

# 资源转移视角下的 RCPSP 鲁棒资源分配方法

胡雪君<sup>1</sup>, 王建江<sup>2\*</sup>, 崔南方<sup>3</sup>, 黄浩<sup>4</sup>

(1. 湖南大学工商管理学院, 湖南长沙 410082; 2. 国防科技大学系统工程学院, 湖南长沙 410073;  
3. 华中科技大学管理学院, 湖北武汉 430074; 4. 中国移动互联网有限公司, 广东广州 510600)

**摘要:** 为使项目在工期不确定环境下既能按计划稳定执行又能维持较低的成本, 以项目鲁棒性和资源转移成本为优化对象, 构建了一个鲁棒资源分配优化模型. 引入一种开始时间关键度指标作为项目的解鲁棒性目标, 不同于已有研究均采用基于活动的资源流描述, 模型定义了基于资源的二元决策变量, 以表示某一资源单元在项目活动之间的转移次序. 结合遗传算法和模拟退火算法的优点, 提出了遗传退火混合智能算法对模型求解, 模拟实验结果证明了所提算法在寻优效果和收敛速度方面的优越性. 最后通过真实项目案例, 进一步验证了模型和算法的实用性与有效性.

**关键词:** 资源受限项目调度问题; 资源流网络; 鲁棒性; 资源转移成本; 混合智能算法

中图分类号: F224 文献标识码: A 文章编号: 1000-5781(2020)02-0173-15

doi: 10.13383/j.cnki.jse.2020.02.004

## Robust resource allocation method for the RCPSP from a resource transferring perspective

Hu Xuejun<sup>1</sup>, Wang Jianjiang<sup>2\*</sup>, Cui Nanfang<sup>3</sup>, Huang Hao<sup>4</sup>

(1. Business School, Hunan University, Changsha 410082, China;  
2. College of Systems Engineering, National University of Defense Technology, Changsha 410073, China;  
3. School of Management, Huazhong University of Science and Technology, Wuhan 430074, China;  
4. China Mobile Internet Company Limited, Guangzhou 510600, China)

**Abstract:** In order to achieve schedule stability of a project plan and to maintain comparatively low project costs under activity duration uncertainty, this paper proposes a robust resource allocation optimization model considering both the robustness and resource transfer cost objectives. First, the model adopts a starting time criticality index as a surrogate measure of solution robustness. Second, different from the existing activity-based resource-flow formulations, the model uses resource-oriented binary decision variables to indicate the sequence of activities that will use a particular resource unit. Then, a hybrid intelligent algorithm (GSA) is developed for solving the model, which combines the merits of the genetic algorithm and the simulated annealing algorithm. The results of a simulation experiment demonstrate the superiority of the GSA algorithm in terms of optimization ability and convergence rate. Finally, this article provides a case study of a real project. The comparison results further validate the practicability and effectiveness of the proposed model and algorithm.

**Key words:** resource-constrained project scheduling problem; resource flow network; robustness; resource transfer cost; hybrid intelligent algorithm

收稿日期: 2018-04-06; 修订日期: 2018-11-15.

基金项目: 国家自然科学基金资助项目(71701067; 71801218; 71572010); 湖南省自然科学基金资助项目(2019JJ50039); 国防科技大学科研计划资助项目(ZK18-03-16).

\*通信作者

## 1 引言

资源受限项目调度问题(resource-constrained project scheduling problem, RCPSP)是在活动工序约束及(可更新)资源约束的条件下,制定出满足项目特定目标(比如完工时间最短)的基准调度计划.传统 RCPSP 研究大多考虑的是确定的模型参数.由于当前市场环境快速变化,复杂项目(如“一带一路”跨国项目、PPP 项目、复杂研发项目等)所面临的风险和不确定性不断增大,例如员工缺勤、机器故障、活动执行时间偏离预期、物料延迟到达、恶劣天气影响开工以及设计变更等<sup>[1]</sup>.这些不可控因素会干扰项目的顺利执行,造成基准计划的指导价值降低甚至变得不可行,而必要的调整和修复又会产生大量的额外成本<sup>[2]</sup>.同时,现代项目本身的规模和结构越来越复杂,对稀缺资源的竞争逐渐加剧,导致越来越多的项目面临进度延误和成本超支的风险.因此,如何得到稳健的项目调度计划与合理的资源分配方案,是企业管理者 and 项目从业者所面临的艰巨挑战.

鲁棒性项目调度作为解决不确定条件下项目调度问题的有效方法,受到了学者们的广泛关注<sup>[3-5]</sup>.该方法在充分挖掘和利用不确定性信息的基础上,旨在生成“受到保护的”、抗干扰能力强的基准调度方案,以有效应对项目执行中的突发事件.学术界将项目调度的鲁棒性分为质鲁棒性和解鲁棒性两种:质鲁棒性(quality robustness)是指基准调度计划对应的目标函数值(比如项目工期、项目准时完工率、项目净现值等)对干扰因素的不敏感性,又称为完工鲁棒性;解鲁棒性(solution robustness)是指基准调度计划与项目实际执行时进度计划之间的差别大小,也称之为计划鲁棒性/稳定性<sup>[6-9]</sup>.可更新资源分配是制定鲁棒性基准调度计划的有效方法之一.该方法旨在为调度计划确定资源传递路线(资源流),它描述了可更新资源在完成一个项目活动后向另一个活动转移所形成的路径,所有资源的转移路径交织在一起形成项目资源流网络(resource flow network).通常情况下,同一个项目调度计划可能存在不同的资源分配方案,对应不同的资源流网络,而资源流显然会影响到项目计划的鲁棒性,因此在生成项目调度计划的同时优化资源分配具有重要意义<sup>[10]</sup>.

Artigues 等<sup>[11]</sup>探讨了资源分配对基准计划稳定性的影响,采用并行进度生成机制构建可行的资源流网络. Leus 等<sup>[12]</sup>, Leus<sup>[13]</sup>提出保护基准计划不受活动工期变动干扰的鲁棒资源分配模型,采用分支定界算法求解单资源小规模问题,其目标函数是最小化活动实际开始时刻偏离计划开始时刻的惩罚成本. Policella<sup>[14]</sup>提出两阶段算法,利用满足资源和紧前关系的基准调度计划,通过“链接”的方式生成稳定的资源流网络,文章将流动性和灵活性作为项目的鲁棒评价指标. Deblaere 等<sup>[15]</sup>考虑活动工期的不确定性,提出了三种基于整数规划的启发式算法(MinEA, MaxPF 和 MinED)构建资源流网络,提升调度计划的鲁棒性.这三个算法有不同目标函数:MinEA 最小化由于资源分配引起的额外弧的数量;MaxPF 最大化项目活动间成对出现的自由时差总和;MinED 最小化项目活动的实际开始时间与计划开始时间之间偏差的期望值.随着问题规模的增大,三个算法的计算量呈指数增长,算法效率大大降低.为此,Deblaere 等<sup>[15]</sup>又提出一种 MABO(myopic activity-based optimization)算法,通过局部寻优的方式构建资源流网络,并设计模拟实验对比以上算法的绩效,结果表明 MABO 算法构建的调度计划鲁棒性最强.崔南方等<sup>[2]</sup>进一步改进了 MABO 算法,采用一种 EPC(expected penalty cost)指标评估可行资源分配方案的鲁棒性,实验证明该方法保证了资源流网络的唯一性,并且大大降低了算法计算量.

以上针对不确定型资源分配问题的研究大都以项目计划鲁棒性最优为目标,对于调度成本的控制却鲜有探讨.在项目实践中,可更新资源在项目活动间的转移是存在一定转移成本的,比如制造行业中的机器加工不同的生产材料要考虑相应的转产成本,电网维修人员奔赴不同地点抢修电网要考虑交通成本等.降低项目成本对于企业增加利润、提高效益具有重要意义,因此需要通过研究资源流网络,获得最佳资源分配方案,最小化项目调度成本. Krüger 等<sup>[16]</sup>在确定型多项目调度问题中考虑了资源转移时间,提出改进的基于优

先规则的串行和并行进度生成机制来确定资源转移路径. Krüger 等<sup>[17]</sup> 引入资源转移成本, 针对多种不同类型的资源转移, 提出混合整数规划模型, 目标分别是最小化多项目总工期和最小化多项目成本(包括资源转移成本、资源闲置成本及项目延期成本). 此外, Poppenborg 等<sup>[18]</sup> 以项目活动间的资源转移数量作为决策变量, 采用禁忌搜索算法求解带有资源转移时间的 RCPSP 问题, 通过对当前资源弧进行重构或者选择其逆流来定义问题解的邻域空间.

可以看出, 针对资源分配问题, 以上两方面研究都是独立开展的, 在不确定环境下仅关注鲁棒性目标, 在确定条件下仅考虑工期/成本目标. 本文引入活动工期不确定性和资源转移成本, 采用一种开始时间关键度指标作为项目的解鲁棒性目标, 在资源转移视角下提出一种新的基于资源链的流网络模型, 寻求工期不确定环境下稳定性最强、成本最优的资源分配方案. 结合遗传算法(GA)和模拟退火算法(SA)的优点, 设计了性能和稳定性更优的遗传退火混合智能算法(GSA)对模型求解, 提出了适应问题特征的基于个体差异度的选择策略、交叉策略及非法解递归修复策略. 最后, 通过实际案例研究和仿真实验, 验证了所提模型和算法的有效性与优越性. 本文工作在理论上延伸了鲁棒性项目调度的问题领域, 也进一步丰富了资源流网络研究的内容, 扩充了混合智能算法在项目调度领域的应用. 在实践方面, 由于考虑了活动工期不确定性、项目计划鲁棒性、活动间资源转移关系及成本, 使得所研究的问题更契合复杂工程项目实际, 研究成果有望应用于企业新产品开发、软件开发、面向订单的生产、工程建设以及维修服务等项目管理实践中.

## 2 基于资源链的鲁棒资源分配优化模型

为了便于理解, 本节首先基于一个案例对所研究的问题进行描述, 给出了文中用到的参数及其定义; 然后建立数学优化模型, 从决策变量、目标函数和约束条件三方面对模型进行具体表述.

### 2.1 问题描述

采用节点式网络  $G = (N, A)$  表示一个项目, 其中  $N = \{0, 1, \dots, n + 1\}$  代表项目活动节点集合, 序号 0 和  $n + 1$  分别表示虚拟首活动和虚拟尾活动,  $A$  代表活动之间的结束-开始型工序优先关系集合. 其它相关参数见表 1.

表 1 参数说明  
Table 1 Description of parameters

参数	取值
$n$	实体(non-dummy)活动的数量
$K$	资源类别集合
$R_k$	第 $k$ 种资源的单位时间可用量, $k \in K$
$d_j$	活动 $j$ 的期望工期(计划工期), $j \in N$
$d_j^R$	项目执行时活动 $j$ 的实际工期(随机变量), $j \in N$
$s_j$	活动 $j$ 的计划开工时间, $j \in N$
$s_j^R$	项目执行时活动 $j$ 的实际开工时间, $j \in N$
$r_{jk}$	活动 $j$ 对资源 $k$ 的单位时间资源需求量, $j \in N, k \in K$
$f(i, j, k)$	活动 $i$ 流向活动 $j$ 的第 $k$ 种资源的数量, $i, j \in N, k \in K$
$e_{kl}$	资源 $k$ 的第 $l$ 个资源单元, $k \in K, l = 1, 2, \dots, R_k$
$c_k^{ij}$	资源 $k$ 在活动 $i$ 与活动 $j$ 之间的单位转移成本, $i, j \in N, k \in K$

图 1 给出了一个项目网络图实例, 该项目包含 9 个实体活动的, 所需单资源的可用量为 10 个单位. 以资源流为研究对象的项目调度问题通常是先通过求解 RCPSP, 获得工序及资源可行的最短工期计划, 得到各活动计划开工时间, 图 2 是采用分支定界法<sup>[19]</sup> 生成的工期最短的基准调度计划. 在基准计划与原始网络  $G = (N, A)$  的基础上, 在具有资源转移关系的两个活动之间加入资源弧  $A_R$ , 就构成了集成资源流的项目网络图  $G' = (N, A \cup A_R)$ , 项目实际执行时同时受到紧前紧后关系网络  $G = (N, A)$  和资源流网络  $G_R = (N, A_R)$  的约束.

需要注意的是,针对同样一个项目调度计划,通常存在多种不同的资源分配方案,不同方案对应不同的资源转移关系.例如,图 3 和图 4 给出了图 2 最短工期计划下资源流网络不同的两种资源分配方案,其中实箭线表示活动间的原始工艺关系约束,虚箭线描述了额外资源弧形成的新工序约束.

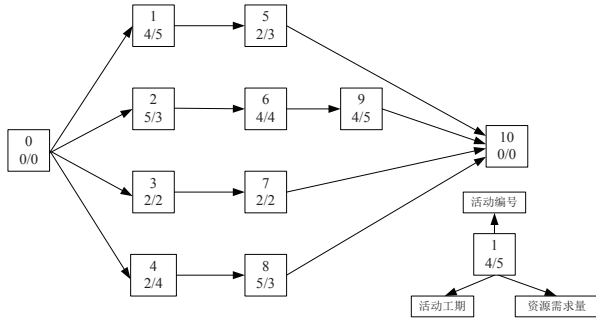


图 1 项目网络图示例

Fig. 1 An example project network

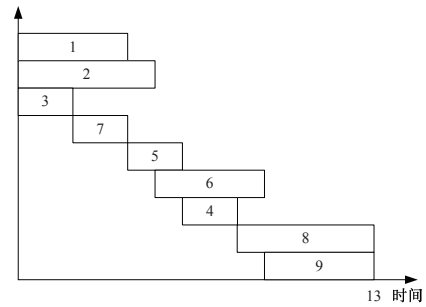


图 2 工期最短的基准调度计划

Fig. 2 A baseline schedule with minimum makespan

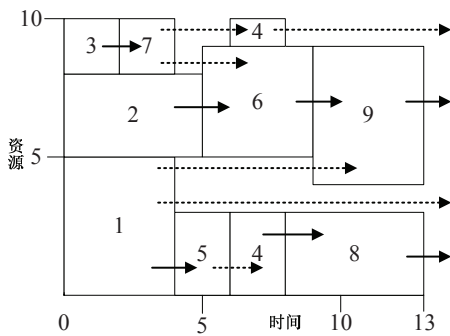


图 3 资源流网络 I

Fig. 3 Resource flow network I

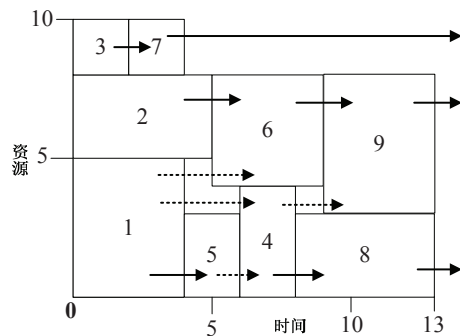


图 4 资源流网络 II

Fig. 4 Resource flow network II

分析图 3 和图 4 可知,一方面,资源流网络的优劣会影响项目调度计划的鲁棒性,不同的资源分配方案在同等程度干扰下的总体延迟效应会呈现较大的差异.比如,资源分配方案 I 中活动 9 所需要的资源由活动 6 和 1 提供,而在方案 II 中则是由活动 6 和 4 提供.此外,方案 I 中额外资源弧数量为 6,而方案 II 中额外资源弧数量为 4,因此方案 II 的资源流网络通常更稳定<sup>[2,15]</sup>.另一方面,资源流直接决定了调度方案的资源转移成本,尤其是在多项目环境下,资源在不同项目之间以及项目内部活动之间的调拨都存在一定的转移成本,同一个活动所需资源的来源通常会有多种选择,此时就需要考虑资源转移成本择优选取.

综上所述,本文试图以项目鲁棒性和资源转移成本为优化对象构建项目调度的资源流网络优化模型.

## 2.2 模型构建

### 决策变量

以往文献大多采用基于活动的资源流模型,即以活动之间的资源转移数量  $f(i, j, k)$  为决策变量.本文以每一资源单元是否在两个活动之间发生转移作为决策变量(0-1决策变量),提出一种基于资源链的流网络模型.该模型不仅能得到活动间资源转移的数量,还可以明确具体转移的是哪些资源单元.决策变量定义为

$$y_{kl}^{ij} = \begin{cases} 1, & \text{如果资源单元 } e_{kl} \text{ 从活动 } i \text{ 流向活动 } j \\ 0, & \text{否则.} \end{cases}$$

### 目标函数

Leus 等<sup>[12]</sup>提出用各项目活动实际开始时刻偏离计划开始时刻所产生的惩罚成本(stability cost, SC) 来

衡量项目调度计划的解鲁棒性, 表示为

$$SC = \sum_{j \in N} w_j E [|s_j - s_j^R|], \quad (1)$$

其中权重  $w_j$  表示活动  $j$  实际开始时刻偏离计划开始时刻的边际成本(比如因计划变更而产生的各种管理费用和协调费用、进度误工的惩罚成本等<sup>[20]</sup>).

由于期望值难以获得, 通常采用蒙特卡洛仿真进行模拟得到 SC 值. 但是, 模拟方法通常需要耗费大量的计算资源和时间, 并且不能在不同环境下保持一致性<sup>[10,11]</sup>. 因此, 为了更有效地衡量调度计划的解鲁棒性, 有必要定义新的解鲁棒性替代指标.

### 1) 解鲁棒性目标

van de Vonder 等<sup>[22]</sup>在研究中提出的活动开始时间关键度(starting time criticality, stc) 指标提供了一种合理的鲁棒性评价指标, 该指标衡量的是活动延迟开工造成的期望损失, 从概率论的角度反映了项目计划解鲁棒性. 活动  $j$  的  $stc_j$  定义为

$$stc_j = w_j \Pr(s_j^R > s_j) = w_j \sum_{\forall i: (i,j) \in T(A \cup A_R)} \Pr(d_i^R > s_j - s_i - LPL(i, j)), \quad (2)$$

其中  $\Pr(s_j^R > s_j)$  表示活动  $j$  受到其紧前工序的影响而延迟开工的概率, 计算公式为

$$\Pr(s_j^R > s_j) = \Pr\left(\bigcup_{\forall i: (i,j) \in T(A \cup A_R)} k(i, j)\right), \quad (3)$$

其中  $T$  代表新网络  $G' = (N, A \cup A_R)$  中所有活动直接与间接前后关系集合,  $(i, j) \in T(A \cup A_R)$  表示活动  $i$  是  $j$  的前序活动. 事件  $k(i, j)$  表示活动  $j$  的计划开始时间受到了其前序活动  $i$  的影响, 则这一事件发生的概率为  $\Pr(k(i, j)) = \Pr(s_i^R + d_i^R + LPL(i, j) > s_j)$ . 其中  $LPL(i, j)$  表示项目网络  $G'$  中活动  $i$  到活动  $j$  的最长路径时间.

计算式(3)时需要遵循两点假设<sup>[2,22]</sup>: (a) 假设活动  $j$  的前序活动  $i$  按照基准调度计划的开始时间执行, 即  $s_i^R = s_i$ ; (b) 假设任意时刻只有一个紧前活动  $i$  影响活动  $j$  的按时开工. 那么, 活动  $j$  延迟开工的概率就等于所有紧前活动(考虑工序约束和资源转移关系而形成的网络  $G'$  中与  $j$  有直接或间接紧前关系的活动)影响其正常开工的概率之和, 即

$$\Pr(s_j^R > s_j) = \sum_{\forall i: (i,j) \in T(A \cup A_R)} \Pr(d_i^R > s_j - s_i - LPL(i, j)). \quad (4)$$

由于活动随机工期  $d_i^R$  的分布已知(本文采用对数正态分布假设), 基于  $s_i, s_j, LPL(i, j)$  以及活动工期的概率分布, 就可以计算出任意活动  $j$  的  $stc_j$  值. 该指标的优点在于兼顾了调度计划中的活动权重、活动执行时间变动、活动间工序约束及资源转移关系等项目特征, 并且不需要通过模拟项目执行过程就可以定量计算出来, 大大节省了计算时间.

项目 stc 是所有活动  $stc_j$  之和, 因此本文模型的解鲁棒性目标为

$$stc = \sum_{j \in N} stc_j. \quad (5)$$

### 2) 资源转移成本目标(resource transfer cost, RTC)

$$RTC = \sum_{i,j \in N} \sum_{k \in K} c_k^{ij} \sum_{l=1}^{R_k} y_{kl}^{ij}, \quad (6)$$

其中  $c_k^{ij}$  代表资源  $k$  在活动  $i$  与活动  $j$  之间的单位转移成本.

需要注意的是, 本文方法模型同样适用于项目间共享资源的多项目环境, 此时资源转移成本包括可更新

资源在项目内部活动间的转产成本以及资源在项目间的转场成本.

约束条件

$$\begin{cases} (s_i + d_i - s_j)y_{kl}^{ij} \leq 0, & \forall i, j \in N, i \neq j, s_i < s_j, k \in K, l = 1, 2, \dots, R_k \\ y_{kl}^{ij} = 0, & \text{否则,} \end{cases} \quad (7)$$

$$\sum_{j \in N} y_{kl}^{0j} = 1, \quad \forall k \in K, l = 1, 2, \dots, R_k, \quad (8)$$

$$\sum_{j \in N} y_{kl}^{j0} = 0, \quad \forall k \in K, l = 1, 2, \dots, R_k, \quad (9)$$

$$\sum_{j \in N} y_{kl}^{j, n+1} = 1, \quad \forall k \in K, l = 1, 2, \dots, R_k, \quad (10)$$

$$\sum_{j \in N} y_{kl}^{n+1, j} = 0, \quad \forall k \in K, l = 1, 2, \dots, R_k, \quad (11)$$

$$\sum_{j \in N} y_{kl}^{ij} = \sum_{j \in N} y_{kl}^{ji} \leq 1, \quad \forall i \in N \setminus \{0, n+1\}, \forall k \in K, l = 1, 2, \dots, R_k, \quad (12)$$

$$\sum_{j \in N} \sum_{l=1}^{R_k} y_{kl}^{ij} = \sum_{j \in N} \sum_{l=1}^{R_k} y_{kl}^{ji} \geq r_{ik}, \quad \forall i \in N \setminus \{0, n+1\}, \forall k \in K, \quad (13)$$

$$s_j^R = f(s_j, Y, \sigma), \quad \forall j \in N, \quad (14)$$

$$y_{kl}^{ij} = \{0, 1\}, \quad \forall i, j \in N, \forall k \in K, l = 1, 2, \dots, R_k. \quad (15)$$

式(7)是资源转移约束,表示任何存在资源转移的活动*i*和*j*,执行时间不允许出现交叠,该约束同时保证了任意时刻资源消耗总量不超过其可用量.式(8)和式(9)是虚拟首活动约束,表示每个资源单元起始必须从虚拟首活动流向其他活动,并且资源单元不能从其他活动流向虚拟首活动.式(10)和式(11)表示虚拟尾活动约束,表示每个资源单元最终必须从其他活动流向虚拟尾活动,并且资源单元不能从虚拟尾活动流向其他活动.式(12)是资源流平衡约束,表示对于每个实际活动,资源如果流入该活动,必然从该活动流出,流向其他活动.式(13)是资源需求约束,表示项目调度过程中任意实际活动对每种资源的需求必须得到满足.不确定环境下,项目活动的实际开工时间无法唯一确定,但通过上文分析可知,活动*j*的实际开工时间 $s_j^R$ 受其计划开工时间 $s_j$ ,资源分配方案(用所有 $y_{kl}^{ij}$ 构成的集合 $Y$ 表示)以及不确定性程度(用工期标准差 $\sigma$ 表示)的影响,因此式(14)用抽象函数的方式表示这种映射关系.式(15)是决策变量值域约束,其中 $y_{kl}^{ij} = 1$ 表示资源单元 $e_{kl}$ 从活动*i*流向活动*j*.此决策的目的是满足约束式(7)~式(15),使得目标函数式(5)和式(6)达到最小.

### 3 求解算法设计

上述模型作为RCPSP的拓展问题,显然也是NP难问题,相对于精确求解算法,智能算法越来越表现出其优越性而受到国内外学者的青睐.其中,模拟退火算法(simulated annealing, SA)和遗传算法(genetic algorithm, GA)是解决组合优化问题较为常用的两种智能算法.SA算法局部搜索能力强,但是搜索域较小,对问题解空间的覆盖率较低,容易陷入局部最优;GA算法具备较大的搜索域但收敛较慢,对初始解的依赖性极大,导致算法稳定性不够强.鉴于此,本文结合两种算法各自的优点,构造新的遗传退火算法(GSA)求解考虑项目鲁棒性与资源转移成本的资源流网络模型.

#### 3.1 GSA算法编码与解码设计

##### 1) 编码与初始化

式(5)~式(15)中的决策变量  $y_{kl}^{ij}$  表示资源单元  $e_{kl}$  在活动  $i$  与活动  $j$  之间是否发生了转移, 据此设计基于资源链的二维矩阵方式对决策变量集进行编码, 图 5 给出了解模型编码示意图. 为了表述方便, 下文将每单位资源看作一个资源单元, 资源可用量  $R_k$  视作资源  $k$  所有资源单元的集合, 编码矩阵中每一列对应一个活动  $j \in N$ , 矩阵每一行对应一个资源单元  $l \in R$ ,  $R = \bigcup_{k \in K} R_k$ . 可见每一单元的资源都是独一无二的, 即使资源种类相同, 不同资源单元的资源链并在一起构成的二维矩阵包含了所有资源的转移信息. 编码矩阵中的数值只能取 0 或 1, 取值为 1 表示该行代表的资源单元流过了该列代表的活动, 取值为 0 表示该行代表的资源单元并没有被该列代表的活动使用, 结合各活动的计划开工时间就可以获得该资源单元具体的转移路径.

采用随机分配资源单元的方法生成初始种群, 具体做法是将所有活动按照计划开始时间由小到大的顺序排列, 组成活动列表  $L$ . 针对每一种资源  $k(k \in K)$ , 逐个选择列表  $L$  中的活动  $j$ , 从  $R_k$  个资源单元中计算得到活动  $j$  开始时刻  $s_j$  可用的资源单元集合, 记为  $R'_k (R'_k \subseteq R_k)$ , 从可用资源单元集合  $R'_k$  中随机选择  $r_{jk}$  个资源单元分配给活动  $j$ , 即令资源链矩阵中第  $j$  列中  $r_{jk}$  个资源单元对应元素为 1, 然后令已分配的  $r_{jk}$  个资源单元在活动  $j$  执行时间段  $[s_j, s_j + d_j]$  不可用.

	活动									
资源单元	1	1	0	0	0	0	0	0	0	1
	1	1	0	0	1	0	0	0	0	1
	1	1	0	0	1	1	0	0	1	1
	1	1	0	0	1	1	0	0	1	1
	1	1	0	0	1	1	0	0	1	1
	1	0	1	0	0	0	1	0	0	1
	1	0	1	0	0	0	1	0	0	1
	1	0	1	0	0	0	1	0	0	1
	1	0	0	1	0	0	1	1	0	1
	1	0	0	1	0	0	0	1	0	1

图 5 基于资源链的二维矩阵编码

Fig. 5 Resource-oriented two-dimensional matrix coding

2) 解码

对基于资源链的二维矩阵编码进行解码的操作需要借助已知的项目调度计划, 首先得到编码矩阵中每一行取值为 1 的列所对应的活动集合, 然后将这些活动按照计划开工时间的先后顺序排列, 则该序列中每相邻的两个活动都存在资源单元转移. 此外, 可以通过同类资源合并的方式得到项目活动间不同资源转移的数量  $f(i, j, k)$ , 进而画出基于活动的资源流网络图.

3.2 GSA 算法算子设计

GSA 算法继承了遗传算法包括选择、交叉和变异等所有的算子操作, 另外考虑到上述编码在算子操作过程中可能会出现非法解的情况, 本文对非法解将采取修复操作. 以下将对 GSA 算法的选择、交叉、变异和修复策略分别进行详细介绍.

1) 选择

GSA 算法为了强化遗传算法的广度搜索能力, 将采取无差别选择的策略, 即不区分种群中的个体, 所有个体都 100% 参与交叉, 这种方法能够在每一代中产生尽可能多的邻域解, 极大地扩大了解的搜索范围.

除此之外, 对于被选择参与交叉的个体如何确定具体的交叉对象也是需要解决的问题. 在给出具体的交叉对象选择策略之前, 首先定义个体差异度的概念. 个体差异度是针对 0-1 矩阵编码方式提出的, 表示种群中两个个体等位基因取值不同的位置数占染色体基因总数的比率, 计算公式为

$$\Delta(A, B) = n_{AB} / \text{num}, \tag{16}$$

其中  $n_{AB}$  表示个体  $A$  和个体  $B$  等位基因取值不同的数, num 表示基因总数.

如图 6 所示,  $A$  和  $B$  分别代表资源流模型的两个解, 由此很容易得到  $A$  和  $B$  的个体差异度为  $4/110$ . 本文在确定具体的交叉对象时, 先计算种群中两两个体的个体差异度, 将其从大到小排序, 每次选择交叉的两个父代都是种群中剩余个体差异度最大的两个. 基于个体差异度的选择策略避免了父代出现近亲繁殖的现象, 这就保证了种群每次都能出现较大幅度的进化, 进一步扩大了解的搜索范围.

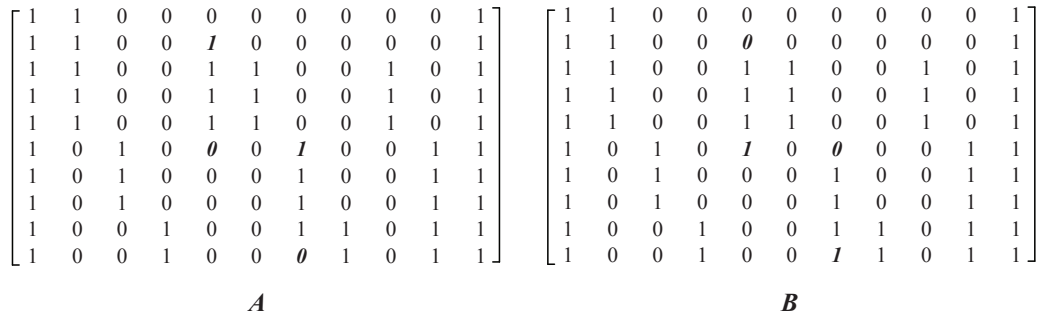


图 6 个体差异度示意图

Fig. 6 Illustration of the individual difference

注: 黑色斜体表示个体  $A$  和个体  $B$  等位基因取值不相同.

2) 交叉

本文基于双点交叉策略对父代个体进行遗传操作. 为了保证交叉操作能够尽可能将优良基因遗传给后代, 同时考虑到后续修复操作的复杂性与时间要求, 本文所设计的交叉策略限制了交叉片段的长度, 每次只需交换自交叉点以后固定长度的基因片段. 如图 7 所示, 将交叉片段长度设为 2, 对图 6 中的个体  $A$  和个体  $B$  进行交叉.

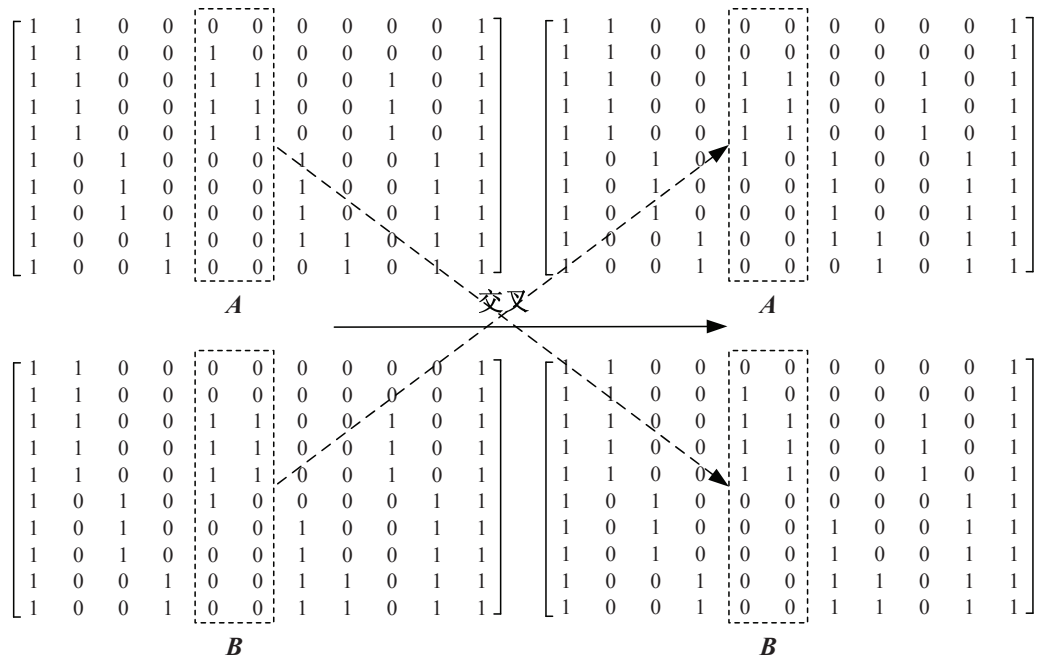


图 7 双点交叉示意图

Fig. 7 Illustration of two-point crossover

3) 变异

采取选择性变异的策略强化 GSA 算法的局部搜索能力, 对于由父代交叉产生的子代, 如果其平均适应性优于父代的平均适应性, 则此子代直接加入新种群不必参与变异, 否则对父代进行概率性的变异, 并将变



异后的父代加入新种群.

具体变异方法通常有逆序变异、等位基因取反变异等, 对于二维矩阵型编码, 采取列倒位的方式实现变异操作, 即随机选择编码矩阵的一列, 将该列的基因以倒序排列的方式加入到原矩阵列中. 图 8 是对图 6 中个体  $A$  采取变异操作的示意图.

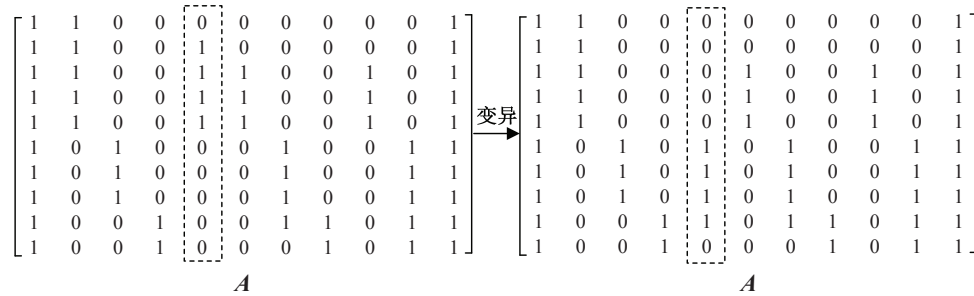


图 8 列倒位变异示意图

Fig. 8 Illustration of column inversion-based mutation

#### 4) 修复

由 3.2 节构建的优化模型可知, 资源约束除了满足活动对资源的需求之外, 还必须满足任何时刻正在使用的资源量不得超过该类资源总量的条件, 这在基于资源链的矩阵编码中等价于同一资源链中取值为 1 的活动不能同时执行. 然而这个约束在进行交叉和变异操作时极易被打破, 从而产生非法解. 遗传算法对于非法解的处理通常有直接舍弃和修复两种, 本文 GSA 算法选择对非法解采取修复措施. 针对矩阵编码的特点, 采取递归修复的策略, 基本思想总结如下:

首先对于交叉/变异区域内按照逐列逐行的顺序对可能存在冲突的元素(值为 1)逐个检测, 如果第  $i$  行第  $j$  列元素  $(i, j)$  存在冲突, 则在第  $j$  列中寻找取值为 0 且可以变为 1 而不引起冲突的元素  $(i', j)$ , 如果存在这样的  $(i', j)$ , 则  $(i, j) \leftarrow 0, (i', j) \leftarrow 1$ , 冲突消解成功. 如果不存在满足条件的  $(i', j)$ , 则在第  $j$  列中寻找交叉前为 1, 现在变为 0 的元素  $(i'', j)$ ,  $(i, j) \leftarrow 0, (i'', j) \leftarrow 1$ , 现在  $(i'', j)$  为冲突元素, 针对  $(i'', j)$  继续上述冲突消解步骤, 直到不存在冲突为止.

具体步骤如下:

**步骤 1** 设资源链矩阵中交叉/变异区域为  $A(i_0 : i_1, j_0 : j_1), i \leftarrow i_0, j \leftarrow j_0$ ;

**步骤 2** 如果  $i > i_1$  或者  $j > j_1$ , 则算法终止, 得到可行解, 否则, 转到步骤 3;

**步骤 3** 如果  $A(i, j) = 0$ , 转到步骤 4, 否则转到步骤 5;

**步骤 4** 如果  $j < j_1$ , 则  $j \leftarrow j + 1$ , 否则  $i \leftarrow i + 1, j \leftarrow j_0$ . 转到步骤 2;

**步骤 5** 获得与活动  $j$  抢占资源单元  $i$  的冲突活动集合  $\text{ConflictSet}_{ij}$ ;

**步骤 6** 如果  $\text{ConflictSet}_{ij} = \emptyset$ , 转到步骤 4, 否则转到步骤 7;

**步骤 7** 在第  $j$  列中搜索得到可用于修复的行  $\text{FeaRow}_j$ , 即当前值为 0, 变为 1 不会引起其他活动冲突的行;

**步骤 8** 如果  $\text{FeaRow}_j \neq \emptyset$ , 转到步骤 9, 否则转到步骤 10;

**步骤 9** 在  $\text{FeaRow}_j$  中随机选择一行  $i'$ , 令  $A(i, j) = 0, A(i', j) = 1$ , 转到步骤 4;

**步骤 10** 在第  $j$  列中搜索得到可用于交换的行  $\text{ExaRow}_j$ , 即当前值为 0, 交叉/变异前为 1 的行;

**步骤 11** 在  $\text{ExaRow}_j$  中随机选择一行  $i'$ , 令  $A(i, j) = 0, A(i', j) = 1$ ;

**步骤 12** 得到资源单元  $i'$  上与活动  $j$  冲突的活动集合  $\text{ConflictSet}_{i'j}$ ;

**步骤 13** 对于  $\text{ConflictSet}_{i'j}$  中的每一个冲突活动, 递归调用步骤 5~步骤 12, 消解所有冲突, 转到步骤 4, 算法结束.

### 3.3 GSA 算法适应值函数设计

适应值是评判遗传算法产生的解优劣的标准, 对解适应值的优化等价于对组合问题的优化, 因此算法适应值函数必须与组合问题的目标函数具有较强相关性, 并且容易计算. 对于单目标优化问题, 通常直接使用目标函数或者目标函数的倒数作为适应值函数, 本文结合逼近理想解法(Topsis)和马氏距离相关理论设计 GSA 算法的适应值函数.

Topsis 法又称为双基点法, 它通过评价当前解与理想解、负理想解的距离对各可行方案进行排序, 若当前解最靠近理想解同时又最远离负理想解, 则为最优, 否则不为最优. 其中, 理想解是各指标值都达到评价指标最优值的解, 而负理想解是各指标值都作为评价指标最差值的解. 本文对项目鲁棒性和资源转移成本的优化属于计量单位不一致的双目标优化, 借鉴 Topsis 法, 忽略负理想解的影响, 采用双目标函数值与理想解目标函数值间的马氏距离作为 GSA 算法适应值计算指标.

马氏距离是一种计算两个来自同一总体的样本集的相似度的方法, 它表示数据的协方差距离, 马氏距离与原始数据的量纲无关. 设  $\mathbf{X}$  和  $\mathbf{Y}$  分别表示来自同一总体的两个点,  $\mathbf{Q}^{-1}$  表示总体的协方差矩阵的逆矩阵,  $\mathbf{X}$  与  $\mathbf{Y}$  之间的马氏距离为

$$d_m^2(\mathbf{X}, \mathbf{Y}) = (\mathbf{X} - \mathbf{Y})^T \mathbf{Q}^{-1} (\mathbf{X} - \mathbf{Y}). \quad (17)$$

综上所述, GSA 算法适应值函数的计算公式表示为

$$f(c) = \left( \begin{bmatrix} f_c^1 \\ f_c^2 \end{bmatrix} - \begin{bmatrix} f_0^1 \\ f_0^2 \end{bmatrix} \right) \mathbf{Q}^{-1} \left( \begin{bmatrix} f_c^1 \\ f_c^2 \end{bmatrix} - \begin{bmatrix} f_0^1 \\ f_0^2 \end{bmatrix} \right), \quad (18)$$

其中  $f_c^1$  和  $f_c^2$  分别表示 GSA 算法解码后个体  $c$  的项目 stc 和资源调度成本 RTC,  $f_0^1$  和  $f_0^2$  表示理想解解码后的项目 stc 和资源调度成本 RTC,  $\mathbf{Q}^{-1}$  为解空间协方差矩阵的逆矩阵.

### 3.4 GSA 算法步骤

本文设计的 GSA 算法的核心在于结合 GA 算法和 SA 算法的优点, 强化 GA 的广度搜索能力和 SA 的深度搜索能力, 以达到更好的寻优效果. 算法详细步骤总结如下:

**步骤 1** 初始化参数, 设定种群规模 POP\_NUM, 初始温度  $T_s$ , 终止温度  $T_e$  和降温率  $\lambda (\lambda \in (0, 1))$ , 当前温度  $T \leftarrow T_s$ ;

**步骤 2** 随机生成个体数为 POP\_NUM 的初始种群  $P_0$ ;

**步骤 3** 算种群个体差异度, 基于个体差异度, 确定种群中个体的交叉组合集;

**步骤 4** 选择交叉组合集中个体差异度最大的两个个体  $\mathbf{A}$  和  $\mathbf{B}$  作为父代, 将其进行交叉生成两个子代个体, 并对非法子代采取修复措施, 产生健康子代  $\mathbf{A}'$  和  $\mathbf{B}'$ ;

**步骤 5** 分别计算父代和子代个体的适应度  $f(\mathbf{A})$ ,  $f(\mathbf{B})$ ,  $f(\mathbf{A}')$  和  $f(\mathbf{B}')$ , 并以此计算出父代和子代的平均适应值分别为  $\bar{f}_0$  和  $\bar{f}_1$ ;

**步骤 6** 比较  $\bar{f}_0$  和  $\bar{f}_1$ , 如果  $\bar{f}_1 < \bar{f}_0$ , 则接受子代, 将个体  $\mathbf{A}'$  和  $\mathbf{B}'$  加入新种群; 否则将父代  $\mathbf{A}$  和  $\mathbf{B}$  以概率  $\exp(-|\bar{F}_0 - \bar{F}_1|/T)$  进行变异, 并对非法变异个体采取修复措施, 无论变异成功与否都将其加入到新种群中;

**步骤 7** 将个体  $\mathbf{A}$  和  $\mathbf{B}$  移出交叉组合集, 判断交叉组合集中是否还有元素, 如果有, 转入步骤 4, 如果没有元素,  $T \leftarrow \lambda T$ , 判断  $T = T_e$  是否成立, 如果成立, 则转入步骤 8, 不成立则转入步骤 3;

**步骤 8** 算法终止, 输出结果.

## 4 模拟实验与案例研究

### 4.1 算法性能分析

为了验证 GSA 算法的有效性, 同时测试算法对原 SA 算法和 GA 算法的改进效果, 本节设计了三组对

比实验, 分别从寻优效果、收敛速度、执行效率三方面进行比较分析. 所选取的项目算例来自 Patterson 标准测试数据集<sup>[23]</sup>, 每组实验选取项目例库中资源紧密度和活动复杂度相似的 3 个项目组合成单项目, 其中每个子项目包含 8 个实活动和 2 个虚活动, 三组实验的可更新资源种类数分别为 1, 2, 3.

为了保证对比研究的可信度, 本文对实验的参数严格控制, 相关参数设置见表 2. 采用 MATLAB 编译程序, 算法运行在 CPU 为 8 核 4 GHz、内存 16 G 的个人电脑, 操作系统为 Windows 7. 为了消除实验的偶然性, 每组项目分别用同一算法求解 30 次, 统计实验结果如表 3 所示.

表 2 算法参数设置

Table 2 Parameter settings of the algorithms

控制参数	参数值
SA/ GSA 初始温度	1 000
SA/ GSA 终止温度	0.001
SA/ GSA 降温速率	0.95
GA/ GSA 种群规模	50
GA 终止代数	500

表 3 三种算法性能对比结果

Table 3 Performance comparison of three algorithms

实验组	资源种类	算法	距离最优解 10% 范围比率	平均最优适应值	平均收敛 迭代次数	平均执行时间(s)
A 组	1	GA	80.00 %	0.083 5	254.48	55.24
		SA	76.67 %	0.084 3	111.05	47.37
		GSA	86.67 %	0.079 8	198.92	68.67
B 组	2	GA	70.00 %	0.072 6	372.33	82.94
		SA	66.67 %	0.074 8	237.21	63.11
		GSA	80.00 %	0.064 9	336.47	103.65
C 组	3	GA	53.33 %	0.299 7	487.23	114.82
		SA	50.00 %	0.315 3	293.67	85.44
		GSA	70.00 %	0.286 6	373.35	152.36

通过对表 3 实验结果的分析, 可以得出以下结论:

1) 随着资源种类的增加, 项目资源分配问题变得更复杂, 三种算法在寻优效果、收敛速度以及执行效率方面的表现均呈下降趋势. 2) GSA 算法的寻优能力强于 GA 和 SA, 而且随着资源种类的增加 GSA 算法的这种优势越来越明显. 3) 收敛速度方面, SA 算法表现最好, 其次是 GSA 算法, 收敛最慢的是 GA 算法. 4) 算法时间复杂度方面, GSA 算法执行效率最低, 其次是 GA 算法, SA 算法表现最好. 由于 GSA 算法和 GA 算法都存在编码和解码的操作, 而这都需要消耗大量的时间, 另外 GSA 算法的局部寻优过程同样比较耗时, 因此, GSA 算法求解效率总体上较 GA 算法和 SA 算法差.

总体来看, GSA 算法很好地汲取了 GA 算法和 SA 算法的优点, 求解效果和收敛速度都得到了优化, 唯一不足的是运行效率不如人意, 这意味着 GSA 算法在求解规模巨大的项目资源分配问题时可能比较困难.

## 4.2 案例研究

本节以某高校道路施工项目作为案例材料, 旨在将本文所提的考虑鲁棒性和资源转移成本的项目资源分配优化模型与另外三种优化模式进行比较. 该工程项目基本信息如下.

M 建筑公司成功中标 H 高校 2016 年“两横三纵”校园道路改造工程, 该工程分为 A, B, C, D 和 E 等 5 条校园主干道路的施工项目. 由于 5 条道路的施工属于同类工程, 现工程组决定在现有资源条件下同期开展这 5 个项目, 每个项目的工序流程基本一致, 如图 10 所示. 工程组清点公司派发的施工设备时, 发现挖掘机、装载机和钢筋弯曲机 3 种机器无法满足计划人员按照关键路径法设定的项目计划, 实际工程可能会出

现项目活动等待资源和项目间资源借调的情况出现,此外,原先计划是计划人员根据以往工程经验预估项目各活动的执行时间而制定的,没有考虑到实际工程中项目活动的偏差风险.为了降低项目偏离计划的风险,同时保证在这种资源受限条件下项目内部以及项目间的资源调度成本不至过高,工程计划人员需要重新制定相应的项目调度计划和资源分配方案.

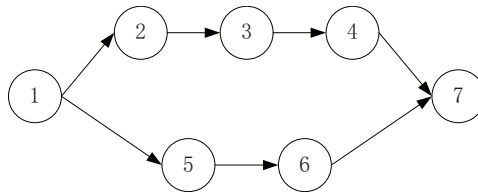


图 9 单个项目网络图

Fig. 10 Network of a single project

该工程 5 个项目依次用代号 A、B、C、D、E 表示,虽然 5 个项目的工序流程图完全一样,但由于不同路段所处的地理位置以及路面长度差异的存在,不同项目同一工序所需的施工时间和资源量完全不同,表 4 和表 5 分别给出了各项目活动的预估执行时间以及对三种受限资源的需求量,其中  $r_1$ 、 $r_2$  和  $r_3$  分别代表挖掘机、装载机和钢筋弯曲机三种设备资源,总量分别为 6 台,8 台和 6 台.

针对案例描述的问题,先不考虑活动执行时间的不确定性,将该问题用解决传统 RCPSP 问题的方法求解,得到该工程期望最短完工期为 89 d,其中 5 个项目各个活动的计划开工时间如表 6 所示.

表 4 项目各活动预估执行时间  
Table 4 Planned processing times of project activities

项目代号	活动						
	1	2	3	4	5	6	7
A	3	5	10	8	8	15	2
B	2	6	9	8	9	12	2
C	2	4	8	7	8	10	1
D	3	6	7	4	12	15	2
E	1	3	5	3	10	11	3

表 6 项目各活动计划开始时间  
Table 6 Planned starting times of project activities

项目代号	活动						
	1	2	3	4	5	6	7
A	0	20	36	73	20	46	81
B	0	20	37	46	28	54	66
C	1	4	12	81	8	72	88
D	0	37	66	75	3	46	79
E	0	1	4	9	26	61	72

表 5 项目各活动对资源的需求量  
Table 5 Resource requirements of project activities

项目代号	活动																				
	1			2			3			4			5			6			7		
	$r_1$	$r_2$	$r_3$	$r_1$	$r_2$	$r_3$	$r_1$	$r_2$	$r_3$	$r_1$	$r_2$	$r_3$	$r_1$	$r_2$	$r_3$	$r_1$	$r_2$	$r_3$	$r_1$	$r_2$	$r_3$
A	0	2	0	1	1	0	2	3	2	2	2	0	3	4	2	3	2	3	0	2	0
B	0	3	0	2	1	0	1	2	1	1	3	0	2	2	1	1	3	1	0	1	0
C	0	1	0	2	2	0	2	3	3	3	2	0	2	2	1	2	1	2	0	1	0
D	0	1	0	3	2	0	3	2	2	1	3	0	2	1	2	2	3	2	0	2	0
E	0	2	0	2	1	0	2	4	2	2	3	0	3	2	2	3	1	1	0	3	0

将案例基础数据和表 6 求出的项目活动计划开始时间代入第 3 节构建的优化模型,利用 GSA 算法求解,获得最优的资源分配方案,其对应的 stc 值和资源转移成本分别为 291.43 和 2 737.86(此为计划阶段).为了验证所提模型的有效性(即评价计划阶段得到的资源流网络的优劣),现将由该模型求出的最优资源分配方案与随机资源分配方案<sup>[11]</sup>,只优化鲁棒性以及只优化资源成本的资源分配方案进行对比,在仿真环境下模拟四种资源分配方案下项目基准计划的实际运行情况(此为执行阶段),比较四种优化模式在不确定环境下鲁棒性和资源成本的表现.实验相关参数的设置见表 7,实验结果如表 8 所示.

通过对实验仿真结果数据的分析,可以得到以下结论:

1) 随着不确定性水平  $\sigma$  的变化, 相同优化模式下的资源转移成本 RTC 是一样的. 这是因为, 在计划阶段已求解得到项目的资源分配优化方案, 项目模拟时可更新资源即按照已确定的资源流网络关系在活动间发生转移, 因此同一种优化模式对应的资源转移成本是定值. 此外, 通过分析资源转移成本目标函数(式(6))也可以看出, RTC 与项目实际执行时的不确定性水平无关.

表 7 案例实验参数设置  
Table 7 Parameter settings used in the experimental case

控制参数	参数值
工期不确定性水平	$\sigma = 0.3, 0.6, 0.9$
活动实际工期 $d_j^R$	服从对数正态分布 $\text{logrnd}(u(j), \sigma^2)$ , $u(j) = \ln(d_j) - \sigma^2/2$
工程截止工期	$\delta = 89 \text{ d}$
活动权重 $w_j$ (离散三角分布 <sup>[21,22]</sup> )	$\Pr(w_j = x) = 0.21 - 0.02x$ , $x = 1, 2, \dots, 10$ , $\forall j \in N \setminus \{0\}$ , $w_0 = 0$
资源在活动之间的单位转移成本 $c_k^{ij}$	$\Pr(c_k^{ij} = x) = 0.21 - 0.02x$ , $x = 1, 2, \dots, 10$ , $\forall i, j \in N$ , $k \in K$
进度生成机制	并行调度(parallel schedule generation scheme)
项目调度策略	时刻表策略(railway scheduling) <sup>[24]</sup>
模拟执行次数	$M = 2000$
资源转移成本 RTC	
项目绩效指标	惩罚成本 $SC = \sum_{j \in N} w_j E s_j - s_j^R $ 准时完工率 $TPCP = \Pr(s_{n+1}^R \leq \delta)$

表 8 案例实验对比结果  
Table 8 Comparison results for the experimental case

不确定程度	优化模式	RTC	SC	RTC+SC	TPCP
$\sigma = 0.3$	随机	3 139.10	1 530.93	4 670.03	92.10 %
	优化鲁棒性	3 069.48	1 316.84	4 386.32	96.90 %
	优化资源成本	2 700.48	1 477.64	4 178.12	93.30 %
	同时优化	2 737.86	1 333.92	4 071.78	95.70 %
$\sigma = 0.6$	随机	3 139.10	3 926.19	7 065.29	49.60 %
	优化鲁棒性	3 069.48	3 305.12	6 374.60	57.25 %
	优化资源成本	2 700.48	3 619.71	6 320.19	54.95 %
	同时优化	2 737.86	3 415.95	6 153.81	57.65 %
$\sigma = 0.9$	随机	3 139.10	6 729.28	9 868.38	34.05 %
	优化鲁棒性	3069.48	5 894.72	8 964.20	39.05 %
	优化资源成本	2 700.48	6 333.38	9 033.86	37.50 %
	同时优化	2 737.86	5 954.64	8 692.50	40.35 %

2) 相同  $\sigma$  的情况下, 不同优化模式对应的资源转移成本 RTC 具有规律性的差异, 对于任意  $\sigma$ , RTC 大小排序为: 随机 > 优化鲁棒性 > 同时优化 > 优化资源成本, 另外还可以发现同时优化模式和只优化资源成本模式对应的RTC差异并不大, 且都远小于随机和只优化鲁棒性两种模式, 这证明本文提出的双目标优化模型对项目资源转移成本的优化是有效的;

3) 相同  $\sigma$  的情况下, 不同优化模式对应的惩罚成本SC 具有规律性的差异, 对于任意  $\sigma$ , SC 大小排序为: 随机 > 优化资源成本 > 同时优化 > 优化鲁棒性, 这说明本文提出的双目标优化对项目计划解鲁棒性的优化是有效的. 另外还可以发现虽然不同优化模式对应的准时完工率 TPCP 并没有呈现固定规律的差异(同时优化和只优化鲁棒性对应的 TPCP 的大小顺序并不固定), 但同时优化和只优化鲁棒性对应的 TPCP 均大于随机和只优化资源转移成本, 这说明通过对项目计划解鲁棒性的优化在一定程度上也能优化项目计划的质鲁棒性, 即解鲁棒性与质鲁棒性存在一定的相关性.

## 5 结束语

本文在鲁棒项目调度问题中考虑可更新资源的转移成本,以项目鲁棒性和资源转移成本为优化对象,建立了活动工期不确定条件下的资源流网络优化模型.针对问题模型的NP难特性,提出结合GA和SA优点的GSA算法进行求解,设计了基于资源链的编码方式及适应问题特征的遗传算子.未来研究将考虑构建一阶段的鲁棒项目调度与资源分配集成优化模型,同时优化活动开始时间和资源流,这将会极大地简化优化过程.其次,如何在工期/资源不确定环境中考虑资源转移时间建立资源流模型,保证调度计划的鲁棒性,值得进一步探索.最后,本文对项目调度成本和鲁棒性的优化还不够全面,项目调度的成本还包括资源闲置成本、项目延期成本等,鲁棒性项目调度还需考虑质鲁棒性,因此多目标综合优化是未来项目调度研究的重要方向之一,这也对设计更高效的多目标优化算法提出了挑战.

### 参考文献:

- [1] Hu X, Cui N, Demeulemeester E, et al. Incorporation of activity sensitivity measures into buffer management to manage project schedule risk. *European Journal of Operational Research*, 2016, 249(2): 717–727.
- [2] 崔南方, 梁洋洋. 基于资源流网络与时间缓冲集成优化的鲁棒性项目调度. *系统工程理论与实践*, 2018, 38(1): 102–112.  
Cui N F, Liang Y Y. Robust project scheduling based on the integrated optimization between resource flow network and time buffers. *Systems Engineering: Theory & Practice*, 2018, 38(1): 102–112. (in Chinese)
- [3] 田文迪, 胡慕海, 崔南方. 不确定性环境下鲁棒性项目调度研究综述. *系统工程学报*, 2014, 29(1): 135–144.  
Tian W D, Hu M H, Cui N F. Review of studies on robust project scheduling under uncertainty. *Journal of Systems Engineering*, 2014, 29(1): 135–144. (in Chinese)
- [4] 何正文, 宁敏静, 徐渝. 前摄性及反应性项目调度方法研究综述. *运筹与管理*, 2016, 25(5): 278–287.  
He Z W, Ning M J, Xu Y. A survey of proactive and reactive project scheduling methods. *Operations Research and Management Science*, 2016, 25(5): 278–287. (in Chinese)
- [5] Demeulemeester E, Herroelen W. *Robust Project Scheduling*. Hanover: Now Publishers Inc., 2011.
- [6] 崔南方, 赵雁, 田文迪. 基于智能算法的双目标鲁棒性项目调度. *系统管理学报*, 2015, 24(3): 379–388.  
Cui N F, Zhao Y, Tian W D. Bi-objective robust project scheduling based on intelligent algorithms. *Journal of Systems & Management*, 2015, 24(3): 379–388. (in Chinese)
- [7] Deblaere F, Demeulemeester E, Herroelen W. Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 2011, 214(2): 308–316.
- [8] 张静文, 刘耕涛. 基于鲁棒性目标的关键链项目调度优化. *系统工程学报*, 2015, 30(1): 135–144.  
Zhang J W, Liu G T. Critical chain project scheduling problem with the robust objective. *Journal of Systems Engineering*, 2015, 30(1): 135–144. (in Chinese)
- [9] 王伟鑫, 葛显龙, 王旭, 等. 基于关键链的非抢占式多项目调度多属性优化. *系统工程学报*, 2016, 31(5): 689–699.  
Wang W X, Ge X L, Wang X, et al. Multi-attribute optimization for non-preemptive multi-project scheduling based on critical chain. *Journal of Systems Engineering*, 2016, 31(5): 689–699. (in Chinese)
- [10] 张沙清, 陈新度, 陈庆新, 等. 基于优化资源流约束的模具多项目反应调度算法. *系统工程理论与实践*, 2011, 31(8): 1571–1580.  
Zhang S Q, Chen X D, Chan Q X, et al. Reactive scheduling algorithm for multiple mould and die projects based on optimized resource flow constraints. *Systems Engineering: Theory & Practice*, 2011, 31(8): 1571–1580. (in Chinese)
- [11] Artigues C, Michelon P, Reusser S. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 2003, 149(2): 249–267.
- [12] Leus R, Herroelen W. Stability and resource allocation in project planning. *IIE Transactions*, 2004, 36(1): 1–16.
- [13] Leus R. *The Generation of Stable Project Plans*. Leuven: Katholieke Universiteit Leuven, 2003.
- [14] Policella N. Scheduling with uncertainty: A proactive approach using partial order schedules. *AI Communications*, 2005, 18(2): 165–167.

- [15] Deblaere F, Demeulemeester E, Herroelen W, et al. Robust resource allocation decisions in resource-constrained projects. *Decision Sciences*, 2007, 38(1): 5–37.
- [16] Krüger D, Scholl A. A heuristic solution framework for the resource constrained (multi-)project scheduling problem with sequence-dependent transfer times. *European Journal of Operational Research*, 2009, 197(2): 492–508.
- [17] Krüger D, Scholl A. Managing and modelling general resource transfers in (multi-)project scheduling. *OR Spectrum*, 2010, 32(2): 369–394.
- [18] Poppenborg J, Knust S. A flow-based tabu search algorithm for the RCPSP with transfer times. *OR Spectrum*, 2016, 38(2): 305–334.
- [19] Demeulemeester E, Herroelen W. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 1992, 38(12): 1803–1818.
- [20] Hu X, Demeulemeester E, Cui N, et al. Improved critical chain buffer management framework considering resource costs and schedule stability. *Flexible Services & Manufacturing Journal*, 2017, 29(2): 159–183.
- [21] Lambrechts O, Demeulemeester E, Herroelen W. A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics*, 2008, 111(2): 493–508.
- [22] Van de Vonder S, Demeulemeester E, Herroelen W. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operation Research*, 2008, 189(3): 723–733.
- [23] Patterson J. A comparison of exact approaches for solving the multiple constrained resource project scheduling problem. *Management Science*, 1984, 30 (7): 854–867.
- [24] Tian W, Demeulemeester E. Railway scheduling reduces the expected project makespan over roadrunner scheduling in a multi-mode project scheduling environment. *Annals of Operations Research*, 2014, 213(1): 271–291.

#### 作者简介:

胡雪君(1989—), 女, 湖北襄阳人, 博士, 副教授, 研究方向: 项目调度, 组合优化, 生产运作管理, Email: xuejun.hu@hnu.edu.cn;

王建江(1986—), 男, 新疆乌鲁木齐人, 博士, 讲师, 研究方向: 卫星资源规划调度, 组合优化, 项目调度, Email: jianjiang-wang@nudt.edu.cn;

崔南方(1963—), 男, 江西宁都人, 博士, 教授, 博士生导师, 研究方向: 项目管理, 生产运作管理, 供应链管理, Email: nfcui@mail.hust.edu.cn;

黄 浩(1991—), 男, 湖北黄冈人, 硕士, 研究方向: 项目调度, 组合优化, Email: 173904839@qq.com.