BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 6
Special Issue on New Developments in Scalable Computing
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0059

New Local Search Procedure for Workforce Planning Problem

Stefka Fidanova[1], *Gabriel Luque*[2]

[1]*IICT-BAS, Acad. G. bonchev str. Bl 25A, 1113 Sofia, Bulgaria*
[2]*DLCS University of Malaga, 29071 Malaga, Spain*
E-mails: stefka@parallel.bas.bg gabriel@lcc.uma.es

Abstract**:** *Optimization of workforce planning is important for any production area. This leads to an improvement in production process. The aim is minimization of the assignment costs of the workers, who will do the jobs. The problem is to select and assign employers to the jobs to be performed. The constraints are very strong, coming both from the specifics of the production process and from the legislation. Sometimes it is difficult to find feasible solutions. The complexity of the problem is very high and the needed number of calculations is exponential, therefore only specially developed algorithms can be applied. The objective is to minimize the assignment cost, while respecting all requirements. We propose a new hybrid metaheuristic algorithm to solve the workforce-planning problem, which is a combination between Ant Colony Optimization* (*ACO*) *and suitable local search procedure.*

Keywords**:** *Metaheuristics, ant colony optimization, local search, workforce planning, combinatorial optimization.*

1. Introduction

The workforce planning is a tricky issue for human resource. It is one of the most important decision making problem, common for all branches of industry. Workforce planning is the process of determining the skills and human capital needed to accomplish a task. The problem is NP-hard. This problem consists of two main decisions: selection and assignment. The first decision determines employees from available workers. The second decision defines which worker which job will perform. The aim is to minimize assignment cost, while all requirements are fulfilled. The workforce planning is a part of the human resource management. Workforce planning is a systematic, fully integrated organizational process that involves proactively planning ahead. The traditional numerical methods as well as the exact methods have not a capacity to solve this hard problem for instances with realistic size. These kinds of methods can be applied only on some simplified variants of the problem.

A deterministic workforce planning problem is studied in [11, 16]. Workforce planning 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 [16] the model includes differences between the workers and the opportunity of workers training and improvement. In [13, 17] a variant with random demands of the problem is proposed. Two-stage program with random demands of scheduling and allocating is considered in [3]. Other authors include uncertainty, [12, 14, 15, 22, 23]. Statistical approaches for similar problems are applied in [19-21]. Most of the authors simplify the problem by omitting some of the constraints. In [5] mixed linear programming is applied and in [17] decomposition method is applied, but the convex methods are not applicable for the more complex non-linear workforce planning problems. In this case, some heuristic method including genetic algorithm [1], memetic algorithm in [18], and scatter search in [1] is applied. So far, the Ant Colony Optimization (ACO) Algorithm is proved to be very effective solving various complex optimization problems [6, 10]. ACO Algorithm for workforce planning is developed in our previous work [7]. There we consider the workforce planning problem variant proposed in [1]. Current paper is the upgrade of [8]. There we proposed a variant of hybrid ACO Algorithm. In the current paper we improve the local search procedure. The aim is to improve the algorithm performance.

2. Workforce planning problem

We propose a new hybrid ACO Algorithm for solving Workforce Planning Problem (WPP), proposed in [1, 9]. For a fixed-time period a set of jobs $J = \{1, \ldots, m\}$ have to be completed. The completion time for job j is d_j hours. The set of candidate for assignment workers is $I = \{1, \ldots, n\}$, and $h_{\min}$ hours is the minimum time, a worker need to perform any of assigned work to work efficiently. Availability of the worker i is s_i hours. A worker is assigned maximum to $j_{\max}$ jobs. A_i is the set of jobs, worker i is qualified. The maximum number of assigned workers fixed period is t, or at most t workers may be assigned from the set I of workers. The selected workers need to have capability to execute all the jobs. c_{ij} is the assignment cost of the worker i to the job j. The objective is to minimize the sum of the assignment cost of all assigned workers. The aim is to find feasible solution, which minimizes the objective function.

The mathematical model of WPP can be described in a following way:

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

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

z_{ij} = number of hours that worker i is assigned to perform job j;

Q_j = set of workers qualified to perform job j,

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

subject to

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

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

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

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

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

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

$$y_i \in \{0,1\}, \quad i \in I,$$

$$z_{ij} \ge 0, \quad i \in I, j \in A_i.$$

The objective function is the total assignment cost and has to be minimized. The number of hours for each selected worker is limited (Inequality (2)). All jobs need to be finished (Inequality (3)). The number of jobs, assigned to a single worker is limited (Inequality (4)). There is minimal number of hours, which every job needs to be performed by every assigned worker to be able to work in efficient way (Inequality (5)). The number of assigned workers is limited (Inequality (6)).

A various objective functions can be used in different models. The aim of this paper is minimization of total assignment cost. If $c1_{ij}$ is the cost the worker i to performs the job j for one time units (for example one hour), then the objective can be to minimize the cost of the all jobs to be finished:

(7) $$f(x) = \min \sum_{i \in I} \sum_{j \in A_i} \mathrm{c1}_{ij} x_{ij} z_{ij}.$$

The workers can have preferences to the jobs they are qualified to perform. In this case, the objective can be to maximize the satisfaction of their preferences. The WPP is difficult to be solved, because of very restrictive constraints, especially the relation between the parameters $h_{\min}$ and d_j. For structured problems (d_j is a multiple of $h_{\min}$) is more easier to find feasible solutions, than for unstructured problems (d_j and $h_{\min}$ are not related).

3. New hybrid ACO Algorithm

ACO is a metaheuristic method, which follows real ants behaviour when looking for a food and returns back to the nest. Ants in the nature use chemical substance called pheromone, to mark their path. Without pheromone, an ant moves in randomly and when it detects a pheromone, it makes a decision whether to follow it and strengthen it with a new added pheromone. Thus, the used trail becomes more attractive. With the help of their swarm intelligence, the ants can find a shorter path between the food source and their nest.

3.1. Main ACO Algorithm

Most of the real life and industrial problems require exponential number of calculations. It is impractical to apply on them exact algorithm or traditional numerical methods, for a large problems. One option is some metaheuristics to be applied. The idea is to find a good solution for a reasonable time [4].

First Eric Bonabeau, Dorigo and Theraula [2] realise the idea to use ant behaviour to solve optimization problem. Later various modifications and improvements are proposed, mainly in pheromone updating rules [4]. One of the main point is graph representation of the problem. Thus, the solutions are paths in a graph. We look for a shorter path, which fulfil problem constraints. ACO Algorithm has following requirements:

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

The transition probability p_{ij}, which is used to make decision about the new included node j, when the current node is i, is a product of the heuristic information η_{ij} and the level of the pheromone τ_{ij} related to this node, where $i, j = 1, \ldots, n$,

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

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

When the value of the heuristic information and/or the value of the related pheromone is higher, then the probability this node to be chosen becomes higher and it is more profitable. The initial value of the pheromone is the same for all elements of the graph and is set to a small positive value τ_0, $0 < \tau_0 < 1$. The algorithm is iterative. The ants update the pheromone values of used elements of the graph, at the end of the iteration. Different ACO Algorithms apply different procedures to update the pheromone values [4].

The main pheromone trail update rule is

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

where, ρ imitates the evaporation in the nature and decrease the value of the old pheromone. The new added pheromone $\Delta\tau_{ij}$ depends to the quality of the founded solution. The quality of the solution is measured by the corresponding value of the objective function.

Method of diversifying the search in a search space is random starting node for every ant and every iteration. Random start helps to use small ant population (up to 20 ants), comparing with other population based metaheuristics. The heuristic information represents the current knowledge of the problem, which is used to better manage the ants. The pheromone reflects the global experience of the ants in search of the optimal solution. The pheromone is a mechanism for concentration of the search around the best found so far solution.

3.2. ACO Algorithm for workforce planning problem

This section describes the ACO Algorithm for WPP from our previous work [7]. In [7] is proposed the main algorithm without local search procedure. Essential moment in ACO is the representation of the problem by a graph in a proper way. We represent the problem with 3D graph. The node (i, j, z) corresponds to the worker with number i assigned to the job j for time Z. The graph is asymmetric, because the maximal value of Z depends of the value of j and different jobs need different time to be completed. An ant starts to construct its solution from random node of the graph. Three random numbers are generated for every ant. The first random number corresponds to the assigned worker and is in the interval $[0, \ldots, n]$. The second random number corresponds to the job, this worker will perform and is in the interval $[0, \ldots, m]$. When the worker is not qualified to perform this job, we chose in a random way a new job. The third number corresponds to the number of hours the worker i is assigned to perform the job j and is in the interval $[h_{\min}, \min\{d_j, s_i\}]$. After, ant includes a next node in a partial solution, applying transition probability rule. An ant repeats this procedure until the solution is completed, or there is not a possibility to include a new node. Proposed heuristic information is as follows, for worker i to perform the job j for time l:

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

With this heuristic information, the more attractive is to assign the cheapest worker for a longer time. The node with the highest probability is included as a next node in the solution. In the case there are more than one candidate nodes with the same probability, the next node is chosen between them in a random way. The problem constraints need to be taken in to account, when a new node is accepted. The constraints are:

- Number of workers assigned till now;
- Time intervals every worker is assigned till now;
- Number of assigned time intervals per job until now.

If some of the problem constraints is not satisfied, then corresponding value of the transition probability function is set to 0. When for all possible nodes the value of the probability function is 0 the solution construction stops, because it is impossible to include new node in the current partial solution. If the constructed solution is feasible, we calculate the value of the objective function as a sum of the assignment cost of the assigned workers. For not feasible solution, the value of the objective function is set to be –1. The elements of feasible solutions update its pheromone. The new added pheromone is equal to the reciprocal value of the objective function,

$$\Delta\tau_{ij} = \frac{\rho - 1}{f(x)}. \tag{11}$$

Thus, the elements of the graph belonging to the solutions with less value of the objective function, will receive more pheromone and will be more attractive in the next iteration. At the end of the iteration we update the global best so far solution. If the iteration best solution is better than the current global best solution, it becomes the new global best solution.

3.3. New local search procedure

Very often the stochastic algorithms are combined with a local search procedure. The aim is to avoid trapping of the algorithm in local optimums and thus to reduce the time to find the best solutions and eventually to improve the achieved solutions. The local search procedure is applied only on infeasible solutions. We apply it until the solution becomes feasible. It is the main difference with our previous work [8]. We observe that the infeasible solutions after first iteration have reduced a lot. Thus the procedure is applied mainly at the beginning of the algorithm. The constraints of the WPP are very restrictive and part of the ants does not succeed to find feasible solution only applying ACO. With the new proposed local search procedure the ants find feasible solutions and the expectation to improve current solution increases.

When the solution is infeasible, we remove half of the assigned workers. The workers are chosen to be removed in a random way. We consider this solution as a partial solution of the problem. After that we assign new workers and thus we construct new solution from the partial one, applying the ACO Algorithm rules on this partial solution. The algorithm is stochastic, thus with a high probability, the new solution will be different from the previous one. We apply local search procedure until the solution becomes feasible.

4. Experimental results

The proposed Hybrid ACO Algorithm is tested on 10 structured and 10 unstructured test problems. The achieved results are compared with hybrid ACO Algorithm from our previous work [8]. The ACO Algorithm from [8] applies local search procedure on infeasible solutions only ones, no matter whether the achieved solution is feasible or not. In [8] we investigate the influence of the number of the removed workers from the solution, when the local search procedure is applied. We found that the best algorithm performance is when half of the workers are removed.

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. The algorithm is tested on 20 artificially generated problem instances used in [1, 7, 8]. The test problem instances parameters are shown in Table 1.

Table 1. Test problems characteristics

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

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. The parameter d_j is proportional to the parameter $h_{\min}$ for structured problems and for unstructured problem it is not proportional. It is shown in our previous work shown that the ACO Algorithm, without local search procedure, outperforms the genetic and scatter search algorithms from [1]. The stopping criteria is founding the best found solution from [7, 8] for the same test problems. Table 2 shows the ACO Algorithm parameters. They are the same as in [7, 8] and are found experimentally.

Table 2. ACO Algorithm parameters

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

The constraints of WPP are very restrictive. Only several ants find feasible solutions. There are iterations without any valid solution. This makes the search process complicated. The aim of our local search procedure is infeasible solutions to become feasible. So the ability of ants to find a good solution increases and the needed number of iterations and computational time respectively decreases. Our local search procedure is economical, because we apply it only on infeasible solutions. Moreover, we have noticed that the number of infeasible solutions decreases sharply on subsequent iterations. It is another reason for not increasing the calculation time.

Table 3. Time in seconds to find best solution

Best solution	One local search	Many times local search
S1	4.012 s	**3.752 s**
S2	19.970 s	**4.485** s
S3	32.960 s	**17.572** s
S4	**37.229** s	46.64 s
S5	**3.787** s	**3.796 s**
S6	5.128 s	**4.628 s**
S7	**31.231** s	36.181 s
S8	31.352 s	**28.984** s
S9	**19.176** s	22.280 s
S10	**10.194** s	15.788 s
U1	**5.252 s**	13.296 s
U2	2.072 s	**1.760 s**
U3	**4.880 s**	**4.860 s**
U4	3.112 s	**2.536 s**
U5	7.980 s	**3.228 s**
U6	**6.744 s**	11.228 s
U7	**20.304 s**	22.296 s
U8	**4.172 s**	**4.124 s**
U9	18.688 **s**	**12.984 s**
U10	**5.640 s**	**6.224 s**

When we apply local search procedure we remove the half of the assigned by ACO workers, because from our experience [8] it is the best number of removing workers. We perform 30 independent runs with every test problem, because the algorithm is stochastic and to guarantee the robustness of the average results. We are interested of the calculation time to find the best solution.

Table 3 shows the average calculation time of 20 test problems. The second column reports the execution time, when local search is applied only one time on infeasible solutions. The third column reports the execution time, when the local search is applied till the solution becomes feasible. We observe that in nine of the tests the new hybrid ACO Algorithm is faster than the other hybrid ACO Algorithm. In three of the tests the time is statistically the same for both algorithms and for eight of the tests the new hybrid ACO Algorithm is slower. Analyzing the results we can conclude that the new hybrid ACO Algorithm is slightly faster.

5. Conclusion

We propose a new hybrid ACO Algorithm for solving WPP. The main ACO Algorithm is combined with suitable local search procedure. The local search procedure is applied only on infeasible solutions until they become feasible. The main idea is to remove half of the assigned workers in a random way and to assign new workers on their place. The proposed local search procedure is compared with the hybrid ACO Algorithm from [8]. We show that the new hybrid ACO Algorithm performs faster. As a future work we will try with other hybridisation and other improvements of the algorithm performance.

***Acknowledgements*:** The work presented here is partially supported by the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program and co-financed by European Union through the European structural and Investment funds, and by the National Scientific Fund of Bulgaria under the Grant DFNI DN 02/10.

References

1. Alba, E., G. Luque, F. Luna. Parallel Metaheuristics for Workforce Planning. – J. Mathematical Modelling and Algorithms, Vol. **6**, Springer, 2007, No 3, pp. 509-528.
2. Bonabeau, E., M. Dorigo, G. Theraula. Swarm Intelligence: From Natural to Artificial Systems. New York, Oxford University Press, 1999.
3. Campbell, G. A Two-Stage Stochastic Program for Scheduling and Allocating Cross-Trained Workers. – J. Operational Research Society, Vol. **62**, 2011, No 6, pp. 1038-1047.
4. Dorigo, M., T. T. Stutzle. Ant Colony Optimization. MIT Press, 2004.
5. Easton, F. Service Completion Estimates for Cross-Trained Workforce Schedules under Uncertain Attendance and Demand. – Production and Operational Management, Vol. **23**, 2014, No 4, pp. 660-675.
6. Fidanova, S., O. Roeva, M. Paprzycki., P. Gepner. InterCriteria Analysis of ACO Start Startegies. – In: Proc. of 2016 Federated Conference on Computer Science and Information Systems, 2016, pp. 547-550.
7. Fidanova, S., G. Luquq, O. Roeva, M. Paprzycki, P. Gepner. Ant Colony Opimization Algorithm for Workforce Planning. FedCSIS'2017, IEEE Xplorer, IEEE Catalog Number CFP1585N-ART, 2017, pp. 415-419.
8. Fidanova S., G. Luque, O. Roeva, M. Paprzycki, P. Gepner. 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.

9. Glover, F., G. Kochenberger, M. Laguna, T. Wubbena. Selection and Assignment of a Skilled Workforce to Meet Job Requirements in a Fixed Planning Period. – In: Proc. of MAEB'04, 2004, pp. 636-641.

10. Grzybowska, K., G. Kovács. Sustainable Supply Chain-Supporting Tools. – In: Proc. of 2014 Federated Conference on Computer Science and Information Systems, Vol. **2**, 2014, pp. 1321-1329.

11. Hewitt, M., A. Chacosky, S. Grasman, B. B. Thomas. Integer Programming Techniques for Solving Non-Linear Workforce Planning Models with Learning. – European J. of Operational Research, Vol. **242**, 2015, No 3, 942-950.

12. Hu, K., X. Zhang, M. Gen, J. Jo. A New Model for Single Machine Scheduling with Uncertain Processing Time. – J. Intelligent Manufacturing, Vol. **28**, Springer, 2015, No 3, pp. 717-725.

13. Li, G., H. Jiang, T. T. He. A Genetic Algorithm-Based Decomposition Approach to Solve An Integrated Equipment-Workforce-Service Planning Problem. – Omega, Vol. **50**, Elsevier, 2015, pp. 1-17.

14. Li, R., G. Liu. An Uncertain Goal Programming Model for Machine Scheduling Problem. – J. Inteligent Manufacturing, Vol. **28**, 2014, No 3, Springer, pp. 689-694.

15. Ning, Y., J. Liu, L. Yan. Uncertain Aggregate Production Planning. – Soft Computing, Vol. **17**, Springer, 2013, No 4, pp. 617-624.

16. Othman, M., N. Bhuiyan, G. Gouw. Integrating Workers' Differences into Workforce Planning. – Computers and Industrial Engineering, Vol. **63**, 2012, No 4, pp. 1096-1106.

17. Parisio, A., C. N. Jones. A Two-Stage Stochastic Programming Approach to Employee Scheduling in Retail Outlets with Uncertain Demand. – Omega, Vol. **53**, Elsevier, 2015, pp. 97-103.

18. Soukour, A., L. Devendeville, C. Lucet, A. Moukrim. A Memetic Algorithm for Staff Scheduling Problem in Airport Security Service. – Expert Systems with Applications, Vol. **40**, 2013, No 18, pp. 7504-7512.

19. Todorov, V., I. Dimov. Monte Carlo Methods for Multidimensional Integration for European Option Pricing. – In: Proc. of AIP Conference 1773, 2016. DOI: 10.1063/1.4965003.

20. Todorov, V., N. Ikonomov, S. Apostolov, I. Dimov, R. Georgieva, Y. Dimitrov. An Improved "Walk on Equations" Monte Carlo Algorithm for Linear Algebric Systems, Recent Advances in Computational Optimization. – Studies in Computational Intelligence, Vol. **838**, Springer, 2020, pp. 215-236.

21. Todorov, V., I. Dimov. Efficient Stochastic Approach for Multidimensional Integrals in Bayesian Statistics, Large Scale Scientific Computing. – Lecture Notes in Computer Science, Vol. **11958**, Springer, 2020, pp. 454-462.

22. Yang, G., W. Tang, R. Zhao. An Uncertain Workforce Planning Problem with Job Satisfaction. – Int. J. Machine Learning and Cybernetics, Springer, 2016. DOI:10.1007/s13042-016-0539-6.

23. Zhou, C., W. Tang., R. Zhao. An Uncertain Search Model for Recruitment Problem with Enterprise Performance. – J. Intelligent Manufacturing, Vol. **28**, Springer, 2014, No 3, pp. 295-704. DOI:10.1007/s10845-014-0997-1.

Received: 20.08.2020; Second Version: 18.10.2020; Third version: 21.10.2020; Accepted: 23.10.2020