

# 资源受限项目调度模型的施工进度管理

王 静<sup>1,2</sup>, 曾莎洁<sup>2</sup>, 琚 娟<sup>2</sup>

(1. 同济大学 经济与管理学院, 上海 200092; 2. 上海建科工程咨询有限公司, 上海 200032)

**摘要:** 综合资源受限项目调度问题的研究和实际建设工程项目施工进度管理的需求, 提出广义优先关系下的差分进化人工蜂群算法的资源受限项目调度模型。其中差分进化算法用于处理交叉变异问题, 人工蜂群算法用于处理寻优过程的各阶段推进, 并以广义优先关系串行调度产生方案获得的总工期作为适应度值确定优先级编码。根据实际工程项目的应用进一步提出由逆向反推计算关键路径、关键任务的资源及历时优化、非关键任务的资源均衡等组成的资源约束进度管理方法。在 eclipse 环境下用 Java 语言进行编程, 快捷准确地获得进度和资源相关的资源约束进度计划、约束资源分布情况, 以及资源与工期的优化方案。

**关键词:** 进度管理; 资源约束; 差分进化; 人工蜂群; 广义优先关系

**中图分类号:** TU722

**文献标志码:** A

## Construction Schedule Management Using Resource-Constrained Project Scheduling Model

WANG Jing<sup>1,2</sup>, ZENG Shajie<sup>2</sup>, JU Juan<sup>2</sup>

(1. School of Economics and Management, Tongji University, Shanghai 200092, China; 2. Shanghai Jianke Engineering Consulting Co. Ltd., Shanghai 200032, China)

**Abstract:** Combining the study of resource-constrained project-scheduling problem with the demand of construction schedule management in real projects, this paper integrates a hybrid algorithm with generalized precedence relations (GPRs) scheduling model, and then proposes better solutions for resource-constrained schedule management. The hybrid algorithm presents crossover and mutation operations from differential evolution (DE) with artificial bee colony (ABC) to balance exploration and exploitation phases of the optimization process. Furthermore, the proposed model applies a serial method with generalized precedence relations to reflect individual-vector priorities into the task schedule to calculate project duration. In order to generalize the use of most real-world construction projects, the resource-constrained

schedule management contains some methods by calculating critical paths, applying resource leveling of non-critical tasks, using correlation analysis to tradeoffs between time and cost of critical tasks. In the eclipse environment of Java programming language, it could obtain stable and accurate results, which project managers may use to make optimal decisions by resource-constrained schedule, resource allocation, the optimized schemes of resources and duration.

**Key words:** schedule management; resource-constrained; differential evolution; artificial bee colony; generalized precedence relations

现阶段施工进度管理主要以传统任务分解后的工艺约束制定进度计划指导现场施工管理。然而每项施工任务的执行都会占用相应资源, 如果某项任务需要的资源被其他任务所占有, 该任务只能等待其所需资源的释放, 这样必然会影响到实际施工进度, 使得计划进度和实际进度间产生较大偏差, 弱化施工进度管理的效用。因此, 同时满足工艺约束和资源约束的进度计划更适合指导现场施工管理。而且在实际工程施工管理中, 由于各种不确定因素的影响和管理目标侧重点的变化, 需要及时调整资源及工期优化的施工进度方案。

资源受限项目调度问题(resource-constrained project scheduling problem, RCPSP)是一类典型的运筹学难题, 是近年来的研究热点之一, 根据研究问题的设定和侧重点的差异, 研究人员提出了各类调度方案和求解算法。聂慧<sup>[1]</sup>针对广义优先关系下资源受限多项目、多目标、多技能人力资源调度和均衡问题, 提出改进的布谷鸟搜索算法、多目标改进布谷鸟搜索算法、差分进化布谷鸟搜索算法和文化基因算法等仿生算法对各个问题求解。虽提出多种问题组合所对应的算法, 但仅针对人力资源的约束。Tran

收稿日期: 2016-12-30

基金项目: 上海市科委科研计划(15DZ1203401); 上海市国资委项目(S15004)

第一作者: 王 静(1983—), 女, 博士生, 主要研究方向为建设工程管理。E-mail: 0910040077@tongji.edu.cn

等<sup>[2]</sup>提出并验证差分进化人工蜂群算法适用于多种资源约束的项目调度问题,但涉及的调度方案过于简化,没有考虑符合项目实际情况的广义优先关系的项目调度问题。

Karaboga 等<sup>[3-4]</sup>将人工蜂群算法与遗传算法、粒子群算法、粒子群进化算法、差分进化算法和进化策略分别使用不同的适应性函数进行比较,得出的结论是人工蜂群算法优于其他算法,且具有运算过程简洁和全局最优搜索等性能。Li 等<sup>[5-6]</sup>将差分进化算法和人工蜂群算法进行组合,发现混合算法比单独使用各自算法有益于加快搜索优解的速度,达到快速收敛的目的。文献<sup>[7-9]</sup>将差分进化人工蜂群算法应用于各个领域并验证其有效性和适用性。而针对广义优先关系问题,文献<sup>[10-11]</sup>从运筹学的角度提出用广义优先关系处理项目网络并求解资源约束和资源均衡等问题。本文采用差分进化人工蜂群算法进行资源约束的施工进度管理的求解,调度方案采用广义优先关系的串行调度方案。

## 1 资源受限项目调度算法及模型

资源受限项目调度的基本要素可归纳为任务、资源、约束和目标。施工项目任务分解形成调度任务,任务的资源占用和工艺关系形成调度约束,进度计划的优化形成调度时间目标,非关键路径上的资源均衡形成调度资源目标。

### 1.1 改进差分进化人工蜂群算法

差分进化算法由 Price 等<sup>[12]</sup>提出,是一种基于群体进化的算法,具有记忆个体最优解和种群内信息共享等特点,在一定程度上是一种自适应、自组织和自学习的迭代寻优过程。人工蜂群算法(artificial bee colony, ABC)由土耳其学者 Karaboga 等<sup>[13]</sup>提出,将人工蜂群分为引领蜂(employed bees)、跟随蜂(outlooker bees)和侦查蜂(scout bees)3种角色。角色转换是人工蜂群算法的特有机理,3种角色的蜜蜂具有不同的功能作用,引领蜂用于寻找并维持最优解,跟随蜂用于加快收敛速度,侦查蜂用于避免局部最优。通过差分进化算法改进蜂群,运用人工蜂群算法避免领域搜索,运用两种算法的优势,实现快速收敛,全局搜索优解的效果。

根据人工蜂群算法初始化蜜源位置  $X_{ij}$ , 设求解问题的维数为个体向量维度  $D$ , 蜜源位置被抽象成解空间中的点, 代表求解问题的潜在解, 蜜源  $i$  的质量对应解的适应度  $f_i$ ,  $N_p$  为蜜源的数量, 一般引领

蜂和跟随蜂各占蜂群的一半, 蜜蜂的总数量等于蜜源的数量, 且每个蜜源同一时间内只有一只引领蜂。

$$X_{ij} = L_j + \text{rand}[0, 1](U_j - L_j) \quad (1)$$

$$j = 1, 2, \dots, D; i = 1, 2, \dots, N_p$$

式中:  $L_j$  和  $U_j$  分别为搜索的下限和上限;  $\text{rand}[0, 1]$  为  $[0, 1]$  间均匀分布随机概率。

根据差分进化算法进行变异操作和交叉操作。随机选取种群中 2 个不同的个体, 将其向量差缩放后与待变异个体进行向量合成, 设每一代进化目标向量为  $\mathbf{X}_{i,G}$ ,  $i = 1, 2, \dots, N_p$ 。

$$\mathbf{V}_{i,G+1} = \mathbf{X}_{r_1,G} + F(\mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G}) \quad (2)$$

式中:  $\mathbf{X}_{i,G} = [x_{i1,G}, x_{i2,G}, \dots, x_{iD,G}]$ ; 向量编号  $r_1, r_2, r_3 \in \{1, 2, \dots, N_p\}$  且  $r_1 \neq r_2 \neq r_3 \neq i$ ;  $F$  为变异因子, 是差分进化的主要控制参数之一,  $F \in [0, 1]$ 。

交叉操作是为了增加种群群体的多样性, 使之跳出局部最优的局限, 对第  $G$  代种群  $\mathbf{X}_{i,G}$  及其变异的中间体  $\mathbf{V}_{i,G+1}$  ( $\mathbf{V}_{i,G+1} = [v_{i1,G+1}, v_{i2,G+1}, \dots, v_{iD,G+1}]$ ) 进行交叉变异得到  $\mathbf{U}_{i,G+1}$  ( $\mathbf{U}_{i,G+1} = [u_{i1,G+1}, u_{i2,G+1}, \dots, u_{iD,G+1}]$ )。

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1}, \text{rand}_j[0, 1] \leq C \text{ 或 } j = j_{\text{rand}} \\ x_{ij,G}, \text{其他} \end{cases} \quad (3)$$

$$j = 1, 2, \dots, D$$

式中:  $C$  为范围在  $[0, 1]$  之间的交叉概率;  $j_{\text{rand}}$  为  $[0, D]$  之间的随机整数。

将交叉操作所产生的  $\mathbf{U}_{i,G+1}$  和  $\mathbf{X}_{i,G}$  代入目标函数进行适应度评估, 选择适应度值优的向量作为新的种群, 即

$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G+1}, f(\mathbf{U}_{i,G+1}) < f(\mathbf{X}_{i,G}) \\ \mathbf{X}_{i,G}, \text{其他} \end{cases} \quad (4)$$

选择好后, 判断是否达到设定的最大迭代次数或停止条件, 不满足则再次循环迭代, 直至满足, 输出优解。

跟随蜂根据引领蜂的共享信息, 以跟随概率  $p_i$  决定是否跟随。

$$p_i = f_i(X_i) / \sum_{i=1}^{N_p} f_i(X_i), \quad (5)$$

$$p_i = 0.9 \times f_i(X_i) / \max[f(X_i)] + 0.1$$

判断引领蜂是否需要转变成侦查蜂, 在搜索过程中, 如果蜜源  $X_{ij}$  经过  $t$  次迭代搜索到达阈值  $l$  而没有找到更好的蜜源时, 该蜜源  $X_{ij}$  将会被放弃, 与之对应的引领蜂角色转变为侦查蜂。侦查蜂将在搜索空间随机产生一个新的蜜源代替  $X_{ij}$ 。

$$X_{ij,G+1} = \begin{cases} L_i + \text{rand}[0, 1] \times (U_i - L_i), t \geq l \\ X_{ij,G}, t < l \end{cases} \quad (6)$$

改进差分进化人工蜂群算法:一是对优先编码随机样本做了样本筛选和样本数量的改进;二是根据广义优先关系,增加了时序约束的类型.具体步骤见图 1.

(1) 设定蜂群规模、访问次数、循环最大次数、变异系数、交叉系数等初始信息.

(2) 为了实现全局搜索,通过有限次的循环,每次循环根据公式(1)随机生成一组解,再根据公式(2)进行变异,根据公式(3)进行交叉.

(3) 将变异交叉的值代入目标函数和约束条件模型计算适应度值,根据公式(4)进行适应度值的比

较,判断当前循环的最优解.

(4) 根据最优解的判断进行替换,即领域搜索得到的解的适应度若好于当前的最优解则更新,否则继续搜索直到领域搜索循环结束.

(5) 获得每次循环的最优解,根据公式(5)计算跟随概率,跟随蜂重复变异、交叉、对比和更新.

(6) 根据公式(6)判断,若领域循环结束且超出限制,用随机解替代当前解,进行全局循环.

(7) 判断是否达到终止条件,达到输出当前最优解.

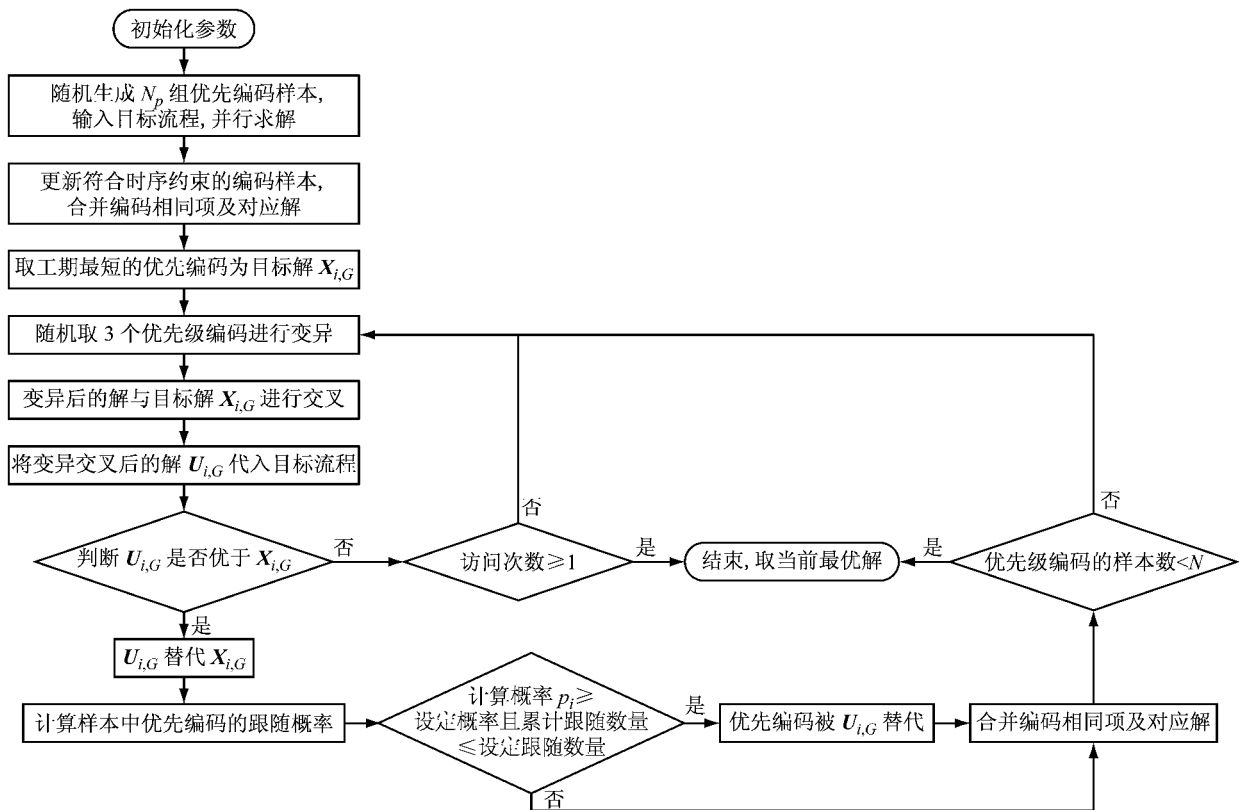


图 1 改进差分进化人工蜂群算法流程图

Fig. 1 Improvement of artificial bee colony with differential evolution

1.2 广义优先关系资源受限调度目标模型

建设工程施工项目中时序的广义优先关系(generalized precedence relations, GPRs)主要有含最小最大时间间隔(time-lag)的开始-开始型(start-start, SS)、完成-开始型(finish-start, FS)、完成-完成型(finish-finish, FF)、开始-完成型(start-finish, SF). 设任务  $i$  是任务  $j$  的广义优先活动,定义  $l_{ij}^{\min}$ 、 $l_{ij}^{\max}$  分别表示任务  $j$  相对于任务  $i$  的最小和最大开始-开始的滞后量,那么任务  $j$  的开始时间约束为  $s_i + l_{ij}^{\min} \leq s_j \leq s_i + l_{ij}^{\max}$ ,其中  $s_i$  和  $s_j$  分别为任务  $i$  和  $j$  的开始时间. 那么其他 3 种类型的时序关系均可以

转化成开始-开始型,见式(7).

$$s_j \geq s_i + l_{ij} \text{ 或 } f_j \geq f_i + l_{ij} \quad (7)$$

其中 SS 型  $l_{ij} = l_{ij}^{\min}$ , SF 型  $l_{ij} = l_{ij}^{\min} - d_j$ , FS 型  $l_{ij} = l_{ij}^{\min} + d_i$ , FF 型  $l_{ij} = l_{ij}^{\min} + d_i - d_j$ ,  $d_i$  和  $d_j$  分别为任务  $i$  和  $j$  的历时.

根据建设工程项目施工任务的分解原则、工艺和持续时间的规定,对资源受限项目调度问题(RCPSP)进行描述. 假设一个项目包含  $N$  个任务,由于施工工艺的要求,任务之间存在时序关系  $G = \{0, 1, \dots, n, n+1\}$ ,其中任务 0 和任务  $n+1$  为项目开始和结束的虚任务,虚任务不占用任何时间和资

源.  $P_i$  为任务集合,  $A_t$  为  $t$  时刻正在执行的任务集合. 用  $T$  表达总工期, 任务  $i$  的持续时间(历时)为  $d_i$ , 开始时间为  $s_i$ , 结束时间为  $f_i$ , 其中  $s_i = f_i - d_i$ .

项目的  $N$  个任务共享  $K$  类可更新资源, 其中第  $k$  类资源的总量为  $R_k, k=1, 2, \dots, K$ , 任务  $i$  对资源类型  $k$  的需求量为  $r_{ik}$ .

资源受限项目调度问题是在满足时序约束和资源约束的前提下, 对所有任务进行排序, 确定各任务的开始时间和完成时间, 并使项目总工期最短. 如式(8)<sup>[14]</sup>所示

$$\begin{aligned} & \min \{ \max f \mid i = 1, 2, \dots, N \} \\ & \text{s. t.} \\ & f_j - f_i \geq l_{ij}; \forall j \in P_i; i = 1, 2, \dots, N \\ & \sum_{A_t} r_{ik} \leq R_k; k = 1, 2, \dots, K; t = s_1, s_2, \dots, s_N \end{aligned} \quad (8)$$

其中, 目标函数表达的是在所有任务组合排序的最大工期中取其最小值; 任务时序约束是通过广义优先关系和时序约束矩阵联合表达; 任务可更新资源约束表达的是在  $t$  时刻所有任务所需资源类型  $k$  的数量总和不能超过总资源  $R_k$ .

通过差分进化人工蜂群算法和广义优先关系资源受限调度, 可以获得受时序约束和资源约束的每个任务的最早开始时间和完成时间, 如图 2 所示, 主

要步骤如下:

(1) 初始化值及参数: 未调度任务集合  $D_n$ 、资源池最大资源供给量  $R_k$ 、任务执行占用资源量  $R_i$ 、任务历时  $D$ 、开始时间  $s_0 = E_S$ 、优先级编码  $\pi$ 、时序约束矩阵  $G$ 、任务数  $n=0$ .

(2) 获得未调用任务  $D_n$  集合.

(3) 根据差分进化人工蜂群算法的可能解获得优先级编码.

(4) 选择  $D_n$  集合中具有最高优先级且满足时序约束矩阵  $G$  的任务  $i$  执行.

(5) 获得该任务  $i$  的持续时间  $d_i$ .

(6) 从集合  $S_n$  中获得任务  $i$  最早可能开始时间  $E_{S_i}$ , 如果是开始-开始和完成-开始类型, 紧后任务的已知时间为开始时间, 执行时间记为  $[s_i, s_i + d_i]$ . 如果是完成-完成和开始-完成类型, 紧后任务的已知时间为完成时间, 执行时间记为  $[f_i, f_i + d_i]$ .

(7) 扫描资源池该时刻的剩余  $R_{k_t}$  是否满足任务  $i$  执行的需求, 满足, 执行, 不满足, 最早开始时间加 1.

(8) 计算任务  $i$  的完成时间, 即  $f_i = s_i + d_i$ .

(9) 将任务  $i$  从  $D_n$  集合中删除, 同时更新  $S_n$  集合, 使得  $S_{n+1} = S_n \cup \{i\}$ .

(10) 设置  $n = n + 1$ , 重复上述步骤.

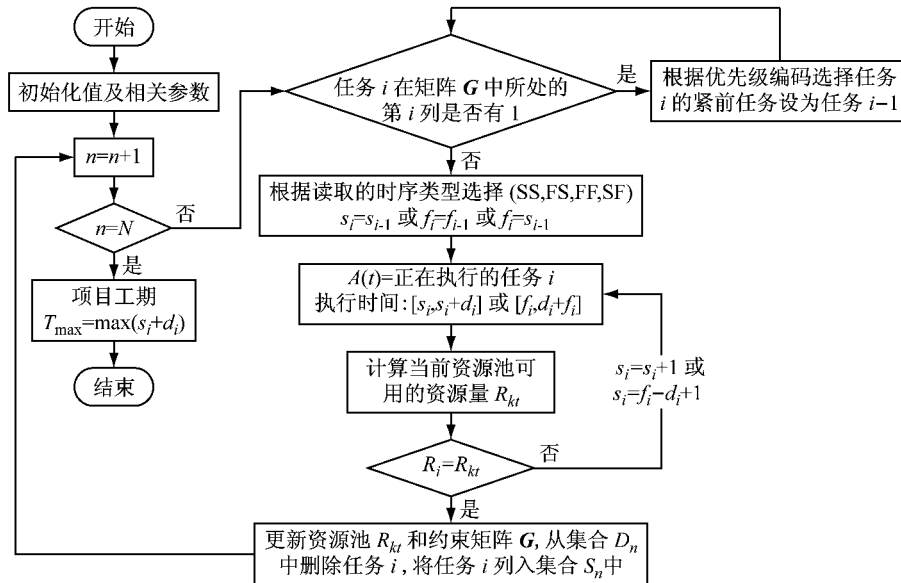


图 2 广义优先关系资源受限调度流程

Fig. 2 Resource-constrained project-scheduling with generalized precedence relations

## 2 资源约束的进度管理

实际施工进度管理需要标记出施工进度计划的关键路径. 缩短工期需要优化关键路径上的关键任

务历时和对应资源, 为了施工安全, 需要控制某些设备的集中使用强度, 在不影响工期的情况下需要均衡非关键任务对应的资源. 为了使上节资源受限项目调度算法和模型更适用于实际建设工程项目的施工进度管理, 对该模型的功能进行扩展和补充.

### 2.1 逆向反推计算关键路径

逆向反推计算关键路径的方法是结合拓扑排序、时序约束矩阵  $G$  和广义优先关系,建立关键任务的拓扑序列,形成资源约束进度计划的关键路径.其中,拓扑排序是指将不带回路的单代号网络图中的所有任务排成一个线性序列,使得每个任务的所有前驱任务都排在该任务之前,所得到的任务序列则称为拓扑序列.如果存在平行任务,则拓扑序列不是唯一排序.

在时序约束  $G$  矩阵中加入因资源冲突产生的任务逻辑关系,从最后完成的任务开始进行计算并判断(见公式(9)),直到计算到没有紧前任务.

$$s_k = \max\{s_h + l_{hk}, s_i + l_{ik}, \dots, s_j + l_{jk}\} \quad (9)$$

每个节点最晚开始的任務组成的拓扑排序即为逆向反推的关键路径.

### 2.2 关键任务的资源及历时优化

方式一,将关键路径中因资源冲突改变时序关系的任务标记出来,根据实际需求改变冲突资源在资源池中的最大供应量,缓解资源冲突,恢复任务原来的时序关系.

方式二,根据数理统计的方法对各类资源进行相关性分析,并对具有相关性的资源根据预算定额人材机的配比和施工经验设置相关系数.同理,对各个任务与其对应资源进行相关性分析并设置相关系数.在此基础上根据实际需求再对各个关键任务进行优化,即根据设定的资源利用率水平提高资源利用率,根据历时与相关资源的关系缩短关键任务历时,并验证单位时间内资源的使用量是否仍满足资源池中的最大资源约束条件.

### 2.3 非关键任务的资源均衡

根据非关键路径上任务的自由浮动时间,改变该任务的开始时间,计算资源占用发生变化的时间段内该资源最大使用量与最小使用量的差值,判断该差值改变前后的差异,若改变后差值变小则达到不影响总工期的资源均衡目标.

$$\begin{aligned} t_{ij} &= t_j - t_i \\ s'_i &= s_i \pm n, t_i \leq s_i \\ f'_i &= f_i \pm n, f'_i \leq t_j \\ \Delta R_{ij} &= R_{\max} - R_{\min} \\ \min \Delta R &= \Delta R'_{i,j} < \Delta R_{i,j} \end{aligned} \quad (10)$$

式(10)中: $t_{ij}$ 为资源均衡的时间段; $t_i$ 和 $t_j$ 为资源均衡的起止时间节点; $s_i$ 和 $f_i$ 为非关键任务未改变前的起止时间节点; $s'_i$ 和 $f'_i$ 为非关键任务改变后的起止时间节点; $n$ 为调节的时间步长; $R_{\min}$ 和 $R_{\max}$ 为时间段 $t_{ij}$ 内资源 $R_{ij}$ 的最大值和最小值; $\Delta R_{ij}$ 和 $\Delta R'_{ij}$ 为时间段 $t_{ij}$ 内改变前后的资源差.在满足自由浮动时间约束的条件下,找到该时间段内 $\min \Delta R$ 资源差最小的非关键任务的起止时间节点.

### 2.4 资源约束进度管理流程

资源约束的进度管理流程如图 3 所示.

(1)将初始化数据进行整理代入混合算法的广义优先关系资源受限项目调度模型,运算获得初步优解.

(2)对当前解进行逆向反推计算关键路径.

(3)优化关键任务的资源和历时,将优化结果代入资源受限项目调度模型,运算获得再次优解,如此循环,直到获得期望总工期,并详细记录每次数据的变化.

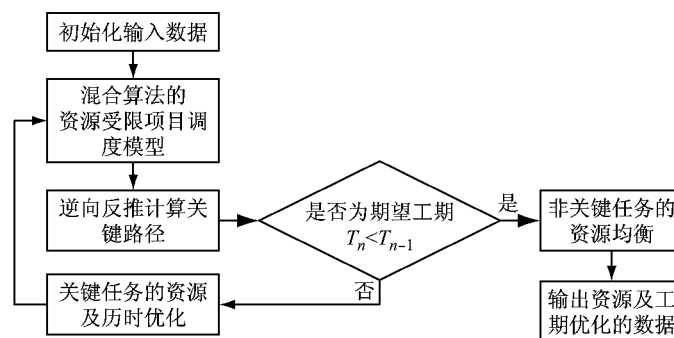


图 3 资源约束进度管理流程

Fig. 3 Resource-constrained dynamic scheduling

(4)对期望总工期最优方案中的非关键路径上的任务进行资源均衡.

该进度管理流程根据初步数据的输入获得初步的优化结果,通过对第  $n-1$  次输出数据的分析,改

进第  $n$  次的输入数据,形成一个反馈迭代优化的闭环,直到达到设定的优解条件.在实际项目施工管理中,往往是先知道某些施工任务需要在某个时间节点完成,根据可用时间合理配置资源.即已知期望工

期和工艺约束,需要获得资源配置的逆向资源受限项目调度问题的求解.可见,资源约束的进度管理方法可以实现根据实际需求的工期期望调节资源配置的管理目标.

### 3 案例应用及数据分析

对项目案例的实际工程数据进行收集整理,并根据上述算法、模型和方法进行数据运算及分析,获得辅助项目管理决策的资源及工期的图表和优化方案.

#### 3.1 案例简介及应用

某在建超高层养老院及社区卫生服务中心项目,现处于基础施工阶段.项目在基础后浇带进行开挖、浇捣、砌筑、平整等施工过程中,有大量挖出的土方需暂放在缓冲堆放区,然后通过运输设备和人员

运至集中堆放场地,同时又有大量的钢筋、混凝土等建筑材料需要运往施工操作场地.往往由于人员、运输设备、操作场地、缓冲堆放区等资源的限制,影响实际施工进度的发展.

选取 36 个施工任务和 6 个主要约束资源,其中  $R_1$  为人力资源( $R_{1max} = 30$  人); $R_2$  为水平运输设备资源( $R_{2max} = 10$  台); $R_3$  为 S 轴区域的操作场地( $R_{3max} = 95 \text{ m}^2$ ); $R_4$  为 S 轴区域的缓冲堆场区( $R_{4max} = 78 \text{ m}^2$ ); $R_5$  为 A 轴区域的操作场地( $R_{5max} = 120 \text{ m}^2$ ); $R_6$  为 A 轴区域的缓冲堆场区( $R_{6max} = 69 \text{ m}^2$ ).图 4 为在 eclipse 环境下用 Java 语言进行编程,并设置 excel 数据导入和导出接口,在界面 input 目录下的 excel 文件中导入数据,点击界面的运行按钮,实现快速输出运行结果并自动存入 output 目录下的 excel 文件中.

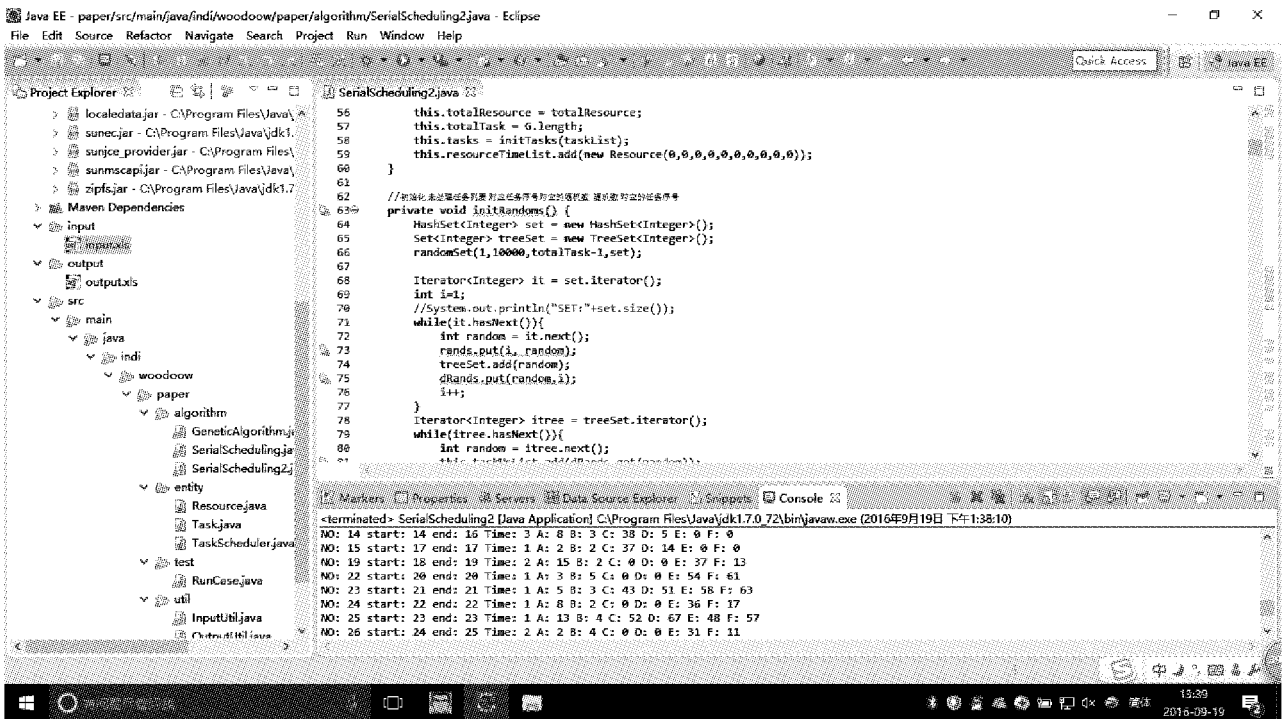


图 4 资源约束进度管理的程序运行界面

Fig. 4 Program running interface of resource-constrained schedule management

#### 3.2 运算结果及数据分析

根据差分进化人工蜂群算法和广义优先关系的资源受限项目调度模型,及以此模型为基础的资源约束进度管理方法,可获得施工项目的进度计划、约束资源分布情况、资源及工期优化方案等.由于篇幅有限,在此仅展示部分进度计划和资源优化方案.

将自动运算的输出数据 excel 表(含 36 个施工任务的开始时间和完成时间等数据)直接导入到进

度管理软件 project 中,获得进度计划横道图如图 5 所示.

##### 3.2.1 资源均衡与工期的关系

根据进度计划及逆向反推算法,标记出关键路径上的任务编号为 1、11、12、13、8、19、22、23、24、25、30、29、33、36、27、5、16、20、21、4、18、7.在不影响总工期的前提条件下对非关键路径上的任务进行资源均衡.以资源  $R_1$  为例可进行资源均衡的任务编号为

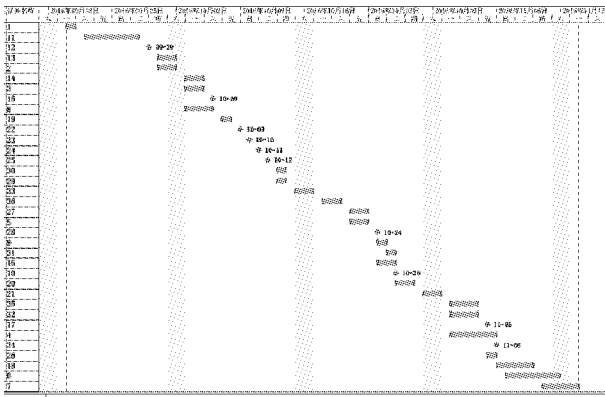


图 5 案例进度计划横道图  
Fig. 5 Gantt charts of scheduled plan in a real construction case

3 和 9,通过任务 3 和任务 9 均延后一天,使得第 14 天和第 17 天的  $R_1$  资源使用量分别从原来的 14 和 5 均衡到 11 和 8,第 35 天和第 37 天的  $R_1$  资源使用量分别从原来的 29 和 25 均衡到 27 和 27,如图 6 案例  $R_1$  资源均衡前后对比图实现了资源均衡局部填谷削峰的效果。

### 3.2.2 资源约束与工期的关系

如果该案例不受任何资源约束,在 input 输入数据中的  $R_1 \sim R_6$  的资源使用量均设为无穷大,运算结果显示所有任务完成的总工期为 31 d,而受  $R_1 \sim R_6$  资源约束的总工期为 57 d,可见多资源约束对工期影响非常大。

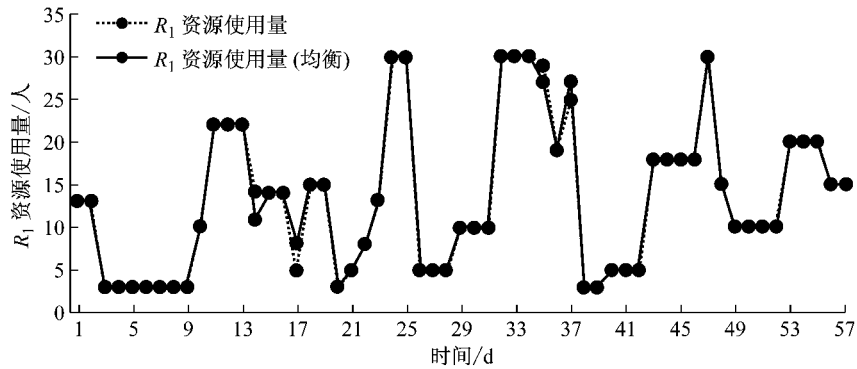


图 6 案例  $R_1$  资源均衡前后对比图  
Fig. 6 Comparison of  $R_1$  before and after resource leveling in a real construction case

### 3.2.3 资源类型与工期的关系

如果一个资源设为无穷大,其他资源保持不变: $R_1$  资源无约束,总工期为 48 d; $R_2$  资源无约束,总工期为 46 d; $R_3$  资源无约束,总工期为 52 d; $R_4$  资源无约束,总工期为 52 d; $R_5$  资源无约束,总工期为 54 d; $R_6$  资源无约束,总工期为 53 d. 运算结果显示,场地类的资源约束之间具有较强的联系,释放一个场地资源而不释放其有关联的场地资源,对施工总进度影响不大。

### 3.2.4 资源利用率与工期的关系

案例  $R_1 \sim R_6$  资源分配图中关键路径上的任务 11,在资源  $R_1, R_2$  和  $R_3$  上的资源利用率偏低. 根据实际项目进展的需求对各个资源及任务历时要素的相关性分析和相关系数设定,以此获得增加资源和缩短历时的优化处理,如图 7 所示. 处理后使得任务 11 的资源利用率增加近 1 倍,总工期提前 3 天,如表 1 所示。

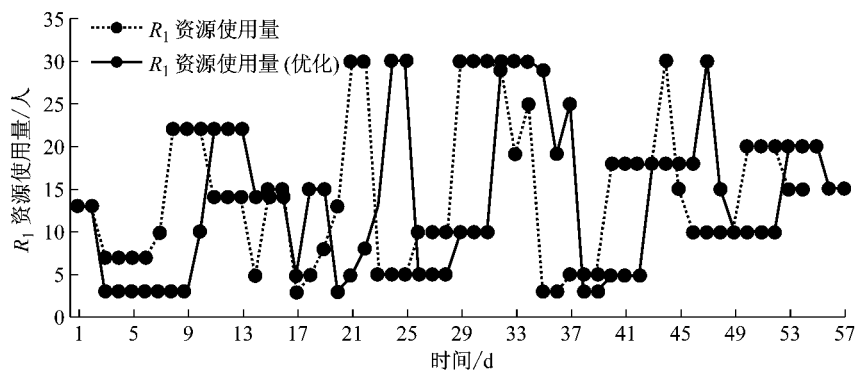


图 7 调整任务 11 的资源  $R_1$  优化前后对比  
Fig. 7 Comparison of  $R_1$  before and after resource improvement by adjusting task NO.11

表 1 调整任务 11 的资源利用率与工期优化结果

Tab. 1 Results of resource utilization ratio and duration optimization by adjusting task NO.11

资源进度计划	总工期/ d	R <sub>1</sub> 资源 利用率/%	R <sub>2</sub> 资源 利用率/%	R <sub>3</sub> 资源 利用率/%
任务 11 优化前	57	10.00	40.00	36.84
任务 11 优化后	54	23.33	80.00	68.42

可见,对相同情况的任务进行类似的优化处理,能提高整个进度计划的资源利用率和总工期.总之,文中提到的算法、模型和方法能快速获得资源约束的进度计划和约束资源的分布情况,可根据项目进展的需求,平衡并优化工期和资源的关系.

## 4 结论与展望

根据资源受限项目调度问题和仿生算法的文献分析,结合实际项目施工管理的现状,提出对多组优先级编码的初始化值进行并行计算,差分进化算法改进蜂群,人工蜂群算法避免领域搜索,以串行调度产生方案结合广义优先关系获得的工期作为适应度值判断差分进化人工蜂群算法中优先级编码的优劣,以此形成循环、迭代、搜索的寻优过程.建立含逆向反推计算关键路径、关键任务的资源和历时的相关性分析及优化,以及非关键任务资源均衡的资源约束进度管理方法.在 eclipse 环境下用 Java 语言进行编程,并设置 excel 数据导入和导出接口,针对需要控制的施工资源进行有效管理.如运输设备、操作场地、运输人员等施工资源在满足施工工艺约束的条件下,获得符合施工管理目标的资源约束进度计划、资源分布情况,以及资源与工期的优化方案.

在建设信息模型 (building information modeling, BIM) 加进度计划的 4D 软件 (如 Synchro 软件) 中,通过将施工进度计划和 BIM 模型挂接突显因空间资源冲突导致任务工艺冲突,解除发现的工艺冲突而优化施工进度计划.文中提出结合广义优先关系的改进差分进化人工蜂群算法不仅能处理空间资源冲突而导致任务工艺冲突,还能处理更多类型的资源冲突.可见,资源约束进度管理方法,今后可用于 4D 软件的二次开发,强化 BIM 模型加进度计划加资源配置的项目施工管理.

### 参考文献:

[1] 聂慧. 建设工程施工项目人力资源调度的模型与方法研究

[D]. 北京:中国矿业大学, 2014.

NIE Hui. Study on models and methods for workforce scheduling of construction projects [D]. Beijing: China University of Mining and Technology, 2014.

- [2] TRAN D H, CHENG M Y, CAO M T. Solving resource-constrained project scheduling problems using hybrid artificial bee colony with differential evolution [J]. Journal of Computing in Civil Engineering, 2016, 30(4): 04015065.
- [3] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm [J]. Journal of Global Optimization, 2007, 39(3):459.
- [4] KARABOGA D, AKAY B. A comparative study of artificial bee colony algorithm [J]. Applied Mathematics & Computation, 2009, 214(1):108.
- [5] LI X, YIN M. Hybrid differential evolution with artificial bee colony and its application for design of a reconfigurable antenna array with discrete phase shifters [J]. IET Microwaves Antennas & Propagation, 2012, 6(6):1573.
- [6] LI Y, WANG Y, LI B. A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow [J]. International Journal of Electrical Power & Energy Systems, 2013, 52(11):25.
- [7] GAO W, LIU S. Improved artificial bee colony algorithm for global optimization [J]. Information Processing Letters, 2011, 111(11):871.
- [8] GONZÁLEZ-ÁLVAREZ D L, VEGA-RODRIGUEZ M A. Hybrid multi-objective artificial bee colony with differential evolution applied to motif finding [C] // European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. Berlin: Springer-Verlag, 2013:68-79.
- [9] HAN Y Y, GONG D, SUN X. A discrete artificial bee colony algorithm incorporating differential evolution for the flow-shop scheduling problem with blocking [J]. Engineering Optimization, 2015, 47(7):927.
- [10] SCHNELL A, HARTL R F. On the efficient modeling and solution of the multi-mode resource-constrained project scheduling problem with generalized precedence relations [J]. OR Spectrum, 2016, 38(2):283.
- [11] BIANCO L, CARAMIA M, GIORDANI S. Resource levelling in project scheduling with generalized precedence relationships and variable execution intensities [J]. OR Spectrum, 2016, 38(2):405.
- [12] PRICE K V, STORN R M, LAMPINEN J A. Differential evolution [J]. Springer Optimization & Its Applications, 2005, 141(2):1.
- [13] KARABOGA D, BASTURK B. On the performance of artificial bee colony (ABC) algorithm [J]. Applied Soft Computing, 2008, 8(1):687.
- [14] CHENG M Y, TRAN D H, WU Y W. Using a fuzzy clustering chaotic-based differential evolution with serial method to solve resource-constrained project scheduling problems [J]. Automation in Construction, 2014, 37(1): 88.