

用于约束优化的简洁多目标微粒群优化算法

张 勇¹, 巩敦卫¹, 任永强^{1,2}, 张建华¹

(1. 中国矿业大学信息与电气工程学院, 江苏徐州 221116; 2. 徐州空军学院, 江苏徐州 221002)

摘 要: 本文提出了一种少控制参数的约束多目标微粒群优化算法. 该算法利用关于微粒全局和个体最优点的高斯分布来更新微粒的位置, 无需设置惯性权重和学习因子等控制参数; 利用非可行储备集保存所得非可行解, 给出一种改进的储备集更新方法; 为均衡微粒对未知可行域和已知可行域的开发/探索能力, 提出一种线性递减策略, 用来分配微粒从非可行储备集中选择全局最优点的概率. 最后, 实验验证了所提算法的有效性.

关键词: 多目标优化; 约束; 微粒群; 高斯分布

中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2011) 06-1436-05

Barebones Multi-Objective Particle Swarm Optimizer for Constrained Optimization Problems

ZHANG Yong¹, GONG Dun-wei¹, REN Yong-qiang^{1,2}, ZHANG Jian-hua¹

(1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China;

2. Xuzhou Air Force College, Xuzhou, Jiangsu 221002, China)

Abstract: This paper presents a constrained multi-objective particle swarm optimization algorithm with few control parameters to solve constrained multi-objective optimization problems. In this algorithm, a Gaussian distribution based on the global/local best positions is developed to update the particles' positions. It makes unnecessary to perform fine tuning on such control parameters as inertia weight and acceleration coefficients. Using an infeasible archive to save infeasible solutions, an improved update method of the infeasible archive is proposed. In order to balance the algorithm's capabilities to exploit known feasible regions and to explore unknown feasible regions, a linear decreasing strategy is introduced to assign the probability, based on which the particles select their global best positions from the infeasible archive. Finally, feasibility of the proposed algorithm is validated by simulation results.

Key words: multi-objective optimization; constraint; particle swarm optimization; Gaussian distribution

1 引言

在日常生活和工程科学等领域, 存在很多含有约束的多目标优化问题 (Constrained Multi-objective Optimization Problems, CMOP). 一种有效的优化技术必须能够在处理约束的同时求得上述问题的 Pareto 最优解集.

微粒群优化^[1] (Particle Swarm Optimization, PSO) 作为一种群体智能优化技术, 由于具有收敛速度快和操作简单等优点, 近些年来已广泛用于各类无约束的多目标优化问题^[2-5]. 然而, 对于含约束多目标优化问题, 相关研究成果较少, 并且已有文献大都采用源于多目标进化算法的直接比较法^[2,6].

为处理 CMOP, 本文提出一种少控制参数的简洁多目标微粒群优化算法 (BareBones Multi-Objective Particle Swarm Optimizer, BB-MOPSO). 具体工作如下: ①为帮助微粒以更快的速度构建优秀模块, 提出一种简洁的微粒

更新方法. 该方法利用高斯分布更新微粒的位置, 具有无需设置惯性权重和学习因子等控制参数的优点; ②提出一种新的非可行储备集的更新方法, 将支配可行储备集中元素的非可行解, 以及未被可行储备集中元素支配且位于稀疏 (已有解较少) 区域的非可行解, 同时保存在非可行储备集中. ③为均衡算法的全局搜索和局部开发能力, 给出一种线性递减策略, 用来分配微粒从非可行储备集中选择全局最优点的概率.

2 相关知识及文献综述

2.1 微粒群优化算法

PSO 源于对鸟类和鱼群等捕食行为的模拟. 在鸟类捕食的群体行为中, 每只鸟被看作一个微粒, 而每个微粒代表被优化问题的一个解. 第 i 个微粒在 D 维空间里的位置表示为矢量 $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$, 飞行速度表示为矢量 $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$. 每个微粒都有一个

由优化函数决定的适应值,并且知道目前为止自己发现的最好位置 $\mathbf{p}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,D})$,即常说的微粒个体最优值,以及目前为止邻域微粒发现的最好位置 $\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_D)$,即通常说的微粒全局最优值.第 $t+1$ 次迭代时,微粒根据自身经验和同伴微粒的经验决定下一步运动的状态.以带惯性权重的基本 PSO 算法^[6]为例,微粒位置和速度的更新公式如下:

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + r_1c_1(p_{ij}(t) - x_{ij}(t)) \\ &\quad + r_2c_2(g_j(t) - x_{ij}(t)) \\ x_{ij}(t+1) &= x_{ij}(t) + v_{ij}(t+1) \end{aligned} \quad (1)$$

其中, w 为惯性权重, c_1 和 c_2 为学习因子, r_1 和 r_2 为 $[0,1]$ 间随机数; $j=1,2,\dots,D$, $i=1,2,\dots,N_S$, N_S 为微粒规模.

2.2 约束多目标微粒群优化算法

在处理含约束多目标优化问题时,大多数微粒群优化算法采用的是直接比较方法^[2,6].对于任意两个解,直接比较方法如下:①如果它们都为可行解,那么,由 Pareto 支配关系决定其优劣;②如果一个为可行解,另一个为非可行解,那么,可行解优于非可行解;③如果它们都为非可行解,那么,约束违背程度小的解优于约束违背程度大的解.

文献[7]指出,对于一些位于全局最优值附近的、高适应值的非可行解,如果进一步利用它们,将增加算法找到最优解的可能性.基于此,作者给出了一种改进的直接比较法.对于给定的常数 ϵ ($\epsilon > 0$) 和任意两个解:①如果两个解都是可行的,适应值大的一个胜出;②如果它们都为非可行解,那么,两个解都被淘汰;③如果一个为可行解,另一个为非可行解但其违背约束的程度小于 ϵ ,那么,适应值大的一个胜出.

为提高微粒对未知可行域的搜索能力,文献[8]采用可行和非可行储备集分别存储迭代中所得到的可行非劣解和非可行非劣解,并以相同概率从两个储备集中选择微粒的全局最优值.进一步,作者提出了一种基于可行解比率(微粒群中可行解所占的比例)和非可行解违背(约束)程度的自适应学习因子.

3 算法描述

3.1 含约束支配关系

考虑约束违背程度和 Pareto 支配关系,给出用于比较微粒优劣的含约束支配关系.对于微粒 \mathbf{x} ,由下式计算其约束违背程度:

$$cv(\mathbf{x}) = \frac{1}{K} \sum_{j=1}^K cv_j(\mathbf{x}) = \frac{1}{K} \sum_{j=1}^K \frac{dc_j(\mathbf{x})}{dc_{\max}^j} \quad (2)$$

$$dc_j(\mathbf{x}) = \begin{cases} \max\{|h_j(\mathbf{x})| - \delta, 0\}, & j=1,2,\dots,J \\ \max\{g_j(\mathbf{x}), 0\}, & j=J+1,\dots,K. \end{cases}$$

$$dc_{\max}^j = \max_{\mathbf{x} \in S} dc_j(\mathbf{x}), j=1,2,\dots,K.$$

其中, $h_j(\mathbf{x})=0$ 和 $g_j(\mathbf{x}) \leq 0$ 分别为等式和不等式约束函数, δ 为关于等式约束违背程度的容忍系数, S 为当前微粒群, K 为约束数目.如果 $dc_j(\mathbf{x})$ 大于 0,则表示 \mathbf{x} 违背第 j 个约束.

考虑包含 M 个目标函数的优化问题,对于给定的微粒 \mathbf{x} 和 \mathbf{y} ,如果它们满足下述条件之一:① $cv(\mathbf{x}) < cv(\mathbf{y})$;② $cv(\mathbf{x}) = cv(\mathbf{y})$,但对于 $\forall i \in \{1,2,\dots,M\}$ 有 $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$,且 $\exists j \in \{1,2,\dots,M\}$ 满足 $f_j(\mathbf{x}) < f_j(\mathbf{y})$,则称 \mathbf{x} 含约束支配 \mathbf{y} .

3.2 可行和非可行储备集的更新

利用非可行解搜索可行最优解有两个目的:一是,在搜索过程中通过有效均衡可行和非可行解的数目,可以改善种群的多样性;二是,利用非可行解作为开发孤立可行域的桥梁,不仅可以搜索更好的可行解,而且有助于处理(相对整个搜索空间)可行解比例较小的约束优化问题^[8].为提高非可行储备集中元素的质量,本文给出一种新的储备集更新方法.

在每次迭代时,为更新算法的两个储备集,首先把新生微粒划分为可行解和非可行解两类.基于上述划分,先更新可行储备集,再更新非可行储备集.

利用 Pareto 支配关系更新可行储备集.首先,合并可行储备集中已有元素和微粒群中新生可行解,组成一个新种群;接着,利用 Pareto 支配关系,选出该种群中互不支配的元素,并将这些元素保存到可行储备集中.如果可行储备集中元素的数目超过其固有容量 N_a ,则利用文献[9]中方法,计算每个元素的拥挤距离值,并保留分布最稀疏,即具有较大拥挤距离值的 N_a 个元素.

基于更新后的可行储备集,再更新非可行储备集.首先,合并非可行储备集中已有元素和微粒群中新生非可行解,组成一个新种群;接着,从该种群中重新选择进入非可行储备集的元素.对于新种群中的一个元素,如果可行储备集中存在一个元素被它支配,或者它与可行储备集中元素互不支配但恰好位于(已有可行非劣解较少的)稀疏区域,那么,保存该元素到非可行储备集.为判断一个元素是否落在稀疏区域,我们先把它放入可行储备集中,并计算其拥挤距离值.如果它的拥挤距离值较大,那么有理由认为该元素位于稀疏区域.由此方法,依次验证新种群中所有元素.如果保存在非可行储备集中元素的数目超过了其固有容量 N'_a ,那么,计算每个元素的拥挤距离值,并保留拥挤距离值较大的 N'_a 个元素.

当采用上述方法更新非可行储备集时,目标值较高和分布性较好的两类非可行解被保存下来.选择前一类非可行解作为微粒的全局最优值,具有改善可行

解的潜能;选择后一类非可行解作为微粒的全局最优点,将提高微粒的全局搜索能力。

3.3 微粒全局最优点的选择

为有效均衡微粒对未知可行域和已知可行域的探索和开发能力,本文给出一种动态分配策略,以线性递减的概率从非可行储备集中选择微粒的全局最优点。在算法迭代过程中,如果产生的随机数 $r_3 \in [0, 1]$ 小于概率 $p_s (0 \leq p_s \leq 1)$, 则从非可行储备集中选择当前微粒的全局最优点,否则,从可行储备集中选择当前微粒的全局最优点,其中

$$p_s = 0.5 + 0.3(1 - 2t/T_{\max}) \quad (3)$$

T_{\max} 为算法终止代数。由式(3)可以看出,在算法初期阶段,微粒以较大的概率从非可行储备集中选择它的全局最优点,这有助于保持微粒群的多样性,使微粒搜索到更多的可行区域。随着迭代次数的增加,微粒从可行储备集中选择全局最优点的概率不断增大。这意味着,在迭代后期,算法将有更多的机会开发已知可行区域,达到深度开发已得可行解的目的。

当非可行储备集为空集时,仅从可行储备集中选择微粒的全局最优点;同样,当可行储备集为空集时,仅从非可行储备集中选择微粒的全局最优点。

3.4 微粒位置的更新

为克服传统 PSO 算法过分依赖惯性权重等控制参数的不足,本文利用一种关于微粒个体最优点和全局最优点的高斯分布来更新微粒的位置:

$$x_{i,j} = \begin{cases} N(r_4 p_{i,j} + (1 - r_4) g_{i,j}, |p_{i,j} - g_{i,j}|), & \text{if } U(0, 1) < 0.5 \\ g_{i,j}, & \text{otherwise} \end{cases} \quad (4)$$

其中 r_4 为 $[0, 1]$ 间随机数。

上述更新公式无需设置惯性权重和学习因子等控制参数。每个新生微粒的位置由高斯采样点和微粒全局最优点共同决定,类似进化算法中的交叉算子,这将帮助微粒以更快的速度构建出优秀的进化模块。此外,由于每个微粒的全局最优点随机选自包含若干元素的储备集,因此,当以 50% 的概率保留微粒全局最优点信息时,不会导致微粒群丧失其多样性。

3.5 算法步骤

BB-MOPSO 的步骤可描述为:

Step1 设定所需参数,包括微粒群规模 N_s , 可行储备集和非可行储备集的容量 N_a 和 N'_a , 算法终止代数 T_{\max} 等;

Step2 初始化微粒群、微粒的个体最优点,将可行储备集和非可行储备集设为空集;

Step3 计算微粒群中每个微粒的约束违背程度及

适应值;

Step4 将微粒划分为可行解和非可行解两类,并利用第 3.2 小节所提方法依次更新可行储备集和非可行储备集;

Step5 判断是否满足终止准则,即是否达到终止代数。若满足,则停止算法;

Step6 对每个微粒,依次执行如下操作:

①由第 3.3 小节方法,从非可行储备集和可行储备集中选择其全局最优点;

②取微粒目前为止找到的最佳位置作为它的个体最优点;

③由公式(4)产生新的微粒位置。

Step7 返回 Step3。

3.6 算法复杂性分析

BB-MOPSO 算法的计算花费主要集中在储备集的更新操作。不失一般性,考虑一个包含 M 个目标和 K 个约束的优化问题。新生微粒群中可行解和非可行解的数目分别为 N'_s 和 N'_a 。

首先,分析可行储备集的更新过程。在规模为 $N'_s + N_a$ 的新种群中,判断一个微粒的优劣需要进行 $M \times (N'_s + N_a)$ 次比较。考虑最差情况(种群中所有微粒皆是互不支配的),从 $N'_s + N_a$ 个微粒中选出全部非劣解需要 $M \times (N'_s + N_a)^2$ 次比较;另一方面,采用拥挤距离维护储备集时的计算复杂度为 $O(M \times (N'_s + N_a) \times \log(N'_s + N_a))^{[8]}$ 。由此,更新可行储备集的计算复杂度为 $O(M \times (N'_s + N_a)^2)$ 。

接着,分析非可行储备集的更新过程。判断新种群(新生非可行解+非可行储备集)中一个微粒是否被可行解支配,需要 $M \times N_a$ 次比较。如果该微粒与可行储备集中所有元素是互不支配的,那么,需要计算它在可行储备集中的拥挤距离值。因为在计算微粒的拥挤距离值时,需要针对每一个目标函数进行适应值排序,因此,此时需要 $M \times (1 + N_a)$ 次比较。考虑最差的情况(相对可行储备集,新种群中所有微粒皆是非劣的),上述两个过程的计算复杂度为 $O(M \times (1 + N_a) \times (N'_s + N'_a))$ 。当上述两个过程保存下来的元素数目 N_1 大于 N'_a 时,需采用拥挤距离维护非可行储备集,其计算复杂度为 $O(M(N_1) \log(N_1))$ 。而当 $N_1 = N'_s + N'_a$ 时,非可行储备集的维护花费最大,为 $O(M(N'_s + N'_a) \log(N'_s + N'_a))$ 。由于 $\log(N'_s + N'_a) < (1 + N_a)$, 因此,更新非可行储备集的计算复杂度为 $O(M \times (1 + N_a) \times (N'_s + N'_a))$ 。

综上所述, BB-MOPSO 算法的计算复杂度为

$$O(M \times (N'_s + N_a)^2) + O(M \times (1 + N_a) \times (N'_s + N'_a))。$$

当 $N_a \geq N'_a$ 且新生微粒全为可行解时,上式可简

化为 $O(M(N'_s + N_a)^2)$.

4 实验及分析

4.1 实验

选择 BNH、OSY、CON、TNK、CTP2、CTP4、CTP5 和 CTP7 作为测试函数,其表达式见文献[9].

选用基于直接比较法的 MOPSO 算法(CMOPSO)^[2]和约束多目标微粒优化算法(Co-MOPSO)^[8]作为比较算法,参数设置如下:

①CMOPSO 算法:种群规模和储备集规模取 100,网格划分数目为 30,变异概率 = 0.5,惯性权重 $w = 0.4$.

②Co-MOPSO 算法:种群规模取 100,可行储备集和非可行储备集规模取 100,惯性权重为 [0.1, 0.7] 中随机变化值,学习因子采用动态调整策略.

对于本文所提 BB-MOPSO 算法,种群规模和两个储备集的规模皆取 100.3 种算法设置相同的函数评价次数 50000 作为终止条件.采用 Spacing-测度(SP)^[3]评价算法所得结果的分布性;采用 Two Set Coverage(TSC)^[10]测度比较算法所得结果的收敛程度. $TSC(A1, A2) = 1$ 表示算法 A1 所得结果支配算法 A2 所得结果或者两种算法得到相似的最优解集.

4.2 结果分析

在相同硬件条件下,利用 3 种算法分别优化测试函数 30 次,表 1 和表 2 分别出示了所得关于 SP 和 TSC 测度的统计结果.表 2 中 A 表示本文所提算法, B 表示 CMOPSO 算法, C 表示 Co-MOPSO 算法.

表 1 3 种算法所得关于 SP 测度的统计结果

函数	BB-MOPSO		CMOPSO		Co-MOPSO	
	均值	方差	均值	方差	均值	方差
BNH	0.4214	0.0506	0.8244	0.0543	0.5284	0.0286
OSY	1.1799	0.1577	1.6119	1.2149	2.2169	0.5894
CON	0.0391	0.0014	0.0489	0.0040	0.0427	0.0026
TNK	0.0061	0.0007	0.0071	0.0008	0.0039	0.0006
CTP2	0.0024	0.0005	0.0157	0.0027	0.0083	0.0057
CTP4	0.0389	0.0157	0.0174	0.0080	0.0143	0.0104
CTP5	0.0276	0.0051	0.0186	0.0025	0.0121	0.0037
CTP7	0.0301	2E-5	0.0322	0.0207	0.0183	0.0055

表 2 3 种算法所得关于 TSC 测度的统计结果

函数	TSC(A, B)		TSC(B, A)		TSC(A, C)		TSC(C, A)	
	均值	方差	均值	方差	均值	方差	均值	方差
BNH	0.20	0.03	0.02	0.008	0.04	0.01	0.03	0.007
OSY	0.79	0.14	0.01	0.02	0.91	0.04	0.01	0.03
CON	0.25	0.05	0.04	0.02	0.07	0.02	0.08	0.03
TNK	0.29	0.05	0.12	0.03	0.26	0.06	0.14	0.03
CTP2	1	0	0	0	1	0	0	0
CTP4	1	0	0	0	1	0	0	0
CTP5	1	0	0	0	1	0	0	0
CTP7	1	0	0	0	1	0	0	0

对于函数 BNH 和 CON,由表 1 可以看出,本文所提算法得到了最好的平均 SP 值;由表 2 可知,本文算法所

得结果以较大的比例支配 CMOPSO 所得结果,而 Co-MOPSO 和 CMOPSO 所得结果支配本文算法的最高比例仅为 0.08.

对于函数 OSY,本文算法不仅得到了最好的平均 SP 值,而且所得结果均以较高的比例支配 CMOPSO 和 Co-MOPSO 的结果.表 2 中 TSC(A, B)和 TSC(A, C)值分别远大于 TSC(B, A)值和 TSC(C, A)值.

对于函数 TNK, Co-MOPSO 得到了最好的平均 SP 值;相比之下,本文算法得到了相对较差但依然较小的平均 SP 值.然而,由于 TSC(A, B)和 TSC(A, C)明显高于 TSC(B, A)和 TSC(C, A),本文算法所得结果的收敛性好于其它两种比较算法.因此,对于函数 TNK, BB-MOPSO 可以得到与 CMOPSO 和 Co-MOPSO 相竞争的优化结果.

考虑 CTP 系列函数.由表 1 可以看出,对于函数 CTP2,本文算法得到了最好的平均 SP 值.而对于其它 3 个函数,本文算法所得 SP 值却远差于 Co-MOPSO.分析出现这种现象的原因,主要是上述 3 个测试函数具有相对不规则的真正 Pareto 前端.例如,CTP4 的真正 Pareto 前端只有少数几个离散点.其中任意一个最优解的缺失,都将显著影响所得结果的分布性.进一步,由图 1 可以看出,对于测试函数 CTP4、CTP5 和 CTP7, Co-MOPSO 和 CMOPSO 只得到了一个分布比较规则的局部 Pareto 前端.此外,由表 2 可以看出,对于上述 4 个函数,皆有 $TSC(A, B) = TSC(A, C) = 1$ 和 $TSC(B, A) = TSC(C, A) = 0$.这表明,对于这些函数,本文算法得到了完全支配 Co-MOPSO 和 CMOPSO 结果的非劣解集.

图 1 展示了 3 种算法优化部分函数时,所得一组 Pareto 前端.其中,实曲线为无约束条件下 CTP 系列函数的真正 Pareto 前端.可以看出,对于函数 OSY,除所提算法外,CMOPSO 和 Co-MOPSO 未发现全部 4 个 Pareto 子前端;对于 CTP 系列函数,经过 500 次迭代,本文算法得到了一个接近真正 Pareto 前端的最优解集,而其它两种算法得到了一个局部 Pareto 前端.

5 结论

为解决约束多目标优化问题,本文提出了一种少控制参数的简洁多目标微粒群优化算法.通过合理利用非可行解和可行解,显著地提高了算法的搜索性能.与两种已知的多目标微粒群优化算法进行比较,实验表明,对于所选测试函数本文所提算法皆能得到具有高竞争力的 Pareto 最优解集.

然而,必须承认,由于非可行储备集更新时带来的额外花费, BB-MOPSO 所需运行时间要略高于其它两种算法.在保证算法性能不受影响的基础上,如何降低储备集更新操作的计算复杂度、提高算法的计算效率,是我们将来的研究内容.

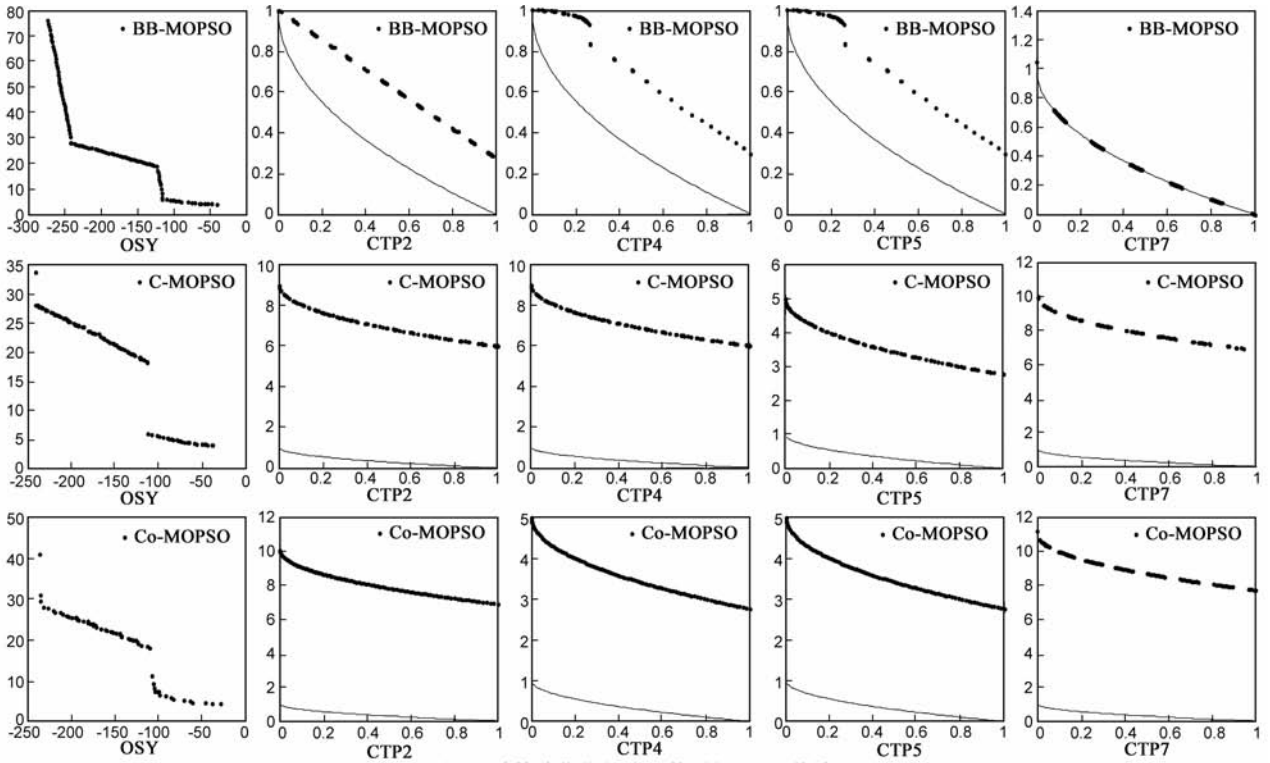


图1 3种算法优化部分函数所得Pareto前端

参考文献

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[A]. Proceedings of IEEE International Conference on Neural Networks [C]. NJ: IEEE Piscataway, 1995. 1942 – 1948.
- [2] Coello Coello C A, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256 – 279.
- [3] Tsai S J, Sun T Y, et al. An improved multi-objective particle swarm optimizer for multi-objective problems[J]. Expert Systems with Applications, 2010, 37(8): 5872 – 5886.
- [4] 彭志平, 陈珂. 一种消解协商僵局的多目标粒子群优化算法[J]. 电子学报, 2007, 35(8): 1452 – 1457.
Peng Z P, Chen K. A multi-objective particle swarm optimization algorithm for solving negotiation deadlock[J]. Acta Electronica Sinica, 2007, 35(8): 1452 – 1457. (in Chinese)
- [5] 陈民铖, 张聪誉, 罗辞勇. 自适应进化多目标粒子群优化算法[J]. 控制与决策, 2009, 24(12): 1851 – 1855, 1864.
Chen M Y, Zhang C Y, Luo C Y. Adaptive evolutionary multi-objective particle swarm optimization algorithm[J]. Control and Decision, 2009, 24(12): 1851 – 1855, 1864. (in Chinese)
- [6] Wang Y J, Yang Y P. Particle swarm optimization with preference order ranking for multi-objective optimization[J]. Information Sciences, 2009, 179(12): 1944 – 1959.
- [7] Shi Y H, Eberhart R C. A modified particle swarm optimizer [A]. Proceedings of the IEEE International Conference on Evolutionary Computation[C]. NJ: IEEE Piscataway, 1998. 63 – 79.
- [8] Leong W F. Multiobjective Particle Swarm Optimization: Integration of Dynamic Population and Multiple-swarm Concepts and Constraint Handling[D]. Stillwater: Oklahoma State University, 2008.
- [9] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.
- [10] Coello Coello C A, Van Veldhuizen D A, Lamont G B. Evolutionary Algorithms for Solving Multi-objective Problems [M]. Norwell, MA: Kluwer, 2002.

作者简介



张勇 男, 1979年9月出生于山东省莱芜市, 讲师, 博士, 主要研究方向: 群体智能和机器人协作等。

E-mail: yongzh401@126.com



巩敦卫 男, 1970年3月出生于江苏铜山, 博士, 教授, 博士生导师, 主要研究方向: 基于搜索的软件工程、智能优化与控制。

E-mail: dwgong@vip.163.com