

Effectiveness of metaheuristics applied to Human Resource Allocation Problem in Short-Term Employment Sector – a case study

Paweł B. Myszkowski^{a,b}, Michał W. Przewoźniczek^{a,b}

^aWrocław University of Science and Technology

Faculty of Information and Communication Technology

ul. Ignacego Łukasiewicza 5, 50-371 Wrocław, Poland

email: {pawel.myszkowski, michal.przewozniczek}@pwr.edu.pl

^bEWL Group, Warsaw, Poland

Łukasz Kopociński

email: lkopocinski@gmail.com

Abstract—This work identifies and defines the real-world Human Resource Allocation Problem in Short-Term Employment Sector (HRAP-STE). HRAP is a subclass of the classic Human Resource Allocation Problem, adopted to the short-term employment sector, where the main everyday objective is assigning employees to the customer facilities (warehouses, factories, logistic centres, etc.). This process has three types of actors: customers, employees, and the company from the short-term employment sector, which provides a platform for cooperation. Usually, customers require significantly more employees than their available number. Since employee assignment is usually a subject of long-term cooperation, all customers should be satisfied (at least partially) even if they do not bring the highest profit. Thus, for a company in the short-term sector, HRAP refers to three objectives: profit from projects, priority of projects, and balance in the project portfolio to satisfy all clients. In this work, we define a specific HRAP-STE problem, consider its crucial elements, and define a benchmark set of real and artificial instances. To investigate the HRAP-STE as a real case study, we apply and compare well-known (meta)heuristics (shown effective in solving real-world problems) dedicated to solving discrete problems. The computational results show the advantages of (meta)heuristics in solving instances of a larger size.

I. INTRODUCTION

DECISION-MAKERS in the short-term employment sector aim to satisfy many contradicting objectives by assigning employees to customer facilities (warehouses, factories, logistic centres, etc.). First, assign employees to selected customers to ensure the highest profit. However, employee assignment is frequently a subject of long-term cooperation. Therefore, some customers should be selected even if they are less profitable. Finally, all customers' demands should be satisfied at least partially. Consequently, in strict cooperation with EWL GROUP company, we identify and define the Human Resource Allocation Problem in the Short-Term Employment Sector (HRAP-STE). HRAP-STE is a subclass of Human Resource Allocation Problem (HRAP). Therefore, it is an NP-hard combinatorial optimization problem [5].

The detailed analysis of the EWL GROUP business model showed that in the HRAP-STE problem there are several metrics, equally important from a business point of view. Some candidate assignments can be more profitable for the company, as they generate higher commissions. At the same time, certain jobs may be more important because they are linked to high-priority projects or key clients. Nevertheless, it is essential to offer all jobs *proportionally* to maintain *balance* in the company's portfolio and ensure that no client is overlooked in the assignment process. These metrics are crucial for the decision-makers in company practice when different decisions can be made each day, depending on the situation (i.e., emerging projects' deadlines), the importance of projects (i.e., changed clients' requirements), already realized portfolio, and finally, candidates available in this moment.

All the above metrics could be defined as objective functions, and the problem is considered as multi-objective HRAP-STE (as MO-HRAP-STE [8]). Moreover, three used objectives defined in MO-HRAP-STE are specific to the company domain: standard profit, balance defined not classically (e.g., in [5]), and priority of offers. Here, we redefine HRAP-STE as a single-objective problem, which allows us to solve and investigate problem features using classic (meta)heuristics. To the best of our knowledge, such a model does not exist in the literature. That makes HRAP-STE a novel model in the Operational Research domain. Additionally, as before the HRAP-STE model implementation, most allocation processes of the company's human resource management work are completed by hand. Thus, without HRAP-STE, the company's efficiency in human resource management is not optimal and is very susceptible to human errors.

The **main motivations** behind this paper are as follows. First, we define and investigate the proposed real-world HRAP-STE problem, which is very important in the practice of short-term employment sector companies like EWL Group company. HRAP-STE is defined as a discrete combinatorial

single-objective optimization problem. Second, we introduce 16 benchmark dataset that contains a real and artificial instances to support research. Finally, the effectiveness of six well-known (meta)heuristics, known as effective in HRAP, is empirically verified and compared in application to HRAP-STE.

The rest of the paper is organized as follows. In Sec.II, a short related work is given. The investigated HRAP-STE problem is defined in Sec.III. An investigated (meta)heuristics are given in Sec.IV. Sec. V includes results of experiments for six (meta)heuristics applications to the proposed HRAP-STE and lastly, the paper is concluded in Sec.VI.

II. RELATED WORK

HRAP is a group of optimization problems known as NP-hard [5] arising from practice. Frequently, HRAP is a variation of Resource Assignment Problem (RAP) [1], where the goal is to assign tasks to machines to optimize a quality measure (or a set of them) to satisfy all constraints. The proposed HRAP-STE is a specific type of HRAP that is a real-world problem defined in cooperation with EWL GROUP. According to our knowledge, in the literature there is no such HRE-STE model, but there are works related to ours.

In the survey [5] an exhaustive taxonomy of HRAP definitions is presented. It consists of single- and multi-objective optimization problems, several assignment problem variations with qualifications, bottleneck assignment, categorized assigned, etc. The heuristics and metaheuristics are used as effective HRAP resolution methods, such as Genetic Algorithm, Particle Swarm Optimization, Tabu Search, etc. In work [5] the very large spectrum of HRAP applications are given, e.g., production management, health care systems, project management etc.

The application of the Particle Swarm Optimization (PSO) metaheuristic to enterprise HRAP is presented in [3]. The proposed approach uses three measures: the functions of society, the economy, and the environment. The effectiveness of Genetic Algorithm (GA) in solving HRAP-based in Software Project Management was investigated in [4]. The proposed approach uses four measures (like cost, concentration efficiency, and concentration and balance of allocation) implemented as a fitness function that consists of the weighted sum of four measures.

Classic HRAP problems may consider many different solution quality metrics. However, their main feature is a direct mapping of a given employee to production tasks [2] (that may also be denoted as projects [3] depending on the considered type of industry). In general, a direct *resource to task* or *task to resource* allocation is typical for many resource assignment problems [1] also when the considered resource is other than employees [5], [6]. In some cases, instead of a direct resource-to-task assignment, the solution-building algorithm may be used. Then, the solution is frequently encoded as the order in which tasks or resources are greedily assigned to each other [7].

As presented in the next section, the nature of HRAP-STE is different. We do not assign an employee (a resource) to a project directly because we do not know how many employees will be available. Thus, a solution to HRAP-STE shall be considered as a plan of profit maximization in the assumed situation in which the amount of resources is uncertain and almost certainly insufficient. Therefore, our objective is to create a resource allocation plan based on the amount of resources available at a given moment.

Finally, three metrics defined in strict cooperation in HRAP-STE are specific to the company domain: standard profit, balance not defined classically (like [5] – as), and priority of offers. To the best of our knowledge, such a model is not presented in the domain literature, which makes HRAP-STE a novel model in the Operational Research domain.

III. HUMAN RESOURCE ALLOCATION PROBLEM IN SHORT-TERM EMPLOYMENT SECTOR – A FORMULATION

To define HRAP-STE, several variables should be given, see Tab.I. There are given a *job offers* set, where each job offer o is defined by a number of employees to recruit o^{cap} and already recruited o^r . All available positions in the i th job offer are $AvPos(o_i) = o_i^{cap} - o_i^r$. Where a *slot* defines a single position available in a job offer. It is assumed that each job offer may have many slots, but a single slot is a part of one job offer. For each assigned job offer $Offer(s)$ there are two measures included in the model: a *Profit*(o) and *Priority*(o) connected to the job offer o (and the s slot). The *Profit* is gained by recruiting a single employee for the job offer o , and the *Priority* of this job offer is defined as $Profit(Offer(s))$ and $Priority(Offer(s))$, respectively. However, none of the clients (the job offerer) can be ignored in the assignments, so the jobs should be offered *proportionally* to keep *Balance* in the company portfolio.

The main goal of HRAP-STE is to satisfy $O = 1, \dots, o_{max}$ job offers that have $n = \sum_{i=1}^{o_{max}} (o_i^{cap} - o_i^r)$ available slots with all available candidates $C = 1, \dots, c_{max}$. However, the practice of short-term employment companies shows that frequently $n > c_{max}$ or $n \gg c_{max}$ – it means that it is impossible to assign employees to *all* available slots. Furthermore, in general c_{max} is not known in advance. Thus, to cover that we define **three quality measures** as *Profit*, *Priority* and *Balance* – it allows the foreplanning that is later used while making the decisions.

Let $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ be a slot-processing sequence where π_1 and π_n indicate the first and the last slot to be processed. For instance, if there are three employees available, then the expected profit of their recruiting will be $Profit(Offer(\pi_1)) + Profit(Offer(\pi_2)) + Profit(Offer(\pi_3))$. Since we do not know the number of available employees in advance, then we define the measure

TABLE I
HRAP-STE – VARIABLES AND NOTATIONS (BASED ON [8])

notation	variables
o	The job offer
o^{cap}	The number of employers to recruit
o^r	The number of employers already recruited
s	The single available position, as <i>slot</i>
$Offer(s)$	The job offer assignment
$Profit(Offer(s))$	The profit of recruited slot s in o job offer
$Priority(Offer(s))$	The priority of o job offer, so called: project
c_{max}	The number of available candidates, <i>not known</i> in advance
n	The number of available job positions, where $n >> c_{max}$
$\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$	The slots-processing sequence, where π_1 is the first and π_n the last slot
$max PlanProfit(\pi)$	The summarized profit of all job assignments
$max PlanPriority(\pi)$	The summarized priority of all job assignments (and projects)
$max PlanBalance(\pi)$	The balance measure that covers job offers (projects) distribution

of expected profit as:

$$PlanProfit(\pi) = \sum_{i=1}^n g(i, \pi) \quad (1)$$

$$g(i, \pi) = \sum_{j=1}^i Profit(Offer(\pi_j))$$

We define the *Priority* referring to job offer priorities as:

$$PlanPriority(\pi) = \sum_{i=1}^n p(i, \pi) \quad (2)$$

$$p(i, \pi) = \sum_{j=1}^i Priority(Offer(\pi_j))$$

In a single-objective optimization, it is easy to find the optimal solution only for these two measures simply by ordering the most profitable or prioritized slots first. However, if these two measures contradict, e.g., the slots with low profit have high priority, then the weighted single-objective problem made from these two measures becomes hard to solve.

However, companies that coordinate the short-term employment process cannot limit their activity only to maximize the expected profit and the priority of the declared job offer. They are also expected to ensure that all job offers from the company portfolio will be assigned, in small part. So the *Balance* measure must be considered too, defined as follows:

$$PlanBalance(\pi) = \sum_{i=1}^n b(i, \pi) \quad (3)$$

$$b(i, \pi) = \min_{o \in O} (o^r(i, \pi) / o^{cap}(i, \pi))$$

where $o^r(i, \pi)$ and $o^{cap}(i, \pi)$ refer to the number of recruited employees and the overall number of employees that are to be recruited for the o^{th} offer when the first i slots in sequence π are assigned employees.

The goal of the HRAP-STE problem is to maximize the values of all three measures. The main difficulty in optimizing the above problem is that the considered objectives contradict each other, which is a typical feature of multi-objective optimization [8]. Moreover, the offers with the highest priority do not necessarily bring the highest profit. Finally, the *PlanBalance*

objective may be considered as contradicting both other objectives. To optimize *PlanBalance*, we shall always choose the slots that refer to the offer with the lowest percentage of occupied slots. Thus, optimization of *PlanBalance* will lead to equalization of the percentage of slots occupied for all i in Eq.3. In the paper we investigate the problem, solution landscape and instances, thus the simplified version of HRAP-STE is considered – as defined in Eq.4 by weighted sum of the objectives.

$$f(\pi) = w_1 * PlanProfit(\pi) + w_2 * PlanPriority(\pi) + w_3 * PlanBalance(\pi) \quad (4)$$

In Eq.4 the three weight values $w_1, w_2, w_3 \in < 0.0, 1.0 >$ that define the 'importance' of the selected objectivity. In this paper, all weights are equal to 1.0 for investigations. Moreover, to avoid the domination of some objectivity, all values for the objectives are normalized.

A. Solution encoding example

HRAP-STE solutions are encoded using permutation (π). Each value of the permutation refers to a given slot. Let us consider a HRAP-STE instance with two jobs o_1 and o_2 , each with two slots. The slot profit is $Profit(o_1) = 10$ and $Profit(o_2) = 5$, while priority is $Priority(o_1) = 2$ and $Priority(o_2) = 1$.

The first two values in the permutation refer to o_1 and the latter two to o_2 . We consider solution $\pi^a = [4, 1, 2, 3]$.

$$[Offer(\pi_1), Offer(\pi_2), Offer(\pi_3), Offer(\pi_4)] =$$

$$[Offer(4), Offer(1), Offer(2), Offer(3)] = [o_2, o_1, o_1, o_2].$$

Thus, the quality measures' values will be:

$$PlanProfit(\pi) = \sum_{i=1}^n g(i, \pi) = \sum_{j=1}^1 Profit(Offer(\pi_j)) + \dots + \sum_{j=1}^4 = 5 + (5 + 10) + (5 + 10 + 10) + (5 + 10 + 10 + 5) = 75$$

$$PlanPriority(\pi) = \sum_{i=1}^n p(i, \pi) = \sum_{j=1}^i Priority(Offer(\pi_j)) + \dots + \sum_{j=1}^4 Priority(Offer(\pi_j)) = 1 + (1 + 2) + (1 + 2 + 2) + (1 + 2 + 2 + 1) = 15$$

$$\begin{aligned}
PlanBalance(\pi) &= \sum_{i=1}^n b(i, \pi) = \\
&\min_{o \in O} (o^r(1, \pi) / o^{cap}(1, \pi)) + \dots + \\
&\min_{o \in O} (o^r(4, \pi) / o^{cap}(4, \pi)) = \\
&\min(\frac{1}{2}, 0) + \min(\frac{1}{2}, 0) + \min(\frac{1}{2}, \frac{1}{2}) + \dots + \min(1, 1) = \\
&0 + (0 + \frac{1}{2}) + (0 + \frac{1}{2} + \frac{1}{2}) + (0 + \frac{1}{2} + \frac{1}{2} + 1) = 3\frac{1}{2}
\end{aligned}$$

IV. METHODS

Six well-known **methods** have been investigated in solving an HRAP-STE problem to give a complementary case study: 3 heuristics and 3 metaheuristics that are effective in solving HRAP [5]. The reference method *RandomSearch* is used, and as its improved version *RandomLocalSearch*. In addition, as a reference, the classic algorithm *Greedy* has been used. In experiments, well-known metaheuristics presented in HRAP survey [5], such as classic *GeneticAlgorithm*, *ParticleSwarmOptimisation* (PSO) [3], and *TabuSearch*, have been used to verify their effectiveness in solving HRAP-STE.

The proposed encoding (see sec.III-A) for HRAP-STE is defined as a permutation, so a classic permutation-based **operators** could be applied. We investigated crossover operators for *GeneticAlgorithm* such as *Cycle Crossover* and *Partially Matched Crossover* (PMX). As mutation *Swap*, *Inversion* and *Insert* is investigated. In addition, mutation operators were investigated as the neighborhood generator for *TabuSearch* and heuristics.

V. EXPERIMENTS

The main goal of developed experiments is to investigate the effectiveness of well-known (meta)heuristics, applied to different HRAP-STE scenarios (instances, see sec.V-A). For each investigated method, the best-found configuration and experimental procedure are set (sec.V-B). Finally, the results of the conducted experiments are presented in sec.V-C.

A. Instances

In experiments, a set of benchmark HRAP-STE¹ real and artificial instances are used. Artificial instances are split into 9 *easy* and 3 *hard* one's scenarios.

All HRAP-STE instances presented see Tab.V-A have varying number of job offers, slots, and profits/priorities to define the problem instances. Furthermore, 4 real instances were prepared to show the influence of constraints on real scenario difficulty. Such instances include anonymized EWL GROUP company data from about a month, containing 39-99 job offers and 2-67 slots each offer.

The 10 collected features of HRAP-STE instances, i.e. number of slots, jobs, and priorities with their statistics (min,avg,max) allow to make a PCA analysis of HRAP-STE instances landscape and visualization. The graph in Fig.1 shows that *easy* instances are near, except *easy8*. The *hard* instances are also in near localization. The most surprising is the long distance for *real* instances, which could be interpreted as they model different situations in the company. Additionally, a long distance from *easy* to *real* instances showed that

¹All used HRAP-STE instances and gained results are published in .

TABLE II
HRAP-STE INSTANCES

name	jobs	slots	priorities	profits
easy1	2	3	1-3	30-60
easy2	2	2-3	1-3	45-60
easy3	3	2-5	1-3	20-105
easy4	3	2-3	2-3	10-25
easy5	3	2-4	2-4	10-25
easy6	6	1-3	1-3	60-400
easy7	3	2-7	1-4	15-115
easy8	3	3	1-8	25-1595
easy9	5	2	1-2	20
hard1	9	15	1-5	20-1810
hard2	14	10-24	1-4	10-115
hard3	60	2-9	1-4	10-100
real1	43	2-25	0-3	1.4-2.08
real2	39	2-14	0-3	1.4-2.08
real3	43	2-14	0-3	1.58-3.61
real4	99	2-67	0-3	1.28-3.61

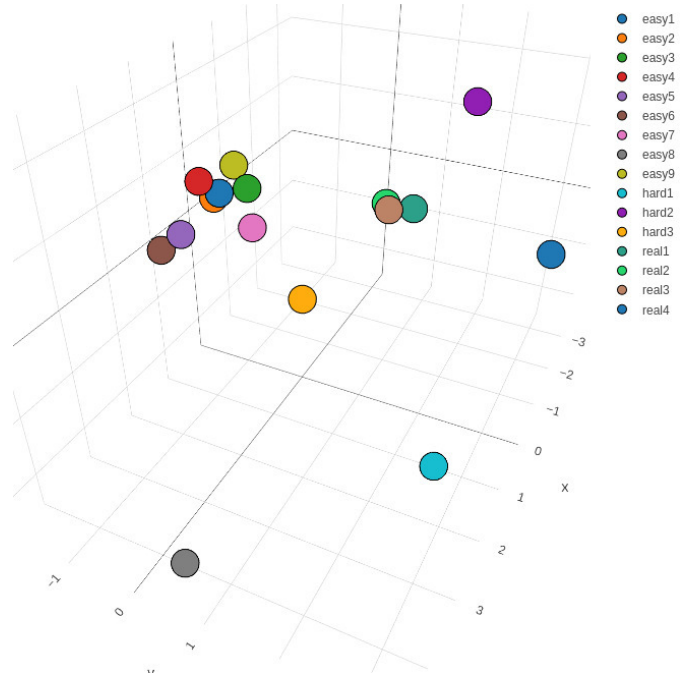


Fig. 1. PCA analysis of HRAP-STE instances

easy are only testing instances that are literally far away from *real* cases.

B. Experimental setup and procedure

We consider a relatively large **computation budgets** (500 000 fitness function evaluations, FFE) for all instances and methods to eliminate the situation in which the best method is simply the one that is the fastest to converge. For each investigated method, the tuning procedure have been run to find best-found configuration – presented in Tab. (see Tab.V-B).

No parameters are given for heuristics like the *Greedy* algorithm, *RandomSearch*, and *RandomLocalSearch*, as the neighborhood operator is defined as *insert*. For metaheuristics, specific parameters should be set. For *GeneticAlgorithm*

TABLE III
THE BEST-FOUND CONFIGURATIONS

method	configuration
RandSearch	-
RandLocalSearch	operator=swap
Greedy	operator=swap
PSO	$c_1 = 0.5$ $c_2 = 0.05$ $psize = 100$ $w = 0.9$
GeneticAlgorithm	$P_{pmx}^x = 0.0$ $P^m = 0.001$ $psize = 200$ $T_{size} = 3$
TabuSearch	operator=insert $tabu_{tenure} = 10$

crossover P^x and mutation P^m probability, size of tournament selection T_{size} and size of population $psize$. For TabuSearch neighborhood operator $insert$ and $tabu_{tenure}$. Finally, for the PSO size of the swarm ($psize$) and specific parameters: inertia weight w , cognitive c_1 , and social c_2 acceleration coefficient.

The experimental results have been evaluated on all HRAP-STE instances. Due to the non-deterministic nature of metaheuristics, all runs have been repeated 30 times, and results averaged. To verify the statistical significance of the presented results, the Wilcoxon signed-rank test is used with $pvalue = 0.05$.

The research environment with all investigated methods has been implemented in Rust and Python. All experiments were developed using the following configuration: 2,6 GHz 6-Core Intel Core i7, 16GB RAM, and OS: Sequoia 15.4.1.

C. Results

All results of experiments use 6 (meta)heuristics in solving 16 HRAP-STE instances – see Tab.V-C. For all *easy* instances, almost all methods gain the same results. The difference can be seen in *hard* and *real* instances, where *GeneticAlgorithm* outperforms other methods. The second place gets *RandLocalSearch*. Although the difference between the two best methods appears to be very small, the Wilcoxon Signed Rank test confirms the statistical significance: for instances *hard* instances $pvalue < 0.0001$ and for *hard* ones do not exceed $pvalue < 0.03$.

The results presented in Tab.V-C encourage a more detailed analysis of results for more difficult instances. Fig.2 presents averaged results for methods solving *hard3* instance – it shows *RandomLocalSearch* in lower budget wins, but finally, the Genetic Algorithm gets the best results. The wider context for *hard3* instance gives a boxplot diagram from Fig.4.

A similar situation occurs in *hard4* instances – see Fig.3, where the Genetic Algorithm relatively quickly, in the computation budget context, outperforms other methods. The boxplot presented in Fig.5 confirms that the Genetic Algorithm for *real4* instance is very competitive.

The budget defined by FFE is useful for comparing results for (meta)heuristics. However, some of them have specific operations unrelated to FFE. In such situations, the computational time for investigated methods could be compared. For *easy* instances, *Geneticalgorithm* needs 13-31 seconds, whereas other methods gain results in less than 1 second. For *hard* instances, *GeneticAlgorithm* works within 26-54

seconds, while *PSO* needs 11-251 seconds and *TS* 24-62 seconds respectively. A similar situation occurs in *real* instances, but there is an exception for *real3* instances, where *GeneticAlgorithm* needs 217 seconds and *TS* 954 seconds. Such differences in computation times for various instances are strictly connected to the 'size' of instances, i.e., how large is the solution landscape.

D. Summary

Experiments presented in previous sections showed that HRAP-SA can be effectively solved by both heuristic and metaheuristic. Heuristic *RandomLocalSearch* is very competitive for a low computational budget, especially for easier instances. However, metaheuristics (like *GeneticAlgorithm*) usage is recommended when efficiency is needed more.

VI. CONCLUSIONS AND FUTURE WORK

In this work, HRAP-STE is defined as a real-world problem that extends the HRAP problem, known as NP-hard. The proposed HRAP-STE definition also consists of representation and fitness function. To evaluate the wider context of HRAP-STE, the 16 benchmark instances that include artificial and real scenarios are proposed. That allows to give a complementary case study, and evaluate the effectiveness of six well-known methods of solving an HRAP-STE problem: 3 heuristics and 3 metaheuristics.

The experiments presented in this paper showed that metaheuristics are effective HRAP-STE solvers. In lower budgets, heuristics are more effective; for larger budgets, metaheuristics outperform other methods. Such dualism encourages linking methods from these groups and defining hybridization – one of the most successful and promising research field in optimization [5]. Additionally, the representation and operators used to solve HRAP-STE in this paper are not specialized. Thus, a further research direction could be pointed out to include *domain knowledge* in new representations and operators. Last but not least, HRAP-STE could be defined as a multi-objective problem, which is considered in [8] – in this context, more extensive research connected to the specialized representations and operators could build a more effective tool for the decision-makers in the company.

ACKNOWLEDGEMENT

This work was supported by The National Center for Research and Development (NCBiR), Poland under Grant POIR.01.01.01-00-1042/20.

Special thanks to EWL S.A. company team, especially for Andrzej Korkus (CEO) and Adam Korkus (CTO).

REFERENCES

- [1] Grillo, H., Alemany, M. & Caldwell-Marin, E. Human Resource Allocation Problem in the Industry 4.0: A Reference framework. *Computers & Industrial Engineering*. **169** pp. 108110 (2022,3)
- [2] Lin, C. & Gen, M. Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm. *Expert Systems With Applications*. **34**, 2480-2490 (2008)

TABLE IV
RESULTS COMPARISON FOR 30 INDEPENDENT RUNS

	GeneticAlgorithm		Greedy		PSO		RandSearch		RandLocalSearch		TabuSearch	
	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
easy1	0.4497	0.0000	0.4497	0.0000	0.4497	0.0000	0.4497	0.0000	0.4497	0.0000	0.4497	0.0000
easy2	0.5778	0.0000	0.5778	0.0000	0.5778	0.0000	0.5778	0.0000	0.5778	0.0000	0.5778	0.0000
easy3	0.3108	0.0000	0.3108	0.0000	0.3108	0.0000	0.3108	0.0000	0.3108	0.0000	0.3108	0.0000
easy4	0.3939	0.0000	0.3939	0.0000	0.3939	0.0000	0.3939	0.0000	0.3939	0.0000	0.3939	0.0000
easy5	0.4354	0.0000	0.4354	0.0000	0.4354	0.0000	0.4354	0.0000	0.4354	0.0000	0.4354	0.0000
easy6	0.3603	0.0000	0.3555	0.0075	0.3603	0.0000	0.3602	0.0001	0.3549	0.0077	0.3603	0.0000
easy7	0.2554	0.0000	0.2550	0.0009	0.2554	0.0000	0.2554	0.0000	0.2550	0.0009	0.2554	0.0000
easy8	0.4099	0.0000	0.4099	0.0000	0.4099	0.0000	0.4099	0.0000	0.4099	0.0000	0.4099	0.0000
easy9	0.4606	0.0000	0.4606	0.0000	0.4606	0.0000	0.4606	0.0000	0.4606	0.0000	0.4606	0.0000
hard1	0.4120	0.0001	0.4121	0.0001	0.4099	0.0005	0.3983	0.0008	0.4120	0.0001	0.4121	0.0000
hard2	0.1846	0.0004	0.1829	0.0007	0.1765	0.0015	0.1614	0.0011	0.1842	0.0004	0.1707	0.0019
hard3	0.3936	0.0006	0.3837	0.0040	0.3612	0.0029	0.3192	0.0023	0.3918	0.0010	0.3364	0.0056
real1	0.3191	0.0011	0.3183	0.0020	0.3003	0.0026	0.2691	0.0021	0.3174	0.0011	0.2913	0.0059
real2	0.1846	0.0004	0.1829	0.0007	0.1765	0.0015	0.1614	0.0011	0.1842	0.0004	0.1707	0.0019
real3	0.2261	0.0019	0.2223	0.0021	0.2110	0.0023	0.1812	0.0017	0.2227	0.0018	0.2066	0.0041
real4	0.2905	0.0030	0.2176	0.0029	0.2678	0.0033	0.2390	0.0014	0.2921	0.0025	0.2213	0.006

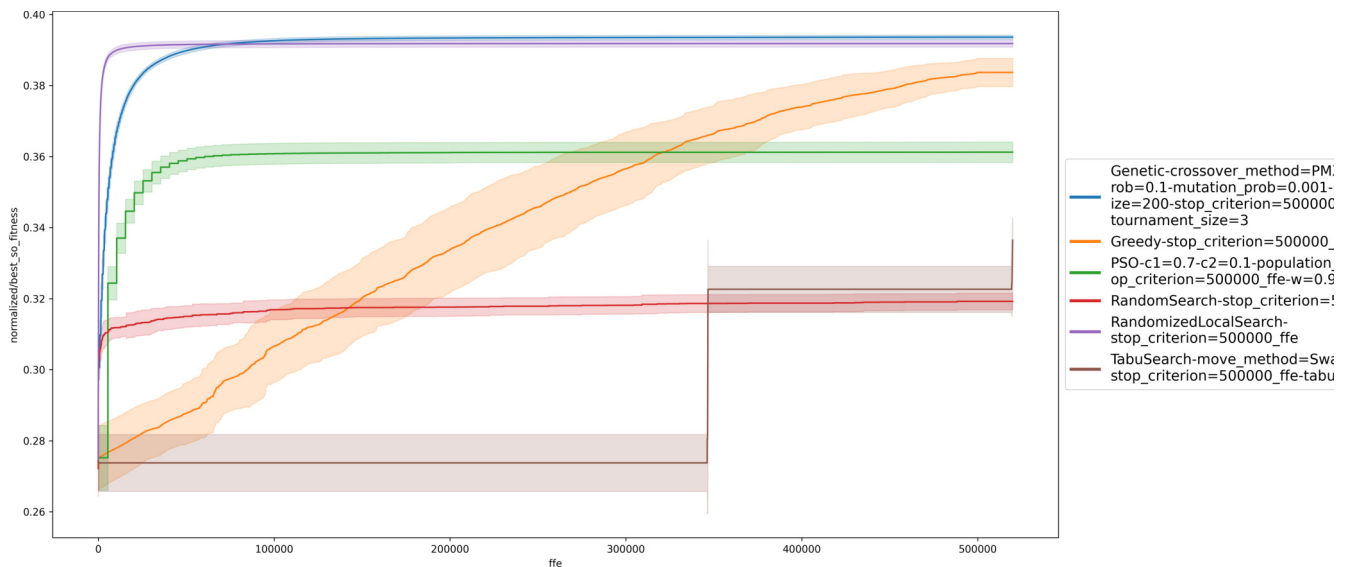
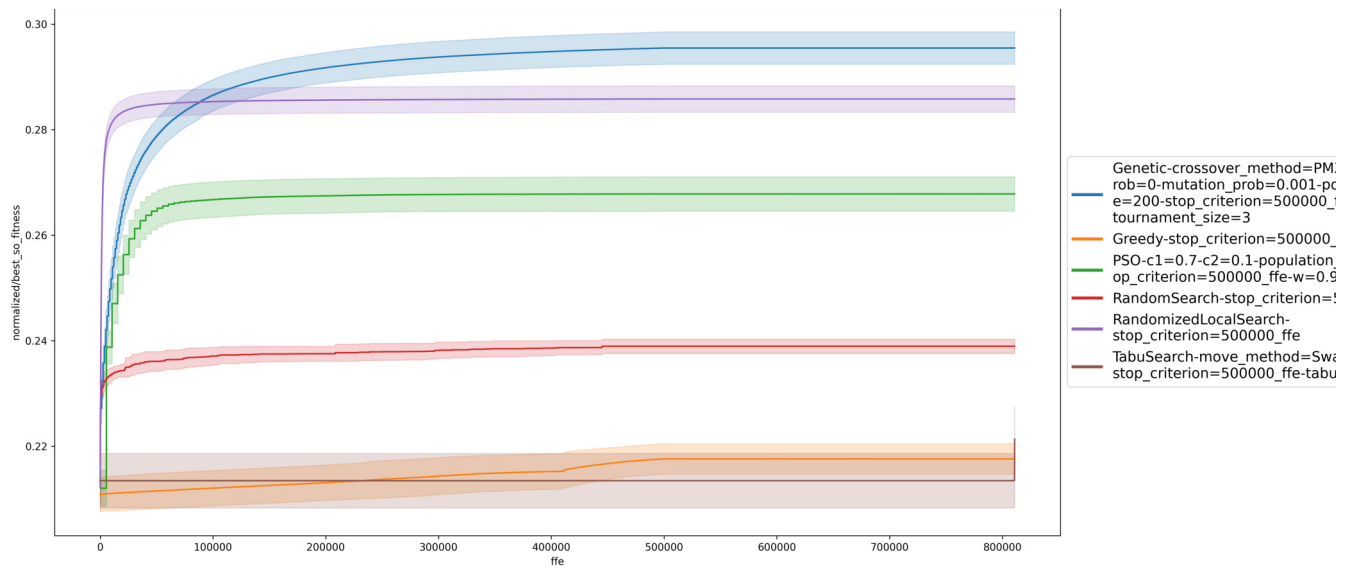
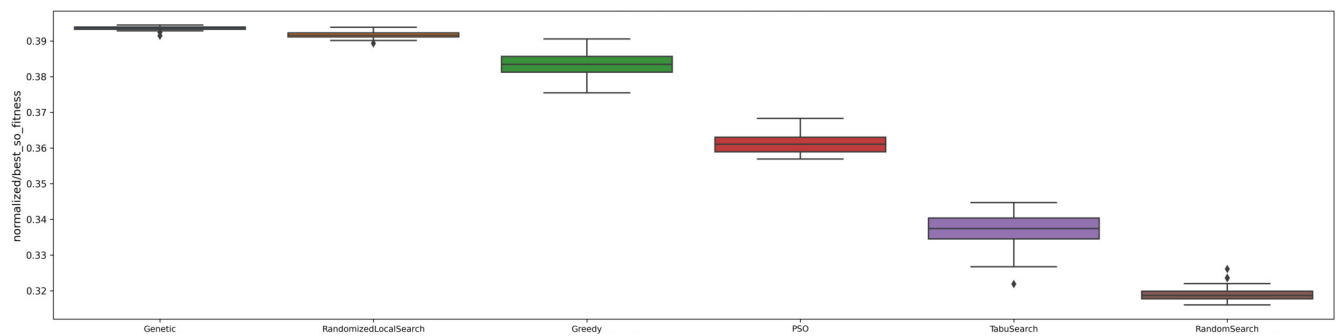
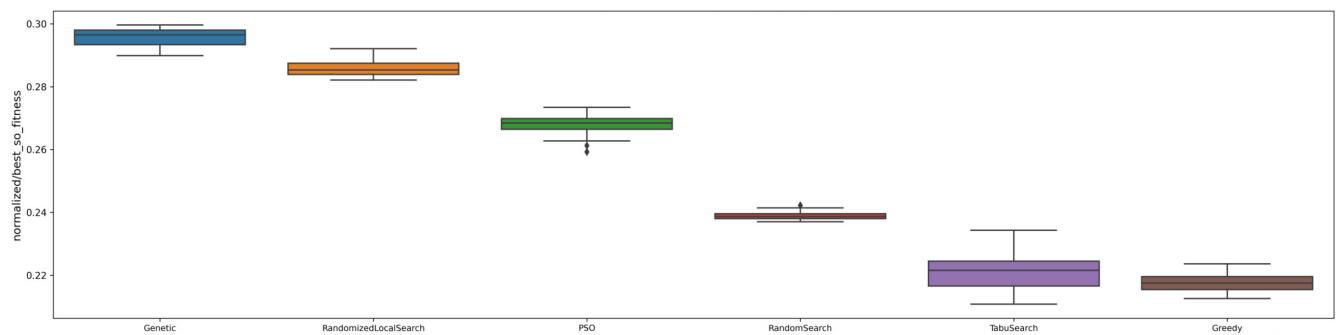


Fig. 2. Computational results for *hard3* instance.

- [3] Wang, Z. Enterprise Human Resource Allocation Optimization Model Based on Improved Particle Swarm Optimization Algorithm. *Wireless Communications And Mobile Computing*. **2022** pp. 1-9 (2022,3)
- [4] Park, J., Seo, D., Hong, G., Shin, D., Hwa, J. & Bae, D. Human Resource Allocation in Software Project with Practical Considerations. *International Journal Of Software Engineering And Knowledge Engineering*. **25** pp. 5-26 (2015,2)
- [5] Bouajaja, S. & Dridi, N. A survey on human resource allocation problem and its applications. *Operational Research*. **17** pp. 339-369 (2017)
- [6] Huynh, N. & Chien, C. A hybrid multi-subpopulation genetic algorithm for textile batch dyeing scheduling and an empirical study. *Computers & Industrial Engineering*. **125** pp. 615-627 (2018), <https://www.sciencedirect.com/science/article/pii/S0360835218300068>
- [7] Taillard, E. Benchmarks for basic scheduling problems. *European Journal Of Operational Research*. **64**, 278-285 (1993), <https://www.sciencedirect.com/science/article/pii/037722179390182M>, Project Management and Scheduling
- [8] Przewozniczek, M.W., Myszkowski, P.B., Kosciukiewicz, W., Wojcik, M., Gonczarek, A., Korkus, A., On discovering and analysing variable dependencies to construct an effective and efficient optimiser dedicated to solving the new real-world multi-objective resource-allocation problem, **in review**, *Applied Soft Computing Journal*.

Fig. 3. Computational results for *real4* instance.Fig. 4. Results comparison for *hard3* instance.Fig. 5. Results comparison for *real4* instance.