The Task of Compiling the Project Execution Plan in the Multi-agent Model

A. S. Zraenko

Ural Federal University, Yekaterinburg, Russian Federation,
zraenko@yandex.ru

Abstract. This article describes one solution for works execution planning in multi-agent system. The basic requirements for a technique of compiling works execution plan are identified. The analysis of scheduling theory methods is done. Then, the task of drawing up the plan is formalized, according to the method of Johnson and the new method, including time limits of work execution. A comparison of the new method with the closest analogue is done.

Keywords: Agent, Multi-agent systems, Scheduling theory, Resources transformation process.

1 Introduction

In multi-agent models [1]-[7] one of the actual tasks types is drawing up the works execution plan for agents and coalitions. When talking about the plan, we mean the list of proposed works or activities with a defined sequence, volume and deadlines [8].Since all the factors influencing the works execution plan are almost impossible to consider, and interests of the agents are often different, the task of its compilation is multi-criteria, with fuzzy number of factors. Such tasks are usually performed by the means of the scheduling theory (ST) in two stages [9]. The first is drawing up a plan using particular method on the basis of the initial conditions of works execution and the formalized constraints. The second stage is subsequent revision of the plan to consider informal constraints. The greatest contribution to the development of ST was made by: R. Akoff, R. Bellman, G. Dantzig, G. Kuhn, T. Saati [10], S. M. Johnson [11], J. A. Zack [12], [13], A. Kofman, R. Ford, etc. To develop the algorithm of compiling the works execution plan it is necessary to formalize requirements to the method of plan compiling in the coalition model of MRTP (multi-agent resource transformation process).

2 Requirements for the method of compiling the works execution plan

Purposes of establishing requirements for the method of the project execution plan compiling are to "equip" an agent with this method as part of the Coalition model MPR and to provide the ability to solve practical problems in planning at industrial enterprises. One can formulate the requirements for the method of compiling the project execution plan:

1) Compliance to MRTP subject area. One should take into account following main features of MRTP: the focus on different types of resources and funds; the possibility of hierarchical representation of the process structure and calculation of its characteristics at each level of the hierarchy.

2) Simplicity in calculations using the method in SIM.

3) Focusing on practical tasks solving for the system consisting of two sets of resources and funds, considering time limits.

4) The optimal time schedule obtained.

As far as the number of methods described in the framework of the ST is huge, let us analyze the most well-known methods, in the first approximation corresponding to the requirements specified above.

3 A General description of the applied problem of planning

For planning implementation let us create a multi-agent model for a preventive assessment of resources and funds volume for MRTP. It is necessary to establish the balance between costs and speed of works to maximize the profit of the company, which operates this MRTP.

The key point is system's dynamism, because orders occur at arbitrary time. In this regard, to maximize profit, it is expedient to make a reserve of some resources and funds and, if necessary, to withdraw part of resources and funds in use for more urgent orders.

Let us describe the coalition MRTP models operation for planning.

a) **Initiation of work – the receipt of the order**. In this stage we are to appoint an agent A_i – the coordinator of works, and allocate human resources $\{Res_1^{A_i}, ..., Res_n^{A_i}\}$ and mechanisms $\{Mech_1^{A_i}, ..., Mech_m^{A_i}\}$ from General reserve (Pool) under his coordination. Here are possible goals of agent A_i:

- minimization of time of works performance $t^{A_i}(W_i) \rightarrow \min$;

- minimization of expenses (*i.e.*, agent's profit maximization): $S^{A_i} \rightarrow \max$;

- fewer number of "common" human resources and mechanisms due to probability of "parallel" order: $N^{A_i} \rightarrow \min$.

b) **Receipt of the "parallel" order**. In the course of order execution there is a chance of another order income. The system appoints the agent A_j, who analyzes the possibility of performing work W_j within the specified time t_j and available backup human resources $\{Res_1^{A_j}, ..., Res_t^{A_j}\}$ and mechanisms $\{Mech_1^{A_j}, ..., Mech_\nu^{A_j}\}$. In the absence of the required resources or funds, agent A_j can arrange the interaction with other agents $\{A_k, ..., A_l\}$ in WT, which may result in the creation of the coalition K_p for the joint execution of work W_j. Further, the third "parallel" order can emerge, *etc.*

c) **Conflicts**. In lack of resources or funds there is a need in interactions between agents (agent A_j and agent A_i in this example), based on their communication strategies, $\{Str^{A_j}, Str^{A_i}\}$. Thus, depending on the strategies one may need to hold the auction.

Then we shall plan the execution of work according to the method of Johnson with the aim of obtaining an optimal works execution plan by agents.

4 Methods of scheduling theory (ST) analysis

The purpose of the analysis of existing methods of work execution planning is finding the ST method and, if necessary, its revision under the above requirements.

4.1 Method of branches and borders

The first method was proposed by Land and Doig [14] in 1960 for solving the General problem of integer linear programming. The method is based on the idea of repeatedly partitioning the set of feasible solutions into subsets [15]. At each step of the method, the elements of the partition are checked to investigate if this subset contains an optimal solution. The check is made by computing lower bounds for the objective function for this subset. If the lower estimation is not less than the record – the best founded solution - the subset can be discarded. The checked subset can also be discarded if it is possible to find the best solution in it. If the value of the criterion function is less than the record, it will replace the record. The record obtained at the end of the algorithm, is the result of its work.

If it is possible to discard all the elements of the decomposition, then the obtained record is the optimal solution of the problem. Otherwise, we are to select between remained subsets the most promising one (*e.g.* lowest value for lower bounds), and perform its decomposition.

There are several methods of reducing the complexity of the algorithm, one of which is artificial overestimate lower bounds [15]. Suppose that we are interested in not only the optimal solution, but also in approximate solutions with a relative error no more than e. Then the overestimation of the lower bounds in $(1 + e)$ times leads to the desired result.

Thus, one can conclude the computational complexity of this method. The application of the method to plan the execution of works in MAS (multi-agent system) imposes serious requirements to computing resources and increases simulation time. In this regard, we consider the method of branches and borders inappropriate for plan the execution of works in MAS.

4.2 The Little's Algorithm

This algorithm [16] is used to find the optimal closed (Hamiltonian) path in the graph, which has N vertices, each vertex i is connected with any other vertex j

by bidirectional arc. Each arc has a weight $C_{i,j}$, and weights of edges are strictly positive ($C_{i,j} \geq 0$).Weights of edges form a matrix value. All the elements on the diagonal of the matrix are equated to infinity ($C_{i,j} = \infty$).

If a pair of vertices i and j are not connected (fully connected graph), then we attribute a weight equal to the length of the minimal path between the vertices i and j to the corresponding element of value matrix. If arc (i, j) is included in the resulting contour, it must be replaced by the corresponding path. The general idea of the algorithm is to divide a huge number of options into classes and to make estimations. It will allow discarding not suitable options by whole classes. The difficulty is to find a division into classes (branches) and assessments (border) which provide procedure effectiveness. The Little's algorithm can be formalized as follows [16].

a) Find the smallest element in each line of the cost matrix and subtract it from each element of the line. Do the same for columns not containing zero. As a result you will have the value matrix, each line and each column of which contains at least one zero element.

b) For each zero element of the matrix $C_{i,j}$ calculate the coefficient $H_{i,j}$, which is equal to the sum of the smallest element i of the line (excluding the element $C_{i,j}$ =0) and the smallest element j of column. Then choose the maximum $H_{i,j}$ coefficient: $H_{k,l} = \max\{H_{i,j}\}$. Add an arc (k, l) in a Hamiltonian circuit.

c) Remove the line k and the column l, change the value of the element $C_{l,k}$ for infinity (since arc (k, l) is included in the path, the reverse path from l to k is invalid).

d) If the matrix order $\neq 2$, go to step "a".

e) Introduce two missing arcs which are uniquely determined by a matrix of the same order 2 in the current oriented graph. As a result you will have Hamiltonian circuit.

During the algorithm implementation the constant calculation of the lower bound current value is made. The lower bound is the sum of all subtracted elements in lines and columns. The final value of the lower bound must match the length of the resulting path.

Thus, this method is computationally complex for works execution planning in MAS.

4.3 The random search method

Generally, the solution can be represented as a sequence of elections. If you make this election using random mechanism, then the solution is fast enough. You only need to remember the "record", *i.e.* the best of the encountered solutions. This method can be improved by considering the perspective of this or that elections in a random mechanism, *i.e.* combining random search with heuristic and local search method; or any other method of scheduling theory (ST).

Thus, this method is algorithmically simple, but the solution obtaining requires a pretty much time. In this regard, we will consider possible application of this method in the task of works execution planning in MAS. This method does not guarantee optimal timing.

4.4 Johnson's method

The separate task to solve in the framework of the ST is to make a working plan for the system with two sets of resources and funds. According to the theorem of Johnson [11], there is an optimal schedule for this case. This method will be further discussed in detail in this paper.

Thus, this method is not computationally complex for use in the task of works execution planning in MAS. The method guarantees the optimal time schedule. We will use this method for the system with two sets of human resources and funds. Here is a description of the applied planning problem for practical application of this method.

5 Preparation of project execution plan according to the Johnson's method

The separate task to solve in the framework of the ST is to make a working plan for the system with two sets of resources and funds. Let us consider this method and the possibility of using obtained results for practical ST working problems.

Let us consider a system of n tasks and m sets of human resources and mechanisms which are used to perform these tasks. We believe that our system is a conveyor-type, *i.e.* the sequence of used resources and facilities is the same for each task (the task corresponds to the order). If a lot of human resources and funds are not being used in work, the value of the regular criteria for such task is 0. We will consider time as regular criteria; you can also use other criteria (*e.g.*, money).

Let us establish designations of Johnson's problem [11]:

$$n \mid m \mid F \mid F_{max}, \tag{1}$$

where: n – number of tasks; m – number of sets of resources and facilities (explore the case of $m = 2$); F – criterion, which indicates what work is performed in conveyor type system; F_{max} – the criterion of minimizing the maximum duration of the work.

A_i - is the duration of using the first set of resources and funds for the i –th operation; B_i – the duration of using the second set of resources and funds for the i –th operation; F_i- is the duration of the i–th operation. Some A_i and B_i can be equal to 0. Simplification of the model is that:

- consider that one set of human resources and funds performs no more than one task at each moment;

- no work may be done in two sets at the same time.

Consider all A_i and B_i are known to us, *i.e.* the complexity of the work execution are known in advance. The goal is to minimize the maximum duration of a passing operation F_{max} for n works. According to [11]:

$$F_{max} \geq \sum_{i=1}^{n} A_i + B_n, \tag{2}$$

$$F_{max} \geq A_1 + \sum_{i=1}^{n} B_i. \tag{3}$$

Because the sum does not depend on the sequence of tasks execution, it is possible to minimize F_{max} only by minimizing the A_i and B_n by ordering works. From (2) it follows that:

$$F_{max} = \sum_{i=1}^{n} A_i + B_n + X, \tag{4}$$

where: X is the total idle time of the first set of human resources and mechanisms in the execution of all tasks.

According to Johnson, we can restrict consideration of schedules and take into account only those in which tasks with the first lot of resources and funds are "packed" tightly, starting with the first. *I.e.* the first set is running without downtime till A_n is completed. Bringing any timetable of this type will not lead to increase in the maximum work execution duration F_{max}. So we can assume $X = 0$.

According to Johnson's theorem: if the timetable for the system of $n \mid m \mid F \mid F_{max}$ and all works are available simultaneously, then the optimum is searched among the schedules that retain the order of execution of works on the first two (1 and 2) and two last ($m - 1$ and m) machines. By "machine" we mean a lot of resources and funds administered by the agent.

On the basis of a number of assertions presented in [11], we derive the algorithm of compiling the project execution plan (schedule) based on this theorem.

1) Set $A = \min\{A_i, i = 1, n\}$; $B = \min\{B_i, i = 1, n\}$. Accept: if any operation is performed on the array of resources and funds, then its duration is equal to 0, but this duration is involved in the comparison to ensure commonality.

2) Form a schedule in the form of a table. If $A - B \leq 0$, then loop over the set A_i, starting with the 1st element and looking for the first element is equal to A. The work i will be executed first, place it at the beginning of the schedule (on the first pass of the loop it will be the first).

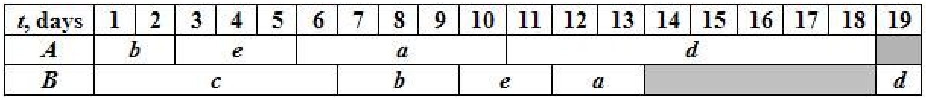
3) Otherwise, if $A - B > 0$, then loop over the set B_i, starting with the 1st item and looking for the first element equal to B. The work i will be executed last, it will be placed at the end of the schedule (on the first pass of the loop it will be the last).

4) Eliminate work posted in the schedule from the many considered works: set again A=min $\{ A_i, i = 1, n - 1\}$; B=min $\{B_i, i = 1, n - 1\}$.

5) Go to step 2, the loop runs $n - 1$ times, leaving one work.

In accordance with the theorem of Johnson, the ordering of works produced by this algorithm is optimal, *i.e.* it minimizes the maximum length of the passage. An example of a project execution plan, according to a theorem of Johnson is presented in Fig. 1.

Total time is 19 days. This is the optimum time according to the Johnson's theorem.



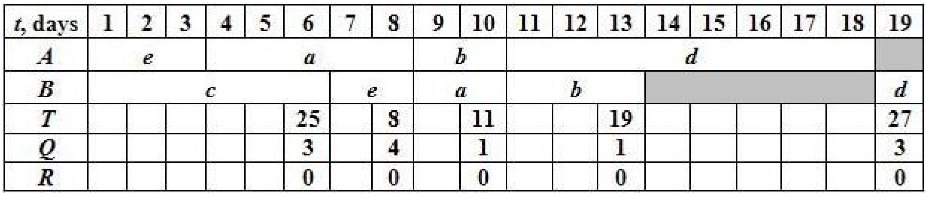
t, days	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	*b*		*e*			*a*					*d*								▨
B	*c*						*b*			*e*		*a*		▨					*d*

Fig. 1. The project execution plan by Johnson's theorem

Thus, application of this algorithm provides the optimal time of work performance. At the same time, the time limit of works execution is not being taken into account, which is necessary to solve practical problems. Therefore we need to develop a new algorithm based on the algorithm of Johnson, taking into account the time limit of execution.

6 Preparation of project execution plan taking into account time limits

Let us introduce additions to the described algorithm based on the theorem of Johnson. These additions are developed by the author of this paper. For each task we introduce critical time or date that determines the maximum duration of its execution. In case of failure to meet time constraints, undesirable consequences such as fines, penalties, *etc.* appear. In practice, time restrictions are the most common and it makes sense to investigate this algorithm especially. An example of a project execution plan taking into account the time limits is provided in Fig. 2.



t, days	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	*e*			*a*					*b*		*d*								▨
B	*c*						*e*		*a*		*b*			▨					*d*
T						25		8		11			19						27
Q						3		4		1			1						3
R						0		0		0			0						0

Fig. 2. The project execution plan by theorem taking into account time limits

Thus, we build a schedule S^{**} with constraints on the execution time of each task. Total time is 19 days, as well as in the Johnson's schedule S^*. *I.e.*, the algorithm provides the optimal time of work performance. However, no work has exceeded the reference time and there is no penalty. In the classic Johnson's schedule S^* the penalty is 5 points from a possible 12.

Thus, the proposed algorithm provides the least duration of works in terms of restrictions on the execution time of each task, as well as their performance with minimization of penalty. In this regard, we will use this algorithm to practical problems of works execution planning in manufacturing plants.

Further it is necessary to conduct comparison of the algorithm of Johnson with regard to time limits with peers.

7 The comparison of algorithm which takes into account the time limit with analogs

Analysis of the literature sources showed that, despite researches by a number of authors [11]-[16], examples of modifications of the classical Johnson's algorithm with timing constraints are almost non-existent, although this task is of great practical interest (analogous conclusion was made by Yu. A. Zach in [12]: "the Study of mathematical properties of such tasks has not received adequate attention in the literature"). The closest analogue of the algorithm developed by the author is the method of Yu. A. Zach [12], [13].

To demonstrate the differences between algorithms, the experimental scheduling with use of Yu. A. Zack's algorithm was made. As a result we obtained schedules with larger total time of operations (136 units), than in algorithm developed by the author of the paper, but with a smaller quantity of penalties (3 units) for delayed execution of certain works and a smaller number of overdue work (1 work). Thus, the algorithm Yu. A. Zack does not provide the optimal time of work performance.

8 Conclusion

The article describes one of the possible solutions of the problem of compiling the project execution plan in the multi-agent model. The requirements to the method of compiling the project execution plan for the multi-agent model are formalized and an overview of applied tasks of work planning in multi-agent models is provided. To select a method of problem solving a comparison of some methods of scheduling theory is done, and the Johnson's algorithm is selected. The application of the Johnson's algorithm for the planning of agents' work in the model provided the optimal time of work performance. This result is obtained without considering the time limit of work execution that is required for practical problems solving. In this regard, the author of this paper developed a new algorithm based on the algorithm of Johnson, taking into account the time limit of works execution.

The algorithm provides the optimal time of work performance, as well as lower penalty in comparison with Johnson's algorithm. The new algorithm has several features that distinguish it from algorithm Yu. A. Zack. A new algorithm is developed to "equip" the agent's mechanism of drawing up optimal plans of works performance (as part of the Coalition model EPR) and the need to solve specific practical problems in works execution planning.

References

1. Vittikh, V. A. Multi–agent systems for modeling of self–organization and cooperation processes [Electronic resource]. 2006-2013. Available at: http://www.cs.brandeis.edu/dept/faculty/mataric

2. Gorodetsky V. Agents and Data Mining Integration. Lecture Notes in Artificial Intelligence / V. Gorodetsky et al. (co–Editor) / vol. 5980, Springer (2010)
3. Shvetsov, A. N. Agent–oriented systems: from formal models to industrial applications / all-Russian competitive selection analytical survey articles on the priority direction "Information and telecommunication systems", 2008. – 101 p. (2008)
4. Gorodetsky, V. I., Karsaev, O. V., Samoilov, V. V., Serebryakov, S. V. Tools for open networks of agents / Izvestiya RAN. Theory and Control Systems. – M.: Nauka, 2008. – P. 106-124 (2008)
5. Skobelev, P. O. Open multi-agent system for operative processing of information in the decision-making processes: Diss. ... d–RA. tech. Sciences: 05.13.01 / SamSTU. – M.: RSL, 2003. – 418 p. (2003)
6. Wooldridge, M., Wooldridge, M., Jennings, N. Intelligent Agent: Theory and Practice / Knowledge Engineering Review, 1995. – i. 10 (2). – P. 115-152 (1995)
7. Bordini, R., Fisher, M., Visser, W., Wooldridge, M. Model Checking Rational Agents / Intelligent Systems, 2003. – Vol.18. – P. 40-47 (2003)
8. Chepasov, V. I., Kulov, S. K., Raimov, F. F. Algorithmic and software implementation of the task on the schedule: textbook / Orenburg: Orenburg. state University, 1999. – 192 p. (1999)
9. Shakhbazyan, K. V., Tushkina, T. A. the Review of the methods of scheduling for multiprocessor systems / Leningrad: Nauka, 1975. – P. 229-258 (1975)
10. Saati, T. decision-Making. Analytic hierarchy process / M.: Radio and communication (1993)
11. Johnson, S. M. Optimal two– and three–stage production schedules with setup times included / Santa Monica, California: The Rand Corp., 1953. – 402 p. (1953)
12. Zack, Yu. A. solution of the generalized Johnson problem with constraints on the timing of tasks and times of work machines. Part 1. / Problems of management, 2010. – No.3. – P. 17-27 (2010)
13. Zack, Yu. A. Solution of the generalized Johnson problem with constraints on the timing of tasks and times of work machines. Part 2. / Problems of management, 2010. – No.4. – P. 12-19 (2010)
14. Land, A. H., Doig, A. G. An automatic method of solving discrete programming problems / A. H. Land, // Econometrica. – 1960. – V. 28. – P. 497-520 (1960)
15. Method of branches and borders [Electronic resource]. M.: Institute of mathematics. S. L. Sobolev SB RAS, 2006-2013. Available at: http://math.nsc.ru/AP/benchmarks/UFLP/uflp_bb.html.
16. Little, J. D. C., Murty, K. G., Sweeney, D. W., Karel, C. An algorithm for the traveling salesman problem / Operations Research, 1963. – V. 11. – P. 972-989 (1963)