Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 955–959
DOI: 10.15439/2023F9586
ISSN 2300-5963 ACSIS, Vol. 35

Ant Colony Optimization for Workforce Planning with Hybridization

Stefka Fidanova
Institute of Information and Communication Technology
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 25A,
Sofia, Bulgaria
E-mail: stefka.fidanova@iict.bas.bg

Maria Ganzha
System Research Institute
Polish Academy of Sciences
Warsaw, Poland
E-mail: maria.ganzha@ibspan.waw.pl

***Abstract*—Production organization plays a key role in the success of any enterprise. Optimizing workforce planning can improve the overall organization of production. The main goal is to minimize the assignment cost of the workers who will perform the planned work. The problem is known to be NP-hard, therefore we will apply methods from the field of artificial intelligence. For this reason, most of the existing methods hardly find feasible solutions. We propose Ant Colony Optimization Algorithm with hybridization, combination with local search procedures. We compare and analyze their performance.**

***Index Terms*—Workforce Planning, Ant Colony Optimization, Metaheuristics, Hybrid Method, Local Search**

I. INTRODUCTION

PROPER management of human resources plays an important role in the organization of production. It is common problem for all industrial sectors. It is NP-hard optimization problem, which includes a lot of level of complexity. Workforce planning is the process of determining the skills and human resources needed to perform a given task. The problem consists of two parts: selection and assignment. First the employers are selected from the set of available workers. After they are assigned to jobs, which they will perform. The aim is minimization of assignment cost, while staying within the framework of work requirements. Human resource management includes workforce planning. Exact methods as well as traditional numerical methods are unable to solve this problem for instances with realistic size. These types of methods can be applied only on special simplified variants of the problem.

It exist various metaheuristic algorithms applied on workforce planning problem. They include genetic algorithm [1], memetic algorithm [11], scatter search [1] etc.

Ant Colony Optimization (ACO) algorithm is proven to be very effective solving various complex optimization problems [5], [10]. In our previous work [6], [7] we propose ACO algorithm for workforce planning. We have considered the variant of the workforce planning problem proposed in [1].

Current paper is the continuation of [7] and [8]. Other variant of hybridization is proposed. They are compared and discussed. The aim is to improve algorithm efficiency .

The rest of the paper is organized as follows. The mathematical description of the problem is presented in Section 2. ACO algorithm for workforce planing problem is presented in Section 3. Computational results, comparisons of different hybridization and discussion are done in Section 4 . A conclusion and directions for future work are proposed in Section 5.

II. WORKFORCE PLANNING PROBLEM

In this section we will give definition and description of the variant of Workforce Planing Problem (WPP) we solve. We intend the variant of the problem considered by Alba [1] and Glover [9].

There is a fixed period of time and a set of jobs $J = \{1, \ldots, m\}$. All jobs need to be finished during this period. For every job j is known that it requires d_j hours to be completed. There are workers, which are candidates for assignment to perform the jobs, the set $I = \{1, \ldots, n\}$. In terms of work quality and efficiency, each worker must work on each of their assigned jobs for a minimum of h_{min} hours. We know the availability of every worker, worker i is available for s_i hours. Workers may have different qualifications and may not be qualified for all the tasks to be performed. The set A_i contains the jobs, for which worker i is qualified. There is a limit t to the maximum number of workers that can be assigned during this period. This means that at most t workers can be selected from a set I of workers, and this must be done in such a way that they are able to perform and complete the planned work. The worker i is assigned to perform job j at c_{ij}. The purpose is to find feasible solution, that minimize assignment price, which is the objective function of this problem.

The following is the description of the mathematical model of the workforce planing problem:

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

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

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

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

Q_j = set of workers qualified to perform job j

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

Subject to

$$\sum_{j \in A_i} z_{ij} \leq s_i.y_i \qquad i \in I \tag{2}$$

$$\sum_{i \in Q_j} z_{ij} \geq d_j \qquad j \in J \tag{3}$$

$$\sum_{j \in A_i} x_{ij} \leq j_{max}.y_j \qquad i \in I \tag{4}$$

$$h_{min}.x_{ij} \leq z_{ij} \leq s_i.x_{ij} \qquad i \in I, j \in A_i \tag{5}$$

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

$$\begin{array}{ll} x_{ij} \in \{0,1\} & i \in I, j \in A_i \\ y_i \in \{0,1\} & i \in I \\ z_{ij} \geq 0 & i \in I, j \in A_i \end{array}$$

Every manufacturer strives to reduce the cost of production. This can be achieved with good organization and optimization of production process. One of the biggest costs is the cost of hiring workers. Therefore, workforce planning and optimization is a fundamental issue for every enterprise. The goal of the problem of workforce planning is the minimization of the total assignment cost, respecting the constraints. Inequality 2 represents the limitation of the number of hours the selected worker can be assigned. Inequality 3 show the completion time for hall jobs. The limitation of the number of jobs, that every worker can perform is done by the inequality 4. If a worker works too short on a job, his work will be inefficient and often of poor quality. Therefore, a minimum amount of time is required for each worker to work on each of their assigned jobs. This requirement is represented by inequality 5. There is always some reason to limit the number of workers working at the same time. This may be the available space; the amount of tools; number of machines or something else. The limitation of the number of the assigned workers is represented by inequality 6.

This mathematical model of the workforce planning problem can be used with a variety of objective functions, depending on what our goal is and what we want to optimize. Regarding the goal, there are various variants of the problem. The focus of this paper is minimization of total assignment cost. Let's $\tilde{c}_{ij}$ is the cost the worker i to performs the job j for one hour. The cost of assigning workers to complete all assigned jobs is represented by function 7. Minimizing this function is the objective function used in this paper.

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

Some of the workers may have preferences for some of the activities for which they are qualified. In this case, the objective function would be the maximum satisfaction of their desires. Another option is for the task to have two objective functions. Simultaneous minimization of the total cost of appointment and maximum satisfaction of preferences.

Workforce planning problems fall into two broad groups: structured and unstructured. The problem is structured when the time to complete a job is proportional to the minimal time the worker need to work on separate job, or parameter d_j is proportional to the parameter h_{min}. When d_j is not proportional to the parameter h_{min} the problem is unstructured. The algorithms find more frequently feasible solutions for structured problems, then for unstructured.

III. Hybrid Ant Colony Optimization Algorithm

One of the most successful methods for solving combinatorial optimization problems is Ant Colony Optimization (ACO). It is a metaheuristics, following the real ants behavior when looking for a food. Normally ants use chemical substance, called pheromone, to mark their path ant to can return back.

A. Main ACO Algorithm

NP-hard problems and in particular combinatorial optimization problems require exponential number of calculations and memory use. So large problems can not be solved for reasonable time by exact algorithms or traditional numerical methods [3].

First realization of the idea to use ant behavior is applied by Marco Dorigo [2] for solving Traveling Salesman Problem. Later some modifications and improvements are proposed, mainly in pheromone updating rules [3] and the method was applied on big variety of combinatorial optimization problems. The ACO methodology is based on the ants behavior simulation. One of the main things in the algorithm is the representation of the problem by graph, called graph of the problem. This allows solutions to be represented as paths in the graph. The problem boils down to finding a shortest path in a graph subject to given constraints.

The transition probability $P_{i,j}$ leads the ants how to choose the next node j to be added to the partial solution, when the last node selected is i. It is a product of the heuristic information $\eta_{i,j}$ and the pheromone trail quantity $\tau_{i,j}$ corresponding to the move from node i to the node j, where $i, j = 1, \ldots, n$. The transition probability formula is as follows:

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

where $Unused$ is the set of unused nodes of the problem graph, a and b are the influence of the pheromone and the heuristics information, respectively.

Equality 8 shows that the attractiveness of a node increases, when the heuristic information and/or the quantity of the pheromone related to it increases, because the probability the node to be selected increases and it becomes more advantageous.

The level of the initial pheromone is the same for all graph elements and is set to a small positive constant value τ_0, $0 < \tau_0 < 1$. The algorithm is iterative and the goal is on the next iteration the ant to try to construct new better solutions, taking in to account the information from previous iterations. At the end of every iteration the ants update the pheromone values of graph elements according the quality of the achieved solution during the iteration. Different ACO algorithms utilize different procedures for updating pheromone values [3]. A node, from the problem graph, becomes more desirable if it accumulates more pheromone, but the accumulation of too much pheromone can lead to stagnation and repetition of the same solutions without their improvement.

The main update rule for the pheromone trail level is:

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

where ρ is the evaporation parameter. It decreases the value of the old pheromone, because the old information is not so current. On the other hand, we do not lose it, but only reduce its influence. Thus we mimics evaporation in a nature and try to prevent early stagnation and help the ants to avoid local minima. $\Delta\tau_{i,j}$ is a new added pheromone, and it is proportional to the quality of the newly constructed solution. The quality of the newly constructed solutions is measured by the values of the objective function, corresponding to these solutions.

B. Workforce Planing Problem ACO Algorithm

In this section we describe ACO algorithm for workforce planning without local search procedure from our previous paper [6]. Proper graph representation of the problem play important role in ACO algorithm application. The problem is described by 3 dimensional graph. Essential is which elements of the problem are represented by the nodes and what is the meaning of the arcs. In our problem the node (i, j, z) represents the worker i assigned to the job j for time z. The maximal value of z is dependent of the completion time of job j. Completion time is different for different jobs, so the graph of the problem is asymmetric.

As we mentioned in the subsection above, an ant starts solution construction from a random node of the graph of the problem. Thus at the beginning of every iteration we generate three random numbers for every ant. The first random number belongs to the interval $[0, \dots, n]$ and shows to the worker who is chosen to be assigned. The second random number belong to the interval $[0, \dots, m]$. It is related with the job, the worker is assigned to do. In case the worker is not qualified to do this job, a new job is chosen in a random way. The third random number belong to the interval $[h_{min}, \min\{d_j, s_i\}]$ and is related with number of hours worker i is assigned to performs job j.

By traditional ACO algorithm next nodes are included applying transition probability rule. These steps are repeated till all ants construct their solutions. The termination condition of the solution construction process is the impossibility of adding new nodes without violating any of the constraints of the problem.

We propose the following heuristic information to be applied, where worker i, performs job j for time l, formula 10:

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

This heuristic information incentives the assignment of the cheaper workers for as long as possible, thereby reducing the overall cost of assigning the workers. Following the rules of ACO algorithm the next included node in the partial solution is the node with highest probability. If there happen to be several nodes with a probability equal to the maximum, then one of them is chosen at random as the next node in the partial solution. Each time a new node is included, it is checked whether the constraints of the problem are not violated, only then the new node is accepted.

If any of the constraints is not satisfied, then the value of the transition probability function corresponding to this node is set to be 0. If for all possible nodes the value of the probability function is 0 the solution construction stops, since it is impossible to include new node in the current partial solution. When the achieved solution is feasible, the value of the objective function is calculated as a sum of assignment cost of all assigned workers. The value of the objective function can not be negative. Therefore we set the value of the objective function to be -1 for infeasible solutions.

We deposit additional pheromone only on the elements of feasible solutions and it reflects the quality of the problem solution, which is measured by the value of the objective function. Workforce planning problem is a minimization problem, so the new added pheromone is proportional to the reciprocal value of the objective function:

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

So the elements of the graph of the problem, belonging to better solutions with less value of the objective function will accumulate more pheromone than others and will be more wanted in the next iteration. The global best so far solution is updated at the end of every iteration. We compare the iteration best solution with the current global best one and if the iteration best solution is better, with less value of the objective function, we accept it as a new global best solution. In our application as end condition we apply number of iterations. When the algorithm reaches the pre-fixed number of iterations, it stops further calculations.

IV. Local Search Procedures

A common practice is to combine a metaheuristic algorithm with some other algorithm. This can be another metaheuristic algorithm, a numerical method, an exact method, or a local search procedure. These are the so called hybrid approaches.

The purpose of combining algorithms can be in several directions. The combination can be aimed at avoiding local optima or falling into a region of infeasible solutions. In this case, a combination with a local search procedure is usually used. Another goal may be to prevent early stagnation of the algorithm and find better solutions. The combination of methods, especially if it is applied to each iteration, leads to an increase in the time to run the iteration. Combining methods can lead to finding good solutions at an earlier stage, with fewer iterations, and in turn reduce the time to solve the problem.

In this paper we propose several variants of local search procedures, which are specifically tailored to the workforce planning problem. Our aim is decrease the number of infeasible solutions and thus to increase the diversification.

Local search procedures generate one or more solutions to the problem based on a current solution. These solutions are called neighborhood solutions. If the neighboring solutions thus generated are feasible, then we compare the best among the feasible neighboring solutions with the current solution. If the neighboring solution is better than the current one, then we replace the current solution of the problem with the neighboring one.

As noted, the local search procedure increases the execution time of a single iteration. If it is not efficient enough, it could also increase the execution time of the algorithm, the time to find good solutions. We apply the local search procedure only on the infeasible solutions. Our goal is to increase the number of feasible solutions and thus increase the choice. This, in turn, could lead to finding good solutions at an early stage of algorithm execution, which would reduce the time to solve the problem.

The main thing in the local search procedures that we offer is the removal of some of the appointed workers and the appointment of new ones in their place. After removing part of the workers, we get a partial solution, which is supplemented by assigning new workers by the use of ACO algorithm. The algorithm is stochastic, thus with a high probability, the new solution will be different from the previous one. From our previous research [7], we have found that it is best to remove half of the assigned workers.

We have compared three variants of the local search procedure:

- The workers to be removed are randomly selected. The procedure is applied once, regardless of whether the new solution is feasible or not [7];
- The workers to be removed are randomly selected. The procedure is repeated until a valid solution is constructed [8];
- The most expensive workers are removed. The procedure is applied once, regardless of whether the new solution is feasible or not.

The workforce planning problem is very complex with tight constraints. Because of this, it happens that there are iterations in which no ant succeeds in finding a feasible solution. We observe that after applying any of the listed procedures for local search, the number of infeasible solutions in subsequent iterations is greatly reduced. So the local search procedure is mainly applied to the first iterations. In subsequent iterations, it is less and less necessary to apply it. Due to the application of the local search procedure only on the infeasible solutions and reducing the need to apply it on subsequent iterations, it does not significantly increase the execution time of the algorithm.

V. Computational Results and Discussion

In this section are shown and compared the test results of application of proposed hybridization. The proposed local search procedures, combined with the ACO algorithm are tested on 10 structured and 10 unstructured problems. In our previous work [7] we research on the impact of the number of the removed workers from the solution. We tested with removing a quarter of the assigned workers, removing half of the assigned workers and removing all of the assigned workers (full restart). We found that the best results are achieved when removing half of the assigned workers. So in this work, when we apply any of the proposed local search procedures , we remove half of the assigned workers and complete the solution applying ACO algorithm.

The software, which realizes the algorithm is written in C programming language and is run on Pentium desktop computer at 2.8 GHz with 4 GB of memory. The proposed hybridizations are tested on artificially generated problem instances from [1].

The set of test problems consist of 10 Structured problems, enumerated from S1 to S10 and 10 Unstructured problems, enumerated respectively from U1 to U10. A problem is structured, when parameter d_j is proportional to the parameter h_{min} and it is unstructured when d_j is not proportional to the parameter minimal working time h_{min}. In our previous work [6] is shown that our ACO algorithm without hybridization outperforms Genetic algorithm and Scatter search from [1]. The stopping criteria is achieving the best found solution for the same test instance from [7], [8]. We apply same parameter settings for all variants of hybridization of ACO algorithm and they are fixed after several experiments.

The process of searching and constructing solutions in solving the workforce planning problem is very complex because of the strict constraints. The aim of the application of local search procedure is as many infeasible solutions of the problem, from the current iteration, become feasible, as well as to reduce the number of infeasible solutions found by the traditional ACO algorithm in the next iterations. This increases the chance that the underlying algorithm will find better solutions, as well as reduces the number of iterations needed to find those solutions. The proposed local search procedures do not spend much computational time because they are applied only over the infeasible solutions. Moreover, there is a sharp reduction in the number of iterations required to find these solutions.

We perform 30 independent runs with every of the test problems, because the algorithm is stochastic and to guarantee the robustness of the average results. We apply ANOVA

TABLE I: Calculation time in seconds

test instance	random remove	many times remove	maximal remove
S1	4.01 s	**3.75 s**	**3.92 s**
S2	19.97 s	**4.48 s**	**4.52 s**
S3	32.96 s	17.57 s	**7.75 s**
S4	37.22 s	46.64 s	**14.50 s**
S5	3.78 s	3.79 s	**2.40 s**
S6	5.12 s	**4.26** s	7.07 s
S7	31.23 s	36.18 s	**10.50 s**
S8	31.35 s	28.98 s	**16.88 s**
S9	**19.17** s	22.28 s	21.56 s
S10	10.19 s	15.78 s	**4.23 s**
U1	**5.25 s**	13.296 s	7.98 s
U2	2.07 s	**1.76 s**	4.15 s
U3	**4.88 s**	**4.86 s**	7.89 s
U4	3.11 s	**2.53 s**	18.84 s
U5	7.98 s	**3.22 s**	4.53 s
U6	**6.74 s**	11.22 s	14.24 s
U7	20.30 s	22.29 s	**11.11 s**
U8	4.17 s	4.12 s	**3.48 s**
U9	18.68 s	**12.98 s**	17.37 s
U10	**5.64 s**	6.224 s	9.95 s

test for statistical analysis to guarantee the significance of the difference between the average results. We compare the calculation time to find the best solution for every of the 20 tests.

Table I shows the needed calculation time to find best solution. The first column is the name of the test. The second column shows the needed time to find best solution, when we remove from infeasible solutions randomly chosen half of the workers, no matter if the new solution is feasible. The third column shows the needed time to find best solution, when we apply random remove of the half of the workers till the solution become feasible. The fourth column shows the needed time to find best solution when we remove from infeasible solution half for the workers, which are most expensive, no matter if the new solution is feasible. With the bold is shortest time to find best solution. Comparing structured problems, we observe that the hybrid ACO algorithm with local search procedure removing half for the workers, which are most expensive, needs less time to achieve best solution, eight of the ten cases. Regarding unstructured problems result is different. Hybrid ACO algorithm with local search procedure removing half fo the workers in a random way and applied one time, achieves best solution for a least time four times, when the local search procedure is applied many times till achieving feasible solution the least time is five times and when the local search procedure removes the most expensive workers, algorithm achieves the least time only two times. We observe big difference in hybrid algorithms performance when they are applied on structured and on unstructured problems. We can confirm that for structured problems is better to apply hybrid ACO algorithm with local search removing most expensive workers and the local search procedure can be applied only ones, no matter if the new solution is feasible. For unstructured problems it seems better to apply many times local search procedure till the solution becomes feasible. The difference comes from the fact that in unstructured problems it is more difficult to reach feasible solutions.

VI. Conclusion

In this paper we apply hybrid ACO algorithms to solve workforce planning problem. The traditional ACO algorithm is combined with several local search procedures. The local search procedures remove half of the assigned workers. Two of the procedures chose the removed workers in a random way and the third removes the most expensive workers and try to assign more cheapest. All local search procedures are applied only on infeasible solutions. The proposed hybrid algorithms are tested on 10 structured and 10 unstructured test instances. We observe that for structured instances, best performance has the local search procedure, which removes most expensive workers.

Acknowledgment

The work is supported by National Scientific Fund of Bulgaria under the grant DFNI KP-06-N52/5. and the rant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program and co-financed by the European Union through the European structural and Investment funds, and by the Polish-Bulgarian collaborative grant “Practical aspects for scientific computing”.

References

[1] Alba E., Luque G., Luna F., *Parallel Metaheuristics for Workforce Planning*, J. Mathematical Modelling and Algorithms, Vol. 6(3), Springer, 2007, 509-528.

[2] Bonabeau E., Dorigo M. and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York,Oxford University Press, 1999.

[3] Dorigo M, Stutzle T., *Ant Colony Optimization*, MIT Press, 2004.

[4] Easton F., *Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand*, Production and Operational Management 23(4), 2014, 660–675.

[5] Fidanova S., Roeva O., Paprzycki M., Gepner P., *InterCriteria Analysis of ACO Start Startegies*, Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, 2016, 547-550.

[6] Fidanova S., Luquq G., Roeva O., Paprzycki M., Gepner P., *Ant Colony Optimization Algorithm for Workforce Planning*, FedCSIS’2017, IEEE Xplorer, IEEE catalog number CFP1585N-ART, 2017, 415-419.

[7] Roeva O., Fidanova S., Luque G., Paprzycki M., Gepner P., *Hybrid Ant Colony Optimization Algorithm for Workforce Planning*, Annals of Computer Science and Information Systems, Vol. 15, 2018, pp. 233-236. ISSN: 2300-5963, DOI: http://dx.doi.org/10.15439/2018F47.

[8] Fidanova S., Luque G., New Local Search Procedure for Workforce Planning Problem, CYBERNETICS AND INFORMATION TECHNOLOGIES, Vol. 206, 2020, 40-48, DOI: 10.2478/cait-2020-0059

[9] 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:MAEB’04, 2004, 636–641.

[10] Grzybowska K., Kovács, G., *Sustainable Supply Chain - Supporting Tools*, Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Vol. 2, 2014, 1321–1329.

[11] Soukour A., Devendeville L., Lucet C., Moukrim A., *A Memetic algorithm for staff scheduling problem in airport security service*, Expert Systems with Applications, Vol. 40(18), 2013, 7504–7512.