Intercriteria Analysis of ACO Performance for Workforce Planning Problem

Olympia Roeva, Stefka Fidanova, Gabriel Luque and Marcin Paprzycki

Abstract The workforce planning helps organizations to optimize the production process with the aim to minimize the assigning costs. The problem is to select a set of employees from a set of available workers and to assign this staff to the jobs to be performed. A workforce planning problem is very complex and requires special algorithms to be solved. The complexity of this problem does not allow the application of exact methods for instances of realistic size. Therefore, we will apply Ant Colony Optimization (ACO) algorithm, which is a stochastic method for solving combinatorial optimization problems. The ACO algorithm is tested on a set of 20 workforce planning problem instances. The obtained solutions are compared with other methods, as scatter search and genetic algorithm. The results show that ACO algorithm performs better than other the two algorithms. Further, we focus on the influence of the number of ants and the number of iterations on ACO algorithm performance. The tests are done on 16 different problem instances – ten structured and six unstructured problems. The results from ACO optimization procedures are discussed. In order to evaluate the influence of considered ACO parameters additional investigation is done. InterCriteria Analysis is performed on the ACO results for the regarded 16 problems. The results show that for the considered here workforce

---

O. Roeva
Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, Sofia, Bulgaria
e-mail: olympia@biomed.bas.bg

S. Fidanova (✉)
Institute of Information and Communication Technology, Bulgarian Academy of Sciences, Sofia, Bulgaria
e-mail: stefka@parallel.bas.bg

G. Luque
Department of Languages and Computer Science, University of Mlaga, 29071 Mlaga, Spain
e-mail: gabriel@lcc.uma.es

M. Paprzycki
System Research Institute, Polish Academy of Sciences, Warsaw and Management Academy, Warsaw, Poland
e-mail: marcin.paprzycki@ibspan.waw.pl

S. Fidanova (ed.), *Recent Advances in Computational Optimization*, Studies in Computational Intelligence 795,
https://doi.org/10.1007/978-3-319-99648-6_4

planning problem the best performance is achieved by the ACO algorithm with five ants in population.

Keywords Workforce planning · Ant colony optimization · Metaheuristics InterCriteria analysis

1 Introduction

The workforce planing is an essential question of the human resource management. This is an important industrial decision making problem. It is a hard optimization problem, which includes multiple levels of complexity. This problem contains two decision sets: selection and assignment. The first set selects employees from the larger set of available workers. The second set assigns the employees to the jobs to be performed. The aim is minimal assignment cost while the work requirements are fulfilled.

For this very complex problem with strong constraints it is impossible to apply exact methods for instances with realistic size. A deterministic workforce planing problem is studied in [26, 32]. In the work [26] workforce planning models that contain non-linear models of human learning are reformulated as mixed integer programs. The authors show that the mixed integer program is much easier to solve than the non-linear program. In [32] a model of workforce planning, that includes workers differences, as well as the possibility of workers training and improving, is considered. A variant of the problem with random demands is proposed in [19, 33]. In [19] a two-stage stochastic program for scheduling and allocating cross-trained workers is proposed considering a multi-department service environment with random demands. In some problems uncertainty has been employed [27, 30, 31, 39, 41]. In such cases the corresponding objective function and given constraints are converted into crisp equivalents and then the model is solved by traditional methods [31] or the considered uncertain model is transformed into an equivalent deterministic form as it is shown in [39]. Most of the approaches simplify the problem by omitting some of the constraints. Some conventional methods can be applied to workforce planning problem as mixed linear programming [21] and decomposition method [33]. However, for the more complex non-linear workforce planning problems, the convex methods are not applicable. In this case some heuristic methods including genetic algorithm [1, 29], memetic algorithm [37], scatter search [1], are applied.

In this work we propose an Ant Colony Optimization (ACO) algorithm for workforce planning problem. So far the ACO algorithm has been proven to be very effective in solving various complex optimization problems [23, 25]. We consider the variant of the workforce planning problem proposed in [1]. Our ACO algorithm performance is compared with the performance of the genetic algorithm and scatter search shown in [1]. Moreover, we focus on optimization of the algorithm parameters in order to find the minimal number of ants which is needed to find the best solution. It is known that when the number of ant doubles, the computational time and the used memory doubles, too. When the number of iterations doubles, only the computational time doubles. We look for a minimal product between number of ants and number of iterations that is sufficient to find the best solution.

In addition, we apply the recently developed approach – InterCriteria Analysis (ICrA) [11]. ICrA is an approach aiming to go beyond the nature of the criteria involved in a process of evaluation of multiple objects against multiple criteria, and, thus to discover some dependencies between the ICrA criteria themselves [11]. Initially, ICrA has been applied for the purposes of temporal, threshold and trends analyses of an economic case-study of European Union member states' competitiveness [15–17]. Further, ICrA has been used to discover the dependencies of different problems as [38, 40] and analysis of the performance of some metaheuristics as GAs and ACO [2, 22, 35, 36]. Published results show the applicability of the ICrA and the correctness of the approach.

ICrA could be appropriate approach for establishing the correlations between different ACO algorithms, based on their performance. ICrA may lead to additional exploration of the considered here problem. Due to that reason, in this paper, ICrA is applied to facilitate the analysis of the number of ants and number of iterations influence on the ACO performance in the considered here workforce planing problem.

The rest of the paper is organized as follows. In Sect. 2 the mathematical description of the workforce planing problem is presented. In Sect. 3 the ACO algorithm for workforce planing problem is proposed. The theoretical background of the ICrA is given in Sect. 4. The numerical results from ACO application for workforce planing problem are summarized and discussed in Sect. 5. In Sect. 6 ACO performance based on differently tuned algorithm parameters is investigated. The presented results are discussed in terms of which ACO algorithm is best to solve the workforce planing problem. The results from ICrA application are discussed in Sect. 7. In Sect. 8 some conclusions and directions for future works are done.

2 The Workforce Planning Problem

In this paper we use the description of workforce planing problem given by Glover et al. [24]. There is a set of jobs $J = \{1, \dots, m\}$, which must be completed during a fixed period (week, for example). Every job j requires d_j hours to be completed. The set of available workers is $I = \{1, \dots, n\}$. For efficiency reason each worker must perform all assigned to him jobs for minimum h_{min} hours. The worker i is available for s_i hours. The maximal number of assigned jobs to the same worker is j_{max}. Workers have different skills and the set A_i shows the jobs that the worker i is qualified to perform. The maximal number of workers which can be assigned during the planed period is t, i.e., at most t workers may be selected from the set I of workers and the selected workers must be capable to complete all the jobs. The aim is to find feasible solution that optimizes the objective function.

Each worker i and job j are related with cost c_{ij} of assigning the worker to the job. The mathematical model of the workforce planing problem is as follows:

$$x_{ij} = \begin{cases} 1 \text{ if the worker } i \text{ is assigned to job } j \\ 0 \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if worker } i \text{ is selected} \\ 0 \text{ otherwise} \end{cases}$$

z_{ij} = number of hours that worker i

is assigned to perform job j

Q_j = set of workers qualified to perform job j

$$\text{Minimize} \sum_{i \in I} \sum_{j \in A_i} c_{ij} \, . \, x_{ij} \quad (1)$$

Subject to

$$\sum_{j \in A_i} z_{ij} \le s_i \, . \, y_i \quad i \in I \quad (2)$$

$$\sum_{i \in Q_j} z_{ij} \ge d_j \quad j \in J \quad (3)$$

$$\sum_{j \in A_i} x_{ij} \le j_{max} \, . \, y_j \quad i \in I \quad (4)$$

$$h_{min} \, . \, x_{ij} \le z_{ij} \le s_i \, . \, x_{ij} \quad i \in I, j \in A_i \quad (5)$$

$$\sum_{i \in I} y_i \le t \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad i \in I, j \in A_i$$
$$y_i \in \{0, 1\} \quad i \in I$$
$$z_{ij} \ge 0 \quad i \in I, j \in A_i$$

The objective function of this problem minimizes the total assignment cost. The number of hours for each selected worker is bounded (inequality 2). The work must be done in full (inequality 3). The number of the jobs, that each worker can perform is limited (inequality 4). There is minimal number of hours that each job must be performed by all assigned workers to can work efficiently (inequality 5). The number of assigned workers is limited (inequality 6).

Different objective functions can be optimized with the same model. In this paper our aim is to minimize the total assignment cost. If $\tilde{c}_{ij}$ is the cost the worker i to performs the job j for one hour, than the objective function can minimize the cost of the all jobs to be finished (on hourly basis).

$$f(x) = \text{Min} \sum_{i \in I} \sum_{j \in A_i} \tilde{c}_{ij} \,.\, x_{ij} \tag{7}$$

Some workers can have preference to perform part of the jobs he is qualified and the objective function can be to maximize the satisfaction of the workers preferences or to maximize the minimum preference value for the set of selected workers.

As we mentioned above in this paper the assignment cost is minimized (equation 1). This problem is similar to the Capacitate Facility Location Problem (CFLP). The workforce planning problem is difficult to be solved because of very restrictive constraints especially the relation between the parameters h_{min} and d_j. When the problem is structured (d_j is a multiple of h_{min}), it is far easier to find feasible solution, than for unstructured problems (d_j and h_{min} are not related).

3 Ant Colony Optimization

The ACO is a metaheuristic methodology which follows the real ant colonies behaviour when they look for a food and return back to the nest. Real ants use chemical substance, called pheromone, to mark their path ant to be able to return back. An isolated ant moves randomly, but when an ant detects a previously laid pheromone it can decide to follow the trail and to reinforce it with additional quantity of pheromone. The repetition of the above mechanism represents the auto-catalytic behavior of a real ant colony, where the more ants follow a given trail, the more attractive that trail becomes. Thus the ants can collectively find a shorter path between the nest and the source of the food. The main idea of the ACO algorithms comes from this natural behaviour.

3.1 Main ACO Algorithm

Metaheuristic methods are applied on difficult in computational point of view problems, when it is impractical to use traditional numerical methods. A lot of problems coming from real life, especially from the industry. These problems need exponential number of calculations and the only option, when the problem is large, is to apply some metaheuristic methods in order to obtain a good solution for a reasonable time [20].

ACO algorithm is proposed by Dorigo et al. [18]. Later some modifications have been proposed mainly in pheromone updating rules [20]. The artificial ants in ACO algorithms simulate the ants behaviour. The problem is represented by graph. The solutions are represented by paths in a graph and we look for shorter path corresponding to given constraints. The requirements of ACO algorithm are as follows:

- Suitable representation of the problem by a graph;
- Suitable pheromone placement on the nodes or on the arcs of the graph;
- Appropriate problem-dependent heuristic function, which manage the ants to improve solutions;
- Pheromone updating rules;
- Transition probability rule, which specifies how to include new nodes in the partial solution.

The structure of the ACO algorithm is shown on Fig. 1.

The transition probability $p_{i,j}$, to choose the node j, when the current node is i, is a product of the heuristic information $\eta_{i,j}$ and the pheromone trail level $\tau_{i,j}$ related with this move, where $i, j = 1, \ldots, n$.

$$p_{i,j} = \frac{\tau_{i,j}^a \eta_{i,j}^b}{\sum\limits_{k \in Unused} \tau_{i,k}^a \eta_{i,k}^b}, \quad (8)$$

where $Unused$ is the set of unused nodes of the graph.

A node becomes more profitable if the value of the heuristic information and/or the related pheromone is higher. At the beginning, the initial pheromone level is the same for all elements of the graph and is set to a small positive constant value τ_0, $0 < \tau_0 < 1$. At the end of every iteration the ants update the pheromone values. Different ACO algorithms adopt different criteria to update the pheromone level [20].

The main pheromone trail update rule is:

$$\tau_{i,j} \leftarrow \rho \tau_{i,j} + \Delta \tau_{i,j}, \quad (9)$$

where ρ decreases the value of the pheromone, like the evaporation in a nature. $\Delta \tau_{i,j}$ is a newly added pheromone, which is proportional to the quality of the solution. The quality of the solution is measured by the value of the objective function of the solution constructed by the ant.

An ant start to construct its solution from a random node of the graph of the problem. The random start is a diversification of the search. Because of the random start

Fig. 1 Pseudo-code of ACO algorithm

```
Ant Colony Optimization
Initialize number of ants;
Initialize the ACO parameters;
while not end condition do
        for k = 0 to number of ants
            ant k chooses start node;
            while solution is not constructed do
                ant k selects higher probability node;
            end while
        end for
        Update pheromone trails;
end while
```

a relatively small number of ants can be used, compared to other population based metaheuristics. The heuristic information represents the prior knowledge of the problem, which we use to better manage the ants. The pheromone is a global experience of the ants to find optimal solution. The pheromone is a tool for concentration of the search around the best so far solutions.

3.2 ACO Algorithm for Workforce Planning

One of the essential point of the ant algorithm is the proper representation of the problem by graph. In our case the graph of the problem is 3 dimensional and the node (i, j, z) denotes to worker i to be assigned to the job j for time z. At the beginning of every iteration each ant starts to construct its solution, from random node of the graph of the problem. For each ant are generated three random numbers. The first random number is in the interval $[0, \dots, n]$ and corresponds to the worker we assign. The second random number is in the interval $[0, \dots, m]$ and corresponds to the job which this worker will perform. The third random number is in the interval $[h_{min}, \dots, \min\{d_j, s_i\}]$ and corresponds to the number of hours worker i is assigned to perform the job j. Subsequently, the ant applies the transition probability rule to include next nodes in the partial solution, until the solution is completed.

We propose the following heuristic information:

$$\eta_{ijl} = \begin{cases} l/c_{ij} & l = z_{ij} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

This heuristic information stimulates to assign the most cheapest worker as longer as possible. The ant chooses the node with the highest probability. When an ant has several possibilities for next node (several candidates have the same probability to be chosen), the next node is chosen randomly among them.

When a new node is included we take in to account how many workers are assigned currently, how many time slots every worker is currently assigned and how many time slots are currently assigned per job. When some move of the ant does not meet the problem constraints, the probability of this move is set to 0. If it is impossible to include new nodes from the graph of the problem (for all nodes the value of the transition probability is 0), the construction of the solution stops. When the constructed solution is feasible the value of the objective function is the sum of the assignment cost of the assigned workers. If the constructed solution is not feasible, the value of the objective function is set to -1.

Only the ants, which constructed feasible solution are allowed to add new pheromone to the elements of their solutions. The newly added pheromone is equal to the reciprocal value of the objective function:

$$\Delta\tau_{i,j} = \frac{\rho - 1}{f(x)} \quad (11)$$

Thus, the nodes of the problem graph, which belong to better solutions (less value of the objective function) receive more pheromone than the other nodes and become more desirable in the next iteration.

At the end of every iteration we compare the best solution with the best so far solution. If the best solution from the current iteration is better than the best so far solution (global best solution), we update the global best solution with the current iteration best solution.

The end condition used in our ACO algorithm is the number of iterations.

4 InterCriteria Analysis

InterCriteria analysis, based on the apparatuses of index matrices [3, 5, 7–9] and intuitionistic fuzzy sets (IFSs) [4, 6, 10], is given in details in [11]. Here, for completeness, the proposed idea is briefly presented.

An intuitionistic fuzzy pair (IFP) [12] is an ordered pair of real non-negative numbers $\langle a, b\rangle$, where $a, b \in [0, 1]$ and $a + b \leq 1$, that is used as an evaluation of some object or process. According to [12], the components (a and b) of IFP might be interpreted as degrees of "membership" and "non-membership" to a given set, degrees of "agreement" and "disagreement", degrees of "validity" and "non-validity", degrees of "correctness" and "non-correctness", etc.

The apparatus of index matrices is presented initially in [5] and discussed in more details in [7, 8]. For the purposes of ICrA application, the initial index set consists of the criteria (for rows) and objects (for columns) with the index matrix elements assumed to be real numbers. Further, an index matrix with index sets consisting of the criteria (for rows and for columns) with IFP elements determining the degrees of correspondence between the respective criteria is constructed, as it is going to be briefly presented below.

Let the initial index matrix is presented in the form of Eq. (12), where, for every p, q, $(1 \leq p \leq m, 1 \leq q \leq n)$, C_p is a criterion, taking part in the evaluation; O_q – an object to be evaluated; $C_p(O_q)$ – a real number (the value assigned by the p-th criteria to the q-th object).

$$A = \begin{array}{c|ccccccc} & O_1 & \dots & O_k & \dots & O_l & \dots & O_n \\ \hline C_1 & C_1(O_1) & \dots & C_1(O_k) & \dots & C_1(O_l) & \dots & C_1(O_n) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_i & C_i(O_1) & \dots & C_i(O_k) & \dots & C_i(O_l) & \dots & C_i(O_n) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_j & C_j(O_1) & \dots & C_j(O_k) & \dots & C_j(O_l) & \dots & C_j(O_n) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_m & C_m(O_1) & \dots & C_m(O_k) & \dots & C_m(O_l) & \dots & C_m(O_n) \end{array}, \quad (12)$$

Let O denotes the set of all objects being evaluated, and $C(O)$ is the set of values assigned by a given criteria C (i.e., $C = C_p$ for some fixed p) to the objects, i.e.,

$$O \stackrel{\text{def}}{=} \{O_1, O_2, O_3, \ldots, O_n\},$$
$$C(O) \stackrel{\text{def}}{=} \{C(O_1), C(O_2), C(O_3), \ldots, C(O_n)\}.$$

Let $x_i = C(O_i)$. Then the following set can be defined:

$$C^*(O) \stackrel{\text{def}}{=} \{\langle x_i, x_j \rangle | i \neq j \;\&\; \langle x_i, x_j \rangle \in C(O) \times C(O)\}.$$

Further, if $x = C(O_i)$ and $y = C(O_j)$, $x \prec y$ iff $i < j$ will be written.

In order to find the agreement of different criteria, the vectors of all internal comparisons for each criterion are constructed, which elements fulfil one of the three relations R, $\overline{R}$ and $\tilde{R}$. The nature of the relations is chosen such that for a fixed criterion C and any ordered pair $\langle x, y \rangle \in C^*(O)$:

$$\langle x, y \rangle \in R \Leftrightarrow \langle y, x \rangle \in \overline{R}, \tag{13}$$
$$\langle x, y \rangle \in \tilde{R} \Leftrightarrow \langle x, y \rangle \notin (R \cup \overline{R}), \tag{14}$$
$$R \cup \overline{R} \cup \tilde{R} = C^*(O). \tag{15}$$

For example, if "R" is the relation "<", then $\overline{R}$ is the relation ">", and vice versa.

Hence, for the effective calculation of the vector of internal comparisons (denoted further by $V(C)$) only the considering of a subset of $C(O) \times C(O)$ is needed, namely:

$$C^{\prec}(O) \stackrel{\text{def}}{=} \{\langle x, y \rangle | \; x \prec y \;\&\; \langle x, y \rangle \in C(O) \times C(O),$$

due to Eqs. (13)–(15). For brevity, $c^{i,j} = \langle C(O_i), C(O_j) \rangle$.

Then for a fixed criterion C the vector of lexicographically ordered pair elements is constructed:

$$V(C) = \{c^{1,2}, c^{1,3}, \ldots, c^{1,n}, c^{2,3}, c^{2,4}, \ldots, c^{2,n}, c^{3,4}, \ldots, c^{3,n}, \ldots, c^{n-1,n}\}. \tag{16}$$

In order to be more suitable for calculations, $V(C)$ is replaced by $\hat{V}(C)$, where its k-th component ($1 \leq k \leq \frac{n(n-1)}{2}$) is given by:

$$\hat{V}_k(C) = \begin{cases} 1, & \text{iff } V_k(C) \in R, \\ -1, & \text{iff } V_k(C) \in \overline{R}, \\ 0, & \text{otherwise.} \end{cases}$$

When comparing two criteria the degree of "agreement" is determined as the number of matching components of the respective vectors (divided by the length of the vector for normalization purposes). This can be done in several ways, e.g. by counting the matches or by taking the complement of the Hamming distance. The degree of "disagreement" is the number of components of opposing signs in the two vectors (again normalized by the length).

If the respective degrees of "agreement" and "disagreement" are denoted by $\mu_{C,C'}$ and $\nu_{C,C'}$, it is obvious (from the way of computation) that $\mu_{C,C'} = \mu_{C',C}$ and $\nu_{C,C'} = \nu_{C',C}$. Also it is true that $\langle \mu_{C,C'}, \nu_{C,C'} \rangle$ is an IFP.

In the most of the obtained pairs $\langle \mu_{C,C'}, \nu_{C,C'} \rangle$, the sum $\mu_{C,C'} + \nu_{C,C'}$ is equal to 1. However, there may be some pairs, for which this sum is less than 1. The difference

$$\pi_{C,C'} = 1 - \mu_{C,C'} - \nu_{C,C'} \quad (17)$$

is considered as a degree of "uncertainty".

The following index matrix is constructed as a result of applying the ICrA to A [Eq. (12)]:

$$\begin{array}{c|ccc} & C_2 & \dots & C_m \\ \hline C_1 & \langle \mu_{C_1,C_2}, \nu_{C_1,C_2} \rangle & \dots & \langle \mu_{C_1,C_m}, \nu_{C_1,C_m} \rangle \\ \vdots & & \vdots & \\ C_{m-1} & & \dots & \langle \mu_{C_{m-1},C_m}, \nu_{C_{m-1},C_m} \rangle \end{array},$$

that determines the degrees of correspondence between criteria $C_1, \ldots, C_m$.

In this paper we use μ-biased algorithm **Algorithm 1** for calculation of intercriteria relations [34]. An example pseudocode of the **Algorithm 1** is presented below.

Algorithm 1 Calculating $\mu_{C,C'}$ and $\nu_{C,C'}$ between two criteria

Require: Vectors $\hat{V}(C)$ and $\hat{V}(C')$

function DEGREES OF AGREEMENT AND DISAGREEMENT($\hat{V}(C)$, $\hat{V}(C')$)
    $V \leftarrow \hat{V}(C) - \hat{V}(C')$
    $\mu \leftarrow 0$
    $\nu \leftarrow 0$
    for $i \leftarrow 1$ to $\frac{n(n-1)}{2}$ **do**
        if $V_i = 0$ **then**
            $\mu \leftarrow \mu + 1$
        else if $\mathrm{abs}(V_i) = 2$ **then** ▷ $\mathrm{abs}(V_i)$: the absolute value of V_i
            $\nu \leftarrow \nu + 1$
        end if
    end for
    $\mu \leftarrow \frac{2}{n(n-1)} \mu$
    $\nu \leftarrow \frac{2}{n(n-1)} \nu$
    return μ, ν
end function

5 Application of ACO for Workforce Planning Problem

In this section we report test results and compare them with results achieved by other methods. We analyse the algorithm performance and the quality of the achieved solutions. The software, which realizes the algorithm is written in C and is run on Pentium desktop computer at 2.8 GHz with 4 GB of memory.

We use the artificially generated problem instances considered in [1]. The test instances characteristics are shown in Table 1.

The set of test problems consists of ten structured and ten unstructured problems. The problem is structured when d_j is proportional to h_{min}. The structured problems are enumerated from $S01$ to $S10$ and unstructured problems are enumerated from $U01$ to $U10$.

The number of iterations (stopping criteria) is fixed to be 100. The parameter settings of our ACO algorithm are shown in Table 2. The values are fixed experimentally.

The algorithm is stochastic and from a statistical point of view it needs to be run minimum 30 times to guarantee the robustness of the average results. We perform 30 independent runs of the algorithm. Afterwards statistical analysis of the results applying ANOVA test was done.The test shows that there is significant difference between the results achieved by different methods, or the results are not statistically the same.

Table 1 Test instances characteristics

Parameters	Value
n	20
m	20
t	10
s_i	[50, 70]
j_{max}	[3, 5]
h_{min}	[10, 15]

Table 2 ACO parameter settings

Parameters	Value
Number of iterations	100
ρ	0.5
τ_0	0.5
Number of ants	20
a	1
b	1

Let us compare the numerical results achieved by our ACO algorithm and those achieved by genetic algorithm (GA) and scatter search (SS) presented in [1]. Table 3 shows the achieved results for structured instances, while Table 4 shows the obtained results for unstructured instances.

We observe that ACO algorithm outperforms the other two algorithms. ACO is a constructive method and when the graph of the problem and heuristic information are appropriate and they represent the problem in a good way, it can help a lot of for better algorithm performance and achieving good solutions. Our graph of the problem has a star shape. Each worker and job are linked with several nodes, corresponding to the time, for which the worker is assigned to perform this job. The proposed heuristic information stimulates the cheapest workers to be assigned for longer time. It is a greedy strategy. After the first iteration the pheromone level reflects the experience

Table 3 Average results for structured problems

Test problem	Objective function value		
	SS	GA	ACO
*S*01	936	963	807
*S*02	952	994	818
*S*03	1095	1152	882
*S*04	1043	1201	849
*S*05	1099	1098	940
*S*06	1076	1193	869
*S*07	987	1086	812
*S*08	1293	1287	872
*S*09	1086	1107	793
*S*10	945	1086	825

Table 4 Average results for unstructured problems

Test problem	Objective function value		
	SS	GA	ACO
*U*01	1586	1631	814
*U*02	1276	1264	845
*U*03	1502	1539	906
*U*04	1653	1603	869
*U*05	1287	1356	851
*U*06	1193	1205	873
*U*07	1328	1301	828
*U*08	1141	1106	801
*U*09	1055	1173	768
*U*10	1178	1214	818

of the ants during the searching process thus affects the strategy. The elements of good solutions accumulate more pheromone, during the algorithm performance, than others and become more desirable in the next iterations.

Now we will compare the execution time of the proposed ACO algorithm with the execution time of the other two algorithms – GA and SS [1]. The algorithms are run on similar computers. In Table 5 the parameters of the GA and SS algorithms are presented.

The average execution times over 30 runs of each of the algorithms are reported in Tables 6 and 7.

It is seen that the ACO algorithm finds the solution faster than GA and SS. Considering the execution time the GA and SS algorithms have similar performance. By the numerical results presented in Tables 3, 4, 6 and 7 we can conclude that ACO algorithm gives very encouraging results. It achieves better solutions in shorter time than

Table 5 Algorithms parameter settings, as given in [1]

Parameters	Genetic algorithm
Population size	400
Crossover rate	0.8
Mutation rate	0.2
Parameters	Scatter search
Initial population	15
Reference set update and creation	8
Subset generated	All 2-elements subsets
ρ_i	0.1

Table 6 Average time for structured problems

Test problem	Execution time, s		
	SS	GA	ACO
$S01$	72	61	26
$S02$	49	32	21
$S03$	114	111	22
$S04$	86	87	25
$S05$	43	40	21
$S06$	121	110	23
$S07$	52	49	23
$S08$	46	42	24
$S09$	70	67	20
$S10$	105	102	22

Table 7 Average time for unstructured problems

Test problem	Execution time, s		
	SS	GA	ACO
*U*01	102	95	22
*U*02	94	87	20
*U*03	58	51	20
*U*04	83	79	20
*U*05	62	57	23
*U*06	111	75	22
*U*07	80	79	21
*U*08	123	89	20
*U*09	75	72	26
*U*10	99	95	20

the other two algorithms, SS and GA. If we compare the used memory, the ACO algorithm uses less memory than GA (GA population size is 400 individuals) and similar memory to SS (initial population size is 15 and reference set is 8 individuals) [1].

6 Influence of ACO Parameters on Algorithm Performance

In this section we analyse the ACO performance according to the number of ants and the quality of the achieved solutions. We use the same artificially generated problem instances, considered in [1].

If the number of ants of ACO algorithm increases, the computational time and the used memory increase proportionally. If the number of iterations increases, only the computational time increases. If the computational time is fixed and we vary only the number of ants it means that we vary the number of iteration too, but in opposite direction, or if the time is fixed it is equivalent to fixing the product of number of ants and number of iterations.

We apply number of ants from the set {5, 10, 20, 40} and respectively, number of iteration – {400, 200, 100, 50}. Because of stochastic nature of the algorithm we run the ACO algorithm for all 16 test problems with each one of the four ACO algorithms ($ACO_{5\times400}$, $ACO_{10\times200}$, $ACO_{20\times100}$ and $ACO_{40\times50}$) 30 times. We look for the maximal number of iterations, within the fixed computational time, which is needed to find the best solution. We compare the product between the number of ants and number of required iterations for the best performed ACO algorithm. The parameter settings for our ACO algorithm are shown in Table 8.

Tables 9 and 10 show the resulting product between the number of ants and number of iterations that have been used to find the best solution. When the observed product is the same for different ACO algorithms (with different number of ants) the best

Table 8 ACO parameter settings

Parameters	Value
Number of iterations	400, 200, 100, 50
ρ	0.5
τ_0	0.5
Number of ants	5, 10, 20, 40
a	1
b	1

Table 9 Computational results for structured problems

Test problem	Product ants × Iterations			
	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
*S*01	195	200	260	**120**
*S*02	**195**	330	340	640
*S*03	**475**	490	580	1160
*S*04	**1540**	**1540**	**1540**	1560
*S*05	415	**320**	420	920
*S*06	**165**	250	420	520
*S*07	**570**	720	860	880
*S*08	**1125**	1130	1140	1120
*S*09	**855**	860	720	880
*S*10	**230**	230	520	840

Table 10 Computational results for unstructured problems

Test problem	Product ants × Iterations			
	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
*U*00	**775**	780	780	2060
*U*01	**330**	340	340	440
*U*02	**160**	340	340	400
*U*03	1000	1000	**560**	640
*U*04	**760**	760	820	820
*U*05	295	295	**280**	280

results is this produced by ACO with lesser number of ants, because in this case the used memory is less. The best results are shown in bold.

Let us discuss the results reported in Tables 9 and 10. Regarding the structured problems, for 8 of them the best execution time is when the number of ants is 5. Only for two of the test problems (*S*01 and *S*05) the execution time is better for 40 and 10 ants, respectively, but the result is close to this achieved with 5 ants. Regarding unstructured test problems, for four of them the best execution time is achieved when the number of used ants is 5. For two test problems ($U03$ and $U05$) the best results are achieved using ACO with 20 ants. For test $U05$, the result is quite close to the result with 5 ants. Only for test $U03$ the difference with the results using 5 ants is significant. Thus, we can conclude that for this problem the best algorithm performance, using less computational resources, is when the number of used ants is 5.

7 InterCriteria Analysis of the Results

In this section we use ICrA to obtain some additional knowledge about considered four ACO algorithms. Based on the results in Tables 9 and 10 we construct the following index matrix, Eq. (18):

	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
S01	195	200	260	120
S02	195	330	340	640
S03	475	490	580	1160
S04	1540	1540	1540	1560
S05	415	320	420	920
S06	165	250	420	520
S07	570	720	860	880
S08	1125	1130	1140	1120
S09	855	860	720	880
S10	230	230	520	840
U00	775	780	780	2060
U01	330	340	340	440
U02	160	340	340	400
U03	1000	1000	560	640
U04	760	760	820	820
U05	295	295	280	280

(18)

From Eq. (18) it can be seen that the ICrA objects (S01, S02, ..., S10, U00, ..., U05) are the different test problems and the ICrA criteria ($ACO_{5\times400}$, $ACO_{10\times200}$, $ACO_{20\times100}$ and $ACO_{40\times50}$) are the ACO algorithms with different number of ants.

After application of ICrA, using the software ICrAData [28], to index matrix Eq. (18) we obtained the two index matrices with the relations between considered four criteria. The resulting index matrices for $\mu_{C,C'}$, $\nu_{C,C'}$ and $\pi_{C,C'}$ values are shown in Eqs. (19)–(21).

$\mu_{C,C'}$	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
$ACO_{5\times400}$	1	0.88	0.78	0.74
$ACO_{10\times200}$	0.88	1	0.78	0.71
$ACO_{20\times100}$	0.78	0.78	1	0.81
$ACO_{40\times50}$	0.74	0.71	0.81	1

(19)

$\nu_{C,C'}$	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
$ACO_{5\times400}$	0	0.10	0.18	0.23
$ACO_{10\times200}$	0.10	0	0.20	0.27
$ACO_{20\times100}$	0.18	0.20	0	0.14
$ACO_{40\times50}$	0.23	0.27	0.14	0

(20)

$\pi_{C,C'}$	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
$ACO_{5\times400}$	0	0.02	0.04	0.03
$ACO_{10\times200}$	0.02	0	0.03	0.03
$ACO_{20\times100}$	0.04	0.03	0	0.05
$ACO_{40\times50}$	0.03	0.03	0.05	0

(21)

The obtained ICrA results are visualized on Fig. 2 within the specific triangular geometrical interpretation of IFSs.

The results show that the following criteria pairs, according to [13], are in:

- positive consonance:
  $ACO_{5\times400}$-$ACO_{10\times200}$ – with degree of "agreement" $\mu_{C,C'} = 0.88$;
- weak positive consonance:
  $ACO_{20\times100}$-$ACO_{40\times50}$,
  $ACO_{20\times100}$-$ACO_{5\times400}$ and

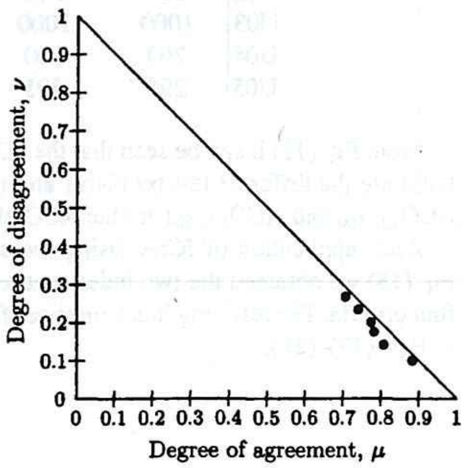

Fig. 2 Presentation of ICrA results in the intuitionistic fuzzy interpretation triangle

$ACO_{20\times100}$-$ACO_{10\times200}$ – with degree of "agreement" $\mu_{C,C'} = 0.81$, $\mu_{C,C'} = 0.78$ and $\mu_{C,C'} = 0.78$, respectively;
- weak dissonance:
$ACO_{40\times50}$-$ACO_{5\times400}$ and
$ACO_{40\times50}$-$ACO_{10\times200}$ – with degree of "agreement" $\mu_{C,C'} = 0.74$ and $\mu_{C,C'} = 0.71$, respectively.

ACO algorithms with close values of number of ants (5 and 10, 10 and 20, 20 and 40) show similar performance. The same result is obtained for ACO algorithms with 5 and 20 ants. The ACO algorithms with bigger difference in the number of ants (5 and 40, 10 and 40) are in weak dissonance, i.e. their performance is not similar (Table 11).

In [14] the author propose to rank the criteria pairs in both dimensions simultaneously (degrees of "agreement" $\mu_{C,C'}$ and "disagreement" $\nu_{C,C'}$ of the intuitionistic fuzzy pairs). This can be done by calculation for each point in the Fig. 2 its distance from the point $\langle 1, 0\rangle$. The formula for the distance $d_{C,C'}$ of the pair C, C' to the $\langle 1, 0\rangle$ point is:

$$d_{C,C'} = \sqrt{(1 - \mu_{C,C'})^2 + \nu_{C,C'}{}^2} \quad (22)$$

The results are presented in Table 12.

Table 11 Consonance and dissonance scale, according to [13]

Interval of $\mu_{C,C'}$	Meaning
(0.00–0.05]	Strong negative consonance
(0.05–0.15]	Negative consonance
(0.15–0.25]	Weak negative consonance
(0.25–0.33]	Weak dissonance
(0.33–0.43]	Dissonance
(0.43–0.57]	Strong dissonance
(0.57–0.67]	Dissonance
(0.67–0.75]	Weak dissonance
(0.75–0.85]	Weak positive consonance
(0.85–0.95]	Positive consonance
(0.95–1.00]	Strong positive consonance

Table 12 Index matrix of criteria distances from point $\langle 1, 0\rangle$

	$ACO_{5\times400}$	$ACO_{10\times200}$	$ACO_{20\times100}$	$ACO_{40\times50}$
$ACO_{5\times400}$	0	0.154	0.279	0.348
$ACO_{10\times200}$	0.154	0	0.301	0.395
$ACO_{20\times100}$	0.279	0.301	0	0.238
$ACO_{40\times50}$	0.348	0.395	0.238	0

In this case the criteria pairs (different ACO algorithms) are ordered according to their $d_{C,C'}$ sorted in decreasing order, as follows:

- $ACO_{5\times400}$-$ACO_{10\times200}$;
- $ACO_{20\times100}$-$ACO_{40\times50}$;
- $ACO_{5\times400}$-$ACO_{20\times100}$;
- $ACO_{10\times200}$-$ACO_{20\times100}$;
- $ACO_{5\times400}$-$ACO_{40\times50}$;
- $ACO_{40\times50}$-$ACO_{10\times200}$.

As it can be seen the similar performance of ACO algorithms with 5 and 10, and 20 and 40 ants is approved by these results too. The similarity between ACO with 5 and ACO with 20 ants is observed again. The next three criteria pairs, are ranked in the same manner, too. Thus the obtained ICrA results are confirmed by two different approaches – when using the scale, proposed in [13] and according simultaneously to the degrees of "agreement"$\mu_{C,C'}$ and "disagreement" $\nu_{C,C'}$ of the intuitionistic fuzzy pairs [14].

On the other hand, ICrA confirms the conclusion that for this problem the best algorithm performance, i.e., using less computational resources, is shown ACO algorithm with five number of ants.

8 Conclusion

In this article we propose ACO algorithm for solving workforce planning problem. We compare the performance of our algorithm with other two metahuristic methods, genetic algorithm and scatter search. The comparison is done by various criteria. We observed that ACO algorithm achieves better solutions than the two other algorithms. Regarding the execution time the ACO algorithm is faster. The ACO population consists of 20 individuals and the memory used by the algorithm is similar to the one used by the SS and less than the memory used by the GA. We achieved very encouraging results. As a future work we will combine our ACO algorithm with appropriate local search procedure for eventual further improvement of the algorithm performance and solutions quality.

In this paper we solve workforce planning problem applying ACO algorithm. We focus on influence of the algorithm parameters on its performance. We try to find minimal number of ants and minimal execution time and memory, which are needed to find best solution. The results show that for most of the test problems, the minimal computational resources are used when the number of ants is five. As a future work we will combine our ACO algorithm with appropriate local search procedure for eventual further improvement of the algorithm performance and solutions quality.

Acknowledgements Work presented here is partially supported by the National Scientific Fund of Bulgaria under grants DFNI-DN 02/10 "New Instruments for Knowledge Discovery from Data, and their Modelling" and DFNI-DN 12/5 "Efficient Stochastic Methods and Algorithms for Large Scale Problems".

References

1. Alba, E., Luque, G., Luna, F.: Parallel metaheuristics for workforce planning. J. Math. Model. Algorithms **6**(3), 509–528 (2007)
2. Angelova, M., Roeva, O., Pencheva, T.: InterCriteria analysis of crossover and mutation rates relations in simple genetic algorithm. In: Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, vol. 5, pp. 419–424 (2015)
3. Atanassov, K.: Index Matrices: Towards an Augmented Matrix Calculus. Springer, Switzerland (2014)
4. Atanassov, K.: Intuitionistic fuzzy sets. VII ITKR session, Sofia, 20–23 June 1983. Int. J. Bioautom. **20**(S1), S1–S6 (2016)
5. Atanassov, K.: Generalized index matrices. Comptes rendus de l'Academie bulgare des Sciences **40**(11), 15–18 (1987)
6. Atanassov, K.: On Intuitionistic Fuzzy Sets Theory. Springer, Berlin (2012)
7. Atanassov, K.: On index matrices, Part 1: standard cases. Adv. Stud. Contemp. Math. **20**(2), 291–302 (2010)
8. Atanassov, K.: On index matrices, Part 2: intuitionistic fuzzy case. Proc. Jangjeon Math. Soc. **13**(2), 121–126 (2010)
9. Atanassov, K.: On index matrices. Part 5: 3-dimensional index matrices. Adv. Stud. Contemp. Math. **24**(4), 423–432 (2014)
10. Atanassov, K.: Review and new results on intuitionistic fuzzy sets, mathematical foundations of artificial intelligence seminar, Sofia, 1988, Preprint IM-MFAIS-1-88. Int. J. Bioautom. **20**(S1), S7–S16 (2016)
11. Atanassov, K., Mavrov, D., Atanassova, V.: Intercriteria decision making: a new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets. Issues in on Intuitionistic Fuzzy Sets and Generalized Nets **11**, 1–8 (2014)
12. Atanassov, K., Szmidt, E., Kacprzyk, J.: On intuitionistic fuzzy pairs. Notes Intuitionistic Fuzzy Sets **19**(3), 1–13 (2013)
13. Atanassov, K., Atanassova, V., Gluhchev, G.: InterCriteria analysis: ideas and problems. Notes on Intuitionistic Fuzzy Sets **21**(1), 81–88 (2015)
14. Atanassova, V.: Interpretation in the intuitionistic fuzzy triangle of the results, obtained by the InterCriteria analysis. In: Proceedings of the 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), pp. 1369–1374 (2015)
15. Atanassova, V., Mavrov, D., Doukovska, L., Atanassov, K.: Discussion on the threshold values in the InterCriteria decision making approach. Notes on Intuitionistic Fuzzy Sets **20**(2), 94–99 (2014)
16. Atanassova, V., Doukovska, L., Atanassov, K., Mavrov, D.: Intercriteria decision making approach to EU member states competitiveness analysis. In: Proceedings of the International Symposium on Business Modeling and Software Design - BMSD'14, pp. 289–294 (2014)
17. Atanassova, V., Doukovska, L., Karastoyanov, D., Capkovic, F.: InterCriteria decision making approach to EU member states competitiveness analysis: trend analysis. In: Intelligent Systems'2014, Advances in Intelligent Systems and Computing, vol. 322, pp. 107–115 (2014)
18. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
19. Campbell, G.: A two-stage stochastic program for scheduling and allocating cross-trained workers. J. Oper. Res. Soc. **62**(6), 1038–1047 (2011)
20. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
21. Easton, F.: Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand. Prod. Oper. Manage. **23**(4), 660–675 (2014)
22. Fidanova, S., Roeva, O., Paprzycki, M.: InterCriteria analysis of ACO start strategies. In: Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, vol. 8, pp. 547–550 (2016)

23. Fidanova, S., Roeva, O., Paprzycki, M., Gepner, P.: InterCriteria analysis of ACO start startegies. In: Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, pp. 547–550 (2016)
24. Glover, F., Kochenberger, G., Laguna, M., Wubbena, T.: Selection and assignment of a skilled workforce to meet job requirements in a fixed planning period. In: MAEB04, pp. 636–641 (2004)
25. Grzybowska, K., Kovcs, G.: Sustainable supply chain—Supporting tools. In: Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, vol. 2, pp. 1321–1329 (2014)
26. Hewitt, M., Chacosky, A., Grasman, S., Thomas, B.: Integer programming techniques for solving non-linear workforce planning models with learning. Eur. J. Oper. Res. **242**(3), 942–950 (2015)
27. Hu, K., Zhang, X., Gen, M., Jo, J.: A new model for single machine scheduling with uncertain processing time. J. Intell. Manufact. **28**(3), 717–725 (2015)
28. Ikonomov, N., Vassilev, P., Roeva, O.: ICrAData software for InterCriteria analysis. Int. J. Bioautom. **22**(2) (2018) (in press)
29. Li, G., Jiang, H., He, T.: A genetic algorithm-based decomposition approach to solve an integrated equipment-workforce-service planning problem. Omega **50**, 1–17 (2015)
30. Li, R., Liu, G.: An uncertain goal programming model for machine scheduling problem. J. Intell. Manufact. **28**(3), 689–694 (2014)
31. Ning, Y., Liu, J., Yan, L.: Uncertain aggregate production planning. Soft Comput. **17**(4), 617–624 (2013)
32. Othman, M., Bhuiyan, N., Gouw, G.: Integrating workers' differences into workforce planning. Comput. Ind. Eng. **63**(4), 1096–1106 (2012)
33. Parisio, A., Jones, C.N.: A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand. Omega **53**, 97–103 (2015)
34. Roeva, O., Vassilev, P., Angelova, M., Su, J., Pencheva, T.: Comparison of different algorithms for InterCriteria relations calculation. In: 2016 IEEE 8th International Conference on Intelligent Systems, pp. 567–572 (2016)
35. Roeva, O., Fidanova, S., Paprzycki, M.: InterCriteria analysis of ACO and GA hybrid algorithms. Stud. Comput. Intell. **610**, 107–126 (2016)
36. Roeva, O., Fidanova, S., Vassilev, P., Gepner, P.: InterCriteria analysis of a model parameters identification using genetic algorithm. Proceedings of the Federated Conference on Computer Science and Information Systems **5**, 501–506 (2015)
37. Soukour, A., Devendeville, L., Lucet, C., Moukrim, A.: A Memetic algorithm for staff scheduling problem in airport security service. Expert Syst. Appl. **40**(18), 7504–7512 (2013)
38. Todinova, S., Mavrov, D., Krumova, S., Marinov, P., Atanassova, V., Atanassov, K., Taneva, S.G.: Blood plasma thermograms dataset analysis by means of InterCriteria and correlation analyses for the case of colorectal cancer. Int. J. Bioautom. **20**(1), 115–124 (2016)
39. Yang, G., Tang, W., Zhao, R.: An uncertain workforce planning problem with job satisfaction. Int. J. Mach. Learn. Cybern. (2016). https://doi.org/10.1007/s13042-016-0539-6
40. Zaharieva, B., Doukovska, L., Ribagin, S., Radeva, I.: InterCriteria decision making approach for Behterev's disease analysis. Int. J. Bioautom. **22**(2) (2018) (in press)
41. Zhou, C., Tang, W., Zhao, R.: An uncertain search model for recruitment problem with enterprise performance. J. Intell. Manufact. **28**(3), 295–704 (2014)