

文章编号: 1001-0920(2010)05-0789-05

## 多项目多资源均衡问题及其基于 Pareto 的向量评价微粒群算法

郭 研<sup>1,2</sup>, 李 南<sup>1</sup>, 李兴森<sup>2</sup>

(1. 南京航空航天大学 经济与管理学院, 南京 210016; 2. 浙江大学 宁波理工学院, 浙江 宁波 315100)

**摘 要:** 首先针对多项目多资源均衡问题的特点, 建立描述问题的多目标优化模型; 然后将 Pareto 方法嵌入向量评价微粒群算法 (VEPSO), 提出一种新的基于 Pareto 的向量评价微粒群算法 (VEPSO-BP); 最后利用一个算例测试了 VEPSO-BP 的性能, 并与 VEPSO 进行了对比. 实验结果表明, VEPSO-BP 的收敛性能优于 VEPSO, 实现了对多项目多资源均衡问题的高质量求解.

**关键词:** 微粒群算法; Pareto 最优; 资源均衡; 多项目管理

**中图分类号:** F407.9

**文献标识码:** A

## Multi-resource leveling in multiple projects and vector evaluated particle swarm optimization based on Pareto

GUO Yan<sup>1,2</sup>, LI Nan<sup>1</sup>, LI Xing-sen<sup>2</sup>

(1. College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; 2. Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China. Correspondent: GUO Yan, E-mail: gunyanbox@126.com)

**Abstract:** Based on the characteristics of multi-resource leveling in multiple projects scheduling problem, a multi-objectives optimization model is setup. By applying Pareto optimal method into vector evaluated particle swarm optimization (VEPSO), a new vector evaluated particle swarm optimization based on Pareto (VEPSO-BP) is proposed. Finally, the performance of VEPSO-BP is tested with a testing example which is compared with VEPSO. Experiment results show that, VEPSO-BP is better than VEPSO in convergence efficiency, which also performs well in solving multi-resource leveling in multiple projects scheduling problem.

**Key words:** Particle swarm optimization; Pareto optimal; Resource leveling; Multiple projects management

### 1 引 言

资源均衡问题是网络计划技术中的一类具有重要经济意义的典型问题. 目前, 国内外对于资源均衡问题的研究主要集中在以下 3 方面:

1) 单项目单资源均衡. 这类问题较为简单, 研究也较为成熟, 主要算法有启发式算法、遗传算法<sup>[1]</sup>和微粒群算法<sup>[2]</sup>.

2) 单项目多资源均衡. 目前, 这类问题一般采用权重法<sup>[3]</sup>将其转化为单项目单资源均衡问题进行处理. 但权重法存在一些不可克服的缺点: 首先, 决策者在设置各种资源的权重时, 往往带有一定的主观性和随机性; 其次, 权重法每次只能给出一种最优方案, 不可避免地可能遗漏更适合实际问题的其他方案.

3) 多项目单资源均衡. 目前, 处理这类问题的算法有启发式算法和遗传算法<sup>[4]</sup>. 但是, 启发式算法过

多地依赖于实际问题, 可移植性较差; 遗传算法涉及复杂的编码问题和遗传算子设计问题, 实现比较麻烦, 且存在过早收敛的问题.

在许多企业的实际工作中, 普遍存在着多种资源在多个项目中的优化利用问题, 与传统的资源均衡问题相比, 具有内部结构复杂、涉及因素众多、需要同时协调多种资源等特点. 本文在深入研究多项目多资源均衡问题的基础上, 为该问题建立了对应的多目标优化模型. 在算法设计上, 本文将 Pareto 方法嵌入向量评价微粒群算法 (VEPSO)<sup>[5]</sup>, 提出一种新的基于 Pareto 的向量评价微粒群算法 (VEPSO-BP). 在 VEPSO-BP 中, 各个子微粒群的全局最优位置直接来源于所有子微粒群所共同产生的 Pareto 非支配解, 所以该算法能有效避免 VEPSO 只能在单个子微粒群中选择全局最优位置的弊端, 加强了微粒飞

收稿日期: 2009-05-26; 修回日期: 2009-10-18.

作者简介: 郭研(1978—), 男, 浙江宁波人, 讲师, 博士生, 从事工程与项目管理的研究; 李南(1956—), 女, 重庆人, 教授, 博士生导师, 从事系统工程、技术与创新管理等研究.

行方向的引导性.

## 2 基于 Pareto 的向量评价微粒群算法

### 2.1 多目标优化问题的基本概念

多目标优化可描述为: 一个由满足一定约束条件的决策变量组成的向量, 使得一个由多个目标函数组成的向量函数最优化<sup>[6]</sup>, 其数学模型为

$$\begin{aligned} \min Y &= (f_1(U), f_2(U), \dots, f_p(U))^T, \\ U &= (u_1, u_2, \dots, u_N)^T; \\ \text{s.t. } h_j(U) &\leq 0, j = 1, 2, \dots, q. \end{aligned}$$

其中: 决策向量  $U \in R^N$ , 目标向量  $Y \in R^p$ ,  $f_i(U)$  ( $i = 1, 2, \dots, p$ ) 为目标函数;  $h_j(x) \leq 0$  为约束条件. 在大多数情况下, 多目标优化问题不存在唯一的全局最优解, 但可以存在这样的解: 对于一个或几个目标函数不可能进一步优化, 而对于其他目标函数不至于劣化, 这样的解称为 Pareto 非支配解 (Non-dominated solutions). 由所有 Pareto 非支配解组成的集合, 称为多目标优化问题的 Pareto 最优解集.

### 2.2 VEPSO-BP 算法

VEPSO-BP 与 VEPSO 一样, 都是基于联合进化 (Co-evolutionary) 技术的多目标微粒群算法. 其核心思想是将微粒群划分为若干个种群规模相等的子微粒群, 分别对多个目标函数进行优化, 并且各个子微粒群的适应值要受到其他子微粒群的影响.

VEPSO-BP 详细算法如下:

有  $p$  个种群规模都为  $M$  的子微粒群<sup>[1]</sup>Subswm, <sup>[2]</sup>Subswm,  $\dots$ , <sup>[p]</sup>Subswm, 同时对多目标优化问题的  $p$  个目标函数进行优化, 其中第  $k$  个子微粒群<sup>[k]</sup>Subswm 只负责优化第  $k$  个目标函数  $f_k(U)$ .

**注 1** 在本文中, 每个变量的左上标  $[k]$  表示该变量对应的是第  $k$  个子微粒群,  $k = 1, 2, \dots, p$ .

VEPSO-BP 局部最优位置的更新方式与 VEPSO 一致, 即第  $k$  个子微粒群的局部最优位置  $[k]p_{\text{best}}$  是按照第  $k$  个目标函数进行更新. 但是, 在全局最优位置的赋值和更新上, VEPSO-BP 与 VEPSO 有本质区别: VEPSO-BP 需要建立一个外部记忆体, 并将所有  $p$  个子微粒群中的 Pareto 非支配解都放入该记忆体中, 程序迭代时, 需按照一定的算法从外部记忆体中取  $p$  个 Pareto 非支配解 (可重复) 分别作为各个子微粒群的全局最优位置. 具体选取算法为: 对应外部记忆体中的每个非支配解, 将  $f_1(U)$  最小的非支配解的微粒位置作为第  $p$  个子微粒群的全局最优位置<sup>[p]</sup> $g_{\text{best}}$ , 将  $f_2(U)$  最小的非支配解的微粒位置作为第 1 个子微粒群的全局最优位置<sup>[1]</sup> $g_{\text{best}}$ , 将  $f_3(U)$  最小的非支配解的微粒位置作为第 2 个子微粒群的全

局最优位置<sup>[2]</sup> $g_{\text{best}}$ . 以此类推, 将  $f_p(U)$  最小的非支配解的微粒位置作为第  $p - 1$  个子微粒群的全局最优位置<sup>[p-1]</sup> $g_{\text{best}}$ . 通过这种交叉选取的方法, 使得各个子微粒群都能共享外部记忆体中所有非支配解的信息, 进而指导微粒尽快向 Pareto 前沿靠近.

此外, 适用于 VEPSO-BP 的进化方程为

$$\begin{aligned} [k]v_{ad}(\text{gen} + 1) &= \\ [k]\omega \times [k]v_{ad}(\text{gen}) &+ \\ [k]c_1 \times r_1 \times [k]p_{\text{best}_{ad}} - [k]x_{ad}(\text{gen}) &+ \\ [k]c_2 \times r_2 \times [k]g_{\text{best}_{ad}} - [k]x_{ad}(\text{gen}), & \quad (1a) \end{aligned}$$

$$[k]x_{ad}(\text{gen} + 1) = [k]x_{ad}(\text{gen}) + [k]v_{ad}(\text{gen} + 1). \quad (1b)$$

其中:  $[k]x_{ad}(\text{gen})$ ,  $[k]v_{ad}(\text{gen})$ ,  $[k]p_{\text{best}_{ad}}$ ,  $[k]g_{\text{best}_{ad}}$  分别为第  $\text{gen}$  时刻, 第  $k$  个子微粒群中的第  $a$  个微粒在第  $d$  维分量下的坐标、速度、局部最优位置和全局最优位置;  $[k]\omega$  为惯性权值;  $[k]c_1$  和  $[k]c_2$  为微粒的加速因子;  $r_1$  和  $r_2$  为两个在  $[0, 1]$  范围内变化的随机数.

## 3 多项目多资源均衡问题及其数学模型

### 3.1 问题描述

某企业有  $n$  个项目需要实施, 每个项目由若干个作业组成, 每个作业在施工时需要消耗  $p$  种资源. 对相关特征量作如下定义:  $\{\text{TA}(i)\}$  为第  $i$  个项目的作业集合;  $T_{ij}$  为第  $i$  个项目中的第  $j$  个作业,  $R_m(t)$  为第  $t$  个工作日  $n$  个项目在第  $m$  种资源上的消耗量,  $R_{mt}(T_{ij})$  表示第  $t$  个工作日作业  $T_{ij}$  在第  $m$  种资源上的消耗量,  $R_m(T_{ij})$  表示作业  $T_{ij}$  在单位时间里对第  $m$  种资源的消耗量, 其中  $m = 1, 2, \dots, p$ ,  $t = 1, 2, \dots, T$ ,  $T$  为整个工程的总工期;  $\text{ES}(T_{ij})$ ,  $\text{LS}(T_{ij})$ ,  $\text{AST}(T_{ij})$ ,  $\text{AFT}(T_{ij})$ ,  $D(T_{ij})$ ,  $S(T_{ij})$ ,  $\{\text{vag}(T_{ij})\}$  分别表示作业  $T_{ij}$  的最早开工时间、最迟开工时间、实际开工时间、实际完工时间、工期、松弛时间和紧前作业集.

解决多项目多资源均衡问题的基本思路是: 通过合理调整各个作业的实际开工时间  $\text{AST}(T_{ij})$ , 使总工期内各资源消耗的时间均衡度最大, 即资源方差 (RV) 最小.

对于整个工程的总工期  $T$ , 本文定义如下:

**定义 1** 整个工程的总工期等于最早开工项目的开始时间与最迟完工项目的结束时间之间的时间跨度, 即

$$T = \max\{\text{LS}(T_{ij}) + D(T_{ij})\} - \min\{\text{ES}(T_{ij})\}.$$

本文将最早开工项目的开始时间作为多项目工程工期的基点. 因此有  $\min\{\text{ES}(T_{ij})\} = 0$ , 总工期  $T = \max\{\text{LS}(T_{ij}) + D(T_{ij})\}$ .

此外, 在多项目多资源均衡问题中, 所有作业的

实际开工时间受到两类约束条件的限制:

**约束 1** 实际开工时间应该被限定在最早开工时间与最迟开工时间之间;

**约束 2** 实际开工时间受到其紧前作业的实际完工时间的限制.

### 3.2 数学模型

根据以上分析, 适用于多项目多资源均衡问题的数学模型如下:

$$\begin{cases} \text{目标函数} \\ \min RV_1 = \frac{1}{T} \sum_{t=1}^T (R_1(t) - \bar{R}_1)^2, \bar{R}_1 = \frac{1}{T} \sum_{t=1}^T R_1(t); \\ \min RV_2 = \frac{1}{T} \sum_{t=1}^T (R_2(t) - \bar{R}_2)^2, \bar{R}_2 = \frac{1}{T} \sum_{t=1}^T R_2(t); \\ \vdots \\ \min RV_p = \frac{1}{T} \sum_{t=1}^T (R_p(t) - \bar{R}_p)^2, \bar{R}_p = \frac{1}{T} \sum_{t=1}^T R_p(t). \end{cases} \quad (2)$$

约束条件

$$ES(T_{ij}) \leq AST(T_{ij}) \leq LS(T_{ij}); \quad (3a)$$

$$\max\{AFT(\text{vag}(T_{ij}))\} \leq AST(T_{ij}); \quad (3b)$$

$$R_m(t) = \sum_{i=1}^n \sum_{j \in \{TA(i)\}} R_{mt}(T_{ij}); \quad (3c)$$

$$\begin{cases} R_{mt}(T_{ij}) = \\ \begin{cases} R_m(T_{ij}), & AST(T_{ij}) < t \leq AFT(T_{ij}); \\ 0, & t \leq AST(T_{ij}) \text{ or } t > AFT(T_{ij}); \end{cases} \end{cases} \quad (3d)$$

$$S(T_{ij}) = LS(T_{ij}) - ES(T_{ij}). \quad (3e)$$

## 4 基于 VEPSO-BP 的多项目多资源均衡算法设计

### 4.1 编码设计

本文将非关键作业的实际开工时间作为编码对象. 对应 VEPSO-BP, 可以将目标问题的可行解空间假想为微粒的  $N$  维搜索空间,  $N$  代表该问题中所有项目非关键路线上的作业总数. 对于  $p$  种资源的资源均衡问题, 需要同时设置  $p$  个种群规模都为  $M$  的子微粒群.

微粒位置  $^{[k]}x_a = \{^{[k]}x_{a1}, ^{[k]}x_{a2}, \dots, ^{[k]}x_{ad}, \dots, ^{[k]}x_{aN}\}$  对应问题的一个可行解, 微粒位置的第  $d$  维分量  $^{[k]}x_{ad} (a = 1, 2, \dots, M; d = 1, 2, \dots, N)$  对应第  $d$  个非关键作业的实际开工时间.

### 4.2 算法流程

**Step 1** 随机产生  $p$  个种群规模都为  $M$  的子微粒群, 并初始化所有微粒的速度和位置.

初始化微粒速度: 为防止微粒速度  $^{[k]}v_{ad}$  过大,

可通过微粒最大速度  $v_{d\max}$  对微粒速度进行限制,  $^{[k]}v_{ad} \in [-v_{d\max}, v_{d\max}]$ , 微粒最大速度  $v_{d\max}$  与作业的松弛时间成正比, 即  $v_{d\max} = \beta S(T_{ij})$ ,  $0.1 \leq \beta \leq 1$ .

初始化微粒位置: 对于微粒位置  $^{[k]}x_{ad}$ , 可首先在  $[ES(T_{ij}), LS(T_{ij})]$  上随机产生, 然后检查是否满足约束条件 (3b). 如果不满足, 则在  $[\max\{AFT(\text{vag}(T_{ij})), ES(T_{ij})\}, LS(T_{ij})]$  上再随机产生一次.

最后根据式 (2) 计算每一微粒所对应的各项资源的资源方差.

**Step 2** 初始化外部记忆体. 首先将第 1 个子微粒群中第 1 个微粒放入外部记忆体中; 然后对随后随机产生的每个微粒与记忆体中的所有微粒进行比较.

比较规则如下:

**规则 1** 如果记忆体中的某个微粒的所有资源方差均大于此微粒, 则将记忆体中的那个微粒从记忆体中删去.

**规则 2** 如果记忆体中存在一个微粒, 其所有资源方差均小于该微粒, 则该微粒不添加到记忆体中; 否则, 将该微粒添加到记忆体中.

如此初始化结束后, 记忆体中保存的就是目前所有子微粒群所能得到的全体非支配解.

**Step 3** 初始化每个子微粒群的全局最优位置和每个微粒的局部最优位置.

**Step 4** 按进化方程更新所有微粒的位置和速度.

在资源均衡问题中, 作业的实际开工时间是一个整数量, 所以必须对式 (1a) 进行取整, 以保证微粒速度和微粒位置都为整数. 修改后的进化方程为

$$\begin{aligned} ^{[k]}v_{ad}(\text{gen} + 1) = \\ \text{int}(^{[k]}\omega \times ^{[k]}v_{ad}(\text{gen})) + \\ \text{int}(^{[k]}c_1 \times r_1 \times \{^{[k]}p_{\text{best}_{ad}} - ^{[k]}x_{ad}(\text{gen})\}) + \\ \text{int}(^{[k]}c_2 \times r_2 \times \{^{[k]}g_{\text{best}_{ad}} - ^{[k]}x_{ad}(\text{gen})\}), \end{aligned} \quad (4a)$$

$$\begin{aligned} ^{[k]}x_{ad}(\text{gen} + 1) = \\ ^{[k]}x_{ad}(\text{gen}) + ^{[k]}v_{ad}(\text{gen} + 1). \end{aligned} \quad (4b)$$

其中式 (4a) 中的  $\text{int}$  为取整函数.

最后通过 Step 1 中的动态监测方法, 判断微粒位置是否满足约束条件 (3a) 和约束条件 (3b). 如果不满足, 则应用进化方程来重新更新微粒的速度和位置.

**Step 5** 按 Step 2 中比较规则, 更新外部记忆体.

**Step 6** 更新每个微粒的局部最优位置和每个子微粒群的全局最优位置.

如果  $^{[k]}x_a$  在第  $k$  种资源上的资源方差小于局部最优位置  $^{[k]}p_{\text{best}_a}$  在第  $k$  种资源上的资源方差, 则将  $^{[k]}x_a$  作为局部最优位置  $^{[k]}p_{\text{best}_a}$ .

对应外部记忆体中的所有非支配解, 将  $RV_1$  最小的非支配解的微粒位置作为  $^{[p]}g_{best}$ , 将  $RV_2$  最小的非支配解的微粒位置作为  $^{[1]}g_{best}$ , 将  $RV_3$  最小的非支配解的微粒位置作为  $^{[2]}g_{best}$ . 以此类推, 将  $RV_p$  最小的非支配解的微粒位置作为  $^{[p-1]}g_{best}$ .

**Step 7** 重复 Step 4, 直至达到给定的最大迭代次数 Maxgen. 此时外部记忆体中所保存的数据就是算法所得到的 Pareto 最优解集.

### 5 算例分析

有两个工期不等的项目都需要使用人力资源  $R_1$  和资金  $R_2$ , 项目的初始网络分别如图 1 和图 2 所示. 经分析, 得到该算例的数学模型如下:

$$\begin{cases} \min RV_1 = \frac{1}{20} \sum_{t=1}^{20} (R_1(t) - \bar{R}_1)^2, \\ \min RV_2 = \frac{1}{20} \sum_{t=1}^{20} (R_2(t) - \bar{R}_2)^2; \\ \begin{cases} 0 \leq AST(T_{11}) \leq 11, 0 \leq AST(T_{12}) \leq 6, \\ AST(T_{12}) + 4 \leq AST(T_{13}) \leq 10, \\ 0 \leq AST(T_{14}) \leq 5, \\ 8 \leq AST(T_{15}) \leq 13, 2 \leq AST(T_{21}) \leq 17, \\ 5 \leq AST(T_{22}) \leq 11, 5 \leq AST(T_{23}) \leq 16. \end{cases} \end{cases}$$

从以上数学模型可知, 本算例共有 8 个非关键作业, 对应于 VEPSO-BP, 每个微粒代表一个作业安排方案, 相当于每个微粒是 8 维空间中的一个点, 即  $N = 8$ . 具体的编码方案如图 3 所示. 对应两种不同的资源  $R_1$  和  $R_2$ , 需要同时设置两个这样的子微粒群:  $^{[1]}Subswm$ ,  $^{[2]}Subswm$ . 其中  $^{[1]}Subswm$  用于优化  $R_1$ ,

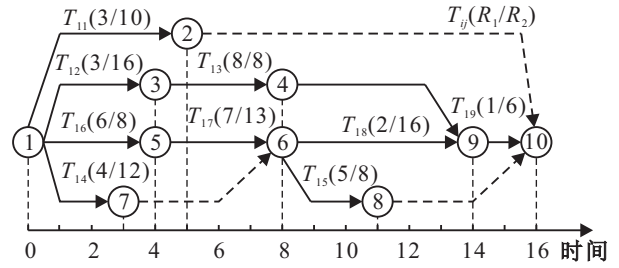


图 1 项目 1 的初始网络图

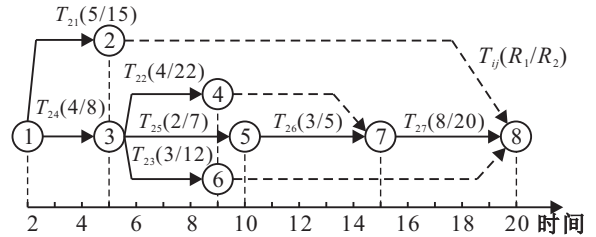


图 2 项目 2 的初始网络图

$X_{a_1}$	$X_{a_2}$	$X_{a_3}$	$X_{a_4}$	$X_{a_5}$	$X_{a_6}$	$X_{a_7}$	$X_{a_8}$
AST ( $T_{11}$ )	AST ( $T_{12}$ )	AST ( $T_{13}$ )	AST ( $T_{14}$ )	AST ( $T_{15}$ )	AST ( $T_{21}$ )	AST ( $T_{22}$ )	AST ( $T_{23}$ )

图 3 VEPSO-BP 编码方案

$^{[2]}Subswm$  用于优化  $R_2$ . 将外部记忆体中  $RV_2$  最小的非支配解的微粒位置作为第 1 个子微粒群的全局最优位置  $^{[1]}g_{best}$ , 将外部记忆体中  $RV_1$  最小的非支配解的微粒位置作为第 2 个子微粒群的全局最优位置  $^{[2]}g_{best}$ . 令每个子微粒群的种群规模  $M = 50$ , 设置最大迭代次数 Maxgen = 10000, 微粒最大速度  $v_{dmax} = 0.5S(T_{ij})$ .

表 1 由 VEPSO-BP 优化后得到的 Pareto 非支配解

序号	$RV_1$	$RV_2$	AST( $T_{11}$ )	AST( $T_{12}$ )	AST( $T_{13}$ )	AST( $T_{14}$ )	AST( $T_{15}$ )	AST( $T_{21}$ )	AST( $T_{22}$ )	AST( $T_{23}$ )
1	2.64	50.928	11	0	8	0	12	17	5	14
2	2.94	41.928	10	0	8	0	12	17	5	14
3	4.94	36.328	10	0	9	0	13	17	5	14
4	6.84	31.328	0	0	8	5	8	17	11	14
5	8.74	29.728	0	3	7	0	8	17	11	14
6	8.84	26.128	0	0	7	4	8	17	11	14
7	9.64	24.928	3	0	7	0	8	17	11	14
8	10.04	22.528	0	0	7	5	8	17	11	14

表 2 由 VEPSO 优化后得到的 Pareto 非支配解

序号	$RV_1$	$RV_2$	AST( $T_{11}$ )	AST( $T_{12}$ )	AST( $T_{13}$ )	AST( $T_{14}$ )	AST( $T_{15}$ )	AST( $T_{21}$ )	AST( $T_{22}$ )	AST( $T_{23}$ )
1	3.54	99.1275	11	3	8	0	12	17	6	14
2	4.64	94.3275	4	0	9	0	13	16	6	13
3	4.74	72.3275	5	0	8	0	13	17	11	14
4	4.84	68.4275	1	0	8	5	12	14	10	16
5	5.24	64.9275	10	0	8	3	12	17	6	15
6	5.64	63.5275	4	0	8	0	13	16	11	7
7	5.94	52.8275	11	0	10	0	8	14	5	16
8	6.04	48.1275	0	0	7	4	13	16	11	16
9	8.54	47.7275	0	0	6	4	9	17	11	13
10	10.24	28.2275	0	0	7	5	8	14	11	16

计算结果表明, VEPSO-BP 经过 2 148 次迭代计算后, 得到 8 个 Pareto 非支配解, 由 VEPSO-BP 优化后得到的非支配解如表 1 所示; VEPSO 经 7 168 次迭代计算后, 得到 10 个 Pareto 非支配解, 但都劣于 VEPSO-BP 所得到的非支配解, 具体数据如表 2 所示。

此外, 为了比较这两种算法的收敛速度和执行效率, 本文每隔 20 次运算便将两种算法得到的 Pareto 非支配解中  $R_1$  的最小资源方差  $RV_{1\min}$  和  $R_2$  的最小资源方差  $RV_{2\min}$  记录下来, 经过整理, 得到表 3。从表 3 可知, VEPSO-BP 在经过 660 次迭代运算后得到的优化效果已经优于 VEPSO 经过 7 180 次迭代运算后的优化效果。

表 3 收敛速度对比

算 法	迭代次数	$RV_{1\min}$	$RV_{2\min}$
VEPSO-BP	1	7.04	105.727 5
	20	7.04	56.827 5
	40	5.24	56.827 5
	60	4.94	56.827 5
	80	4.94	34.527 5
	140	4.14	34.527 5
	200	3.94	34.527 5
	320	3.94	28.227 5
	460	3.94	24.927 5
	660	3.24	24.927 5
	820	2.94	24.927 5
VEPSO	1560	2.94	22.527 5
	2160	2.64	22.527 5
	1	11.34	120.427 5
	20	6.94	88.427 5
	40	6.94	62.627 5
	60	5.64	62.627 5
	180	5.64	53.727 5
	1 100	5.64	47.627 5
	1 200	3.54	47.627 5
	1 740	3.54	36.427 5
7 180	3.54	28.227 5	

从该算例分析可知, VEPSO-BP 在处理多项多资源均衡问题时, 算法的执行效率是 VEPSO 的 10.88 倍, 且最终得到的 Pareto 最优解集中各项资源的最小资源方差比 VEPSO 降低了 25.4% 和 20.2%。

## 6 结 论

本文建立了适用于多项目多资源均衡问题的数学模型, 并提出了一种新的基于 Pareto 的向量评价微粒群算法 VEPSO-BP。最后通过算例的计算分析, 将 VEPSO-BP 得到的 Pareto 最优方案与 VEPSO 得到的方案进行比较分析。分析结果表明, 所提出的 VEPSO-BP 在优化多项目多资源均衡问题上具有一定的可行性。

## 参考文献(References)

- [1] Sou-Sen Leu, Chung-Huei Yang, Jiun-Ching Huang. Resource leveling in construction by genetic algorithm-based optimization and its decision support system application[J]. Automation in Construction, 2000, 10(1): 27-41.
- [2] 陈志勇, 杜志达, 周华. 基于微粒群算法的工程项目资源均衡优化[J]. 土工工程学报, 2007, 40(2): 93-96. (Chen Z Y, Du Z D, Zhou H. Research on the unlimited resource leveling optimization with PSO[J]. China Civil Engineering J, 2007, 40(2): 93-96.)
- [3] 王祖和, 亓霞. 多资源均衡的权重优选法[J]. 管理工程学报, 2002, 16(3): 91-93. (Wang Z H, Qi X. The weight optimal choice method of multi-resource leveling[J]. J of Industrial Engineering /Engineering Management, 2002, 16(3): 91-93.)
- [4] 郭研, 宁宣熙. 利用遗传算法求解多项目资源平衡问题[J]. 系统工程理论与实践, 2005, 25(10): 78-82. (Guo Y, Ning X X. Using genetic algorithms for multi-project resource balance[J]. Systems Engineering-Theory & Practice, 2005, 25(10): 78-82.)
- [5] Omkar S N, Dheevatsa Mudigere, Narayana Naik G, et al. Vector evaluated particle swarm optimization (VEPSO) for multi objective design optimization of composite structures[J]. Computers and Structures, 2008, 86(1): 1-14.
- [6] 蓝艇, 刘士荣, 顾幸生. 基于进化算法的多目标优化方法[J]. 控制与决策, 2006, 21(6): 601-605. (Lan T, Liu S R, Gu X S. Approaches of evolutionary multiobjective optimization[J]. Control and Decision, 2006, 21(6): 601-605.)