# An Adaptive Hybrid Evolutionary Approach for a Project Scheduling Problem that Maximizes the Effectiveness of Human Resources

Virginia Yannibelli[1,2](✉)

[1] ISISTAN Research Institute, UNCPBA University, Campus Universitario,
Paraje Arroyo Seco, 7000 Tandil, Argentina
virginia.yannibelli@isistan.unicen.edu.ar
[2] CONICET, National Council of Scientific and Technological Research,
Buenos Aires, Argentina

**Abstract.** In this paper, an adaptive hybrid evolutionary algorithm is proposed to solve a project scheduling problem. This problem considers a valuable optimization objective for project managers. This objective is maximizing the effectiveness of the sets of human resources assigned to the project activities. The adaptive hybrid evolutionary algorithm utilizes adaptive processes to develop the different stages of the evolutionary cycle (i.e., adaptive parent selection, survival selection, crossover, mutation and simulated annealing processes). These processes adapt their behavior according to the diversity of the algorithm's population. The utilization of these processes is meant to enhance the evolutionary search. The performance of the adaptive hybrid evolutionary algorithm is evaluated on six instance sets with different complexity levels, and then is compared with those of the algorithms previously reported in the literature for the addressed problem. The obtained results indicate that the adaptive hybrid evolutionary algorithm significantly outperforms the algorithms previously reported.

**Keywords:** Project scheduling · Human resource assignment
Multi-skilled resources · Evolutionary algorithms
Adaptive evolutionary algorithms · Hybrid evolutionary algorithms

## 1 Introduction

Project scheduling is a highly relevant and complex issue in most real-world organizations [1, 2]. Project scheduling generally involves defining feasible start times and human resource assignments for the project activities, such that a given optimization objective is achieved. Moreover, to define human resource assignments, it is necessary to consider the available knowledge about the effectiveness of human resources respecting the project activities. This is important since the development and the results of project activities mainly depend on the effectiveness of the human resources assigned to such activities [1, 2].

I. Batyrshin et al. (Eds.): MICAI 2018, LNAI 11288, pp. 33–49, 2018.
https://doi.org/10.1007/978-3-030-04491-6_3

In the past four decades, many kinds of project scheduling problems have been formally presented and addressed in the literature. However, to the best of the author's knowledge, only few project scheduling problems consider human resources with different effectiveness levels [3, 4], a very important aspect of real-world project scheduling. These project scheduling problems suppose very different assumptions in respect of the effectiveness of human resources.

The project scheduling problem presented in [5] supposes that the effectiveness level of a human resource depends on several factors inherent to its work context (i.e., the project activity to which the resource is assigned, the skill to which the resource is assigned within the project activity, the set of human resources assigned to the project activity, and the attributes of the resource). This assumption about the effectiveness of human resources is really valuable. This is because, in real-world project scheduling problems, human resources generally have different effectiveness levels with respect of different work contexts, and therefore, the effectiveness level of a human resource is considered with respect of the factors inherent to its work context [1, 2]. To the best of the author's knowledge, the influence of the work context on the effectiveness level of human resources is not considered in other project scheduling problems presented in the literature. The problem presented in [5] also considers a valuable optimization objective for project managers: maximizing the effectiveness of the sets of human resources assigned to the project activities.

The project scheduling problem presented in [5] is a variant of the known RCPSP (Resource Constrained Project Scheduling Problem) [6] and, therefore, is an NP-Hard optimization problem. Because of this, heuristic search and optimization algorithms are required to solve different problem instances in an acceptable amount of time. In this respect, to the best of the author's knowledge, four heuristic search and optimization algorithms have been presented so far in the literature to solve this problem. Specifically, a traditional evolutionary algorithm was presented in [5]. In [7], a traditional memetic algorithm was presented which includes a hill-climbing algorithm into the framework of an evolutionary algorithm. In [8], a hybrid evolutionary algorithm was presented which incorporates an adaptive simulated annealing algorithm within the framework of an evolutionary algorithm. In [9–11], a hybrid evolutionary algorithm was presented which utilizes semi-adaptive crossover and mutation processes, and an adaptive simulated annealing algorithm.

These four algorithms follow the stages of the traditional evolutionary cycle (i.e., parent selection, crossover, mutation, and survival selection stages) to develop the evolutionary search. Nevertheless, these algorithms use non-adaptive processes to carry out all or many of the stages of the evolutionary cycle. In this respect, the four algorithms use non-adaptive parent selection and survival selection processes, and the first three of these algorithms use non-adaptive crossover and mutation processes. The fourth of these algorithms uses semi-adaptive crossover and mutation processes; however, the adaptability of the behavior of these processes during the evolutionary search is very limited.

In this paper, the project scheduling problem presented in [5] is addressed with the aim of proposing a better heuristic search and optimization algorithm to solve it. In this respect, an adaptive hybrid evolutionary algorithm is proposed which utilizes adaptive processes to develop the different stages of the evolutionary cycle (i.e., adaptive parent selection, survival selection, crossover, mutation and simulated annealing processes). These processes adapt their behavior based on the diversity of the algorithm's population. The utilization of these adaptive processes is meant for improving the performance of the evolutionary search [12–14].

The above-mentioned adaptive hybrid evolutionary algorithm is proposed mainly because of the following reasons. Evolutionary algorithms with adaptive selection, crossover and mutation processes have been proven to be more effective than evolutionary algorithms with non-adaptive selection, crossover and mutation processes in the resolution of a wide variety of NP-Hard optimization problems [12–17]. Thus, the proposed adaptive hybrid evolutionary algorithm could outperform the heuristic search and optimization algorithms presented so far in the literature to solve the addressed problem.

The remainder of the paper is organized as follows. Section 2 presents a brief review of project scheduling problems reported in the literature which consider the effectiveness of human resources. Section 3 describes the project scheduling problem addressed. Section 4 presents the adaptive hybrid evolutionary algorithm proposed for the problem. Section 5 presents the computational experiments developed to evaluate the performance of the adaptive hybrid evolutionary algorithm and also an analysis of the results obtained. Finally, Sect. 6 presents the conclusions of the present work.

## 2 Related Works

Over the past four decades, different kinds of project scheduling problems which consider the effectiveness of human resources have been presented in the literature [3, 4]. However, these project scheduling problems suppose very different assumptions concerning the effectiveness of human resources. In this regards, to the best of the author's knowledge, only few project scheduling problems consider human resources with different effectiveness levels [3, 4], a very important aspect of real-world project scheduling problems. In this section, the focus is on reviewing the main assumptions about the effectiveness that have been considered in project scheduling problems presented in the literature.

In the multi-skill project scheduling problems presented in [19–23], project activities require a given number of skills for their development, and a given number of human resources for each skill required. The human resources available for the project activities master one or several skills. These problems suppose that the human resources that master a given skill have the same effectiveness level in respect of such skill.

In the skilled workforce project scheduling problems presented in [24–26], each project activity requires only one human resource with a given skill. Moreover, the human resources available for project activities master one or several skills. These problems suppose that the human resources which master a given skill have the same effectiveness level in relation to such skill.

The multi-skill project scheduling problem reported in [27] considers hierarchical levels of skills. In this regard, the problem supposes that the human resources which master a given skill have different effectiveness levels in respect of such skill. Moreover, the project activities of this problem require a given number of skills for their development, a given minimum level of effectiveness for each one of the skills required, and a given number of human resources for each pair skill-level. Then, this problem supposes that the human resource sets feasible for a given project activity have the same effectiveness level with respect to the development of such activity.

In the multi-skill project scheduling problems presented in [28–30], most project activities require only one human resource with a given skill. The human resources available for project activities master one or several skills. These problems suppose that the human resources which master a given skill have different effectiveness levels in respect of such skill. In addition, these problems suppose that the effectiveness level of a human resource in a given project activity only depends on the effectiveness level of the human resource with respect to the skill required for the activity.

In contrast with the project scheduling problems previously mentioned, the project scheduling problem presented in [5] supposes that the effectiveness level of a human resource depends on several factors inherent to its work context. Then, different effectiveness levels can be defined for each human resource regarding different work contexts. This assumption about the effectiveness of human resources is really important. This is because, in the context of real-world project scheduling problems, human resources have very different effectiveness levels in respect of different work contexts, and therefore, the effectiveness level of a human resource is considered in respect of the factors inherent to its work context [1, 2]. To the best of the author's knowledge, the influence of the work context on the effectiveness level of human resources is not considered in other project scheduling problems presented in the literature. Based on the mentioned above, the project scheduling problem presented in [5] supposes a valuable and novel assumption concerning the effectiveness level of human resources in the context of project scheduling problems.

## 3 Project Scheduling Problem Description

In this paper, the project scheduling problem introduced in [5] is addressed. A description of this problem is presented below.

Suppose that a project contains a set $A$ of $N$ activities, $A = \{1, \ldots, N\}$, that has to be scheduled. Specifically, a starting time and a human resource set have to be defined for each project activity of the set $A$. The duration, human resource requirements, and precedence relations of each project activity are known.

The duration of each project activity $j$ is notated as $d_j$. Besides, it is considered that pre-emption of project activities is not allowed. This means that, when a project activity starts, it must be developed period by period until it is completed. Specifically, the $d_j$ periods of time must be consecutive.

Among the project activities, there are precedence relations. This is because usually each project activity requires results generated by other project activities. Thus, the precedence relations establish that each project activity $j$ cannot start until all its immediate predecessors, given by the set $P_j$, have completely finished.

To be developed, project activities require human resources skilled in different knowledge areas. Specifically, each project activity requires one or several skills and also a given number of human resources for each skill required.

It is considered that qualified workforce is available to develop the activities of the project. This workforce is made up of a number of human resources, and each human resource masters one or several skills.

Set $SK$ contains the $K$ skills required in order to develop the activities of the project, $SK = \{1, \ldots, K\}$, and set $AR_k$ contains the available human resources with skill $k$. Then, the term $r_{j,k}$ represents the number of human resources with skill $k$ required for activity $j$ of the project. The values of the terms $r_{j,k}$ are known for each project activity.

It is considered that a human resource cannot take over more than one skill within a given activity, and also a human resource cannot be assigned more than one activity at the same time.

Based on the assumptions previously mentioned, a human resource can be assigned different project activities but not at the same time, can take over different skills required for a project activity but not simultaneously, and can belong to different possible sets of human resources for each activity.

Therefore, different work contexts can be defined for each available human resource. It is considered that the work context of a human resource $r$, denoted as $C_{r,j,k,g}$, is made up of four main components. In this respect, the first component refers to the project activity $j$ which $r$ is assigned (i.e., the complexity of $j$, the domain of $j$, etc.). The second component refers to the skill $k$ which $r$ is assigned within project activity $j$ (i.e., the tasks associated to $k$ within $j$). The third component is the set of human resources $g$ that has been assigned $j$ and that includes $r$ (i.e., $r$ must work in collaboration with the other human resources assigned to $j$). The fourth component refers to the attributes of $r$ (i.e., his or her educational level regarding different knowledge areas, his or her level regarding different skills, his or her experience level regarding different tasks and domains, the kind of labor relation between $r$ and the other human resources of $g$, etc.). In respect of the attributes of $r$, it is considered that these attributes could be quantified from available information about $r$ (e.g., curriculum vitae of $r$, results obtained from evaluations made to $r$, information about the participation of $r$ in already executed projects, etc.).

The four components previously mentioned are considered the main factors that determine the effectiveness level of a human resource. Because of this, it is assumed that the effectiveness level of a human resource depends on all the components of his or her work context. Then, different effectiveness levels can be considered for each human resource in respect of different work contexts.

The effectiveness level of a human resource $r$, in respect of a possible context $C_{r,j,k,g}$ for $r$, is notated as $e_{rC_{r,j,k,g}}$. The term $e_{rC_{r,j,k,g}}$ refers to how well $r$ can take over, within activity $j$, the tasks associated to skill $k$, considering that $r$ must work in collaboration with the other human resources of set $g$. The term $e_{rC_{r,j,k,g}}$ takes a real value over the range [0, 1]. The values of the terms $e_{rC_{r,j,k,g}}$ inherent to each human resource available for the project are known. It is considered that these values could be obtained from available information regarding the participation of the human resources in already carried out projects.

The problem of scheduling a project involves to determine feasible start times (i.e., the precedence relations among the project activities must not be violated) and feasible human resource assignments (i.e., the human resource requirements of project activities must be met) for project activities such that the optimization objective is achieved. In this respect, an optimization objective valuable for project managers is considered. This optimization objective implies maximizing the effectiveness of the sets of human resources assigned to the project activities. This objective is modeled by Formulas (1) and (2).

Formula (1) maximizes the effectiveness of the sets of human resources assigned to the *N* project activities. In this formula, set *S* contains all the feasible schedules for the project in question. The term *e(s)* refers to the effectiveness level of the sets of human resources assigned to the project activities by schedule *s*. The term *R(j, s)* refers to the set of human resources assigned to activity *j* by schedule *s*. The term $e_{R(j,s)}$ refers to the effectiveness level corresponding to *R(j, s)*.

Formula (2) estimates the effectiveness level of the set of human resources *R(j,s)*. This effectiveness level is estimated by calculating the mean effectiveness level of the human resources belonging to *R(j, s)*.

For a more detailed discussion of the project scheduling problem described here and, in particular, of Formulas (1) and (2), the readers are referred to the work [5] which has introduced this problem.

$$\max_{\forall s \in S} \left( e(s) = \sum_{j=1}^{N} e_{R(j,s)} \right) \tag{1}$$

$$e_{R(j,s)} = \frac{\sum_{r=1}^{|R(j,s)|} e_{rC_{r,j,k(r,j,s),R(j,s)}}}{|R(j,s)|} \tag{2}$$

## 4 Adaptive Hybrid Evolutionary Algorithm

To solve the addressed problem, an adaptive hybrid evolutionary algorithm is proposed. This algorithm utilizes adaptive processes to develop the different stages of the evolutionary cycle. These processes adapt their behavior according to the diversity of the evolutionary algorithm population, to promote either the exploration or exploitation of the search space. The utilization of these adaptive processes aims to enhance the performance of the evolutionary search [12–14].

The general behavior of the algorithm is described as follow. This algorithm is an iterative or generational process. This process starts from an initial population of solutions. Each solution encodes a feasible schedule for the project to be scheduled. Besides, each solution has a fitness value that represents the quality of the related schedule in respect of the optimization objective of the addressed problem. As mentioned in Sect. 3, such objective implies maximizing the effectiveness of the sets of human resources assigned to the project activities. The iterative process ends when a predefined number of iterations or generations is reached. Once this happens, the iterative process provides the best solution of the last population as a solution to the problem.

In each iteration, the algorithm develops the following stages. First, an adaptive parent selection process is applied in order to determine which solutions of the current population will compose the mating pool. Once the mating pool is composed, the solutions in the mating pool are paired, and a crossover process is applied to each pair of solutions with an adaptive crossover probability in order to generate new feasible ones. Then, a mutation process is applied to each solution generated by the crossover process, with an adaptive mutation probability. The mutation process is applied in order to introduce diversity in the new solutions generated by the crossover process. Then, an adaptive survival selection process is applied to create a new population from the solutions in the current population and the new solutions generated by crossover and mutation. Finally, an adaptive simulated annealing algorithm is applied to each solution of the new population, excepting the best solution of this population. The best solution remains in the population. Thus, the adaptive simulated annealing algorithm modifies the solutions of the new population.

## 4.1 Encoding of Solutions

In order to encode the solutions of the population, the encoding introduced in [5] for project schedules was used. By using this encoding, each solution is encoded by two lists with a length equal to $N$, considering that $N$ is the number of activities in the project to be scheduled.

The first list is a traditional activity list. Each position on this list contains a different activity $j$ of the project. Each activity $j$ of the project can appear on this list in any position higher than the positions of all its predecessor activities. The activity list represents a feasible order in which the activities of the project can be added to the schedule.

The second list is an assigned resources list. This list contains information about the human resources of each skill $k$ assigned to each activity of the project. Specifically, position $j$ on this list contains a detail about the human resources of each skill $k$ assigned to activity $j$ of the project.

To decode or build the schedule related to the encoding previously described, the serial schedule generation method presented in [5] was used. By this method, each activity $j$ is scheduled at the earliest possible time.

In order to generate the encoded solutions of the initial population according to the encoding previously described, the random generation process introduced in [5] was used. By using this process, a very diverse initial population is obtained. This is meant in order to avoid the premature convergence of the evolutionary search [12].

## 4.2 Fitness Function

To evaluate the encoded solutions, a fitness function specially designed was used. Considering a given encoded solution, the fitness function decodes the schedule $s$ from the solution by using the serial schedule generation method mentioned in Sect. 4.1. Then, the fitness function calculates the value of the term $e(s)$ corresponding to $s$ (Formulas (1) and (2)). This value defines the fitness value of the solution. Note that the term $e(s)$ takes a real value over the range [0, …, $N$].

To calculate the value of term $e(s)$, the fitness function uses the values of the terms $e_{rCr,j,k,g}$ inherent to $s$ (Formula 2). As mentioned in Sect. 3, the values of the terms $e_{rCr,j,k,g}$ inherent to each available human resource $r$ are known.

## 4.3 Adaptive Parent Selection Process

To develop the parent selection on the current population, an adaptive tournament selection process was defined. This process is an adaptive variant of the well-known tournament selection process with replacement [12].

In this process, the tournament size $T$ is defined by Formula (3), where $PD$ refers to the diversity of the current population, and $PD_{MAX}$ refers to the maximum $PD$ attainable. Then, $T^H$ and $T^L$ refer to the upper and lower bounds for the tournament size, respectively.

The term $PD$ is defined by Formula (4), where $f_{max}$ is the maximal fitness of the current population, $f_{avg}$ is the average fitness of the current population, and $(f_{max} - f_{avg})$ is a measure of the diversity of the current population. This measure has been proposed by Srinivas and Patnaik [31], and is one of the population diversity measures most well-known in the literature [12].

The term $PD_{MAX}$ is defined by Formula (5), where $f_{MAX}$ and $f_{MIN}$ represent to the maximum and minimum fitness values attainable, respectively. Note that $f_{MAX}$ and $f_{MIN}$ correspond to the upper and lower bounds of the fitness function described in Sect. 4.2.

By Formula (3), the tournament size $T$ is adaptive according to the diversity of the current population. Specifically, when the population is very diverse, $T$ is increased, promoting the selection of the solutions with high fitness values. This favors the exploitation of the search space. When the diversity of the population reduces, $T$ is decreased, increasing the selection chances of the solutions with low fitness values. This is meant to preserve the diversity of the population and thus to favor the exploration of the search space, with the aim of avoiding the premature convergence of the evolutionary search.

$$T = \left\lceil \frac{PD}{PD_{MAX}} * \left(T^H - T^L\right) + T^L \right\rceil \tag{3}$$

$$PD = \left(f_{max} - f_{avg}\right) \tag{4}$$

$$PD_{MAX} = \left(f_{MAX} - f_{MIN}\right) \tag{5}$$

### 4.4 Adaptive Crossover and Adaptive Mutation Processes

In relation to the crossover process and the mutation process, processes feasible for the encoding of solutions were defined.

The crossover process is composed by a crossover operator feasible for activity lists and a crossover operator feasible for assigned resources lists. Regarding the crossover for activity lists, the one-point crossover operator for activity lists [18] was applied. Regarding the crossover for assigned resources lists, the uniform crossover operator was applied [12].

The mutation process is composed by a mutation operator feasible for activity lists and a mutation operator feasible for assigned resources lists. In relation to the mutation for activity lists, a variant of the simple shift operator for activity lists [18] was applied. In relation to the mutation for assigned resources lists, the random resetting operator [12] was applied.

These crossover and mutation processes are applied with adaptive crossover and mutation probabilities, respectively. In this regards, an adaptive crossover probability $AP_c$ and an adaptive mutation probability $AP_m$ were defined by Formulas (6)–(7). In these formulas, $PD$ refers to the diversity of the current population, and $PD_{MAX}$ refers to the maximum $PD$ attainable, as was mentioned in Sect. 4.3. In Formula (6), the terms $C^H$ and $C^L$ represent to the upper and lower bounds for the crossover probability, respectively. In Formula (7), the terms $M^H$ and $M^L$ represent to the upper and lower bounds for the mutation probability, respectively. The term $f_{max}$ is the maximal fitness of the population, $f_{min}$ is the minimal fitness of the population, and $f$ is the fitness of the solution to be mutated.

By Formula (6)–(7), $AP_c$ and $AP_m$ are adaptive according to the diversity of the current population. In this respect, when the diversity of the population reduces, $AP_c$ and $AP_m$ are increased, promoting the exploration of the search space. This is important for preventing the premature convergence of the evolutionary search. When the population is very diverse, $AP_c$ and $AP_m$ are decreased, promoting the exploitation of the search space. Therefore, probabilities $AP_c$ and $AP_m$ are adaptive according to the diversity of the population, to promote either the exploitation or exploration of the search space.

By Formula (7), $AP_m$ is also adaptive according to the fitness of the solution to be mutated. In this respect, lower values of $AP_m$ are defined for high-fitness solutions, and higher values of $AP_m$ are defined for low-fitness solutions. This is meant in order to preserve high-fitness solutions, while disrupting low-fitness solutions to promote the exploration of the search space.

$$AP_c = \left(\frac{PD_{MAX} - PD}{PD_{MAX}}\right) * \left(C^H - C^L\right) + C^L \tag{6}$$

$$AP_m = \left(\frac{f_{max} - f}{f_{max} - f_{min}}\right) * \left(\frac{PD_{MAX} - PD}{PD_{MAX}}\right) * \left(M^H - M^L\right) + M^L \tag{7}$$

## 4.5 Adaptive Survival Selection Process

The survival selection process is applied to create a new population from the solutions in the current population (parent solutions) and the new solutions generated by crossover and mutation (offspring solutions).

To develop the survival selection, an adaptive deterministic crowding process was defined. This process is an adaptive variant of the well-known deterministic crowding process [12, 32].

In this process, offspring solutions compete with their respective parent solutions to be included in the new population. Specifically, each offspring competes with its closest parent, considering that the closeness between an offspring solution and a parent solution is defined based on the distance between their fitness values. When the fitness value of an offspring solution is better than that of its closest parent solution, the offspring solution is accepted for the new population. Otherwise, when the fitness value of an offspring solution is not better than that of its closest parent solution, the offspring solution is accepted for the new population with a probability $p_{offspring}$. If the offspring solution is not accepted, the parent solution is accepted for the new population directly.

The probability $p_{offspring}$ is defined as follows: $exp(-\Delta_{DC}/T)$, where $\Delta_{DC}$ is the difference between the fitness value of the parent solution and the fitness value of the offspring solution, and the term $T$ is defined according to the diversity of the current population. Specifically, $T$ is inversely proportional to the diversity of the current population, and is calculated as follows: $T = 1/PD$, where $PD$ refers to the diversity of the current population, as was mentioned in Sect. 4.3.

By the definition of the probability $p_{offspring}$, this probability is adaptive based on the diversity of the current population. In this respect, when the population is very diverse, the value of $T$ is very low, and therefore the probability $p_{offspring}$ of the process is also low. Thus, the process preserves the best solutions for the new population, favoring the exploitation of the search space. When the diversity of the population decreases, the value of $T$ increases, and therefore the probability $p_{offspring}$ of the process also increases. Thus, the process introduces diversity into the new population, favoring the exploration of the search space. This is important to prevent the premature convergence of the evolutionary search.

## 4.6 Adaptive Simulated Annealing Algorithm

After obtaining a new population by the survival selection process, an adaptive simulated annealing algorithm was applied to each solution of this population, except to the best solution of this population which is maintained. This adaptive simulated annealing algorithm is a variant of the one presented in [9–11], and is described below.

The adaptive simulated annealing algorithm is an iterative process. This process starts from a given encoded solution $s$, and a given initial value $T_0$ for the temperature parameter. The iterative process ends when a given number of iterations $I$ is reached, or the current value $T_i$ of the temperature parameter is lower than or equal to 0. After this happens, the solution obtained by the process is provided.

In each iteration, the process generates a new encoded solution *s'* from the current encoded solution *s* by applying a move operator. Then, the process analyzes if the current solution *s* should be replaced by the new solution *s'*. When the fitness value of the current solution *s* is worse than that of the new solution *s'*, the process replaces to the solution *s* by the new solution *s'*. Otherwise, when the fitness value of the current solution *s* is better than or equal to that of the new solution *s'*, the process replaces to the solution *s* by the new solution *s'* with a probability $p_{new\_solution}$. This probability is defined as follows: $p_{new\_solution} = exp(-\Delta_{SA}/T_i)$, where $\Delta_{SA}$ is the difference between the fitness value of the current solution *s* and the fitness value of the new solution *s'*, and $T_i$ is the current value of the temperature parameter. The probability $p_{new\_solution}$ mainly depends on the current value $T_i$ of the temperature parameter. If $T_i$ is high, $p_{new\_solution}$ is also high, and if $T_i$ is low, $p_{new\_solution}$ is also low. The value $T_i$ of the temperature is reduced by a given cooling factor $\alpha$ at the end of each iteration.

The initial value $T_0$ of the temperature parameter is defined before applying the simulated annealing algorithm to the solutions of the population. In this case, the value $T_0$ is defined according to the diversity of the population. In particular, $T_0$ is inversely proportional to the diversity of the population, and is calculated as follows: $T_0 = 1/PD$, where *PD* refers to the diversity of the population, as mentioned in Sect. 4.3. By this definition of $T_0$, when the population is diverse, the value $T_0$ is low, and therefore the probability $p_{new\_solution}$ of the algorithm is also low. Thus, the algorithm fine-tunes the solutions of the population, promoting the exploitation of the search space. When the diversity of the population reduces, the value $T_0$ increases, and therefore the probability $p_{new\_solution}$ of the algorithm also increases. Thus, the algorithm introduces diversity into the population, promoting the exploration of the search space. This is important for avoiding the premature convergence of the evolutionary search. Based on the above-mentioned, the algorithm is adaptive according to the population diversity, in order to promote either the exploitation or exploration of the search space.

This simulated annealing algorithm utilizes a move operator in order to generate a new encoded solution from a given encoded solution. In this respect, a move operator feasible for the encoding of solutions was defined. This move operator is composed by a move operator feasible for activity lists and a move operator feasible for assigned resources lists. In respect of the move operator for activity lists, the adjacent pairwise interchange operator [18] was applied. For assigned resources lists, an operator which is a variant of the random resetting operator [12] was applied.

## 5 Computational Experiments

To develop the computational experiments, the six instance sets introduced in [7] were used. Table 1 presents the main characteristics of these six instance sets. Each instance set contains 40 instances. Each instance of these six instance sets contains information about a number of activities to be scheduled, and information about a number of available human resources for developing these activities. For a more detailed description of these instance sets, the readers are referred to [7].

Each instance of these six instance sets has a known optimal solution with a fitness value *e(s)* equal to *N* (*N* refers to the number of activities in the instance). These know optimal solutions are considered here as references to evaluate the performance of the adaptive hybrid evolutionary algorithm.

The adaptive hybrid evolutionary algorithm was evaluated on the six instance sets. Specifically, the algorithm was run a number of 40 times on each instance of the six instance sets. In order to develop these runs, the parameter setting detailed in Table 2 was used. It is necessary to mention that such parameter setting was defined based on exhaustive preliminary experiments that showed that this setting led to the best and most stable results.

**Table 1.** Main characteristics of the instance sets introduced in [7].

| Instance set | Activities per instance | Possible sets of human resources per activity | Instances |
|---|---|---|---|
| j30_5 | 30 | 1 to 5 | 40 |
| j30_10 | 30 | 1 to 10 | 40 |
| j60_5 | 60 | 1 to 5 | 40 |
| j60_10 | 60 | 1 to 10 | 40 |
| j120_5 | 120 | 1 to 5 | 40 |
| j120_10 | 120 | 1 to 10 | 40 |

**Table 2.** Parameter setting of the adaptive hybrid evolutionary algorithm

| Parameter | Value |
|---|---|
| Population size | 90 |
| Number of generations | 300 |
| Parent selection process | |
| $T^H$ | 10 |
| $T^L$ | 4 |
| Crossover process | |
| $C^H$ | 0.9 |
| $C^L$ | 0.6 |
| Mutation process | |
| $M^H$ | 0.3 |
| $M^L$ | 0.01 |
| Simulated annealing algorithm | |
| *I* (number of iterations) | 25 |
| α (cooling factor) | 0.9 |

Table 3 presents the results obtained by the adaptive hybrid evolutionary algorithm for each of the six instance sets. The second column presents the average percentage deviation from the optimal value (Av. Dev. (%)) for each instance set. The third column presents the percentage of instances for which the optimal value was reached at least once among the 40 runs developed (Opt. (%)).

For the instance sets j30_5, j30_10, j60_5, j60_10 and j120_5 (i.e., the five less complex instance sets), the algorithm obtained Av. Dev. (%) values equal to 0% and Opt. (%) values equal to 100%. These results indicate that the algorithm has reached an optimal solution in each of the 40 runs developed on each instance of these sets.

For the instance set j120_10, the algorithm obtained an Av. Dev. (%) value equal to 0.01%. Considering that the instances of j120_10 have known optimal solutions with a fitness value *e(s)* equal to 120, this result indicates that the average fitness value of the solutions obtained by the algorithm is 119.99. Therefore, the algorithm has reached very high-fitness solutions for the instances of j120_10. Moreover, the algorithm obtained an Opt. (%) value equal to 100% for j120_10. This result indicates that the algorithm has reached an optimal solution at least once among the 40 runs developed on each instance of j120_10.

**Table 3.** Results obtained by the adaptive hybrid evolutionary algorithm

| Instance set | Av. Dev. (%) | Opt. (%) |
|---|---|---|
| j30_5 | 0 | 100 |
| j30_10 | 0 | 100 |
| j60_5 | 0 | 100 |
| j60_10 | 0 | 100 |
| j120_5 | 0 | 100 |
| j120_10 | 0.01 | 100 |

### 5.1 Comparison with Competing Heuristic Algorithms

To the best of the author's knowledge, four heuristic search and optimization algorithms have been presented so far in the literature to solve the addressed problem: a traditional evolutionary algorithm [5], a traditional memetic algorithm [7] which incorporates a hill-climbing algorithm into the framework of an evolutionary algorithm, a hybrid evolutionary algorithm [8] which integrates an adaptive simulated annealing algorithm into the framework of an evolutionary algorithm, and a hybrid evolutionary algorithm [9–11] which utilizes semi-adaptive crossover and mutation processes as well as an adaptive simulated annealing algorithm.

The four algorithms above-mentioned utilize non-adaptive parent selection and survival selection processes to develop the evolutionary search. Besides, the first three of these algorithms use non-adaptive crossover and mutation processes. The fourth of these algorithms uses semi-adaptive crossover and mutation processes; however, the adaptability of the behavior of these processes during the evolutionary search is very limited.

In [7–9], the four algorithms above-mentioned have been evaluated on the six instance sets presented in Table 1. The results obtained by each of the four algorithms for these six instance sets are detailed in Table 4, as were reported in [7–9].

**Table 4.** Results obtained by the heuristic algorithms reported in the literature for the addressed problem

| Instance set | Evolutionary algorithm [5] | | Memetic algorithm [7] | | Hybrid evolutionary algorithm [8] | | Hybrid evolutionary algorithm [9–11] | |
|---|---|---|---|---|---|---|---|---|
| | Av. Dev. (%) | Opt. (%) | Av. Dev. (%) | Opt. (%) | Av. Dev. (%) | Opt. (%) | Av. Dev. (%) | Opt. (%) |
| j30_5 | 0 | 100 | 0 | 100 | 0 | 100 | 0 | 100 |
| j30_10 | 0 | 100 | 0 | 100 | 0 | 100 | 0 | 100 |
| j60_5 | 0.42 | 100 | 0 | 100 | 0 | 100 | 0 | 100 |
| j60_10 | 0.59 | 100 | 0.1 | 100 | 0 | 100 | 0 | 100 |
| j120_5 | 1.1 | 100 | 0.75 | 100 | 0.64 | 100 | 0.1 | 100 |
| j120_10 | 1.29 | 100 | 0.91 | 100 | 0.8 | 100 | 0.36 | 100 |

According to the results in Table 4, the performance of the algorithm presented in [9–11] is better than those of the algorithms presented in [5, 7, 8]. Thus, the algorithm presented in [9–11] may be considered as the best algorithm presented so far in the literature for solving the addressed problem.

Below, the performance of the algorithm presented in [9–11] is compared with that of the adaptive hybrid evolutionary algorithm proposed here. For simplicity, the algorithm presented in [9–11] will be referred as algorithm HEA.

Comparing the results obtained by the algorithm HEA (as detailed in Table 4) with those obtained by the adaptive hybrid evolutionary algorithm (as detailed in Table 3), the following points may be mentioned. Both algorithms have obtained an optimal effectiveness level for j30_5, j30_10, j60_5 and j60_10 (i.e., the four less complex instance sets). However, the effectiveness level obtained by the adaptive hybrid evolutionary algorithm for j120_5 and j120_10 (i.e., the two more complex instance sets) is significantly higher than that obtained by the algorithm HEA. Therefore, the adaptive hybrid evolutionary algorithm outperforms the algorithm HEA on the more complex instance sets. This is mainly because of the following reasons.

The adaptive hybrid evolutionary algorithm uses adaptive processes (i.e., adaptive parent selection, crossover, mutation, and survival selection processes) to develop the evolutionary search. Such processes adapt their behavior according to the population diversity, to promote either the exploration or exploitation of the search space, and therefore, improve the performance of the evolutionary search. In contrast with the adaptive hybrid evolutionary algorithm, the algorithm HEA utilizes non-adaptive processes (i.e., non-adaptive parent selection and survival selection processes) and semi-adaptive processes (i.e., semi-adaptive crossover and mutation processes) to develop the evolutionary search. The non-adaptive processes disregard the population diversity, and are not able to adapt their behavior during the evolutionary search to improve the performance of the evolutionary search. The semi-adaptive processes consider the population diversity; however, the adaptability of the behavior of these processes during the evolutionary search is very limited. Based on the mentioned above, the adaptive hybrid evolutionary algorithm has significant advantages to develop the evolutionary search.

## 6 Conclusions and Future Work

In this paper, an adaptive hybrid evolutionary algorithm was proposed to solve the project scheduling problem introduced in [5]. This problem considers a valuable optimization objective for project managers. Such objective involves maximizing the effectiveness of the sets of human resources assigned to the project activities.

The proposed adaptive hybrid evolutionary algorithm uses adaptive processes to develop the different stages of the evolutionary cycle (i.e., adaptive parent selection, survival selection, crossover, mutation and simulated annealing processes). These processes adapt their behavior according to the diversity of the evolutionary algorithm population, to promote either the exploration or exploitation of the search space. The utilization of these adaptive processes is meant to improve the evolutionary search.

The performance of the adaptive hybrid evolutionary algorithm was evaluated on six instance sets with very different complexity levels. After that, the performance of this algorithm on these six instance sets was compared with those of the algorithms previously reported in the literature for solving the addressed problem. Based on the obtained results, it may be stated that the proposed adaptive hybrid evolutionary algorithm significantly outperforms the algorithms previously reported.

In future works, other adaptive processes will be evaluated into the framework of the hybrid evolutionary algorithm. In particular, other adaptive parent selection, adaptive survival selection, adaptive crossover and adaptive mutation processes will be evaluated. Besides, other population diversity measures will be evaluated to adapt the behavior of the adaptive processes during the evolutionary search.

## References

1. Heerkens, G.R.: Project Management. McGraw-Hill, New York City (2002)
2. Wysocki, R.K.: Effective Project Management, 3rd edn. Wiley, Hoboken (2003)
3. De Bruecker, P., Van den Bergh, J., Beliën, J., Demeulemeester, E.: Workforce planning incorporating skills: state of the art. Eur. J. Oper. Res. **243**(1), 1–16 (2015)
4. Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L.: Personnel scheduling: a literature review. Eur. J. Oper. Res. **226**(3), 367–385 (2013)
5. Yannibelli, V., Amandi, A.: A knowledge-based evolutionary assistant to software development project scheduling. Expert Syst. Appl. **38**(7), 8403–8413 (2011)
6. Blazewicz, J., Lenstra, J., Rinnooy, K.A.: Scheduling subject to resource constraints: classification and complexity. Discrete Appl. Math. **5**, 11–24 (1983)

7. Yannibelli, V., Amandi, A.: A memetic approach to project scheduling that maximizes the effectiveness of the human resources assigned to project activities. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012. LNCS (LNAI), vol. 7208, pp. 159–173. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28942-2_15
8. Yannibelli, V., Amandi, A.: A diversity-adaptive hybrid evolutionary algorithm to solve a project scheduling problem. In: Corchado, E., Lozano, José A., Quintián, H., Yin, H. (eds.) IDEAL 2014. LNCS, vol. 8669, pp. 412–423. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10840-7_50
9. Yannibelli, V., Amandi, A.: Hybrid evolutionary algorithm with adaptive crossover, mutation and simulated annealing processes to project scheduling. In: Jackowski, K., Burduk, R., Walkowiak, K., Woźniak, M., Yin, H. (eds.) IDEAL 2015. LNCS, vol. 9375, pp. 340–351. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24834-9_40
10. Yannibelli, V., Amandi, A.: Scheduling projects by a hybrid evolutionary algorithm with self-adaptive processes. In: Sidorov, G., Galicia-Haro, Sofía N. (eds.) MICAI 2015. LNCS (LNAI), vol. 9413, pp. 401–412. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27060-9_33
11. Yannibelli, V., Amandi, A.: Project Scheduling: a memetic algorithm with diversity-adaptive components that optimizes the effectiveness of human resources. Polibits **52**, 93–103 (2015)
12. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. NCS. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-44874-8
13. Rodriguez, F.J., García-Martínez, C., Lozano, M.: Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: taxonomy, comparison, and synergy test. IEEE Trans. Evol. Comput. **16**(6), 787–800 (2012)
14. Talbi, E. (ed.): Hybrid Metaheuristics. SCI, vol. 434. Springer, Heidelberg (2013)
15. Cheng, J., Zhang, G., Caraffini, F., Neri, F.: Multicriteria adaptive differential evolution for global numerical optimization. Integr. Comput.-Aided Eng. **22**(2), 103–117 (2015)
16. Wang, R., Zhang, Y., Zhang, L.: An adaptive neural network approach for operator functional state prediction using psychophysiological data. Integr. Comput.-Aided Eng. **23**, 81–97 (2016)
17. Zhu, Z., Xiao, J., Li, J.Q., Wang, F., Zhang, Q.: Global path planning of wheeled robots using multi-objective memetic algorithms. Integr. Comput.-Aided Eng. **22**(4), 387–404 (2015)
18. Kolisch, R., Hartmann, S.: Experimental investigation of heuristics for resource-constrained project scheduling: an update. Eur. J. Oper. Res. **174**, 23–37 (2006)
19. Li, H., Womer, N.K.: Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming. Eur. J. Oper. Res. **246**(1), 20–33 (2015)
20. Heimerl, C., Kolisch, R.: Scheduling and staffing multiple projects with a multi-skilled workforce. OR Spectrum **32**(4), 343–368 (2010)
21. Li, H., Womer, K.: Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. J. Sched. **12**, 281–298 (2009)
22. Drezet, L.E., Billaut, J.C.: A project scheduling problem with labour constraints and time-dependent activities requirements. Int. J. Prod. Econ. **112**, 217–225 (2008)
23. Bellenguez, O., Néron, E.: A branch-and-bound method for solving multi-skill project scheduling problem. RAIRO – Oper. Res. **41**(2), 155–170 (2007)
24. Braekers, K., Hartl, R.F., Parragh, S.N., Tricoire, F.: A bi-objective home care scheduling problem: analyzing the trade-off between costs and client inconvenience. Eur. J. Oper. Res. **248**(2), 428–443 (2016)
25. Aickelin, U., Burke, E., Li, J.: An evolutionary squeaky wheel optimization approach to personnel scheduling. IEEE Trans. Evol. Comput. **13**(2), 433–443 (2009)

26. Valls, V., Pérez, A., Quintanilla, S.: Skilled workforce scheduling in service centers. Eur. J. Oper. Res. **193**(3), 791–804 (2009)
27. Bellenguez, O., Néron, E.: Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. In: Burke, E., Trick, M. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 229–243. Springer, Heidelberg (2005). https://doi.org/10.1007/11593577_14
28. Silva, T., De Souza, M., Saldanha, R., Burke, E.: Surgical scheduling with simultaneous employment of specialised human resources. Eur. J. Oper. Res. **245**(3), 719–730 (2015)
29. Gutjahr, W.J., Katzensteiner, S., Reiter, P., Stummer, Ch., Denk, M.: Competence-driven project portfolio selection, scheduling and staff assignment. CEJOR **16**(3), 281–306 (2008)
30. Hanne, T., Nickel, S.: A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects. Eur. J. Oper. Res. **167**, 663–678 (2005)
31. Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybern. **24**(4), 656–667 (1994)
32. Mahfoud, S.W.: Crowding and preselection revised. Parallel Problem Solving from Nature **2**, 27–36 (1992)