. L. Thompson, eds.), Prentice-Hall, Inc, pp. 225–251, 1963.

[64] Foxley, E. and K. Lockyer, “The construction of examination timetables by computer.” *The Computer Journal*, 1968.

[65] Ganapathy, V., S. Marimuthu and S. G. Ponnambalam, “Tabu Search and Simulated Annealing Algorithms for Lot Streaming in Two-Machine Flow-shop.” *IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[66] Garey, M. R. and D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness.”, W. H. Freeman, New York, 1979.

[67] Garrido, P. and M-C. Riff, "An Evolutionary Hyperheuristic to Solve Strip-Packing Problems", LNCS, Springer-Verlag, 4881, p: 406-415, 2007.

[68] Gaw, A., P. Rattadilok and R. S. K. Kwan, “Distributed Choice Function Hyper-

Heuristics for Timetabling and Scheduling.” In Proc. of the 2004 International Conference on the Practice and Theory of Automated Timetabling (PATAT'04), Pittsburgh USA, 495–497, 2004.

[69] Gilmore, P. C. and R. E. Gomory, "A linear programming approach to the cutting-stock problem-Part II." Operations Research 11, 864–888, 1963.

[70] Gilmore, P. C. and R. E. Gomory, "Multistage cutting stock problems of two and more dimensions." Operations Research 13, 94–120, 1965.

[71] Glover, F., "Tabu Search - Part I." ORSA Journal on Computing 1, 190-206, 1989.

[72] Glover, F., "Tabu Search - Part II." ORSA Journal on Computing 2, 4-32, 1990.

[73] Goldberg, D. E., “Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction.” *Complex Systems*, 129–152, 1989.

[74] Goldberg, D. E., “Genetic Algorithms and Walsh Functions: Part II, Deception and Its Analysis.” *Complex Systems*, 153–171, 1989.

[75] Gratch, J., S. Chein and G.de Jong. “Learning search control knowledge for deep space network scheduling.” *In Proceedings of the Tenth International Conference on Machine Learning*, pp 135-142, 1993.

[76] Griewangk, A. O., “Generalized Descent of Global Optimization.” *Journal of Optimization Theory and Applications*, Vol. 34, 11–39, 1981.

[77] Han, L. and G. Kendall, “Investigation of a Tabu Assisted Hyper-Heuristic Genetic Algorithm.” In Proc. of Congress on Evolutionary Computation (CEC2003), 2230–2237 (Vol. 3), 2003.

[78] Han, L. and G. Kendall, “Guided Operators for a Hyper-Heuristic Genetic Algorithm.” In Proc. of AI-2003: Advances in Artificial Intelligence. *The 16th Australian*

Conference on Artificial Intelligence (AI'03), Tamás D Gedeon and Lance Chun Che Fung (eds.), Perth, Australia 3-5 Dec 2003, Springer LN in Artificial Intelligence Vol. 2903, 807–820, 2003.

[79] Hansen, P. and N. Mladenovic, "Variable Neighbourhood Search." In: Burke E.K. and Kendall G. (eds.) Search Methodologies: Introductory Tutorials in Optimisation and Decision Support Techniques. Springer, Chapter 8, 211-238, 2005.

[80] Hansen, P. and N. Mladenovic, "Variable Neighbourhood Search: Principles and Applications." European Journal of Operational Research, 130: 449-467, 2001.

[81] Hart, W., P. M. Ross and J. Nelson, "Solving a real-world problem using an evolving heuristically driven schedule builder." *Evolutionary Computing*, Vol. 6 No. 1, 61–80, 1998.

[82] Holland, J. H., *Adaptation in Natural and Artificial Systems*. Univ. Mich. Press, 1975.

[83] Johnson, D. S., C. R. Aragon, L. A. McGeoch and C. Schevon, "Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning." Operations Research. 37, 865-892, 1989.

[84] Johnson, D. S., C. R. Aragon, L. A. McGeoch and C. Schevon, "Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Colouring and Number Partitioning." Operations Research. 39, 378-406, 1991.

[85] Kaelbling, L. P., M. L. Littman and A.W. Moore, "Reinforcement learning: A survey." Journal of Artificial Intelligence Research, 4:237–285, 1996.

[86] Kendall, G. and N. M. Hussin, "Tabu Search Hyper-heuristic Approach to the Examination Timetabling Problem at University of Technology MARA." *The 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT'04),* E. K. Burke and M. Trick (eds.), Springer LNCS, Vol. 3616, 2005, 270–293, 2004.

[87] Kendall, G. and M. Mohamad, “Channel Assignment in Cellular Communication Using a Great Deluge Hyper-heuristic.” In Proc. of the 2004 IEEE International Conference on Network (ICON2004), 2004.

[88] Kendall, G. and M. Mohamad, "Channel Assignment Optimisation Using a Hyper-heuristic." Proceedings of the IEEE Conference on Cybernetic and Intelligent Systems (CIS2004), pp. 790-795, 2004.

[89] Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing." Science, Vol 220, Number 4598, pages 671-680, 1983.

[90] Kira, K. and L. A. Rendell, “The feature selection problem: Traditional methods and a new algorithm.” In: *Proceedings of Ninth National Conference on Artificial Intelligence*, 129-134, 1992.

[91] Koza, J. R., "Genetic Programming: on the Programming of Computers by Means of Natural Selection." The MIT Press, Boston, Massachusetts, 1992.

[92] Krasnogor, N., “Studies on the Theory and Design Space of Memetic Algorithms.” PhD Thesis, University of the West of England, Bristol, UK, 2002.

[93] Krasnogor, N. and J. E. Smith, “Multimeme Algorithms for the Structure Prediction and Structure Comparison of Proteins.” In Proc. of the Bird of a Feather Workshops, GECCO, 42–44, 2002.

[94] Krasnogor, N. and J. E. Smith, “Emergence of Profitable Search strategies Based on a Simple Inheritance Mechanism.” In Proc. of the Genetic and Evolutionary Computation Conference, GECCO, 432–439, 2001.

[95] Krasnogor, N. and J. E. Smith, “A Memetic Algorithm With Self-Adaptive Local Search: TSP as a case study.” In Proc. of the Genetic and Evolutionary Computation Conference, GECCO, 987–994, 2000.

[96] Leake, D. B., "CBR in Context: The Present and Future." AAAI Press/MIT Press, California, 1996.

[97] Leighton, F. T., “A Graph Coloring Algorithm for Large Scheduling Problems.” *Journal of Reasearch of the National Bureau of Standards*, 84:489–506, 1979.

[98] Lotfi, V. and R. Cerveny, “A final-exam-scheduling package.” *The Journal of the Operational Research Society*, Vol.42, No.3.(Mar.), 205–216, 1991.

[99] Marin, H. T., “Combinations of GAs and CSP Strategies for Solving Examination Timetabling Problems.” Ph. D. Thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey, 1998.

[100] Marin, H. T., C., J. F. Zarate, P. Ross and M. valenzuela-Rendon, "Comparing two models to generate hyper-heuristics for the 2d-regular bin-packing problem." Proceedings of the 9th annual conference on Genetic and evolutionary computation, p: 2182 - 2189, 2007.

[101] Merlot, L. T. G., N. Boland, B. D. Hughes and P. J. Stuckey, “A Hybrid Algorithm for the Examination Timetabling Problem.” In Proc. of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT'02), Gent, August 21-23, 348–371, 2002.

[102] Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller. "Equations of State Calculations by Fast Computing Machines." Journal of Chemical Physics, 21(6):1087-1092, 1953.

[103] Mitchell, M. and S. Forrest, “Fitness Landscapes: Royal Road Functions.” *Handbook of Evolutionary Computation*, T. Baeck, D. Fogel and Michalewiz Z. (eds.), Institute of Physics Publishing and Oxford University, Bristol, 1–25, 1997.

[104] Mladenovic, N. and P. Hansen, "Variable neighbourhood search." Computers Ops. Res., 24(11), pp 1097-1100, 1997.

[105] Moscato, P. and M. G. Norman, “A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems.” *Parallel Computing and Transputer Applications*, 177–186, 1992.

[106] Nareyek, A., “Choosing Search Heuristics by Non-Stationary Reinforcement Learning.” *Metaheuristics: Computer Decision Making*, Kluwer Academic Publishers, 523–544, 2004.

[107] Ozcan, E., “Towards an XML based standard for Timetabling Problems: TTML.” *Multidisciplinary Scheduling: Theory and Applications*, Springer Verlag, 163 (24), 2005.

[108] Ozcan, E., B. Bilgin and E. E. Korkmaz, “Hill climbers and mutational heuristics in hyperheuristics.” In Runarsson, T., Beyer, H.G., Burke, E., J.Merelo-Guervos, J., Whitley, D., Yao, X. (eds.): *LNCS*, In Proc. of the 9th International Conference on Parallel Problem Solving from Nature (PPSN'06). Vol. 4193., Reykjavik, Iceland. 202–211, 2006.

[109] Ozcan, E., B. Bilgin and E. E. Korkmaz, “A Comprehensive Analysis of Hyper-heuristics.” *Intelligent Data Analysis*, 12:1, pp. 3-23, 2008.

[110] Ozcan, E. and E. Ersoy, “Final Exam Scheduler – FES.” In Proc. of 2005 IEEE Congress on Evolutionary Computation, Vol. 2, (2005) 1356–1363, 2005.

[111] Paquete, L. F. and C. M. Fonseca, “A Study of Examination Timetabling with Multiobjective Evolutionary Algorithms.” In Proc. of the 4th Metaheuristics International Conference (MIC 2001), Porto, 16-20 July 2001, 149–154, 2001.

[112] Petrovic, S., V. Patel and Y. Yang, “Examination Timetabling with Fuzzy Constraints.” *The 5th International Conference on the Practice and Theory of Automated Timetabling* (PATAT'05), Burke, E. K. and M. Trick, (eds.), LNCS 3616, Springer-Verlag 2005, 313–333, 2005.

[113] Petrovic, S., Y. Yang and M. Dror, "Case-based Initialisation for Examination Time-tabling." In Proc. of 1st Multidisciplinary Intl. Conf. on Scheduling: Theory and Applications (MISTA 2003), Nottingham, UK, Aug 13-16, 137–154, 2003.

[114] Petrovic, S., Y. Yang and M. Dror, "Case-based selection of initialisation heuristics for metaheuristic examination timetabling." *Expert Systems with Applications: An International Journal*. Vol. 33, Iss. 3, October, 772–785, 2007.

[115] Qu, R., E. K. Burke, B. McCollum, L. T. G. Merlot and S. Y. Lee, "A Survey of Search Methodologies and Automated Approaches for Examination Timetabling." Technical Report NOTTCS-TR-2006-4, School of CSiT, University of Nottingham, 2006.

[116] Qu, R., and E. K. Burke, "Adaptive Decomposition and Construction for Examination Timetabling Problems". *Multidisciplinary International Scheduling: Theory and Applications 2007 (MISTA'07)*, 418-425, Paris, France, Aug, 2007.

[117] Qu, R. and E. K. Burke, "Hybrid Variable Neighborhood HyperHeuristics for Exam Timetabling Problems." In: *Proceedings of the sixth metaheuristics international conference, MIC 2005*, Vienna, Austria, 2005.

[118] Qu, R. and G. Ochoa, Lecture Slides from Summer School Course on Hyper-heuristics. Istanbul Technical University, Turkey, 31st July – 3rd August, 2007.

[119] Radcliffe, N. J. and P. D. Surry, "Fitness variance of formae and performance prediction." *Foundations of Genetic Algorithms 3*, L. D. Whitley and M. D. Vose, Ed. San Mateo, CA: Morgan Kaufmann, Vol. 3, 51–72, 1994.

[120] Rastrigin, L. A., "Extremel Control System." *In Theoritical Foundations of Engineering Cyberneics Series*, Moscow, Nauka, Russian, 1974.

[121] Redl, A. T., "University Timetabling via Graph Coloring: An Alternative Approach." Thirty-Eighth Southeastern International Conference on Combinatorics, Graph Theory, and Computing, 2007.

[122] Robbins, S. P. and D. A. DeCenzo, *Fundamentals of Management*. (4th ed.), Prentice Hall, 2003.

[123] Rodriguez J. A. V., S. Petrovic and A. Salhi, "A Combined Meta-Heuristic with Hyper-Heuristic Approach to the Scheduling of the Hybrid Flow Shop with Sequence Dependent Setup Times and Uniform Machines." In *Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications*, (Baptiste P., Kendall G., Munier-Kordon A., and Sourd F., Eds.), pages 506-513. Paris, France, August 2007.

[124] Rodriguez, J. A. V., S. Petrovic and A. Salhi, "An Investigation of Hyper-heuristic Search Spaces." In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3776-3783, 2007.

[125] Rodriguez, J. A. V. and A. Salhi, "A Robust Meta-Hyper-Heuristic Approach to Hybrid Flow Shop Scheduling", Evolutionary Scheduling, Springer, Vol. 49, p:125-142, 2007.

[126] Ross, P., *Hyper-heuristics, Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (eds. E.K.Burke & G.Kendall), pp 529-556, Springer, 2005.

[127] Ross, P, S. Schulenburg, J. G. Marin-Blázquez and E. Hart, “Hyperheuristics: learning to combine simple heuristics in bin-packing problems.” In Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, July 9-13, 942–948, 2002.

[128] Ross, P., S. Schulenburg, J. G. Marin-Blázquez and E. Hart, “Learning a Procedure that Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heuristics.” In Proc. Genetic and Evolutionary Computation (GECCO 2003), Springer LNCS, Vol. 2724, 1295–1306, 2003.

[129] Ross, P., J. G. Marin-Blázquez and E. Hart, “Hyper-heuristics applied to class and exam timetabling problems.” *The Congress on Evolutionary Computation*, Vol. 2, 1691–1698, 2004.

[130] Rossi-Doria, O. and B. Paechter, "An hyperheuristic approach to course timetabling problem using an evolutionary algorithm." *The 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)*, 2003.

[131] Sawaragi, Y., H. Nakayama and T. Tanino, "Theory of Multiobjective Optimization." London: Academic Press, 1985.

[132] Schaerf, A., "A Survey of Automated Timetabling." Artificial Intelligence Review 13 (2) 87-127, 1999.

[133] Schwartz, A. E., “Group decision-making.” *The CPA Journal*, 64, 8: 60–62, 1994.

[134] Schwefel, H. P., *Numerical Optimization of Computer Models*. John Wiley & Sons, New York, 1981.

[135] Sutton, R. S. and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.

[136] Thabtah, F. and P. Cowling, "Mining the data from a hyperheuristic approach using associative classification." Expert Systems with Applications 34, 1093–1101, 2008.

[137] Tounsi, M. and S. Ouis, “An Iterative local-search framework for solving constraint satisfaction problem”, Applied Soft Computing, Elsevier, 2008.

[138] Welsh, D. J. A. and M. B. Powell, “An upper bound to the chromatic number of a graph and its application to timetabling problems.” *The Computer Journal*, 10:85–86, 1967.

[139] Whitley, D., “Fundemental Principles of Deception in Genetic Search.” In G.J.E.

Rawlings (Ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann, San Matco, CA, 1991.

[140] Wilson, S. W., "Classifer Systems Based on Accuracy." Evolutionary Computation. 3(2):149-175, 1995.

[141] Wolpert, D. and W. G. MacReady, "No free lunch theorems for optimization." *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82, 1997.

[142] Wong, T., P. Côté and P. Gely, "Final Exam Timetabling: A Practical Approach." In Proc. of IEEE Canadian Conference on Electrical and Computer Engineering, Winnipeg, 12-15 May 2002, Vol. 2, 726–731, 2002.

[143] Wood, D. C., "A system for computing university examination timetables." *The Computer Journal*, 11(1): 41–47, 1968.

[144] Wood, D. C., "A technique for coloring a graph applicable to large scale time-tabling problems." *The Computer Journal*, 12:317, 1969.

[145] Zitzler, E., "Evolutionary Algorithms for Multiobjective Optimization." Evolutionary Methods for Design, Optimisation and Control, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou and T. Fogarty (Eds.), 2002.