Index merge in application to multi-skill project scheduling

Dmitry Arkhipov[1], Olga Battaïa[2]

[1] V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation
miptrafter@gmail.com
[2] KEDGE Business School, Bordeaux, France
olga.battaia@kedgebs.com

Keywords: project scheduling, multi-skill resources, workforce scheduling.

1 Introduction

In this research the index merge method is proposed to decrease the number of variables of Multi-Skilled Resource Constraint Project Scheduling Problem (MSRCPSP). This problem was formulated in (Neron E. and Baptista B. 2002) It generalizes Resource-Constraint Project Scheduling Problem (RCPSP) which is proved to be NP-hard in the strong sense (Garey M. and Johnson D. 1975). In (De Bruecker P. *et. al.* 2015) the survey of approaches to solve this problem is presented. Integer and mixed-integer statements of MSRCPSP are compared in (Almeida B.F. *et. al.* 2019).

The problem can be formulated as follows. There is a set of tasks N, set of workers W and a set of specialities S. If the worker w_i has skill s_l than $h_{il} = 1$, otherwise $h_{il} = 0$. For each task $j \in N$ the following parameters are given: p_j – processing time and a_{jk} – required number of workers with speciality s_k. Precedence relation e_{ij} can be defined for a pair of tasks, means that the task i has to be completed before the task j starts. The objective is to process all tasks without preemptions in the shortest time. For easier understanding of presented method non-human resources are not considered in this extended abstract.

The very important difference between MSRCPSP and RCPSP: *it is necessary to assign workers not only to tasks but also the speciality for which this worker is responsible for.* The following short example describes it. There is a task j which needs one worker with speciality s_1 and one with speciality s_2, a set of workers $W = \{w_1, w_2, w_3\}$ and a set of specialities $S = \{s_1, s_2, s_3\}$. Worker w_1 has specialities s_1 and s_2, w_2 – s_3, w_3 – s_1. Suppose that there is a binary variable x_{ij} which equals 1 if the worker i is assigned to task j, otherwise $x_{ij} = 0$. Suppose that speciality constraints are modelled as resources, i.e.

$$\forall l = 1, 2 : \sum_{i=1}^{3} x_{1i} \cdot h_{il} \geq 1 \tag{1}$$

– there are enough workers for each speciality required for processing task j. The problem is that inequality (1) allows to assign workers w_1 and w_2 to the task j in feasible solution, which is not correct. In this case worker w_1 have to act as a specialist s_1 and s_2 simultaneously, which is not possible.

This means that the variable with three indices (task, worker, speciality) have to be introduced together with variables related to the start times of tasks. Such a large number of variables makes the problem very hard to solve.

2 Proposed approach

We propose a data-preprocessing method to decrease the number of variables by index merge. To solve MSRCPSP it is necessary to set task processing intervals and to assign workers to tasks. For each worker the speciality under which he operates the task has to be chosen. In the paper (Stadnicka D. *et. al.* 2017), the Hall's marriage theorem was used in integer linear programming (ILP) model to formalize operator assignments. In this work, we propose an approach based on the creation of task processing *scenarios*.

Let we call scenario z – the assignment list of workers $z = \{x_1^z, \ldots, x_{|W|}^z\},\ x^z = 1$. If scenario z is *correct* for task $j \in N$, then it is possible to assign the set of workers $W_z = \{w_i | x_i^z = 1\}$ to all specialities required for processing j. The set of all correct scenarios for task j is denoted by Z_j. To decrease the number of scenarios in Z_j, only those with the number of workers not exceeding the required number for task j are considered.

2.1 The correctness of task scenario

The problem to verify the correctness of scenario z for task j can be formulated as follows.

Problem 1. *There is a set of workers defined by scenario z and a set of specialists required for job j. Each worker can have several specialities. How to verify that there is an assignment of workers to required specialities, such that each worker is assigned to the speciality he has and for each speciality the required number of workers are assigned?*

Let S_j – the ordered set of specialities, which includes a_{jk} elements of speciality k. Note that $|S_j| = |W_z|$. Then, problem 1 means that the set of workers W_z can be paired with the set of S_j. In terms of graph theory this problem is equivalent to the *perfect matching problem in bipartite graph.*

Problem 2. *There is a bipartite graph with two disjoint sets of vertices related to W_j and S_j. Vertex $i \in W_j$ is connected to vertex $s_j^k \in S_j$ related to speciality s if and only if $h_{is} = 1$. Is there a perfect matching for this graph?*

If there is a perfect matching, then we can assign workers to specialities they are matched with. If there is no perfect matching, then the workers cannot be assigned to specialities and task j cannot be processed under scenario z. Verification of scenarios for the example, presented in the previous section is illustrated on the Fig. 1.

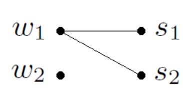
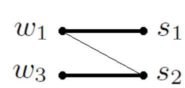
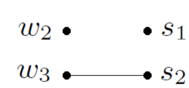
Scenarios for task j	{1, 1, 0}	{1, 0, 1}	{0, 1, 1}
Graph representation to find a perfect matching			
Is this scenario correct?	No	Yes	No

Fig. 1. Example: scenario verification.

Problem 2 can be solved by the Hopcroft–Karp algorithm (Hopcroft J. and Karp R. 1973) in $O(|W_z|^{5/2})$ operations. Therefore the complexity of the verification of the correctness of scenario z for task j is the same.

2.2 Creation of correct scenarios

To create the set of correct scenarios Z_j the following algorithm can be used.
Algotithm 1.

1. Calculate the number of workers $k_j = \sum_{s \in S} a_{js}$ required for processing task $j \in N$.
2. Generate all the scenarios of k_j workers.
3. Cycle all generated scenarios and check each scenario if it is correct for processing task j. If yes, add it to Z_j.

Number of scenarios to be verified – $C^{k_j}_{|W_z|} = \frac{|W_z|!}{k_j!(|W_z|-k_j)!}$ which is not more than $C^{\lceil |W_z|/2 \rceil}_{|W_z|}$. By the Stirling's formula this value can be asymptotically approximated by

$$C^{|W_z|}_{\lceil |W_z|/2 \rceil} \backsim \frac{2^{|W_z|+1/2}}{\sqrt{\pi |W_z|}}.$$

Then, subject to Hopcroft–Karp algorithm, the complexity of Algorithm 2 can be evaluated as $O(2^{|W_z|}|W_z|^2)$ operations. Creation of the correct scenarios for entire set of tasks N takes $O(n2^{|W|}|W|^2)$.

2.3 Using scenarios in MSRCPSP models

In case of Mixed-Integer Linear Programming models the variable with one index (task scenario) can be used instead of the variable with three indices (task, worker, speciality). In Constraint Programming models, interval variables associated with optional task scenarios can be used as follows.

Constraint programming MSRCPSP model.
Task processing optional interval variables: $\forall j \in N, z \in Z_j : \ int_z$ with size $|int_z| = p_j$.
Constraints:

- $\forall j \in N : \ \sum_{z \in Z_j} presenceOf(int_z)$ – for each task only one scenario is presented in the solution;
- Let $f_i(t)$ – cumulative function defined for all $i \in W$ by the number of intervals associated with scenarios $z \in Z$ which involves the worker i ($x^z_i = 1$). Then the number of tasks processed simultaneously be the worker i can be modelled by $f_i(t) \leq 1$.
- $\forall e_{ij} \in E, z_1 \in Z_i, z_2 \in Z_j : \ endOf(int_{z_1}) \leq startOf(int_{z_2})$ – precedence relations have to be satisfied.

Objective – minimal makespan:

$$\min \max_{z \in Z} endOf(int_z).$$

3 Numerical experiments & analysis

In numerical experiments, we compared the presented model with CP model based on the IBM ILOG example: /examples/opl/sched_sequence. Both models were implemented using IBM CP Optimizer 12.6.2 and tested on Intel(R) Core(TM) i7-7700 HQ 2.8 GHz with 8 Gb RAM. We generated 800 random instances with 10, 20, 30 and 40 tasks and different number of workers, precedences and skills. Time limit for instances with 10 and 20 jobs was equal to 120 seconds and for instances with 30 and 40 jobs – 600 seconds. For 100% of generated instances proposed method gave better results. The results are presented in Table 1.

Table 1. Numerical experiments result.

	Without scenarios		With scenarios	
Tasks	Solutions found %	Optimum proved %	Solutions found %	Optimum proved %
10	53	26	100	89
20	44	19	100	63
30	18	6	100	39
40	6	0	100	0

The presented approach can be applied for other models to merge the variable indices and decrease the number of variables. Method allows to decrease the number of variables by considering constraints on the pre-processing stage and works especially efficiently if the pre-processing eliminates a large number of indices combinations as it is shown by numerical experiments.

The proposed idea has the following weaknesses.

- A larger number of constraints. In comparison with classic models, all constraints involving index i have to be applied to all merged combinations of indices including i.
- The need to store scenarios and a large number of constraints leads to the large amount of memory required.
- It is necessary to develop fast pre-processing procedures to eliminate the forbidden combinations of indices.

4 Conclusion

In this paper, an index merge method was presented and applied to MSRCPSP. The efficiency of the proposed model was evaluated theoretically and by a comparative analysis with default IBM ILOG CP model.

Acknowledgements

This research was supported by the Russian Foundation for Basic Research (grant 18-37-00295 mol_a).

References

Almeida B.F., I. Correia and F. Saldanha-da-Gama, 2019, “Modeling frameworks for the multiskill resource-constrained project scheduling problem: a theoretical and empirical comparison", *Intl Trans in Oper Res.*, Vol. 26, I. 3, pp. 946-967.

De Bruecker P., J. Van den Bergh, J. Beliën and E. Demeulemeester, 2015, “Workforce planning incorporating skills: State of the art", *Eur J Oper Res*, Vol. 243, I. 1, pp. 1-16.

Garey M., D. Johnson, 1975, “Complexity results for multiprocessor scheduling under resource constraints", *SIAM Journal on Computing*, Vol. 4, I. 4, pp. 397-411.

Hopcroft J., R. Karp, 1973, “An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs", *SIAM Journal on Computing*, Vol. 2, I. 4, pp. 225-231.

Neron E., B. Baptista, 2002, “Heuristics for multi-skill project scheduling problem", *In:International Symposium on Combinatorial Optimization.*

Stadnicka D., D. Arkhipov, O. Battaïa and R.M. Chandima Ratnayake, 2017, “Skill management in the optimization of aircraft maintenance processes", *IFAC PapersOnLine*, Vol. 50, I. 1, pp. 6912-6917.