

基于合同双方交互作用的项目调度优化^①

何正文, 刘人境, 胡信布
(西安交通大学管理学院, 西安 710049)

摘要: 基于实际项目管理中, 承包商和业主的利益与行为相互交织的基本事实, 研究合同双方交互作用下的项目调度优化问题. 首先对所研究问题进行界定, 其中, 业主通过支付安排对承包商施加影响以取得一个理想的项目收益, 而承包商则通过调整活动进度对业主的安排产生反作用, 并借此追求自身项目收益的最大化. 在此基础上, 以项目净现值最大化为目标, 构建了由业主和承包商两个子模型构成的项目调度优化模型. 鉴于问题的强 NP-hard 属性, 设计专门的禁忌搜索启发式算法, 并在随机生成的标准算例集合上对算法进行测试, 结果表明该算法是求解问题的有效算法. 最后, 用算例对研究进行说明, 得到如下结论: 随着项目截止日期的延后或承包商收益底线的提高, 业主的收益呈下降趋势而承包商的收益呈上升趋势.

关键词: Max-npv 项目调度; 合同双方交互作用; 优化模型; 禁忌搜索

中图分类号: C935; F224.33 **文献标识码:** A **文章编号:** 1007-9807(2014)08-0048-12

0 引言

以净现值 NPV (net present value) 最大化作为活动进度安排目标的项目调度问题称为 Max-npv 项目调度问题^[1]. 经典的 Max-npv 项目调度问题的研究多从承包商视角出发, 很少考虑业主的作用^[2]. 然而在现实中, 业主会通过多种途径对承包商安排活动进度施加影响. 特别是在 Max-npv 项目调度问题中, 承包商现金流 (即支付) 安排的主动权便掌握在业主的手中, 因此, 忽略业主的作用无疑会影响到研究成果的应用价值.

在实际项目管理中, 合同双方存在着如下交互作用: 一方面, 业主通过设置截止日期、安排支付进度等手段, 对承包商进行有效的约束和激励, 促使承包商尽可能地按照业主的意愿实施项目, 以实现业主项目收益的最大化; 另一方面, 由于活动进度安排的主动权掌握在承包商的手中, 因此, 承包商也可以根据业主的安排调整活动进度, 从而对业主产生反作用并据此实现自身项目收益的

最大化. 上述事实表明, 承包商视角的 Max-npv 项目调度优化是仅考虑单方主体的局部优化, 而基于合同双方交互作用的 Max-npv 项目调度优化则是综合考虑双方主体的全局优化. 相比于前者来说, 后者更接近于现实情况, 因而也更具实用价值. 其结果不仅可以为承包商安排活动进度提供支持, 而且还可以为业主安排支付进度提供借鉴, 并促使双方合作协调, 取得双赢的结果.

关于 Max-npv 项目调度问题的研究始于 1970 年的 Russell^[3], 他首先采取 AoA (activity-on-arc) 项目网络表述方式, 构建了无资源约束的 Max-npv 项目调度优化模型并设计了近似迭代算法. 此后, 众多学者将注意力投入到该问题的研究中, 使其成为项目调度领域中仅次于工期最小化问题的一大分支. 1997 年, Herroelen 等^[2]对该分支 20 多年来的 70 篇文献进行了梳理, 从项目网络的表述方式、是否考虑资源约束、求解算法为最优算法还是近似算法等不同的角度对这些文献进行了分类, 初步勾勒出 Max-npv 项目调度问题的

^① 收稿日期: 2011-05-23; 修订日期: 2013-05-01.

基金项目: 国家自然科学基金资助项目 (71371150; 70971105); 新世纪优秀人才支持计划资助项目 (NCET-13-0460).

作者简介: 何正文 (1967-), 男, 山西运城人, 博士, 副教授. Email: zhengwenhe@mail.xjtu.edu.cn

研究体系. 近年来, Max-npv 项目调度问题的新的研究成果不断涌现, 例如, Jcmeli 和 Erengüç^[4] 针对资源约束问题设计了分枝定界精确算法; Padman 等^[5] 用启发式算法对问题进行求解; Etgar^[6] 研究了现金流线性依赖于时间的 Max-npv 项目调度问题; Vanhoucke 等^[7] 考虑了可更新资源的限制; Mika 等^[8] 将问题扩展到活动具有多种执行模式条件下; 而 Waligóra^[9] 用禁忌搜索算法求解了离散-连续 Max-npv 项目调度问题. 此外, 国内也有部分学者对该问题进行了研究, 例如, 柴国荣等^[10] 将柔性资源约束引入该问题中; 何正文等^[11] 分析了现金流平衡约束对调度结果的影响; 任世科等^[12] 研究了银行授信额度约束下的 Max-npv 项目调度问题. 值得注意的是, 经典的 Max-npv 项目调度问题均假定支付安排是给定的. 1997 年, Dayanand 和 Padman^[13] 首先放松了这一假设条件, 将支付安排视为决策变量, 提出了所谓的项目支付调度 (project payment scheduling) 问题. 项目支付调度问题可视为 Max-npv 项目调度问题的一个最新分支, 关于该问题, Ulusoy 和 Cebeli^[14] 设计了双环路遗传算法以获取平等的支付进度安排; Dayanand 和 Padman^[15] 针对问题的特点, 开发了两阶段模拟退火启发式算法; Dayanand 和 Padman^[16] 还从业主的视角构建了项目支付调度的优化模型, 并给出相应的求解算法; Szmereskovsky^[17] 基于承包商对业主的影响, 构建了项目支付调度模型并给出分枝定界算法; He 和 Xu^[18] 发现奖惩机制可以改善支付安排的柔性并提高双方的收益; Kavlak 等^[19] 定义了谈判能力指数, 将 Ulusoy 和 Cebeli^[14] 的成果扩展到更为一般的情形下; He 等^[20] 用模拟退火与禁忌搜索算法求解了多模式项目支付调度问题; 文献 [21] 针对资金约束多模式项目支付调度问题开发了多种启发式算法; 文献 [22] 从双方联合的视角出发, 提出了基于融资费用分担的项目支付调度模型. 最后需要特别强调的是, 尽管关于 Max-npv 项目调度问题的研究已经产生了不少成果, 但迄今尚未发现对合同双方交互作用做过深入研究.

本文鉴于上述理论和实际现状, 研究基于合同双方交互作用的 Max-npv 项目调度问题.

1 问题界定

本文采用基于事件的研究方法 (event-based

method), 即项目网络采用 AoA 方式表述, 正负现金流均与项目事件相联系. 考虑一个具有 N 个活动、 M 个事件的项目, 活动 n ($n = 1, 2, \dots, N$) 具有 Q_n 种执行模式, 以模式 q ($q = 1, 2, \dots, Q_n$) 执行时的工期和费用分别为 d_{nq} 和 c_{nq} , 实现事件 m ($m = 1, 2, \dots, M$) 所需费用为 e_m .

$$e_m = \zeta \sum_{n \in SS_m} c_{nq} + (1 - \zeta) \sum_{n \in ES_m} c_{nq}$$

其中, SS_m 为从事件 m 开始活动的集合; ES_m 为到事件 m 结束活动的集合; ζ ($0 \leq \zeta \leq 1$) 为活动费用在其开始和结束事件之间的分配比例. 活动 n 的挣值为 w_n , 事件 m 的挣值为 v_m ($v_m = \sum_{n \in ES_m} w_n$),

项目合同总价格为 U ($U = \sum_{m=1}^M v_m$). 现金流的折现率为 α , 项目截止日期为 D .

由于基于合同双方交互作用的 Max-npv 项目调度问题包含承包商和业主两个主体, 因此, 该问题实质上由如下两个子问题构成:

1) 业主子问题 将项目在未来可为业主带来的收益折现到项目结束时刻, 认为业主在项目结束时刻有较大的预期收益 ER . 在项目实施过程中, 业主对承包商进行 K ($K \leq M$) 次支付, 支付量等于支付时承包商的累计挣值与支付比例 θ ($0 \leq \theta \leq 1$) 的乘积, 项目结束时支付总量必须等于项目合同总价格 U . 业主有权在 M 个事件上随意安排这 K 次支付, 但最后一次支付必须安排在项目终事件 M 上. 业主所要面对的问题是, 如何在事件上安排这些支付以使得自身的项目净现值最大化.

2) 承包商子问题 对于该项目来说, 承包商有一个收益底线 $NPV_{\text{cont}}^{\text{low}}$, 亦即业主所给出的支付事件安排必须确保承包商的项目净现值不低于 $NPV_{\text{cont}}^{\text{low}}$; 否则, 承包商有权要求业主对支付事件安排进行调整, 直至达到其收益底线 $NPV_{\text{cont}}^{\text{low}}$ 为止. 在确保项目完成时间不晚于截止日期 D 的前提下, 承包商有权随意安排活动执行模式和事件发生时间. 他所要面对的问题是, 在业主所给定的支付事件安排下, 如何安排活动执行模式和事件发生时间以使得自身的项目净现值最大化.

从以上对两个子问题的界定可见, 合同双方的利益和行为是相互交织在一起的: 首先, 业主需要在承包商安排活动执行模式和事件发生

时间的基础上,才可对其支付事件安排进行优化,以便取得自身项目净现值的最大化;而反过来,承包商又必须在业主给出支付事件安排之后,才能以自身项目净现值最大化为目标,对活动执行模式和事件发生时间的安排进行优化.

$$\begin{aligned}
 x_{km} &= \begin{cases} 1 & \text{第 } k \text{ 次支付在事件 } m \text{ 上进行} \\ 0 & \text{其他} \end{cases} & k = 1, 2, \dots, K; m = 1, 2, \dots, M \\
 y_{nq} &= \begin{cases} 1 & \text{活动 } n \text{ 采用执行模式 } q \\ 0 & \text{其他} \end{cases} & n = 1, 2, \dots, N; q = 1, 2, \dots, Q_n \\
 z_{mt} &= \begin{cases} 1 & \text{事件 } m \text{ 在 } t \text{ 时刻发生} \\ 0 & \text{其他} \end{cases} & m = 1, 2, \dots, M; t = 1, 2, \dots, D
 \end{aligned}$$

为了随后的表述需要,在3组决策变量的基础上进一步定义如下3个决策向量

$$\begin{aligned}
 X &= (x_{km}, m = 1, 2, \dots, M); \\
 Y &= (q: y_{nq} = 1, n = 1, 2, \dots, N); \\
 Z &= (t: z_{mt} = 1, m = 1, 2, \dots, M)
 \end{aligned}$$

将业主和承包商的项目收入净现值分别记为 NPV_{client} 和 NPV_{cont} ,那么,本文所研究的问题最终可界定如下:承包商基于业主给出的 X ,对 Y 和 Z 进行优化以使 NPV_{cont} 最大化;业主在承包商优化 Y 和 Z 的基础上,对 X 进行优化以使 NPV_{client} 最大化;通过二者之间的循环迭代,获得合同双方均满意的 X^* 、 Y^* 和 Z^* .

2 优化模型

在对问题进行界定的基础上,构建由如下两个子模型构成的基于合同双方交互作用的 Max-npv 项目调度优化模型.

1. 业主子模型

$$\max NPV_{client} = ER \sum_{t=E_M}^{L_M} [\exp(-\alpha t) z_{Mt}] - \sum_{k=1}^K \left\{ p_k \sum_{m=1}^M \left[x_{km} \sum_{t=E_m}^{L_m} (\exp(-\alpha t) z_{mt}) \right] \right\} \quad (1)$$

$$\text{s. t. } \sum_{m=1}^{M-1} x_{km} = 1, \quad k = 1, 2, \dots, K-1 \quad (2)$$

$$x_{KM} = 1 \quad (3)$$

$$\sum_{k=1}^K x_{km} \leq 1, \quad m = 1, 2, \dots, M \quad (4)$$

$$p_k = \theta \left[\sum_{m=1}^M \left(v_m \sum_{t=0}^{T_k} z_{mt} \right) - \sum_{m=1}^M \left(v_m \sum_{t=0}^{T_{k-1}} z_{mt} \right) \right]$$

由此可见,要取得 Max-npv 项目调度的全局优化,必须对两个子问题进行循环求解并反复迭代才可实现.

根据上述对两个子问题的讨论,定义如下3组决策变量

$$k = 1, 2, \dots, K-1 \quad (5)$$

$$P_K = U - \sum_{k=1}^{K-1} P_k \quad (6)$$

$$NPV_{cont} \geq NPV_{cont}^{low} \quad (7)$$

$$x_{km} \in \{0, 1\} \quad (8)$$

式中 $T_k \left(T_k = \sum_{t=0}^D \left[t \sum_{m=1}^M (x_{km} z_{mt}) \right] \right)$ 和 $T_{k-1} \left(T_{k-1} = \sum_{t=0}^D \left[t \sum_{m=1}^M (x_{k-1, m} z_{mt}) \right] \right)$ 分别是第 k 及 $k-1$ 次支付的时间; NPV_{cont} 由式(9)给出.

在业主子模型中,目标要求式(1)为最大化业主的项目净现值;约束条件式(2)将支付安排到事件 M 上;式(3)确保最后一次支付安排在项目终事件 M 上;式(4)限定了在某一给定事件上,最多只能安排一次支付;式(5)计算前 $K-1$ 次支付的支付量;式(6)使得所有支付量的总和等于项目合同总价格;式(7)保证承包商的项目净现值不低于其收益底线;式(8)为决策变量的定义域约束.注意,在该子模型中 z_{mt} 为输入变量而非决策变量.

2. 承包商子模型

$$\max NPV_{cont} = \sum_{k=1}^K \left\{ p_k \sum_{m=1}^M \left[x_{km} \sum_{t=E_m}^{L_m} (\exp(-\alpha t) z_{mt}) \right] \right\} - \sum_{m=1}^M \left\{ e_m \sum_{t=E_m}^{L_m} [\exp(-\alpha t) z_{mt}] \right\} \quad (9)$$

$$\text{s. t. } \sum_{q=1}^{Q_n} y_{nq} = 1, \quad n = 1, 2, \dots, N \quad (10)$$

$$e_m = \zeta \sum_{n \in SS_m} \sum_{q=1}^{Q_n} (c_{nq} y_{nq}) + (1 - \zeta) \times \sum_{n \in ES_m} \sum_{q=1}^{Q_n} (c_{nq} y_{nq}), \quad m = 1, 2, \dots, M \quad (11)$$

$$\sum_{t=E_m}^{L_m} z_{mt} = 1 \quad m = 1, 2, \dots, M \quad (12)$$

$$\sum_{t=E_{g_n}}^{L_{g_n}} (z_{g_n t}) + \sum_{q=1}^{Q_n} (d_{nq} y_{nq}) \leq \sum_{t=E_{h_n}}^{L_{h_n}} (z_{h_n t}) \quad n = 1, 2, \dots, N \quad (13)$$

$$\sum_{t=E_m}^{L_m} (z_{Mt} t) \leq D \quad (14)$$

$$y_{nq}, z_{mt} \in \{0, 1\} \quad (15)$$

式中 g_n 和 h_n 分别表示活动 n 的开始和结束事件; p_k ($k = 1, 2, \dots, K$) 由式 (5) 和 (6) 计算.

在承包商子模型中, 目标要求式 (9) 最大化承包商的项目净现值 NPV_{cont} ; 约束条件式 (10) 为每个活动确定一种执行模式; 式 (11) 基于所选择的执行模式计算各事件的费用; 式 (12) 为每个事件在其时间窗内安排一个发生时间; 式 (13) 确保各事件的发生时间遵守网络优先关系的限制; 式 (14) 使得终事件 M 的发生时间 (亦即项目的完成时间) 不晚于项目截止日期; 式 (15) 为决策变量的定义域约束. 注意, 在该子模型中, x_{km} 为输入变量而非决策变量.

3 禁忌搜索启发式算法

不失一般性, 令 $\alpha = 0, K = 1$ 则本文所研究

问题的承包商子问题可简化为离散时间 / 费用权衡问题的工期问题^[23], 而后者已被证明为一个强 NP-hard 问题^[24]. 因此, 前者也必然为一强 NP-hard 问题, 从而导致本文所研究问题也为一个强 NP-hard 问题. 鉴于上述原因, 此处采用禁忌搜索启发式算法求解该问题. 之所以选择禁忌搜索, 是因为该种算法已被众多学者应用于项目调度问题, 且被证明为求解此类问题的最有效算法之一^[8-9, 25-31].

3.1 算法的总体思路

对上述优化模型进行分析可以发现, 业主子模型和承包商子模型是相互嵌套在一起的: 一方面, 承包商子模型需要在业主子模型给定 X 之后才能求解; 另一方面, 在求解业主子模型时, 也需要承包商子模型输出的 Z 和 NPV_{cont} . 基于两个子模型之间的上述关系, 制定如图 1 所示的算法流程. 由图 1 可见, 在求解过程中, 首先生成一个初始可行的 X ; 然后依次求解承包商子模型和业主子模型, 对两个子模型进行反复交叉的迭代循环运算, 直至满足算法的终止条件为止; 最后再求解一次承包商子模型, 输出得到的最终结果. 由于承包商子模型的求解实质上是内嵌在业主子模型的求解中的, 所以, 下面首先给出承包商子模型的求解步骤, 然后再说明业主子模型的求解过程.

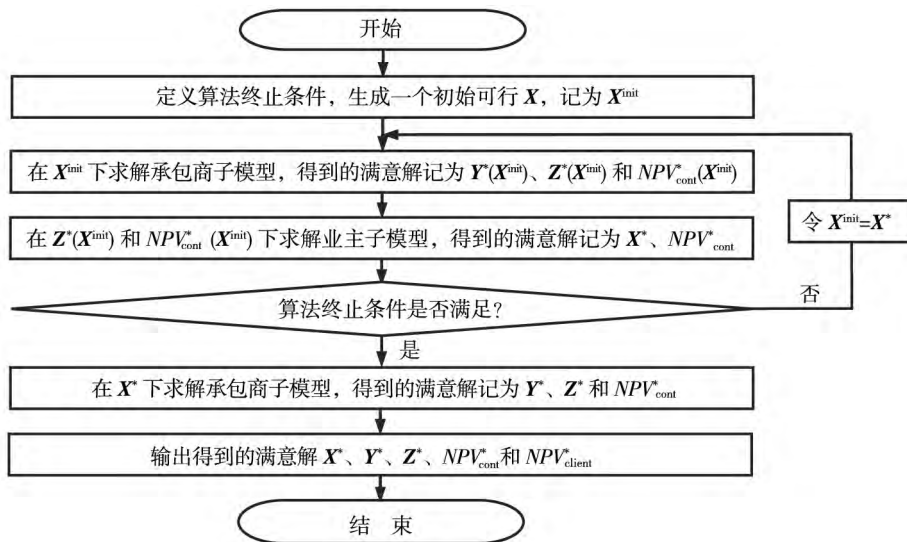


图 1 算法流程图

Fig. 1 Flow chart of algorithms

3.2 承包商子模型求解步骤

1) 初始可行解构造 承包商子问题的初始

可行解 Y^{init}, Z^{init} 通过下述步骤构造:

步骤 1 随机地为每个活动安排一种执行模

式,由此得到一个 Y^{init} . 计算每个事件的时间窗,如果项目终事件 M 的最早发生时间不晚于项目截止日期 D ,则接受 Y^{init} 为一可行的 Y^{init} ;否则,重复该操作直至得到一个可行的 Y^{init} 为止.

步骤 2 在不违反网络优先关系约束的前提下,为每个事件在其时间窗中随机地安排一个发生时间,由此得到一个 Z^{init} . 如果项目终事件 M 的发生时间不晚于项目截止日期 D ,则接受 Z^{init} 为可行的 Z^{init} ;否则,重复该操作直至得到可行的 Z^{init} 为止.

2) 邻点生成机理 当前可行解 Y^{curr} 、 Z^{curr} 的邻点 Y^{neig} 、 Z^{neig} 通过如下两个算子随机生成:

①Y算子 在 Y^{curr} 中 随机地选择一个活动,将其当前执行模式随机地变成另外一种执行模式,从而将 Y^{curr} 转化为它的一个邻点 Y^{neig} . 计算各事件在 Y^{neig} 下的时间窗,如果项目终事件 M 的最早发生时间不晚于项目截止日期 D ,则接受 Y^{neig} 为可行的邻点;否则,重复上述操作直至得到可行的 Y^{neig} 为止. 在不违反网络优先关系和项目截止日期约束的前提下,将 Z^{curr} 中各事件的发生时间调整到由 Y^{neig} 所决定的新时间窗中,从而将 Z^{curr} 同时变换为 Z^{neig} .

②Z算子 在 Z^{curr} 中,从除项目开始事件之外的其余事件中随机地选择一个事件,将其发生时间在其时间窗内随机地变化一个单位,调整其他事件的发生时间以确保网络优先关系约束得到满足,由此得到 Z^{curr} 的一个邻点 Z^{neig} . 如果 Z^{neig} 中的项目终事件 M 的发生时间不晚于项目截止日期 D ,则接受 Z^{neig} 为可行邻点;否则,重复上述操作直至获得可行的 Z^{neig} 为止. 注意,在该算子的操作过程中, Y^{curr} 保持不变,即将 Y^{curr} 直接记为 Y^{neig} .

3) 禁忌列表 在算法搜索过程中,新生成邻点移动的逆向移动被加入到禁忌列表中,从而避免搜索返回刚刚探测过的可行解. 禁忌列表的长度设置为 20,采用“先进先出 FIFO(first-in-first-out)”的原则进行更新:每当生成当前解的可行邻点时,该移动的逆向移动从底部加入到禁忌列表中. 与此同时,最早进入列表的逆向移动从顶部移出列表,列表中其余逆向移动向上递进一位. 所有位于禁忌列表中的逆向移动都是被禁止的,但当一个被

禁止的逆向移动能够生成比当前最好解还要好的邻点时,那么它的禁忌状态可以被激活:即将其从禁忌列表中移出,其下所有逆向移动向上递进一位,同时将该逆向移动的逆向移动加入到禁忌列表的底部.

4) 求解步骤 基于上述讨论,即可给出承包商子模型的具体求解步骤. 由于在承包商子模型中,向量 X 为输入变量,因此,将该求解步骤记为 $Cont(X)$:

步骤 1 生成初始可行解 Y^{init} 、 Z^{init} 并计算其目标函数值 NPV_{cont}^{init} . 定义算法终止条件,即探测可行解总数 Num_{cont}^{stop} (设置为 $N \cdot 1000$);初始化禁忌列表 TL_{cont} ;令计数器 $Num_{cont} = 0$;将当前解及当前最好解赋值为初始解: $Y^{curr} = Y^{best} = Y^{init}$, $Z^{curr} = Z^{best} = Z^{init}$, $NPV_{cont}^{curr} = NPV_{cont}^{best} = NPV_{cont}^{init}$. 其中 NPV_{cont}^{curr} 为当前解所对应的目标函数值, Y^{best} 、 Z^{best} 和 NPV_{cont}^{best} 为当前最好解及其目标函数值.

步骤 2 从 Y 算子和 Z 算子中随机地选出一个算子,生成当前解 Y^{curr} 、 Z^{curr} 的一个邻点 Y^{neig} 、 Z^{neig} ,计算其目标函数值 NPV_{cont}^{neig} . 检查生成邻点操作的移动是否位于禁忌列表 TL_{cont} 中,若是转步骤 4;否则,转步骤 3.

步骤 3 令 $Y^{curr} = Y^{neig}$, $Z^{curr} = Z^{neig}$, $NPV_{cont}^{curr} = NPV_{cont}^{neig}$, $Num_{cont} = Num_{cont} + 1$,更新禁忌列表 TL_{cont} . 如果 $NPV_{cont}^{curr} > NPV_{cont}^{best}$,进一步令 $Y^{best} = Y^{curr}$, $Z^{best} = Z^{curr}$, $NPV_{cont}^{best} = NPV_{cont}^{curr}$,转步骤 4;否则,直接转步骤 5.

步骤 4 如果 $NPV_{cont}^{neig} > NPV_{cont}^{best}$,激活生成邻点移动的禁忌状态,令 $Y^{curr} = Y^{best} = Y^{neig}$, $Z^{curr} = Z^{best} = Z^{neig}$, $NPV_{cont}^{curr} = NPV_{cont}^{best} = NPV_{cont}^{neig}$, $Num_{cont} = Num_{cont} + 1$,更新禁忌列表 TL_{cont} ,转步骤 5;否则,转步骤 2.

步骤 5 判断 $Num_{cont} \geq Num_{cont}^{stop}$ 是否成立,若成立转步骤 6;否则,转步骤 2.

步骤 6 输出得到的满意解 $Y^*(X) = Y^{best}$, $Z^*(X) = Z^{best}$, $NPV_{cont}^*(X) = NPV_{cont}^{best}$.

3.3 业主子模型求解步骤

在承包商子模型求解步骤 $Cont(X)$ 的基础上,可给出业主子模型的禁忌搜索求解步骤:

步骤 1 生成初始可行解 X^{init} ,调用

$\text{Cont}(X^{\text{init}})$, 得到 $Y^*(X^{\text{init}})$ 和 $Z^*(X^{\text{init}})$, 计算在 $X^{\text{init}}, Y^*(X^{\text{init}}), Z^*(X^{\text{init}})$ 下的业主子模型目标函数值, 记为 $NPV_{\text{client}}^{\text{init}}$. 定义算法终止条件, 即探测可行解总数 $Num_{\text{client}}^{\text{stop}}$ (设置为 $M \cdot 100$); 初始化禁忌列表 TL_{client} ; 令计数器 $Num_{\text{client}} = 0$; 将当前解及当前最好解赋值为初始解: $X^{\text{curr}} = X^{\text{best}} = X^{\text{init}}$, $NPV_{\text{client}}^{\text{curr}} = NPV_{\text{client}}^{\text{best}} = NPV_{\text{client}}^{\text{init}}$. 其中 $NPV_{\text{client}}^{\text{curr}}$ 为当前解所对应的目标函数值, X^{best} 和 $NPV_{\text{client}}^{\text{best}}$ 为当前最好解及其目标函数值.

步骤 2 随机生成当前解 X^{curr} 的一个可行邻点 X^{neig} , 调用 $\text{Cont}(X^{\text{neig}})$, 得到 $Y^*(X^{\text{neig}})$ 和 $Z^*(X^{\text{neig}})$, 计算在 $X^{\text{neig}}, Y^*(X^{\text{neig}}), Z^*(X^{\text{neig}})$ 下的业主子模型目标函数值, 记为 $NPV_{\text{client}}^{\text{neig}}$. 检查生成邻点操作的移动是否位于禁忌列表 TL_{client} 中, 若是转步骤 4; 否则, 转步骤 3.

步骤 3 令 $X^{\text{curr}} = X^{\text{neig}}$, $NPV_{\text{client}}^{\text{curr}} = NPV_{\text{client}}^{\text{neig}}$, $Num_{\text{client}} = Num_{\text{client}} + 1$, 更新禁忌列表 TL_{client} . 如果 $NPV_{\text{client}}^{\text{curr}} > NPV_{\text{client}}^{\text{best}}$, 进一步令 $X^{\text{best}} = X^{\text{curr}}$, $NPV_{\text{client}}^{\text{best}} = NPV_{\text{client}}^{\text{curr}}$, 转步骤 5; 否则, 直接转步骤 5.

步骤 4 如果 $NPV_{\text{client}}^{\text{neig}} > NPV_{\text{client}}^{\text{best}}$, 激活生成邻点移动的禁忌状态, 令 $X^{\text{curr}} = X^{\text{best}} = X^{\text{neig}}$, $NPV_{\text{client}}^{\text{curr}} = NPV_{\text{client}}^{\text{best}} = NPV_{\text{client}}^{\text{neig}}$, $Num_{\text{client}} = Num_{\text{client}} + 1$, 更新禁忌列表 TL_{client} , 转步骤 5; 否则, 转步骤 2.

步骤 5 判断 $Num_{\text{client}} \geq Num_{\text{client}}^{\text{stop}}$ 是否成立, 若成立转步骤 6; 否则, 转步骤 2.

步骤 6 输出得到的满意解 $X^* = X^{\text{best}}$, $NPV_{\text{client}}^* = NPV_{\text{client}}^{\text{best}}$, 以及在 X^* 下的承包商子模型的满意解及对应的目标函数值: $Y^*(X^*) = Y^*(X^{\text{best}})$, $Z^*(X^*) = Z^*(X^{\text{best}})$, $NPV_{\text{cont}}^*(X^*) = NPV_{\text{cont}}^*(X^{\text{best}})$.

在上述搜索步骤中, 初始可行解 X^{init} 按以下方法生成: 从除项目终事件 M 之外的其余事件中随机地选择 $K - 1$ 个事件, 将向量 X 中与所选事件及终事件 M 所对应的元素设置为 1, 其他的元素设置为 0, 由此得到一个 X^{init} ; 调用 $\text{Cont}(X^{\text{init}})$, 得到 $NPV_{\text{cont}}^*(X^{\text{init}})$, 如果 $NPV_{\text{cont}}^*(X^{\text{init}}) \geq NPV_{\text{cont}}^{\text{low}}$, 则接受 X^{init} 为一可行的初始解; 否则, 拒绝 X^{init} 并重复上述操作直至获得可行的初始解为止. 当前解 X^{curr} 的可行邻点

X^{neig} 通过下述 X 算子生成:

X 算子 在 X^{curr} 中, 除最后一个元素外, 随机地选出一个取值为 1 的元素, 将其与另一随机选出的取值为 0 的元素交换位置, 从而把 X^{curr} 转化为它的一个邻点 X^{neig} ; 调用 $\text{Cont}(X^{\text{neig}})$, 得到 $NPV_{\text{cont}}^{\text{low}}(X^{\text{neig}})$, 如果 $NPV_{\text{cont}}^*(X^{\text{neig}}) \geq NPV_{\text{cont}}^{\text{low}}$, 则接受 X^{neig} 为一可行的邻点; 否则, 拒绝 X^{neig} 并重复上述操作直至获得一个可行的邻点为止.

禁忌列表 TL_{client} 的长度设置为 10, 管理方式与承包商子模型求解步骤中的相同.

4 算法测试

由于本文所研究问题之前并无学者进行过深入研究, 因此没有现成的算法可为此处的算法测试提供比较基准. 借鉴 Mika 等^[8, 27]、Waligóra^[9]、He 等^[20-21] 在评价禁忌搜索算法时的已有作法, 作者也选用多重迭代改进 MSII (multi-start iterative improvement) 和随机生成 RG (random generation) 两个算法作为对比. MSII 算法从一个与禁忌搜索相同的初始可行解开始搜索, 使用同样的邻点生成机理, 在搜索过程中选择改进最为显著的邻点进行迭代; 如果不存在可以改进的邻点, 则算法从另外一个随机生成的可行解重新启动再进行搜索, 当已探测的可行解达到规定的数目时, 算法终止并将保存的最好解输出为满意解. RG 算法随机地生成规定数目的可行解, 从中选出最好解作为问题的满意解. 为使 3 个算法具有相同的比较基准, 在测试中令其使用同样的终止条件, 亦即对于承包商和业主两个子问题, 当 MSII 和 RG 算法探测的可行解数目分别达到 $Num_{\text{cont}}^{\text{stop}}$ 和 $Num_{\text{client}}^{\text{stop}}$ 时, 算法终止并输出满意解.

算法测试在由 ProGen^[32] 随机生成的标准算例上进行, ProGen 的参数设置见表 1. 表中各参数的选择简要解释如下: “在某一非虚活动数下生成的算例数”、“算例的起始和终止活动数”、“最大紧前和紧后活动数”和“活动的执行模式数”这 4 个参数的取值, 是按照项目调度问题标准算例的构造惯例来设定的. “活动在执行模式 1 下的工期 d_{n1} ”、“活动在执行模式 1 下的费用 c_{n1} ”、“活

动在执行模式 2 下的工期 d_{n2} ”、“活动在执行模式 2 下的费用 c_{n2} ”和“活动挣值 w_n ”这 5 个参数的取值 均在某一范围内随机抽取以确保算例活动参数的随机性,同时 d_{n2} 不大于 d_{n1} 且 c_{n2} 不小于 c_{n1} , 保证了在算例生成过程中不会产生无效的“活动费用分配比例 ζ ”、“折现率 α ”、“支付次数 K ”、“支付比例 θ ”和“业主的预期收益 ER ”这 5 个参数,主要是从基本符合现实情形的角度出发来设定其取值的. 最后,需要特别说明的是,“项目非虚活动数 N ”、“承包商收益底线 NPV_{cont}^{low} ”和“项目截止日期 D ”是要进行全因素试验的 3 个关键参数,所以,将它们分别设定为不同的值以分析这种变化对计算结果的影响. N 的取值为 4 种,其余 2 个参数取值为 3 种,在每种取值组合下生成的算例数为 10 个,由此获得 $10 \times 4 \times 3 \times 3 = 360$ 个算例.

定义如下 4 个指标以评价算法的绩效:
 $AR_1(\%)$ 对于承包商子问题来说,满意解距离已知最好解的平均相对偏差;
 $MR_1(\%)$ 对于承包商子问题来说,满意解距离已知最好解的最大相对偏差;
 $AR_2(\%)$ 对于业主子问题来说,满意解距离已知最好解的平均相对偏差;
 $MR_2(\%)$ 对于业主子问题来说,满意解距离已知最好解的最大相对偏差;
 $AT(s)$ 算法的平均计算时间;
 $MT(s)$ 算法的最大计算时间.
 其中,已知最好解是指在 3 个算法所获得的满意解中,相应子问题目标函数值最大的一个(注意,对于两个子问题来说,已知最好解有可能不同). 算法采用 Visual Basic 6.0 编程,在 CPU 主频为 1.6GHz、内存为 256MB 的个人计算机上运行.

表 1 ProGen 的参数设置
 Table 1 Parameter setting of ProGen

参数	取值
算例的非虚活动数 N	分别设置为 10、20、30、40
在某一非虚活动数下生成的算例数	10
算例的起始和终止活动数	从 2、3 和 4 中随机选取
最大紧前和紧后活动数	4
活动的执行模式数	2
活动在执行模式 1 下的工期 d_{n1}	从区间 [1, 10] 中随机选取
活动在执行模式 1 下的费用 c_{n1}	从区间 [10, 20] 中随机选取
活动在执行模式 2 下的工期 d_{n2}	$\rho_1 \cdot d_{n1}$, 其中 ρ_1 从区间 [0.8, 1] 中随机选取
活动在执行模式 2 下的费用 c_{n2}	$\rho_2 \cdot c_{n1}$, 其中 ρ_2 从区间 [1, 1.2] 中随机选取
活动挣值 w_n	$\rho_3 \cdot c_{n2}$, 其中 ρ_3 从区间 [1.3, 1.5] 中随机选取
活动费用分配比例 ζ	0.5
折现率 α	0.01
支付次数 K	从 3、4、5 中随机选取
支付比例 θ	从 0.7、0.8、0.9 中随机选取
业主的预期收益 ER	$\rho_4 \cdot \sum_{n=1}^N w_n$, 其中 ρ_4 从 1.6、1.8、2.0 中随机选取
项目截止日期 D	$\rho_5 \cdot (CL_{max} - CL_{min}) + CL_{min}$, 其中 CL_{max} 和 CL_{min} 分别为所有活动在执行模式 1 和 2 下的项目网络关键路径的长度 ρ_5 分别设置为 0.4、0.6 和 0.8
承包商收益底线 NPV_{cont}^{low}	$\rho_6 \cdot [(\sum_{n=1}^N w_n - \sum_{n=1}^N c_{n2}) \exp(-\alpha D)]$, 其中 ρ_6 分别设置为 0.8、1.0 和 1.2

算法的测试结果见表 2. 首先从获得的满意解的质量看, 禁忌搜索启发式算法的 AR_1 、 MR_1 、 AR_2 和 MR_2 均小于 MSII 和 RG 算法的相应指标, 而且当问题规模(用项目活动数 N 表示)增大时, 禁忌搜索以上两个指标均呈下降趋势, 而其他两种算法却呈上升趋势, 表明禁忌搜索获得的满意解的质量不仅优于其他两种算法, 而且这种优势随着问题规模的增大而扩大; 其次从计算时间看, 禁忌搜索算法的 AT 和 MT 都大于其他两种算法的相应指标, 说明禁忌搜索虽然获得的满意解质量较高, 但其在相同终止条件下计算时间较长. 这一结果可以解释如下: 本文所设计的禁忌搜索算法基于两个子模型的嵌套关系, 通过反复交叉的迭代循环运算寻找问题的满意解, 较好地体现了本文所研究问题的特点; 而且, 禁忌搜索本身是一种智能优化算法, 通过对禁忌列表的检查、判断和更新, 可以有效地避免算法重复探测同一可行解, 由

此改善其搜索效率, 所以禁忌搜索获得的满意解的质量较高. MSII 算法仅选择可以改进的邻点进行迭代, 且不对已探测过的可行解进行检查判断, 因而易陷于局部最优并存在重复探测相同可行解的缺陷, 所以其得到的满意解的质量较差. RG 算法则更为简单, 它甚至不对邻域进行搜索, 仅从随机生成的可行解中挑出最好解为满意解, 因此该算法获得的满意解的质量最差. 上述分析表明, 虽然禁忌搜索启发式算法的计算时间较长, 但对于本文所研究问题来说这种代价是值得的, 与其他两种算法相比, 它可以有效地提高获得的满意解的质量; 而且对于所测试的全部算例来说, 禁忌搜索启发式算法的计算时间没有超过 123.01 s, 仍处于可以接受的范围内. 所以可以认为, 本文所设计的禁忌搜索启发式算法, 是求解基于合同双方交互作用的 Max-npv 项目调度问题的有效算法.

表 2 算法测试结果

Table 2 Test results of algorithms

N	禁忌搜索启发式算法						MSII 算法						RG 算法					
	AR_1	MR_1	AR_2	MR_2	AT	MT	AR_1	MR_1	AR_2	MR_2	AT	MT	AR_1	MR_1	AR_2	MR_2	AT	MT
10	0.35	1.14	0.14	0.66	0.56	1.37	0.67	1.59	0.93	2.22	0.33	0.87	2.12	5.37	3.32	5.97	0.14	0.27
20	0.17	0.56	0.05	0.21	5.87	9.11	1.93	4.35	2.83	4.37	2.44	6.16	7.76	13.26	7.96	14.16	1.32	2.71
30	0.09	0.32	0.01	0.03	33.44	54.62	5.80	11.67	6.27	13.91	17.25	25.67	17.21	31.44	19.46	33.30	9.28	15.09
40	0.03	0.10	0.00	0.01	86.03	123.01	13.16	22.32	15.06	26.12	56.83	84.43	36.87	54.23	38.67	58.33	23.60	35.78

5 示 例

用图 2 所示的随机生成的算例对本文的研究进行说明. 各活动的挣值及在不同执行模式下的工期和费用见表 3, 项目的其他参数如下: $\zeta = 0.5$, $\theta = 0.8$, $K = 5$, $\alpha = 0.01$, $ER = 768$, $D = 54$, $NPV_{cont}^{low} = 65$. 利用本文所开发的禁忌搜索启发式算法, 可以求得该项目的满意安排如下:

$$X = (0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1);$$

$$Y = (2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1,$$

$$2, 1, 2, 2);$$

$$Z = (0, 5, 9, 18, 11, 14, 27, 13, 23, 19, 33, 37, 41, 51, 54);$$

$$NPV_{client} = 154.36; NPV_{cont} = 66.39.$$

上述结果表明, 通过合同双方的交互作用, 业主将支付安排在事件 2、9、10、12 和 15 上; 而承包商在该支付事件安排下, 选择活动 1、9、12、17、19、20 采用执行模式 2, 其余活动采用执行模式 1, 整个项目在截止日期 54 完成. 最终, 业主获得了满意的项目收益 154.36, 承包商的收益也超过了其收益底线 65, 达到 66.39.

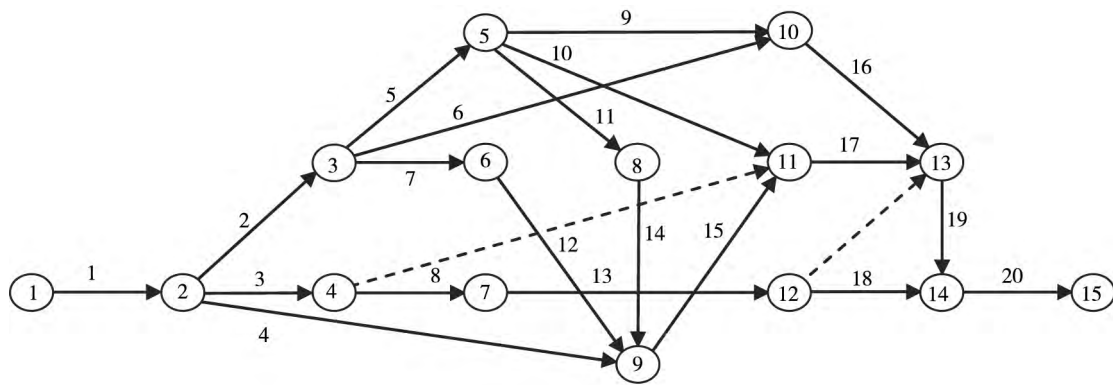


图2 项目网络图

Fig. 2 Project network

表3 活动的挣值及其在不同执行模式下的工期和费用

Table 3 Activities' earned values and their durations and costs under different payment modes

活动编号	挣值	执行模式 1		执行模式 2		活动编号	挣值	执行模式 1		执行模式 2	
		工期	费用	工期	费用			工期	费用	工期	费用
1	16	6	11	5	12	11	17	2	11	2	12
2	24	4	14	4	16	12	13	11	8	9	9
3	23	9	14	8	16	13	18	8	13	7	14
4	28	4	18	3	22	14	22	10	16	9	17
5	17	2	12	2	13	15	25	10	17	8	19
6	29	8	20	7	23	16	19	6	13	5	14
7	27	5	19	4	22	17	25	10	16	8	19
8	11	9	7	8	8	18	26	13	17	10	19
9	20	10	13	8	15	19	27	12	18	10	20
10	22	12	14	10	17	20	17	4	11	3	12

此外,为了分析项目截止日期 D 和承包商收益底线 NPV_{cont}^{low} 对双方交互作用及最终项目收益的影响,在其他参数保持不变的前提下,令这两个参数取不同的值,分别求解优化模型,得到的结果见表4。依据表4绘制的合同双方项目收益随两个参数的变化曲线见图3。图3表明,随着项目截止日期的延后,业主的收益呈下降趋势而承包商的收益则逐步上升;而当承包商的收益底线提高时,承包商的收益随之提高,但业主的收益却随之下降。上述现象的原因解释如下:项目截止日期是业主对承包商安排活动进度施加的限制,当该日期延后时,这一限制变得宽松,承包商可以选择更多的活动采用费用较低而工期较长的执行模式1,从而导致其总费用降低而使项目收益提高;而对于业主来说,由于其在项目结束时有一个较大的预期

收益 ER 截止日期的延后必然导致项目完成时间的延后,这会造成业主获得 ER 的时间迟,进而使得其项目收益下降。反过来,收益底线 NPV_{cont}^{low} 是承包商对业主安排支付事件施加的限制,该底线上时,承包商的收益自然会向着改善的方向变化,而业主的收益则会由于安排支付事件的自由度减小而下降。需要特别指出的是,当项目截止日期 D 变为50时,算法无法找到承包商子问题的可行解,意味着业主要求承包商完成项目的时间过于严苛,使得承包商即使选择所有的活动都采取执行模式2(此时项目活动网络关键路径的长度为51)也无法在规定的截止日期50完成项目;相似地,当承包商的收益底线提高至71时,算法无法找到业主子问题的可行解,表明此时承包商对项目收益要求太高,业主无论怎样安排支付事件也无法达到承包商的要求。

表 4 关键参数对项目收益的影响

Table 4 Effects of key parameters on project profits

D	NPV_{client}	NPV_{cont}	NPV_{cont}^{low}	NPV_{client}	NPV_{cont}
50	-	-	63	155.37	63.37
51	158.08	65.05	64	154.69	64.82
52	156.69	65.70	65	154.36	66.39
53	155.10	65.87	66	152.61	68.41
54	154.36	66.39	67	148.54	68.49
55	149.44	66.51	68	148.54	68.49
56	145.67	67.75	69	146.86	69.07
57	142.39	68.00	70	146.49	70.32
58	142.83	68.41	71	-	-

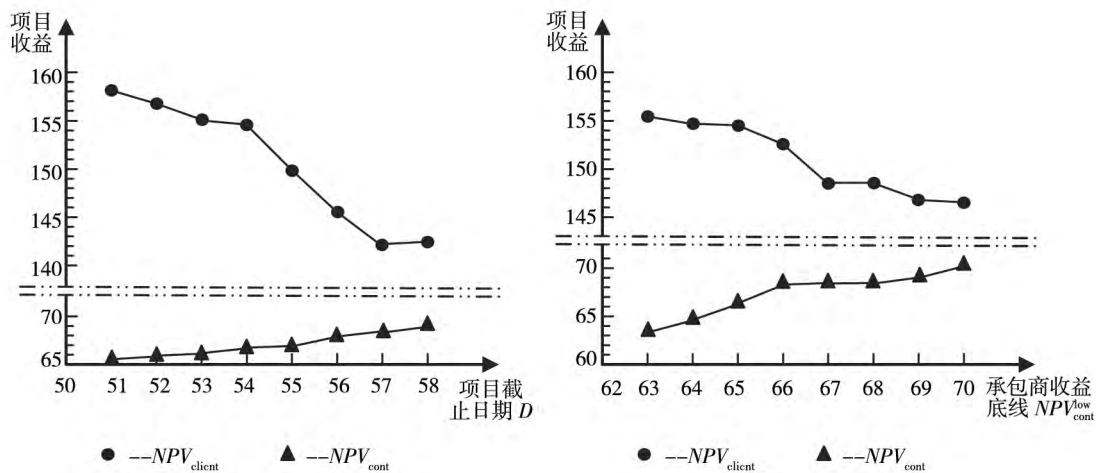


图 3 合同双方项目收益随关键参数的变化曲线

Fig. 3 Variation curves of two parties' project profit with key parameters

6 结束语

本文从项目管理的现实情形出发,研究合同双方交互作用下的 Max-npv 项目调度问题. 首先对所研究问题进行了界定,其中包含业主和承包商两个子问题,业主借助支付事件的安排对承包商施加影响以实现自身项目收益的最大化,而承包商则通过选择活动执行模式和事件发生时间对业主的安排产生反作用,并借此使得自身项目收益最大化. 随后,构建由两个子模型构成的 Max-npv 项目调度优化模型,通过对两个子模型的反复求解及循环迭代,可以确定出合同双方均满意的项目支付及活动进度安排. 鉴于问题的强 NP-hard 属性,设计了两个子模型反复交叉的迭代循环运算的禁忌搜索启发式算法. 以 MSII 和 RG 两个算法为对比基准,在随机生成的标准算例

集合上对算法进行了测试,结果表明:尽管计算时间较长,但禁忌搜索启发式算法获得的满意解的质量要远好于其他两个算法,因而可以认为是求解本文所研究问题的有效算法. 最后用一个算例对研究进行了说明,分析了项目截止日期和承包商收益底线对合同双方交互作用及最终项目收益的影响,得到如下结论:随着项目截止日期的延后或承包商收益底线的提高,业主的收益呈下降趋势而承包商的收益则呈上升趋势. 本文的研究可以为承包商和业主关系的协调提供支持,有助于合同双方取得双赢的结局. 最后需要指出的是,由于本文所研究问题尚无精确算法,所以算法的测试是通过与另外两种启发式算法的比较完成的,这样的评价无疑具有局限性. 为解决这一局限性,精确算法的设计将会是该问题后续研究的重点.

参考文献:

- [1]何正文,徐渝,朱少英. Max-npv项目进度问题研究评述[J]. 管理工程学报,2005,19(4): 60-63.
He Zhengwen, Xu Yu, Zhu Shaoying. A survey of max-npv project scheduling problems [J]. Journal of Industrial Engineering and Engineering Management, 2005, 19(4): 60-63. (in Chinese)
- [2]Herroelen W, Dommelen V, Demeulemeester E. Project network models with discounted cash flows: A guided tour through recent developments [J]. European Journal of Operational Research, 1997, 100(1): 97-121.
- [3]Russell A H. Cash flows in networks [J]. Management Science, 1970, 16(5): 357-373.
- [4]Icmeli O, Erengüç S S. A branch-and-bound procedure for the resource-constrained project scheduling problem with discounted cash flows [J]. Management Science, 1996, 42(10): 1395-1408.
- [5]Padman R, Smith-Daniels D E, Smith-Daniels V L. Heuristic scheduling of resource-constrained projects with cash flows [J]. Naval Research Logistics, 1997, 44(4): 365-381.
- [6]Etgar R. Scheduling project activities to maximize the net present value the case of linear time-dependent cash flows [J]. International Journal of Production Research, 1999, 37(2): 329-339.
- [7]Vanhoucke M, Demeulemeester E, Herroelen W. On maximizing the net present value of a project under renewable resource constraints [J]. Management Science, 2001, 47(8): 1113-1121.
- [8]Mika M, Waligóra G, Wezglarz J. Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models [J]. European Journal of Operational Research, 2005, 164(3): 639-668.
- [9]Waligóra G. Discrete-continuous project scheduling with discounted cash flows—A tabu search approach [J]. Computers and Operations Research, 2008, 35(7): 2141-2153.
- [10]柴国荣,徐渝,张静文,等. 多模式柔性资源约束型折现流项目进度问题研究[J]. 系统工程,2005,23(2): 14-18.
Chai Guorong, Xu Yu, Zhang Jingwen, et al. A study on multi-mode flexible resource constrained project scheduling with discounted cash flow [J]. Systems Engineering, 2005, 23(2): 14-18. (in Chinese)
- [11]何正文,刘入境,胡信布,等. 现金流平衡约束下的 Max-npv项目调度[J]. 系统工程理论与实践,2009,29(3): 132-141.
He Zhengwen, Liu Renjing, Hu Xinbu, et al. Study on Max-npv project scheduling problems with cash flow balance constraints [J]. Systems Engineering-Theory & Practice, 2009, 29(3): 132-141. (in Chinese)
- [12]任世科,何正文,徐渝. 基于银行授信额度的 Max-npv项目调度问题研究[J]. 管理工程学报,2009,23(2): 85-91.
Ren Shike, He Zhengwen, Xu Yu. Study on bank credit limit based Max-npv project scheduling problems [J]. Journal of Industrial Engineering and Engineering Management, 2009, 23(2): 85-91. (in Chinese)
- [13]Dayanand N, Padman R. On modeling progress payments in project networks [J]. Journal of the Operational Research Society, 1997, 48(9): 906-918.
- [14]Ulusoy G, Cebelli S. An equitable approach to the payment scheduling problem in project management [J]. European Journal of Operational Research, 2000, 127(2): 262-278.
- [15]Dayanand N, Padman R. A two stage search heuristic for scheduling payments in projects [J]. Annals of Operation Research, 2001, 102(1): 197-220.
- [16]Dayanand N, Padman R. Project contracts and payments schedules: The client's problem [J]. Management Science, 2001, 47(12): 1654-1667.
- [17]Szmerekovsky J G. The impact of contractor behavior on the client's payment-scheduling problem [J]. Management Science, 2005, 51(4): 629-640.
- [18]He Z, Xu Y. Multi-mode project payment scheduling problems with bonus-penalty structure [J]. European Journal of Operational Research, 2008, 189(3): 1191-1207.
- [19]Kavlak N, Ulusoy G, Şerifoglu F S, et al. Client-contractor bargaining on net present value in project scheduling with limited resources [J]. Naval Research Logistics, 2009, 56(2): 93-112.
- [20]He Z, Wang N, Jia T, et al. Simulated annealing and tabu search for multi-mode project payment scheduling [J]. European Journal of Operational Research, 2009, 198(3): 688-696.
- [21]He Z, Liu R, Jia T. Metaheuristics for multi-mode capital-constrained project payment scheduling [J]. European Journal of Operations Research, 2012, 223(3): 605-613.

- [22] He Z, Wang N, Li P. Simulated annealing for financing cost distribution based project payment scheduling from a joint perspective [J]. *Annals of Operations Research*, 2012, published online, DOI 10.1007/s10479-012-1155-9.
- [23] De P, Dunne E J, Ghosh J B, et al. The discrete time-cost tradeoff problem revisited [J]. *European Journal of Operational Research*, 1995, 81(2): 225-238.
- [24] De P, Dunne E J, Ghosh J B, et al. Complexity of the discrete time-cost tradeoff problem for project networks [J]. *Operations Research*, 1997, 45(2): 302-306.
- [25] Icmeli O, Erengüç S S. A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows [J]. *Computers and Operations Research*, 1994, 21(8): 841-853.
- [26] Tsai Y W, Gemmill D D. Using tabu search to schedule activities of stochastic resource-constrained projects [J]. *European Journal of Operational Research*, 1998, 111(1): 129-141.
- [27] Mika M, Waligóra G, Węglarz J. Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times [J]. *European Journal of Operational Research*, 2008, 187(3): 1238-1250.
- [28] Lambrechts O, Demeulemeester E, Herroelen W. A tabu search procedure for developing robust predictive project schedules [J]. *International Journal of Production Economics*, 2008, 111(2): 493-508.
- [29] 何正文, 刘入境, 徐渝. 基于不同支付条件的现金流均衡项目调度优化 [J]. *管理科学学报*, 2011, 14(8): 75-85.
He Zhengwen, Liu Renjing, Xu Yu. Study on Max-npv project scheduling problems with cash flow balance constraints [J]. *Journal of Management Sciences in China*, 2011, 14(8): 75-85. (in Chinese)
- [30] 胡祥培, 孙丽君, 王雅楠. 物流配送系统干扰管理模型研究 [J]. *管理科学学报*, 2011, 14(1): 50-60.
Hu Xiangpei, Sun Lijun, Wang Yanan. A model for disruption management in urban distribution systems [J]. *Journal of Management Sciences in China*, 2011, 14(1): 50-60. (in Chinese)
- [31] 李健, 达庆利, 何瑞银. 多车次同时集散货物路线问题研究 [J]. *管理科学学报*, 2010, 13(10): 75-85.
Li Jian, Da Qingli, He Ruiyin. Simultaneous deliveries and pickups [J]. *Journal of Management Sciences in China*, 2010, 13(10): 75-85. (in Chinese)
- [32] Kolisch R, Sprecher A. PSPLIB-A project scheduling problem library [J]. *European Journal of Operational Research*, 1996, 96(1): 205-216.

Project scheduling optimization based on interaction between two parties of contracts

HE Zheng-wen, LIU Ren-jing, HU Xin-bu

School of Management, Xi'an Jiaotong University, Xi'an 710049, China

Abstract: Based on the fact that, in the actual project management, actions of the contractor and the client are interacted with each other and their profits are related closely, this paper investigates the project scheduling problem considering the interaction between the two parties of a contract. The authors identify the studied problem firstly, where the client exerts an effect on the contractor through the payment arrangement so as to obtain a desirable profit, while the contractor reacts to the client by adjusting activity schedule with the objective of maximizing his own profit. Then, taking the maximization of the project's net present value as the objective, the authors construct a project scheduling optimization model which is composed of two submodels, namely, the contractor and the client submodel. For the strong NP-hardness of the problem, a special tabu search heuristic algorithm is developed. The algorithm is tested on a standard instance set generated randomly and the results show that the algorithm is an efficient algorithm for the problem. Ultimately, an example is utilized to illustrate the research and the following conclusions are drawn: With the postponement of the project deadline or the augment of the contractor's profit baseline, the profit of the client tends to decrease while that of the contractor tends to increase.

Key words: Max-npv project scheduling; interaction between two parties of contract; optimization model; tabu search