

Feasibility Identification for Networks with Generalized Precedence Relations (GPRs)

Wanan Cui¹, Bin Song², Xin Xie², Chaoyuan Yue¹, and Zheng Yin¹

¹ Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, PR China, cuiwanan@163.com,

WWW home page: <http://cse.hust.edu.cn>

² Institute of public management, China University of Geosciences, Wuhan, 430074, PR China, admin@whsumi.com,

WWW home page: <http://unit.cug.edu.cn/glxy/>

Abstract. Network with Generalized Precedence Relations (GPRs) is a very important model in network planning technology which is the core technology of project management software. The most important problem should be solved first is to identify the network feasibility under GPRs in project planning phase, which determines whether its successive works can be processed successfully or not. This paper puts forward the definition of parallel subnetwork and studies the methods of its identification and feasibility discrimination. And then, we propose distinguishing methods for network feasibility in an acyclic network with GPRs, study the identifying methods for subprojects and GPRs arcs which need to be adjusted when the network is infeasible. The analysis of time complexity and computational example manifests that the proposed algorithm is feasible and effective in practice.

1 Introduction

Network planning technology is the core technology of project management software which is the most important part in large project information systems. With the help of network models, we can establish close relations among subprojects, between resources needed by each subproject and cost, which is an important basis for optimizing project scheduling, investment management, resources allocation etc by computer. Most network planning software used at present such as Primavera Project Planner (P3), Microsoft Project for Windows and so on are based on classical CPM [1] or PERT [2] model.

Please use the following format when citing this chapter:

Cui, W., Song, B., Xie, X., Yue, C., Yin, Z., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.711-720.

As we known, the classical CPM [1] or PERT [2] model reflects the strict partial ordering [3] that exists among subprojects of a project, i.e. a successor subproject only can be started when its all predecessor subprojects have already been finished. However, there are other precedence relations called generalized precedence relations (GPRs) [3] between subprojects of a project besides the strict precedence relation in project planning in practice. Therefore, GPRs network model is more suitable to large project management than the classical CPM [1] or PERT [2].

Elmaghraby and Kamburowski [3] constructed the structure of GPRs network model based on the works of Roy [4] , Kerbosh and Schell [5]. Other related studies include De Reyck and Herroelon [6], Sakellaropoulos and Chassiakos [7], etc.

There are very complex relationships of time lags between different nodes in GPRs networks. Therefore, it is difficult to consider all constraint conditions carefully and completely in project planning phase. As a result, there may be contradictable time lag constraints which will lead to network infeasible. However, this phase determines whether its successive works such as identifying critical paths and critical subnetworks can be processed successfully or not. Therefore, the most important problem should be solved first is to identify network feasibility under GPRs.

Elmaghraby and Kamburowski [3] and Cesta et al [8] gave the definition of network feasibility in relation to this problem. In this paper, we study identifying methods for the feasibility of acyclic networks with fixed subproject durations under GPRs based on their works.

The remainder of this paper is organized as follows. Section 1 introduces the notation and terminology used in this paper. Section 2 gives some important definitions about networks with GPRs. Section 3 is problem statement. Section 4 puts forward identifying methods for network feasibility and analyzes the time complexity about the proposed algorithm. Section 5 gives a computational example. Section 6 is reserved for our overall conclusion.

2 Notation, Terminology, and Basic Definitions

We mainly use notation and terminology in line with Elmaghraby and Kamburowski [3] and modify some of them if it is necessary.

$G = (N, \mathcal{A}, \mathcal{P}, d, g)$: a network under GPRs, where N is the set of nodes, A is the set of arcs, \mathcal{A} is the set of subprojects arcs, \mathcal{P} is the set of GPRs arcs and $A = \mathcal{A} \cup \mathcal{P}$.

π (or $\pi_{ij}, \pi_k, \pi_{h,PS_k}$): path (or a path between node i and j , a path numbered k , path h on parallel subnetwork PS_k).

d_k (or $d, d_k^l, d_k^u, d_{ij}, d_\pi, d_{PS_k}^l, d_{PS_k}^u$): the duration of a subproject k (or a subproject (i,j) , path π , PS_k). The superscript l (or u) indicates the lower (or upper) bound of the subproject's (or path's, PS_k 's) duration.

g_{ij} (or $g, g_{ij}^l, g_{ij}^u, g_{ij}^{\pi}, g_{ij}^{l,up}, g_{ij}^{u,up}$): the length of a GPRs arc between node i and j . The superscript l (or u) and up denote the lower (or upper) bound and updated value of the length respectively, and the subscript π denotes the total length of GPRs arcs on path π .

g_{pub}^l (or g_{pub}^u): the minimal (or maximal) accumulative total time lag of several common arcs.

g_{pub,PS_k}^l (or g_{pub,PS_k}^u): The minimal (or maximal) accumulative total time lag of several common arcs can be taken in parallel subnetwork k .

$g_{pub,PS}^{l,up}$ (or $g_{pub,PS}^{u,up}, g_{ij,PS}^{l,up}, g_{ij,PS}^{u,up}$): The updated minimal (or maximal) accumulative total time lag of several common arcs (or common arc (i,j)) can be taken in the set of parallel subnetworks.

g_{priv}^l (or g_{priv}^u): The minimal (or maximal) accumulative total time lag of several GPRs arcs only passed by one parallel subnetwork.

t_j : the realization time of node j . s_h : the start-time of subproject h .

f_h : the finish-time of subproject h . PS: the set of parallel subnetworks.

$PS_{i_0j_0}$: a parallel subnetwork with begin-node i_0 and end-node j_0 .

P_{ij} (or $P_{PS_{i_0j_0}}$): the set of paths between node i and j (or the set of paths on $PS_{i_0j_0}$).

$PS^{CA(i,j)}$ ($PS_{i_0j_0}^{CA(i,j)} \in PS^{CA(i,j)}$): the set of parallel subnetworks which pass common arc (i,j) , where $PS_{i_0j_0}^{CA(i,j)}$ is a parallel subnetwork passing (i,j) .

$deg_G^-(i)$ (or $deg_G^+(i)$): in-degree (or out-degree) of node i .

$CA(i,j)$: common arc (i,j) . ML $_{ij}$: a minimal loop between node i and j .

Subnetwork (SN): A subnetwork is composed by all paths between node i and j .

Parallel Subnetwork (PS): The meaning is the same as completely parallel network in Dodin [9], i.e. a parallel subnetwork is composed by at least two parallel paths with identical direction between two nodes. For example, path $(1 \rightarrow 2 \rightarrow 6)$ and $(1 \rightarrow 5 \rightarrow 6)$ compose PS_{16} in Fig. 1.

Network Feasibility (NF): We incorporate the definition of network feasibility given by Elmaghraby and Kamburowski [3] and Cesta et al [8] to define it as follows:

A network $G = (N, \mathcal{A}, \mathcal{P}, d, g)$ is called feasible if there are project schedules which satisfy conditions: (1). $t_j - t_i = d_{ij}$, $\forall (i, j) \in \mathcal{A}$; (2). $g_{ij}^l \leq t_j - t_i \leq g_{ij}^u$, $\forall (i, j) \in \mathcal{P}$; (3). $t_0 = 0$; (4). $t_j, t_i \geq 0$.

The same definition of network feasibility is used to define the feasibility of a parallel subnetwork. d, g)

Parallel Subnetwork Feasibility: A parallel subnetwork $PS = (N, \mathcal{A}_{PS}, \mathcal{P}_{PS}, d, g)$ is called feasible if there are project schedules which satisfy conditions: (5). $t_j - t_i = d_{ij}$, $\forall (i, j) \in \mathcal{A}_{PS}$; (6). $g_{ij}^l \leq t_j - t_i \leq g_{ij}^u$, $\forall (i, j) \in \mathcal{P}_{PS}$; (7). $t_j, t_i \geq 0$, where \mathcal{A}_{PS} and \mathcal{P}_{PS} are the set of subprojects and GPRs arcs in the PS respectively.

Minimal Loop (ML): A loop which does not contain any other loops is called a minimal loop. For example, path (0→1→2) and (0→3→4→13) compose $ML_{0,13}$ in Fig. 1.

Common Arc (CA): A GPRs arc passed by several parallel subnetworks is called a common arc. For example, $CA(6,10)$ is passed by $PS_{5,10}$ and $PS_{6,16}$ in Fig. 1.

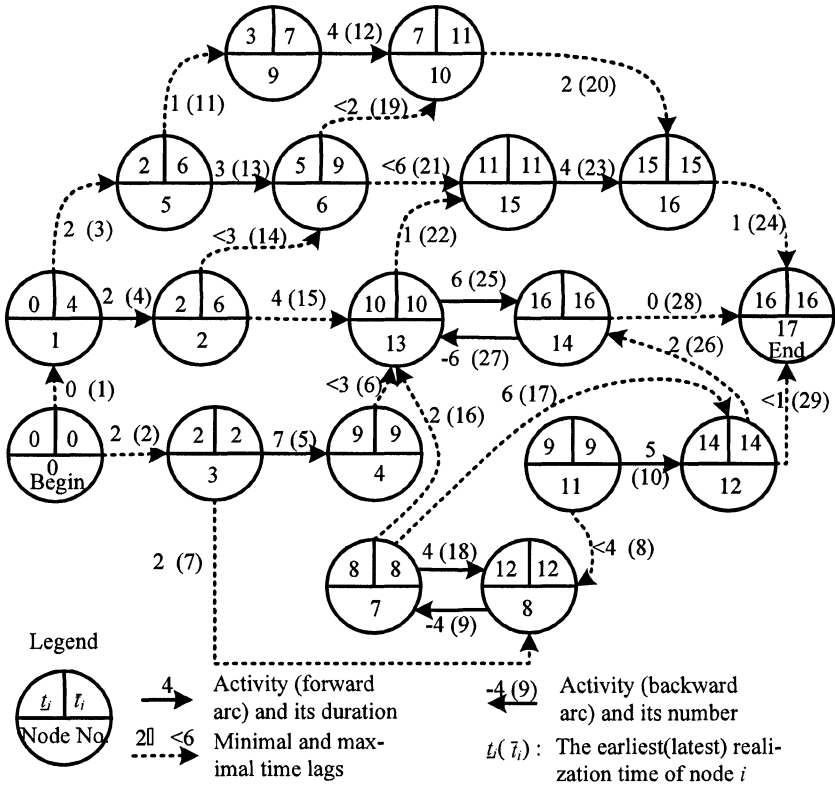


Fig. 1. An improved example of Elmaghraby and Kamburowski [3]

3 Problem Statement

A project is composed by many subprojects. According to technological conditions and resource constraints, we can get the datum of GPRs between different subprojects. However, we can not judge whether there are contradictions with these datum by our experiences or intuitions due to complex GPRs, so we can not determine whether the network schedule is feasible or infeasible.

Our problem is: How can we identify whether there is at least one feasible project schedule in a given set of subprojects durations $d_{ij} (\forall (i, j) \in \mathcal{A})$ and GPRs length interval $[g_{ij}^l, g_{ij}^u] (\forall (i, j) \in \mathcal{P})$ in an acyclic network under GPRs? And should we adjust which subprojects' durations or/and the length of GPRs if there is no feasible project schedule?

The first problem can be solved by identifying the solution of the linear program (1)-(4). Simplex method and ellipsoid method can be used to identify the network feasibility. The former performs very well in practice but needs exponential time in the worst conditions. The later is a polynomial time algorithm but has little value in practice [10].

For the second problem, Bartusch [11] and Zhan [12] used cycle structure to identify some contradict GPRs if subprojects have fixed durations. However, the cycle structure does not conform to the reality of a project because there is no work to be done over again in the project under GPRs.

Furthermore, these methods or models are helpless for our problems if the durations of subprojects are stochastic.

We base our models and algorithms on an acyclic network structure with GPRs and aim at to extend them to stochastic durations of subprojects. The later will be discussed in another paper.

4 Feasibility Identification and Complexity Analysis

4.1. Feasibility Identification for Parallel Subnetworks

Parallel subnetwork feasibility can be identified by theorem 1.

Theorem 1: A parallel subnetwork is feasible if and only if its intersection of the duration intervals of all parallel paths is not empty.

Proof: Necessity: Suppose that parallel subnetwork PS_{i_0, j_0} is composed by m paths named $\pi_1, \pi_2, \dots, \pi_m$ respectively.

If PS_{i_0, j_0} is feasible $\Rightarrow t_{j_0}$ certainly exists. $\Rightarrow t_{j_0} \in \bigcap_{k=1}^m [d_{\pi_k}^l, d_{\pi_k}^u] \neq \Phi$, where $[d_{\pi_k}^l, d_{\pi_k}^u] (k=1, 2, \dots, m)$ is the duration interval of path k .

Sufficiency: $\bigcap_{k=1}^m [d_{\pi_k}^l, d_{\pi_k}^u] \neq \Phi \Rightarrow t_{j_0}$ certainly exists. $\Rightarrow PS_{i_0, j_0}$ is feasible.

Corollary 1: For parallel subnetwork PS_{i_0, j_0} , If $\forall \pi_k (k = 1, \dots, m), \exists [g_{ij}^l, g_{ij}^u] = [g_{ij}^l, +\infty]$ then PS_{i_0, j_0} is feasible, where $P_{PS_{i_0, j_0}} = \{\pi_1, \pi_2, \dots, \pi_m\}$.

Theorem 2: The necessary condition for GPRs network feasibility is that all parallel subnetworks in network G are feasible.

Proof: The conclusion is obvious.

4.2. An Identifying Method for Network Feasibility (Algorithm 1)

Suppose that there are only one network begin- and end-nodes.

4.2.1. Basic Ideas of Algorithm 1

①. Searching all minimal loops in network G under GPRs, then combining minimal loops which have the same start- and finish-nodes, and so getting all parallel subnetworks (including some parallel subnetworks generated by eliminating some arcs of subnetworks).

②. The feasibility of parallel subnetworks is the necessary condition of the network feasibility and it can be identified easily. Therefore, we check the feasibility of all parallel subnetworks firstly.

③. Searching the set $PS^{CA(i,j)}$ which passes $CA(i,j)$, then computing the maximal interval value of $CA(i,j)$ in each parallel subnetwork $PS_{i_0j_0}^{CA(i,j)} \in PS^{CA(i,j)}$ and the intersection of these maximal intervals which is used as the new interval value of $CA(i,j)$ (called $CA(i,j)$ updated).

④. Checking the feasibility of all parallel subnetworks with updated value of the $CA(i,j)$, if all of them are feasible then the network G is feasible, otherwise, the network G is infeasible.

4.2.2. Steps of Algorithm 1

Step 1. Searching all minimal loops in network G under GPRs: ①.Numbering arcs in regularity: We can modify the method of numbering nodes in regularity or ternary tab method for an optimal tree. ②.Algorithm of searching minimal loops: We can modify ROOTP or ROOT algorithm to search all minimal loops in a network.

Step 2. Searching the set of parallel subnetworks: Searching minimal loops which have identical start-node i and finish-node i_{end} , then incorporating them as parallel subnetwork $PS_{i,i_{end}} (i \in \{0, \dots, n-1\}, i_{end} \in \{1, \dots, n\} \text{ and } i_{end} > i)$, and regarding other minimal loops as parallel subnetworks. All parallel subnetworks belong to the set of parallel subnetworks.

Step 3. Searching parallel subnetworks which pass $CA(i,j) \in \mathcal{P}$ and incorporating them into the set of $PS^{CA(i,j)}$.

Step 4. Computing the duration interval of $PS_{i_0j_0}^{CA(i,j)} \in PS^{CA(i,j)}$. If the interval is empty then $PS_{i_0j_0}^{CA(i,j)}$ is infeasible. Adjusting the durations of subprojects or/and the length of GPRs arcs which belong to $PS_{i_0j_0}^{CA(i,j)}$, making $PS_{i_0j_0}^{CA(i,j)}$ feasible.

Step 5. Computing the maximal interval of $CA(i,j)$ in $PS_{i_0j_0}^{CA(i,j)} \in PS^{CA(i,j)}$ in the condition of guaranteeing parallel subnetworks $PS_{i_0j_0}^{CA(i,j)}$ feasible: ①.Let the duration of parallel subnetwork k be $[d_{PS_k}^l, d_{PS_k}^u]$, computing the maximal interval of common arc g_{pub} . ②. The maximal interval of common arc g_{pub} in parallel subnetwork PS_k can be computed by formula (8).

$$[\mathcal{G}_{pub,PS_k}^l, \mathcal{G}_{pub,PS_k}^u] = [\max\{d_{PS_k}^l - \sum_{(i,j) \in \pi_{h,PS_k}} d_{ij} - \sum_{\pi_{h,PS_k}} \mathcal{G}_{priv}^u, \sum_{\pi_{h,PS_k}} \mathcal{G}_{pub}^l\}, \min\{d_{PS_k}^u - \sum_{(i,j) \in \pi_{h,PS_k}} d_{ij} - \sum_{\pi_{h,PS_k}} \mathcal{G}_{priv}^l, \sum_{\pi_{h,PS_k}} \mathcal{G}_{pub}^u\}] \quad (8)$$

③. The maximal accumulative total interval of several common arcs in the set of parallel subnetworks can be determined by formula (9).

$$[\mathcal{G}_{pub,PS}^{l,up}, \mathcal{G}_{pub,PS}^{u,up}] = \bigcap_{k=1}^n [\mathcal{G}_{pub,PS_k}^l, \mathcal{G}_{pub,PS_k}^u] \quad (9)$$

④. The maximal interval of $CA(i,j)$ which belongs to \mathcal{G}_{pub} in the set of parallel subnetworks can be determined by formula (10).

$$[\mathcal{G}_{ij,PS}^{l,up}, \mathcal{G}_{ij,PS}^{u,up}] = \bigcap_{k=1}^n [\max\{(\mathcal{G}_{pub,PS_k}^{l,up} - \sum_{(i',j') \in \mathcal{G}_{pub,PS_k}, (i',j') \neq (i,j)} \mathcal{G}_{ij'}^u), \mathcal{G}_{ij}^l\}, \min\{(\mathcal{G}_{pub,PS_k}^{u,up} - \sum_{(i',j') \in \mathcal{G}_{pub,PS_k}, (i',j') \neq (i,j)} \mathcal{G}_{ij'}^l), \mathcal{G}_{ij}^u\}], (i,j) \in \mathcal{G}_{pub} \quad (10)$$

Step 6. Computing the intersection of maximal intervals of the $CA(i,j)$ in $PS_{i_0,j_0}^{CA(i,j)} \in PS^{CA(i,j)}$. If the intersection is empty then the network is infeasible, so adjusting the durations of subprojects or/and the length of GPRs arcs which belong to $PS_{i_0,j_0}^{CA(i,j)}$, making $PS_{i_0,j_0}^{CA(i,j)}$ feasible.

Step 7. Checking the feasibility of each parallel subnetwork according to the updated value of the $CA(i,j)$. If all of them are feasible then the network is feasible, otherwise, the network is infeasible.

4.3. The Time Complexity Analysis of Algorithm 1

Table 1 is the time complexity analysis of each step of algorithm 1. As a result, the time complexity degree [13] of algorithm 1 is $O(|N|^2 \times |A| + |\mathcal{P}| \times |A|^2)$. Therefore, algorithm 1 is effective in practice.

Table 1. The Time Complexity Analysis of Algorithm 1

Steps of algorithm 1	Computing times	Complexity degree
Step 1	$ N ^2 \times A + A ^2$	$O(N ^2 \times A + A ^2)$
Step 2	$(A - 3) \times (A - 2) / 2$	$O(A ^2)$
Step 3	$ \mathcal{P} \times A \times (A - 2)$	$O(\mathcal{P} \times A ^2)$
Step 4	$[2(A - 2) + A - 1] \times (A - 2) \times \mathcal{P} $	$O(A ^2 \times \mathcal{P})$
Step 5	$2(A - 2)^2 \times \mathcal{P} $	$O(A ^2 \times \mathcal{P})$
Step 6	$2(A - 2) \times \mathcal{P} $	$O(A \times \mathcal{P})$
Step 7	$[2(A - 2) + A - 1] \times (A - 2)$	$O(A ^2)$
Algorithm 1		$O(N ^2 \times A + \mathcal{P} \times A ^2)$

4.4. A Theorem

Theorem 3: Algorithm 1 does not identify a network infeasible if it is feasible in reality, or vice versa.

Proof: The precondition of maximizing intervals of common arcs is to guarantee each parallel subnetwork to be feasible, which ensures that the intersection of these arcs intervals in different parallel subnetwork is maximal. Therefore, updating GRPs arcs according to the intersection does not decrease the solution space of feasible schedules of the network.

According to theorem 2, algorithm 1 checks the feasibility of parallel subnetworks two times, which is the necessity condition to guarantee network feasibility. Therefore, algorithm 1 does not identify a network infeasible if it is feasible in reality.

Algorithm 1 does not identify a network feasible if it is infeasible in reality: If this happens then at least one of the four conditions (1)-(4) in line with network feasible is not satisfied. However, according to algorithm 1, A feasible network satisfies $t_j - t_i = d_{ij}, \forall (i, j) \in \mathcal{A}; t_j - t_i \in [g_{ij}^{l,up}, g_{ij}^{u,up}] \subseteq [g_{ij}^l, g_{ij}^u], \forall (i, j) \in \mathcal{P}$; and $t_0=0$. This contradicts the assumption. \square

5 An Example for Algorithm 1

We use algorithm 1 to identify the feasibility of figure 1 and all results are shown in Table 2 and 3.

Table 2. Searching Minimal Loops

Iterative order	ML _{ij}	Deleted arcs	Iterative order	ML _{ij}	Deleted arcs
①	ML _{0,13} = $\begin{matrix} 0-1-2-13 \\ 0-3-4-13 \end{matrix}$	$\begin{matrix} 0-1 \\ 0-3 \end{matrix}$	③	ML _{2,15} = $\begin{matrix} 2-6-15 \\ 2-13-15 \end{matrix}$	$\begin{matrix} 2-6 \\ 2-13 \end{matrix}$
②	ML _{1,6} = $\begin{matrix} 1-5-6 \\ 1-2-6 \end{matrix}$	$\begin{matrix} 1-5 \\ 1-2 \end{matrix}$		ML _{7,14} = $\begin{matrix} 7-13-14 \\ 2-12-14 \end{matrix}$	$\begin{matrix} 7-13 \\ 7-12 \end{matrix}$
	ML _{3,13} = $\begin{matrix} 3-4-13 \\ 3-8-7-13 \end{matrix}$	$\begin{matrix} 3-4-13 \\ 3-8 \end{matrix}$	④	ML _{6,16} = $\begin{matrix} 6-10-16 \\ 6-15-16 \end{matrix}$	$\begin{matrix} 6-10-16 \\ 6-15 \end{matrix}$
	ML _{11,12} = $\begin{matrix} 11-8-7-12 \\ 11-12 \end{matrix}$	$\begin{matrix} 11-8-7 \\ 11-12 \end{matrix}$		ML _{13,17} = $\begin{matrix} 13-15-16-17 \\ 13-14-17 \end{matrix}$	$\begin{matrix} 13-15-16-17 \\ 13-14 \end{matrix}$
③	ML _{5,10} = $\begin{matrix} 5-9-10 \\ 5-6-10 \end{matrix}$	$\begin{matrix} 5-9 \\ 5-6 \end{matrix}$		ML _{12,17} = $\begin{matrix} 12-14-17 \\ 12-17 \end{matrix}$	$\begin{matrix} 12-14-17 \\ 12-17 \end{matrix}$

Table 3. Searching Parallel Subnetworks

CA	Step 3	Feasibility of PS_{ij}	Step 5	Step 6	Feasibility of PS_{ij} with updated GPRs
(2,6)	$\{PS_{1,6}; PS_{2,15}\}$	$\sqrt{\textcircled{2}}$	{3}; [-1,3]	{3}	$\sqrt{}$
(2,13)	$\{PS_{2,15}; PS_{0,13}\}$	$\sqrt{}$	[4,8]; [4,∞]	[4,8]	$\sqrt{}$
(4,13)	$\{PS_{0,13}; PS_{3,13}\}$	$\sqrt{}$	[-∞,3]; [-7,3]	[-7,3]	$\sqrt{}$
(7,13)	$\{PS_{3,13}; PS_{7,14}\}$	$\sqrt{}$	[2,12]; [2,∞]	[2,12]	$\sqrt{}$
(7,12)	$\{PS_{7,14}; PS_{11,12}\}$	$\sqrt{}$	[6,∞]; [6,∞]	[6,∞]	$\sqrt{}$
(6,10)	$\{PS_{5,10}; PS_{6,16}\}$	$\sqrt{}$	{2}; {2}	{2}	$\sqrt{}$
(6,15)	$\{PS_{6,16}; PS_{2,15}\}$	$\sqrt{}$	{0}; [2,6]	{0}	$\sqrt{}$
(13,15)	$\{PS_{2,15}; PS_{13,17}\}$	$\sqrt{}$	[1,5]; [1,∞]	[1,5]	$\sqrt{}$
(12,14)	$\{PS_{7,14}; PS_{12,17}\}$	$PS_{12,17}^{\times}$	[2,∞]; Φ	Φ	$\sqrt{}$
(14,17)	$\{PS_{13,17}; PS_{12,17}\}$	$PS_{12,17}^{\times}$	[0,∞]; Φ	Φ	$\sqrt{}$
(12,14) [ⓐ]	$\{PS_{7,14}; PS_{12,17}\}$	$\sqrt{}$	[2,∞]; {2}	{2} [ⓐ]	$\sqrt{}$
(14,17) [ⓐ]	$\{PS_{13,17}; PS_{12,17}\}$	$\sqrt{}$	[0,∞]; {2}	{2}	$\sqrt{}$

Note: ① The computing results are obtained with updated GPRs. ② symbol “ $\sqrt{}$ ” and “ \times ” represent feasible and infeasible respectively in table 3. ③ $PS_{12,17}$ is infeasible, so we have to adjust the durations of subprojects or/and the length of GPRs arcs of $PS_{12,17}$ and make it feasible. Here, we adjust the maximal lag of arc (12,17) to 2, then re-compute the maximal intervals of $CA(12,14)$ (and $CA(14,17)$) in $PS_{12,17}$ and $PS_{7,14}$ respectively and the intersection of these maximal intervals.

The computational results manifest that algorithm 1 is effective in identifying the feasibility of a network. At the same time, all infeasible parallel subnetworks can be found. So, we can adjust the durations of subprojects or/and the length of GPRs arcs to make a feasible project schedule.

6 Conclusions

This paper puts forward the definition of parallel subnetwork and studies the methods of its identification and feasibility discrimination. And then, according to the definitions of network feasibility defined by Elmaghraby and Kamburowski [3] and Cesta et al [8], we propose distinguishing methods for network feasibility with fixed subproject durations under GPRs; study the identifying methods for subprojects and GPRs arcs which need to be adjusted when the network is infeasible. The analysis of time complexity and computational example manifests that the proposed algorithm is feasible and effective in practice.

Acknowledgments

We would like to thank the anonymous referees for their helpful suggestions.

The research was partially supported by Yun Nan HuanNeng Hydroelectric Company contracts of “Mathematical models for hydroelectric project Management and project risk management” and “Materials scheduling and dispatching system”, which is gratefully acknowledged.

The authors benefit from the discussion with Professors Xianglin Wu, Yong Zhao, Jiangwen Xiao, Yang Chen and other members participating in the projects mentioned above. The viewpoints and findings are those of writers and don't necessarily represent the views of other people mentioned above.

References

1. J. E. Kelly, Critical-path Planning and Scheduling: Mathematical Basis, *Operations Research* **9**, 296- 320 (1961).
2. D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazer, Application of a Technique for Research and Development Program Evaluation, *Operations Research* **7**(5), 646-669 (1959).
3. S. E. Elmaghraby, and J. Kamburowski, The Analysis of Activity Networks under Generalized Precedence Relations (GPRs), *Management Science* **38**(9), 1245-1263 (1992).
4. B. Roy, Graphes et ordonnancement, *Revue Francaise de Recherche Operationelle*, 323-333(1962).
5. J. A. G. M. Kerbosh and H. J. Schell, Network Planning by the Extended Metra Potential Method, Report KS-1.1, 1975, University of Technology Eindhoven, Department of Industrial Engineering.
6. B. De Reyck and W. Herroelen, A Branch-And-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations, *European Journal of Operational Research* **111**(1), 152-174 (1998).
7. S. Sakellaropoulos and A.P. Chassiakos, Project Time-cost Analysis under Generalised Precedence Relations, *Advance Engineering Software* **35**, 715-724 (2004).
8. A. Cesta, A. Oddi, and S. F. Smith, A Constraint-Based Method for Project Scheduling With Time Windows, *Journal of Heuristics*. **8**(1), 109-136 (2002).
9. B. Dodin and M. Sirvanci, Stochastic Networks and the Extreme Value Distribution, *Comput. Operations Research* **17**(4), 397-409 (1990).
10. M. K. Richard, Computational Complexity of Network Problem, *Proceedings of Symposia in Applied Mathematics* **26**, 142-155 (1982).
11. M. Bartusch, R. H. Möhring, and F. J. Radermacher, Scheduling Project Networks with Resource Constraints and Time Windows, *Annals of Operations Research* **16**, 201-240 (1988).
12. J. Zhan, Heuristics for Scheduling Resource-Constrained Projects in MPM Networks, *European Journal of Operational Research* **76**(1), 192-205 (1994).
13. K. C. Lu, *Algorithm and Complexity* (Higher Education Press, Bei Jing, 1995) [in Chinese].