Pamukkale Univ Muh Bilim Derg, XX(X), XX-XX, 20XX

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Population-based local search algorithms forcross-domain search

Alanlar-arası arama için popülasyona dayalı yerel arama algoritmaları

Berna KİRAZ[1], Fatma CORUT ERGİN[2]*

[1]Department of Computer Engineering, Faculty of Engineering, Fatih Sultan Mehmet Vakif University, İstanbul, Turkey.
bkiraz@fsm.edu.tr

[2]Department of Computer Engineering, Faculty of Engineering, Marmara University, İstanbul, Turkey.
fatma.ergin@marmara.edu.tr

Received/Geliş Tarihi: 09.08.2023 Revision/Düzeltme Tarihi: 22.03.2024 doi: 10.5505/pajes.2024.90390
Accepted/Kabul Tarihi: 18.04.2024 Research Article/Araştırma Makalesi

Abstract

Population-based local search is a meta-heuristic algorithm combining the principles of the population-based search and the local search. This study presents an extensive comparison of two population-based local search approaches, specifically, the steady state memetic algorithm (SSMA) and a population-based iterated local search (PILS). To the best of our knowledge, PILS is proposed first for cross-domain search. Both approaches are implemented in Hyper-heuristics Flexible Framework (HyFlex) which contains different operators for different problem domains. The operators used in PILS and SSMA are the ones defined in HyFlex and the operator selection is done using two heuristic selection methods, namely, Simple Random and Reinforcement Learning with Tournament selection. The performance of the proposed methods with the selection methods is assessed over nine problem domains in HyFlex. The results reveal the success of the presented approaches for the cross-domain search.

Keywords: Population-based Local Search, Memetic Algorithms, Hyper-heuristics, Iterated Local Search, Combinatorial Optimization

Öz

Popülasyona dayalı yerel arama, popülasyona dayalı arama ve yerel aramanın ilkelerini birleştiren meta-sezgisel bir algoritmadır. Bu çalışma, iki farklı popülasyona dayalı yerel arama yaklaşımının kapsamlı bir karşılaştırmasını sunmaktadır: kararlı durum memetik algoritma (SSMA) ve popülasyona dayalı iteratif yerel arama (PILS). PILS, bildiğimiz kadarıyla, alanlar arası arama için ilk önerilen yöntemdir. Her iki yaklaşım da farklı problem alanları için farklı operatörler içeren Hyper-heuristics Flexible Framework (HyFlex) üzerinde uygulanmıştır. PILS ve SSMA'da kullanılan operatörler, HyFlex'te tanımlanan operatörlerdir ve bu operatörler arasından seçim yapmak için Basit Rastgele ve Turnuva seçimi ile Pekiştirmeli Öğrenme yöntemleri kullanılmaktadır. Önerilen yöntemlerin her iki seçim yöntemiyle performansı HyFlex' teki dokuz farklı problem üzerinden değerlendirilmiştir. Sonuçlar, alanlar arası arama için sunulan yaklaşımların başarılı olduğunu ortaya koymaktadır.

Anahtar kelimeler: Popülasyona dayalı yerel arama, Memetik algoritma, Üst-sezgiseller, Yinelemeli Yerel Arama, Kombinatoryal Optimizasyon

Düzenlenmemiş Sürüm - Unedited Version

1 Introduction

There are many different heuristic/meta-heuristic methods that can be employed for solving NP-complete combinatorial optimization problems [1]-[7]. Most of the time, these methods perform well in finding an optimal/near-optimal solution to the problem they are designed for. However, they will need to be redesigned to obtain good performance in other specific problems. Hyper-heuristics [8], on the other hand, are used to automate the selection (selection hyper-heuristics) or generation of heuristics (generation hyper-heuristics) to solve combinatorial optimization problems. This study uses selection hyper-heuristics, which seeks a good method based on heuristic selection approaches. The method is searched among a set of heuristics, which are called low-level heuristics in the context of hyper-heuristics. The chosen low-level heuristic is considered to produce a new solution from the current one. Subsequently, the new solution is assessed and either approved or discarded according to the acceptance criteria. Hyper-heuristics Flexible Framework (HyFlex) enables researchers to design and test general-purpose heuristic search methods with a focus on selection hyper-heuristics. It contains implementations of many problem-specific operators as low-level heuristics for nine different problem domains. This study uses HyFlex as a design and test environment for the proposed methods in cross-domain search.

Cross-domain search tasks, where the goal is to find the best solution to a problem that spans multiple domains or disciplines, can be effectively studied with local search approaches. The key advantage of local search methods is that they may be used to efficiently explore a large solution space, even when there is no obvious path to the ideal solution. Moreover, using a population of candidate solutions to explore the search space allows exploration of a larger portion of the search space. In summary, population-based search enables concurrent exploration of many locations within the search space, while local search helps to converge toward promising solutions in each region. A memetic algorithm (MA) is a population-based meta-heuristic approach that combines local search techniques with genetic algorithms. This combination enables the MA to discover solutions that neither a genetic algorithm nor a local search approach could discover alone. In its simplest version, a local search method is employed after crossover and mutation operators in the genetic algorithms. The MAs have been applied effectively to many different problems [9]-[13]. Similar to other meta-heuristic methods, the memetic algorithms require designing problem-specific operators for different problem domains.

*Corresponding author/Yazışılan Yazar

In this study, we introduce two population-based local search approaches for cross-domain search, specifically, population-based iterated local search (PILS) and steady-state memetic algorithm (SSMA). The implementations of these approaches are done within HyFlex. PILS maintains a set of solutions (population), which allows to explore the search space more effectively. At each step, a new single solution is created by perturbing the current solution and applying a local search operator. Regardless of the quality of the new solution, it is accepted and inserted into the population, replacing it with the worst-performing solution in the population. Perturbation and local search operators are selected from a set of operators (low-level heuristics) provided in HyFlex. The local search operator is chosen randomly, while the perturbation operator is selected based on the heuristic selection component in the selection hyper-heuristics. To select the operator, two heuristic selection techniques are used, namely Simple Random (SR) and Reinforcement Learning with Tournament Selection (RLT). Simple Random uses no feedback, while Reinforcement Learning with Tournament Selection, which is a learning selection method, gets feedback during the search. The second approach, SSMA, is proposed in [14]. It is implemented within HyFlex and proved to perform well on cross-domain search. This algorithm applies genetic operators, specifically, parent selection, crossover, and mutation, to generate a candidate solution at each iteration. Then, a local search is employed to the candidate solution to improve its quality. Finally, the candidate solution is replaced by the worst solution in the current population. Since the algorithm is implemented within HyFlex, an operator is randomly selected from a corresponding set of low-level heuristics, which corresponds to Simple Random selection method. In this study, as an improvement to the original SSMA, RLT is also incorporated as another selection method besides SR.

The experiments are conducted in two phases: Experiments on **(1)** six problem domains provided in the original HyFlex and **(2)** three additional HyFlex problem domains. In the first phase, the performances of four local search approaches, namely *SSMA-SR, SSMA-RLT, PILS-SR*, and *PILS-RLT*, are assessed on six problem domains in HyFlex. We consider five instances for each problem domain used in Cross-domain Heuristic Search Challenge held in 2011 (CHeSC 2011). The experimental results reveal that PILS-RLT outperforms the other approaches based on the statistical results. We also compare four approaches to the selected approaches including twenty CHeSC competitors, a self-adaptive self-configuring steady state multimeme memetic algorithm (SSMMA) [15], and a multi-stage hyper-heuristic (MSHH) [16]. All algorithms are ranked according to the Formula 1 point system. SSMA-RLT ranks third among twenty-six approaches, beating SSMA-SR and PILS-based algorithms. Furthermore, the second phase includes the experiments on three additional HyFlex problem domains presented in [17]. In this phase, ten instances for each problem domain are used to assess the performance of four approaches. In this phase, PILS-SR yields better performance than the others according to the statistical results. In addition, our approaches are compared with six algorithms presented in [17]. The median values generated by the approaches and the scores for all approaches according to the Formula 1 point system are presented in this phase. The results show that PILS-SR gives competitive results for all problem instances, and it ranks second among ten approaches.

Motivation and Contribution

Population-based local search approaches can effectively address cross-domain search tasks [14-15]. Through the integration of population-based strategies, which exploit diverse candidate solutions and collective intelligence, these approaches present a promising method for improving solution quality, enhancing exploration capabilities, and achieving convergence toward optimal solutions. This study introduces two population-based local search approaches for cross-domain search. The following is a summary of the main contributions of this paper:

- PILS algorithm with two heuristic selection methods is proposed for the cross-domain search.
- RLT is integrated into the SSMA along with SR.
- A thorough experimental examination of all approaches is conducted.

The subsequent sections of the paper are structured in the following manner: Section 2 provides background information on hyper-heuristics, HyFlex, and local search-based algorithms, which is followed by the methodology. Then, the design of experiments and results are provided. The last section gives a conclusion.

2 Background

An overview of selection hyper-heuristics, population-based iterated local search, HyFlex, and local search algorithms proposed for HyFlex is presented in this section.

2.1 Selection hyper-heuristics

Selection hyper-heuristics decide on which low-level heuristic (LLH) will be employed to the current solution among a set of LLHs defined for the problem. After applying the chosen LLH, the created solution is accepted as the current solution or rejected, depending on a move acceptance approach. This search continues until the termination condition is satisfied. A thorough survey of selection hyper-heuristics can be read in [18]. In this section, we will outline some basic techniques and recent research in the field of selection hyper-heuristics.

Three basic categories of heuristic selection approaches as a component of selection hyper-heuristics are introduced in [19], i.e., random, greedy, and choice-function based. In the first category, they propose Simple Random (SR) and Random Permutation (RP) to apply a different LLH at each iteration and Random Permutation Descent (RPD) and Random Descent (RD) to select a new LLH only if the current LLH does not improve anymore. When necessary, RD selects a random LLH, whereas RPD selects a LLH regarding a permutation of LLHs generated randomly at the beginning. The greedy approach (GR) computes the difference between the fitness value of the present solution and the solution produced by each LLH at each step. The LLH that causes the best improvement is then applied. For the last category, a choice-function (CF) is introduced to choose the next LLH. The performance of each heuristic selection method is examined with Only Improving (OI) and All Moves (AM) move acceptance methods, where AM accepts all moves and OI accepts only the moves that yield a better solution. As a result of an experimental study, in [19], it is concluded that combining the use of a choice function to select the LLH and accepting all moves performs better than all other combinations.

Sequence-based Selection (SS) hyper-heuristics utilizing the Markov model are identified to perform well in different

Düzenlenmiş Sürüm - Uncorrected Version

problem domains [20]-[31]. In [20], besides the SS method, a set of basic selection methods is paired with a set of move acceptance methods. Their basic selection methods include SR, RD, RP, RPD, and GR. In the experimental study, they pair each of these selection methods with the move acceptance methods: OI, Improve or Equal (IE), Great Deluge (GD), Late Acceptance (LA), and Simulated Annealing (SA). IE accepts the moves that do not result in a worse solution. LA, GD, and SA are non-deterministic move acceptance approaches; in GD and SA, the improving and equal moves are accepted; in LA, a solution is accepted or rejected following a comparison with another solution that was created several steps earlier. Moreover, the move that results in a worse solution is accepted if it is below a dynamically changing threshold in GD. On the other hand, it is accepted with a probability which decreases as the search continues in SA. As a result, the SS-GD pair is found to perform the best for urban transit route design problems.

2.2 Population-based Iterated Local Search

Iterated Local Search (ILS) is a single point-based search algorithm consisting of two steps: perturbation and local search. Many successful applications of ILS are presented in the literature [23],[24]. On the other hand, some recent studies make use of the population for exploring the different regions in the search space [25],[26].

Evolutionary iterated local search algorithm (EILS), in which ILS is integrated with evolutionary operators, is introduced to solve the problem of antenna positioning in cellular networks [25]. It is also a population-based ILS algorithm including two neighborhood structures for local search, crossover and mutation operators, and a selection procedure based on a binomial distribution. The main steps of EILS are as follows: Initially, a population of individuals and a candidate solution are randomly created. At each iteration, the local search is employed to the candidate solution to obtain an improved one. The improved solution is inserted into the current population, replacing it with the worst solution in the population. Then, the two-point crossover is applied to generate a new population. The mutation is applied to the candidate solution which is selected based on the selection procedure for a new cycle. This process is iterated until a specified stopping criterion is met.

In [26], a population-based ILS approach is developed for solving dynamic vehicle routing problems. First, the initial population is generated randomly. Then, a new solution is created by ILS at each iteration. The new solution is inserted in the current population concerning its quality and diversity. In ILS part, the skewed variable neighborhood descent (SVND) is considered as a local search method. In the perturbation step, the current solution is perturbed by three different rules to generate a new one. This perturbation procedure provides diverse solutions and high quality. The first two rules use the crossover operator, but the third one uses the ruin-recreate operator. At each iteration, a rule is chosen using the ϵ-greedy strategy. Finally, the new solution has been accepted using the exponential Monte Carlo criterion (EMS).

2.3 HyFlex

HyFlex (Hyper-heuristics Flexible Framework) [27] is a software framework that contains the implementations of problem-specific components for a set of problem domains. The framework was implemented for the Cross-domain Heuristic Search Challenge (CHeSC) held in 2011 [28]. The initial set of problems implemented in the original Hyflex includes Vehicle Routing (VRP), Permutation Flow Shop (PFS), Travelling Salesman (TSP), Personnel Scheduling (PS), One Dimensional Bin Packing (BP), and Boolean Satisfiability (SAT) problems. Then, in 2015, Adriaensen et. al. [17] implemented three more problem domains, i.e., 0–1 Knapsack Problem (KP), Quadratic Assignment Problem (QAP), and MaxCut (MAC) in HyFlex. This version of HyFlex including three new problem domains will be referred to as *extended HyFlex* throughout the paper. For each problem domain, the solution representation, initial solution generation, fitness function evaluation, and low-level perturbative heuristics are implemented in the framework. Therefore, this framework makes it possible for the user not to think of details of the problem domain, rather, the user can concentrate on the hyper-heuristic methods. As a result, HyFlex is commonly used for research on cross-domain selection perturbative hyper-heuristics.

In the HyFlex implementation, several problem instances are included for each problem domain. Also, four different types of low-level heuristics (LLH) are implemented for each domain. The LLH types are crossover (XO), mutational (MU), hill-climbing (HC), and ruin-recreate (RR). Mutational heuristics are also called perturbation heuristics, and they result in a small change in the solution. Ruin-recreate heuristics, on the other hand, first destructs some part of the solution (ruin) and then reconstructs this part (recreate), which leads to a larger change in the solution. The process of hill climbing heuristics involves making incremental changes to a solution and only accepting the new solution if it is better than the original. The final type of heuristic, called crossover, combines two solutions to create a single offspring. The number of different LLHs implemented in HyFlex for each type and for each domain is given in Table 1.

Table 1. The number of different LLHs implemented in HyFlex for each type and for each domain.

HyFlex	Problem	MU	RR	HC	XO	Total
Original	SAT	6	1	2	2	11
	BP	3	2	2	1	8
	PS	1	3	5	3	12
	PFS	5	2	4	4	15
	TSP	5	1	3	4	13
	VRP	3	2	3	2	10
Extended	KP	5	2	6	3	16
	QAP	2	3	2	2	9
	MAC	2	3	3	2	10

There is a wide variety of selection hyper-heuristics that are proposed and evaluated for the problem domains provided in both the original and the extended HyFlex [18]. Some methods include a hidden Markov model-based method [29], a tensor-based selection hyper-heuristic [30], an iterated multi-stage hyper-heuristic approach [16], iterated local search based hyper-heuristic [31], a multi-armed bandit selection mechanism for hyper-heuristics [32], and a simulated annealing approach [33].

Moreover, the performance of six selection hyper-heuristics, namely, dominance-based, and random descent hyper-heuristic, sequence-based selection hyper-heuristic, fuzzy late acceptance-based hyper-heuristic, simple random-great deluge, robinhood (round-robin neighborhood) hyper-heuristic, modified choice function, are compared on three problem domains in the extended HyFlex in [34].

2.4 Local search algorithms for hyFlex

Iterated local search is used as a selection hyper-heuristic in literature. In addition to the heuristic selection and acceptance

Elements Sunum - Uncorrected Version

method, a local search is included in the ILS-based selection hyper-heuristic. The fair-share iterated local search (FS-ILS), proposed in [35], is one of the well-known selection hyper-heuristics for cross-domain search. In this method, a constructive heuristic is considered to create the initial solution. In the perturbation step, a LLH is chosen among mutation and ruin-recreate heuristics according to the acceptance rate of previous solutions and applied to the current solution. In the local search step, a heuristic is selected from the hill-climbing heuristics provided in HyFlex and applied in tabu-search manner. Then, the metropolis acceptance condition is used to accept or discard the new solution. The search process is restarted when there is no improvement for a certain time. The FS-ILS performance has been evaluated in six HyFlex problem domains. It is reported that FS-ILS outperforms the competing algorithms from the CHeSC 2011 competition. In [17], three problems provided in the extended HyFlex are used to examine the performance of FS-ILS. FS-ILS is compared to five selection hyper-heuristics including the winner of CHeSC 2011 (AdapHH), an evolutionary programming hyper-heuristic (EH), FS-ILS without restart (NR-FS-ILS), and two simple single point hyper-heuristics (AA-HH and ANW-HH). The results reveal that AdapHH outperforms the others for the extended HyFlex.

Two recent studies on the iterated local search hyper-heuristics for cross-domain search are presented in [36],[37]. In [36], probabilistic learning is employed in the perturbation step. This algorithm combines the advantages of Thompson Sampling algorithm and FS-ILS. Unlike FS-ILS, the current solution is perturbed according to four operators: mutation heuristic followed by ruin-recreate, ruin-recreate followed by ruin-recreate, only mutation heuristic, and only ruin-recreate heuristic. The performance of the proposed approach is assessed on six problem domains in HyFlex. The experimental findings suggest that the algorithm described in this study has superior performance compared to FS-ILS and the competing approaches in CHeSC 2011. It is significantly better than FS-ILS for BP, PS, and PFS. The second approach is the evolutionary iterated local search hyper-heuristic (EA-ILS), combining the principles of ILS and the evolutionary algorithm [37]. In this algorithm, a sequence of low-level heuristics is created and updated in accordance with ILS steps that include first implementing multiple perturbation heuristics and finally finishing with a local search heuristic. A new mutation operator has been proposed to regulate the order of low-level heuristics. The performance of EA-ILS is examined on three combinatorial optimization problems provided in extended HyFlex. It provides superior performance in these problem domains. Its performance is also evaluated on the problem instances used in CHeSC 2011. It gives competitive results for these instances.

In general, the problem-specific operators are designed in MAs, requiring a different implementation for each problem domain [10]. However, in [14], the MA is implemented without any modification for various optimization problems in HyFlex. The authors present two MAs, namely steady-state memetic algorithm (SSMA) and trans-generational memetic algorithm (TGMA) for the six problem domains used in CHeSC2011 competition. In the SSMA, an offspring is created at each generation. To create an offspring; crossover, mutation, and hill climbing operators are sequentially applied. Binary tournament selection is considered to pick two parents for crossover. The selection of operators, namely crossover, mutation, and hill climbing, is performed randomly from a set of defined operators. The offspring substitutes for the worst-performing individual in the population. On the other hand, TGMA creates an offspring population at each generation instead of generating a single offspring. It uses the same selection rules to pick a genetic operator as in SSMA. The experiments are conducted under the competition conditions. The results indicate that SSMA yields superior performance compared to TGMA for most cases.

One of the main issues in meta-heuristic approaches is parameter tuning. The experiments for the parameter tuning of the SSMA are conducted using a *design of experiments* approach [38] across several problem domains provided in HyFlex. Based on the experimental results, the tuned SSMA performs better in most instances. The results also show that the crossover operators did not influence the performance of SSMA. The authors compare the performance of SSMA with the best settings, referred to as SSMA-Best, with that of SSMA, TGMA, and twenty competing approaches in CHeSC 2011. The experimental results show that the SSMA-Best outperforms SSMA and TGMA and ranks fourth among twenty-three algorithms according to the Formula 1 point system. Moreover, the performance of SSMA-Best is tested on the problems in the extended HyFlex and compared with some selection hyper-heuristic approaches. SSMA-Best gives competitive results for these problem domains.

A self-adaptive steady state multimeme memetic algorithm (SSMMA) is presented in [15]. In this method, an individual consists of two parts: chromosome (solution) and memeplex (memetic information). A memeplex contains five memes, each of which encodes a selection of algorithmic components, namely the intensity of mutation parameter, crossover operator, mutation or ruin-recreate operator, and hill-climbing operator, depth of search parameter. Five memes within the memeplex utilize scores for each operator and their parameter settings. Tournament selection is used to select the operator and parameter setting. In the experiments, six optimization problems in HyFlex are used to assess the performance of SSMMA. The results show that SSMA yields better performance than the other memetic algorithms including SSMA, TGMA, and a self-adaptive multimeme memetic algorithm (MMA) for most cases.

3 Methodology

In this study, a population-based iterated local search (PILS) and the steady state memetic algorithm (SSMA) are utilized for cross-domain search. This section presents the details of PILS, SSMA, and the operator selection methods based on the heuristic selection methods.

3.1 Population-based Iterated local search

In this study, we propose a population-based iterated local search (PILS) algorithm for the cross-domain search. This algorithm has three main components: a population of individuals, ILS steps (perturbation, local search, acceptance criterion), and a population update mechanism. The initial population is randomly created and the best solution in the population is picked as the current solution. Then, the current solution undergoes a randomly selected hill-climbing operator (line 3 in Figure 1). At each step, a candidate solution is generated by ILS and the population is updated. In the perturbation step (lines 5 and 6 in Figure 1), an operator is selected from a set of crossover, mutation, and ruin-recreate heuristics according to the operator selection method (see Section 3.3). If the selected operator is a crossover heuristic, two parents are selected by using the binary tournament selection. Then, the selected crossover operator is applied to generate a candidate solution. It should be noted that the crossover operators provided in HyFlex generate only one solution. Otherwise, it is applied to the current solution (S_c) to create a candidate solution (S_n). In the local search step, a hill-climbing operator is selected randomly among the set of hill-climbing operators provided in HyFlex and is employed to the candidate solution (lines 7 and 8 in Figure 1). Then, the candidate solution is accepted regardless of its quality (line 9 in Figure 1). The candidate solution replaces the individual having the lowest performance in the population. Figure 1 presents the pseudo-code of the PILS.

```
1-  Create random individuals for initial population (pop)
2-  Select the best solution in the population as the current
    solution (Sc)
3-  Apply a randomly selected hill climbing operator to Sc
4-  while (termination criterion is not satisfied) do
5-      opID ← SelectPerturbationOperator()
6-      Sn ← ApplyPerturbationOperator(opID, Sc)
7-      opID ← SelectHillClimbing()
8-      Sn ← ApplyHillClimbing(opID, Sn)
9-      Sc ← Sn
10-     Replace the lowest-performing individual in the
        population by
          the candidate solution (Sc)
11-     Update the score values of operators if necessary
12- end while
```

Figure 1. The pseudo-code of the PILS algorithm.

3.2 Steady state memetic algorithm

We consider the steady state memetic algorithm (SSMA) proposed in [14]. The pseudo-code of the SSMA is presented in Figure 2. First, the individuals in the population are generated randomly. Then, a randomly selected hill-climbing operator is employed to each individual in the population. Afterwards, the main cycle of the SSMA is employed: Tournament selection is used to select two parents for recombination. These parents undergo crossover, mutation, and hill climbing operators to generate offspring. Crossover, mutation, and hill climbing operators are randomly selected. Then, at the end of each iteration, the offspring replaces the individual with the lowest performance within the population. In this method, the ruin-recreate heuristics of HyFlex are used as mutation operators. In Hyflex, the solution representation, initial solution generation, fitness function evaluation, low-level perturbative heuristics, and genetic operators are implemented for each problem domain. Therefore, those operators are used as they are defined in the framework.

3.3 Operator selection method

In SSMA [14], an operator from the set of operators for crossover, mutation, and hill climbing is randomly chosen. This is the same as the Simple Random (SR) which is one of the heuristic selection methods in selection hyper-heuristics. In addition, we also employed SR in the PILS algorithm for selecting the perturbation operators (line 5 in Figure 1).

Additionally, we use another heuristic selection method incorporating a learning mechanism, namely Reinforcement Learning with Tournament selection (RLT) [15]. RLT maintains a utility score for each operator. The initial score of each operator is set to the same value. An operator is selected based on tournament selection according to the score values whenever an operator is needed. The score of the selected operator is updated depending on the change in the solution quality. If an offspring generated by a crossover operator is better than any one of the parents, it is said that there is an improvement in solution quality. Similarly, if an offspring generated by a mutation operator is better than the one before mutation, there is an improvement. On the other hand, in this method, a hill-climbing operator is randomly selected at each iteration to maintain diversity.

```
Create random individuals for initial population (pop)
Apply a randomly selected hill climbing operator to
each individual
while (termination criterion is not satisfied) do
    Parent1 ← TourSelect(pop, tour_size)
    Parent2 ← TourSelect(pop, tour_size)
    opID ← SelectOperator(crossoverList)
    Offspring ← ApplyXover(opID, Parent1, Parent2)
    opID ← SelectOperator(mutationList)
    Offspring ← ApplyMutation (opID, Offspring)
    opID ← SelectOperator(hillClimbingList)
    Offspring ← ApplyHillClimbing(opID, Offspring)
    Replace the lowest-performing individual in the
    population by
        Offspring
    Update the score values of operators if necessary
end while
```

Figure 2. The pseudo-code of the SSMA.

4 Experimental setup

Experiments are conducted in accordance with the following CHeSC 2011 competition rules. Each algorithm is executed for 31 runs. We consider six problem domains in HyFlex, namely Boolean Satisfiability (SAT), Travelling Salesman (TSP), Personnel Scheduling (PS), One Dimensional Bin Packing (BP), Permutation Flow Shop (PFS), and Vehicle Routing (VRP) problems. The five competition instances are employed for each problem. Three additional problem domains in the extended HyFlex are also used in experiments: 0–1 Knapsack Problem (KP), Quadratic Assignment Problem (QAP), and MaxCut (MAC). Each of these domains consists of 10 instances. A benchmarking program on the competition website determines the equivalent time limit on test machines, which corresponds to 10 minutes (600 seconds) on the competition machine. The experiments are conducted on a PC with 8 GB RAM and Intel Core i7-2600K Processor.

We utilize the CHeSC 2011 ranking methodology that relies on Formula 1 points from before 2010 to evaluate and compare the performance of the approaches. In this system, for each problem instance, the top eight approaches are determined after comparing the median values of all approaches obtained over 31 runs. The approach in the first place is awarded 10 points, the second place receives 8 points, and each subsequent approach receives 6, 5, 4, 3, 2, and 1 points, respectively. Any other approaches get 0 points. If there is a tie for a given instance, the mean of corresponding points is calculated for the tied position and assigned for each approach. The final score of an approach is calculated as the sum of points earned over all problem instances.

The parameters of SSMA-SR are tuned using Taguchi's design of experiments method [38]. The best settings of the parameter among 25 different configurations are reported as: Population size is 5, intensity of mutation in mutation and ruin-recreate operators is 0.2, depth of search in local search operators is 1.0, and the tournament size is 5. Here, the intensity of mutation is a parameter that specifies the extent to which a given solution is perturbed by the corresponding operator and the depth of search parameter is the termination condition (i.e., the number of steps) in local search. The algorithm with the best setting is referred to as SSMA-Best (SR). We consider the same settings for all operator selection methods. In addition, we use the following settings for RLT in both algorithms recommended in [15]: the initial score of each operator is set to 0. The score of the selected operator is increased by 1 if it improves the solution; otherwise, there is no change in the value of the score. The tournament size for RLT is set to 2.

For the statistical comparison, we perform the Wilcoxon signed-ranks test at a 95% confidence interval for paired approaches and one-way ANOVA and Tukey's HSD test at a 95% confidence interval for more than two approaches. The overall win/tie/loss *(w / t / l)* counts over 30 instances are provided in the results tables to provide a summary for the statistical comparisons.

5 Results and discussion

In this study, we provide a comprehensive analysis of local search-based algorithms for cross-domain search. The experimental evaluation of these algorithms consists of two parts: First, we explore the performance of algorithms and compare them to the methods participating in the CHeSC 2011 competition on six problem domains in the original HyFlex. Then, the performance of these algorithms is tested on three problem domains provided in the extended HyFlex. The results and discussion for each part of the experiments are presented in this section.

5.1 Comparison of local search-based algorithms

In this part of the experiments, we conduct experiments to compare the performance of local search-based algorithms on six problem domains in HyFlex. Table 2 presents the average fitness values generated by these methods for each instance of six problem domains in HyFlex. Each row in the table presents the performance of these approaches for the corresponding instance. Based on the results, we can conclude that:

- ✓ For all problem instances of SAT, PILS-RLT is the best-performing approach. It performs significantly better than the others for 10 cases. There are no statistically significant differences between PILS-RLT and PILS-SR for the SAT problem. In addition, PILS-based algorithms are significantly better than SSMA-based algorithms. The differences between SSMA-SR and SSMA-RLT are not statistically significant for all instances.
- ✓ PILS-based algorithms are also significantly better than SSMA-based algorithms for all problem instances of BP. PILS-RLT is significantly better than PILS-SR for instance 4, but it is significantly worse for instance 4. Others have no statistically significant differences. In addition, SSMA-RLT is slightly worse than SSMA-SR for instance 1, however, it is significantly better than SSMA-SR for the other instances.
- ✓ The differences between the approaches are not statistically significant for the following problem instances of PS: instance 1 and instance 4. SSMA-SR is significantly better than PILS-SR for instances 2, 3, and 5. SSMA-based algorithms significantly outperform the PILS-based algorithms for instance 5.
- ✓ SSMA-RLT delivers good performance for almost all problem instances of PFS except for instance 3. ILS-RLT yields better performance for this instance; however, there is no statistical difference between them. In addition, PILS-SR performs worse than SSMA-based algorithms.
- ✓ For the problem instances of TSP, SSMA-RLT is the best-performing approach overall. There are no statistical differences between all approaches for instance 1. However, SSMA-based algorithms outperform PILS-SR for the other instances.
- ✓ For the problem instances of VRP, PILS-based algorithms produce better performance than SSMA-based algorithms. PILS-based algorithms are significantly better than SSMA-based algorithms for instance 1 and 2. However, there are no statistical differences between all approaches for instance 3 and 4.

To summarize the result of the statistical comparison between approaches, the overall win/tie/loss counts for operator selection methods on six problem domains provided in HyFlex based on Tukey's multi-comparison test at a 95% confidence level is calculated. In HyFlex, there are 30 instances, and an algorithm is compared to the other four algorithms for each problem instance. Therefore, an algorithm can achieve 120 significance states in total over the other algorithms: *win (w)* count gives the number of cases where the corresponding algorithm is statistically better than the others. *tie (t)* count gives the number of cases in which the corresponding algorithm is not statistically different from others. *loss (l)* count gives the number of cases where the corresponding algorithm is statistically worse than the others. These results are presented in the last row of Table 2. PILS-RLT is the best-performing approach overall. It performs significantly better than the others for 31 cases, especially for SAT, BP, and VRP problems. Besides, SSMA-SR delivers poor performance with the 12 best significance states.

Figure 3 illustrates the convergence behavior of all approaches (SSMA-SR, SSMA-RLT, PILS-SR, and PILS, RLT) for the selected instances of each problem. All algorithms can converge to a minimum in a short time for most instances.

The performance of local search-based methods is also compared with selected approaches from the literature, including twenty competing approaches from CHeSC 2011, dominance-based and relay hybridization multi-stage hyper-heuristic (MSHH) [16], SSMA with the best parameter setting (SSMA-Best) [38], and a self-adaptive self-configuring steady state multimeme memetic algorithm (SSMMA) [15]. These approaches are selected since they are well-known approaches tested on six problem domains in HyFlex. It is reported that MSHH outperforms 20 competing approaches from CHeSC 2011 [16]. SSMMA is based on a steady state memetic algorithm and RLT is used for selecting components.

The results of MSHH and SSMMA are provided by the authors of the corresponding papers. We run the SSMA-RLT, SSMA-Best (SR), PILS-SR, and PILS-RLT for all instances under the same termination condition used in the CHeSC competition. Table 3 presents the score values of each approach according to Formula 1 point system with respect to median values for each problem. MSHH ranks first among the 24 approaches. The winner of CHeSC 2011, AdapHH, takes the second place overall.

Table 2. Average fitness values generated by different local search-based algorithms for instances of six problem domains in HyFlex.

Problem	ID	SSMA-SR	SSMA-RLT	PILS-SR	PILS-RLT
SAT	1	17.323	17.419	11.290	**11.226**
	2	44.871	44.710	33.226	**27.000**
	3	27.516	27.774	16.323	**13.935**
	4	21.516	22.968	15.516	**14.484**
	5	17.871	17.258	14.290	**13.968**
BP	1	0.05444	0.05009	**0.02868**	0.03359
	2	0.01047	0.01053	0.00684	**0.00677**
	3	0.03073	0.02619	**0.01297**	0.01323
	4	0.10968	0.10922	0.10895	**0.10870**
	5	0.05705	0.04782	**0.02594**	0.02615
PS	1	20.77	**20.68**	22.10	22.19
	2	9560.71	**9519.16**	9944.10	9791.90
	3	**3210.77**	3224.39	3253.65	3239.61
	4	**1573.55**	1578.06	1625.94	1603.48
	5	**308.90**	313.71	335.52	335.97
PFS	1	6251.65	**6249.97**	6261.19	6258.23
	2	26799.84	**26774.16**	26813.55	26799.32
	3	6348.06	6351.06	6357.06	**6347.94**
	4	11375.45	**11369.81**	11396.39	11390.23
	5	26624.13	**26592.35**	26641.61	26615.06
TSP	1	**48218.62**	**48218.62**	48225.61	48230.47
	2	**20892206.89**	20917311.69	21137038.29	21074726.35
	3	6812.84	**6808.94**	6819.90	6814.59
	4	66754.38	**66707.01**	67256.04	66946.82
	5	53541.38	**53250.28**	53850.54	53271.37
VRP	1	71641.79	75218.71	**60020.50**	60070.81
	2	13881.05	13916.49	13415.97	**13414.84**
	3	147510.17	147502.31	147304.53	**147287.83**
	4	21081.29	21343.58	**21078.73**	21205.92
	5	147549.95	148170.72	147541.30	**147083.33**
	w / t / l	*12 / 48 / 30*	*22 / 42 / 26*	*25 / 41 / 24*	***31 / 49 / 10***

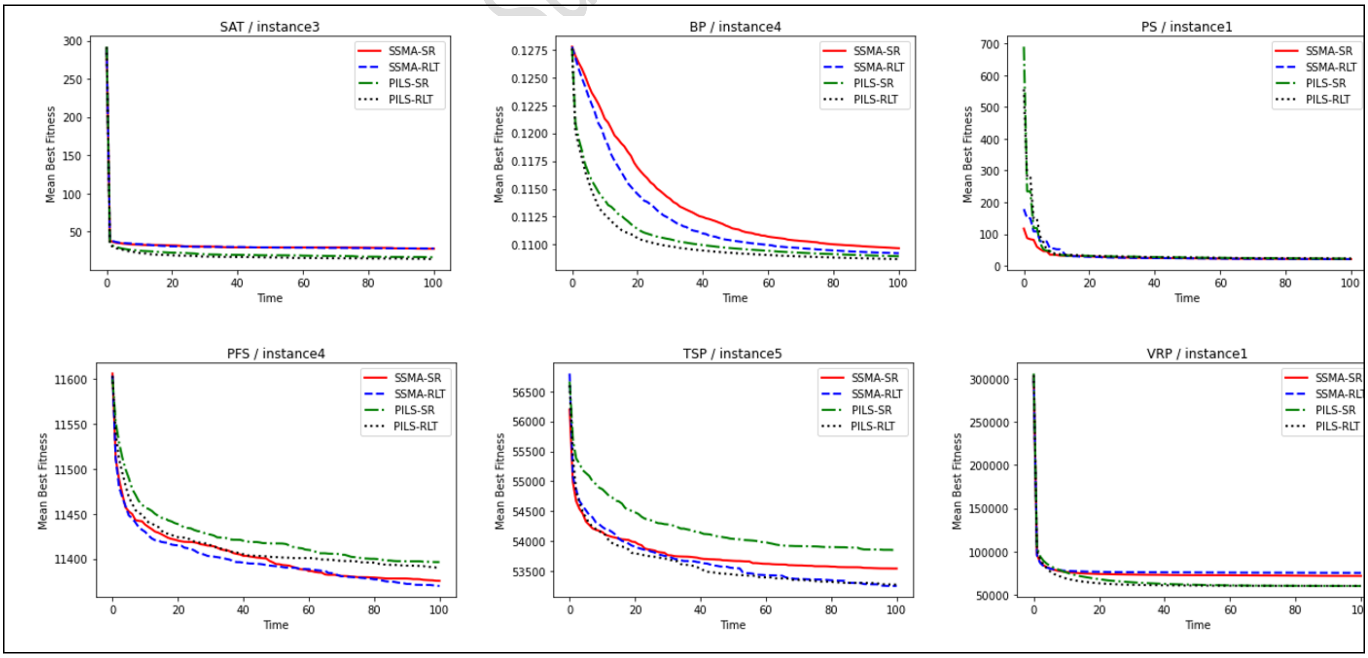

Figure 3. Convergence plot of the approaches (SSMA-SR, SSMA-RLT, PILS-SR, and PILS, RLT) for the selected instances.

Table 1. The ranking and scores for each approach according to the Formula 1 point system across six problem domains in HyFlex.

Rank	Algorithm	SAT	BP	PS	PFS	TSP	VRP	TOTAL
1	MSHH [16]	48	36	4	24	41	0	153
2	AdapHH [39]	23.45	43	4	25	29.2	5	129.65
3	**SSMA-RLT**	**0**	**0**	**34.83**	**29.5**	**32.2**	**3**	**99.53**
4	ML [27]	6.5	7	21.83	27.5	3	19	84.83
5	VNS-TW [27]	23.45	0	26.5	23.5	6.2	2	81.65
6	**SSMA-Best (SR)**	**0**	**0**	**38.5**	**11.5**	**17**	**9**	**76**
7	SSMMA [15]	28.95	16	8	0	0	8	60.95
8	**PILS-RLT**	**0**	**3**	**4**	**14.75**	**7.5**	**28**	**57.25**
9	PHunter [40]	4.5	1	7	2.5	15.2	25	55.2
10	NAHH	8.83	15	0	17.5	7	5	53.33
11	EPH	0	4	3	10	24.2	11	52.2
12	HAHA [41]	21.95	0	15.5	0.25	0	9	46.7
13	ISEA	1.5	21	10.83	0	6	1	40.33
14	KSATS-HH	15.2	6	1	0	0	17	39.2
15	**PILS-SR**	**0**	**4**	**0**	**3.5**	**1.5**	**16**	**25**
16	HAEA	0	0	0	0.25	3	18	21.25
17	ACO-HH	0	15	0	4.25	1	0	20.25
18	AVEG-Nep	8.5	0	0	0	0	7	15.5
19	XCJ	2.17	10	0	0	0	3	15.17
20	GenHive	0	8	2	1	0	3	14
21	GISS	0	0	6	0	0	4	10
22	DynILS	0	0	8	0	0	2	10
23	SA-ILS	0	6	0	0	1	0	7
24	MCHH-S	2	0	0	0	0	0	2
25	SelfSearch	0	0	0	0	0	0	0
26	Ant-Q	0	0	0	0	0	0	0

In addition, SSMA-RLT performs poorly with a score of 0 for SAT and BP problem domains; however, it ranks 3rd with a total score of 99.53 overall among the 26 approaches. It also gets the highest score for the PFS problem. PILS-RLT and PILS-SR rank the 8th and 15th, respectively. PILS-RLT gets the highest score for VRP. SSMA-Best outperforms the PILS-based algorithms but is worse than SSMA-RLT.

Figure 4 shows the boxplots for the fitness values of the approaches, namely MSHH, SSMMA, SSMA-SR, SSMA-RLT, PILS-SR, and PILS-RLT, for the selected instances. The comparison is done with MSHH and SSMMA since the MSHH is the best-performing approach and SSMMA is the improved version of SSMA. SSMA and PILS give competitive results for these instances except for SAT and BP problems.

5.2 Comparison of local search-based algorithms on extended HyFlex problems

In this part, the performance of the presented algorithms is assessed on the extended HyFlex problems. As mentioned before, the extended HyFlex includes three problem domains, namely the Knapsack Problem (KP), Quadratic Assignment Problem (QAP), and MaxCut (MAC) [19]. The experiments are conducted on ten instances for each problem domain.

Table 4 provides the average fitness values generated by different operator selection methods for each instance of three problem domains in extended HyFlex. The last row of the table shows the overall (w/t/l) counts for operator selection methods on three problem domains provided in extended HyFlex based on Tukey's multi-comparison test at a 95% confidence level. In general, PILS-SR is the best-performing approach overall. It performs significantly better than the others for 30 cases. SMA-SR and PILS-RLT give competitive results. Based on the average fitness values and the statistical analysis results, it can be revealed that:

- ✓ For the problem instances of KP, ILS-SR performs significantly better than SSMA-based algorithms in 11 cases. The difference between PILS-SR and PILS-RLT is not statistically significant for all instances. In addition, there are no significant differences between all algorithms for the problem instances 0, 2, and 4 of KP.
- ✓ ILS-SR generates better results than the others for 7 out of 10 instances of QAP. Even though PILS-RLT is better than PILS-SR for 2 instances, the difference between them is not statistically significant for all instances. SSMA-SR performs slightly better than other algorithms for instance 7. Besides, there are no significant differences between all algorithms for the following problem instances of QAP: instances 5 and 7.

To evaluate the performance of local search-based methods on the extended HyFlex, they are compared with six algorithms presented in [17] including Adap-HH (the winner of CHeSC 2011), the fair-share iterated local search (FS-ILS), an evolutionary programming hyper-heuristic with co-evolution (EPH), two single point selection hyper-heuristics (ANW-HH and AA-HH). Figure 5 presents the bar plot of the score values for each approach according to the Formula 1 point system across the three problems in the extended HyFlex. AdapHH ranks first with a score of 183.36. PILS-SR takes second place with a score of 167.03. The results show that the proposed PILS-based algorithms perform better than FS-ILS algorithm.

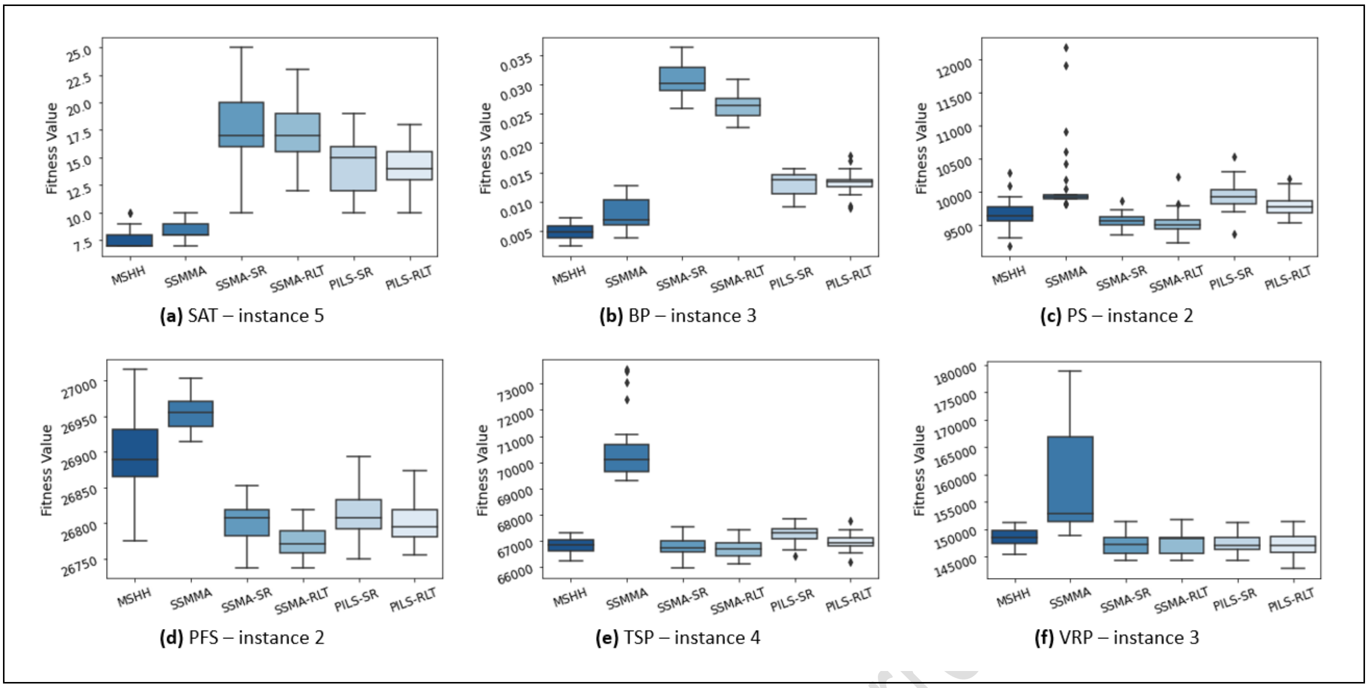

Figure 4. Boxplots of fitness values for a statistical comparison of the approaches (MSHH, SSMMA, SSMA-SR, SSMA-RLT, PILS-SR, and PILS, RLT) for the selected instances.

Table 4. Average fitness values generated by different local search-based algorithms for each instance of three problem domains in extended HyFlex.

Problem	ID	SSMA-SR	SSMA-RLT	PILS-SR	PILS-RLT
KP	0	-104022.80	-104037.20	**-104045.10**	-104044.70
	1	-1217160.30	-1218397.70	**-1257186.90**	-1255727.00
	2	-242198.50	-242265.70	**-242279.20**	-242129.50
	3	**-431356.80**	-431356.10	-431353.60	-431353.60
	4	-396166.50	**-396167.00**	**-396167.00**	-396166.50
	5	-4263544.40	-4258581.30	-4303583.30	**-4304513.20**
	6	-927956.80	-926313.90	**-940832.30**	-939046.70
	7	-1574340.60	-1574067.50	**-1577174.50**	-1576901.40
	8	-1530258.80	-1530366.80	-1530470.60	**-1530476.70**
	9	**-1467384.20**	-1467372.80	-1467359.30	-1467358.70
QAP	0	152531.30	152511.50	**152356.50**	152371.90
	1	154396.30	154460.60	**154257.80**	154307.50
	2	148543.30	148432.30	**148173.50**	148230.80
	3	150125.20	150338.60	150025.90	**150011.90**
	4	21374964.60	21390657.80	**21357006.10**	21375836.50
	5	1191614554.30	1191742050.40	1189277953.10	**1188300657.50**
	6	503374277.50	503424131.60	**502365685.20**	502540552.70
	7	**44839289.00**	44844394.50	44848103.90	44847149.70
	8	8177869.90	8181833.00	**8165227.50**	8170522.30
	9	273701.00	273605.30	**273465.70**	273586.00
MAC	0	-40749828.50	-40683723.90	**-40865627.30**	-40801985.10
	1	**-274386421.40**	-273434188.10	-271117693.60	-268593670.40
	2	-3023.60	-3024.70	**-3026.50**	-3023.80
	3	-3004.20	-3003.90	**-3010.30**	-3004.60
	4	-3009.60	-3009.30	**-3014.50**	-3009.30
	5	**-13100.00**	-13087.40	-13095.90	-13088.60
	6	**-1293.70**	-1290.30	-1292.10	-1280.10
	7	**-9938.80**	-9923.40	-9895.70	-9865.90
	8	-428.70	-424.80	**-431.60**	-427.40
	9	**-2747.40**	-2734.60	-2737.00	-2718.10
	w / t / l	*12 / 62 / 16*	*7 / 62 / 21*	***30 / 54 / 6***	*12 / 60 / 18*

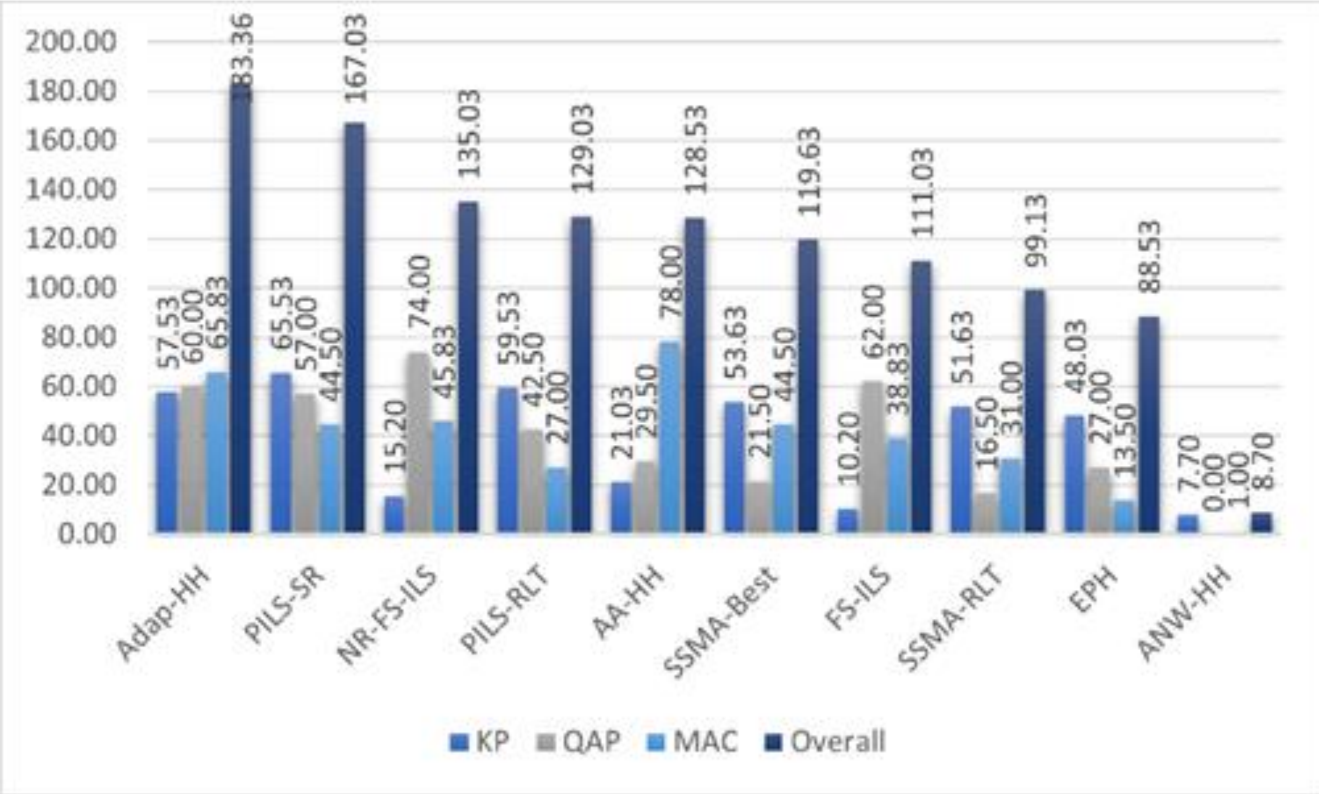

Figure 5. The scores for all approaches according to Formula 1 point system across the three problems in the extended HyFlex.

6 Conclusion and future works

In this study, we present two population-based local search approaches, namely a population-based iterated local search (PILS) and the steady state memetic algorithm (SSMA), for HyFlex. In both approaches, at each iteration, a single candidate solution is generated and replaced with the worst individual in the current population. To create a new solution, PILS uses a perturbation operator (crossover or mutation) and a local search operator, while SSMA applies a crossover, a mutation, and a local search operator, respectively. Two heuristic selection methods, namely Simple Random (SR) and Reinforcement Learning with Tournament Selection (RLT), are utilized to select an appropriate operator other than hill-climbing operators. A hill-climbing operator is randomly selected when needed.

For the performance evaluation of four approaches addressed in this study, we conduct the experiments in two parts: **(1)** tests on six problem domains in the original HyFlex; and **(2)** tests on three additional domains in the extended HyFlex. In the first part, all approaches are tested on a total of 30 instances used in CHeSC 2011. The statistical results show that PILS-RLT is the best-performing approach, which is significantly better than the others for 31 cases. Moreover, four approaches are compared to the selected approaches including twenty CHeSC competitors, a multi-stage hyper-heuristic, and a self-adaptive self-configuring steady state multimeme memetic algorithm according to the Formula 1 points system. In this system, the median values obtained over 31 runs are used to rank all approaches. According to the results, SSMA-RLT ranks third among twenty-six approaches. In the second part, four approaches are evaluated on three additional HyFlex problem domains, each of which has 10 problem instances. Based on the statistical results, PILS-SR yields better performance, which is significantly better than the others for 30 cases. We also compare the performance of our approaches to the six methods presented in [17]. The median values and ranking of all algorithms according to the Formula 1 point system are presented. The results reveal that PILS-SR ranks second and gives competitive results. In conclusion, the presented approaches are applicable to a variety of combinatorial problems, which is the main objective for designing automated methodologies as hyper-heuristics.

In future works, PILS can be applied to continuous optimization problems. Besides, PILS accepts the generated solution regardless of its quality, which corresponds to all moves as the acceptance criterion in the selection hyper-heuristic methods. Other acceptance criteria such as improving equality, only improving, etc. can be employed in PILS.

7 Acknowledgements

8 Author contribution statements

In the scope of this study, Author 1 contributed to forming the idea, literature review, performing the analysis, examining the results, and writing the article; Author 2 contributed to the creation of ideas, assessment of the results, and writing the article.

9 Ethics committee approval and conflict of interest statement

This article does not necessitate ethics committee approval. The authors do not have any competing interests to disclose that are pertinent to the subject matter of this study.

10 References

[1] Şahin Y, Karagül K. "Solving travelling salesman problem using hybrid fluid genetic algorithm (HFGA)". *Pamukkale University Journal of Engineering Sciences,* 25(1), 106-114, 2019.

[2] Demir İ, Kiraz B, Corut Ergin F. "Experimental evaluation of meta-heuristics for multi-objective capacitated multiple allocation hub location problem". *Engineering Science and Technology, an International Journal*, 29, 101032, 2022.

[3] Du KL, Swamy MNS. *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*, 1st ed, Switzerland, Birkhauser Cham, 2016.

[4] Martí R, Pardalos PM, Resende MGC. *Handbook of Heuristics*, Switzerland, Springer, 2018.

[5] Karaoğlan AD. "Optimization of welding job-shop scheduling problem under variable workstation constraint: an industrial application with Arena simulation based genetic algorithm". *Pamukkale University Journal of Engineering Sciences*, 28(1), 139-147, 2022.

[6] Kocer HG, Türkoğlu B, Uymaz SA. "Chaotic golden ratio guided local search for big data optimization". *Engineering Science and Technology, an International Journal*, 41, 101388, 2023.

[7] Türkoğlu B, Eroğlu H. *Genetic Algorithm for Route Optimization*. Editor: Dey N. Applied Genetic Algorithm and Its Variants, 51-79, Springer Nature Switzerland AG., 2023.

[8] Burke EK, Gendreau M, Hyde MR, Kendall G, Ochoa G, Özcan E, Qu R. "Hyper-heuristics: A Survey of the State of the Art", *Journal of Operational Research Society*, 64(12), 1695–1724, 2013.

[9] Chen X, Ong YS, Lim MH, Tan KC. "A multi-facet survey on memetic computation". *IEEE Transactions on Evolutionary Computation*, 15(5), 591–607, 2011.

[10] Neri F, Cotta C. "Memetic algorithms and memetic computing optimization: A literature review". *Swarm and Evolutionary Computation*, 2, 1–14, 2012.

[11] Gong G, Deng Q, Chiong R, Gong X, Huang H. "An effective memetic algorithm for multi-objective job-shop scheduling". *Knowledge-Based Systems*, 182, 104840, 2019.

[12] Moscato P, Mathieson L. *Memetic algorithms for business analytics and data science: a brief survey*. Editors: Moscato P, de Vries NJ. Business and Consumer Analytics: New Ideas, 545–608, 2019.

[13] Hemanth DJ, Kumar BV, Manavalan GK. *Recent Advances on Memetic Algorithms and Its Applications in Image Processing*, Singapore, Springer, 2020.

[14] Özcan E, Asta S, Altıntaş C. "Memetic algorithms for Cross-domain Heuristic Search". *13th UK Workshop on Computational Intelligence*, United Kingdom, 2013.

[15] Özcan E, Drake JH, Altıntaş C, Asta S. "A self-adaptive Multimeme Memetic Algorithm co-evolving utility scores to control genetic operators and their parameter settings". *Applied Soft Computing Journal*, 49, 81-93, 2016.

[16] Kheiri A, Özcan E. "An iterated multi-stage selection hyper-heuristic". *European Journal of Operational Research*, 250(1), 77–90, 2016.

[17] Adriaensen S, Ochoa G, Nowe A. "A benchmark set extension and comparative study for the hyflex framework". *IEEE Congress on Evolutionary Computation,* Sendai, Japan, 2015.

[18] Drake JH, Kheiri A, Özcan E, Burke EK. "Recent advances in selection hyper-heuristics". *European Journal of Operational Research*, 285(2), 405–428, 2020.

[19] Cowling P, Kendall G, Soubeiga E. "A hyper-heuristic approach to scheduling a sales summit". *Practice and Theory of Automated Timetabling III: Third International Conference*, Berlin, Heidelberg, 2000.

[20] Ahmed L, Mumford C, Kheiri A. "Solving urban transit route design problem using selection hyper-heuristics". *European Journal of Operational Research*, 274(2), 545–559, 2019.

[21] Kheiri A, Gretsista A, Keedwell E, Lulli G, Epitropakis MG, Burke EK. "A hyper-heuristic approach based upon a hidden Markov model for the multi-stage nurse rostering problem". *Computers and Operations Research*, 130, 105221, 2021.

[22] Kheiri A, Keedwell E. "A hidden markov model approach to the problem of heuristic selection in hyper-heuristics with a case study in high school timetabling problems". *Evolutionary Computation*, 25(3), 473–501, 2017.

[23] Brandao J. "A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem". *European Journal of Operational Research*, 284(2), 559–571, 2020.

[24] Lourenço HR, Martin OC, Stützle T. *Iterated Local Search: Framework and Applications*. Editors: Gendreau M, Potvin JY. Handbook of Metaheuristics, 129–168, Springer, 2019.

[25] Benmezal L, Benhamou B, Boughaci D. "Evolutionary iterated local search meta-heuristic for the antenna positioning problem in cellular networks". *Computational Intelligence*, 38(3), 1183–1214, 2022.

[26] Sabar NR, Goh SL, Turky A, Kendall G. "Population-based iterated local search approach for dynamic vehicle routing problems". *IEEE Transactions on Automation Science and Engineering*, 19(4), 2933–2943, 2022.

[27] Ochoa G, Hyde M, Curtois T, Vazquez-Rodriguez JA, Walker J, Gendreau M, Kendall G, McCollum B, Parkes AJ, Petrovic S, Burke EK. *Hyflex: A benchmark framework for cross-domain heuristic search*. Editors: Hao JK, Middendorf M. Evolutionary Computation in Combinatorial Optimization, 136–147, Berlin, Heidelberg, Springer, 2012.

[28] Burke EK, Gendreau M, Hyde M, Kendall G, McCollum B, Ochoa G, Parkes AJ, Petrovic S. *The cross-domain heuristic search challenge – an international research competition*. Editor: Coello CAC. Learning and Intelligent Optimization, 631–634, Berlin, Heidelberg, Springer, 2011.

[29] Kheiri A, Keedwell E. "A sequence-based selection hyper-heuristic utilising a hidden markov model". *Annual Conference on Genetic and Evolutionary Computation*, New York, USA, 2015.

[30] Asta S, Özcan E. "A tensor-based selection hyper-heuristic for cross-domain heuristic search". *Information Sciences*, 299, 412–432, 2015.

[31] Meignan D, Schwarze S, Voß S. "Improving local-search metaheuristics through look-ahead policies". *Annals of Mathematics and Artificial Intelligence*, 76(1-2), 59–82, 2016.

Düzenlenmemiş Sürüm - Uncorrected Version

[32] Ferreira AS, Goncalves RA, Pozo A. "A multi-armed bandit selection strategy for hyper-heuristics". *2017 IEEE Congress on Evolutionary Computation*, Donostia, Spain, 2017.
[33] Jackson WG, Özcan E, John RI. "Tuning a simulated annealing metaheuristic for cross-domain search". *2017 IEEE Congress on Evolutionary Computation*, Donostia, Spain, 2017.
[34] Almutairi A, Özcan E, Kheiri A, Jackson WG. "Performance of selection hyper-heuristics on the extended hyflex domains". *International Symposium on Computer and Information Sciences*, Krakow, Poland, 2016.
[35] Adriaensen S, Brys T, Nowe A. "Fair-share ils: A simple state-of-the-art iterated local search hyperheuristic". *2014 Annual Conference on Genetic and Evolutionary Computation*, New York, USA, 2014.
[36] Adubi SA, Oladipupo OO, Olugbara OO. "Configuring the perturbation operations of an iterated local search algorithm for cross-domain search: A probabilistic learning approach". *2021 IEEE Congress on Evolutionary Computation*, Krakow, Poland, 2021.
[37] Adubi SA, Oladipupo OO, Olugbara OO. "Evolutionary algorithm-based iterated local search hyper-heuristic for combinatorial optimization problems". *Algorithms*, 15(11), 2022.
[38] Gümüş DB, Özcan E, Atkin J. "An investigation of tuning a memetic algorithm for cross-domain search". 2016 IEEE Congress on Evolutionary Computation, Vancouver, British Columbia, Canada, 2016.
[39] Mısır M, Verbeeck K, De Causmaecker P, Vanden Berghe G. "An Intelligent Hyper-Heuristic Framework for CHeSC 2011". International Conference on Learning and Intelligent Optimization, Lecture Notes in Computer Science. Springer, 461-466, 2012.
[40] Chan CY, Xue F, Ip WH, Cheung CF. "A hyper-heuristic inspired by pearl hunting". International Conference on Learning and Intelligent Optimization, Lecture Notes in Computer Science. Springer, 349-353, 2012.
[41] Lehrbaum A, Musliu N. "A New Hyperheuristic Algorithm for Cross-Domain Search Problems". International Conference on Learning and Intelligent Optimization, Lecture Notes in Computer Science. Springer, 437-442, 2012.

Düzenlenmemiş Sürüm - Uncorrected Version