Optimization of planned solutions based on network models for large-scale problems

Yu N Kondrashov and A A Ermakov

Moscow Aviation Institute (National Research University), 4 Volokolamskoe shosse, Moscow, Russia

Email: aleral@mail.ru

Abstract. This article covers the development of the project planning algorithm based on network models for solving large-scale problems. The planning algorithm which brings the problem of plan optimization based on a network model down to a general problem of non-linear programming with the possibility of flexible formation of the goal function and constraints is completed by the iterative aggregation method for problems of non-linear programming which allows increasing the efficiency of solving large-scale problems.

1. Introduction

Network models are commonly used in planning. However, as for the optimization of planned solutions, heuristic algorithms or the reduction to specific optimization problems are used which do not allow to flexibly form the problem formulation in accordance with actual planning situations. Modern automated systems (the widely used MS Project system in particular) use one heuristic algorithm for coordination of time schedules of existing resources with the time schedules of their use by projects without the possibility of adjusting the algorithm in accordance with the actual planned situation. The authors of this article suggest a universal algorithm for optimal project planning based on network models with the use of non-linear programming and an effective method for its implementation for solving large-scale problems.

2. Planning method

The work [1] describes an original project planning algorithm based on network models. This algorithm is based on the specified coefficients of the usage rate of resources when performing project work.

The labor intensity value $V_{ij}^r$ is known for each resource $r \in R$ ($R$ — the set of types of the used resources) needed to perform the $(i, j)$-th work of the network schedule of the project. When the value of the work execution time is $t_{ij}$ the equation

$$\varphi^r(i,j)_{cp} = \frac{V_{ij}^r}{t_{ij}}$$

(1)

determines the number of units of the $r$-th resource consumed by the work during its execution. The values of the work execution time $t_{ij}$ are subject to technological constraints

$$t_{r \min}(i,j) \leq t^r(i,j) \leq t_{r \max}(i,j)$$

(2)
where: \( t^{\min}(i,j) \), \( t^{\max}(i,j) \) are minimum and maximum technologically acceptable amounts of time of the execution of the \((i,j)\)-th work. These values may vary for different types of resources consumed by the work (using the index \( r \)).

We can change the constraint 2 as follows

\[
\frac{v^r(i,j)}{t^{\min}(i,j)} \geq \frac{v^r(i,j)}{t^{(i,j)}} \geq \frac{v^r(i,j)}{t^{\max}(i,j)}
\]  

(3)

and derive the constraint for the number of units of the \( r \)-th resource consumed by the work

\[
\varphi^r(i,j)_{\max} \geq \varphi^r(i,j)_{cp} \geq \varphi^r(i,j)_{\min}
\]  

(4)

By changing the inequality (4), we derive the inequality (5):

\[
\frac{\varphi^r(i,j)_{\max}}{\varphi^r(i,j)_{cp}} \geq \frac{\varphi^r(i,j)_{\min}}{\varphi^r(i,j)_{cp}}
\]  

(5)

The factor of intensity \( y_r(i,j) \) of putting the \( r \)-th resource into the \((i,j)\)-th work is defined as the ratio

\[
y_r(i,j) = \frac{\varphi^r(i,j)}{\varphi^r(i,j)_{cp}}
\]  

(6)

The intensity factor values define the ratio of the number of units of the resource usage to its average value during work execution. The intensity factor is non-dimensional.

When \( \varphi^r(i,j) = \varphi^r(i,j)_{cp} \), the intensity factor value equals 1 and determines the average intensity of putting the used resource into the work.

The area of definition of the intensity factor is subject to the constraint

\[
y_r(i,j)_{\max} \geq y_r(i,j) \geq y_r(i,j)_{\min}
\]  

(7)

The number of units of the \( r \)-th resource consumed by the work during its execution is calculated by the formula

\[
\varphi^r(i,j) = \varphi^r(i,j)_{cp} y_r(i,j)
\]  

(8)

The work execution time for the \( r \)-th resource is calculated by the formula

\[
t^r(i,j) = \frac{v^r(i,j)}{\varphi^r(i,j)}
\]  

(9)

There is an inverse correlation between the work execution time and the intensity factor. When the value of the work intensity factor for the \( r \)-th resource increases, the number of units of the resource consumed by the work increases as well, while the work execution time for this resource decreases.

When the value of the work intensity factor for this resource decreases, the number of units of the resource consumed by the work decreases as well, while the work execution time for this resource increases.

The work execution time is determined as the maximum of the execution time for all used resources with the re-calculation of the units of the rest of the resources consumed by the work.

The specified intensity factor is interpretable, which makes it convenient to use it for solving project planning problems.

Resource constraints for project planning problems for each type of the resource \( r \) are set by the values \( \Phi_{rs}(r \in R, s \in S) \), where \( s \) is the stage of the planned period. The stages of the planned period \( \tau_s = [t_{s-1}, t_s], s = 1, 2, ..., S, t_0 = 0, t_S = T \) are defined by the planned period \([0, T]\) and divide it into \( S \) segments. The length of the segments is calculated by the applied planning problem. For the analyzed network schedules of the software development, the length of the segment amounted to one week (such length of the planned period segment is maintained in the MS Project system in which the analyzed optimization algorithms were tested for setting resource constraints). The function \( \Phi_{rs}(r \in R, s \in S) \) is piecewise-constant.
The constraints of the planning problem take the following form

$$\sum_{n \in N} \sum_{i,j} \psi_n^r(i,j) \sigma_{cp}^r(i,j) \leq \Phi_{rs} \quad r \in R, s \in S, n \in N$$  \hspace{2cm} \text{(10)}$$

where $n \in N$ is the set of projects executed during the period $[0,T]$ which may be at different stages of execution.

The values of functions $\Phi_{rs}$ may vary for different types of resources when solving planning problems which indicates different strategies for resource development at the company or one of its divisions. For example, the strategy determines the dynamics of changes in the company’s staff with a certain qualification.

When setting the criterion $F(\bar{y}) \rightarrow \min (\max)$, the planning problem based on the analyzed resource network model can be brought down to a non-linear programming problem for which standard optimization methods and algorithms can be used.

The used criteria can be divided into temporary and resource ones. Temporary criteria include criteria of minimization of project execution time, as well as the minimization of deviation of project completion dates from the estimated ones. Resource criteria include criteria of minimization of deviation of the resource consumption function from the given resource constraints.

The number of arguments for solving the problem of the plan optimization is defined by the number of projects, included works and types of resources used in the work. For the actual software development projects used for optimization, the number of used resource types varied from 5 to 9. In this case, the number of arguments varies from 97 to 279. This means that the optimization problem is a large-scale problem, which significantly increases the time needed to solve it.

One of the possible ways to solve the problem of the increase in the plan optimization time based on the resource network model is to use iterative aggregation methods [2].

The general method of iterative aggregation for solving non-linear optimization problems is as follows. There is a problem of the following form

$$f(x) \rightarrow \max$$

$$g_s(x) \geq 0 \quad s \in S$$  \hspace{2cm} \text{(12)}$$

The authors of the work [3] suggest a general solution pattern in which, according to the structure of variable aggregation, the variable vector $\tilde{x}$ is divided into $V$ subvectors $\tilde{x} = (\tilde{x}^1, \tilde{x}^2, ..., \tilde{x}^V)$, the vector of dual variables $\tilde{y}$ which corresponds to the constraints (12) is divided into $M$ subvectors $\tilde{y} = (\tilde{y}^1, \tilde{y}^2, ..., \tilde{y}^M)$ which define the aggregation structure of the constraint. In this way, variables and constraints included in the corresponding subvectors are aggregated.

In order to find aggregated arguments, $V$ vectors $\bar{\Phi}_v^\sigma, v = 1, V$ shall be set (where $\sigma$ is the iteration index) which will serve as vectors of the direction of changes in variables during the solution of the aggregated problem (correlation of block problems). The vectors $\bar{\Phi}_v^\sigma$ are set as follows:

$$\bar{\Phi}_v^\sigma = (0,0,...,0,x_k^{\sigma-1},x_{k+1}^{\sigma-1},...,x_{k+l}^{\sigma-1},0,...,0)$$  \hspace{2cm} \text{(13)}$$

where $x_k^{\sigma-1}, x_{k+1}^{\sigma-1}, ..., x_{k+l}^{\sigma-1}$ is the block of variables of a block model aggregated into the $v$-th variable. A substitution is applied to the aggregated problem

$$\tilde{x} = \tilde{x}^{\sigma-1} + \sum_{v=1}^V z_v \bar{\Phi}_v^\sigma$$  \hspace{2cm} \text{(14)}$$

where $z_v$ is the variable of the aggregated problem. Thus, one direction of change is sought for a group of aggregated variables during the solution of the aggregated problem (solutions to block problems are corrected). Whereas the number of arguments of the aggregated problem is less than that of the detailed problem. When $z_v = 0$, $x_k^\sigma = x_k^{\sigma-1}$, i.e. the solutions of the detailed and aggregated problems at two subsequent iterations match together. Taking into account (13), the formula (14) takes the following form

$$\tilde{x} = \tilde{x}^{\sigma-1} + \sum_{v=1}^V z_v \bar{\Phi}_v^\sigma$$  \hspace{2cm} \text{(14)}$$
\[ \bar{x} = \sum_{v=1}^{V} (z_v + 1) \bar{q}^\sigma_v \]  

(15)

Similarly, the process of changing the vector of dual variables \( \bar{y} \) is arranged which is divided into \( M \) subvectors. The following vectors are given

\[ \bar{y}^\sigma_m = (0,0,...,0,y_{i_1}^{\sigma-1},y_{i_1+1}^{\sigma-1},...,y_{i_1+1}^{\sigma-1},0,...,0) \quad m = 1,M \]

(16)

which results in a formula for dual variables that is similar to (15)

\[ \bar{y} = \sum_{m=1}^{M} (\lambda_m + 1) \bar{y}^\sigma_m \]

(17)

where \( \lambda_m \) define directions of change in groups of dual variables of aggregated constraints during the solution of the aggregated problem.

The iteration process is arranged as follows. Let the full vector \( \bar{x}^{\sigma-1} \) and the full vector of constraint estimates \( \bar{y}^{\sigma-1} \) be defined after the \( \sigma - 1 \)-th iteration. After the substitution of (15) and (16), the aggregated problem is solved

\[ f(\bar{z}) + \sum_{S=1}^{S} y_S^{\sigma-1} g_S(\bar{z}) - \alpha \sum_{v=1}^{V} z_v^2 \rightarrow \max \]

(18)

with the aggregated constraints of

\[ \sum_{s \in S_m} y_s^{\sigma-1} g_s(z) \geq 0 \quad m = 1,M \]

(19)

Its solution results in the determination of \( z_j (j = 1,N) \) and disaggregation of the found solution, which is a starting point for solving block problems

\[ x_n^\sigma = x_n^{\sigma-1} + z_j x_n^{\sigma-1} \]

(20)

Aggregated constraint estimates \( \lambda_m^\sigma \) can be calculated by the formula

\[ \lambda_m^\sigma = [\lambda_m^{\sigma-1} + \alpha \sum_{s \in S_k} y_s^{\sigma-1} g_l(z)]^+ \]

(21)

based on which the constraint estimates that are initial for solving block problems are calculated by the formula (17).

The block problems are solved as follows. There a corresponding variable subvector \( \bar{x}_k, k \in K_j \) for each block problem. If the subvectors \( \bar{x}_k \) and \( \bar{x}_p \) (define the aggregation structure) match, then the problem is a block problem. Otherwise, there is no blocking. This means that this method can be used for solving large-scale optimization problems with no blocking. However, practice shows that blocking causes the increase in the efficiency of using the iterative aggregation method.

For each subvector \( \bar{x}_k, \ k \in K_j \), the following problem is solved in the \( j \)-th block

\[ 2f(\bar{x}^\sigma,...,\bar{x}_{k-1}^\sigma,\bar{x}_k,\bar{x}_{k+1}^\sigma,...,\bar{x}_n^\sigma) - \alpha \sum_{n \in K} (x_n - \bar{x}_n^\sigma)^2 \rightarrow \max \quad k \in K_j \]

(22)

with the constraints of

\[ g_l(\bar{x}_1^\sigma,...,\bar{x}_{k-1}^\sigma,\bar{x}_k,\bar{x}_{k+1}^\sigma,...,\bar{x}_n^\sigma) \geq 0 \quad s = 1,S \]

(23)

Thus, \( J \) optimization problems are solved in each \( j \)-th of which the values of the elements of the subvector \( \bar{x}_k, \ k \in K_j \) are determined at constant values of the other elements of the vector which were derived by disaggregating the solution of the aggregated problem. The values of all components of the vector \( \bar{x}_k, \ k \in K_j \) are obtained from the set of local problems.

Based on the results of the solution of all local problems, constraint estimates are determined which are calculated by the formula

\[ y_s^\sigma = [y_s^\sigma - \alpha g_s(x)]^+ \quad s = 1,S \]

(24)

and the iteration repeats.
The iteration process ends if the results of solving problems in blocks at two consecutive iterations match with a specified accuracy \( |x_k^\sigma - x_k^{\sigma-1}| \leq \varepsilon \) (where \( k \in K, \varepsilon \) is the specified accuracy of the solution). It is proved that the optimal solution of the problem (11)-(12) can be obtained in the range of such iterative process. Different applied large-scale non-linear optimization problems can be formulated based on this general method.

With respect to the analyzed problem of plan optimization based on a network model with the use of intensity factors as arguments, the structure of block and aggregated problems can be determined based on the network model. The network schedule of the project has a hierarchy of works in which the "summary problem" (this term is used in the MS Project system) and the network schedule fragment that elaborates it can be distinguished. Such hierarchy is the basis for formulating aggregated and block problems during implementation of the iterative aggregation algorithm.

Block problems are solved for the elaborating fragments of the network schedule, while the aggregated problem is solved based on the network schedule of "summary problems". The correlation between temporal relationships of the elaborating fragments is set by additional constraints (matching information \( T_i \) and \( T_j \) — the starting and completion times of the "summary problem" obtained during the solution of the aggregated problem) and gives a full picture of the original network schedule.

The dimension of block problems is smaller due to the fewer number of arguments (a subset of arguments \( y_{\sigma}^i(i, j) \) related to the works that elaborate one "summary problem" is considered).

To formulate arguments of an aggregated problem, the variables of the original planning problem shall are divided into blocks \( \bar{y} = (\bar{y}^1, \bar{y}^2, ..., \bar{y}^w) \) which correspond to the aggregation structure (the number of arguments of the aggregated problem is equal to W). The aggregation structure is determined by the "summary problems".

W vectors \( \bar{y}_{\sigma}^w, w = 1, W \) are given (where \( \sigma \) is the iteration index) which will serve as vectors of the direction of change in the intensity factors during the solution of the aggregated problem.

The vectors \( \bar{y}_{\sigma}^w \) are set as follows

\[
\bar{y}_{\sigma}^w = (0, ..., 0, y_{nir}^\sigma, ..., y_{nip}^\sigma, ..., y_{nk}^\sigma, 0, ..., 0)
\]  

The non-zero elements of each \( w \)-th vector correspond to the arguments of the detailed problem that are aggregated into the \( w \)-th variable. \( (y_{nir}^\sigma, ..., y_{nip}^\sigma, ..., y_{nk}^\sigma) \) is a block of variables of the detailed model that are aggregated into the \( w \)-th variable.

In accordance with the general method for the aggregation of non-linear problems, a substitution is applied to the aggregated problem

\[
\bar{y} = \bar{y}^{\sigma-1} + \sum_{w=1}^w z_w \bar{y}_{\sigma}^w
\]  

where \( z_w \) is the variable of the aggregated problem.

It means that if a subset of network schedule operations is aggregated into one "summary problem", then for all work intensity factors that elaborate this "summary problem", a common corrective action is sought during the solution of the aggregated problem.

The aggregated problem takes the following form

\[
f(Z_w) - c \sum_{l=1}^N z_l^2 \rightarrow \max(\min)
\]

\[
\sum_{n \in N} \sum_{(i,j) \in U_n} \bar{q}_n^r(i, j) c_p Z_{w_r} \leq \Phi_{rs} \quad r \in R, s \in S, n \in N
\]

Solving the aggregated problem allows to determine the values of the arguments \( Z_{w_r}^\sigma, w = 1, W \) and calculate the original values of arguments of the detailed problems

\[
y_{n}^r(i, j) = \sum_{w=1}^w (Z_{w_r}^\sigma + 1) \bar{q}_n^r \quad n \in N, \{i, j\} \in U_n
\]  

The results of solving the aggregated problem including the matching information \( T_{lw} \) and \( T_{lw} \) for each summary problem can be used for solving block problems.

The block problems (the \( w \)-th block) take the following form
\[ f(y_n^r(i,j)) \rightarrow \max(\min), \{i,j\} \in U_{nw} \]  
\[ \sum_{n \in N} \sum_{\{i,j\} \in U_{nw}} q_n^r(i,j) \sigma y_n^r(i,j) \leq \Phi_{rs} \forall r \in R, s \in S, n \in N \]  

When \( z_w = 0 \), \( y_n^r(i,j) = y_n^{r-1}(i,j) \) and the solutions of the detailed and aggregated problems at two subsequent iterations match together, it means that the optimal solution is obtained.

3. Findings

Bringing planning based on a network model down to a non-linear programming problem allows to flexibly form criteria and constraints in accordance with the specific planning situation. When selecting criteria, the sets of projects and their priorities, scheduled project baselines, resource constraints and their subsets included in the criteria, as well as the constraints may differ. This fact sets the analyzed planning algorithm apart from the existing ones that solve specific planning problems based on priorities or the optimization of the cost-time relationship.

The analyzed planning algorithm was implemented as an addition to the well-known MS Project system, tested for solving real problems of planning software development projects and proved itself to be effective. Namely, the estimated project execution time in the mentioned examples was reduced to 13\% compared to the standard planning algorithm implemented in the MS Project system. The Powell non-linear programming algorithm was used for optimization.

The use of the iterative aggregation method to optimize plans based on network models allows to reduce the solution time. The experiments carried out for real software development projects and their comparison with the results of direct optimization mentioned above demonstrated the convergence of the iterative method of solution to the optimal solution with a simultaneous decrease (up to 40\%) in the optimization time.

4. Conclusion

The analyzed method of optimization of planned decisions, along with the iterative aggregation algorithm, allows to improve the existing mechanism of planning based on network models. The described algorithms were implemented as additional modules of the MS Project system. Compared to the planning algorithm implemented in the system, the modules provide a flexible formation of the planned problem by selecting and forming a structure of criteria and constraints and its effective solution for large-scale problems. The results of the comparative analysis of the use of standard algorithms of the MS Project system and the developed modules show an improvement of the quality of made decisions when planning actual software development projects.

References
[1] Kondrashov Yu N and Chashin M O 2016 Project planning algorithm Bulletin of the Financial University 4(94) 78-84
[2] Dudkin L M 1979 Iterative Aggregation and Its Application in Planning (Moscow, USSR: Ekonomika)
[3] Theory and practice of using aggregation methods for planning and management: Materials of the Meeting of the Academy of Sciences of the Armenian SSR. Yerevan, 1983.
[4] Bellenguez-Morineau O and Neron E 2007 A Branch-and-Bound method for solving Multi-Skill Project Scheduling Problem RAIRO-Oper. Res. 41(2) 155-70
[5] Chaparro A, Madrid A, Bustos K and Rios A 2011 Designing a strategic planning tool applying artificial neural network theory Revista Internacional Administracion & Finanzas 4(4) 31-44
[6] Clough R, Sears G A and Sears S 2000 Construction Project Management 4th ed. (New York, US: John Wiley & Sons) p 360
[7] Epstein M 2002 Risk Management of Innovative R&D Project (Helsinki, Finland: Helsinki School of Economics) p 283
[8] Lee J W and Kim S H 2000 Using analytic network process and goal programming for interdependent information system project selection Computers & Operations Research 27
[9] Maserang S 2002 Project Management: Tools & Techniques *Systems Analysis & Design* Retrieved from http://www.umsl.edu/~sauterv/analysis/488_f02_papers/ProjMgmt.html, last accessed 10.01.2020