Soft Comput (2015) 19:3599–3619
DOI 10.1007/s00500-014-1455-x

FOCUS

Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem

Paweł B. Myszkowski · Marek E. Skowroński · Łukasz P. Olech · Krzysztof Oślizło

Published online: 16 September 2014

Abstract In this paper, hybrid ant colony optimization (HAntCO) approach in solving multi-skill resource-constrained project scheduling problem (MS-RCPSP) has been presented. We have proposed hybrid approach that links classical heuristic priority rules for project scheduling with ant colony optimization (ACO). Furthermore, a novel approach for updating pheromone value has been proposed based on both the best and worst solutions stored by ants. The objective of this paper is to research the usability and robustness of ACO and its hybrids with priority rules in solving MS-RCPSP. Experiments have been performed using artificially created dataset instances based on real-world ones. We published those instances that can be used as a benchmark. Presented results show that ACO-based hybrid method is an efficient approach. More directed search process by hybrids makes this approach more stable and provides mostly better results than classical ACO.

Keywords Ant colony optimization · Project scheduling problem · Metaheuristics · Hybrid ACO · Multi-objective optimization · Benchmark dataset

Communicated by E. Lughofer.

P. B. Myszkowski (✉) · M. E. Skowroński · Ł. P. Olech · K. Oślizło
Department of Artificial Intelligence, Wroclaw University of Technology, Wrocław, Poland
e-mail: pawel.myszkowski@pwr.wroc.pl

M. E. Skowroński
e-mail: m.e.skowronski@pwr.wroc.pl

Ł. P. Olech
e-mail: 179214@student.pwr.wroc.pl

K. Oślizło
e-mail: 163753@student.pwr.wroc.pl

1 Introduction

Resource-constrained project scheduling problem (RCPSP) is one of the most investigated types of scheduling problems. Its goal is to find the resource-to-task assignments to make the finite project plan the cheapest or shortest. Description of RCPSP in Blazewicz et al. (1983) as combinatorial, NP-hard problem encouraged scientists to find *good enough* methods that would be able to produce approximate, (sub)optimal solutions in finite, polynomial computing time. Those methods are called (meta)heuristics and are used to solve problems for which finding optimal solution in an acceptable time is impossible.

Beside Evolutionary Algorithms (EA), Taboo Search (TS), Simulated Annealing (SA) and some other techniques, metaheuristics contain also a group of methods called *swarm intelligence* methods, as particle swarm optimization (PSO) or ant colony optimization (ACO). Those methods assume that separate individuals, representing given problem solutions, can interact with each other and cooperate to achieve their common goals. In this point of view, swarm intelligence techniques are similar to EA. However, they assume that there is one, constant population of individuals that can evolve in time but cannot be replaced by new individuals. ACO, as the name stands, simulates the behavior of ants, traveling between the ant's nest and the source of food. The optimization goal is to find the optimal path between food and nest, while definition of *path's quality* is varied and dependent on the considered problem.

The *real-life* nature of RCPSP comes from business. Project managers in companies struggle to build effective project schedule, meeting duration, cost and other constraints. What is more, many constraints have to be satisfied, while manual scheduling often leads to violating of those constraints. It is a common problem for project managers. Hence, computer-aided, (semi-)automatic tools are desired by the industry. Furthermore, obtaining the project plan by computer-driven methods is less time-consuming than obtained manually.

Developing RCPSP to a more practical problem, we have introduced the skills domain, transforming it to the multi-skill RCPSP (MS-RCPSP) extension. In MS-RCPSP, resources dispose of some given pool of skills, while every task requires some skills in a given level to be performed. It means not every resource is capable of performing every task. As solution space in MS-RCPSP is more constrained, it is more difficult to build *good enough* solution—project schedule. Furthermore, we have added another criterion—project schedule performance cost, transforming the classical single-objective (duration) RCPSP into multi-objective (duration vs. cost) MS-RCPSP.

We have decided to create hybrid methods by combining ACO-based approach with some heuristics described in Skowroński et al. (2013b). Therefore, classical heuristics have been also investigated. Based on results obtained in that paper, we have chosen given heuristics that could be used to obtain the initial solution for ACO mechanism and stand as a hybrid ant colony optimization (HAntCO). A very significant fact is that depending on optimized criterion (duration or cost), various priority rules could be used. Therefore, we are able to decide whether using HAntCO allows to get better solutions than using ACO mechanism not supported by any priority rule.

Investigating ACO-based approach was motivated by the willingness to compare results obtained using several collective intelligence methods and other metaheuristics, such as TS or SA (Myszkowski et al. 2013) to solve this problem. As we had researched EA-based approach before Skowroński et al. (2013a), we made a comparison of different approaches in case of their robustness, effectiveness and stability, while those terms would be explained further.

The dataset for experiments has been created artificially, but instances are based on the real-world ones obtained from an international enterprise. What is more, presented MS-RCPSP could be generalized to the PSPLIB (Kolisch et al. 1996) dataset model that is regarded as a benchmark for methods solving project scheduling problems.

The rest of the paper is organized as follows. Section 2 describes selected ways of solving the (MS-) RCPSP using metaheuristics, especially ACO. Section 3 presents the MS-RCPSP problem statement, while Sect. 4 describes the approaches proposed in this paper. Section 5 provides conducted experiments of proposed methods in a given dataset. Finally, Sect. 6 presents the conclusions of obtained results and suggests some directions of future work.

2 Related work

Metaheuristics are very often used to solve RCPSP because of its NP-hard nature. EA (Hartmann 1998; Valls et al. 2001, 2008), TS (Thomas et al. 1998; Tsai et al. 1998; Verhoeven 1998), SA (Bouleimen et al. 2003; Das et al. 2011) are well explored and widely applied to solve MS-RCPSP. It is worth a mention that ACO is not the only swarm intelligence metaheuristic used in solving (MS-) RCPSP. PSO approaches could be found in Tam et al. (2006), Zhang et al. (2005, 2009), while bee colony optimization (BCO) method has been investigated in Ziarati et al. (2011). Numerous papers regarding PSO or BCO in solving RCPSP prove that those methods are often investigated and researched.

However, there is still lack of papers regarding multi-objective multi-skill extension of RCPSP. Some approaches solving MS-RCPSP in project duration domain (Al-Anzi et al. 2010; Santos et al. 2011) or project cost domain (Li et al. 2009) could be found. On the other hand, there are methods solving classical RCPSP extended by cost domain but without skills considerations. Such research has been presented in Phruksaphanrat (2014), Jaberi et al. (2014), Gonzalez et al. (2013), Luna et al. (2013) and Yannibelli et al. (2013). Hence, we have decided to combine those two elements: multi-objective optimization and multi-skill domain for project scheduling problem.

Although classical RCPSP is deeply investigated and numerous approaches could be easily compared using PSPLIB instances, it is very hard to find multi-objective MS-RCPSP methods working on datasets that could be regarded as a benchmark. Some papers describe instances artificially generated (Hegazy et al. 2000; Santos et al. 2011), while some others propose methods of PSPLIB dataset adaptation (Al-Anzi et al. 2010; Drezet et al. 2008; Kadrou et al. 2006; Li et al. 2009). However, both of those approaches for handling MS-RCPSP benchmark data are not supplied by any published dataset instances. Hence, the need of proposing our own dataset has arisen.

ACO is inspired by the rules in the real environment of ants. Real ants are capable of finding the shortest path from the source of food to the ant's nest. Every ant from a population leaves a substance called pheromone while getting to the source of food. This substance attracts other ants to come into that direction. However, the pheromone evaporates gradually in every period. It means the shorter path is, the less pheromone would be evaporated and that path would be more attractive to other ants. In that way, more and more ants start to exploit the region of a surface where there was more pheromone—-the path to the source of food was shorter. Finally, all ants move along the same path, what is regarded as the found solution of the problem.

A classical ACO approach with some modifications that made it more robust has been presented in Merkle et al.

(2002). Particularly, the following features have been proposed: combination of two pheromone updating methods, dynamic influence of those methods during ACO runtime and possibility of leaving the best obtained solution by an elitist ant to preserve sticking in local optima. The presented methods have been tested on PSPLIB instances. In many cases, the obtained results were better than the best found so far, what confirms the robustness of that approach.

Various improvements of ACO have been proposed in Luo et al. (2003). A single solution, represented by a single ant, is obtained using serial generation scheme. If generated schedule turns out to be infeasible after adding a given task, the ant can reschedule some beginning fragments of a current schedule to make it feasible. The feasibility is lost when precedence constraints are violated. The following activities that should be added to a current schedule are chosen by combination of classical heuristics: most total successors, latest finish time (LFT) and resource scheduling method. The authors used UBO dataset from ProGen (Kolisch et al. 1996) to verify their approach.

A different ACO approach has been presented in Zhou et al. (2009) as well. The combination of ant colony system (Dorigo 1997) and Max–Min ant system (Stutzle et al. 2000) called MMACS has been proposed. The following improvements have been proposed in this approach: pseudo-random proportional rule for choosing a next activity, updating pheromone only in the base of the best ant from given iteration and serial schedule generation scheme. Furthermore, an extended and RCPSP-adjusted 2opt local search method (Watson et al. 1998) called PS-2opt has been proposed. Results of experiments conducted on PSPLIB stated that PS-2opt and MMACS methods are robust in solving RCPSP.

Another ACO-based approach has been presented in Liang et al. (2004) where activity-on-node task precedence relations representation is considered. Activity selection is performed by forward-parallel method, while the search space exploration and exploitation are performed by tuned online and offline pheromone updating procedure. Conclusions supported by performed experiments on PSPLIB datasets stand that the approach proposed in Liang et al. (2004) gives competitive results in comparison to other (not only) ACO-based approaches.

3 Problem statement

Before the description of the multi-skill extension for RCPSP, the fundamentals of classical RCPSP would be presented. The motivation to investigate RCPSP and its extensions came from industry and would be explained in detail in Sect. 3.3.

3.1 Classical RCPSP description

In RCPSP, a set of tasks is given, while every task is described by its duration, start and finish dates. Tasks are non-preemptive. It means any task cannot be withdrawn if it has been started. Tasks are related to each other by precedence relations, describing which tasks are needed to be completed before some others could be started. Tasks that have to be finished before the start time of another task are called predecessors. In classical RCPSP, resource units are provided. Every resource owns a finite number of units (represented as integer numbers) that could be assigned to various tasks, while tasks require some number of units to be performed. Cumulative number of units of tasks assigned to specified resource in a given period cannot exceed a number of units owned by resource. Not only one resource can be assigned to a given task but also one task can be assigned to given resource in given timestamp. In classical RCPSP, two dummy activities are added: start and finish tasks. It is because, in RCPSP, every task besides the start one has predecessors. Hence, finish time of the last, dummy finish task is the finish time of schedule and the duration of a project could be computed as duration between start time of dummy start task and finish time of finish dummy task. The goal of RCPSP is to find such task-to-resource assignments to make the final schedule feasible and as shortest as possible. Combinatorial nature of the RCPSP makes it NP-hard.

A solution of RCPSP is a feasible schedule—the one in which resource units and precedence constraints are preserved.

3.2 Multi-skill extension of RCPSP

MS-RCPSP extension adds the skills domain to classical RCPSP. Every task requires some skills at given familiarity level to be performed, while every resource disposes some skills pool—subset of skill types (e.g., developer, analyst, tester, architect, etc.) defined in a project with given familiarity level. Therefore, the resource R is capable of performing the task T only if R disposes skill required by T at the same or higher level. The capabilities of performing tasks by resources could be presented as skill matrix. Sample skill matrix is shown in the Fig. 1.

In the skill matrix presented in the Fig. 1, skills required by task to be performed have been written over task definition, while skills owned by resources have been written next to resource definition. This figure presents sample resource capabilities: resource $R1$ disposes skills $Q1$ and $Q2$ with familiarity levels 3 and 2, respectively. It is capable of performing tasks $T1$, $T3$ and $T4$ because all of those mentioned tasks require skill owned by $R1$ at no higher level than it has. $R1$ cannot be assigned to $T2$, because this task requires totally different skill that $R1$ does not dispose of, even at the

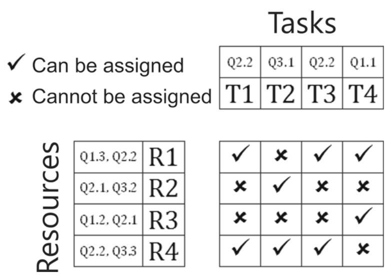

Fig. 1 Example of skill matrix

lowest familiarity level. Analogously, resource $R2$ can be assigned to task $T2$, resource $R3$ is a proper one for task $T3$ and, finally, resource $R4$ can perform tasks $T1$, $T2$ and $T3$. Even though $R3$ disposes of skill $Q2$, it cannot be assigned to $T1$ and $T3$ because those tasks require $Q2$ at higher familiarity level that this resource disposes.

3.3 Model adjustment

As a result of consultations with representatives of various enterprises, we decided to introduce some practical changes in classical RCPSP extended to MS-RCPSP model. Firstly, we introduced resource salary (as an hourly wage) paid for performed work. In that case, resources are regarded only as human ones varied by their salary. We also resigned from introducing start and finish dummy activities as our approach assumes that there could be some tasks that are not connected by precedence relations with any other. Hence, we cannot define the project duration, start time and finish time based on dummy activities.

What is more, resources are not described by units—any resource cannot be assigned to more than one task in an overlapping period—dedicated resources (Bianco et al. 1998). If such a situation occurs, the **conflict** is detected and should be resolved. The conflict fixing procedure is presented in Sect. 4.4. Schedule feasibility for such modified problem is extended from classical RCPSP schedule feasibility definition by skills domain—only resources capable of performing given tasks can be assigned to them.

3.4 Problem formulation

Feasible Project Schedule (PS) consists of $J = 1, \ldots, n$ tasks and $K = 1, \ldots, m$ resources. A non pre-emptive duration d_j, start time S_j and finish time F_j is defined for each task. Predecessors of given task j are defined as P_j. Each resource is defined by its hourly rate salary s_k and owned skills $Q^k = 1, \ldots, r$, while pool of owned skills is a subset of all skills defined in project $Q^k \in Q$. Value l_q denotes the level of given skill, while h_q describes its type and q_j is a skill required by j to be performed. Therefore, by J^k subset of tasks that can be performed by k resource is defined. Duration of a project schedule is denoted as τ. Cost of performing j task by k resource is denoted as $c_j^k = d_j * s_k$, where s_k describes the salary of resource k assigned to j. For simplicity, we have modified the task's performance cost from c_j^k to c_j, because only one resource can be assigned to given task. Hence, there is no need to distinguish various costs for the same task. Moreover, we have introduced variable defining whether k is assigned to j in given time t: $U_{j,k}^t \in \{0; 1\}$. If $U_{j,k}^t = 1$, k is assigned to j in t. Analogously, k is not assigned to j in t if $U_{j,k}^t = 0$.

Feasible project schedule (PS) belongs to the set of all feasible and non-feasible solutions (violating precedence, resource and skills constraints) : $\text{PS} \in \text{PS}_{\text{all}}$.

Formally, the problem could be regarded as optimization (minimization) problem and stated as follows:

$$\min f(\text{PS}) = \min\left[f_\tau(\text{PS}), f_C(\text{PS})\right] \tag{1}$$

Subject to:

$$\forall_{k\in K}\, s_k \geq 0, \forall_{k\in K}\, Q^k \neq \emptyset \tag{2}$$

$$\forall_{j\in J}\, F_j \geq 0; \forall_{j\in J}\, d_j \geq 0 \tag{3}$$

$$\forall_{j\in J, j\neq 1, i\in P_j}\, F_i \leq F_j - d_j \tag{4}$$

$$\forall_{i\in J^k}\, \exists_{q\in Q^k}\, h_q = h_{q_i} \wedge l_q \geq l_{q_i} \tag{5}$$

$$\forall_{k\in K} \forall_{t\in\tau} \sum_{i=1}^{n} U_{i,k}^t \leq 1 \tag{6}$$

$$\forall_{j\in J} \exists_{!t\in\tau, !k\in K}\, U_{j,k}^t = 1 \tag{7}$$

Equation 1 denotes the duration and cost optimization, respectively. Depending on the evaluation function configuration (described below), various optimization modes could be used in an optimization process. $f_\tau(\text{PS})$ is an evaluation function of project schedule's duration, while $f_C(\text{PS})$ is an evaluation function of project schedule's performance cost.

The first constraint (Eq. 2) preserves the positive values of resource salaries and ability to perform at least one task by every resource. Equation 3 states that every task has positive finish date and duration, while Eq. 4 shows the precedence constrains rule. Next two equations: Eq. 5 introduces skill constraints and transforms RCPSP into MS-RCPSP. Constraint (Eq. 6) describes that any resource can be assigned to no more than one task in given time during the project. The last constraint (Eq. 7) says that each task must be performed in schedule PS by one resource assignment.

3.5 Evaluation function

As it was mentioned, the proposed approach allows to set various objectives of optimization: duration- or cost-oriented one. Those two aspects are normalized, weighted and summarized. Normalization is necessary because of different domains of both aspects that are in opposition to each other. Setting optimization as more, cost-oriented causes enlarging the project duration; while setting as more important, the duration aspect of optimization could increase the cost of the project.

The detailed formulation of the evaluation function is presented in Sect. 4.2.

3.6 Solution space size

Because of NP-hard (combinatorial) nature of investigated problem, we have decided to present an estimation of solution space size (SS). It has been computed as follows:

$$\mathrm{SS}(n, m) = n! * m^n \tag{8}$$

The above estimation is valid for all solutions, including non-feasible ones. Computing factorial of tasks number provides the number of combinations of ordering tasks within the timeline. It is easy to notice that such estimation allows to set any order, skipping precedence constraints. The second element of Eq. 8 provides the number of resource-to-task assignments, including a situation that the same resource is assigned to all tasks and no skill constraints are preserved.

To imagine how big the solution space could be, let's take into account a sample project schedule with 100 tasks and 20 resources. Using Eq. 8, the solution space size is equal to $\mathrm{SS}(100, 20) = 1.19 * 10^{288}$ solutions, including both feasible and infeasible ones.

4 Proposed approach

Before we describe the details of the proposed approach, some basic ACO definitions in terms of MS-RCPSP should be introduced. **Colony** is represented as a set of ants: $A = 1, \ldots, p$, where p is a number of ants in population. **Edge** represents a given task and resources that are capable of performing it. Furthermore, edge stores information about the pheromone ($\mathrm{Ph}_j = 1, \ldots, p_j^k$) values for each resource capable of performing a given task. **Surface** is represented as a set of edges: $E = 1, \ldots, j$—all possible task-to-resource assignments, while **path** represents the set of specified task-to-resource assignments. Path is assigned to a given ant that represents a single solution. Surface represents the solution space of skill-feasible solutions.

The pheromone value determines the probability of assigning given resource to given task. In the first step of classical ACO, the initial value of pheromone is given for each resource in every edge; while for a heuristic initialization, pheromone value is the biggest for path reflecting solution found by heuristic. It means that, at the beginning of our approach run, the probability of choosing resource to be assigned to a task is equal in classical ACO or is close to 1 for path representing heuristic found solution and close to 0 for remaining edges in the surface.

Firstly, we have used heuristics from Skowroński et al. (2013b) to find the best approach for duration optimization (DO) and cost optimization (CO) modes. Based on the obtained results, successors list that size-based heuristic (SLS) (Skowroński et al. 2013b) with descending order has been used for DO and resource salary-based (RS) (Skowroński et al. 2013b) with ascending order has been used for CO. Output of scheduling project instances by those heuristics has been used as input for ACO method that has been run with the same parameters' configuration as ACO not boosted by heuristic.

The proposed hybrid ACO-based approach could be briefly described in the following steps:

1. Set initial ant population using heuristics to find *good* initial solution
2. Check the stopping condition.
3. Select edge for each ant.
4. Evaluate solutions.
5. Evaporate given amount of pheromone from each edge.
6. Update solutions.
7. Update pheromone value in edges by selected ants.
8. Return to 2.

The pseudocode of investigated HAntCO approach is presented in Algorithm 1.

Algorithm 1 HAntCO pseudocode

1: $A \leftarrow set\ initial\ solution$
2: **while** $stopping\ criterion\ not\ satisfied$ **do**
3: **for** $a \in A$ **do**
4: **for** $e \in Path(a)$ **do**
5: $e \leftarrow selectEdge(J^k)$
6: $f(a) \leftarrow evaluate(a)$
7: **for** $e \in E$ **do**
8: $p_e \leftarrow decayPheromone()$
9: $A' \leftarrow selectAnts(A)$
10: **for** $a' \in A'$ **do**
11: **for** $e \in Path(a')$ **do**
12: $p_e \leftarrow updatePheromone(e)$
13: $A_b \leftarrow getBestAnt(A)$
14: $A_w \leftarrow getWorstAnt(A)$
15: **if** $f(A_b) < f(A_g)$ **then**
16: $A_g \leftarrow A_b$
17: **if** $f(A_w) > f(A_v)$ **then**
18: $A_v \leftarrow A_w$
19: $A \leftarrow A'$
20: $return\ A_g$

In every iteration, some ants have to be selected (line 9 in Algorithm 1) to update a pheromone on their edges. The decision which ant should be chosen depends on selected pheromone update methods. There could be all ants chosen, only the local and global best or the local best and worst. Choosing ants to update a pheromone is described in detail in Sect. 4.3. After each iteration, pheromone values are updated. Then, local (A_b) and global (A_g) best solutions are updated. After each iteration, solutions in ants are ordered ascending by their evaluation function value (line 13). The first ant from the list is set as the best one (A_b) while the last one—as the worst local one. If the evaluation function value of the best local solution (A_b) is smaller (minimization problem) than evaluation function value for the best global solution (A_g), the best global solution is updated (line 15). Analogously, the global worst solution (A_v) is updated. The local worst solution (A_w) is used in DIFF pheromone update method.

4.1 HAntCO colony initialization

In the first step of classical ACO, the surface of n edges is obtained. For each resource in each edge, the initial pheromone value is set. Then, p ants are defined by choosing random capable resource to j task. To reduce the influence of non-determinism and make search more directed, we have decided to introduce a heuristic initialization in hybrid called HAntCO. In HAntCO, one ant has assigned schedule obtained by heuristic described in Skowroński et al. (2013b). This ant is set as favorable—it can leave much more pheromone than any other ant in a colony. Other ants in the colony are defined in the same way as in classical ACO initial colony definition.

Heuristic used to obtain an initial solution is varied depending on the optimization mode. For the duration optimization mode (DO), the successors' list size (SLS) heuristic has been used, as it provided the best results for DO mode. In this method, tasks are sorted by a number of successors they have in ascending order. Then, for every task from ordered list, a resource is assigned. The decision which resource should be assigned is determined by the earliest time when given resource would finish its previous tasks it has been assigned to.

For cost optimization (CO) mode, resources are sorted ascending by their standard salary rate and then are assigned to tasks from the list given in project definition, preserving skill constraints and avoiding conflicts, by assigning a given task-to-resource no earlier than all previously assigned tasks to resource would be finished.

In the next step, each solution is evaluated to set the pheromone value for each ant in the next iteration. The amount of pheromone left in every iteration is set according to the ant chosen as the best.

As the **stopping criterion**, the number of iterations with no change of global best solution has been proposed in this approach. It is notated as γ.

The probability of selecting resource k to task j in **edge selection** bases on the roulette method is computed as follows:

$$\text{prob}_j^k = \frac{p_j^{k\,\alpha}}{\sum_{i=1}^{n} p_i^{k\,\alpha}} \tag{9}$$

where α is a weight for pheromone values influence. This value is the parameter of ACO approach and should be provided by the user. p_j^k is a pheromone value stored in the edge containing information about k resource performing j task.

4.2 Evaluation solution method

Evaluation function is formulated as follows:

$$\min f(\text{PS}) = w_\tau f_\tau(\text{PS}) + (1 - w_\tau) f_c(\text{PS}) \tag{10}$$

where w_τ is the weight of duration component, $f_\tau(\text{PS})$ is the duration evaluation component, $f_c(\text{PS})$ is the cost evaluation component. Both components are non-negative values, while $w_\tau \in [0; 1]$.

Summing both components' weight to 1 ensures that changing the importance of one aspect would cause also some changes of second aspect's importance.

The time component $f_\tau(\text{PS})$ is calculated as follows:

$$f_\tau(\text{PS}) = \frac{\tau}{\tau_{\max}} \tag{11}$$

where $\tau_{\max}$ is the maximal (pessimistic) possible duration of the schedule PS, computed as the sum of all tasks' duration. It occurs when all tasks are performed serially in project: one-by-one. No matter how many and how flexible resources are.

The cost component $f_c(\text{PS})$ is defined as follows:

$$f_c(\text{PS}) = \frac{\sum_{i=1}^{J} c_j}{c_{\max} - c_{\min}} \tag{12}$$

where $c_{\min}$ is the minimal schedule cost—the total cost of all tasks assigned to the cheapest resource, $c_{\max}$ is the maximal schedule cost—a total cost of all tasks assigned to the most expensive resource. Note: $c_{\max}$ and $c_{\min}$ do not involve skill constraints. It means that $c_{\min}$ value could be reached only for non-feasible solution. Analogously to $c_{\max}$.

4.3 Update pheromone

Pheromone evaporates iterative. It means the pheromone value is decreased by the same value (μ) in every iteration, as it was stated in the Eq. 13.

$$(p_j^k)^{(i+1)} = (p_j^k)^i (1 - \mu) \tag{13}$$

Obtained results for various update pheromone methods strongly depend on values set for the following parameters used in ACO:

- p_{init} is the initial value of pheromone amount in each edge,
- μ is the amount of pheromone evaporated in each iteration,
- δ is the amount of pheromone left in edges by ants,
- $p_{\min}$ is the minimal value of pheromone set for resource in edge.

In the proposed approach, three strategies of setting pheromones have been researched: *ALL* (Liang et al. 2004), *ELITE* (Merkle et al. 2002) and *DIFF*. The last of the proposed ones is the new one, proposed by the authors of this paper.

4.3.1 Update pheromone: ALL

In this approach, every ant can leave the pheromone value in the edge for selected resource (Liang et al. 2004). The better the solution is, the more pheromone could be left by the ant in given edge. The best ant leaves the pheromone in the amount equal to δ. All next ants leave the amount of pheromone equal to δ divided by the ant's position (pos) in the list ordered ascending by the evaluation function value.

$$(p_j^k)^{(i+1)} = (p_j^k)^i + \frac{\delta}{\mathrm{pos}} \tag{14}$$

The main advantage of this approach is the method's resistance to being stuck in local optima. On the other hand, this approach raises a risk of missing the best solutions because of the more exploratory than exploitation-based character of search process.

4.3.2 Update pheromone: ELITE

In this approach, only elite ants are allowed to leave the pheromone on given edges. The set of elite ants always contains two ants: the one with the best solution found in the current iteration (A_{b}) and the global best one (A_{g}) (Merkle et al. 2002)—with the best solution found from the beginning of search process. For both ants, the same pheromone amount update method is set:

$$(p_j^k)^{(i+1)} = (p_j^k)^i + \delta \tag{15}$$

As this approach is more local-optimum oriented, it could lead to getting stuck in local optima. However, the convergence to the optimum of this approach is faster than in ALL method.

4.3.3 Update pheromone: DIFF

In this approach, the ant with the worst or best found solution in a given iteration is selected. Updating the pheromone value by *the worst* allows to explore the search space in other than potentially the best directions and, consequently, escape from local optima. The same like in ELITE approach, only two ants are able to leave the pheromone: the best (A_{b})/worst (A_{w}) in iteration and global best (A_{g})/global worst (A_{v}) found so far. The decision which ant (best or worst) should leave pheromone is made on the basis of satisfaction of the following condition:

$$\pi > \psi \tag{16}$$

where π is regarded as an ant population variety and is computed as follows:

$$\pi = \frac{f_{\mathrm{w}} - f_{\mathrm{b}}}{f_{\mathrm{w}}} \tag{17}$$

where f_{b} and f_{w} are the evaluation function values of the best and worst solutions contained by given ants in specified iteration. The right-sided variable ψ could be regarded as an ant population variety threshold and is set as an ACO parameter.

If condition in Eq. 16 is satisfied, ELITE update pheromone method is used. With every iteration in which condition from Eq. 16 is satisfied, the counter (κ) of possible worst pheromone update strategy use is incremented. If the variety computed in Eq. 16 is not satisfied, it means ants are concentrated near some local optima. Then, to avoid being stuck, the worst update method is launched. It means that not the best but worst ants leave pheromone on their path. Meanwhile, the counter κ is decremented. The worst ant can leave pheromone as long as the ant population variety is smaller than ψ or the κ is not negative. Initial κ value is also set as an ACO parameter.

The value of pheromone left by the global ant is defined in Eq. 18. For the global best (or worst) ant, the pheromone amount update value is defined as follows:

$$(p_j^k)^{(i+1)} = (p_j^k)^i + \frac{\delta}{\gamma} \tag{18}$$

where γ is a number of iterations from the last found new global best.

For the best/worst ant in iteration, the pheromone amount update value is stated as follows:

$$(p_j^k)^{(i+1)} = (p_j^k)^i + \frac{\delta}{\pi} \tag{19}$$

In update pheromone amount method for global ant (Eq. 18) the pheromone amount is reduced, while the pheromone amount for the best local ant is increased (Eq. 19). It enhances the possibility of finding new global optimum, reducing the probability of losing the best solution found so far at the same time.

4.4 Conflict fixing

A conflict appears when more than one task is assigned to the same resource in overlapping periods. In that case, it should be fixed by the following procedure.

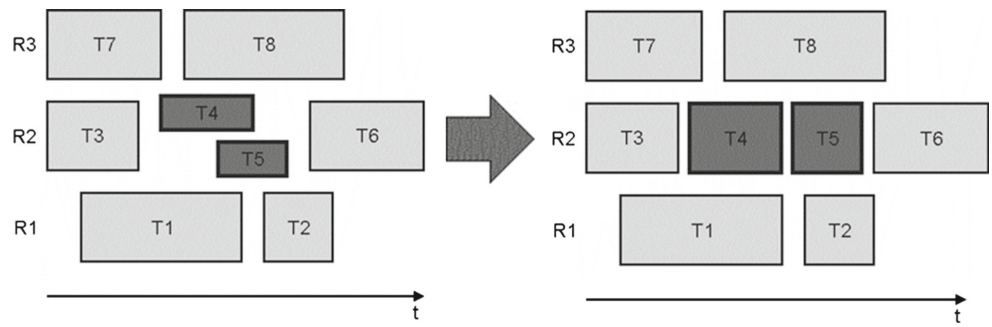

Fig. 2 Example of conflict resolving

It is performed by shifting one of conflicting task's start date. Consequently, the finish date of that task also has to be shifted to keep the task duration. The decision which of conflicting tasks should be shifted depends on which of them starts earlier. If they are set to start at exactly the same time, task to be shifted is selected by the way, which was firstly added to project definition. Furthermore, we do not investigate the velocity of resources. Therefore, job duration is constant regardless of assigned resource and skills it owns.

Conflict fixing procedure illustration is presented in the Fig. 2.

Tasks $T4$ and $T5$ have been assigned to the Resource $R2$ in overlapping period. As a conflict fixing result, a new schedule has been presented, where $T5$ starts just after the $T4$ should be finished. The $T5$ has been shifted, because it was initially set to start later than the $T4$.

5 Experiments and results

The goal of the conducted experiments was to investigate the following issues:

- robustness of ACO approach for MS-RCPSP based on given dataset,
- robustness of various update pheromone methods,
- comparing HAntCO to classical ACO approach and other (meta-)heuristics.

Therefore, we have compared the results obtained for different update pheromone methods and results for hybrids and classical ACO approach. Furthermore, the results for simple heuristic scheduling have been provided to get a reference for the ACO-based mechanism.

The obtained results (project schedules) are described by duration time ([days]) and performance cost ([currency units]). Those project schedule properties have been used to compare the investigated methods.

5.1 *iMOPSE* dataset

Due to evaluating not only the project schedule duration, but also the cost of the schedule, we cannot use the standard PSPLIB benchmark dataset (Kolisch et al. 1996) that does not contain any information about the task performance cost. What is more, PSPLIB dataset instances do not reflect the MS-RCPSP. Hence, lack of benchmark data has encouraged us to prepare the *iMOPSE* dataset, containing 36 project instances that have been artificially created[1], on the basis of real-world instances, obtained from an international enterprise. We recommend other scientists using *iMOPSE* dataset as a benchmark for investigating their approaches in solving MS-RCPSP as defined.

Instances of the dataset have been created according to the analysis made in cooperation with experienced project manager from Volvo IT. We were not allowed to get real project data because of their sensitive character for the enterprise. However, we made a statistical analysis of real projects. Then, we prepared artificial dataset instances according to the analysis result, regarding the most common project characteristics, such as a number of tasks, a number of resources, various skill types in enterprise and the structure of critical chain (a number of tasks involved by precedence relations), etc.

The iMOPSE dataset summary is presented in the Table 1. There are two groups of created project instances: one contains 100 tasks and the second—200 tasks. Within each group, project instances are varied by a number of available resources and the precedence relationship complexity. The number of resources for instances from both groups was chosen in a way to preserve constant average resource load and average task relations ratio for given instances. Hence, for project instances with 200 tasks, the number of possible resources and precedence relations is twice bigger than for project instances containing 100 tasks. The skill vari-

[1] http://imopse.ii.pwr.wroc.pl/—iMOPSE (intelligent multi-objective project scheduling environment) project homepage, containing description of investigated methods, dataset definition and best found solutions.

Table 1 iMOPSE dataset summary

Dataset instance	Tasks	Resources	Relations	Skills
100_20_23_9_D1	100	20	23	9
100_20_22_15	100	20	22	15
100_20_47_9	100	20	47	9
100_20_46_15	100	20	46	15
100_20_65_9	100	20	65	9
100_20_65_15	100	20	65	15
100_10_27_9_D2	100	10	27	9
100_10_26_15	100	10	26	15
100_10_47_9	100	10	47	9
100_10_48_15	100	10	48	15
100_10_64_9	100	10	64	9
100_10_65_15	100	10	65	15
100_5_20_9_D3	100	5	20	9
100_5_20_15	100	5	22	15
100_5_48_9	100	5	48	9
100_5_48_15	100	5	46	15
100_5_64_9	100	5	64	9
100_5_64_15	100	5	64	15
200_40_45_9	200	40	45	9
200_40_45_15	200	40	45	15
200_40_90_9	200	40	90	9
200_40_91_9	200	40	91	15
200_40_130_9_D4	200	40	130	9
200_40_144_15	200	40	133	15
200_20_55_9	200	20	55	9
200_20_54_15	200	20	54	15
200_20_97_9	200	20	97	9
200_20_97_15	200	20	97	15
200_20_150_9_D5	200	20	150	9
200_20_145_15	200	20	145	15
200_10_50_9	200	10	50	9
200_10_50_15	200	10	50	15
200_10_84_9	200	10	84	9
200_10_85_15	200	10	85	15
200_10_135_9_D6	200	10	135	9
200_10_128_15	200	10	128	15

ety has been set-up to 9 or 15 different skill types for each project instance, while any resource can dispose of exactly six different skill types. Because of the different resources' and relations' numbers, the scheduling complexity for each project is varied.

This dataset stands as an extension of dataset presented in Skowroński et al. (2013a, b), Myszkowski et al. (2013), and that is the reason some instances are named with suffix Dx. This suffix refers to dataset instances that have been previously created and presented in those papers. Because of the extension of the dataset, the need of introducing more clear name system has arisen. Suffix has been added to a reference of previously created files, keeping the naming convention applied after dataset extension.

5.2 Experiments' set-up

The experiments have been divided into investigating the influence of ACO parameters' configurations for project duration and performance cost in three various components' weights in evaluation function: duration optimization (DO: $w_\tau = 1$, see. Eq. 10), balanced optimization (BO: $w_\tau = 0.5$) and cost optimization (CO: $w_\tau = 0$). Because of the stochastic nature of ACO-based methods, each experiment for given parameter configuration has been repeated ten times. For K–S test and t test, each experiment has been repeated 50 times (see Tables 9, 10). On the other hand, deterministic character of heuristics allowed us to obtain results for those methods in only one iteration for every parameters' configuration.

The further step of the conducted experiments was to compare results obtained for random initial solution with boosting initial solution using described hybrids. Initial solution has been previously obtained using the above-mentioned heuristics and then set them as input for ACO and made those results as more favorable in local search by enhancing the pheromone value left in this path representing initial solution. We decided to use SLS(D) (Skowroński et al. 2013b) for DO mode and RS(A) (Skowroński et al. 2013b) for CO mode optimization within HAntCO hybrid. Because of some code refactoring, we were able to tune our heuristics and obtain a better solution than the found ones in Skowroński et al. (2013b). That is the reason why the results of those heuristics in this paper are slightly better than the results in Skowroński et al. (2013b) for given dataset instances.

To present averaged results in detail (see Table 4), a standard deviation measure (σ) has been introduced and applied to each average value, presented as a percentage value in relation to the average. We have also added information about the best found solution for a given method (see Table 2) that have been compared with the results obtained by most promising heuristics, described in Skowroński et al. (2013b).

Both for the best and averaged results, pheromone update methods have been compared and the one that provided best results (shortest duration in DO, smallest cost in CO and smallest evaluation function value in BO) is presented in Tables 2 and 4. The notation for methods used in tables with obtained results is as the following: E—update ELITE pheromone method, A—update ALL, D—update DIFF. If more than one pheromone update methods turned out to be the best and gave the same results, they have been presented both separated by "/" (e.g., E/D—both update DIFF

Table 2 Comparison of the best obtained results for DO, BO and CO modes in classical ACO and selected heuristics from Skowroński et al. (2013b)

Dataset instance	ACO									Heuristics				
	DO			BO			CO			DO		CO		
	M	Days	Cost	M	Days	Cost	M	Days	Cost	Days	Cost	C	Days	Cost
100_10_26_15	E	**32**	124,687	E/D	85	70,326	E/D	85	**70,326**	37	126,361	RS(A)	85	**70,326**
100_10_27_9_D2	E	**34**	44,999	D	72	27,120	E/D	129	**26,323**	38	44,309	RS(A)	129	**26,323**
100_10_47_9	E	**36**	143,100	D	105	94,334	E/D	145	**90,992**	41	142,759	RS(A)	145	**90,992**
100_10_48_15	E	**33**	133,062	E/D	81	87,194	E	85	**87,187**	36	135,534	RS(A)	85	**87,187**
100_10_64_9	D	**35**	110,643	D	92	63,934	E/D	121	**62,102**	39	113,124	RS(A)	121	**62,102**
100_10_65_15	E	**35**	150,294	E/D	76	108,312	E/D	98	**106,296**	40	152,955	RS(A)	98	**106,296**
100_20_22_15	D	**20**	120,949	D	56	56,625	D	87	55,240	25	117,493	ADAD	86	**55,240**
100_20_23_9_D1	D	**32**	52,119	D	60	30,900	D	121	30,107	32	53,154	AAAD	119	**30,104**
100_20_46_15	E	**25**	138,565	D	65	69,789	E/D	75	**68,899**	28	138,270	RS(A)	75	**68,899**
100_20_47_9	E	**21**	124,817	D	69	59,196	D	131	**55,197**	21	129,160	RS(A)	131	**55,197**
100_20_65_15	E	**27**	109,831	D	52	57,338	E/D	69	**57,085**	32	11,0503	RS(A)	69	57,085
100_20_65_9	E	**23**	130,934	D	76	61,913	D	114	59,736	25	127,149	RS(A)	114	**59,736**
100_5_20_9_D3	E	**50**	41,029	D	75	31,681	E/D	167	**30,164**	57	40,539	RS(A)	167	**30,164**
100_5_22_15	D	**60**	119,434	D	70	110,145	E/D	86	**109,111**	63	119,266	RS(A)	86	**109,111**
100_5_46_15	E	**67**	204,110	*	125	184,409	E/D	125	**184,409**	75	202,238	RS(A)	125	**184,409**
100_5_48_9	E	**62**	191,712	E/D	127	175,526	E/D	130	**175,225**	72	193,383	RS(A)	130	**175,225**
100_5_64_15	D	**62**	144,972	E/D	123	109,431	E/D	141	**109,091**	71	141,407	RS(A)	141	**109,091**
100_5_64_9	E	**61**	102,777	D	87	74,617	E/D	173	**72,848**	71	102,439	RS(A)	173	**72,848**
200_10_128_15	E	**62**	178,264	D	126	136,643	E	143	136,551	71	180,812	AxAD	159	**134,425**
200_10_135_9_D6	*	**216**	99,375	E	237	72,753	D	274	72,036	216	105,593	RS(A)	256	**71,986**
200_10_50_15	E	**63**	191,856	D	144	85,712	E/D	167	**84,308**	66	189,660	RS(A)	167	**84,308**
200_10_50_9	E	**65**	250,075	D	228	110,218	D	318	105,232	66	251,158	RS(A)	318	**105,198**
200_10_84_9	E	**69**	226,666	D	171	125,715	D	316	117,754	70	224,121	DAAA	338	117,543
200_10_85_15	E	**61**	306,949	E	180	197,767	E	215	**195,820**	65	304,277	RS(A)	215	**195,820**
200_20_145_15	E	36	278,199	D	109	144,694	D	152	143,688	**36**	275,983	RS(A)	158	**143,497**
200_20_150_9_D5	D	186	91,461	D	247	52,620	D	296	51,678	**183**	92,821	ADDA	337	**51,496**
200_20_54_15	E	39	299,993	D	123	161,883	D	131	161,614	**37**	295,786	RS(A)	125	**161,412**
200_20_55_9	D	38	231,094	D	159	75,836	D	250	72,176	**37**	230,150	RS(A)	332	**70,057**
200_20_97_15	D	**42**	280,951	D	115	160,070	D	169	157,202	49	290,399	RS(A)	171	**156,951**
200_20_97_9	E	37	275,819	D	114	102,641	D	150	99,901	**35**	273,378	RS(A)	169	**98,480**
200_40_130_9_D4	*	**112**	94,488	D	132	48,362	D	205	48,419	112	101,879	DAAD	214	**46,133**
200_40_133_15	D	27	281,933	D	93	101,620	D	131	99,329	**24**	276,456	AAAA	155	**97,345**
200_40_45_15	E	**25**	248,717	D	118	95,959	D	161	91,010	31	260,738	RS(A)	213	**87,955**
200_40_45_9	E	26	273,632	D	118	96,375	D	179	94,142	**22**	270,758	AAAA	334	**77,236**
200_40_90_9	E	26	287,694	D	115	97,926	D	142	96,312	**24**	290,028	RS(A)	285	**80,732**
200_40_91_15	E	25	257,927	D	82	91,204	D	132	88,616	**19**	249,909	RS(A)	184	**86,476**

and update ELITE methods gave the same, best results). In Table 2, a sign $*$ has been also introduced to indicate a situation where all three methods provided the same, regarded as the best, result.

All the results presented in tables have been obtained for given ACO parameter configuration: $p = 12$, $\mu = 0.1$, $p_{\text{init}} = 1.5$, $\alpha = 1$, $\delta = 0.05$, $p_{\min} = 0.05$, $h_{\text{init}} = 1$, $\beta = 0$, $\gamma = 150$, $\sigma = 30$, $\psi = 0.1$, $\kappa_{\text{init}} = 20$. This configuration has been regarded as the best, defined as a result of the previous parameter-tuning experiments. The same configuration has been chosen to be used in every pheromone update method (ALL, ELITE, DIFF), every optimization mode approach (DO, BO, CO) both for ACO and HAntCO approaches.

5.3 Experiments' performance

The processing time was varied in relation to the used update method. For ALL method that could be regarded as the simplest, the processing time was relatively small (from 7 to 90 s, depending on processed dataset instance). However, for ELITE and DIFF methods that are regarded as more complex because of the need of sorting ants and choosing best/worst, the processing time varied from 30 to 270 s per one execution in one CPU for given parameter configuration.[2]

5.3.1 The best found results

The best results obtained by ACO for CO and DO modes have been compared with the results obtained using heuristics proposed in Skowroński et al. (2013b). In Table 2, this comparison is presented. For each dataset instance and optimization mode, the best results have been chosen from various pheromone update methods. Indication which method provided the best results is stored in columns named M for every optimization mode.

The obtained best results have been compared with the heuristic results. We decided to omit the name of heuristic if possible to reduce the space covered by the table. For heuristic results in CO, SA heuristic name has been omitted without losing any important information, as the parameter configuration for that method has been written in the table. To give a more detailed view about those methods, please refer to Skowroński et al. (2013b).

Better values from comparison optimization modes between ACO and heuristics have been written in bold. If key values (duration for DO or cost for CO) were equal for ACO and heuristic approaches, the smallest value of the second aspect has been taken into account to choose a better solution. If both project schedule properties turned out to be the same, both solutions were written in bold.

To determine the best obtained result for BO mode, neither duration nor cost has been investigated. Instead of those aspects, the evaluation function value has been taken into account. Furthermore, we were not able to compare strictly the results of BO for ACO with corresponding ones for heuristics, as no evaluation function has been used to evaluate results of heuristics.

A similar analysis has been made for the best found results within investigated hybrid. The best HAntCO results are presented in Table 3. The most significant difference for HAntCO best results table in comparison with table of best results for classical ACO is that there is no BO mode included. It is because hybrid is activated only for DO or CO mode—depending on selected heuristic for initialization.

Taking into account the results gathered in Table 3, we can assume that the ELITE strategy mode for HAntCO generally provides better results than DIFF in DO mode. It provided better results in 26 cases (72 %). However, in CO, we noticed that the DIFF strategy turned out to be more suitable than the ELITE, provided better results in 9 cases (25 %), while the ELITE became better in only one case (less than 3 %). In remaining cases, both strategies gave the same best results. An interesting fact is that for DO, no equal best results for both strategies have been found.

Also comparing HAntCO best results (see Table 3) to single heuristics results (see. Table 2), we can see that hybrid ACO with heuristics is more effective for DO than CO mode. In most instances (89 %), HAntCO found a better solution than simple heuristic or ACO.

5.3.2 Averaged results

Averaged results obtained for various pheromone update methods are presented in Table 4 in a similar way as the ones in Table 2, respectively. We also provided in Table 4 the notation for the method that provided best results (A, D, E, D/E). In opposition to Table 2, no comparison to averaged heuristic results has been introduced, because heuristics are deterministic methods for which result can be obtained in only one iteration. On the other hand, in Table 4, a standard deviation measure (σ) has been introduced to indicate the level of variability of the obtained results. It is presented as a percentage value of an average.

For DO and CO modes, the smallest averaged values of project duration or project cost, respectively, have been taken into account to determine the best pheromone update method. If values of given aspect are equal, the smallest value of the second aspect is taken into account. If there is still no possibility to determine which pheromone update method provides better solutions, the standard deviation of more important aspect is taken into account (duration for DO and cost for CO, respectively) and the method with smaller standard deviation value is regarded as better.

We have also provided averaged results for HAntCO approach, presented in Table 5. Analogously to best HAntCO approach results, averaged ones regard only DO and CO modes. Averaged values are supported by standard deviation values that reflect the variability of the obtained results. We have also decided to count how many times one strategy became better than another also in averaged results. For DO, ELITE strategy became better in 25 cases (69 %), while DIFF became better in the remaining ones. For CO, DIFF strategy provided better results in 14 cases (39 %), while only in one case ELITE strategy became better. For the remaining ones, the obtained averaged results became the same. It leads to conclusion that HAntCO searches space in CO mode in very directed way, being unable to explore other parts of the solu-

[2] Machine for tests was equipped with 8 CPUs Intel Core i7 2.67 GHz each, 24 GB of RAM memory and Ubuntu 12.04 OS.

Table 3 Best results obtained for HAntCO with various pheromone update methods in DO and CO optimization modes

Dataset instance	DO				CO			
	ELITE		DIFF		ELITE		DIFF	
	Days	Cost	Days	Cost	Days	Cost	Days	Cost
100_10_26_15	**31**	126,216	32	125,688	85	70,326	85	70,326
100_10_27_9_D2	**33**	42,199	35	44,022	129	26,323	129	26,323
100_10_47_9	**34**	140,865	34	142,362	145	90,992	145	90,992
100_10_48_15	33	134,692	**33**	133,495	85	87,187	85	87,187
100_10_64_9	**33**	113,774	34	115,998	121	62,102	121	62,102
100_10_65_15	33	149,175	**32**	149,185	98	106,296	98	106,296
100_20_22_15	**19**	123,642	20	118,054	87	55,240	87	55,240
100_20_23_9_D1	**23**	53,358	24	54,309	117	30,104	117	30,104
100_20_46_15	**24**	138,568	24	142,206	75	68,899	75	68,899
100_20_47_9	**18**	134,312	21	133,050	131	55,197	131	55,197
100_20_65_15	**27**	108,991	27	113,275	69	57,085	69	57,085
100_20_65_9	21	126,659	**20**	128,354	114	59,736	114	59,736
100_5_20_9_D3	53	41,310	**53**	40,811	167	30,164	167	30,164
100_5_22_15	**60**	119,158	61	119,218	86	109,111	86	109,111
100_5_46_15	**67**	204,730	70	205,618	125	184,409	125	184,409
100_5_48_9	**62**	191,888	62	192,315	130	175,225	130	175,225
100_5_64_15	61	145,322	**61**	143,956	141	109,091	141	109,091
100_5_64_9	**61**	101,297	62	103,777	173	72,848	173	72,848
200_10_128_15	**60**	178,375	61	180,400	143	136,551	143	136,551
200_10_135_9_D6	**186**	103,561	186	105,515	269	**71,986**	270	71,986
200_10_50_15	**62**	190,956	62	191,149	167	84,308	167	84,308
200_10_50_9	**63**	253,214	64	250,850	318	105,198	318	105,198
200_10_84_9	67	224,639	**66**	222,655	318	117,543	318	117,543
200_10_85_15	62	303,301	**62**	302,064	215	195,820	215	195,820
200_20_145_15	**35**	272,504	35	277,291	158	143,497	158	143,497
200_20_150_9_D5	187	90,548	**177**	92,567	344	51,524	345	**51,496**
200_20_54_15	**34**	298,822	36	295,819	125	161,412	125	161,412
200_20_55_9	**36**	223,879	36	227,449	311	70,967	332	**70,057**
200_20_97_15	42	290,308	**42**	277,860	171	156,951	171	156,951
200_20_97_9	**35**	278,797	36	270,910	155	99,190	169	**98,480**
200_40_130_9_D4	108	106,637	**108**	104,965	225	47,212	216	**46,275**
200_40_133_15	**24**	282,730	24	279,073	141	97,953	144	**97,345**
200_40_45_15	**23**	256,687	23	256,753	201	89,407	213	**87,955**
200_40_45_9	**25**	270,428	26	263,162	270	89,123	315	**82,192**
200_40_90_9	**24**	298,340	25	293,098	229	93,090	247	**84,038**
200_40_91_15	**23**	241,492	23	248,984	176	87,875	184	**86,476**

tion space. Independent character of searching is, in many cases, regardless of applied pheromone update strategy.

To investigate the level of stability of HAntCO in comparison with classical ACO, we have checked how many times 0-equal standard deviation value has been obtained in the conducted experiments. Those results are presented in Table 6. The results gathered in this table prove that the proposed hybrid approach is more directed and thus, the proposed approach found the same solution in many more cases than classical ACO which stochastic nature allows to explore the search space more widely.

The most interesting results found in Table 6 concern CO mode. For that mode, HAntCO found the same cost solutions 21 (58 %) times for ELITE and 16 times (44 %) for DIFF strategies, while the same duration solutions have been found 24 (67 %) and 25 (69 %) times, respectively.

Table 4 Averaged results obtained for classical ACO in various optimization modes

Dataset instance	DO					BO					CO				
	M	Days		Cost		M	Days		Cost		M	Days		Cost	
		Avg	σ	Avg	σ		Avg	σ	Avg	σ		Avg	σ	Avg	σ
100_10_26_15	E	33.2	2.6	125,436	1.5	D	85	0.0	70,326	0.0	E	84.9	0.4	70,363	0.1
100_10_27_9_D2	E	36.2	4.1	43,382	1.8	E	75.4	2.1	27,064	0.1	E	130.5	3.5	26,326	0.0
100_10_47_9	E	37.5	2.7	142,742	0.4	D	104.9	1.3	94,501	0.3	D	144.8	0.4	91,088	0.1
100_10_48_15	E	35.2	4.0	135,563	2.0	E	81	0.0	87,214	0.0	D	85.3	0.5	87,205	0.0
100_10_64_9	E	36.8	2.7	114,538	1.8	D	90.5	1.3	64,231	0.4	D	121	0.0	62,136	0.1
100_10_65_15	E	35.8	3.0	152,033	1.2	E	76.7	1.4	108,266	0.1	E	98	0.0	106,299	0.0
100_20_22_15	D	22	4.5	118,254	2.9	E	52.5	3.8	57,503	1.0	E	84.5	4.0	55,431	0.3
100_20_23_9_D1	A	32	0.0	52,915	2.5	D	63.2	3.2	31,009	0.5	D	115.7	9.2	30,212	0.7
100_20_46_15	E	24.9	3.3	140,271	2.4	D	67.4	3.6	69,574	0.4	D	75.2	0.8	68,932	0.1
100_20_47_9	E	23.3	5.1	128,127	3.2	D	69.7	3.1	59,802	0.9	E	116.6	7.8	56,800	1.8
100_20_65_15	E	27.2	1.5	111,946	4.0	E	51.4	2.3	57,645	0.5	E	66.9	3.7	57,131	0.1
100_20_65_9	E	23.9	2.3	126,709	2.8	E	71.5	4.5	64,189	2.7	D	103.1	10.5	60,929	2.6
100_5_20_9_D3	E	52.4	2.4	41,152	1.1	E	76.5	2.0	31,653	0.1	E	166.9	0.2	30,167	0.0
100_5_22_15	E	61	0.7	119,479	0.4	E	70.2	0.6	110,135	0.0	E	86	0.0	109,111	0.0
100_5_46_15	E	68.2	1.7	204,507	0.3	E	125	0.0	184,409	0.0	E	125	0.0	184,409	0.0
100_5_48_9	E	63.1	1.1	191,911	0.2	E	127	0.0	175,535	0.0	E	130	0.0	175,225	0.0
100_5_64_15	E	62.6	0.8	144,257	0.7	D	123.1	0.2	109,428	0.0	/	141	0.0	109,091	0.0
100_5_64_9	E	63	1.9	103,527	1.3	D	87	0.0	74,617	0.0	E	172.9	0.2	72,850	0.0
200_10_128_15	E	63.3	1.9	178,421	1.2	D	124.9	1.1	136,938	0.2	E	140.7	1.3	136,568	0.0
200_10_135_9_D6	E	216	0.0	100,758	1.6	D	247.2	1.8	72,693	0.5	D	267.3	1.2	72,127	0.1
200_10_50_15	E	65.3	1.9	190,271	2.2	E	134.3	3.2	87,158	0.6	E	166.7	0.4	84,402	0.1
200_10_50_9	E	66.6	1.8	247,741	1.7	E	220.5	2.8	113,340	1.6	D	311	3.0	105,825	0.8
200_10_84_9	E	71.1	2.0	224,680	1.9	E	162.1	2.0	129,065	1.2	E	275.7	7.2	121,478	1.3
200_10_85_15	E	64.3	2.2	307,437	1.0	E	170.2	3.6	199,332	0.7	D	212.3	1.7	196,662	0.4
200_20_145_15	E	38.3	2.6	272,720	1.8	D	108.3	2.1	146,285	0.9	D	143.2	10.5	144,947	1.1
200_20_150_9_D5	D	190.7	1.3	91,095	3.2	D	237	3.1	54,032	2.3	D	266.9	12.1	54,512	8.3
200_20_54_15	E	41.2	3.4	288,063	2.2	D	124.3	1.7	162,514	0.4	D	133.3	4.4	162,498	0.4
200_20_55_9	D	39.7	1.6	228,459	2.5	D	148.3	9.5	80,793	8.5	D	230.5	8.3	75,247	4.3
200_20_97_15	D	43.3	2.7	287,731	1.6	D	114.9	2.6	160,892	0.4	D	160.5	11.1	158,560	1.6
200_20_97_9	D	40.8	3.3	281,754	2.0	D	112.3	2.7	105,641	2.9	D	134	5.2	101,992	1.7
200_40_130_9_D4	E	112	0.0	102,221	3.4	D	141.7	9.8	51,413	11.9	D	185.1	7.2	49,156	1.6
200_40_133_15	E	28.4	3.2	282,463	2.2	D	89.7	3.1	104,442	1.9	D	116.5	10.6	102,689	3.0
200_40_45_15	E	26.9	3.9	247,230	3.8	D	106.8	7.2	102,650	4.0	D	160.8	9.8	94,330	3.6
200_40_45_9	E	28.2	4.1	267,910	2.1	D	102.6	10.4	106,705	6.6	D	182.8	8.3	97,018	2.0
200_40_90_9	E	27.4	3.3	288,861	2.0	D	109.3	12.1	104,403	8.2	D	133	13.1	102,871	7.3
200_40_91_15	E	26.4	3.5	242,588	2.4	D	80.2	7.9	96,756	6.8	D	112.2	10.8	92,724	4.0

5.4 Computational complexity

Our research has been extended by investigating the level of complexity of compared methods. The complexity has been estimated as a number of potential assignments of resources to a given task as dominant operations. As this value is constant regardless of the optimization process and depends only on initial skill constraints, we can compute the level of complexity as a factor of an average number of iterations and a number of possible assignments. The results of those computations are presented in Table 7.

As we decided to set a constant number of iterations in most methods such as TS, EA S and EA C, the complexity level for those methods was easy to compute. For ACO

Table 5 Averaged results obtained for HAntCO with various pheromone update methods in DO and CO optimization modes

Dataset instance	DO								CO							
	ELITE				DIFF				ELITE				DIFF			
	Days		Cost		Days		Cost		Days		Cost		Days		Cost	
	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ
100_10_26_15	**32.5**	0.92	125,889	1,498	32.6	0.49	125,848	1,373	85	0.00	70,326	0	85	0.00	70,326	0
100_10_27_9_D2	**35.1**	1.37	43,644	661	35.8	0.75	43,992	650	129	0.00	26,323	0	129	0.00	26,323	0
100_10_47_9	**34.9**	1.04	142,103	998	35.2	0.75	143,263	944	145	0.00	90,992	0	145	0.00	90,992	0
100_10_48_15	**34**	0.63	134,504	1,507	34.4	0.66	134,568	1,509	85	0.00	87,187	0	85	0.00	87,187	0
100_10_64_9	**34.7**	1.10	113,638	1,871	34.9	0.54	113,230	1,899	121	0.00	62,102	0	121	0.00	62,102	0
100_10_65_15	33.6	0.66	149,474	963	**33.2**	0.60	149,598	1,033	98	0.00	106,296	0	98	0.00	106,296	0
100_20_22_15	20.7	1.00	118,914	2,464	**20.6**	0.49	118,347	2,895	87	0.00	55,240	0	87	0.00	55,240	0
100_20_23_9_D1	**24.5**	0.81	53,810	1,028	25	0.77	53,051	1,243	117	0.00	30,104	0	117	0.00	30,104	0
100_20_46_15	**24.2**	0.40	140,491	2,823	24.2	0.40	141,045	3,922	75	0.00	68,899	0	75	0.00	68,899	0
100_20_47_9	**20.3**	1.10	128,641	2,938	21.7	0.46	127,577	3,023	131	0.00	55,204	19	131	0.00	**55,197**	0
100_20_65_15	27.2	0.40	111,842	2,758	**27**	0.00	113,219	2,501	69	0.00	57,085	0	69	0.00	57,085	0
100_20_65_9	21.9	0.70	126,081	1,789	**21.6**	0.80	125,269	4,271	114	0.90	59,744	24	114	0.00	**59,736**	0
100_5_20_9_D3	**53.3**	0.46	40,917	238	54.4	0.80	41,025	148	167	0.00	30,164	0	167	0.00	30,164	0
100_5_22_15	**61.4**	0.80	119,219	486	61.9	0.83	118,934	787	86	0.00	109,111	0	86	0.00	109,111	0
100_5_46_15	**69.8**	1.54	205,451	555	70.9	0.30	204,973	615	125	0.00	184,409	0	125	0.00	184,409	0
100_5_48_9	**62.8**	0.40	191,934	171	63	0.45	192,103	342	130	0.00	175,225	0	130	0.00	175,225	0
100_5_64_15	**62.6**	1.02	144,256	1,342	62.9	0.94	144,077	813	141	0.00	109,091	0	141	0.00	109,091	0
100_5_64_9	**62.5**	1.12	102,901	1,226	62.8	0.75	103,495	751	173	0.00	72,848	0	173	0.00	72,848	0
200_10_128_15	**61.1**	1.14	179,159	1,773	61.8	0.40	178,981	1,685	143	0.00	136,551	0	143	0.00	136,551	0
200_10_135_9_D6	190.9	7.53	103,411	2,442	**186.8**	2.40	104,042	2,117	268	2.69	71,986	0	268.7	1.73	71,986	0
200_10_50_15	**63.4**	1.43	188,265	2,814	63.8	1.08	189,963	2,903	167	0.00	84,308	0	167	0.00	84,308	0
200_10_50_9	**64**	0.77	250,681	2,505	64.8	0.40	249,281	1,911	318	0.00	**105,198**	1	317.6	1.20	105,217	57
200_10_84_9	67.9	0.83	224,551	1,907	**67.4**	1.02	224,596	1,505	318	0.60	117,549	19	318	0.00	**117,543**	0
200_10_85_15	**62.9**	0.83	303,381	2,050	63.2	0.60	303,335	2,961	215	0.00	195,820	0	215	0.00	195,820	0
200_20_145_15	36.6	0.80	275,546	3,066	**36.5**	0.67	277,057	3,948	158	0.00	143,507	16	158	0.00	**143,497**	0
200_20_150_9_D5	191.6	2.29	90,882	3,176	**184.8**	5.02	92,562	1,457	318	16.31	51,678	74	345.9	1.45	**51,497**	2
200_20_54_15	**36.7**	1.42	295,455	2,829	37.5	0.92	293,412	3,656	125	0.30	161,424	25	125	0.00	**161,412**	0
200_20_55_9	**37**	1.00	229,781	4,000	37.7	0.78	228,500	5,602	310	8.83	71,652	483	328	4.96	**70,154**	92
200_20_97_15	42	0.00	287,989	4,572	**42**	0.00	285,854	5,826	171	0.00	156,951	0	171	0.00	156,951	0
200_20_97_9	**37.3**	1.19	275,710	4,650	37.6	0.80	276,680	5,627	152	5.81	100,450	1, 414	168.7	0.64	**98,500**	43
200_40_130_9_D4	108	0.00	103,493	2,383	**108**	0.00	103,389	1,692	219	4.93	48,022	533	216.1	0.94	**46,663**	329

Table 5 continued

Dataset instance	DO								CO							
	ELITE				DIFF				ELITE				DIFF			
	Days		Cost		Days		Cost		Days		Cost		Days		Cost	
	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ	Avg	σ
200_40_133_15	25.4	0.66	280,950	4,927	**25.4**	0.66	279,931	3,980	138	4.12	98,962	585	145.3	3.47	**97,396**	72
200_40_45_15	**23.6**	0.80	256,232	3,997	24.2	0.60	256,521	3,155	198	3.93	91,369	970	212.3	1.49	**87,974**	31
200_40_45_9	**26**	0.63	271,406	4,939	26.4	0.49	267,745	7,041	266	10.32	93,099	2, 528	301.5	9.74	**83,744**	1,500
200_40_90_9	**25.4**	0.80	292,674	8,765	26	0.63	291,293	5,745	219	10.62	97,899	3, 623	250.6	10.04	**85,915**	1,533
200_40_91_15	**23.7**	0.64	246,065	3,201	24.1	0.54	248,715	6,059	164	7.56	89,262	810	178.6	5.62	**86,590**	133

Table 6 Number of 0-equal standard deviation measures for given pheromone update strategies and optimization modes

Method	ELITE				DIFF			
	DO		CO		DO		CO	
	Days	Cost	Days	Cost	Days	Cost	Days	Cost
ACO	3	0	5	4	3	0	4	1
HAntCO	2	0	24	21	3	0	25	16

Table 7 Average number of dominant operations (divided by 10^3) during optimization process using investigated methods for given parameters' configurations

	D1	D2	D3	D4	D5	D6
TS	200.3	38.3	22.7	234.5	159.6	72.0
EA C	80.3	38.3	22.7	234.5	159.6	72.0
ACO	1,287.9	472.8	205.9	3,038.4	2,221.4	1,063.4
HAntCO	423.9	212.5	86.2	1,925.5	1,481.2	323.3
H	0.803	0.383	0.227	2.345	1.596	0.72

and HAntCO, we decided to get an average number of iterations from all optimization modes (DO, BO, CO) and update pheromone methods (ALL, ELITE, DIFF) as the value that should be multiplied by a number of possible assignments.

Based on the results gathered in Table 7, we can notice that ACO and HAntCO are most processing complex methods. However, the level of complexity for HAntCO is lower than for classical ACO. It is because the number of iterations for hybrids is generally smaller, as searching is started from more directed place in the solution space.

Complexity level of heuristics has been computed as multiplication of a number of possible assignments by 1, as there is only one iteration in heuristic scheduling process. What is more, heuristics are deterministic approaches. Therefore, we always get the same results that are obtained in only one iteration. Hence, heuristic could be used as a powerful tool to get the first glance of optimization possibilities for given dataset instance.

5.5 Results and discussion

Both for the best and averaged results for classical ACO, ELITE update pheromone method turned out to be the best for DO mode, while DIFF update pheromone method became the most suitable for BO mode. However, it is not possible to get such straightforward conclusions for CO mode, because DIFF method became the most suitable choice for the best obtained results, while both DIFF and ELITE methods provided equally good results for average obtained optimization results.

We have also compared pheromone update methods in hybrids performance. For that approach, ELITE mode turned out to be the most suitable for DO, while any (∗) proposed pheromone update method became equally good for CO mode for most project instances. No difference between pheromone update method has been also observed in 15/36 (42 %) cases in CO. It could lead to conclusion that pheromone update method is not as crucial as for classical ACO. It is because initial solution is preferred—hybrid is more exploitation—than exploration-oriented.

We have also compared how many times heuristics provided better results than the best ones obtained from an application of ACO approaches (see Table 2). For DO, SLS heuristic became better 9 times (25 %); while for CO SA or RS, heuristics became better 18 times (50 %). It shows that classical ACO approach proposed in this paper cannot be fully regarded as better in comparison with heuristic methods. However, combining it with heuristic in hybrid (HAntCO) approach turned out to give usually much better results than any other investigated methods, especially for CO mode.

An interesting fact is that DO mode is generally more stable than other based on the provided results. It has been deducted by counting number of bigger than 10 % σ values in Table 4. For DO, there were no such values, while, for BO, there were 3 over 10 % values (2 for duration aspect and 1 for a cost aspect). Finally, for CO, there were 7 over 10 % values of standard deviation—all for a duration aspect.

An interesting conclusion that could be made regardless of the best or averaged results is that a DIFF strategy provided better solutions in DO mode but mostly for dataset instances containing 200 tasks. The best results obtained by a DIFF strategy were better than obtained by an ELITE in 9 cases for 200 task-project instances (50 %), while ELITE strategy provided only one better solution than a DIFF (5 %). Averaged results obtained in a DIFF mode were better in 12 cases (67 %), while ELITE strategy still provided only one better solution in comparison with a DIFF.

Comparing the best results obtained by ACO and HAntCO, it can be noticed that HAntCO outclasses classical ACO, whichever pheromone update method would be used. For DO, classical ACO approach has been better than HAntCO in only 5 from 36 cases; while for CO, HAntCO became better than ACO for every project instance. Analysing averaged results, there are only 3 cases with ACO results better than HAntCO ones. Still only for DO. For CO, ACO has never been better than HAntCO. It proves the legitimacy of using

Table 8 Comparison of best obtained results for investigated methods in DO, BO and CO modes

Method	Mode	Crit.	D1	D2	D3	D4	D5	D6
TS	DO	Days	32	33	51	**92**	179	199
		Cost	40,656	43,542	40,054	88,720	80,448	97,978
	BO	Days	37	49	61	125	184	222
		Cost	38,939	34,240	36,100	50,438	54,181	75,996
	CO	Days	129	179	133	254	481	330
		Cost	30,750	26,444	31,645	46,371	52,425	73,126
EA S	DO	Days	32	34	52	112	179	216
		Cost	41,509	42,804	40,768	66,196	90,753	81,344
	BO	Days	32	40	57	112	188	216
		Cost	42,975	40,387	38,486	87,107	84,067	88,317
	CO	Days	116	133	163	196	417	294
		Cost	30,158	26,691	34,361	52,027	52,400	74,897
EA C	DO	Days	35	52	64	112	183	216
		Cost	41,217	37,248	40,242	87,487	81,555	99,462
	BO	Days	46	77	77	114	211	216
		Cost	37,190	31,888	35,527	79,854	72,918	92,602
	CO	Days	56	94	84	120	230	216
		Cost	35,760	31,328	34,160	78,928	72,338	91,972
ACO	DO	Days	32	34	**50**	112	186	216
		Cost	52,119	44,999	41,029	94,488	91,461	99,375
	BO	Days	60	72	75	132	247	237
		Cost	30,900	27,120	31,681	48,362	52,620	72,753
	CO	Days	121	129	167	205	296	274
		Cost	30,107	**26,323**	**30,164**	48,419	51,678	72,036
HAntCO	DO	Days	**23**	**33**	53	108	**177**	**186**
		Cost	53,358	42,199	40,811	104,965	92,567	103,561
	CO	Days	117	129	167	216	344	267
		Cost	**30,104**	**26,323**	**30,164**	46,342	51,496	71,986
H	DO	Days	32	38	57	112	183	216
		Cost	53,154	44,309	40,539	101,879	92,821	105,593
	CO	Days	119	129	167	214	337	256
		Cost	30,104	**26,323**	**30,164**	**46,133**	**51,496**	**71,986**

hybrids that become robust way of boosting optimization process.

To get bigger awareness of classical ACO and HAnt-CO approaches' robustness, we decided to compare the obtained best results for ACO with best results obtained using other methods, as EA (Skowroński et al. 2013a) and TS (Myszkowski et al. 2013). However, we needed to distinguish the best results obtained for DO and CO modes from BO mode, because no heuristic scheduling method has been proposed for BO. Comparison of DO, BO and CO modes is presented in Table 8.

This comparison has been made only for project instances D1–D6, because only those have been investigated in Skowroński et al. (2013a, b), Myszkowski et al. (2013). The compared methods are Taboo Search (TS), specialized Evolutionary Algorithms (EA S), classic EA (EA C), classical ACO, HAntCO and heuristics (H).

The results presented in Table 8 show that both HAnt-CO and TS outclassed other methods in DO mode, obtaining best cost results for half of investigated project instances for each method (D1, D2, D5, D6 for HAntCO and D3, D4 for TS). For CO mode, classical ACO became the best approach for D2 and D3 instances, while HAntCO obtained the best results for the same instances plus D1. However, the most successful approach for these instances is a heuristic one that allowed to get best results in 5/6 cases.

The averaged results of investigated methods are presented in Table 9. It differs slightly from the results in Table 8, as methods are non-deterministic. However, conclusions are very similar: HAntCO outperforms other methods in almost

Table 9 Comparison of averaged obtained results for investigated methods in DO and CO modes

Method	Mode	Crit.	D1	D2	D3	D4	D5	D6
TS	DO	Days	35.06 ± 2.26	46.14 ± 3.06	71.0 ± 0.0	112 ± 0.0	183.0 ± 0.0	216.0 ± 0.0
		Cost	41,151 ± 201	38,205 ± 950	38,748 ± 0.0	87,691 ± 206	79,927 ± 166	98,538 ± 138
	CO	Days	128 ± 4.99	176.7 ± 11.6	133.4 ± 4.4	248.3 ± 21.4	467.3 ± 23.7	358.2 ± 17.2
		Cost	30,693 ± 2.1	26,424 ± 3.4	31,637 ± 0.0	46,359 ± 128	52,354 ± 43	72,961 ± 0.0
EA S	DO	Days	32 ± 0.00	37.52 ± 1.28	54.68 ± 1.39	112 ± 0.00	**180 ± 1.51**	216 ± 0.00
		Cost	52,781 ± 1,510	43,547 ± 909	41,082 ± 544	104,459 ± 4,194	92,355 ± 3,234	100,002 ± 4,511
	CO	Days	43.9 ± 7.64	150 ± 3.09	110.7 ± 10	234.66 ± 20.4	443.8 ± 25.8	221.6.5 ± 10.88
		Cost	46,492 ± 673	26,344 ± 57	34,834 ± 535	47,600 ± 509	**51,200 ± 220**	93,914 ± 957
EA C	DO	Days	32.0 ± 0.00	46.6 ± 2.27	68.32 ± 1.72	111.88 ± 0.72	181.2 ± 1.48	216.0 ± 0.00
		Cost	52,949 ± 1,850	43,113 ± 1,139	41,026 ± 927	107,021 ± 2,955	87,899 ± 2,687	101,798 ± 1,894
	CO	Days	46.43 ± 5.84	76.45 ± 7.29	114.2 ± 12.11	116.5 ± 5.9	206.36 ± 12.07	219.34.7 ± 6.97
		Cost	45,220 ± 902	36,678 ± 656	34,074 ± 521	94,577 ± 1,586	77,804 ± 1,228	94,218 ± 852
ACO	DO	Days	32 ± 0.0	38.4 ± 1.49	**52.86 ± 1.6**	112 ± 0.0	189.8 ± 2.5	216.24 ± 0.72
		Cost	53,092 ± 1,816	43,271 ± 895	53,092 ± 1,816	104,862 ± 2,928	90,471 ± 2,765	102,075 ± 1,930
	CO	Days	114.06 ± 7.29	127.5 ± 6.5	166.82 ± 0.38	181.52 ± 12.62	252.9 ± 11.93	260.35 ± 8.27
		Cost	30,295 ± 332	26,376 ± 154	30,167 ± 7.21	50,486 ± 1,113	53,110 ± 584	72,767 ± 1,566
HAntCO	DO	Days	**25.1 ± 0.81**	**35.8 ± 1.07**	55.8 ± 0.73	**108.0 ± 0.00**	182.48 ± 5.05	**186.8 ± 2.16**
		Cost	53,527 ± 1,086	44,183 ± 622	56,671 ± 314	104,112 ± 2,217	90,294 ± 3,198	104,510 ± 1,690
	CO	Days	117.0 ± 0.00	128.98 ± 0.13	167.0 ± 0.00	217.1 ± 1.07	341.62 ± 8.02	267.36 ± 1.94
		Cost	**30,104 ± 0.0**	**26,323 ± 3.78**	**30,164 ± 0**	46,554 ± 291	51,514 ±	**71,986 ± 0.00**
H	DO	Days	32 ± 0	38 ± 0	57 ± 0	112 ± 0	183 ± 0	216 ± 0
		Cost	53,154 ± 0	44,309 ± 0	40,539 ± 0	101,879 ± 0	92,821 ± 0	105,593 ± 40
	CO	Days	119 ± 0	129 ± 0	167 ± 0	214 ± 0	337 ± 0	256 ± 0
		Cost	**30,104 ± 0**	**26,323 ± 0**	**30,164 ± 0**	**46,133 ± 0**	51,496 ± 0	**71,986 ± 0**

every case or results are comparable. We developed extra statistical analysis to prove a quality of presented method. We have provided the Kolmogorov–Smirnov (K–S test) to investigate the normality of the distribution of gained results. The K–S test proved that results of used methods are normally distributed and t test can be used. Moreover, a sample size around 50 allows the normality assumptions conducive for performing the unpaired t tests (Flury 1997). We used two tailed t test with 95 % confidence interval (see results in Table 10) for the best and the second best performing methods applied in D1–D6 instances for DO and CO modes.

We found that HAntCO results are the best in most cases. Very interesting results are noticed for EA S, especially for D5 instance (in DO and CO mode) where EA S gives the best (average) solutions.

Only in one case (D3 instance DO mode), ACO gives better average solution. The results are significant in statistical meaning. The statistical significance of results for HAntCO in CO mode comes mostly from the fact that HAntCO is a method directed by a heuristic that finds the best cost-oriented solution (algorithm). Hence, the statistical significance of this method should be mostly investigated in DO mode. In this mode, the results obtained by HAntCO are statistically significant in 3 cases (D1, D2, D6), while DO-oriented results obtained by ACO are statistically significant in only one case (D3). It additionally proves the legitimacy of using proposed hybrid rather than classical ACO approach. We have also investigated results for several methods in BO mode. In this case, classical ACO approach outclassed the rest of examined methods and became the best choice in 5/6 cases. However, it caused enlarging the project schedule duration of analyzed instances and make them mostly the longest from all obtained with various methods. EA with specialized genetic operators gave the smallest project cost for BO mode. It was the best in 5/6 cases. An interesting fact is that the results obtained for ACO are completely different from the results from other methods like TS or EA. The conclusion could be that ACO searches the solution space totally different from the above-mentioned methods. Hence, combining those approaches into one could be possibly effective and potentially give promising results.

Table 10 Results of the unpaired t test between the best and the second best performing methods (for each instances D1–D6) based on Table 8 (heuristic H (Skowroński et al. 2013b) results not included as a part of HAntCO)

Instance	Mode	Best methods	SE	t	95 % CI	Two tailed p	Stat. significance
D1	DO	**HAntCO**, EA S	0.115	60.2350	−7.12 to −6.67	<0.0001	**Extr. significant**
	CO	**HAntCO**, EA S	47.016	4.0574	−284.06 to −97.45	<0.0001	**Extr. significant**
D2	DO	**HAntCO**, EA S	8.072	7.2901	−2.18 to −1.25	<0.0001	**Extr. significant**
	CO	**HAntCO**, TS	0.726	2.6885	−37.71 to −5.68	0.0084	**Very significant**
D3	DO	**ACO**, EA S	0.300	6.0720	−2.41 to −1.22	<0.0001	**Extr. significant**
	CO	**HAntCO**, **ACO**	1.018	3.5355	1.57 to 5.62	0.0006	**Extr. significant**
D4	DO	**HAntCO**, EA C	0.102	38.1052	3.67 to 4.08	<0.0001	**Extr. significant**
	CO	TS, **HAntCO**	44.934	4.3397	−284.16 to −105.83	<0.0001	**Extr. significant**
D5	DO	EA S, EA C	0.299	2.8761	−1.45 to −0.26	0.0049	**Very significant**
	CO	EA S, **HAntCO**	31.673	9.8913	−376.14 to −250.43	<0.0001	**Extr. significant**
D6	DO	**HAntCO**, TS, EA	0.305	95.8523	28.67 to 29.88	<0.0001	**Extr. significant**
	CO	**HAntCO**, **ACO**	221.542	3.5243	341.14 to 1220.43	0.0006	**Extr. significant**

6 Conclusions and further work

In this paper, we have presented hybrid approach for solving multi-skill resource-constrained project scheduling problem. MS-RCPSP is an extension of classical RCPSP with skills and cost domain. Our approach bases on classical ACO metaheuristics for discrete combinatorial problems. However, it has been enhanced by modified pheromone update methods. Furthermore, we have proposed a hybridization of ACO approach (HAnt-CO) using simple heuristics based on priority rules to find an initial solution in optimization process.

What is more, we have prepared and published iMOPSE dataset instances to allow others to investigate their approaches for such defined MS-RCPSP. The dataset consists of 36 instances containing 100 or 200 tasks. All instances are varied by the number of resources, precedence relations and skill types what makes them more or less difficult to be scheduled.

We have also defined evaluation methods for the proposed approaches not only in case of their robustness (how good the final solution is) but also their effectiveness. To evaluate method's quality, we rate not only the project schedule duration, as in classical RCPSP, but also the project schedule performance cost, regarding the MS-RCPSP as multi-objective optimization problem. The method's effectiveness is rated by the number of dominant operations that need to be performed during the optimization process.

Finally, we have compared the results obtained by HAnt-CO and ACO with the ones received with the use of other methods as simple heuristics, Taboo Search and Evolutionary Algorithms with classic and specialized genetic operators that have been published before. The provided results have been also supported by the statistical significance tests. The obtained results lead to the conclusion that ACO-based approaches stand suitable ones for solving MS-RCPSP as they provide mostly the best results from all investigated methods.

6.1 Future work

After observation that pheromone update method in ACO has an impact on the obtained results depending on selected optimization mode (aspect), we are encouraged to use this outcome and propose an approach more dedicated to multi-objective optimization using Pareto front from various ant populations performing in different pheromone update methods. It could provide us a mechanism to find very good solutions leaving the need of setting optimization mode. It could give us good solutions in DO and BO in the same run of ACO-based run.

Pareto-based approach could be implemented in the investigated methods to distinguish non-dominated solutions. By non-dominated solution, a one with the smallest value of given criterion is taken while remaining criterion values are equal. In MS-RCPSP, non-dominated solution is regarded as the one that has smallest cost or duration from subset of solutions with the same duration or cost, respectively. It could make the optimization process more robust and effective, as *good enough* results could be found in a smaller number of iterations within the examined method.

As cost-oriented optimization in ACO and HAntCO has not provided significantly better results than other methods investigated in this paper, we discuss a potential application of dedicated neighborhood definition for ants to make them more oriented to search solutions cheaper.

Investigating the comparison of the results obtained for CO and DO modes could lead to conclusion that ACO is a powerful tool for solving MS-RCPSP, especially if it was boosted by initial solution obtained by heuristic (HAntCO). It leads to conclusion that other hybrids should be investigated using the proposed heuristics. Hence, we would examine and compare the results obtained for EA, TS or SA approaches to check, whether boosting initial solution by heuristic provides better results for other metaheuristics.

According to the experiences with ACO of other researchers, ACO can be extended by additional heuristic (Dorigo 1997) to enhance the potential of optimization. We plan to find suitable, problem-specific heuristic that could be used and investigate whether it could make our approach better in solving MS-RCPSP.

References

Al-Anzi FS, Al-Zamel K, Allahverdi A (2010) Weighted multi-skill resources project scheduling. J Softw Eng Appl 3:1125–1130

Blazewicz J, Lenstra JK, Rinnooy Kan AHG (1983) Scheduling subject to resource constraints: classification and complexity. Discret Appl Math 5:11–24

Bianco L, DellOlmo P, Speranza MG (1998) Heuristics for multimode scheduling problems with dedicated resources. Eur J Oper Res 107:260–271

Bouleimen K, Lecocq H (2003) A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. Eur J Oper Res 149:268–281

Brucker P, Drexl A, Mohring R, Neumann K, Pesch E (1998) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41

Das PP, Acharyya S (2011) Simulated annealing variants for solving resource constrained project scheduling problem: a comparative study. In: Proceedings of 14th international conference on computer and information technology, pp 469–474

Dorigo M (1997) Ant Colony System: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66

Drezet LE, Billaut JC (2008) A project scheduling problem with labour constraints and time-dependent activities requirements. Int J Prod Econ 112:217–225

Flury B (1997) A first course in multivariate statistics, vol 28. Springer, Berlin

Gonzalez F, Ramies Rios D (2013) Multi-objective optimization of the resource constrained project scheduling problem (RCPSP) a heuristic approach based on the mathematical model. Int J Comput Sci Appl 2(2):1–13

Hartmann S (1998) A competitive genetic algorithm for resource-constrained project scheduling. Nav Res Logist 45:733–750

Hegazy T, Shabeeb AK, Elbeltagi E, Cheema T (2000) Algorithm for scheduling with multiskilled constrained resources. J Constr Eng Manag (11–12/2000):414–421

Jaberi M, Jaberi M (2014) A multi-objective resource-constrained project-scheduling problem using mean field annealing neural networks. J Math Comput Sci 9:228–239

Kadrou Y, Najid NM (2006) A new heuristic to solve RCPSP with multiple execution modes and multi-skilled labor. IMACS multiconference on computational engineering in systems applications (CESA), pp 1302–1309

Kolisch R, Sprecher A (1996) PSPLIB—a project scheduling problem library. Eur J Oper Res 96:205–216

Kolisch R, Hartmann S (2000) Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. Eur J Oper Res 127:394–407

Kolisch R, Hartmann S (2006) Experimental investigation of heuristics for resource-constrained project scheduling: an update. Eur J Oper Res 174:23–37

Li H, Womer K (2009) Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. J Sched 12:281–298

Liang Y, Chen A, Kao W, Chyu C (2004) An Ant Colony approach to resource-constrained project scheduling problems. In: Proceedings of the fifth Asia Pacific industrial engineering and management systems conference 2004, pp 31.5.1–31.5.10

Luna F, Gonzalez-Alvarez DL, Chicano F, Vega-Rodriguez MA (2013) The software project scheduling problem: a scalability analysis of multi-objective metaheuristics. Appl Soft Comput 15:136–148

Luo S, Wang C, Wang J (2003) Ant Colony Optimization for resource-constrained project scheduling with generalized precedence relations. In: Proceedings of the 15th IEEE international conference on tools with artificial intelligence (ICTAI03), pp 284–289

Merkle D, Mittendorf M, Schmeck H (2002) Ant Colony Optimization for resource-constrained project scheduling. IEEE Trans Evol Comput 6(4):333–346

Pan HI, Hsaio PW, Chen KY (2008) A study of project scheduling optimization using Tabu Search algorithm. Eng Appl Artif Intell 21:1101–1112

Pan NH, Lee ML, Chen KY (2009) Improved Tabu Search algorithm application in RCPSP. In: Proceedings of the international multiconference of engineers and computer scientists, vol I

Phruksaphanrat B (2014) Multi-objective multi-mode resource-constrained project scheduling problem by preemptive fuzzy goal programming. World Acad Sci Eng Technol/Int J Mech Ind Sci Eng 8(3):99–103

Santos M, Tereso AP (2011) On the multi-mode, multi-skill resource constrained project scheduling problem - computational results. Soft Comput Ind Appl, Adv Intell Soft Comput 96:239–248

Skowroński ME, Myszkowski PB (2013) Specialized genetic operators for multi-skill resource-constrained project scheduling problem. In: 19th international conference on soft computing mendel, pp 57–62

Skowroński ME, Myszkowski PB, Kwiatek P, Adamski M (2013) Tabu Search approach for multi-skill resource-constrained project scheduling problem. Ann Comput Sci Inf Syst 1. Proceedings of the 2013 FedCSIS, pp 153–158

Skowroński ME, Myszkowski PB, Podlodowski Ł (2013) Novel heuristic solutions for multi-skill resource-constrained project scheduling problem. Ann Comput Sci Inf Syst 1. Proceedings of the 2013 FedCSIS, pp 159–166

Stutzle T, Hoos H (2000) Max–Min Ant System. Future Gener Comput Syst 16:889–914

Thomas PR, Salhi S (1998) A Tabu Search approach for the resource constrained project scheduling problem. J Heuristics 4:123–139

Tsai YW, Gemmill DD (1998) Using Tabu Search to schedule activities of stochastic resource-constrained projects. Eur J Oper Res 111:129–141

Valls V, Ballestin F, Quintanilla S (2008) A hybrid genetic algorithm for the resource-constrained project scheduling problem. Eur J Oper Res 185:495–508

Valls V, Ballestin F, Quinanilla S (2001) An evolutionary approach to the resource-constrained project scheduling problem. In: 4th metaheuristics international conference 2001, MIC2001, pp 217–220

Verhoeven MGA (1998) Tabu Search for resource-constrained scheduling. Eur J Oper Res 106:266–276

Wang H, Lin D, Li MQ (2005) A competitive genetic algorithm for resource-constrained project scheduling problem. In: Proceedings of the fourth international conference on machine learning and cybernetics, pp 2945–2949

Watson J, Ross C, Eisele V, Denton J, Bins J, Guerra C, Whitley D, Howe A (1998) The traveling salesrep problem, edge assembly crossover, and 2-opt. Lecture Notes in Computer Science, vol 1498, pp 823–832

Yannibelli V, Amandi A (2013) Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. Expert Syst Appl 40:2421–2434

Zhang H, Xu H, Peng W (2008) A genetic algorithm for solving RCPSP. In: 2008 international symposium on computer science and computational technology, pp 246–249

Zhang H, Li H, Tam C (2006) Particle swarm optimization for resource-constrained project scheduling. Int J Proj Manag 24:83–92

Zhang H, Li X, Li H, Hang F (2005) Particle swarm optimization-based schemes for resource-constrained project scheduling. Autom Constr 14:393–404

Zhang K, Zhao G, Jiang J (2009) Particle swarm optimization method for resource-constrained project scheduling problem. In: The ninth international conferenc on electronic measurement & instruments, ICEMI2009, pp 792–796

Zhou Y, Guo Q, Gan R (2009) Improved ACO algorithm for resource-constrained project scheduling problem. In: 2009 international conference on artificial intelligence and computational intelligence, pp 358–365

Zhou L, Wang D, Peng W (2008) An ACO for solving RCPSP. In: 2008 international symposium on computer science and computational technology, pp 250–253

Ziarati K, Akbari R, Zeighami V (2011) On the performance of bee algorithms for resource-constrained project scheduling problem. Appl Soft Comput 11:3720–3733