

Hybrid Ant Colony Optimization Algorithm for Workforce Planning

Stefka Fidanova
IICT, BAS
Sofia, Bulgaria
E-mail: stefka@parallel.bas.bg

Gabriel Luque
DLCS University of Mlaga
29071 Mlaga, Spain
E-mail: gabriel@lcc.uma.es

Olympia Roeva
IBPhBME, BAS
Sofia, Bulgaria
E-mail: olympia@biomed.bas.bg

Marcin Paprzycki
SRI, PAS
Warsaw, Poland
E-mail: marcin.paprzycki@ibspan.waw.pl

Pawel Gepner
Intel Corporation
Swindon, UK
E-mail: pawel.gepner@intel.com

Abstract—Every organization and factory optimize their production process with a help of workforce planing. The aim is minimization of the assignment costs of the workers, who will do the jobs. The problem is very complex and needs exponential number of calculations, therefore special algorithms are developed to be solved. The problem is to select employers and to assign them to the jobs to be performed. This problem has very strong constraints and it is difficult to find feasible solutions. The objective is to fulfil the requirements and to minimize the assignment cost. We propose a hybrid Ant Colony Optimization (ACO) algorithm to solve the workforce problem, which is a combination between ACO and an appropriate local search procedure.

Keywords: Workforce Planning, Ant Colony Optimization, Metaheuristics, Local search

I. 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 [11], [17]. Workforce planing models are reformulated as mixed integer programming in [11]. The authors show that the mixed integer program is much easier to solve than the non-linear program. In [17] the model includes workers differences and the possibility of workers training and upgrading. In [4] and [18] a variant with random demands of the problem is proposed. Two stage program of scheduling and allocating with random demands is considered in [4]. Other variant of the problem is to include

uncertainty [12], [14], [16], [23], [24]. Most of the authors simplify the problem by omitting some of the constraints. In [6] a mixed linear programming is applied and in [18] 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 [2], [15], [19], [21], [22]. In considered here problem some heuristic method including genetic algorithm [1], [13], memetic algorithm [20], scatter search [1] etc., are applied.

So far the Ant Colony Optimization (ACO) algorithm is proved to be very effective solving various complex optimization problems [7], [10]. In our previous work [8] 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 [8]. 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.

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

II. DEFINITION OF THE WORKFORCE PLANNING PROBLEM

In this paper we solve the workforce planning problem proposed in [1] and [9]. The set of jobs $J = \{1, \dots, m\}$ must be completed during a fixed period of time. The job j requires d_j hours to be completed. $I = \{1, \dots, n\}$ 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. 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:

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

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.

$$f(x) = \text{Min} \sum_{i \in I} \sum_{j \in A_i} \tilde{c}_{ij} \cdot x_{ij} \quad (1)$$

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).

III. 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.

A lot of problems coming from real life and industry needs exponential number of calculations. Therefore the only option is to be applied some metaheuristics. The goal is to find a good solution for a reasonable time [5].

A. ACO Algorithm for Workforce Planning

In this section we will apply the ACO algorithm for workforce planing from our previous work [8], 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:

$$\eta_{ijl} = \begin{cases} l/c_{ij} & l = z_{ij} \\ 0 & otherwise \end{cases} \quad (2)$$

When there are several candidate nodes with a same probability, the next node is chosen between them in a random way. 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 deposited a new pheromone on the elements of their solutions. The new added pheromone is equal to the reciprocal value of the objective function.

$$\Delta\tau_{i,j} = \frac{\rho - 1}{f(x)} \quad (3)$$

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.

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

B. 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. 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.

TABLE I: Test instances characteristics

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

TABLE II: ACO parameter settings

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

IV. COMPUTATIONAL RESULTS

In this section test results are reported and compared with ACO algorithm without local search procedure. We use the artificially generated problem instances considered in [1]. The test instances characteristics are shown in Table I.

The set of test problems consists of 20 used in [1], [8]. In our previous work [8] we show that our ACO algorithm outperforms the genetic and scatter search algorithms from [1]. The number of iterations is fixed to be maximum 100. In Table II the parameter settings of our ACO algorithm are shown. The values are fixed experimentally.

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 calculation 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. We are interested of the number of iterations for finding the best result. It can be very different for different test

TABLE III: Hybrid ACO ranking

	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

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 III 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.

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. The Table III 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 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 calculation time to remove and

TABLE IV: Hybrid ACO comparison according calculation 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	82.244 s	93.98 s	79.63 s	103.012 s

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 calculation 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 III, but taking in to account the calculation time instead number of iterations. Regarding the Table IV the ranking according the time is similar to the ranking according to the number of iterations from the Table III. 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 calculation 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 IV 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.

V. 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.

ACKNOWLEDGMENT

Work presented here is partially supp, 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] Albayrak G., zdemir ., *A state of art review on metaheuristic methods in time-cost trade-off problems*, International Journal of Structural and Civil Engineering Research, Vol. 6(1), 2017, 30-34.
- [3] Bonabeau E., Dorigo M. and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York, Oxford University Press, 1999.
- [4] Campbell G., *A two-stage stochastic program for scheduling and allocating cross-trained workers*, J. Operational Research Society 62(6), 2011, 10381047.
- [5] Dorigo M., Stutzle T., *Ant Colony Optimization*, MIT Press, 2004.
- [6] Easton F., *Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand*, Production and Operational Management 23(4), 2014, 660675.
- [7] Fidanova S., Roeva O., Paprzycki M., Gepner P., *InterCriteria Analysis of ACO Start Strategies*, Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, 2016, 547-550.
- [8] Fidanova S., Luqu 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.
- [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:MAEB04, 2004, 636641.
- [10] Grzybowska K., Kovcs, G., *Sustainable Supply Chain - Supporting Tools*, Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Vol. 2, 2014, 13211329.
- [11] Hewitt M., Chacosky A., Grasman S., Thomas B., *Integer programming techniques for solving non-linear workforce planning models with learning*, European J of Operational Research 242(3),2015, 942950.
- [12] Hu K., Zhang X., Gen M., Jo J., *A new model for single machine scheduling with uncertain processing time*, J Intelligent Manufacturing, Vol 28(3), Springer, 2015, 717-725.
- [13] Li G., Jiang H., He T., *A genetic algorithm-based decomposition approach to solve an integrated equipment-workforce-service planning problem*, Omega, Vol. 50, Elsevier, 2015, 117.
- [14] Li R., Liu G., *An uncertain goal programming model for machine scheduling problem*. J. Intelligent Manufacturing, Vol. 28(3), Springer, 2014, 689-694.
- [15] Mucherino A., Fidanova S., Ganzha M., *Introducing the environment in ant colony optimization*, Recent Advances in Computational Optimization, Studies in Computational Intelligence, Vol. 655, 2016, 147158.
- [16] Ning Y., Liu J., Yan L., *Uncertain aggregate production planning*, Soft Computing, Vol. 17(4), Springer, 2013, 617624.
- [17] Othman M., Bhuiyan N., Gouw G., *Integrating workers' differences into workforce planning*, Computers and Industrial Engineering, Vol. 63(4), 2012, 10961106.
- [18] Parisio A, Jones CN., *A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand*, Omega, Vol. 53, Elsevier, 2015, 97-103.
- [19] Roeva O., Atanassova V., *Cuckoo search algorithm for model parameter identification*, Int. J. Bioautomation, Vol. 20(4), 2016, 483492.
- [20] 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, 75047512.
- [21] Tilahun S.L., Ngnotchouye J.M.T., *Firefly algorithm for discrete optimization problems: A survey*, Journal of Civil Engineering, Vol. 21(2), 2017, 535545.
- [22] Toimil D., Gmes A., *Review of metaheuristics applied to heat exchanger network design*, International Transactions in Operational Research, Vol. 24(1-2), 2017, 726.
- [23] Yang G., Tang W., Zhao R., *An uncertain workforce planning problem with job satisfaction*, Int. J. Machine Learning and Cybernetics, Springer, 2016. doi:10.1007/s13042-016-0539-6 <http://rd.springer.com/article/10.1007/s13042-016-0539-6>
- [24] Zhou C., Tang W., Zhao R., *An uncertain search model for recruitment problem with enterprise performance*, J Intelligent Manufacturing, Vol. 28(3), Springer, 2014, 295-704. doi:10.1007/s10845-014-0997-1