

# 基于分支定界法的多资源约束下项目进度规划

陈勇强, 宋莹, 龚辰

(天津大学 管理学院, 天津 300072)

**摘要:** 基于分支定界法的思想提出了一种新的精确求解多资源约束下项目进度规划问题的最优化算法。首先依据网络图中活动的逻辑关系和资源约束进行分支构建搜索树,然后通过基于广探法思想提出的优选原则对搜索树中的分支节点进行分析比较,淘汰劣解进一步缩小搜索空间并提高求解效率,最终可获得最优解。最后通过实例进行分析表明算法在求解多资源约束条件下项目进度规划问题的有效性,同时对于资源平衡优化问题以及合理权衡项目竣工时间与资源配置两者之间关系都有一定的现实指导意义。

**关键词:** 分支定界法;资源约束;项目进度规划;资源配置

中图分类号: C934/C931.1

文献标识码: A

文章编号: 1009-3370(2009)04-0041-05

项目进度安排最常用的方法是关键路线法(CPM)和计划评审技术(PERT),而这两种方法应用的前提是假定资源供应无约束。如果某一种或几种资源供应能力受到限制,将可能导致整个项目进度的延期。如何在多资源约束的情况下合理进行项目进度规划,保证项目总工期最短,这是一类NP完全问题,求解难度相当大。文献<sup>[1]</sup>基于CPM/PERT网络提出一种资源均衡归一化的启发式优化方法。文献<sup>[2]</sup>提出了资源约束下工序新的排序准则进行网络计划优化的启发式方法。文献<sup>[3]</sup>基于枚举式分支切割法提出了新的多资源约束下施工计划的优化方法。文献<sup>[4]</sup>提出解决该类问题的一般算法,并给出具体的算法步骤。文献<sup>[5]</sup>提出一种基于模糊理论的自适应模拟退火启发算法。文献<sup>[6]</sup>基于遗传算法提出了一种启发式优化方法,并进行了相应的测试。文献<sup>[7]</sup>建立一个基于多Agent系统来解决多资源约束条件下的多项目进度的调度问题。

文献<sup>[8]</sup>提出一种双向的蚁群算法来解决此类问题。但这些方法并不能得到最优解,而只能求出满意解,而且计算结果的满意度在很大程度上受到工程项目不同的复杂度和相关特点的影响<sup>[9]</sup>。

分支定界法作为一种求解策略,在解决现实生活中最优化问题方面取得了发展。与动态规划一样,分支和定界仅是一种决策,不是算法,而决策就是必须配合具体问题的结构以形成一个可实现的解题算法<sup>[10]</sup>。本文基于分支定界法的思想并改进搜索策略,提出一种新的精确求解多资源约束下项目进度规划最优化问题的算法。首先依据数学模型中的约束条

件并结合网络图中活动的逻辑关系和资源约束进行合理分支构建搜索树,避免了组合爆炸问题;然后通过基于广探法思想提出的优选原则对搜索树中的分支节点进行分析比较,仅保留非劣可行解进行下一步分析,进一步缩小搜索空间提高求解效率,最终可获得最优解。最后通过实例进行分析表明该算法在求解多资源约束条件下项目规划问题的有效性,同时对于资源平衡优化问题以及合理权衡项目竣工时间与资源配置两者之间关系都有一定的现实指导意义。

## 一、问题的数学描述

本文所讨论的问题符合以下三个假设条件:

(1) 每项活动都遵循完成-开始的逻辑关系

对于工程项目活动通常存在四种逻辑关系:开始-开始、开始-完成、完成-开始、完成-完成,最常见的是完成-开始逻辑关系,而其他三种逻辑关系都可以通过增加虚工序的方法在网络图中转变为完成-开始的逻辑关系,所以本文统一采用完成-开始逻辑关系。同时,由于单代号网络图(AON)相对于其他网络图在表述活动逻辑关系方面更清晰明确,而且易于算法的实现,所以本文采用单代号网络图进行分析。

(2) 每项活动一旦开始就不能中断

如果存在某些活动允许中断,可以考虑把此项活动按需分成几项单独的不可再分活动进行分析。

(3) 每项活动所需的资源都小于资源的最大供应量

收稿日期: 2008-10-14

基金项目: 国家自然科学基金(70772057)

作者简介: 陈勇强(1964—),男,副教授,管理学博士。E-mail: symbolpmc@vip.sina.com

该假设表明如果实际情况不符合则问题明显无解。

以上对三个假设条件的分析表明上述假设条件不失问题的一般性。

资源约束下项目进度规划问题的整数规划数学模型可以表示如下<sup>[1]</sup>:

$$\begin{aligned} \min f_n \\ f_j - f_i \geq d_j, (i, j) \in H \\ \text{S.T.} \quad \sum_{S_i} r_{ik} \leq b_k, \quad k=1, 2, \dots, K \end{aligned}$$

其中,  $a_1, a_{12}, \dots, a_i$  表示单代号网络图中的活动代号;  $f_i$  表示活动  $a_i$  的完成时间,  $d_j$  表示活动  $a_j$  的持续时间;  $H$  表示所有符合完成-开始逻辑关系的活动集合;  $K$  表示  $K$  种不同资源,  $r_{ik}$  表示活动  $a_i$  单位时间内第  $k$  种资源的需求量,  $b_k$  表示第  $k$  种资源的限量。

### 二、算法的实现

分支定界法(branch and bound)是上世纪60年代初由 Land Doig 和 Dakin 等人提出的, 由于这方法灵活且便于用计算机求解, 现在已是解整数规划的重要方法。最优化求解过程如图1所示, 首先依据网络图中活动的逻辑关系和资源约束条件, 将可行解的集合分成数个小子集, 在搜索树上体现为由  $p$  层母节点生成  $p+1$  层子节点; 然后再根据优选规则对子节点进行分析比较, 仅保留可能产生最优解的节点进行下一步的分析, 虚框节点表示被淘汰的劣解, 不进行下一步的分支求解。此过程不断反复并更新分支节点状态直到取得最优解。

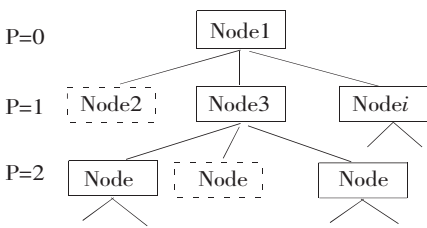


图1 求解过程概念图

#### 1. 分支过程

设节点  $i$  在  $t$  时刻状态由集合  $F_i, S_i, U_i$  表述, 分别表示已完成的集合、已安排的集合、尚未安排的集合。设集合  $E_i$  表示节点  $i$  在  $t$  时刻所对应的在集合  $U_i$  中紧前活动都已完成的所有活动集合, 先假设在无资源约束条件下, 确定由集合  $E_i$  中所有活动组成的组合方案集合  $P_i$ , 然后再分别对  $P_i$  中元素进行验证, 从中选择  $t$  时刻仅满足资源约束条件下的可行方案集合  $A_i$ , 并依此进行分支构建搜索树。

例如假设节点 2 在  $t$  时刻状态为:  $F_2=\{a_1, a_2\}, S_2=\{a_3\}, U_2=\{a_4, a_5, a_6\}$ , 其中  $U_2$  中仅活动  $a_4, a_6$  满足所有紧前活动都已完成条件, 则  $E_2=\{a_4, a_6\}$ , 由  $E_2$  中所有活动在无资源约束条件下组成的组合方案集合  $P_2=\{a_4, a_6, a_4a_6\}$ , 其中  $a_4a_6$  表示两活动同时开始, 但现有资源仅能满足其中一个活动开始实施, 故  $A_2=\{a_4, a_6\}$ , 依此可构建两个分支节点。

#### 2. 优选过程

如果对前面分支过程生成的所有可行节点无目的地继续分支, 进行完全枚举, 将大大增加搜索工作量, 降低算法效率, 而且对于求解大型复杂项目是不现实的。应根据实际问题设计优选原则对可行节点进行分析, 本文基于广探法(breadth-first search)思想提出了三种原则对可行方案集合  $A_i$  中元素进行进一步分析优选。

设  $n(S_i)$  表示集合  $S_i$  中活动的个数;  $f(S_i)$  表示集合  $S_i$  中每一个活动的计划完成时间;  $EF(S_i)$  和  $LF(S_i)$  分别表示集合  $S_i$  中最早和最晚活动的计划完成时间。  $F_i+S_i=F_j+S_j$  表示节点  $i$  和  $j$  中的已完成和已安排的活动组成的集合中元素相同, 例如:  $F_2=\{a_1, a_2\}, S_2=\{a_3\}, F_3=\{a_1\}, S_3=\{a_2, a_3\}$ , 则有  $F_2+S_2=F_3+S_3$ ; 然后可通过下面优选原则对其进行分析。

规则 1: 当  $n(S_i)=1$  时, 如果  $LF(S_i) > LF(S_j)$ , 则节点  $i$  为劣解被淘汰。例如: 在上例中  $n(S_2)=1$ , 如果  $LF(S_2)=6, LF(S_3)=5$ , 则节点 2 被淘汰。

规则 2: 当  $n(S_i) > 1$ , 且  $F_i \subseteq F_j$ , 即  $S_i=S_j+(F_j-F_i)$  时, 如果  $f(S_i) \geq f[S_j+(F_j-F_i)]$ , 即  $i$  节点  $S_i$  中每个活动的完成时间都不早于  $j$  节点, 则节点  $i$  被淘汰。例如: 在上例中  $n(S_3) > 1, F_3 \subseteq F_2$ , 即  $S_3=S_2+(F_2-F_3)=\{a_2, a_3\}$ , 如果  $f(S_3)=\{4, 4\}, f[S_2+(F_2-F_3)]=\{3, 4\}$ , 则节点 3 被淘汰。

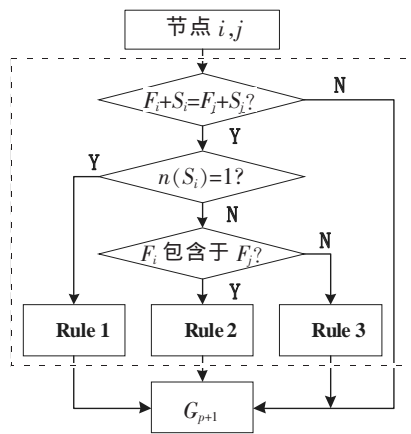


图2 优选流程图

规则 3: 当  $n(S_i) > 1$ , 且  $F_i \not\subseteq F_j$ , 如果  $f(F_i+S_i) \geq f(F_j+S_j)$ , 即  $i$  节点  $S_i$  和  $F_i$  中每个活动的完成时间都

不早于  $j$  节点, 则节点  $i$  被淘汰。例如:  $F_2=\{a_1\}, S_2=\{a_2, a_3\}, F_3=\{a_2\}, S_3=\{a_1, a_3\}$ , 即满足  $n(S_2) > 1$  且  $F_2 \not\subseteq F_3$ , 如果  $f(F_2+S_2)=f(\{a_1, a_2, a_3\})=\{2, 3, 4\}, f(F_3+S_3)=\{2, 1, 4\}$ , 则节点 2 被淘汰。

优选流程如图 2 所示, 其中虚框中表述对节点进行逻辑分析优选过程; 输入为待分析的节点; 输出  $G_{p+1}$  表示通过优选后搜索树中  $p+1$  层保留的可以继续进行求解的节点集合。

3. 算法流程

步骤 1: 初始化

对单代号网络图中的根节点进行初始化赋值, 令  $t_1=0, p=0, S_1=\{a_1\}$ 。

步骤 2: 更新搜索树  $p$  层节点  $i$  的状态

首先更新时钟  $t_i$ , 令  $t_i=EF(S_i)$ , 同时更新  $i$  节点对应的活动状态, 即把所有计划完成时间等于  $t_i$  的活动由  $S_i$  集合更新到  $F_i$  集合, 并且这些被更新的活动不再占用资源;

如果时钟  $t_i$  可以被更新, 转到下一步;

如果时钟  $t_i$  不能被更新, 输出当前状态并转到步骤 2 更新  $p$  层其它节点, 如果没有可更新节点则计算结束。

步骤 3: 确定  $E_i$

节点  $i$  更新后, 确定  $U_i$  中所有紧前活动都已完成的集合  $E_i$ ;

如果  $E_i=\emptyset$ , 转到步骤 2 更新当前节点, 否则转到下一步。

步骤 4: 确定  $A_i$

在无资源约束条件下, 确定由集合  $E_i$  中所有活动组成的组合方案集合  $P_i$ , 然后再分别对  $P_i$  中元素进行验证, 在  $P_i$  中选择  $t$  时刻仅满足资源约束条件下的可行方案集合  $A_i$ , 并依此构建分支节点;

如果  $A_i=\emptyset$ , 转到步骤 2 更新当前节点, 否则转到下一步。

步骤 5: 判断  $p$  层是否还存在尚未分支的其它节点, 存在则转到步骤 2 对搜索树  $p$  层其它节点按照上述步骤进行分支, 否则转到下一步。

步骤 6: 确定  $G_{p+1}$

按照优选原则对  $p+1$  层中的所有新构建的分支节点进行分析比较, 仅保留非劣解  $G_{p+1}$ , 则完成  $p+1$  层搜索树的优选过程并转到步骤 2 进行下一层搜索树的构建和分析。

三、案例分析

本文分别对文献[12]、[13]和[14]中的案例进行了分析并得到了精确的最优解。下面仅对文献[12]中的案例进行详细的分析, 单代号网络图如图 3 所示, 计

算过程如下:

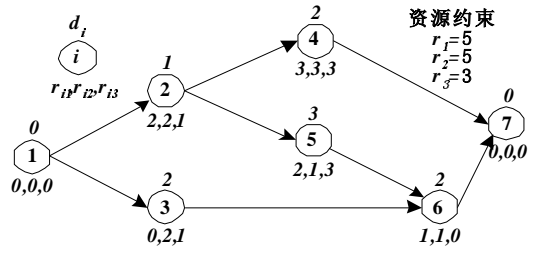


图 3 单代号网络图

进行第一层分支, 如图 4 所示:

步骤 1: 对根节点进行初始化, 令  $t_1=0, p=0, S_1=\{a_1\}$ ;

步骤 2: 更新节点 1 的状态, 由于  $a_1$  为虚活动, 所以  $t_1=EF(S_1)=0$ , 同时  $a_1$  更新为完成状态, 即  $F_1=\{a_1\}$ , 并且  $a_1$  不再占用资源;

步骤 3: 确定  $E_1$ , 由于  $a_1$  已完成, 故  $U_1$  中只有  $a_2, a_3$  满足所有紧前活动都已完成, 则  $E_1=\{a_2, a_3\}$ ;

步骤 4: 确定  $A_1$ , 在不考虑资源约束条件下由  $E_1$  确定其集合中所有活动的组合方案集合  $P_1=\{a_2, a_3, a_2a_3\}$ , 通过对  $P_1$  中元素可行性进行验证表明三种方案都满足资源约束条件, 故  $A_1=\{a_2, a_3, a_2a_3\}$ , 并依此构建分支节点 2、3、4;

步骤 5: 由于第 1 层仅有 1 个母节点, 故第 1 层分支都已完成并转到下一步;

步骤 6: 确定  $G_2$ , 当前第二层节点状态:  $F_2=\{a_1\}, S_2=\{a_2\}; F_3=\{a_1\}, S_3=\{a_3\}; F_4=\{a_1\}, S_4=\{a_2, a_3\}$ 。由于目前还没有满足  $F_i+S_i=F_j+S_j$  条件的节点进行分析比较, 故保留  $A_1$  中所有可行方案, 即  $G_2=\{a_2, a_3, a_2a_3\}$ , 同时转到步骤 2 进行下一层分析。

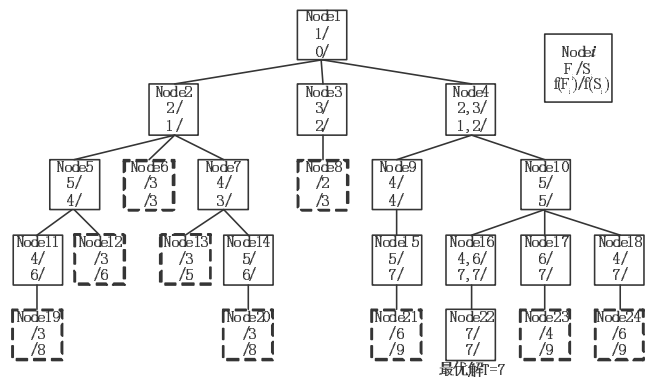


图 4 搜索树

进行第二层分支:

步骤 2: 对节点 2 进行状态更新,  $t_2=EF(S_2)=1, F_2=\{a_1, a_2\}, S_2=\emptyset$ , 同时  $a_1, a_2$  不再占用资源;

步骤 3: 由于  $F_2=\{a_1, a_2\}$ , 则  $E_2=\{a_3, a_4, a_5\}$ ;

步骤 4:  $P_2=\{a_3, a_4, a_5, a_3a_4, a_3a_5, a_4a_5, a_3a_4a_5\}$ , 由于  $P_2$

中仅有前三种方案满足资源约束,故  $A_2=\{a_3, a_4, a_5\}$ , 并依此构建分支节点 5、6、7;

步骤 5: 此时第 2 层分支过程尚未结束, 故转到步骤 2;

步骤 2: 对节点 4 进行更新,  $t_4=EF(S_4)=1, F_2=\{a_1, a_2\}, S_4=\{a_3\}$ , 同时  $a_1, a_2$  不再占用资源;

步骤 3: 由于  $F_2=\{a_1, a_2\}, S_4=\{a_3\}$ , 则  $E_2=\{a_4, a_5\}$ ;

步骤 4:  $P_4=\{a_4, a_5, a_4a_5\}$ , 由于  $S_4=\{a_3\}$ , 表示  $a_3$  正在进行并占用相应资源, 可推出  $P_4$  中三种方案都不满足资源约束, 故  $A_4=\emptyset$ , 并转到步骤 2;

步骤 2: 对当前节点 4 再次进行状态更新,  $t_4=EF(S_4)=2, F_4=\{a_1, a_2, a_3\}, S_4=\emptyset$ , 同时  $a_1, a_2, a_3$  不再占用资源;

步骤 3: 由于  $F_4=\{a_1, a_2, a_3\}, S_4=\emptyset$ , 则  $E_4=\{a_4, a_5\}$ ;

步骤 4:  $P_4=\{a_4, a_5, a_4a_5\}$ , 由于  $P_4$  中  $a_4a_5$  不满足资源约束, 故  $A_4=\{a_4, a_5\}$ , 并依此构建分支节点 9、10;

步骤 5: 此时第 2 层还有节点 3 未分析, 故转到步骤 2 对节点 3 进行分析, 由于此节点较简单, 故这里略去分析过程仅给出生成的分支节点 8;

步骤 5: 此时第二层节点分支过程已完成, 转到步骤 6;

步骤 6: 确定  $G_3$ , 由于节点 6、8 与节点 4 都满足  $F_i+S_i=\{a_1, a_2, a_3\}$ , 故可以通过优选原则进行比较分析, 其中:  $F_4=\{a_1\}, S_4=\{a_2, a_3\}; F_6=\{a_1, a_2\}, S_6=\{a_3\}; F_8=\{a_1, a_3\}, S_8=\{a_2\}$ 。对于节点 6,  $n(S_6)=1$ , 可依据规则 1 分析, 由于  $LF(S_6)=3, LF(S_4)=2, LF(S_6)>LF(S_4)$ , 故节点 6 为劣解被淘汰。对于节点 8,  $n(S_8)=1$ , 由于  $LF(S_8)=3, LF(S_4)=2, LF(S_8)>LF(S_4)$ , 故节点 8 也为劣解被淘汰。

按照上面过程继续构建搜索树进行分析求解, 直到没有可以继续更新状态的节点, 计算结束, 并从搜索过程输出的结果中可以得到最优解。

同时, 本文对于有无资源约束两种条件下的进度安排以及资源使用情况进行了对比分析。首先利用 Project2003 软件针对两种情况下的进度安排生成甘特图, 如图 5、6 所示。对于本例在资源约束下求得的最优解仅比无约束条件下按照关键路线法求得的最短工期多一天。从图 7、8 的三种资源使用量对比中我们可以看到, 在有资源约束条件下的资源使用量更趋于平缓, 而且对于本例从图 8 中我们可以分析出如果资源约束变为  $r_1=4, r_2=4, r_3=3$ , 最短工期不变, 这对于资源平衡优化问题以及合理权衡项目竣工时间与资源配置两者之间关系都有一定的现实指导意义。

同理对文献[13]和[14]中的案例按照此最优化算法进行了分析求解, 最终求解过程直接输出了最优解方案, 再一次证明了该算法的有效性。

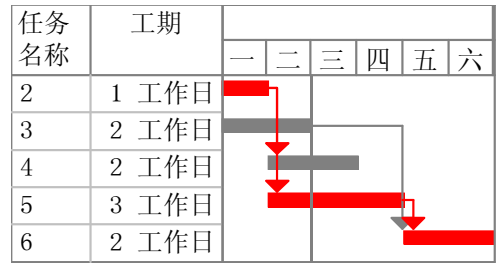


图 5 无资源约束条件下项目进度

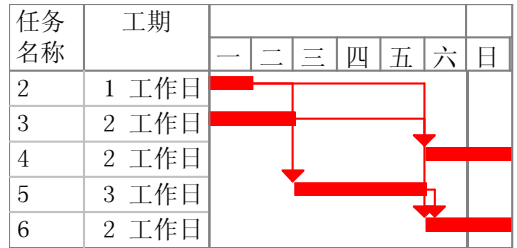


图 6 有资源约束条件下项目进度

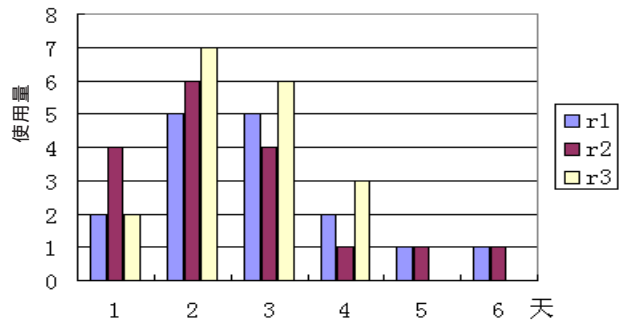


图 7 无资源约束条件下资源使用量

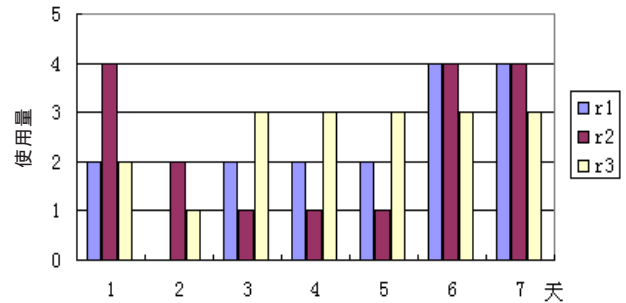


图 8 有资源约束条件下资源使用量

#### 四、结语

本文在前人研究基础上, 基于分支定界法的思想并改进搜索策略提出一种新的精确求解多资源约束下项目进度规划问题的最优化算法, 实例分析表明算法在求解多资源约束条件下项目规划最优化问题的有效性, 同时对于资源平衡优化问题以及合理权衡项目竣工时间与资源配置两者之间关系都有一定的现实指导意义。

## 参考文献:

- [1] 白思俊. 多资源约束的网络计划的启发式优化方法[J]. 系统工程理论与实践, 1995, (7): 42-47.
- [2] 杨永清, 许先云, 等. 多资源约束下的工序排序准则及其优化[J]. 系统工程理论与实践, 1995, (5): 28-31.
- [3] 蒋根谋. 多资源约束下包含资源使用顺序关系最优施工计划的确定[J]. 铁道科学与工程学报, 2005, 2(3): 56-62.
- [4] 林琳, 姚郁. 多资源约束下的多项目作业调度问题研究[J]. 哈尔滨工业大学学报, 2007, 39(7): 1045-1049.
- [5] Shukla Sanjay Kumar, Son Young Jun, M.K. Tiwari. Fuzzy-based adaptive sample-sort simulated annealing for resource-constrained project scheduling[J]. International Journal of Advanced Manufacturing Technology, 2008, 36(9-10): 982-995.
- [6] J.F. Goncalves, J.J.M. Mendes, M.G.C. Resende. A genetic algorithm for the resource constrained multi-project scheduling problem[J]. European Journal of Operational Research, 2008, 189(3): 1171-1190.
- [7] Homberger, J.. A multi-agent system for the decentralized resource-constrained multi-project scheduling problem [J]. International Transactions in Operational Research, 2007, 14(6): 565-589.
- [8] Shou, Yongyi. A bi-directional ant colony algorithm for resource constrained project scheduling [A]. IEEM 2007: 2007 IEEE International Conference on Industrial Engineering and Engineering Management[C], United States, 2007: 1027-1031.
- [9] Ozdamar, L., and Ulusou, G.. A survey on the resource-constrained project scheduling problem. IIE Trans., 1995, 27: 574-586.
- [10] ARNE THESEN. 运筹学的计算机方法[M]. 薛华成, 译. 清华大学出版社, 1986: 3.
- [11] Christofides, N., Alvarez-Valdes, R., and Tamarit, J. M.. Project scheduling with resource-constraints: A branch and bound approach [J]. European Journal of Operational Research, 1987, 29: 262-273.
- [12] Genmou Jiang, Jonathan Shi. Exact algorithm for solving project scheduling problems under multiple resource constraints [J]. Journal of Construction Engineering and Management, 2005, 131(9): 986-992.
- [13] Fortunato Crespo Abril, Concepcion Maroto Alvarez. Scheduling resource-constrained projects using branch and bound and parallel computing techniques[J]. International Journal of Operational Research, 2005, 1(1-2): 172-187.
- [14] Roland Heilmann. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags[J]. European Journal of Operational Research, 2003, 144(2): 348-365.

## Project Scheduling under Multiple Resource Constraints Based on Branch and Bound Procedure

CHEN Yong-qiang, SONG Ying, GONG Chen

(School of Management Tianjin University, Tianjin 300072)

**Abstract:** In this paper a new exact algorithm is proposed for solving project scheduling problems under multiple resource constraints which is based on the branch and bound procedure. The enumeration tree is generated by the feasible branches emanating from a parent node subject to precedence and resource constraints in the network. Then the feasible branches are evaluated by using the pruning rules based on the breadth-first search solution strategy, and the worse ones will be eliminated from the enumeration tree while the better will continue branching until they get the best solution. It is proved by examples that this new exact algorithm is efficient, providing a method for resource balance and a reasonable trade-off between resource allocation and duration.

**Key words:** branch and bound procedure; resource constraints; project scheduling; resource allocation

[责任编辑: 萧姚]