

粒子群优化算法在柔性资源受限项目调度中的研究

陈君兰 叶春明
(上海理工大学 上海 200093)

摘要 为了更有效地解决柔性资源受限项目调度问题,建立了速熟练度的技能供给矩阵,并应用混沌粒子群优化算法来满足工序的先后约束关系,以在技能供给受限的情况下形成优先规则序列,根据串行进度生成机制形成该序列下的最优解,运用嵌入混沌理论的粒子群优化算法更新种群,寻得全局最优解。实验结果验证了混沌粒子群优化算法求解该问题的可行性和有效性,对于项目管理中柔性资源受限问题具有实际应用价值。

关键词 柔性资源,粒子群,项目调度,串行进度生成机制,混沌

中图分类号 TP391 **文献标识码** A

Flexible-resource Constrained Project Scheduling Research Based on Particle Swarm Optimization

CHEN Jun-lan YE Chun-ming

(University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract In order to better solve the flexible-resource constrained project scheduling problem, matrix with proficiency was established to show the relations between resources and skills, and CPSO (Chaos Particle Swarm Optimization) was used in this essay to solve this problem. In consideration of work's priorities and flexible-resource constrained problem, a randomized priority chain was formed and followed by an optimized result based on Serial Schedule Generation Scheme (SSGS) and the best schedule of the whole project was found by updating the population by CPSO. Results prove the possibility and effect of this method in solving this problem. Therefore, this method has its practical application value for the flexible-resource constrained project scheduling problem.

Keywords Flexible-resource, Particle swarm optimization(PSO), Project scheduling, Serial schedule generation scheme (SSGS), Chaos

1 引言

资源受限项目调度问题(Resource-Constrained Project Scheduling Problem, RCPSP)是在资源前后约束下带有最小化项目持续时间目标的项目调度问题,也就是资源限制条件下的工期最短问题^[1]。柔性资源指存在具有不同技能的有限资源。由于完成一项工序需要不同的技能,因此如何在资源不冲突的情况下合理安排各个工序,使整个项目的最终完成时间最短的问题就是柔性资源受限的项目问题。

求解资源受限的问题从传统的关键路径法(CPM)、计划评审技术(PERT)发展到了以分枝定界法为代表的精确算法、基于优先规则的启发式算法以及智能算法等。但是精确算法求解问题规模小,因此启发式算法成为研究热点。

在资源受限问题上,刘士新^[6]总结了目前一些基本项目调度问题的模型及一些求解方法的证明。寿涌毅^[7]总结了一些基于优先规则求解基本项目调度问题的方法,并提出了求解多项目调度问题的拍卖理论。而柔性资源受限问题领域的研究主要集中在算法的研究上,喻小光^[5]等使用 Burgess 和

Killebrew 提出的资源平滑算法^[11]对串行调度形成的调度顺序进行调整形成改进的串行调度,并验证了其在资源水平调度问题中的有效性。黄敏镁^[3]等构造了一种结合遗传算法和最大流的改进遗传算法,其通过基于优先权的自然数编码和拓扑排序^[12]产生先后关系可行任务链表,并将算法运用到产品开发项目的例子中,由此验证了算法的有效性。而在混沌粒子群优化算法方面,刘军民^[4]将混沌算法嵌入粒子群算法来求解基准函数的典型问题,并证明比 PSO 算法其具有更高的收敛性。王华秋^[13]提出了并行混沌粒子群算法并证明其更快速收敛于全局最优解。

本文采用混沌粒子群优化算法生成优先规则,并结合串行调度生成机制来求解柔性资源受限项目调度问题。

2 问题描述

一个项目由工序集 J 组成,每项工作用 j 表示, $j=1, 2, \dots, n$, 表示该项目共有 n 道工序,每道工序都有固定的执行时间,用 TD_j 表示,且工序间都有相应的先后关系约束,而且每道工序需要使用一定数量的不同技能的资源,资源集合用 R

到稿日期:2012-04-05 返修日期:2012-07-08 本文受教育部人文社会科学规划基金项目(10YJA630187),高等学校博士点基金(20093120110008),上海市重点学科建设项目(S30504),上海市教育委员会科研创新项目(12ZS133),上海市研究生创新基金项目(JWCXSL1102)资助。
陈君兰(1988-),女,硕士,主要研究方向为项目管理,E-mail:tracychen0602@163.com;叶春明(1964-),男,博士,教授,主要研究方向为工业工程、企业战略、企业生产计划与控制、生产调度、企业资源计划(ERP)、供应链管理以及企业信息化。

表示。设项目工期的上限为 SUM, 根据各工序的执行时间及先后关系约束, 可计算出各工作的最早、最晚开始时间窗口 $[ES_j, LS_j]$, TS_j 表示工序 j 的开始时间。根据文献[1]提到的方法建立资源-技能矩阵, 设项目的柔性资源的集合为 H , 每种资源用 h 表示, $h=1, 2, \dots, m$, 则柔性资源的总量为 m ; 设项目共有的所需技能集合为 S , 每种技能用 $s=1, 2, \dots, v$ 表示, 则共有技能 v 种; 由此, 可构成技能能力矩阵 H , 这是一个 $m \times v$ 的矩阵行表示资源, 列表示技能, 整个矩阵表示每种资源拥有技能的情况, ρ_{hs} 表示资源 h 拥有技能 s 的能力, 由于前提假设每项资源在同一时间内只能进行一道工序的加工, 且资源对不同技能的熟练程度也各不相同, 因此所构成的技能能力矩阵中的数值在 $[0, 1]$ 间, 0 表示某一资源不具有某一技能, 非 0 表示某一资源掌握某一技能的能力, 其中:

$$\rho_{hs} = \begin{cases} 0, & \text{资源 } h \text{ 不具有技能 } s \\ (0, 1], & \text{资源 } h \text{ 具有技能 } s \text{ 的能力} \end{cases} \quad (1)$$

$$\varphi_{hs} = \begin{cases} 0, & \rho_{hs} = 0 \\ 1, & \rho_{hs} \neq 0 \end{cases} \quad (2)$$

由于项目中的每道工序需要使用一定数量的不同技能的资源, 每道工序对每种技能的需求量用 r_{js} 表示, 因此在分配工序 j 的开始时间时, 要考虑在 $[TS_j, TS_j + TP_j]$ 时间段内空闲且具有所需技能的资源量是否能满足该工序的技能需求。所以问题的数学模型如下:

$$\min TS_n \quad (3)$$

$$TS_j - TS_i \geq TD_i, i \in IP_j \quad (4)$$

$$\sum_{h=1}^m x_{ht} \cdot \varphi_{hs} \geq r_{js}, j=1, 2, \dots, n; s=1, 2, \dots, v; t \in [TS_j, TS_j + TD_j] \quad (5)$$

$$\sum_{s=1}^v r_{js} \leq \sum_{h=1}^m x_{ht} \cdot j \in A_t, t \in [TS_j, TS_j + TD_j] \quad (6)$$

式中, r_{js} 表示在 t 时刻工序 j 对技能 s 的需求量,

$$x_{ht} = \begin{cases} 1, & t \text{ 时刻资源 } h \text{ 可用} \\ 0, & t \text{ 时刻资源 } h \text{ 不可用} \end{cases} \quad (7)$$

式中, IP_j 表示工作 j 的紧前工作集合, IS_j 表示工作 j 的紧后工作集合, A_t 表示 $(t-1, t)$ 时间段内正在进行的工作集合, 式(3)为目标函数, 式(4)为前后关系时间约束, 式(5)表示任意工序加工时间段内具有该工序所需技能且可用的资源数不小于该工序对技能的需求量, 式(6)表示任意时段各种技能总需求量不能超过资源对技能的总供给量, 式(7)表示资源在 t 时刻是否可用。

但是可能存在多个资源同时满足工序的需求, 从而也就存在多种资源分配方案, 如果把每种资源分配方案表示成一种任务执行模式, 可以将柔性资源约束下的项目调度问题转化为多模式资源受限项目调度问题(MRCPSP), 问题的数学模型类似 MRCPSP 的数学模型。

3 算法原理

粒子群优化算法(PSO)是 1995 年由 Eberhart 和 Kennedy 提出的, 它模拟鸟群飞行觅食的行为, 通过信息的交流寻找到最优解。混沌算法具有遍历性、随机性等特点, 因此将混沌算法嵌入粒子群优化算法中。

3.1 混沌模型

目前较为成熟的混沌模型之一为生长模型, 它于 1976 年

由美普林斯顿大学生物系的 May 提出, 而对混沌的严格定义尚不明确, 一般将由确定性方程得到的具有随机性的运动状态称为混沌。Logistic 方程为:

$$x_{n+1} = \mu x_n (1 - x_n), n=0, 1, \dots \quad (8)$$

式中, μ 为控制参数, $x_0 \in [0, 1]$ 为初始值。

利用混沌遍历性和随机性的特点, 对传统粒子群优化算法进行改进, 避免传统粒子群优化算法陷入局部最优。混沌粒子群算法主要是在基本粒子群算法中加入混沌初始化, 主要思想是形成一组与优化变量相同数目的混沌变量作为初始粒子位置, 并同时运用并行的 CPSO 方法, 增大搜索范围。

3.2 粒子群优化算法

在 PSO 模型中将每个搜索空间中的可行解称为粒子, 每个粒子都有其对应的位置和速度, 通过粒子速度的变化对粒子位置进行调整从而寻得最优解。

假设粒子 i 用 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 表示, 其中 D 为粒子的维度, 它经过代数进化后所经历的最好位置称为局部最优解, 用 $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ 表示; 而整个种群(由多个粒子构成)在经过代数进化后所遍历得到的最优解称为全局最优解, 用 $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 表示, 每个粒子的速度用 $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 表示, 其可设定速度的最大值和最小值, 控制速度更新的范围。更新方程如下:

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (p_{ij}^t - x_{ij}^t) + c_2 r_2 (p_{gj}^t - x_{ij}^t) \quad (9)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (10)$$

式中, c_1, c_2 为学习因子, t 为代数, 式(9)为速度更新方程, 式(10)为位置更新方程。 ω 为惯性因子, 按下式取值:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{D_{\max}} t \quad (11)$$

式中, $\omega_{\max}, \omega_{\min}$ 分别为最大和最小惯性权重, D_{\max} 为最大进化代数。

3.3 串行进度生成机制

进度生成机制是多数资源受限问题的核心, 分为串行进度生成机制和并行进度生成机制。串行进度机制以阶段区分, 在每个阶段中, 选取一个工序从该工序的最早开始时间起对工序进行调度, 在满足资源约束的条件下, 尽可能早地安排工序, 并将已调度的工序放入已调度工序的集合中, 并将该工序从未调度工序集合中删去。

4 求解问题的算法

4.1 个体编码及初始化

随机生成满足资源量约束和技能约束的条件的资源-技能矩阵。

初始化所有工序开始时间为 0。

混沌初始化时, 采用式(12), 对应每一道工序随机产生一组 n 维的 $[0, 1]$ 间的数作为混沌变量, 并利用式(8)更新混沌序列, 计算目标函数, 从中选择较好的 POPSIZE 个粒子作为粒子初始位置。其中虚开始工序对应 0, 虚结束工序对应 1, $\text{random}(a, b)$ 表示在 a, b 间随机生成一个小数, $x_s = (x_{s1}, x_{s2}, \dots, x_{sn})$, 粒子位置用 x_s 表示,

$$x_s = \text{random}(0, 1) \quad (12)$$

粒子速度初始化时,采用式(13),在 $[-1,1]$ 间随机取值, $x_v=(x_{v1},x_{v2},\dots,x_{vn}),x_v$ 表示粒子速度,

$$x_v = \text{random}(-1,1) \quad (13)$$

4.2 调度生成

4.2.1 工序的最早、最晚开始时间

设项目工期上限 SUM 为各工序串联情况下工期的总和,由此,设虚开始工序的最早开始时间为 0,再根据各工序的工期和前后约束关系,可计算出各工序的最早、最晚开始时间 $[ES_j,LS_j]$ 。

4.2.2 优先规则形成

对每个粒子中的 n 个数(对应工序号)重新按从小到大排序。之后,根据工序的前后关系约束,对形成的排序进行调整,从形成序列的第二个工序开始,检查是否有该工序的紧后工序出现在该工序在序列中的位置前,如果有出现,则将这两个工序位置互换,如此循环 $n-2$ 次,使每道工序的顺序都满足其前后关系约束。而形成的新顺序,就是调度的优先规则序列。

4.2.3 资源优先选择策略

本文提出的技能能力矩阵表示某一资源掌握某技能的情况,因此在选择资源完成工序时,优先选择具有该项技能高能力值的资源来完成工序。

4.2.4 串行进度生成机制

本文采用串行进度生成机制。从优先规则链表的第一项工序开始调度。

首先,从优先规则链表上的第一道工序 j 开始,令 $TS_j = ES_j$,在从该时刻起的工作时间内 $[TS_j,TS_j+TD_j]$,从该工序所需要的每项技能 i 依次算起,找出会该技能且处于空闲状态下的所有资源集合,并根据 ρ_{is} 的大小对这些资源进行从大到小排序得到集合 hh_{ij} ,若 $\text{length}(hh_{ij}) \geq R(j,i)$,则从 hh_{ij} 中依次选出该工序该技能所需的资源,构成集合 hp_{ij} ,若该工序所需的任意技能的资源集合 hh_{ij} 不能满足上述约束,则时刻下移一格,即 $TS_j = TS_j + 1, TS_j \leq LS_j$,直至找到满足条件的资源为止。

其次,将该工序用于某技能的所有选中资源 hp_{ij} 在这段时间内标记为活动状态,此时的 TS_j 即为该工序调度后的开始时间,更新该工序紧后工序的最早开始时间, $TS_{is_j} = \max\{TS_{is_j}, TS_j + TD_j\}$,并进入下一个工序的调度。

再次,当所有工序都完成调度之后,更新工序开始时间,由于 TS_j 在 $[TS_j,LS_j]$ 中搜索解,因此当在此范围内无法找到合适的解时,该工序的开始时间就是初始化时的时间 0。此时需对形成的调度依次检查,若发现存在某工序的开始时间为 0(不包括虚开始工序),则将此调度的适应值记为无限大,以排除不符合要求的解。

最后,对于合格的调度,计算资源使用率,即某种资源在总工期内处于活动状态的概率,从而控制所形成调度的资源使用率。记录调度工期,由此完成了对一个粒子位置的调度。进入下一粒子位置适应度的计算,计算方法与前面相同。

在串行进度生成机制中,优先规则链表中的每一道工序都在该工序允许范围内将工序开始时间尽可能安排在最早的

时间,而且优先规则链表中的工序调度顺序符合工序的前后约束关系,从而形成该优先规则下的最优化调度,而后通过 4.3 节中的种群更新策略更新优先规则链表,从而形成总项目的最优调度。

4.3 种群更新策略

利用粒子群优化算法协同搜索性强、具有很强的记忆能力来保持局部和全局最优信息的特点,采用粒子群更新式(9)、式(10)分别对粒子位置和速度进行更新,但是,由于粒子速度控制在 $[-1,1]$ 间,因此对于超出边界的粒子速度取边界值,同时,由于粒子位置需控制在 $[0,1]$ 间,因此对于超出边界的粒子位置取边界值。此外由于粒子群优化算法易陷入局部最优解,故当有可能陷入局部最优时,采用混沌算法,利用式(10)更新粒子位置。此外,同时继续运用式(8)对种群混沌化。

4.4 混沌粒子群算法步骤

粒子群优化算法有易陷入局部最优解的缺陷,因此在粒子群优化算法中嵌入混沌算法,利用混沌算法具有遍历性和随机性的特点降低陷入局部最优解的风险。由此,求解柔性资源受限问题的 CPSO 算法具体步骤如下:

步骤 1 随机生成资源-技能矩阵,混沌初始化形成粒子位置和粒子速度。计算各工序最早开始时间、最晚开始时间,并记录初始粒子个体极值和粒子群全局极值。

步骤 2 以目前粒子位置作为混沌变量,利用式(8)更新混沌序列,更新优先规则链表,并计算、评价满足柔性资源约束下的适应度 $F1$ 。

步骤 3 更新粒子群,利用式(9)、式(10)分别更新粒子位置和速度,更新优先规则链表,并计算、评价满足柔性资源约束下的适应度 $F2$,对于超出最大允许工期的解或不符合要求的解,其适应度为无限大,以此排除非法解。

步骤 4 比较两个适应度 $F1$ 和 $F2$,保留较优的解,更新粒子个体极值和粒子群全局极值。

步骤 5 若达到最大迭代次数,则返回全局最优解 $gbest$;否则跳至步骤 2。

5 实例研究

本文对国际标准问题库 <http://129.187.106.231/psplib/> 中的 J3022.2.sm 进行改编的算例,来证明算法的有效性。

本文采用 MATLAB7.0 对以上算法编程,粒子群优化算法的参数设置如下:种群数 POPSIZE=20,学习因子 c_1, c_2 为 2,设进化次数为 500 次,粒子速度取值范围 $[-1,1]$, w 取值范围为 $[0.8,1.2]$ 。本文采用 CPSO 算法求解该实例。

对国际标准问题库进行改编,项目作业信息表如表 1 所列,然后自动生成资源技能矩阵 20 个,资源技能矩阵生成原则符合最大资源为 40(由表 1 中所列工序对技能在同一时刻的最大需求为 30),所以每个资源技能矩阵的行数为 40(表示最大资源量 40 个),列数为 4(表示 4 种技能),且保持生成的随机矩阵满足技能约束条件,样式见表 2,其能技能 i 的资源量表示为 $H_{\max}(i), H_{\max}(1)=12, H_{\max}(2)=24, H_{\max}(3)=19,$

$H_{\max}(4)=19$,在这 20 个随机矩阵下进入测试,并假定无资源使用率约束。

表 1 作业信息表

工序号	工期	紧后工序	技能需求			
			技能 1	技能 2	技能 3	技能 4
1	0	2,3,4	0	0	0	0
2	1	11	2	0	9	0
3	5	5,7,15	2	8	3	0
4	7	6,8,9	6	7	0	0
5	10	17,19	0	0	1	0
6	1	10,13,24	0	8	0	0
7	5	12,16,17	0	5	0	0
8	1	13,18,23	6	0	0	6
9	5	11,22,24	0	3	1	0
10	9	26,30	0	0	0	10
11	3	14	1	9	7	0
12	7	13,29	2	10	0	8
13	4	21	0	0	3	0
14	2	19,21	7	10	0	0
15	7	16	4	7	3	1
16	6	18,28	0	7	9	0
17	7	18,20	3	10	7	10
18	8	21,22	10	0	10	10
19	8	25	3	0	0	6
20	8	22,23,28	0	4	6	0
21	4	26	0	1	1	6
22	5	27	0	0	4	0
23	3	27,31	0	2	0	7
24	5	25,26	0	1	10	0
25	10	27,28	0	0	3	2
26	3	31	4	0	1	3
27	4	29,30	7	0	0	0
28	3	29	0	1	3	0
29	1	32	0	2	0	0
30	3	32	0	0	6	6
31	1	32	0	7	0	0
32	0	0	0	0	0	0

经过计算求得在资源技能矩阵表 2 的情况下,整个工期达到最小工期 58 天,部分项目的最优调度如表 3 所列。

表 2 资源技能信息表

资源	技能 1	技能 2	技能 3	技能 4	资源	技能 1	技能 2	技能 3	技能 4
1	0	0.4	0	0.6	21	0.1	0.5	0	0.4
2	0	0	0.4	0.6	22	0	0	1	0
3	0.5	0	0	0.5	23	0	0.2	0.8	0
4	0.5	0.2	0	0.3	24	0	1	0	0
5	0	0.5	0.5	0	25	0	0.4	0	0.6
6	0.5	0	0	0.5	26	0.3	0.3	0.4	0
7	1	0	0	0	27	0	0.1	0.7	0.2
8	0	0	1	0	28	0.1	0.6	0.2	0.1
9	0	0.7	0.3	0	29	0	1	0	0
10	0	0.6	0	0.4	30	0	0	0	1
11	0	0	0	1	31	0	0.4	0	0.6
12	0.9	0	0.1	0	32	0	0	1	0
13	0	0	0.2	0.8	33	0	0	0	1
14	1	0	0	0	34	0	1	0	0
15	0	1	0	0	35	0	0.6	0.4	0
16	0	0.5	0	0.5	36	0	0	1	0
17	0	1	0	0	37	0	0.6	0.1	0.3
18	0	0.5	0.5	0	38	1	0	0	0
19	0	0.4	0.6	0	39	0.6	0	0.4	0
20	0	0.4	0.5	0.1	40	0.1	0.5	0	0.4

表 3 项目调度方案

工序号	开始时间	技能编号	资源编号
1	0	—	—
2	1	1	4,3
		3	36,32,22,8,23,27,19,20,18
		1	4,3
3	2	2	35,28,10,40,21,18,16,5
		3	36,32,22
		1	38,14,7,12,39,6
4	1	2	34,29,24,17,15,9,37
		3	23
5	7	3	34,29,24,17,15,9,37,31
6	8	2	34,29,24,17,15
7	9	2	6,7,12,14,38,39
8	8	1	1,2,11,13,27,30
		4	25,20,19
9	8	2	8
		3	—
10	30	4	33,30,11,13,31,25,2,1,16,6
		1	7
11	14	2	40,21,18,16,5,20,19,26,4
		3	8,12,22,27,32,36,39
		1	38,14
12	14	2	34,29,24,17,15,9,37,35,28,10
		4	33,30,11,13,31,25,2,1
13	21	3	35,2,9
14	21	1	3,4,6,7,14,28,38
		2	1,10,15,17,24,25,29,31,34,37
		1	4,3,26,40
15	7	2	35,28,10,21,18,16
		3	36,32,22
		4	33
16	17	2	40,21,18,16,5,20,19
		3	8,12,22,23,26,27,32,36,39
		1	38,14,7
17	23	2	34,29,24,17,15,37,28,10,40,21
		3	36,32,22,8,23,27,19
		4	33,30,11,13,31,25,1,16,6,3
		1	38,14,7,12,39,6,4,3,26,40
18	39	3	36,32,22,8,23,27,19,20,18,5
		4	33,30,11,13,31,25,2,1,16,21
		1	38,14,7
19	30	4	3,40,21,10,37,4
		2	34,29,24,17
20	30	3	36,32,22,8,23,27
		2	34
21	47	3	36
		4	33,30,11,13,31,25
22	47	3	32,22,8,23
23	47	2	29,24
		4	2,1,16,6,3,40,21
24	30	2	15
		3	5,9,12,18,19,20,26,28,35,39
25	38	3	35,9,28
		4	10,37
		1	38,14,7,12
26	51	3	36
		4	33,30,11
27	52	1	39,6,4,3,26,40,28
28	48	2	17
		3	27,19,20
29	56	2	34,29
		3	36,32,22,8,23,27
30	56	4	33,30,11,13,31,25
31	54	2	34,29,24,17,15,9,37
32	59	—	—

这种资源技能矩阵情况下,两种算法的运行结果比较如
(下转第 283 页)

[6] Bouguila N, Ziou D, Vaillancourt J. Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application[J]. IEEE Transactions On Image Processing, 2004, 13(11):1533-1543

[7] Fan S. Image thresholding using a novel estimation method in generalized gaussian distribution mixture modeling[J]. Neuro-computing, 2008, 72(1-3):500-512

[8] Zribi M, Ghorbel F. An unsupervised and non-parametric Bayesian classifier[J]. Pattern Recognition Letters, 2003(24):97-112

[9] Cencov N N. Evaluation of an unknown distribution density from observations[J]. Soviet Mathematics, 1962(3):1559-1562

[10] Liu Zhe, Xiao Jian-guo, Song Yu-qing. Image segmentation based on non-parametric mixture model of Legendre orthogonal polynomial[J]. Application Research of Computers, 2010, 27(8):3165-3167

[11] Liu Zhe, Song Yu-qing, Chen Jian-mei, et al. Image Segmentation based on Finite Mixture Models of Non-parametric Chebyshev Orthogonal Polynomials of the Second Kind[J]. Journal of Com-

puter Research and Development, 2011, 11:2008-2014

[12] Sun J. m Dimension Fourier Series over Domains with $m+1$ Direction Mesh[R]. Technical Report, 2002:11-17

[13] Sun J, Li H. Generalized Fourier transform on an arbitrary triangular domain[J]. Advances in Computational Mathematics, 2005(22):223-248

[14] Sun J. Multivariate Fourier transform methods over simplex and super-simplex domains[J]. Journal of Computational Mathematics, 2006(24):55-66

[15] Sun J. Approximate eigen-decomposition preconditioners for solving numerical PDE problems[J]. Applied Mathematics and Computation, 2006(172):772-787

[16] Asmar N H. Partial Differential Equations with Fourier Series and Boundary Value Problems (2nd Edition) [M]. Prentice Hall, 2004

[17] Song T, Gasparovic C, Andreassen N, et al. A hybrid tissue segmentation approach for brain MR images[J]. Medical and Biological Engineering and Computing, 2006, 44(3):242-249

(上接第 244 页)

表 4 所列。

表 4 PSO 与 CPSO 算法的运行结果

算法	距最优方案的平均距离	达到最优值次数的百分比(%)
PSO	1.15	0
CPSO	0.76	46

由表 4 运行结果看出, CPSO 比 PSO 算法在解决该问题时更有效, 距离最优值的平均距离比 PSO 的算法结果好。而且可以达到比 PSO 算法更优的解, 避免其陷入局部最优。

根据图 1 结果可以看出在资源使用不受限情况下运用 CPSO 算法计算得到的项目最小工期为 61 天, 而运用 PSO 算法计算得到的项目最小工期却为 62 天, 从而说明 CPSO 算法可以在一定程度上避免算法陷入局部最优, 且收敛更快。

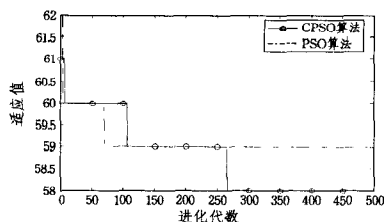


图 1 PSO 与 CPSO 收敛性对比

结束语 本文通过粒子群优化算法的原理产生和更新优先规则, 并运用串行调度生成机制的原理, 在资源受限的情况下, 合理安排资源执行相应工序, 求出最优解。此外, 为避免算法陷入局部最优, 利用混沌初始化形成的混沌序列作为粒子初始位置, 并且采用并行的混沌粒子群算法, 有效增加了算法的全局寻优能力, 从而寻得最优解。通过实例研究发现, 采用这种算法能较快寻找到最优解, 验证了算法的有效性。但是在实际情况中, 出于人性化方面的考虑, 还应考虑到资源使用间隔、影响资源疲劳度的各种因素、平均作业时间等, 这些可在以后进一步研究。

参考文献

[1] 黄敏镁, 罗荣桂. 柔性资源约束下的产品开发项目优化调度研究[J]. 管理工程学报, 2010, 4:143-147

[2] Boctor F F. A New and Efficient Heuristic for Scheduling Projects with Resource Restrictions and Multiple Execution Modes[J]. European Journal of Operational Research, 1996, 90:349-361

[3] Özdamar L, Ulusoy G. A Local Constraint Based Analysis Approach to Project Scheduling under General Resource Constraints[J]. European Journal of Operational Research, 1994, 79:287-298

[4] 刘军民, 高岳林. 混沌粒子群优化算法[J]. 计算机应用, 2008, 28(2):322-325

[5] 喻小光, 战德臣, 聂兰顺, 等. 柔性资源约束的资源水平项目调度问题[J]. 计算机集成制造系统, 2010, 16(9):1967-1976

[6] 刘士新. 项目优化调度理论与方法[M]. 北京:机械工业出版社, 2006

[7] 寿涌毅. 资源受限多项目调度的模型与方法[M]. 杭州:浙江大学出版社, 2010

[8] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proc. IEEE int'l Conf. on Neural Networks Vol. IV, IEEE Service Center, Piscataway, NJ, 1995:1942-1948

[9] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]// 1999 Congress on Evolutionary Computation CEC'99. IEEE Service Center, 1999:1945-1949

[10] Alcaraz J, Maroto C. A robust genetic algorithm for resource allocation in project scheduling[J]. Annals of Operations Research, 2001, 102(1-4):83-109

[11] Burgess, Killebawb. Variation in activity level on a cyclical arrow diagram[J]. Journal of Industrial Engineering, 1962, 13(2):7683

[12] 玄光男, 程润伟. 遗传算法与工程优化[M]. 北京:清华大学出版社, 2004

[13] 王华秋, 曹长修. 并行混沌粒子群优化研究及应用[J]. 计算机仿真, 2005, 22(11):98-101