基于粒子记忆体的多目标微粒群算法 *

章国安[1,2], 周 超[1], 周 晖[1]
(1. 南通大学 电子信息学院, 江苏 南通 226019; 2. 东南大学 移动通信国家重点实验室, 南京 210096)

摘 要: 针对多目标微粒群算法(MOPSO)解的多样性分布问题,提出一种基于粒子记忆体的多目标微粒群算法(dp-MOPSO)。dp-MOPSO 算法为每个微粒分配一个记忆体,保存寻优过程中搜索到的非支配 *p*best 集,以避免搜索信息的丢失。采用外部存档保存种群搜索到的所有 Pareto 解,并引入动态邻域的策略从外部存档中选择全局最优解。利用几个典型的多目标测试函数对 dp-MOPSO 算法的性能进行测试,并与两种著名的多目标进化算法 m-DNPSO、SPEA2 进行比较。实验结果表明,dp-MOPSO 算法可以更好地逼近真实 Pareto 沿,同时所得 Pareto 解分布更均匀。

关键词: 多目标优化; 微粒群算法; 记忆体; 多样性; *p*best
中图分类号: TP18 **文献标志码**: A **文章编号**: 1001-3695(2010)05-1665-04
doi:10.3969/j.issn.1001-3695.2010.05.015

New multi-objective particle swarm optimization based on extended individual memory

ZHANG Guo-an[1,2], ZHOU Chao[1], ZHOU Hui[1]
(1. *School of Electronics & Information*, *Nantong University*, *Nantong Jiangsu* 226019, *China*; 2. *National Mobile Communications Research Laboratory*, *Southeast University*, *Nanjing* 210096, *China*)

Abstract: To deal with the problem of diversity distribution of solution in multi-objective particle swarm optimization, this paper proposed a diversity pbest based multi-objective particle swarm optimization algorithm (dp-MOPSO). In dp-MOPSO, allocated each particle an individual memory to save the solution set of non-dominated *p*best which were fiound in the searching process, avoiding the loss of searching information. Used an external archive to save all the Pareto solutions, and introduced the dynamic neighborhood strategy to select the global optimal solution from the external archive. Tested several multi-objective benchmark functions for comparing the performance of dp-MOPSO with two famous multi-objective evolutionary algorithm m-DNPSO and SPEA2. The results show that dp-MOPSO converges to the true Pareto front more closely, and also all the Pareto solutions are well-distributed.

Key words: multi-objective optimization; particle swarm optimization(PSO); individual memory; diversity; *p*best

0 引言

多目标优化问题(multi-objective optimization problems, MOP)的各个目标往往相互冲突,难以求解。传统的求解 MOP 的方法存在许多不足,比如对 Pareto 沿比较敏感、要求目标函数和约束条件可微、每次运行只产生一个解、效率低等。1985 年,Schaffer 设计了向量评价遗传算法 VEGA,首次利用进化算法求解多目标优化问题,为多目标优化开辟了一条新途径。研究人员已经提出了多种多目标进化算法(multiple objectives evolutionary algorithm, MOEA), 如 MOGA、NSGA、NSGA-Ⅱ、NPGA、SPEA、SPEA2、PAES、PESA 等[1]。

微粒群算法(PSO)是 Kennedy 等人[2,3] 于 1995 年提出的一种基于种群的进化算法,源于对鸟群觅食行为的模拟。PSO 在单目标优化中表现优于遗传算法,但将微粒群算法应用于求解多目标优化问题相对较晚。2002 年,文献[4]首先发表了将 PSO 应用于求解 MOP 问题的成果,使得 PSO 应用于 MOP 成为新的研究热点。根据文献检索,目前已有 30 种不同的多目标微粒群算法(multiple objectives particle swarm optimization, MOPSO),这些算法各有特色。但基于 PSO 的多目标优化算法所得到解的分布性不理想[5]。

针对上述问题,本文提出基于粒子记忆体的多目标微粒群算法 dp-MOPSO(diversity *p*best based multiple objectives particle swarm optimization)。该算法为每个微粒分配一个记忆体,保存搜索信息以避免丢失。仿真实验表明,dp-MOPSO 算法能够避免局部次优,得到的解更逼近真实 Pareto 沿,分布更均匀。

1 多目标微粒群算法回顾

在已经提出的多目标微粒群算法中,本文选择几种具有代表性的算法加以介绍。

2002 年,Hu 等人[4] 提出动态邻域(dynamic neighborhood)

收稿日期: 2009-10-11; **修回日期**: 2009-11-17 **基金项目**: 国家"863"计划资助项目(2007AA01Z330);江苏省高校自然科学重大基础研究资助项目(07KJA51007);江苏省教育厅"青蓝工程"资助项目;东南大学移动通信国家重点实验室开放研究基金资助项目(W200912)

作者简介:章国安(1965-),男,江苏南通人,教授,硕导,博士,主要研究方向为宽带数字移动通信系统理论与技术、智能计算等;周超(1984-),男,江西吉安人,硕士研究生,主要研究方向为网络安全、智能计算等(jsdavid.acc@163.com);周晖(1963-),男,江苏南通人,教授,硕导,博士,主要研究方向为群集智能、网络优化等.

的概念,率先将 PSO 算法应用于多目标优化问题。每个微粒将其每一次迭代搜索到的邻居微粒保存在动态邻域中,每一次所产生的邻域都不同,全局最优解 *l*best 从动态邻域中选择。该算法在保持其他目标不变的前提下,每次只对一个目标进行优化。后来,Hu 等人[6]对该算法进行了改进,引入外部存档保存种群搜索到的所有 Pareto 最优解。

Coello 等人[7]引入 repositiry 用于保存种群搜索到的全局最优解,利用 Pareto 支配来控制微粒的飞行(进化)方向,提出了一种新的 MOPSO 算法。在此基础上,Coello 等人[8]又引入了一种特殊的变异操作算子(mutation operator),提出了一种具有更强搜索能力的多目标微粒群算法。

Fieldsend 等人[9]采用精英归档的方法将搜索到的 Pareto 解存入精英存档中,并在进化方程的速度公式中加入扰动项来影响微粒的飞行方向,提高了算法所得解的多样性。Mostaghim 等人[10]提出了一种新的选择全局最优解的方法(sigma method),算法在提高所得解的多样性的同时收敛得更快。

国内学者也对多目标微粒群算法进行了相关研究。文献[11,12]都采用拥挤度(或拥挤距离)寻找全局最优解,引入变异策略避免早熟,增强了算法的搜索能力。文献[13]提出基于偏好信息的多目标微粒群算法,以栅格策略保持解的多样性。文献[14]通过定义概率支配关系比较所得解的优劣,给出了新的个体最优、全局最优以及外部存档的更新方法。

2　dp-MOPSO 算法

2.1　多目标优化问题

一般地,设 R^n 为 n 维欧氏空间,$D\in R^n$,$\boldsymbol{f}:D\to R^k$ 为目标函数,$g_j(\boldsymbol{x})$,$h_j(\boldsymbol{x})$ 为 D 上的约束函数,$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in R^n$,多目标优化问题定义为

最优化:$\boldsymbol{f}(\boldsymbol{x})=[f_1(\boldsymbol{x}),f_2(\boldsymbol{x},\cdots,f_k(\boldsymbol{x})]$

约束条件:$g_j(\boldsymbol{x})\leqslant 0$ for $j=1,\cdots,p$ and $h_j(\boldsymbol{x})=0$ for $j=p+1,\cdots,m$

在最小化问题中,Pareto 支配、Pareto 最优解、Pareto 沿定义如下。

定义 1　Pareto 支配。

设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$,$\boldsymbol{y}=(y_1,y_2,\cdots,y_n)$,当且仅当 $\forall i\in\{1,2,\cdots,n\}$,$x_i\leqslant y_i$,且 $\exists i\in\{1,2,\cdots,n\}$,$x_i<y_i$,称 $\boldsymbol{x}$ 支配 $\boldsymbol{y}$。

定义 2　Pareto 最优解。

当且仅当不存在 $\boldsymbol{f}\in F$(F 为可行解空间)支配 $\boldsymbol{f}^*\in F$,则称 $\boldsymbol{f}^*$ 为一个 Pareto 最优解。

由上述定义可知,Pareto 最优解不是单一解,而是一组解,也称这组解为非劣解或者非支配解。

定义 3　Pareto 沿。

所有的 Pareto 最优解组成多目标优化问题的 Pareto 沿。

2.2　dp-MOPSO 算法

微粒群算法应用于单目标优化时,按照式(1)和(2)进行速度和位置更新。

$$V_i=w\times V_i+c_1\times r_1\times(p\text{best}_i-X_i)+c_2\times r_2\times(l\text{best}_i-X_i)\quad(1)$$

$$X_i=X_i+V_i\quad(2)$$

其中:X_i 为微粒的位置;V_i 为微粒的速度;w 为惯性权重;c_1 和 c_2 为学习因子;r_1 和 r_2 为两个介于(0,1)之间的随机数;*p*best 为微粒经历的历史最好位置;*l*best 为种群中所有微粒经历的历史最好位置。式(1)的第一部分为惯性部分,由微粒的先前速度决定;第二部分为个体认知部分,表示微粒的自身经验,引导微粒向自身最好位置进化;第三部分为社会认知部分,表示微粒间的协同合作和信息共享,引导微粒向群体的最好位置进化。微粒就是这样通过自身的经验和同伴中最好的经验来决定自己的进化方向的。

将微粒群算法从单目标优化拓展到多目标优化,关键在于改进 *p*best 和 *l*best 的选择方式。在已有算法中,*l*best 主要是从外部存档的 *p*areto 解中选择。*p*best 的选择方式分为以下几种:

a)只有当微粒的当前位置支配其 *p*best 时,才更新 *p*best;否则不更新,如文献[4]。

b)如果微粒的当前位置支配其 *p*best,则用当前位置更新 *p*best;如果两者互不支配,则从两者中随机选择一个更新 *p*best,如文献[6]。

c)如果微粒的当前位置支配其 *p*best,则用当前位置更新 *p*best;如果两者互不支配,则直接用当前位置更新 *p*best,如文献[9]。

方式 a)将搜索到的与微粒当前 *p*best 互不支配的解全部丢弃,只保留支配当前 *p*best 的解;方式 b)和 c)在 a)的基础上有所改进,但当搜索到的解与微粒当前 *p*best 互不支配时,b)会丢弃搜索到的非劣解(不更新 *p*best 时),或丢弃历史非劣 *p*best 解(用搜索到的解更新 *p*best 时),方式 c)会丢弃历史非劣 *p*best 解。这三种方式均保留一个非劣 *p*best 解,而多目标优化在一次搜索中能找到多个非劣解,这势必会造成寻优信息的丢失。

本文提出一种新算法 dp-MOPSO,种群中的每个微粒所带的记忆体保存微粒寻优中各自搜索到的非支配 *p*best 集,这样每个微粒的 *p*best 不再是单一解,而是一个解集。这对于提高解的多样性,避免算法陷入局部次优具有重要意义。以两目标函数求最小值为例,种群中每个微粒的记忆体都保存有 m 个非支配 *p*best,如图 1 所示。

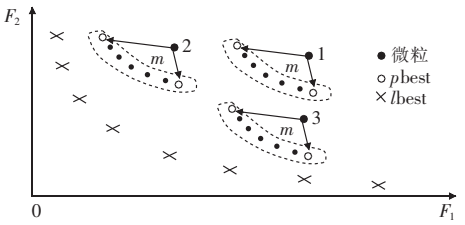


图 1　微粒记忆体存储非支配 *p*best 集

在进化搜索过程中,微粒的当前位置与记忆体中的非劣 *p*best 解比较 Pareto 支配情况,记忆体中被微粒当前位置支配的解将被删除,对记忆体中的解而言是非劣解的新解将被存入记忆体中,这样微粒的记忆体就能得到不断更新。具体的 *p*best 集操作过程如图 2 所示。

引入外部存档用于存储种群在进化中搜索到的所有 Pareto 解,采用文献[6]中的方式将需要优化的多个目标分为两组 F_1 和 F_2(F_1 为邻域目标,F_2 为优化目标),先利用 F_1 计算当前微粒与外部存档中 Pareto 解的距离,选出与当前微粒距离最近的 m 个 Pareto 解组成当前微粒的邻域,再利用 F_2 从邻域中选择一个解作为 *l*best。

算法 dp-MOPSO 描述如下:

a)初始化

设置算法参数,随机初始化各个微粒的位置和速度,并限制其变化范围。

b)评价微粒的适应值

计算种群中各个微粒的每一个目标适应值,并以矢量形式存储。

c)更新外部存档,即更新 Pareto 解集

在目标适应值空间比较当前一代微粒之间的 Pareto 支配情况,得到当前一代的 Pareto 解;这些解与外部存档中的解再次比较支配情况,更新外部存档。如果是第一代搜索,此时外部存档为空,将得到的第一代 Pareto 解直接存入外部存档。

d)选择 *p*best 和 *l*best

具体方式如前文所述。

e)更新微粒的速度和位置

微粒从各自记忆体中随机选择一个 *p*best,与所选择的 *l*best 一起代入更新公式对微粒的速度和位置进行更新。当速度和位置超过各自设定的变化范围时,将其设置为对应的边界值。

f)迭代

判断是否达到最大迭代次数,若没有,则转至 b)。

g)输出 Pareto 沿

达到迭代次数后输出外部存档中的解得到 Pareto 沿。

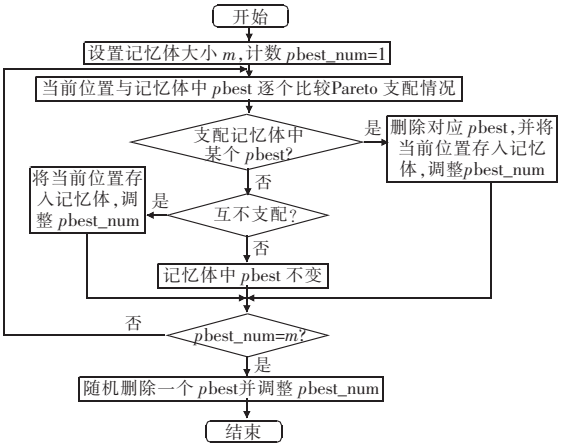


图 2　*p*best 集操作流程图

3　仿真实验

3.1　测试函数

采用 Zitzler 等人[15]中提出的测试函数对多目标优化算法进行性能测试。测试函数的具体形式如表 1 所示。

表 1　测试函数

name	dim	range	objective function
T_1	30	$0\leqslant x_i\leqslant 1$	$\min f_1(x)=x_1$ $\min f_2(x)=g(x)(1-\sqrt{x_1/g(x)})$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$
T_2	30	$0\leqslant x_i\leqslant 1$	$\min f_1(x)=x_1$ $\min f_2(x)=g(x)(1-(x_1/g(x))^2)$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$
T_3	30	$0\leqslant x_i\leqslant 1$	$\min f_1(x)=x_1$ $\min f_2(x)=g(x)[1-\sqrt{x_1/g(x)}-\frac{x_1}{g(x)}\sin(10\pi x_1)]$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$
T_4	30	$0\leqslant x_i\leqslant 1$	$\min f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1)$ $\min f_2(x)=g(x)[1-(f(x)/g(x))^2]$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$

3.2　算法性能的评价标准

多目标优化算法通常利用算法的收敛性与非劣解的多样性衡量算法的性能。

1)收敛性

$$GD=\frac{1}{n}\sqrt{\sum_{i=1}^{n}d_i^2} \tag{3}$$

其中:n 为算法所得非劣解的个数;d_i 为第 i 个非劣解到真实 pareto 最优解的最小欧氏距离。

2)多样性

$$SP=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \tag{4}$$

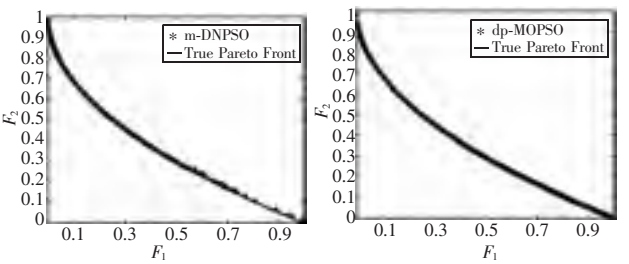
其中:d_i 为相邻两个非劣解之间的欧氏距离;$\bar{d}$ 为所有 d_i 的均值;n 为非劣解的个数。

3.3　实验及结果分析

用表 1 的测试函数对 dp-MOPSO 算法与文献[6]中的 m-DNPSO 算法进行实验,并比较两者的性能。

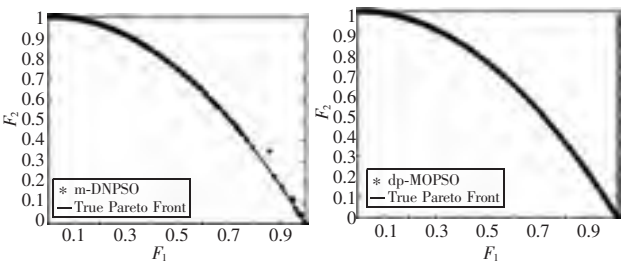
为了便于与 m-DNPSO 进行比较,实验参数的设置必须一致。种群大小为 20,最大的迭代次数为 250,每个测试函数执行 10 次。惯性权重为(0.5 + Rand/2.0),学习因子皆为 1.494 45,最大速度 V_{max} 设置为微粒每一维的数值变化范围,外部存档的大小为 200,邻域大小为 10。分配给每个微粒的记忆体大小可根据实际情况设定,本次实验设为 10。

图 3 ~6 是 m-DNPSO 和 dp-MOPSO 所得的四个测试函数的 Pareto 沿与其真实 Pareto 沿的比较图。从这些图中可以直观地看到两种算法所得非劣解与真实非劣解之间的逼近程度,以及两种算法所得非劣解的多样性差别。表 2 和 3 是 dp-MOPSO、m-DNPSO、SPEA2 三种算法的收敛性标准和多样性标准的均值和方差。SPEA2 为每个测试函数运行 20 次,每次迭代 250 次的结果[16]。



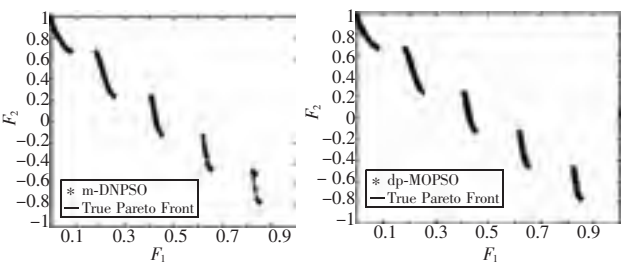
(a) m-DNPSO 得到的 T_1 的 Pareto 沿　　(b) dp-MOPSO 得到的 T_1 的 Pareto 沿

图 3　两种算法得到的 T_1 的 Pareto 沿



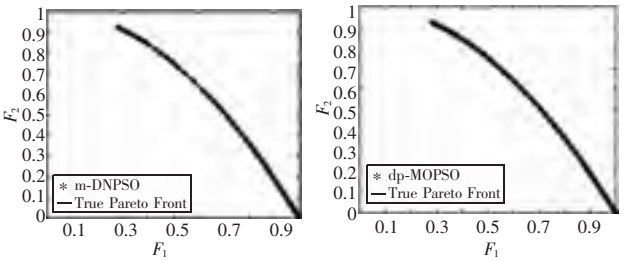
(a) m-DNPSO 得到的 T_2 的 Pareto 沿　　(b) dp-MOPSO 得到的 T_2 的 Pareto 沿

图 4　两种算法得到的 T_2 的 Pareto 沿



(a) m-DNPSO 得到的 T_3 的 Pareto 沿　　(b) dp-MOPSO 得到的 T_3 的 Pareto 沿

图 5　两种算法得到的 T_3 的 Pareto 沿



(a) m-DNPSO 得到的 T_4 的 Pareto 沿　(b) dp-MOPSO 得到的 T_4 的 Pareto 沿

图 6　两种算法得到的 T_4 的 Pareto 沿

从图 3 ~6 可以看出,对于四个测试函数,m-DNPSO 所得解在部分区域不能完全覆盖真实 Pareto 沿,解的分布也不是很均匀,而本文所提算法 dp-MOPSO 得到的非劣解分布更均匀,而且所得 Pareto 沿与真实 Pareto 沿基本上完全重合。

表 2　收敛性标准 GD 的值

算法	GD	T_1	T_2	T_3	T_4
m-DNPSO	均值	0.00026279	0.0088413	0.019184	0.0050645
	方差	3.1271e-008	4.2933e-005	8.5653e-006	5.3665e-006
SPEA2	均值	0.023285	0.16762	0.018409	0.232551
	方差	0	0.000815	0	0.004945
dp-MOPSO	均值	2.9325e-005	0.0067419	0.015383	0.0037082
	方差	1.0314e-010	1.3987e-006	5.9019e-005	8.0149e-007

表 3　多样性标准 SP 的值

算法	SP	T_1	T_2	T_3	T_4
m-DNPSO	均值	0.0078515	0.0077942	0.015145	0.052081
	方差	2.966e-006	3.3199e-005	1.8334e-005	0.002491
SPEA2	均值	0.154723	0.33945	0.4691	1.04422
	方差	0.0008738	0.001755	0.005265	0.158106
dp-MOPSO	均值	0.0023879	0.0013257	0.0075624	0.0011348
	方差	2.7418e-007	2.3023e-007	7.466e-006	6.5169e-008

从表 2 可以看出,dp-MOPSO 与 m-DNPSO 的收敛性均优于 SPEA2(GD 值都更小),但 SPEA2 在 T_1、T_3 上具有更好的稳定性(方差为 0);dp-MOPSO 的收敛性优于 m-DNPSO(GD 值更小),就算法的稳定性而言,两者相差不大。

表 2 表明,与 SPEA2 相比,m-DNPSO、dp-MOPSO 所得非劣解都能更好地收敛于真实的 Pareto 沿。

从表 3 可看出, dp-MOPSO 与 m-DNPSO 所得非劣解的多样性均优于 SPEA2,且算法的稳定性更强。dp-MOPSO 所得非劣解的多样性优于 m-DNPSO。这是因为在 dp-MOPSO 中,每个微粒的记忆体保存搜索到的非劣 *p*best 解集,微粒在进化中具有更多的选择性,可以很好地避免陷入局部次优。表 3 表明,dp-MOPSO 所得非劣解具有更好的多样性,且算法比较稳定。

从实验结果及其分析可以看出,dp-MOPSO 算法在收敛性和多样性方面比 m-DNPSO、SPEA2 表现优越,说明 dp-MOPSO 算法所提带粒子记忆体的策略对提高算法的收敛性和所得解的多样性具有很好的效果。

4　结束语

进化算法中,特别是多目标进化算法中,搜索非常重要,它决定了一个算法的性能,但是对已经搜索到的信息进行保存、处理和利用同样重要,其作用不亚于前者,否则,搜索信息不能充分利用,算法的性能同样不会提高。本文提出一种新的多目标微粒群算法 dp-MOPSO,在搜索信息的保存和利用方面作出了探索,实验结果验证了本文算法的有效性,从而说明了有效利用搜索信息的重要性。这方面的研究工作才刚刚开始,如何充分利用和有效处理多目标优化中已搜索到的信息,有待进一步的研究。

参考文献:

[1] 郑向伟,刘弘. 多目标进化算法研究进展[J]. 计算机科学,2007,34(7):187-192.

[2] KENNEDY J,EBERHART R C. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. 1995: 1942-1948.

[3] EBERHART R C,SHI Yu-hui. Particle swarm optimization: developments, applications and resources [C]//Proc of IEEE Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 2001:81-86.

[4] HU Xiao-hui,EBERHART R C. Multi-objective optimization using dynamic neighborhood particle swarm optimization [C]//Proc of IEEE Congress on Evolutionary Computation. Honolulu, HI: IEEE Press, 2002:1677-1681.

[5] FIELDSEND J E. Multi-objective particle swarm optimization methods,Technical Report No. 419, Exeter [R]. [S. l.]: Department of Computer Science, University of Exeter,2004.

[6] HU Xiao-hui,EBERHART R C,SHI Yu-hui. Particle swarm with extended memory for multi-objective optimization [C]//Proc of IEEE Swarm Intelligence Symposium. 2003:193-197.

[7] COELLO C, CARLOS A, LECHUGA M S. MOPSO: a proposal for multiple objective particle swarm optimization [C]//Proc of IEEE Congress on Evolutionary Computation. Honolulu, HI: IEEE Press, 2002:1051-1056.

[8] COELLO C,CARLOS A,PULIDO G T,*et al*. Handing multiple objectives with particle swarm optimization [J]. IEEE Trans on Evolutionary Computation,2004,8(3):256-279.

[9] FIELDSEND J E,SINGH S. A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence [C]//Proc of the 2002 U. K. Workshop on Computational Intelligence. 2002:37-44.

[10] MOSTAQHIM S,TEICH J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO) [C]//Proc of IEEE Swarm Intelligence Symposium. 2003:26-23.

[11] 王辉,钱锋. 基于拥挤度与变异的动态微粒群多目标优化算法[J]. 控制与决策,2008,23(11):1238-1242.

[12] 李中凯,谭建荣,冯毅雄,等. 基于拥挤距离排序的多目标粒子群优化算法及其应用[J]. 计算机集成制造系统,2008,14(7):1329-1336.

[13] 余进,何正友,钱清泉. 基于偏好信息的多目标微粒群优化算法研究[J]. 控制与决策,2009,24(1):66-70.

[14] 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):921-928.

[15] ZITZLER E,DEB K,THIELE L. Comparison of multi-objective evolutionary algorithms: empirical results [J]. Evolutionary Computation,2000,8(2):173-195.

[16] 张利彪,周春光,马铭,等. 基于极大极小距离密度的多目标微分进化算法[J]. 计算机研究与发展,2007,44(11):177-184.