第39卷第8期
2011年8月

同济大学学报(自然科学版)
JOURNAL OF TONGJI UNIVERSITY(NATURAL SCIENCE)

Vol.39 No.8
Aug. 2011

文章编号：0253-374X(2011)08-1222-05

DOI:10.3969/j.issn.0253-374x.2011.08.023

动态拓扑结构的多目标粒子群优化算法

任子晖，王　坚

（同济大学 CIMS 中心，上海 201804）

摘要：介绍了一种动态拓扑结构的多目标粒子群优化算法（dynamical topology multiple-objective particle swarm optimization，DMPSO）.给出了一种新的储备集更新策略，定义了支配度和邻域拥挤度及粒子差异度的概念，根据支配度及邻域拥挤度的大小来决定储备集的更新，增强了解的多样性和均匀性.为了防止早熟收敛，结合邻域拥挤度和粒子差异度，给出了一种拟小世界动态拓扑邻域结构来平衡粒子的全局搜索能力和局部搜索能力.最后通过对几个例子的数值实验说明算法的可行性，并通过成功地应用在实际工程问题上说明方法的有效性.

关键词：多目标粒子群优化；拟小世界动态拓扑；支配度；拥挤度

中图分类号：TP 18　　**文献标识码**：A

Dynamical Topology Multi-objective Particle Swarm Optimization Algorithm

REN Zihui, *WANG Jian*

(CIMS Research Center, Tongji University, Shanghai 201804, China)

Abstract: The paper presents a dynamical topology multi-objective particle swarm optimization (DMPSO) algorithm and a definition of the degree of domain and congestion and discrepancy of particles as well as a strategy of renewing archives which depends on the degree of domain and congestion around neighborhood. So the diversity and uniformity of solution are enhanced. In order to overcome the premature convergence, a new imitating small world dynamical topology strategy based on congestion degree and discrepancy degree is applied to balancing the ability of global searching and local searching. In the end, the application successfully to engineering shows that the DMPSO is feasible and effective.

Key words: multi-objective particle swarm optimization (MPSO); imitating small world dynamical topology; domain degree; congestion degree

多目标优化问题（multi-objective optimization problem，MOP）大量地存在于科学实践、工程系统设计、社会生产活动及投资组合等问题中.近年来，越来越多的学者采用进化算法来求解多目标的优化问题，提出了各种各样的多目标的粒子群优化算法（multi-objective particle swarm optimization，MPSO）.如 Wen－Fung Leong 等[1]提出了一种基于动态群体规模和自适应局部存档的 MPSO 算法，通过增加及减少粒子的规模来达到加快收敛的目的.R.Brits 等[2]提出了一种小生境的 MPSO 算法，在原始种群的基础上增加一些独立优化的子群体.Praveen Kumar Tripathi 等[3]提出了一种基于时间变化权重和加速系数的 MPSO 算法.含区间参数多目标系统的微粒群优化算法[4]通过概率支配关系来确定解的优劣.张利彪等[5]提出一种多子群进化算法，并在群体中实现信息交换来保证群体分布的均匀性.曲红等[6]提出了一种动态的 MPSO 算法，对粒子位置和速度的更新公式依据杂交概率进行了动态的改进.丛琳等[7]根据多目标的特点，提出了正交免疫克隆粒子群算法.上述的 MPSO 算法均缺少对其拓扑结构的分析，对于互不支配的 2 个解是否进入档案库（储备集）也没有给出有效的解决办法.此外，文献[8－12]也分别对粒子群的拓扑结构进行了改进，但是均没有应用在多目标的粒子群优化上且在改变边的连接结构时针对性也不是很强.本文针对这些缺点给出了一种动态拓扑结构的 MPSO 算法.在对粒子拓扑结构分析的基础上，给出解的支配度和拥挤度及粒子差异性的概念，同时对粒子的拓扑

收稿日期：2010－04－28

基金项目：国家自然科学基金重大研究计划集成项目(91024131)；“十一五”国家科技支撑计划(2006BAG01A02)；上海市科技发展基金(08201201905，08DZ1120802)

第一作者：任子晖(1983—)，女，博士生，主要研究方向为计算智能及其应用.E-mail：renzihui2006@126.com

通讯作者：王　坚(1961—)，男，教授，博士生导师，工学博士，主要研究方向为智能计算、仿真技术研究等.E-mail：jwang@tongji.edu.cn

结构进行调整，根据决策容忍系数的大小更新储备集，由此提出了一种动态拓扑结构的 MPSO 算法(dynamical topology MPSO，DMPSO).

1　一些基本概念

通常多目标优化问题的描述如下：

$$\begin{aligned}&\min\ \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))\\ &\text{s.t.}\ \ g_j(\boldsymbol{x}) \leqslant 0, j = 1,2,\cdots k\end{aligned} \tag{1}$$

式中：$\boldsymbol{f}(\boldsymbol{x})$为目标向量，$\boldsymbol{f}(\boldsymbol{x}) \in \mathbf{R}^m$，$\boldsymbol{x} \in \mathbf{R}^D$ 为决策变量，$f_i(\boldsymbol{x})$是第 i 个目标函数；$g_j(\boldsymbol{x})$为第 j 个约束条件. 当$\forall$ i$\in\{1,2,\cdots,m\}$，有 $f_i(\boldsymbol{x}^*) \leqslant f_i(\boldsymbol{x})$且$\exists\, i_0 \in \{1,2,\cdots,m\}$，满足 $f_{i0}(\boldsymbol{x}^*) < f_{i0}(\boldsymbol{x})$时，称向量 $\boldsymbol{x}^*$ 支配向量 $\boldsymbol{x}$. $\boldsymbol{x}^* = (x_1^*, x_2^*, \cdots, x_D^*)$称为 Pareto 解当且仅当不存在 $i_0 \in \{1,2,\cdots,m\}$使得 $f_{i0}(\boldsymbol{x}) < f_{i0}(\boldsymbol{x}^*)$成立. 所有 Pareto 解的集合构成 Pareto 最优集. 所有 Pareto 最优解对应的目标函数值形成的区域 P_{F} 称为 Pareto 最优前端或均衡面. 具体参见文献[5－7].

2　动态拓扑的多目标粒子群优化算法

PSO(particle swarm optimization)以模拟鸟的群体智能为特征，以求解连续变量优化问题为背景. 在 PSO 中，每只鸟被称为一个粒子，每个粒子用其几何位置和速度向量表示，每个粒子参考自己的既定方向及所经历的最优方向和整个鸟群所公共认识到的最优方向来确定自己的飞行.

PSO 是 Kennedy 和 Eberhert[13] 于 1995 年首先提出，为便于对粒子的拓扑结构进行分析研究，在文献[14]中采用式(2)对粒子群的位置 $\boldsymbol{x}_i$ 和速度 $\boldsymbol{v}_i$ 进行更新：

$$\begin{aligned}&\boldsymbol{v}_i = \chi\left(\boldsymbol{v}_i + \sum_{i=1}^{N_i} u(0,\varphi)(p_{\mathrm{nbr}(n)} - \boldsymbol{x}_i)/N_i\right)\\ &\boldsymbol{x}_i = \boldsymbol{x}_i + \boldsymbol{x}_i\end{aligned} \tag{2}$$

式中：$v_i \in [-v_{\max}, v_{\max}]$，$v_{\max}$是常数，$i = 1,2,\cdots, m$；$\chi$ 为影响粒子轨迹收敛的一个约束因子，通常取 0.729[15]；N_i为 i 的邻域集合；$u(0,\varphi)$是一个$(0,\varphi)$间的均匀分布的随机数，φ 是加速因子，φ 控制粒子的收敛速度，通常取 4.1[15]；下标nbr(n)为 i 的第 n 个邻域；第 i 个粒子的位置用一个 D 维的向量 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$表示，它在空间的飞行速度其用 $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$表示. 迭代终止条件根据具体问题一般选为最大迭代次数或粒子群迄今为止搜索到的最优位置满足预定最小适应阀值.

2.1　储备集更新策略

在多目标的求解过程中，最关键的问题就是如何确定 Pareto 最优前端，而在最优前端确定的过程中重要的步骤就是储备集(外部档案)的更新策略. 在叙述该文储备集更新策略之前先给出几个定义.

假设 a，b 都是非劣解，那么 a，b 是互不支配的，因为 a 在f_2方向支配 b，b 在f_1 的方向支配 a. 下面给出支配度的概念.

定义 1　假设 x_i，x_j为式(1)的非劣解，若 x_i在 $f_{p_1}, f_{p_2}, \cdots, f_{p_k}$方向支配$x_j$，$x_j$在$f_{q_1}, f_{q_2}, \cdots, f_{q_k}$方向支配$x_i$，其中 $p_1, p_2, \cdots, p_k, q_1, q_2, \cdots, q_k \in \{1, 2, \cdots, m\}$，那么称 $\rho(x_i, x_j)$为 x_i对 x_j的支配度.

$$\begin{aligned}\rho(x_i, x_j) &= \rho_{f_1}(x_i, x_j) + \rho_{f_2}(x_i, x_j) + \cdots\\ &+ \rho_{f_m}(x_i, x_j) = \sum_{k=f_1}^{f_m} \rho_k(x_i, x_j)\end{aligned} \tag{3}$$

其中，$\rho_{f_k}(x_i, x_j)$，$k \in \{1,2,\cdots,m\}$称为 x_i在f_k方向上对x_j的支配度，其计算公式为

$$\rho_{f_k}(x_i, x_j) = \frac{f_k(x_j) - f_k(x_i)}{\sum_{p=1}^{m} |f_p(x_i) - f_p(x_j)|} \tag{4}$$

定义 2　将目标空间按照文献[1]的方法进行划分，每次进化后计算每一方格 k 内进入储备集的粒子个数N_{k_i}与整个方格内粒子数N_k的比值$\tau_k(t)$称为第 t 代时第k 个方格内粒子的拥挤度.

$$\tau_k(t) = N_{k_i}(t)/\, N_k(t) \tag{5}$$

在更新储备集时，若储备集的大小未超过规定的定值，非劣解将直接进入储备集. 若储备集的大小超过规定的定值，且新解 x_i支配储备集中的解x_j，则新解 x_i将进入储备集，解 x_j将从储备集中删除；否则根据新解 x_i对储备集中解支配度的大小及新解所在方格内拥挤度的大小判断被替代的解，这里取支配度大于给定的值且拥挤度小于给定值的解来替代储备集中的解. 因为当新解 x_i所在方格内的拥挤度很小时，说明此方格内进入储备集的粒子数很少，为了保证解的多样性和均匀性，故在新解对储备集中部分解中的支配度适中情况下，选取方格内(邻域内)拥挤度最小的解替代储备集中的部分解.

设 $A(t)$为第 t 代储备集，$N_{A(t)}$表示第 t 代储备集中元素的个数，N_0 是给出的储备集中元素个数的规定值，α，β 分别为支配度和拥挤度的决策容忍系数，α，β 常取为初始的平均支配度和平均拥挤度的值，本文取值为 1/3. 下面简单给出储备集更新的步骤.

若 $N_{A(t)} < N_0$，$x_i(t+1)$ 进入储备集，$A(t+1) = A(t) \bigcup \{x_i(t+1)\}$，$N_{A(t+1)} = N_{A(t)} + 1$；否则判断 $x_i(t+1)$ 是否支配 x_k，$x_k \in A(t)$，若是，$A(t+1) = A(t) \bigcup \{x_i(t+1)\} \setminus \{x_k\}$，$N_{A(t+1)} = N_{A(t)} + 1$；否则计算 $\rho_{f_k}(x_{i+1}(t), x_j)$，计算 $x_i(t+1)$ 所在方格 k 内的拥挤度因子 $\tau_k(t+1)$，若 $\rho_{f_k}(x_{i+1}(t), x_j) \geqslant \alpha$ ，且 $\tau_k(t+1) \leqslant \beta$，则 $A(t+1) = A(t) \bigcup \{x_i(t+1)\} \setminus \{x_j\}$，$N_{A(t+1)} = N_{A(t)} + 1$，其中 $j \in \{1,2,\cdots,N_{A(t)}\}$；否则 $A(t+1) = A(t)$，$N_{A(t+1)} = N_{A(t)}$.

2.2 动态拓扑结构的描述

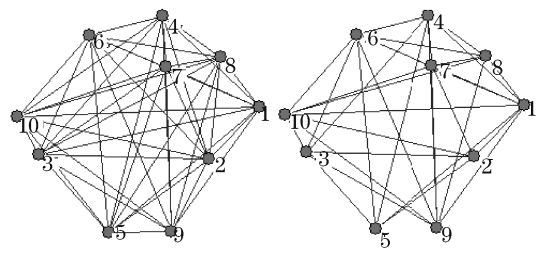
粒子群的拓扑结构是指整个群体中所有粒子之间连接的方式，而粒子群的邻域结构是单个粒子如何与其他粒子相连的方式. 拓扑结构是整体性质，邻域结构是局部性质，邻域结构决定了拓扑结构. 本文的思想就是通过粒子邻域结构变化来达到改变粒子拓扑结构变化的目的.

根据 2.1 节计算出储备集中每个元素所在方格的拥挤度，当拥挤度的最大值和最小值相差很大时，且以后多次迭代得到的解都分布在某一固定方格的周围，这表明拥挤度的差距还将继续增大，说明粒子的多样性不是很好，全局的搜索能力不强，需要改变粒子的邻域结构来增强全局搜索能力；反之需要改变粒子的邻域结构来增强局部搜索能力. 基于这个思想，定义了粒子邻域结构数量控制的自适应函数 $\bar{N}_i$.

$$\bar{N}_i = N_i e^{-1/\{\max\{\tau_k(t)\} - \min\{\tau_k(t)\}\}} \tag{6}$$

其中，$\tau_k(t)$ 为第 t 代时第 k 个方格内粒子的拥挤度。当拥挤度的最大值和最小值相差很大时，根据式(6)增加邻域粒子，使粒子和更多的粒子交换信息，反之减少邻域粒子，使粒子在局部搜索更仔细.

下面介绍 PSO 的两大拓扑结构 Gbest 拓扑和拟小世界拓扑，如图 1Gbest 拓扑结构是指每个粒子都与系统中的其他粒子相连，此时粒子的邻域结构比较庞大，粒子间信息的交互比较完善，导致粒子的收敛速度快，全局搜索能力较强，但容易陷入局部最优，出现早熟现象. 拟小世界拓扑结构是指每个粒子只与其周围的部分粒子相连，粒子间的信息交流较慢，而一旦一个粒子搜索到最优位置，最终这个信息也将逐步传播到整个群体，这种拓扑结构收敛的速度较慢，但具有较强的局部搜索能力，鲁棒性好. 本文将 PSO 的这 2 种拓扑结构很好地结合起来，当粒子的邻域粒子增加时，即 $\bar{N}_i > N_i$ 邻域中粒子的结构采用 Gbest 拓扑拓扑结构，当粒子的邻域粒子减少到 $\bar{N}_i < N_i$ 时，邻域中粒子的结构采用拟小世界拓扑结构，更加有效地平衡了粒子的全局搜索能力和局部搜索能力. 下面重点介绍拟小世界拓扑结构.



a Gbest拓扑　　b 拟小世界拓扑

图 1　拓扑结构

Fig.1　Topology

定义 3　节点 v_i 与节点 v_j 之间差异度的大小定义如下：

$$\rho_{ij} = \frac{|f(v_i) - f(v_j)|}{\sum_{\substack{j=1 \\ j \neq i}}^{\gamma_i} |f(v_i) - f(v_j)|} \tag{7}$$

式中，γ_i 为节点 i 的度. 节点差异度越小，相似度越大，说明这 2 个节点间的信息差异不大，这 2 个节点间无边相连，否则，节点的差异性越大，说明这 2 个节点间信息的差异性越大，此时要想达到快速收敛的目的，这 2 个节点间必须有边相连.

在实际操作中，粒子间起初是全连接拓扑结构，当 $\bar{N}_i < N_i$ 时，根据节点间差异度的大小，断开一些差异度较小的边，保留差异度较大的边. 为了不增加计算复杂度，在比较同一节点差异度的大小时，只需计算 $|f(v_i) - f(v_j)|$ 与给定值 Z 之间的关系，当 $|f(v_i) - f(v_j)| < Z$ 时，断开节点 v_i 与节点 v_j 之间的边，否则保留此边. 这里 Z 取为平均差异度. 最后给出了一个算法收敛的定理.

定理 1　算法 DTMOPSO 是以概率收敛于 pareto 前端的.

3　DTMOPSO 算法验证

3.1　算法的验证

为了对上述提出的 DMPSO 算法进行验证，先针对 2 个问题验证其可行性，然后再将其应用在实际问题中，验证算法的有效性. 参数的设置如下：支配度和拥挤度 $\alpha = \beta = 1/3$；约束因子 $\chi = 0.729$；加速因子 $\varphi = 4.1$；初始粒子规模 $N_0 = 100$；最大迭代次数 $N_c = 2\,000$. 下面给出 2 个经典的多目标函数的测

试例子，具体形式如下：

$$\begin{cases}\min f_1(x) = \sum_{i=1}^{2}(-10\exp(-2.0\sqrt{x_i^2 + x_{i+1}^2}))\\ \min f_2(x) = \sum_{i=1}^{3}(|x_i|^{0.8} + 5\sin x_i^3\\ \text{s.t. } -5 \leqslant x_i \leqslant 5, i = 1,2,3\end{cases} \tag{12}$$

$$\begin{cases}\min f_1(x_1,x_2) = x_1\\ \min f_2(x_1,x_2) = g(x_2)/x_1\\ g(x_2) = 2.0 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - \\ \qquad 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\}\\ \text{s.t. } 0.1 \leqslant x_i \leqslant 1.0, i = 1,2\end{cases} \tag{13}$$

使用上面给出的参数，得到的结果如表 1.

表 1　几种算法求解问题(12)和(13)的误差结果对比

Tab.1　Results comparison of DMPSO algorithm and other algorithms to problem (12) and (13)

算法	最优值	最差值	平均值	中值	标准差	平均计算时间/s
文献[18]结果	0.02,0	0.37,1.01	0.25,0.26	0.26,0.05	0.04,0.40	0.16,0.28
Micro-GA	0.18,0.08	0.36,1.01	0.27,0.25	0.25,0.16	0.05,0.23	0.33,0.14
DMPSO 结果	0.02,0	0.38,1.01	0.24,0.25	0.25,0.04	0.04,0.24	3.19,2.85
NSGA-Ⅱ	0.06,0.02	1.01,1.01	0.56,0.42	0.50,0.12	0.39,0.46	2.43,1.59

注：逗号前后分别为问题(12)和(13)对应的值

从表 1 中的结果可以看出，DMPSO 算法是有效的，但采用 DMPSO 算法得到满意解需要花费的时间比较长，也就是说 DMPSO 算法是在牺牲时间的基础上换回精度的. 主要是因为在进行备集更新和确定动态邻域的时候花费的时间比较大. 下一步将研究如何在确保精度的基础上使花费的时间也尽量减少.

3.2　算法应用于资源受限研发项目进度的问题

将 DMPSO 算法用于资源受限研发项目进度的问题中，并与文献[6]中的结果进行对比. 其中编码和解码方式采用[6]中的方式，图 2 为对比结果.

从图 2 可以看出 DMPSO 算法应用在资源受限研发项目进度问题中得到的结果是满意的，不管是实验工期还是实验资源方差，最后的结果都优于文献[6]的结果.

3.3　算法应用于多目标车辆路径问题

再将 DMPSO 算法应用在多目标车辆路径上的实验结果，编码和解码方式采用文献[16]中的方式，并于文献[16]中的结果进行了对比(表 2). 其中 C_{put} 是指程序运行时间；O_{pt} 定义为 Pareto 解集的最优指数；S_p 用来衡量 Pareto 解集张成的空间的标准，其值越小，说明解的分布越均匀；Q 表示问题名称，Q，O_{pt} 和 S_p 的具体表现形式见文献[16]；F_{avar} 为 Pareto 解的方差.

从表 2 可以看出，DMPSO 算法在多目标车辆路径上的应用也是有效的. 从 Pareto 解的方差来看，得到的 Pareto 解集的最优指数和解分布的均匀性都比文献[16]好，这是因为 DMPSO 算法中拥挤度的衡量在很大程度上提高了解分布的均匀性，同时采用的动态邻域结构在很大程度上也使解之间的信息传送更快、更有效，这也促进了最终得到解的有效性和均匀性. 但是 DMPSO 算法运行的时间花费比文献[13]多，主要原因是在计算储备集更新条件和确定动态邻域的过程花费的时间比较多.

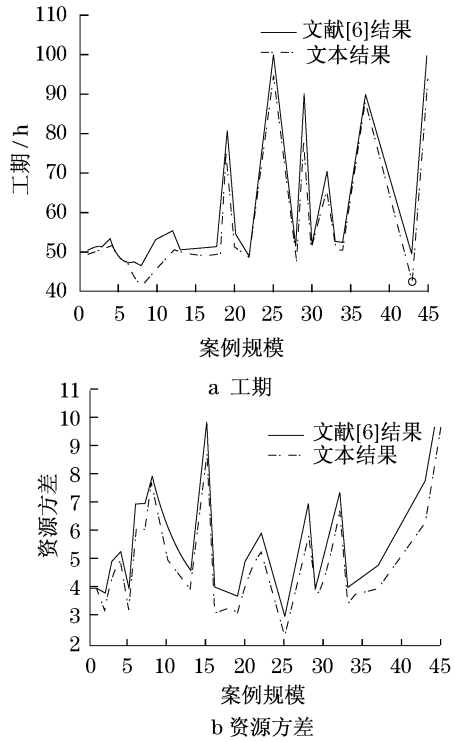


图 2　实验工期及资源方差对比

Fig.2　Graph of experimental period and resources variance comparison

表2 文献[16]和DMPSO算法结果对比

Tab.2 Result comparison of reference [16] and DMPSO algorithm

Q	C_{put}	O_{pt}	S_p	F_{avar}
C_{101}	3.8,4.3	8.9,8.3	4.9,4.1	289.3,278.9
C_{102}	4.3,5.4	6.7,6.4	5.6,5.3	283.3,269.2
C_{109}	3.9,3.9	5.6,4.4	5.0,3.8	299.4,230.9
C_{104}	4.2,4.7	7.9,6.9	6.1,5.3	298.8,261.1
C_{105}	4.2,5.0	5.6,5.3	6.7,6.4	298.8,284.0
R_{101}	4.4,5.2	7.9,7.5	13.9,13.3	746.7,713.7
R_{102}	4.6,5.3	8.1,7.8	17.2,16.3	741.8,705.9
R_{201}	6.3,7.0	7.8,7.4	14.6,13.8	645.4,610.9
R_{202}	5.3,6.1	7.8,7.4	14.2,13.5	632.9,604.5
R_{203}	6.5,7.1	6.8,6.3	9.5,8.9	609.9,570.9
R_{C203}	20.0,21.3	11.9,10.3	4.0,3.4	423.9,366.9
R_{C204}	21.1,22.7	14.0,13.1	7.1,6.6	421.2,393.5
R_{C206}	20.1,21.1	13.2,12.1	36.0,32.5	487.9,440.9
R_{C207}	20.9,22.6	11.1,10.4	33.0,30.9	521.3,487.2
R_{C208}	20.6,21.1	11.0,10.1	29.8,27.6	470.2,434.9

注:逗号前后分别为文献[16]和DMPSO算法结果.

4 结语

给出了一种动态邻域结构的多目标粒子群优化算法,并成功地将此算法应用在一些工程问题中,在储备集更新的过程中,以支配度和拥挤度的度量为依据,使得解分布的均匀性更优;在粒子搜索的过程中,采用了Gbest拓扑邻域结构和拟小世界拓扑邻域结构2种形式,并依据全局搜索和局部搜索的平衡性来动态调整粒子的拓扑结构.使得粒子的搜索更快,在一定程度上避免了早熟收敛现象.实验结果表明,DMPSO算法求解多目标优化问题是可行的、有效的.

参考文献:

[1] Leong Yen,Wen Fung , Gary G. PSO-based multiobjective optimization with dynamic population size and adaptive local archives [J]. IEEE Transactions on Systems, Man, and Cybernetics Part B:Cybernetics,2008,38(5):1270.

[2] Brits R,Engelbrecht A P,F. van den Bergh. Locating multiple optimization using particle swarm optimization [J]. Applied Mathematics and Computation,2007,189:1859.

[3] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, Sankar Kumar Pal. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients [J]. Information Sciences,2007,177:5033..

[4] 张勇,巩敦卫,郝国生,等.含区间参数多目标系统的微粒群优化算法[J].自动化学报,2008,34(8):921.
ZHANG Yong, GONG Dunwei, HAO Guosheng, et al. Particle swarm optimization for multi-objective system with interval parameters[J]. Acta Automatica Sinica,2008,34(8):921.

[5] 张利彪,周春光,刘小华,等.求解多目标优化问题的一种子群体进化算法[J].控制与决策,2007,22(11):1313.
ZHANG Libiao, ZHOU Chunguang, LIU Xiaohua, et al. A multiple sub-swarms evolutionary algorithm for multi-objective optimization problems [J]. Control and Decision, 2007, 22(11):1313.

[6] 曲红,吴娟.基于动态多目标粒子群优化算法的资源受限研发项目进度[J].系统工程,2007,25(9):98.
QU Hong, WU Juan. The resource-constrained & project scheduling problem based on dynamic multi-object particle swarm optimization algorithm[J]. Systems Engineering,2007,25(9):98.

[7] 丛琳,焦李成,沙宇恒.正交免疫克隆粒子群多目标优化算法[J].电子与信息学报,2008,30(10):2320.
CONG Lin, JIAO Licheng, SHA Yuheng. Orthogonal immune clone particle swarm algorithm on multi-objective optimization [J]. Journal of Electronics & Information Technology,2008,30(10):2320.

[8] LI Hui, ZHANG Qingfu. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-I[J]. IEEE Transactions on Evolution Computation,2009,13(1):284.

[9] Tan K C,Yang Y J,Goh C K. A distributed cooperative Co-evolutionary algorithm for multi-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5):527.

[10] Nebro A J,Luna F,Alba E,et al. AbYSS:adapting scatter search to multi-objective optimization [J]. IEEE Transactions on Evolutionary Computation,2008,12(4):439.

[11] Amosa S Bandyopadhyay,Saha S,Maulik U,et al. A simulated annealing-based multi-objective optimization algorithm [J]. IEEE Transactions on Evolutionary Computation, 2008, 12(3):269.

[12] Carlos A Coello, Coello Pulido Gregorio Toscano, Salazar Maximino, et al. Multiple objectives with particle swarm optimization [J]. IEEE Transactions on Evolutionary Computation,2004,8(3):256.

[13] Kennedy J,Eberhert R. Particle swarm optimization[C]// IEEE International Conference on Neural Networks. Perth: IEEE Press,1995(1):1942-1948.

[14] James Kennedy,Rui Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews,2006,36(4):515.

[15] Clerc M,Kennedy J. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space[J]. IEEE Trans Evol Comput,2002,6(1):58.

[16] 徐杰,黄德先.基于混合粒子群算法的多目标车辆路径研究[J].计算机集成制造系统,2007,13(3):573.
XU Jie,HUANG Dexian. Hybrid particle swarm optimization for vehicle routing problem with multiple objectives[J]. Computer Integrated Manufacturing Systems,2007,13(3):573.