文章编号: 1001-0920(2013)01-0131-06

多模式多资源均衡及基于动态种群的多目标微粒群算法

郭 研[1,2], 李 南[1], 李兴森[2]

(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: 研究了多模式多资源均衡问题, 该问题需要动态选取每项任务的执行模式, 并综合考虑项目截止日期和资源限额等约束. 将种群竞争模型嵌入到基于Pareto的向量评价微粒群算法(VEPSO-BP)中, 提出了一种新的基于动态种群的多目标微粒群算法(MOPSO-DP). 通过实例测试了MOPSO-DP的性能, 并与VEPSO-BP进行了对比. 实验结果表明, MOPSO-DP能取得更为丰富且优化效果更好的Pareto非支配解.

关键词: 资源均衡；种群竞争模型；多目标优化；微粒群算法

中图分类号: F407.9 **文献标志码:** A

Multi-mode multiple resources leveling and multi-objective particle swarm optimization with dynamic population

GUO Yan[1,2], *LI Nan*[1], *LI Xing-sen*[2]

(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：guoyanbox@126.com)

Abstract: The multi-mode multiple resources leveling problem is studied, in which activity durations depend on committed resources, and the deadline and resource limit of the project are also taken into considerations. By applying population competition model into vector evaluated particle swarm optimization based on Pareto(VEPSO-BP), a novel multi-objective particle swarm optimization with dynamic population(MOPSO-DP) is utilized. Finally, the performance of MOPSO-DP is tested with a practical example which is compared with VEPSO-BP. Experiment results show that MOPSO-DP is able to obtain more superior Pareto non-dominate solutions than VEPSO-BP.

Key words: resource leveling；population competition model；multi-objective optimization；particle swarm optimization

0 引 言

传统的资源均衡问题研究如何合理调整项目任务的实际开工时间, 使资源的消耗量在整个工期内趋于均衡, 即资源方差最小. 在此类问题中, 项目的每一个任务所需的工期和资源是确定的, 即所有任务都只有一种执行模式. 目前用于单模式资源均衡问题的优化算法有遗传算法[1]和微粒群算法[2]等. 但在建筑工程、软件开发、飞机和轮船制造等单件或小批量生产方式的企业里, 根据多种不同的资源投入量和工期, 每项任务可能有多种不同的执行模式, 对此类项目进行资源均衡优化时, 不仅要调整各项任务的实际开工时间, 还要选取合适的执行模式, 这就是多模式多资源均衡问题. 在多模式多资源均衡问题中, 由于每项任务所需的工期和资源是不确定的, 会随着执行模式的不同而改变, 在求解此类问题时, 除了需要考虑传统资源均衡问题的约束条件外, 还需考虑项目截止日期和资源限额等约束, 增大了问题的复杂性和求解难度. 本质上, 多模式多资源均衡问题属于多模式资源受限项目调度问题[3](MRCPSP)的范畴, 是传统资源均衡问题的扩展, 比一般的资源均衡问题更接近于现实. 目前还较少见到解决多模式多资源均衡问题的研究文献.

本文首先为多模式多资源均衡问题建立了对应的多目标优化模型; 然后将种群竞争模型[4]嵌入到基于Pareto的向量评价微粒群算法(VEPSO-BP)[5]中, 提出了一种新的基于动态种群的多目标微粒群算法(MOPSO-DP). MOPSO-DP将文献[4]所提出的只适用于单目标优化的生态微粒群算法扩展到多目标优化领域. 此外, MOPSO-DP可根据种群竞争模型动态调整各子微粒群的规模, 并能较好地平衡全局搜索及

收稿日期: 2011-06-27；**修回日期:** 2011-11-25.

作者简介: 郭研(1978−), 男, 讲师, 博士生, 从事工程与项目管理的研究；李南(1956−), 女, 教授, 博士生导师, 从事管理科学与工程、项目管理等研究.

局部搜索能力.

1 多模式多资源均衡问题及其数学模型

1.1 问题描述

项目共含有n项任务, 每项任务需消耗r种资源, 其中可更新资源有d种, 不可更新资源有$r-d$种; 每一任务有m种执行模式, 每种执行模式$j(j=1,2,\cdots,m)$代表不同的需求(工期D, 可更新资源RR, 不可更新资源NR), 任务$T_i(i=1,2,\cdots,n)$的执行模式可表示为$(D_{ij},\mathrm{RR}_{ij1},\cdots,\mathrm{RR}_{ijd},\mathrm{NR}_{ij1},\cdots,\mathrm{NR}_{ijr-d})$. 多模式多资源均衡问题可描述为: 在满足项目截止日期和资源限额的前提下, 如何合理安排每项任务的执行模式和实际开工时间, 使项目工期内各资源消耗的时间均衡度最大, 即资源方差最小.

根据约束条件的不同, 可以把多模式多资源均衡问题分为4种类型: 1) 在确定的项目截止日期和资源限额约束下, 使资源方差最小; 2) 在确定的项目截止日期约束下, 使项目的资源方差和资源总量最小; 3) 在确定的资源限额约束下, 使项目的资源方差和项目工期最小; 4) 在项目截止日期和资源限额均不确定的条件下, 找到所有能使项目工期、资源总量、资源方差均趋于最小的最优解集.

问题4) 是求解所有可能的工期和资源组合下的最小资源方差, 故可以看作是前3类问题的综合, 本文即研究这类问题.

对于资源总量, 本文定义如下.

定义 1 对于可更新资源, 资源总量MRR为项目在该资源上的最大日需求量, 即

$$\mathrm{MRR}_l=\max\{\mathrm{RR}_l(t)|\ t\in(t=1,2,\cdots,\mathrm{PD})\}.$$

其中: $l(l=1,2,\cdots,d)$为可更新资源的序号, $\mathrm{RR}_l(t)$表示第t个工作日项目在可更新资源l上的需求量.

定义 2 对于不可更新资源, 资源总量PNR为项目在该资源上的需求总和, 即

$$\mathrm{PNR}_q=\sum_{i=1}^{n}\sum_{j=1}^{m}\mathrm{NR}_{ijq}X_{ij}.$$

其中: $q(q=1,2,\cdots,r-d)$为不可更新资源的序号; X_{ij}是取值为0或1的决策变量, $X_{ij}=1$表示将模式j作为任务T_i的执行模式, $X_{ij}=0$表示模式j未被选中.

1.2 数学模型

以下给出适用于多模式多资源均衡问题的数学模型.

目标函数为

$$\min\ \mathrm{PD}=\sum_{i\in\mathrm{CP}}\sum_{j=1}^{m}D_{ij}X_{ij},\tag{1}$$

$$\min\ \mathrm{MRR}_l=\max\{\mathrm{RR}_l(t)\},\tag{2}$$

$$\min\ \mathrm{PNR}_q=\sum_{i=1}^{n}\sum_{j=1}^{m}\mathrm{NR}_{ijq}X_{ij},\tag{3}$$

$$\min\ \mathrm{RRV}_l=\frac{1}{\mathrm{PD}}\sum_{t=1}^{\mathrm{PD}}(\mathrm{RR}_l(t)-\bar{\mathrm{RR}}_l)^2,\tag{4}$$

$$\min\ \mathrm{NRV}_q=\frac{1}{\mathrm{PD}}\sum_{t=1}^{\mathrm{PD}}(\mathrm{NR}_q(t)-\bar{\mathrm{NR}}_q)^2.\tag{5}$$

各目标函数的计算步骤如下:

Step 1: 根据决策变量X_{ij}计算任务的工期

$$D(T_i)=\sum_{j=1}^{m}D_{ij}X_{ij}.\tag{6}$$

Step 2: 根据任务工期, 应用关键路径法确定项目的关键路径CP, 并按式(1)计算项目工期PD.

Step 3: 根据任务的实际开工时间$\mathrm{ST}(T_i)$, 计算任务的实际完工时间$\mathrm{FT}(T_i)$, 有

$$\mathrm{FT}(T_i)=\mathrm{ST}(T_i)+D(T_i).\tag{7}$$

Step 4: 计算任务在可更新资源l和不可更新资源q上的日需求量$\mathrm{RR}_l(T_i)$和$\mathrm{NR}_q(T_i)$, 这里假设在单个任务的执行过程中, 不可更新资源的消耗是均匀的, 有

$$\mathrm{RR}_l(T_i)=\sum_{j=1}^{m}\mathrm{RR}_{ijl}X_{ij},\tag{8}$$

$$\mathrm{NR}_q(T_i)=\Big(\sum_{j=1}^{m}\mathrm{NR}_{ijq}X_{ij}\Big)\Big/D(T_i).\tag{9}$$

Step 5: 计算工作日t任务在可更新资源l和不可更新资源q上的资源需求量$\mathrm{RR}_{lt}(T_i)$和$\mathrm{NR}_{qt}(T_i)$, 有

$$\mathrm{RR}_{lt}(T_i)=\begin{cases}\mathrm{RR}_l(T_i),\ \mathrm{ST}(T_i)<t\leqslant\mathrm{FT}(T_i);\\ 0,\ t\leqslant\mathrm{ST}(T_i)\ \text{or}\ t>\mathrm{FT}(T_i).\end{cases}$$

$$\mathrm{NR}_{qt}(T_i)=\begin{cases}\mathrm{NR}_q(T_i),\ \mathrm{ST}(T_i)<t\leqslant\mathrm{FT}(T_i);\\ 0,\ t\leqslant\mathrm{ST}(T_i)\ \text{or}\ t>\mathrm{FT}(T_i).\end{cases}$$

Step 6: 计算工作日t项目在可更新资源l和不可更新资源q上的资源需求量$\mathrm{RR}_l(t)$和$\mathrm{NR}_q(t)$, 有

$$\mathrm{RR}_l(t)=\sum_{i=1}^{n}\mathrm{RR}_{lt}(T_i),\ \mathrm{NR}_q(t)=\sum_{i=1}^{n}\mathrm{NR}_{qt}(T_i).$$

Step 7: 按式(2)和(3)分别计算可更新资源l和不可更新资源q的资源总量MRR_l和PNR_q.

Step 8: 计算可更新资源l和不可更新资源q在项目工期内的日平均需求量$\bar{\mathrm{RR}}_l$和$\bar{\mathrm{NR}}_q$, 有

$$\bar{\mathrm{RR}}_l=\frac{1}{\mathrm{PD}}\sum_{t=1}^{\mathrm{PD}}\mathrm{RR}_l(t),\ \bar{\mathrm{NR}}_q=\frac{1}{\mathrm{PD}}\sum_{t=1}^{\mathrm{PD}}\mathrm{NR}_q(t).$$

Step 9: 根据式(4)和(5)计算可更新资源l和不可更新资源q的资源方差RRV_l和NRV_q.

约束条件为

$$\sum_{j=1}^{m} X_{ij} = 1, \tag{10}$$

$$\mathrm{ES}(T_i) \leqslant \mathrm{ST}(T_i) \leqslant \mathrm{LS}(T_i), \tag{11}$$

$$\max\{\mathrm{FT}(T_b)|T_b \in \mathrm{PA}(T_i)\} \leqslant \mathrm{ST}(T_i). \tag{12}$$

其中: 式(10)确保了每项任务仅有一种执行模式被选中; 式(11)表示任务的实际开工时间必须在其最早开工时间 $\mathrm{ES}(T_i)$ 和最晚开工时间 $\mathrm{LS}(T_i)$ 之间; 式(12)表示任务必须等待其所有紧前任务都完工后才能开工, $\mathrm{PA}(T_i)$ 表示任务的紧前任务集.

2　基于动态种群的多目标微粒群算法

2.1　种群竞争模型

在种群竞争模型中, 生态系统一般分为个体、群体和社团3个层次, 其中群体由具有相似生活形态的个体构成, 社团又由不同的群体构成. 为了自身的繁衍, 群体需要与处于同一社团中的不同群体竞争各种资源, 这种群体间的相互竞争会导致某些群体的个体数量减少, 而其他的群体个体数量增加. p 个群体的种群竞争模型(Latka-Volterra模型)[4]可描述为

$$\frac{M_i}{d(\mathrm{iter})} = r_i M_i\Big[\Big(K_i - M_i - \sum_{j=1,j\neq i}^{p} a_{ij}M_j\Big)\Big/K_i\Big]. \tag{13}$$

其中: $M_i(i = 1, 2, \cdots, p)$ 为第 i 个群体的规模(个体数量), iter为时间(在算法中即为迭代次数), r_i 为第 i 个群体的增长系数(群体出生率与死亡率之差), K_i 为第 i 个群体的容量(个体数量上限), a_{ij} 为竞争系数矩阵.

2.2　MOPSO-DP

现有的多目标微粒群算法通常采用多个子微粒群协同进化的方法, 但是在生态系统中, 群体间的相互竞争、优胜劣汰是整个系统得到不断进化的根本保证. 基于该思想, 提出了一种新的基于动态种群的多目标微粒群算法MOPSO-DP.

该算法的微粒群系统由 p 个子微粒群构成, 每个子群相当于该系统下的一个群体, 第 k 个子群 Subpop_k 只负责优化第 k 个目标函数 $f_k(U)$. 各个子群的规模将根据式(13)进行动态调整. 如果某个子群的增长系数较大, 则算法能随机产生1个或多个微粒加入该子群, 有利于提高该子群的多样性; 如果某个子群的增长系数较小, 则删除适应度最小的1个或多个微粒, 以满足优胜劣汰的自然定律. 另外, 通过竞争系数矩阵来实现子群内的自我抑制和子群间的协同竞争, 将整个系统的微粒总规模控制在一定限度内. 此外, 本文将MOPSO-DP中的每个子群划分为 h 个阶层, 每个阶层中的微粒具有不同的惯性因子 $\omega \in [\omega_{\min}, \omega_{\max}]$, 阶层 $s(s = 0, 1, \cdots, h-1)$ 中微粒的惯性因子为

$$\omega_s = \omega_{\min} + \frac{s}{h-1} \times (\omega_{\max} - \omega_{\min}). \tag{14}$$

子群中的微粒还可按一定的概率在不同的阶层间迁移, 这样便能较好地平衡MOPSO-DP算法的全局搜索和局部搜索能力.

MOPSO-DP在外部记忆体、局部最优位置和全局最优位置的初始化和更新方式上与VEPSO-BP一致, 适用于MOPSO-DP的进化方程为

$$\begin{aligned} &v_{kad}(\mathrm{iter}+1) = \\ &\omega_{ks} v_{kad}(\mathrm{iter}) + c_{k1} r_1[\mathrm{pbest}_{kad} - x_{kad}(\mathrm{iter})] + \\ &c_{k2} r_2[\mathrm{gbest}_{kd} - x_{kad}(\mathrm{iter})], \end{aligned} \tag{15}$$

$$x_{kad}(\mathrm{iter}+1) = x_{kad}(\mathrm{iter}) + v_{kad}(\mathrm{iter}+1). \tag{16}$$

其中: $x_{kad}(\mathrm{iter})$, $v_{kad}(\mathrm{iter})$, pbest_{kad}, gbest_{kd} 分别是第iter时刻第 k 个子群中第 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; ω_{ks} 是第 k 个子群第 s 个阶层中微粒的惯性因子; c_{k1}, c_{k2} 是微粒的加速因子; r_1, r_2 是在[0,1]内变化的随机数.

3　基于MOPSO-DP的多模式多资源均衡优化算法设计

3.1　编码设计

本文将任务的决策变量 X_{ij} 和实际开工时间 $\mathrm{ST}(T_i)$ 作为编码对象. 对应MOPSO-DP, 可以将多模式多资源均衡问题的可行解空间假想为微粒的 n 维搜索空间, n 代表项目的任务总数. 微粒位置 $x_{ka} = (x_{ka1}, x_{ka2}, \cdots, x_{kai}, \cdots, x_{kan})$ 对应问题的一个可行解, 其第 i 维分量 $x_{kai}(a = 1, 2, \cdots, M,\ i = 1, 2, \cdots, n)$ 对应第 i 个任务的决策变量和实际开工时间. 微粒速度的编码方式和微粒位置类似, 也由两部分组成: 控制决策变量的速度分量 vX_{ij} 和控制实际开工时间的速度分量 $v\mathrm{ST}(T_i)$.

3.2　算法流程

Step 1　随机产生 $p = 1 + 2r$ 个初始规模都为 M_0 的子微粒群 $\mathrm{Subpop}_1, \mathrm{Subpop}_2, \cdots, \mathrm{Subpop}_p$, 并设置每个子群的容量 K_i 和各子群间的竞争系数矩阵 a_{ij}; 定义各子群的阶层个数 h 和每个阶层微粒个数占子群微粒总数的百分比; 最后初始化所有微粒的位置和速度.

1) 初始化微粒位置中的决策变量, 以确定所有任务的执行模式. 决策变量 X_{ij} 可在0或1中随机产生, 但必须满足式(10)的约束; 根据式(6), (8)和(9)分别计算任务的工期和各种资源的日需求量, 并确定项目的关键路径; 最后计算所有任务的最早开工时间和最晚开工时间.

2) 初始化微粒位置中的实际开工时间. $\mathrm{ST}(T_i)$

首先可在 $[\mathrm{ES}(T_i),\mathrm{LS}(T_i)]$ 上随机产生, 然后检查是否满足式 (12) 的约束, 如果不满足则在 $[\max\{\max\{\mathrm{FT}(T_b)|T_b\in\mathrm{PA}(T_i)\},\mathrm{ES}(T_i)\},\mathrm{LS}(T_i)]$ 上再随机产生一次.

3) 初始化微粒速度. 为了防止微粒速度 v_{kad} 过大, 可通过微粒最大速度 $v_{i\max}$ 对微粒速度进行限制, 有

$$v_{i\max}=$$
$$(vX_{i1\max},vX_{i2\max},\cdots,vX_{im\max},v\mathrm{ST}(T_i)_{\max}),$$

其中控制决策变量的微粒最大速度分量 $vX_{i1\max}=vX_{i2\max}=\cdots=vX_{im\max}=1$. 控制实际开工时间的微粒最大速度分量 $v\mathrm{ST}(T_i)_{\max}$ 与该任务的松弛时间 $\mathrm{TF}(T_i)$ 成正比, 即

$$v\mathrm{ST}(T_i)_{\max}=\beta\mathrm{TF}(T_i),\mathrm{TF}(T_i)=\mathrm{LS}(T_i)-\mathrm{ES}(T_i),$$
$$\beta\in[0,1].$$

4) 根据式 (1)~(5) 分别计算各微粒所对应的项目工期、各资源的资源总量和资源方差.

Step 2 初始化外部记忆体、各子群的全局最优位置和各微粒的局部最优位置.

Step 3 根据种群竞争模型调整所有子群的种群规模, 并更新所有微粒的位置和速度.

1) 根据式 (13) 确定各子群增加或删除的微粒个数, 并按下式来更新各子群的规模:

$$M_i(\mathrm{iter}+1)=M_i(\mathrm{iter})+\frac{dM_i}{d(\mathrm{iter})}. \tag{17}$$

2) 按进化方程更新所有微粒的位置和速度. 由于在多模式多资源均衡问题中, 任务的决策变量和实际开工时间都是整数量, 必须对式 (15) 进行取整, 修改后的进化方程为

$$v_{kad}(\mathrm{iter}+1)=$$
$$\mathrm{int}(\omega_{ks}v_{kad}(\mathrm{iter}))+\mathrm{int}(c_{k1}r_1[\mathrm{pbest}_{kad}-x_{kad}(\mathrm{iter})])+$$
$$\mathrm{int}(c_{k2}r_2[\mathrm{gbest}_{kd}-x_{kad}(\mathrm{iter})]), \tag{18}$$
$$x_{kad}(\mathrm{iter}+1)=x_{kad}(\mathrm{iter})+v_{kad}(\mathrm{iter}+1), \tag{19}$$

其中 int( ) 为取整函数. 最后通过 Step 1 中的监测方法判断微粒位置是否满足式 (10)~(12) 的约束. 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 4 更新外部记忆体、各微粒的局部最优位置和各子群的全局最优位置. 如果 x_{ka} 的第 k 个目标函数值小于局部最优位置 pbest_{ka} 的第 k 个目标函数值, 则将 x_{ka} 作为局部最优位置 pbest_{ka}.

对应外部记忆体中的所有非支配解, 将 PD 最小的非支配解的微粒位置作为 gbest_p, 将 MRR_1 最小的非支配解的微粒位置作为 gbest_1, 以此类推, 将第 p 个目标函数 NRV_{r-d} 最小的非支配解的微粒位置作为 gbest_{p-1}.

Step 5 重复 Step 3 和 Step 4 直至达到给定的最大迭代次数 $\mathrm{iter}_{\max}$, 此时, 外部记忆体中所保存的数据便是算法所得到的 Pareto 最优解集.

4 实例研究

4.1 应用实例

本实例选自现实中的一个软件研发项目, 共包含 8 项研发任务, 每项任务需要人力资源和资金 2 种资源, 每项任务均有 2 种执行模式, 项目网络图如图 1 所示. 对应总工期、人力资源总量、总资金、人力资源方差和资金资源方差 5 个目标, 需设置 5 个子微粒群. 因为 $m=2, X_{i1}+X_{i2}=1$, 为了缩短编码长度, 同时消除式 (10) 的约束, 只将 X_{i1} 放入编码. 若 $X_{i1}=0$, 则表示任务 T_i 的执行模式为 $(D_{i2},\mathrm{HR}_{i2},F_{i2})$; 若 $X_{i1}=1$, 则表示任务 T_i 的执行模式为 $(D_{i1},\mathrm{HR}_{i1},F_{i1})$.

对应 VEPSO-BP, 各子群的种群规模 $M=10$, 惯性因子 $\omega=1$, 最大次数 $\mathrm{iter}_{\max}=1\,000$; 对应 MOPSO-DP, 令各子群容量 $K=25$, 子群的初始规模 $M_0=6$, 由于各目标的优化难度不同, 需设置不同的子群增长系数 $r_1=0.2, r_2=0.2, r_3=0.2, r_4=0.3, r_5=0.3$, 竞争系数矩阵如表 1 所示.

基于所设置的子群初始规模和容量, 将各子群分成 3 个阶层: 位于 H_0 阶层的微粒占微粒总数的 25%, 对应的惯性因子 $\omega_0=0.6$; 位于 H_1 阶层的微粒占微粒总数的 50 %, 对应的 $\omega_1=0.8$; 位于 H_2 阶层的微粒占微粒总数的 25%, 对应的 $\omega_2=1$; 其他的参数设置同 VEPSO-BP.

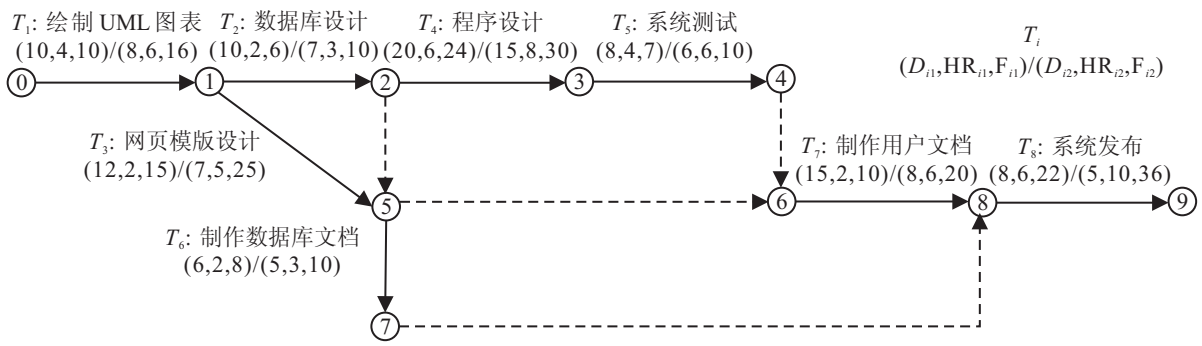

图 1 软件项目网络图

表 1　竞争系数矩阵

a_{ij}	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=1$	0	0.06	0.06	0.15	0.15
$i=2$	0.05	0	0.07	0.1	0.15
$i=3$	0.06	0.06	0	0.15	0.15
$i=4$	0.05	0.05	0.08	0	0.2
$i=5$	0.05	0.05	0.07	0.15	0

4.2　计算结果讨论

4.2.1　种群总规模

在 VEPSO-BP 中, 各子群的种群规模是固定的, 总规模为 50; MOPSO-DP 的各子群的种群规模是可变的. 根据第 4.1 节设置的参数, MOPSO-DP 在初次迭代时种群总规模为 32, 随着迭代次数的增加而增加, 当算法迭代至 12 次时达到最大值 51, 随后 MOPSO-DP 各子群的规模达到平衡, 一直保持在 51. 所以, 虽然理论上 MOPSO-DP 的种群总规模最大可达 125, 但由于各子群之间存在相互制约和竞争, 种群总规模与 VEPSO-BP 相差不大.

4.2.2　非支配解的个数

在种群规模相当的前提下, 非支配解的个数是衡量多目标优化算法性能的重要指标. 两种算法每次迭代所获得的非支配解个数的变动曲线如图 2 所示.

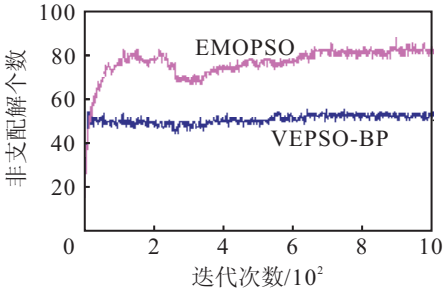

图 2　非支配解个数变动曲线

由图 2 可知, 在初次迭代, 由于 MOPSO-DP 的种群规模小于 VEPSO-BP, MOPSO-DP 只得到 14 个非支配解, 少于 VEPSO-BP 得到的 25 个, 但随着迭代次数的增加, VEPSO-BP 每次迭代获得的非支配解个数一直维持在 50 左右, 而 MOPSO-DP 在经过 600 次迭代后, 每次运算获得的非支配解个数保持在 80 上下. 所以, 通过相同数量的微粒优化, MOPSO-DP 获得的非支配解要明显多于 VEPSO-BP.

4.2.3　算法收敛速度和非支配解的质量

为了比较这两种算法的收敛速度, 记录每次迭代后得到的非支配解中的最小总工期 $\mathrm{PD}_{\min}$, 最小人力资源总量 $\mathrm{MH}_{\min}$, 最小总资金 $\mathrm{PF}_{\min}$, 最小人力资源方差 $\mathrm{HV}_{\min}$ 和最小资金资源方差 $\mathrm{FV}_{\min}$, 经过整理后的数据如表 2 所示. 其中: I 为 MOPSO-DP 算法, II 为 VEPSO-BP 算法.

表 2　算法收敛速度对比分析

算法	次数	$\mathrm{PD}_{\min}$	$\mathrm{MH}_{\min}$	$\mathrm{PF}_{\min}$	$\mathrm{HV}_{\min}$	$\mathrm{FV}_{\min}$
I	1	49	8	108	1.559	0.382
	2	49	8	102	1.526	0.264
	3	49	8	102	0.895	0.264
	5	49	8	102	0.895	0.226
	8	49	8	102	0.895	0.183
	21	49	7	102	0.895	0.183
	573	49	7	102	0.881	0.183
	581	49	7	102	0.746	0.183
	590	49	7	102	0.610	0.183
	1 000	49	7	102	0.610	0.183
II	1	49	8	102	1.667	0.402
	3	49	8	102	1.667	0.311
	13	49	8	102	0.949	0.307
	21	49	8	102	0.763	0.254
	27	49	8	102	0.763	0.244
	34	49	8	102	0.763	0.224
	221	49	8	102	0.746	0.224
	1 000	49	8	102	0.746	0.224

由表 2 可知, 在收敛速度上, MOPSO-DP 要慢于 VEPSO-BP, 但最小资源方差分别比 VEPSO-BP 降低了 18.2% 和 18.3%.

为了评价 MOPSO-DP 和 VEPSO-BP 所得到的非支配解的质量, 采用以下评价指标.

1) 非支配解优劣指标 C.

C 指标由 Zitzler[6]提出, 其定义为

$$C(A,B)=\frac{|\{b\in B|\ \exists a\in A: a\succ b\}|}{|B|}.$$

其中: A 和 B 是两种算法所得到的非支配解集; C 是一种值域定义在 [0,1] 上, 用于刻画 (A,B) 之间偏序性能的指标, $C(A,B)=1$ 表示 B 中所有解都被 A 中的解支配, $C(A,B)=0$ 表示 B 中没有解被 A 中的解支配. 记 DP 和 BP 分别是 MOPSO-DP 和 VEPSO-BP 所得到的非支配解集, 经计算可得

$$C(\mathrm{DP},\mathrm{BP})=0.489,\ C(\mathrm{BP},\mathrm{DP})=0.013.$$

2) 均匀性指标 SP.

本文采用基于海明 (Hamming) 距离的均匀性指标 SP[7],其定义为

$$\mathrm{SP}=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2},\ \bar{d}=\frac{1}{n}\sum_{i=1}^{n}d_i,$$

$$d_i=\min\left\{\sum_{k=1}^{p}|f_k^i(x)-f_k^j(x)|\right\},$$

$$j=1,2,\cdots,n,\ j\neq i.$$

其中: n 为算法所得非支配解的个数, p 为目标函数的个数, d_i 为第 i 个非支配解对应目标向量与其最靠近的目标向量之间的距离. SP 值越小, 非支配解前沿分布越均匀, 经计算可得

$$\mathrm{SP}_{\mathrm{MOPSO-}DP}=2.16,\ \mathrm{SP}_{\mathrm{VEPSO-BP}}=2.581.$$

表 3　MOPSO-DP 优化后得到的 Pareto 非支配解 (部分)

No	PD	MH	PF	HV	FV	决策变量 X_{i1}								实际开工时间 ST(T_i)							
						X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
1	49	10	145	3.061	2.228	0	0	1	0	0	1	0	0	0	8	8	15	30	30	36	44
2	52	10	131	2.904	0.183	0	0	1	0	0	1	0	1	0	8	8	15	30	30	36	44
3	59	8	132	0.610	1.386	0	0	0	1	1	1	0	1	0	8	8	15	35	37	43	51
4	64	7	122	1.703	1.100	1	1	0	1	1	1	0	1	0	10	10	20	40	40	48	56
5	71	8	102	2.423	0.426	1	1	1	1	1	1	1	1	0	10	10	20	40	56	48	63

由计算得到的指标 C 和 SP 可知, MOPSO-DP 得到的非支配解要优于 VEPSO-BP, 且分布也较VEPSO-BP 均匀. 表 3 给出了 MOPSO-DP 优化后最小目标函数值所在的那部分 Pareto 非支配解.

5 结　论

现有文献对资源均衡问题的研究主要集中在可更新资源上, 且一般只考虑资源方差最小这一目标. 但是, 在目前市场资金普遍紧缺、能源危机迫近的大背景下, 如何在满足项目期限和资源限额的条件下, 有效地利用各种不可更新资源和可更新资源就显得尤为重要. 本文研究了多模式多资源均衡问题, 综合考虑了项目工期、资源限额和资源方差等多个目标, 并提出了一种新的基于动态种群的多目标微粒群算法. 该算法除了以协同进化算法为基础, 还进一步考虑了不同子群间的相互竞争和子群体内个体与个体间的相互协调, 大大拓展了传统微粒群算法的生态基础和系统行为. 实例研究部分的计算表明, MOPSO-BP 更易趋向于全局收敛.

参考文献(References)

[1] 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 and Practice, 2005, 25(10): 78-82.)

[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] Jarboui B, Damak N, Siarry P. A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems[J]. Applied Mathematics and Computation, 2008, 195(1): 299-308.

[4] Kang Q, Wang L, Wu Q D. A novel ecological particle swarm optimization algorithm and its population dynamics analysis[J]. Applied Mathematics and Computation, 2008, 205(1): 61-72.

[5] 郭研, 李南, 李兴森. 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[J]. 控制与决策, 2010, 25(5): 789-793.
(Guo Y, Li N, Li X S. Multiple resource leveling in multiple projects and vector evaluated particle swarm optimization based on Pareto[J]. Control and Decision, 2010, 25(5): 789-793.)

[6] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results[J]. IEEE Trans on Evolutionary Computation, 2000, 8(2): 173-195.

[7] 胡广浩, 毛志忠, 何大阔. 基于两阶段领导的多目标粒子群优化算法[J]. 控制与决策, 2010, 25(3): 404-410.
(Hu G H, Mao Z Z, He D K. Multi-objective PSO optimization algorithm based on two stages guided[J]. Control and Decision, 2010, 25(3): 404-410.)

下　期　要　目