1 Introduction

One of the most important decision making problem, common for all branches of industry, is workforce planning. The workforce planing is a part of the human resource management. It includes multiple level of complexity, therefore it is a hard optimization problem (NP-hard). This problem consists of two decision sets: selection and assignment. The first set shows selected employees from available workers. The assignment set shows which worker which job will perform. The aim is to fulfil the work requirements with minimal assignment cost.

As we mentioned the problem is a hard optimization problem with strong constraints and is impossible to be solved with exact methods or traditional numerical methods for instances with realistic size. These kind of methods can be apply only on some simplified variants of the problem. A deterministic workforce planing problem is studied in [1, 2]. Workforce planing models are reformulated as mixed integer programming in [1]. The authors show that the mixed integer program is much easier to solve than the non-linear program. In [2] the model includes workers differences and the possibility of workers training and upgrading. In [3, 4] a variant with random demands of the problem is proposed. Two stage program of scheduling and allocating with random demands is considered in [3]. Other variant of the problem is to include uncertainty [5–9]. Most of the authors simplify the problem by omitting some of the constraints. In [10] a mixed linear programming is applied and in [4] a decomposition method is applied. For the more complex non-linear workforce planning problems, the convex methods are not applicable.

Nowadays, nature-inspired metaheuristic methods receive great attention [11–15]. In considered here problem some heuristic method including genetic algorithm [16, 17], memetic algorithm [18], scatter search [16] etc., are applied.

So far the Ant Colony Optimization (ACO) algorithm is proved to be very effective solving various complex optimization problems [19, 20]. In our previous work [21] we propose ACO algorithm for workforce planning. We have considered the variant of the workforce planning problem proposed in [16]. Current paper is the continuation of [21]. We propose a hybrid ACO algorithm which is a combination of ACO with a local search procedure. The aim is to improve the algorithm performance.

In order to compare the proposed hybrid ACO algorithm with different local search procedures the approach named InterCriteria Analysis (ICrA) is used. ICrA 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 [22]. The approach is based on the apparatus of the index matrices and the intuitionistic fuzzy sets, two approaches that have been actively researched and applied [23–28].

For the first time ICrA has been applied for the purposes of temporal, threshold and trends analyses of an economic case-study of European Union member states' competitiveness [29–31]. The approach already has a lot of different applications [32–34]. ICrA is applied for comparison of different metaheuristics as GAs and ACO [35, 36], too.

In this paper ICrA is applied for analysis of an ACO algorithm for workforce planing combined with various local search procedures. The aim is to analyze the algorithm performance according the local search procedures and to study the correlations between the different variants.

The rest of the paper is organized as follows. In Sect. 2 the mathematical description of the problem is presented. In Sect. 3 hybrid ACO algorithm for workforce planing problem is proposed. In Sect. 4 a brief discussion on InterCriteria Analysis background is done. Section 5 shows numerical calculations, ACO algorithms comparisons and discussion of the results. In Sect. 6 results form InterCriteria Analysis application are discussed. A conclusion and directions for future work are done in Sect. 7.

2 Definition of the Workforce Planning Problem

In this paper we solve the workforce planning problem proposed in [16, 37]. The set of jobs $J = \{1, \ldots, m\}$ must be completed during a fixed period of time. The job j requires d_j hours to be completed. $I = \{1, \ldots, n\}$ is the set of workers, candidates to be assigned. Every worker must perform every of assigned to him job minimum h_{min} hours to can work in efficient way. Availability of the worker i is s_i hours. One worker can be assigned to maximum j_{max} jobs. The set A_i shows the jobs, that worker i is qualified. Maximum t workers can be assigned during the planed period, or at most t workers may be selected from the set I of workers. The selected workers need to be capable to complete all the jobs. The aim is to find feasible solution, that optimizes the objective function.

Let c_{ij} is the cost of assigning the worker i to the job j. The mathematical model of the workforce planing problem can be described as follows:

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

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

$$z_{ij} = \text{number of hours that worker } i$$

$$\text{is assigned to perform job } j$$

$$Q_j = \text{set of workers qualified to perform job } j$$

$$\text{Minimize} \sum_{i \in I} \sum_{j \in A_i} c_{ij} \cdot x_{ij} \tag{1}$$

subject to

$$\sum_{j\in A_i} z_{ij} \le s_i{\cdot}y_i \;\; i \in I \tag{2}$$

$$\sum_{i\in Q_j} z_{ij} \ge d_j \;\; j \in J \tag{3}$$

$$\sum_{j\in A_i} x_{ij} \le j_{max}{\cdot}y_i \;\; i \in I \tag{4}$$

$$h_{min}{\cdot}x_{ij} \le z_{ij} \le s_i{\cdot}x_{ij} \;\; i \in I, j \in A_i \tag{5}$$

$$\sum_{i\in I} y_i \le t \tag{6}$$

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

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

The same model can be used with different objective functions. Minimization of total assignment cost is the aim of this paper. 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 hall jobs to be finished.

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

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}), in this case it is more easier to find feasible solution, than for unstructured problems (d_j and h_{min} are not related).

3 Hybrid Ant Colony Optimization Algorithm

The ACO is a nature inspired method. It is metaheuristics methodology following the behaviour of real ants looking for a food. Real ants use chemical substance, called pheromone, to mark their path ant to can return back. An ant moves in random way and when it detects a previously laid pheromone it decides whether to follow it and reinforce it with a new added pheromone. Thus the more ants follow a given trail, the more attractive that trail becomes. Using their collective intelligent the ants can find a shorter path between the source of the food and the nest.

3.1 Main ACO Algorithm

A lot of problems coming from real life and industry needs exponential number of calculations. It is not practical to solve them with exact methods or traditional numerical methods when the problem is large. Therefore the only option is to be applied some metaheuristics. The goal is to find a good solution for a reasonable time [38].

First idea to use ant behaviour for solving optimization problems is proposed by Marco Dorigo [39]. Later some modifications are proposed, mainly in pheromone updating rules [38]. The basic in ACO methodology is the simulation of 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:

- Appropriate representation of the problem by a graph;
- Appropriate pheromone placement on the nodes or on the arcs of the graph;
- Suitable 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 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 $i, j = 1, \ldots, n$.

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

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 [38].

The main pheromone trail update rule is:

$$\tau_{i,j} \leftarrow \rho\tau_{i,j} + \Delta\tau_{i,j}, \tag{9}$$

where ρ decreases the value of the pheromone, like the evaporation in a nature. $\Delta\tau_{i,j}$ is a new 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.

The starting node for every ant is randomly chosen. It is a diversification of the search. Because the random start a relatively few number of ants can be used, comparing with 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 best so far solutions.

3.2 ACO Algorithm for Workforce Planning

In this section we will apply the ACO algorithm for workforce planing from our previous work [21], which is without local search procedure. One of the main points 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) corresponds worker with number i to be assigned to the job j for time z. The graph of the problem is asymmetric, because the maximal value of z depends of the value of j, different jobs needs different time to be completed. At the beginning of every iteration every ant starts to construct their solution, from random node of the graph of the problem. For every ant are generated three random numbers. The first random number corresponds to the worker we assign and is in the interval $[0, \dots, n]$. The second random number corresponds to the job which this worker will perform and is in the interval $[0, \dots, m]$. We verify if the worker is qualified to perform the job, if not, we chose in a random way another job. The third random number corresponds to the number of hours worker i is assigned to performs the job j and is in the interval $[h_{min}, \dots, \min\{d_j, s_i\}]$. After, the ant applies the transition probability rule to include next nodes in the partial solution, till the solution is completed, or there is not a possibility to include new node.

We propose the following heuristic information:

$$\eta_{ijl} = \begin{cases} l/c_{ij} & l = z_{ij} \\ 0 & otherwise \end{cases} \tag{10}$$

This heuristic information stimulates to assign the most cheapest worker as longer as possible. the node with a highest probability is chosen to be the next node, included in the solution. When there are several candidate nodes with a same probability, the next node is chosen between them in a random way.

When a new node is included we take in to account the problem constraints as follows: how many workers are assigned till now; how many time slots every worker is assigned till now; how many time slots are assigned per job till now. When some move of the ant do not meets the problem constraints, then the probability of this move is set to be 0. If for all possible nodes the value of the transition probability is 0, it is impossible to include new node in the solution and the solution construction 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 be equal to -1.

The ants constructed feasible solutions deposed a new pheromone on the elements of their solutions. The new 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 graph belonging to solutions with less value of the objective function, receive more pheromone than others and become more desirable in the next iteration.

At the end of every iteration we compare the iteration 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 algorithm is the number of iterations.

3.3 Local Search Procedure

The our main contribution in this paper is the hybridization of the ACO algorithm with a local search procedure. The aim of the local search is to decrease the time to find the best solution and eventually to improve the achieved solutions.

We apply local search procedure only on infeasible solutions and only one time disregarding the new solution is feasible or not. Thus, our local search is not time consuming. The workforce planning is a problem with strong constraints and part of the ants do not succeed to find feasible solution. With our local search procedure the possibility to find feasible solution increases and thus increases the chance to improve current solution.

If the solution is not feasible we remove part of the assigned workers and after that we assign in their place new workers. The workers which will be removed are chosen randomly. On this partial solution we assign new workers applying the rules of ant algorithm. The ACO algorithm is a stochastic algorithm, therefore the new constructed solution is different from previous one with a high probability.

We propose three variants of local search procedure:

- removed workers are quarter of all assigned workers;
- removed workers are half of all assigned workers;
- all assigned workers are removed.

4 InterCriteria Analysis

According to [22, 40–42], we will obtain an Intuitionistic Fuzzy Pair (IFP) as the degrees of "agreement" and "disagreement" between two criteria applied on different objects. An IFP is an ordered pair of real non-negative numbers $\langle a, b\rangle$ such that:

$$a + b \leq 1.$$

Let us be given an IM [43] whose index sets for rows consist of the names of the criteria and for columns—objects. We will obtain an IM with index sets consisting of the names of the criteria both for rows and for columns. The elements IFPs of this IM corresponds to the degrees of "agreement" and degrees of "disagreement" of the considered criteria.

The following two points are supposed:

1. All criteria provide an evaluation for all objects and all these evaluations are available.
2. All the evaluations of a given criteria can be compared amongst themselves.

Further, by O we denote the set of all objects $O_1, O_2, \ldots, O_n$ being evaluated, and by $C(O)$ the set of values assigned by a given criteria C to the objects, i.e.

$$O \stackrel{\text{def}}{=} \{O_1, O_2, \ldots, O_n\},$$

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

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

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

In order to find the degrees of "agreement" of two criteria the vector of all internal comparisons of each criteria is constructed. This vector fulfil exactly one of the following three relations—R, $\overline{R}$ and $\tilde{R}$. For a fixed criterion C and any ordered pair $\langle x, y\rangle \in C^*(O)$ it is required:

$$\langle x, y\rangle \in R \Leftrightarrow \langle y, x\rangle \in \overline{R} \tag{12}$$

$$\langle x, y\rangle \in \tilde{R} \Leftrightarrow \langle x, y\rangle \notin (R \cup \overline{R}) \tag{13}$$

$$R \cup \overline{R} \cup \tilde{R} = C^*(O) \tag{14}$$

From the above it is seen that We only need to consider a subset of $C(O) \times C(O)$ for the effective calculation of $V(C)$ (vector of internal comparisons). From Eqs. (12)–(14) it follows that if we know what is the relation between x and y, we also know what is the relation between y and x. Thus, we will only consider lexicographically ordered pairs $\langle x, y\rangle$.

Let:

$$C_{i,j} = \langle C(O_i), C(O_j)\rangle.$$

We construct the vector with exactly $\frac{n(n-1)}{2}$ elements:

$$V(C) = \{C_{1,2}, C_{1,3}, \ldots, C_{1,n}, C_{2,3}, C_{2,4}, \ldots, C_{2,n},$$

$$C_{3,4}, \ldots, C_{3,n}, \ldots, C_{n-1,n}\}.$$

for a fixed criterion C.

Further, we replace the vector $V(C)$ with $\hat{V}(C)$, where for each $1 \leq k \leq \frac{n(n-1)}{2}$ for the k-th component it is true:

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

We determine the degree of "agreement" ($\mu_{C,C'}$) between the two criteria as the number of matching components. 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" ($\nu_{C,C'}$) is the number of components of opposing signs in the two vectors. This may also be done in various ways.

It is obvious that:

$$\mu_{C,C'} = \mu_{C',C},$$

$$\nu_{C,C'} = \nu_{C',C},$$

and $\langle \mu_{C,C'}, \nu_{C,C'} \rangle$ is an IFP.

The difference

$$\pi_{C,C'} = 1 - \mu_{C,C'} - \nu_{C,C'} \tag{15}$$

is considered as a degree of "uncertainty".

5 Results

5.1 Numerical Results

In this section test results are reported and compared with ACO algorithm without local search procedure. The software which realizes the algorithm is written in C computer language and is run on Pentium desktop computer at 2.8 GHz with 4 GB RAM.

We use the artificially generated problem instances considered in [16]. The test instances characteristics are shown in Table 1.

The set of test problems consists of ten structured and ten unstructured problems. The structured problems are enumerated from $S01$ to $S10$ and unstructured problems are enumerated from $U01$ to $U10$. For structured problems d_j is proportional to h_{min}. In our previous work [21] we show that our ACO algorithm outperforms the genetic and scatter search algorithms from [16].

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]

The number of iterations is a stopping criteria for our hybrid ACO algorithm. The number of iterations is fixed to be maximum 100. In Table 2 the parameter settings of our ACO algorithm are shown. The values are fixed experimentally.

Further, the problem instances are enumerated as $S20_{01}$–$S20_{10}$ and $U20_{01}$ to $U20_{10}$, taking into account the number of ants.

The workforce problem has very restrictive constraints. Therefore only 2–3 of the ants, per iteration, find feasible solution. Sometimes exist iterations without any feasible solution. Its complicates the search process. Our aim is to decrease the number of unfeasible solutions and thus to increase the possibility ants to find good solutions and so to decrease needed number of iterations to find good solution. We observe that after the local search procedure applied on the first iteration, the number of unfeasible solutions in a next iterations decrease. It is another reason the computation time does not increase significantly. We are dealing with four cases:

- without local search procedure (ACO);
- local search procedure when the number of removed workers is quarter from the number of all assigned workers (ACO quarter);
- local search procedure when the number of removed workers is half from the number of all assigned workers (ACO half);
- local search procedure when all assigned workers are removed and the solution is constructed from the beginning (ACO restart);

We perform 30 independent runs with every one of the four cases, because the algorithm is stochastic ant to guarantee the robustness of the average results. We apply ANOVA test for statistical analysis to guarantee the significance of the achieved results. The obtained results are presented in Tables 3 and 4. Table 3 presents the minimal number of iterations to achieve the best solution and Table 4—the computation time needed to achieve the best solution.

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

Table 3 Minimal number of iterations to achieve the best solution

	ACO	ACO quarter	ACO half	ACO restart
$S20_{01}$	13	10	15	16
$S20_{02}$	17	28	28	35
$S20_{03}$	29	27	37	33
$S20_{04}$	77	66	41	23
$S20_{05}$	21	21	4	14
$S20_{06}$	21	13	20	1
$S20_{07}$	43	34	29	40
$S20_{08}$	57	15	50	33
$S20_{09}$	36	28	22	48
$S20_{10}$	26	19	16	35
$U20_{01}$	17	23	11	21
$U20_{02}$	17	16	12	15
$U20_{03}$	28	22	20	48
$U20_{04}$	41	56	28	28
$U20_{05}$	14	20	15	4
$U20_{06}$	46	46	45	20
$U20_{07}$	29	44	37	39
$U20_{08}$	11	14	16	26
$U20_{09}$	46	68	41	42
$U20_{10}$	30	30	30	30

We are interested of the number of iterations for finding the best result. It can be very different for different test problems, so we will use ranking of the algorithms. The variant of our hybrid algorithm is on the first place, if it achieves the best solution with less average number of iterations over 30 runs, according other cases and we assign to it 1, we assign 2 to the case on the second place, 3 to the case on the third place and 4 to the case with most number of iterations. On some cases can be assigned same numbers if the number of iterations to find the best solution is the same. We sum the ranking of the cases over all 20 test problems to find final ranking of the different cases of the hybrid algorithm.

We observe that the local search procedure decreases the number of unfeasible solutions, found by traditional ACO algorithm in the next iterations, thus when the number of iterations increase, the need of local search procedure decreases. On Table 5 we report the achieved ranking of different cases of our hybrid algorithm. As we mentioned above, with ACO quarter we call the case when quarter of the workers are removed. ACO half is the case when half of the workers are removed. ACO restart is the case when all workers are removed. It is like to restart the solution construction, to construct the solution from the beginning.

Table 4 Computation time needed to achieve the best solution

	ACO	ACO quarter	ACO half	ACO restart
$S20_{01}$	1.20	0.94	0.96	2.29
$S20_{02}$	3.94	8.62	6.22	14.75
$S20_{03}$	5.19	5.79	11.93	3.06
$S20_{04}$	3.06	16.66	7.00	6.11
$S20_{05}$	0.63	1.312	0.396	0.90
$S20_{06}$	2.48	2.12	2.64	0.59
$S20_{07}$	6.78	4.82	6.78	6.60
$S20_{08}$	6.38	1.87	10.42	8.59
$S20_{09}$	4.68	5.31	4.48	5.70
$S20_{10}$	1.45	1.25	1.28	10.43
$U20_{01}$	3.10	4.48	2.00	2.50
$U20_{02}$	1.98	1.18	0.92	0.93
$U20_{03}$	2.14	2.41	1.54	1.88
$U20_{04}$	3.08	3.35	3.12	3.47
$U20_{05}$	1.55	2.76	2.06	1.056
$U20_{06}$	10.92	11.8	4.36	7.05
$U20_{07}$	4.22	6.55	3.54	3.27
$U20_{08}$	0.89	1.48	1.19	1.77
$U20_{09}$	6.48	8.72	7.10	7.21
$U20_{10}$	3.74	3.88	3.69	10.00

One of the main questions is how many worker to remove from unfeasible solution, so that the new constructed solution to be feasible and close to the best one. We calculate the ranking, regarding the average number of iterations to find best solution over 30 runs of the test. When more than half of the ants find unfeasible solutions, the deviation from the average is larger compared to the tests when the most of the ants achieve feasible solutions. Table 5 shows that the local search procedure decreases the number of iterations needed to find the best solution, when more than half of the workers are removed. The traditional ACO algorithm and hybrid ACO with removing quarter of the workers are 4 times on the first place when either, by chance, the algorithm find the best solution on the first iteration, or all ants find feasible solutions. We observe that the both cases are on the third and forth place 12 times. This means that removing less than half of the workers is not enough to construct feasible solution. The ACO algorithm with removed half of the workers 15 times is on the first or second place and only one time is on the fourth place, which means that it performs much better than previous two cases. When all workers are removed the achieved ranking is similar to the case when half of the workers are removed. Let the maximal number of assigned workers is t. Thus the every one

Table 5 Hybrid ACO ranking over number of iterations

	ACO	ACO quarter	ACO half	ACO restart
First place	4 times	4 times	8 times	8 times
Second place	4 times	4 times	7 times	6 times
Third place	8 times	6 times	4 times	3 times
Forth place	4 times	6 times	1 times	3 times
Ranking	52	54	38	41

Table 6 Hybrid ACO comparison over computation time

	ACO	ACO quarter	ACO half	ACO restart
First place	4 times	3 times	10 times	6 times
Second place	7 times	5 times	5 times	5 times
Third place	8 times	4 times	3 times	4 times
Forth place	1 times	8 times	2 times	5 times
Average time (s)	82.244	93.98	79.63	103.012

of the solutions consists about t workers. If all of the workers are removed, the ant need to add new workers on their place which number is about t. When half of the workers are removed, then the ant will add about $t/2$ new workers. The computation time to remove and add about $t/2$ workers is about two times less than to remove and add about t workers. Thus we can conclude that the local search procedure with removing half of the workers performs better than other cases.

Another way for comparison is the computation time. For every test problem and every case we calculate the average time to achieve best solution over 30 runs. In Table IV we did similar ranking as in Table 5, but taking in to account the computation time instead number of iterations. Regarding the Table 6 the ranking according the time is similar to the ranking according to the number of iterations from the Table 5. The best performance is when half of the worker are removed in the local search procedure and the worst performance is when quarter of the workers are removed. The local search procedure with removing half of the worker is on the first place 10 times and on the forth place only 2 times. The local search procedure with removing quarter of the workers is on the first place 3 times and on the forth place 8 times. Regarding the computation time the local search procedure with removing half of the workers again is the best, but the worst is the local search procedure with removing all workers. Reconstructing a solution from the beginning takes more time than to reconstruct partial solution, therefore ACO algorithm with local search procedure removing all workers performs worst. The results from Table 6 show that removing only quarter of the workers from the solution is not enough for construction of good solution and is time consuming comparing with traditional ACO algorithm. According the both types of comparison, ranking and computation time ACO with local search procedure removing half of the assigned workers performs best.

6 Results from InterCriteria Analisys

The four considered here cases of ACO hybrid algorithms will further be denoted as:

- without local search procedure—$ACO1$ (ACO);
- local search procedure when the number of removed workers is quarter from the number of all assigned workers—$ACO2$ (ACO quarter);
- local search procedure when the number of removed workers is half from the number of all assigned workers—$ACO3$ (ACO half);
- local search procedure when all assigned workers are removed and the solution is constructed from the beginning—$ACO4$ (ACO restart).

The cross-platform software for ICrA approach, ICrAData, is used [44]. The input index matrices for ICrA have the form of Tables 3 and 4. The test problems $S20_{01}$ to $S20_{10}$ and $U20_{01}$ to $U20_{10}$ are considered as objects $O = \{O_1, O_2, \ldots, O_{10}\}$. The four ACO hybrid algorithms ($ACO1 - ACO4$) are considered as criteria $C(O) = \{C(O_1), C(O_2), \ldots, C(O_{10})\}$. In a result, ICrA gives the relation between the proposed ACO hybrid algorithms. The hybrid algorithms are compared based on the obtained results according to the number of iterations and according to the computation time.

6.1 InterCriteria Analisys of Different Hybrid ACO Algorithms According to the Number of Iterations

In Tables 7 and 8 the results of ICrA are presented. To the notation of the hybrid algorithms "*it*" is added (iterations). The obtained ICrA results are analysed based on the proposed in [45] consonance and dissonance scale. The scheme for defining the consonance and dissonance between each pair of criteria is presented in Table 9.

The obtained ICrA results are visualized on Fig. 1 within the specific triangular geometrical interpretation of IFSs, thus allowing us to order these results according simultaneously to the degrees of "agreement" $\mu_{C,C'}$ and "disagreement" $\nu_{C,C'}$ of the intuitionistic fuzzy pairs.

The results show that there is very small values of π, i.e., there is no significant degree of "uncertainty" in the data.

Table 7 Index matrix for $\mu_{C,C'}$ (intuitionistic fuzzy estimations)

$\mu_{C,C'}$	$ACO1_{it}$	$ACO2_{it}$	$ACO3_{it}$	$ACO4_{it}$
$ACO1_{it}$	**1.00**	0.76	0.78	0.60
$ACO2_{it}$	0.76	**1.00**	0.71	0.64
$ACO3_{it}$	0.78	0.71	**1.00**	0.62
$ACO4_{it}$	0.60	0.64	0.62	**1.00**

Table 8 Index matrix for $\nu_{C,C'}$ (intuitionistic fuzzy estimations)

$\nu_{C,C'}$	$ACO1_{it}$	$ACO2_{it}$	$ACO3_{it}$	$ACO4_{it}$
$ACO1_{it}$	**0.00**	0.20	0.17	0.35
$ACO2_{it}$	0.20	**0.00**	0.26	0.34
$ACO3_{it}$	0.17	0.26	**0.00**	0.34
$ACO4_{it}$	0.35	0.34	0.34	**0.00**

Table 9 Consonance and dissonance scale [45]

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

Fig. 1 Presentation of ICrA results in the intuitionistic fuzzy interpretation triangle

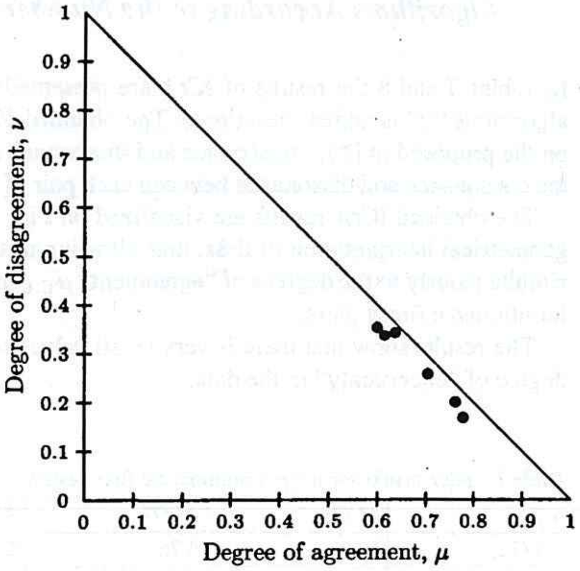

Table 10 ACO hybrid algorithms in weak positive consonance

Pair of ACO hybrid algorithms	$\mu_{C,C'}$-value
$ACO1_{it}$-$ACO3_{it}$	0.78
$ACO1_{it}$-$ACO2_{it}$	0.76

Table 11 ACO hybrid algorithms in weak dissonance

Pair of ACO hybrid algorithms	$\mu_{C,C'}$-value
$ACO2_{it}$-$ACO3_{it}$	0.71

Table 12 ACO hybrid algorithms in dissonance

Pair of ACO hybrid algorithms	$\mu_{C,C'}$-value
$ACO2_{it}$-$ACO4_{it}$	0.64
$ACO3_{it}$-$ACO4_{it}$	0.62
$ACO1_{it}$-$ACO4_{it}$	0.60

Table 13 Index matrix for $\mu_{C,C'}$ (Intuitionistic fuzzy estimations)

$\mu_{C,C'}$	$ACO1_s$	$ACO2_s$	$ACO3_s$	$ACO4_s$
$ACO1_s$	**1.00**	0.77	0.83	0.71
$ACO2_s$	0.77	**1.00**	0.79	0.67
$ACO3_s$	0.83	0.79	**1.00**	0.73
$ACO4_s$	0.71	0.67	0.73	**1.00**

Based on ICrA it is shown that the $ACO1_{it}$-$ACO3_{it}$ and $ACO1_{it}$-$ACO2_{it}$ are the pairs in weak positive consonance (see Table 10). These hybrid algorithms—ACO without local search procedure, ACO quarter and ACO half—show similar performance.

The criteria pair $ACO2_{it}$-$ACO3_{it}$ is in weak dissonance (see Table 11).

Other results show that the pairs $ACO1_{it}$-$ACO4_{it}$, $ACO2_{it}$-$ACO4_{it}$ and $ACO3_{it}$-$ACO4_{it}$ are in strong dissonance, i.e., there is no correlation between them (see Table 12). These results means that the hybrid algorithm ACO restart shows different behaviour compared to the other three ACO algorithms.

6.2 InterCriteria Analisys of Different Hybrid ACO Algorithms According to the Computation Time

Based on the results presented in Table 4 the ICrA is applied. The obtained numerical ICrA results are presented in Tables 13 and 14. To the notation of the hybrid algorithms "*s*" is added (seconds). The analysis is based again on the proposed in [45] scale (Table 9).

The triangular geometrical interpretation of the obtained results is shown on Fig. 2.

Table 14 Index matrix for $\nu_{C,C'}$ (Intuitionistic fuzzy estimations)

$\nu_{C,C'}$	$ACO1_s$	$ACO2_s$	$ACO3_s$	$ACO4_s$
$ACO1_s$	0.00	0.23	0.17	0.29
$ACO2_s$	0.23	0.00	0.21	0.33
$ACO3_s$	0.17	0.21	0.00	0.27
$ACO4_s$	0.29	0.33	0.27	0.00

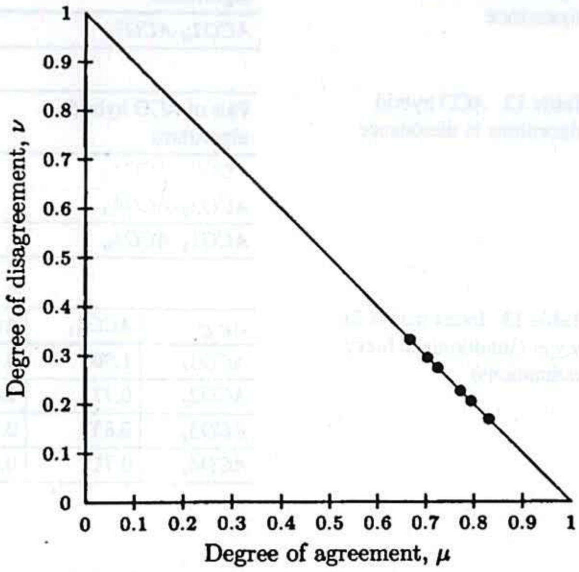

Fig. 2 Presentation of ICrA results in the intuitionistic fuzzy interpretation triangle

Table 15 ACO hybrid algorithms in weak positive consonance

Pair of ACO hybrid algorithms	$\mu_{C,C'}$-value
$ACO1_s$-$ACO3_s$	0.83
$ACO2_s$-$ACO3_s$	0.79
$ACO1_s$-$ACO2_s$	0.77

In this case $\pi = 0$, i.e., there is no "uncertainty" in the data.

Analysis of the results shows that the pairs of hybrid ACO algorithms e.g. $ACO1_s$-$ACO3_s$ and $ACO1_s$-$ACO2_s$ are in weak positive consonance (see Table 15). We obtain the same results as in the case of comparison based on minimal number of iterations to achieve the best solution. In the case of comparison of computation time needed to achieve the best solution a stronger positive consonance is observed: $\mu_{ACO1_{it},ACO3_{it}} = 0.78$ versus $\mu_{ACO1_s,ACO3_s} = 0.83$ and $\mu_{ACO1_{it},ACO2_{it}} = 0.76$ versus $\mu_{ACO1_s,ACO2_s} = 0.77$. Moreover, the criteria pair $ACO2_s$-$ACO3_s$ that in case of comparison based on minimal number of iterations to achieve the best solution shows weak dissonance, in this case shows weak positive consonance with $\mu_{ACO2_{it},ACO3_{it}} = 0.79$ (Table 15).

Table 16 ACO hybrid algorithms in weak dissonance

Pair of ACO hybrid algorithms	$\mu_{C,C'}$-value
ACO3$_s$-ACO4$_s$	0.73
ACO1$_s$-ACO4$_s$	0.71

Table 17 ACO hybrid algorithms in dissonance

Pair of ACO hybrid algorithms	$\mu_{C,C'}$-value
ACO2$_s$-ACO4$_s$	0.67

Analogically, two criteria pairs, *ACO4$_s$-ACO3$_s$* and *ACO1$_s$-ACO4$_s$*, in this case have moved from dissonance to weak dissonance (Table 16). The criteria pair *ACO2$_s$-ACO4$_s$* is in strong dissonance, i.e., there is no correlation between hybrid algorithms ACO quarter and ACO restart (see Table 17).

The presented ICrA results show that there is a difference when the hybrid algorithms are compared based on minimal number of iterations and based on computation time. The comparison based on the computation time is more realistic. In this case, it is compared not only the time to achieve the best solution to compare, but also the time for run of one iteration.

7 Conclusion

In this paper we propose Hybrid ACO algorithm for solving workforce assignment problem. The ACO algorithm is combined with appropriate local search procedure. The local search procedure is applied only on unfeasible solutions. The main idea is to remove part of the workers in the solution in a random way and to include new workers in their place. Three variants of the local search procedure are compared with traditional ACO algorithm, removing quarter of the assigned workers, removing half of the assigned workers and removing all assigned workers. The local search procedure with removing half of the assigned workers performs better than other algorithms.

Further InterCriteria analysis is performed over obtained numerical results solving workforce assignment problem. Additional comparison of the proposed ACO hybrid algorithms is done. Two cases are considered – comparison based on minimal number of iterations to achieve the best solution and based on the computation time needed to achieve the best solution. ICrA results confirm the previously discussed ACO hybrid algorithms performance. Moreover, it is shown that the comparison based on the computation time needed to achieve the best solution is more realistic than the comparison based on minimal number of iterations.

Acknowledgements Work presented here is partially supported by the National Science Fund of Bulgaria under grants DFNI-DN02/10 "New Instruments for Knowledge Discovery from Data and by the Polish-Bulgarian collaborative grant "Practical aspects for scientific computing".

References

1. 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)
2. Othman, M., Bhuiyan, N., Gouw, G.: Integrating workers' differences into workforce planning. Comput. Ind. Eng. **63**(4), 1096–1106 (2012)
3. Campbell, G.: A two-stage stochastic program for scheduling and allocating cross-trained workers. J. Oper. Res. Soc. **62**(6), 1038–1047 (2011)
4. Parisio, A., Jones, C.N.: A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand. Omega (Elsevier) **53**, 97–103 (2015)
5. Hu, K., Zhang, X., Gen, M., Jo, J.: A new model for single machine scheduling with uncertain processing time. J. Intell. Manuf. (Springer) **28**(3), 717–725 (2015)
6. Li, R., Liu, G.: An uncertain goal programming model for machine scheduling problem. J. Intell Manuf. (Springer) **28**(3), 689–694 (2014)
7. Ning, Y., Liu, J., Yan, L.: Uncertain aggregate production planning. Soft Comput. (Springer) **17**(4), 617–624 (2013)
8. Yang, G., Tang, W., Zhao, R.: An uncertain workforce planning problem with job satisfaction. Int. J. Mach. Learn. Cybern. (Springer) (2016). https://doi.org/10.1007/s13042-016-0539-6, http://rd.springer.com/article/10.1007/s13042-016-0539-6
9. Zhou, C., Tang, W., Zhao, R.: An uncertain search model for recruitment problem with enterprise performance. J. Intell. Manuf. (Springer) **28**(3), 295–704 (2014). https://doi.org/10.1007/s10845-014-0997-1
10. Easton, F.: Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand. Prod. Oper. Manag. **23**(4), 660–675 (2014)
11. Albayrak, G., Özdemir, İ.: A state of art review on metaheuristic methods in time-cost trade-off problems. Int. J. Struct. Civil Eng. Res. **6**(1), 30–34 (2017)
12. Mucherino, A., Fidanova, S., Ganzha, M.: Introducing the environment in ant colony optimization, recent advances in computational optimization, studies in computational. Intelligence **655**, 147–158 (2016)
13. Roeva, O., Atanassova, V.: Cuckoo search algorithm for model parameter identification. Int. J. Bioautomation **20**(4), 483–492 (2016)
14. Tilahun, S.L., Ngnotchouye, J.M.T.: Firefly algorithm for discrete optimization problems: a survey. J. Civil Eng. **21**(2), 535–545 (2017)
15. Toimil, D., Gmes, A.: Review of metaheuristics applied to heat exchanger network design. Int. Trans. Oper. Res. **24**(1–2), 7–26 (2017)
16. Alba, E., Luque, G., Luna, F.: Parallel metaheuristics for workforce planning. J. Math. Model. Algorithms (Springer) **6**(3), 509–528 (2007)
17. Li, G., Jiang, H., He, T.: A genetic algorithm-based decomposition approach to solve an integrated equipment-workforce-service planning problem. Omega (Elsevier) **50**, 1–17 (2015)
18. 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)
19. 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)
20. 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)

21. Fidanova, S., Luquq, G., Roeva, O., Paprzycki, M., Gepner, P.: Ant colony optimization algorithm for workforce planning. In: FedCSIS'2017, IEEE Xplorer, IEEE Catalog Number CFP1585N-ART, pp. 415–419 (2017)
22. 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 IFSs and GNs **11**, 1–8 (2014)
23. Traneva, V., Atanassova, V., Tranev, S.: Index matrices as a decision-making tool for job appointment. In: G. Nikolov et al. (eds.) NMA 2018, LNCS , vol. 11189, pp. 1–9. Springer Nature Switzerland AG (2019)
24. Traneva, V., Tranev, S., Atanassova, V.: An intuitionistic fuzzy approach to the hungarian algorithm. In: Nikolov G. et al. (eds.) NMA 2018, LNCS, vol. 11189, pp. 1–9. Springer Nature Switzerland AG (2019)
25. Atanassov, K.T., Vassilev, P.: On the intuitionistic fuzzy sets of n-th type. In: Gaweda A., Kacprzyk J., Rutkowski L., Yen G. (eds.) Advances in Data Analysis with Computational Intelligence Methods. Studies in Computational Intelligence, vol. 738, pp. 265–274. Springer, Cham (2018)
26. Vassilev, P., Ribagin, S.: A note on intuitionistic fuzzy modal-like operators generated by power mean. In: Kacprzyk J., Szmidt E., Zadrony S., Atanassov K., Krawczak M. (eds.) Advances in Fuzzy Logic and Technology 2017. EUSFLAT 2017, IWIFSGN 2017. Advances in Intelligent Systems and Computing, vol. 643, pp. 470–475. Springer, Cham (2018)
27. Marinov, E., Vassilev, P., Atanassov, K.: On separability of intuitionistic fuzzy sets. In: Novel Developments in Uncertainty Representation and Processing, Advances in Intelligent Systems and Computing, vol. 401, pp. 111–123. Springer, Cham (2106)
28. Vassilev, P.: A note on new distances between intuitionistic fuzzy sets. Notes Intuit. Fuzzy Sets **21**(5), 11–15 (2015)
29. Atanassova, V., Mavrov, D., Doukovska, L., Atanassov, K.: Discussion on the threshold values in the intercriteria decision making approach. Notes on Intuit. Fuzzy Sets **20**(2), 94–99 (2014)
30. Atanassova, V., Doukovska, L., Atanassov, K., Mavrov, D.: Intercriteria decision making approach to EU member states competitiveness analysis. In: Shishkov, B. (ed.) Proceedings of the International Symposium on Business Modeling and Software Design—BMSD'14, pp. 289–294 (2014)
31. Atanassova, V. Doukovska, L., Karastoyanov, D., Capkovic, F.: InterCriteria decision making approach to EU member states competitiveness analysis: trend analysis. In: Angelov P. et al. (eds.) Intelligent Systems'2014, Advances in Intelligent Systems and Computing, vol. 322, pp. 107–115 (2014)
32. Roeva, O., Fidanova, S., Vassilev, P., Gepner, P.: InterCriteria analysis of a model parameters identification using genetic algorithm. In: Proceedings of the Federated Conference on Computer Science and Information Systems, vol. 5, pp. 501–506 (2015)
33. Todinova, S., Mavrov, D., Krumova, S., Marinov, P., Atanassova, V., Atanassov, K., Taneva, S.G.: Blood plasma thermograms dataset analysisby means of intercriteria and correlation analyses for the case of colorectal cancer. Int. J. Bioautomation **20**(1), 115–124 (2016)
34. Vassilev, P., Todorova, L., Andonov, V.: An auxiliary technique for intercriteria analysis via a three dimensional index matrix. Notes on Intuit. Fuzzy Sets **21**(2), 71–76 (2015)
35. Angelova, M., Roeva, O., Pencheva, T.: InterCriteria analysis of crossover and mutation rates relations in simple genetic algorithm. In: Proceedings of the Federated Conference on Computer Science and Information Systems, vol. 5, pp. 419–424 (2015)
36. Roeva, O., Fidanova, S., Paprzycki, M.: InterCriteria analysis of ACO and GA hybrid algorithms. Stud. Comput. Intell. **610**, 107–126 (2016)
37. 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)
38. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press (2004)
39. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, New York (1999)

40. Atanassov, K.: On Intuitionistic Fuzzy Sets Theory. Springer, Berlin (2012)
41. Atanassov, K.: Review and new results on intuitionistic fuzzy sets, mathematical foundations of artificial intelligence seminar, sofia, 1988. Preprint IM-MFAIS-1-88, Reprinted: Int. J. Bioautomation 20(S1), S7–S16 (2016)
42. Atanassov, K.: Intuitionistic Fuzzy Sets, VII ITKR Session, Sofia, 20–23 June 1983. Reprinted: Int. J. Bioautomation 20(S1), S1–S6 (2016)
43. Atanassov, K.: On index matrices, Part 1: standard cases. Adv. Stud. Contemp. Math. 20(2), 291–302 (2010)
44. Ikonomov, N., Vassilev, P., Roeva, O.: ICrAData—software for intercriteria analysis. Int. J. Bioautomation 22(1), 1–10 (2018)
45. Atanassov, K., Atanassova, V., Gluhchev, G.: InterCriteria analysis: ideas and problems. Notes Intuit. Fuzzy Sets 21(1), 81–88 (2015)