

改进的 r 支配高维多目标粒子群优化算法

章恩泽[†], 陈庆伟

(南京理工大学 自动化学院, 江苏 南京 210094)

摘要: 高维多目标优化问题是广泛存在于实际应用中的复杂优化问题, 目前的研究方法大都限于进化算法. 本文利用粒子群优化算法求解高维多目标优化问题, 提出了一种基于 r 支配的多目标粒子群优化算法. 采用 r 支配关系进行粒子的比较与选择, 并结合粒子群优化算法收敛速度快的优势, 使得算法在目标个数增加时仍保持较强的搜索能力; 为了弥补由此造成的群体多样性的丢失, 优化非 r 支配阈值的取值策略; 此外, 引入决策空间的拥挤距离测度, 并给出新的外部存储器更新方法, 从而进一步防止算法陷入局部最优. 对多个基准测试函数的仿真结果表明所得解集在收敛性、多样性以及围绕参考点的分布性上均优于其他两种算法.

关键词: 高维多目标优化; 偏好; 粒子群优化

中图分类号: TP301 **文献标识码:** A

Improved r -dominance-based particle swarm optimization for multi-objective optimization

ZHANG En-ze[†], CHEN Qing-wei

(School of Automation, Nanjing University of Science and Technology, Nanjing Jiangsu 210094, China)

Abstract: Multi-objective optimization problems (MOPs) are complex optimization problems existing in practice, for which most of the modern research methods are focused on evolutionary algorithms. In this paper, a multi-objective particle swarm optimization algorithm based on the r -dominance is proposed for investigating the behavior of the particle swarm optimization (PSO) in MOPs. The combination of the r -dominance with the fast convergence properties of PSO maintains strong search capabilities of the algorithm when the number of objectives increases. In particular, the value of the non- r -dominance threshold is varied in an improved way in order to keep desired population diversity. Furthermore, a new updating strategy of the external repository, which incorporates the crowding distance in the variable space, is presented to get rid of the local optimum. Effectiveness of the proposed algorithm is validated by several benchmark test functions. Results indicate that the proposed algorithm outperforms two other existing algorithms in terms of convergence, diversity and distribution over the reference point.

Key words: multi-objective optimization; preference; particle swarm optimization

1 引言(Introduction)

多目标优化问题广泛存在于科学研究和工程应用中, 这类问题具有多个彼此冲突的目标, 因此不存在唯一的最优解. 研究者们对此提出了诸多理论和方法, 这些研究成果多针对2个或3个目标的优化问题, 而许多实际应用问题都涉及多于3个目标, 即高维目标. 研究表明已有的基于Pareto支配的多目标优化算法在求解高维多目标优化问题时存在如下困难^[1-3]: 随着目标个数的增加, 种群中的非支配个体的数量呈指数上升, 大大削弱了算法的选择压力, 减缓了算法的搜索进程, 且求得的大量非支配解会带来较大的决策负担.

事实上, 在实际应用中, 决策者往往只需要得到符合其偏好的一个解或若干解, 而非整个Pareto最优解集, 因此, 在优化过程中融入决策者偏好可以缩小算法的搜索区域, 提高算法在目标个数增加时的搜索能力, 同时减轻决策者的负担. 近年来, 一些研究者结合决策者偏好求解高维多目标优化问题^[4-7]. Said等^[5]采用参考点体现决策者偏好, 提出了一种新的支配关系, r 支配, 并将其应用于非支配排序遗传算法II(non-dominated sorting genetic algorithm II, NSGA-II)算法, 验证了处理高维目标时的有效性.

目前关于高维多目标优化问题的理论和方法大都

收稿日期: 2014-09-30; 录用日期: 2015-02-02.

[†]通信作者. E-mail: yzzez8986@gmail.com; Tel.: +86 15850577123.

国家自然科学基金项目(61074023), 江苏省科技支撑计划项目(BE2012175), 江苏省普通高校研究生科研创新计划项目(KYZZ.0121)资助.

Supported by National Natural Science Foundation of China (61074023), Science and Technology Support Program of Jiangsu Province (BE2012175) and Graduate Student Research Innovation Program of Jiangsu Province College (KYZZ.0121).

限于少数几种算法,如NSGA-II算法.粒子群优化(particle swarm optimization, PSO)算法^[8]是Kennedy等受到飞鸟集群活动的启发而提出的一种群体智能优化算法.相比进化算法,PSO具有易于实现和收敛速度快等优势.近年来PSO在多目标优化领域的研究取得了很大进展^[9-10],但是用于高维多目标优化问题的理论和方法还很少.Wickramasinghe等^[11]利用与参考点的距离测度进行粒子的选择与更新,引导算法向偏好区域搜索.然而,这种方法可能造成解的多样性的丢失,导致算法的早熟收敛,同时无法保证搜索到的最优解是非支配解.Carvalho等^[12]将控制支配区域方法与粒子群优化算法相结合,在一定程度上增强了算法的选择能力,但随着目标个数的持续增加,该方法仍然面临着选择能力退化的问题.Hirano等^[13]针对高维多目标优化问题,提出了一种基于粒子群优化的两步搜索方法.Britto等^[14]则通过基于理想点的外部存储器更新方法提高算法的收敛性.

本文针对高维多目标优化问题,提出一种基于r支配的多目标粒子群优化算法.结合r支配关系与粒子群优化算法的优点,在体现决策者偏好的同时缩小了搜索区域,增强了算法在目标个数增加时的搜索能力;通过改进非r支配阈值的取值方式,保证了算法在“探索”和“开发”之间的平衡;此外,引入决策空间的拥挤距离测度,并给出新的外部存储器更新策略,从而进一步保证解的多样性,防止算法陷入局部最优.仿真结果验证了所提算法在求解高维多目标优化问题时的可行性与有效性.

2 基础工作(Basic work)

2.1 高维多目标优化问题(Many-objective optimization)

以最小化问题为例,多目标优化问题的数学模型可描述如下^[9]:

$$\min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})), \quad (1)$$

$$\text{s.t.} \begin{cases} g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, P, \\ h_j(\mathbf{x}) = 0, j = 1, 2, \dots, Q, \end{cases} \quad (2)$$

其中: $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ 为决策向量, $f_k(\mathbf{x})$ ($k = 1, 2, \dots, M$) 为目标函数, $g_i(\mathbf{x})$ ($i = 1, 2, \dots, P$) 和 $h_j(\mathbf{x})$ ($j = 1, 2, \dots, Q$) 为问题约束.当目标个数 $M \geq 4$ 时,称这类多目标优化问题为高维多目标优化问题(many-objective optimization problems, MaOPs).

2.2 粒子群优化算法(Particle swarm optimization)

粒子群算法^[8]是Kennedy等在1995年提出的一种群智能优化算法,它来源于对鸟群飞行觅食行为的模拟.问题的潜在解被看做是搜索空间中的一个粒子,在搜索空间内以一定的速度飞行,并根据个体和集体的飞行经验来动态调整这个速度.以带惯性权重的基

本PSO算法为例,粒子速度和位置的更新公式如下:

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1r_1(p_{i,j}^t - x_{i,j}^t) + c_2r_2(g_j^t - x_{i,j}^t), \quad (3)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, \quad (4)$$

其中: w 为惯性权重; c_1 和 c_2 为学习因子; r_1 和 r_2 为区间 $[0, 1]$ 上服从均匀分布的随机数; $\mathbf{x}_i^t = (x_{i,1}^t, x_{i,2}^t, \dots, x_{i,n}^t)$ 和 $\mathbf{v}_i^t = (v_{i,1}^t, v_{i,2}^t, \dots, v_{i,n}^t)$ 为第 t 次迭代时粒子 i 在 n 维空间中的位置和速度; $\mathbf{p}_i^t = (p_{i,1}^t, p_{i,2}^t, \dots, p_{i,n}^t)$ 为第 t 次迭代时粒子 i 的个体最好位置; $\mathbf{g}^t = (g_1^t, g_2^t, \dots, g_n^t)$ 为第 t 次迭代时整个种群的全局最好位置.

3 针对高维多目标优化问题的粒子群优化算法(Particle swarm optimization for many-objective optimizations)

针对高维多目标优化问题,本文提出一种基于r支配的多目标粒子群优化算法(multi-objective particle swarm optimization based on r-dominance, r-MOPSO).

3.1 r支配关系(r-dominance relation)

r支配关系^[5]的本质是将Pareto支配关系与参考点方法相结合,在保留Pareto支配关系产生的排序的同时选择更加靠近参考点的解.设有群体 P , 参考点 g 以及权重向量 $\mathbf{w}$, 采用带权重的欧式距离^[15](weighted Euclidean distance)衡量某个解与参考点的距离,其表达式如下:

$$\text{Dist}(x, g) = \sqrt{\sum_{i=1}^M w_i \left(\frac{f_i(x) - f_i(g)}{f_i^{\max} - f_i^{\min}} \right)^2}, \quad (5)$$

其中: $f_i^{\max}$ 和 $f_i^{\min}$ 分别为第 i 个目标函数值的上限和下限; $w_i \in [0, 1]$ 为第 i 个目标对应的权重且 $\sum_{i=1}^M w_i = 1$. 值得注意的是,在目标函数的最大、最小值已知的情况下,分别取其作为 $f_i^{\max}$ 和 $f_i^{\min}$; 在目标函数的真正最大、最小值难以获取的情况下,则取当前种群对应目标函数的最大、最小值作为 $f_i^{\max}$ 和 $f_i^{\min}$. 当满足以下条件之一时,解 x 被认为r支配解 $y(x \prec_r y)$:

1) 解 x Pareto支配解 y ;

2) 解 x 和 y Pareto互不支配且 $D(x, y, g) < -\delta$, 其中 δ 为非r支配门限且 $\delta \in [0, 1]$,

$$D(x, y, g) = \frac{\text{Dist}(x, g) - \text{Dist}(y, g)}{\text{Dist}_{\max} - \text{Dist}_{\min}}, \quad (6)$$

$$\text{Dist}_{\max} = \max_{z \in P} \text{Dist}(z, g), \quad (7)$$

$$\text{Dist}_{\min} = \min_{z \in P} \text{Dist}(z, g). \quad (8)$$

可以看出, r支配关系的主要思想是在Pareto非支配解之间建立严格的偏序关系,即通过参考点体现决策者偏好,据此区分任意两个非支配解.与Pareto支配关系相比, r支配关系不仅增强了算法的选择压力,而且在选择过程中融入了决策者偏好.同时, r支配关系

并未违背Pareto支配关系产生的排序,即在保留Pareto支配关系的同时利用决策者偏好信息指导算法的搜索方向,使得算法最终只求期望区域内的解。

3.2 非r支配阈值取值(Management of the non-r-dominance threshold)

“探索”和“开发”是搜索和优化的核心问题之一,二者的平衡对于算法的优化结果有重要的影响^[16]。本文改进了非r支配阈值的取值方式,将其作为一种控制“探索”和“开发”能力的机制。

由r支配定义可知,当 $\delta = 1$ 时,r支配关系等同于Pareto支配关系;随着 δ 值减小,算法的选择压力增大,同时意味着所得解更能够体现决策者偏好,但较高的选择压力往往会导致解的多样性的丧失。文献[5]按照线性减小的方式调整 δ 的取值以保持群体的多样性,然而值的线性减小会导致算法探索能力的急剧下降,从而导致算法陷入局部最优。为了避免此问题,本文对 δ 的取值进行改进,采用一种非线性的取值方式,如图1所示。

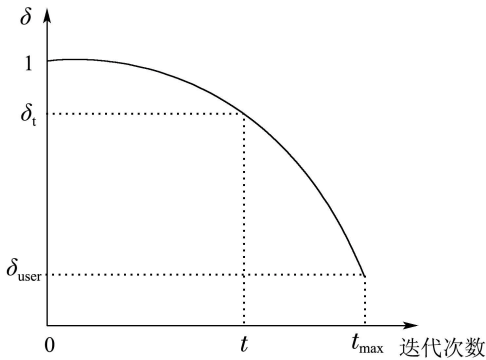


图1 非r支配阈值的取值方式

Fig. 1 Management of the non-r-dominance threshold

其表达式为

$$\delta_t = 1 + (\delta_{\text{user}} - 1) \times (t/t_{\text{max}})^{\frac{1}{c}}, \quad (9)$$

其中: t_{max} 和 t 分别为最大迭代次数和当前迭代次数, δ_{user} 和 δ_t 分别为决策者给出的非r支配阈值以及当前的非r支配阈值, c 为阈值取值参数,依照经验取0.3。在优化初始阶段,即 $t = 0$, δ 值为1,根据Pareto支配关系进行个体的比较与选择;随后 δ 值缓慢减小,增强了早期算法的探索能力,保证了搜索过程中群体的多样性;最后, δ 值减小到决策者设定的值 δ_{user} 。采用上述方式调整 δ 的取值,保证了算法在群体多样性与决策者偏好之间的平衡,防止算法陷入局部最优。

3.3 外部存储器更新(Update of the external repository)

外部存储器主要用来保存算法搜索到的非支配解^[17]。利用拥挤距离(crowding distance)^[18]进行外部存储器的更新以及领导粒子的选择有助于维护解的多样性,保证所得最优解分布的均匀性。现有算法仅

考虑了目标空间的拥挤距离,设有非支配解集 D ,则非支配个体 $d \in D$ 的拥挤距离计算公式如下:

$$I(d) = \sum_{m=1}^M \frac{I_m(d)}{f_m^{\text{max}} - f_m^{\text{min}}}, \quad (10)$$

其中: f_m^{max} 和 f_m^{min} 分别为第 m 个目标函数的最大值和最小值;

$$I_m(d) = \begin{cases} \infty, & f_m(d) = \min\{f_m(d') | d' \in D\} \text{ 或} \\ & f_m(d) = \max\{f_m(d') | d' \in D\}; \\ \min\{f_m(d') - f_m(d'') | d', d'' \in D : \\ & f_m(d'') < f_m(d) < f_m(d')\}, \\ \text{其他.} \end{cases} \quad (11)$$

然而当目标个数增多时,以上拥挤距离不能够很好地反应粒子的密集程度。因此,本文算法同时考虑了决策空间的拥挤距离。

r-MOPSO采取的外部存储器更新方法为:根据r支配关系,存入当前种群搜索到的非支配解,删除受支配解。若外部存储器Rep中最优解个数超过其容量,进行裁剪操作,步骤如下:

Step 1 根据式(10)计算Rep中每个粒子的目标空间拥挤距离,并进行降序排列;

Step 2 对于排在后10%的 N_s 个粒子,计算其中每个粒子的变量拥挤距离。设 $\mathbf{x}_i, \mathbf{x}_j$ 分别为第 i, j 个粒子,则 $\mathbf{x}_i$ 的变量拥挤距离为($\|\cdot\|$ 表示欧氏距离):

$$d\mathbf{x}_i = \min_{1 \leq j \leq N_s, j \neq i} \{\|\mathbf{x}_i - \mathbf{x}_j\|\}; \quad (12)$$

Step 3 对以上变量拥挤距离值进行归一化。记归一化后的变量拥挤距离为 $d\mathbf{x}_i^{\text{norm}}$,其表达式如下:

$$d\mathbf{x}_i^{\text{norm}} = \frac{d\mathbf{x}_i - d\mathbf{x}_i^{\text{min}}}{d\mathbf{x}_i^{\text{max}} - d\mathbf{x}_i^{\text{min}}}, \quad (13)$$

其中 $d\mathbf{x}_i^{\text{max}}$ 和 $d\mathbf{x}_i^{\text{min}}$ 分别为 $d\mathbf{x}_i$ 的最大值和最小值。

Step 4 若 $d\mathbf{x}_i^{\text{norm}} < \delta_d$,从Rep中移除 $\mathbf{x}_i$,否则保留此解,其中 $\delta_d \in [0, 1]$ 为决策空间拥挤距离阈值,依据经验取0.3;

Step 5 判断Rep中最优解个数是否超过其容量,若是则转到Step 1,否则结束裁剪操作。

3.4 算法描述(Description of the algorithm)

基于以上研究, r-MOPSO算法的具体描述如下:

Step 1 设置参数。设定种群规模、外部存储器容量、最大迭代次数,以及PSO更新公式中的常量,如惯性常量 w 、学习因子 c_1 和 c_2 等;

Step 2 初始化种群。初始化粒子的位置、速度以及个体最好位置。计算粒子适应值,将其中非r支配的解放入外部存储器Rep中;

Step 3 选择领导粒子。计算Rep中每个粒子的拥挤距离值并进行降序排列,从前10%的粒子中随机选择领导粒子;

Step 4 更新粒子的位置和速度. 根据式(3)和式(4)更新粒子的位置和速度, 并进行变异操作^[17];

Step 5 更新粒子的个体最好位置. 根据 r 支配关系, 比较当前粒子和个体最好位置 $pbest$, 若当前粒子支配 $pbest$, 则当前粒子位置为新的 $pbest$; 若两者互不支配, 随机选择其一作为新的 $pbest$;

Step 6 采用第3.3节方法更新外部存储器;

Step 7 判断终止条件. 判断是否达到最大迭代次数, 若达到, 则停止搜索; 否则转到Step 3.

4 实验研究(Experimental study)

本节通过仿真验证所提 r -MOPSO算法的有效性. 首先分析参数 δ 和 w 取值对算法优化结果的影响; 其次验证所提非线性阈值的选取能够提高算法的多样性; 最后将 r -MOPSO与两种已有算法 r -NSGA-II^[5], MDEPSO^[11]进行性能比较. 所有实验均在同一台计算机上采用MATLAB R2013a编程实现.

4.1 参数取值分析(Effect of varying parameters)

选用2目标FON测试函数研究阈值 δ 以及权重向量取值对所得解分布的影响. FON函数具有凹Pareto前沿. 参数设置如下: 粒子群规模为50, 外部存储器容量为50, 最大迭代次数为50, 惯性权重为0.6, 学习因子皆为2.

1) 阈值 δ 取值分析.

图2展示了不同阈值 δ 取值下的优化结果. 参考点选为(0.6, 0.5), 在图中用五角星表示, 权重向量为(0.5, 0.5). 当 $\delta=0$ 时, 群体中的所有粒子收敛到了离参考点最近的Pareto最优前; 为了更清楚地显示不同 δ 值下的结果, 将 δ 取其他值的结果偏离真实的Pareto前沿画出; 从图中可以看出, 所得解的分布范围随着 δ 的增大而变广; 当 $\delta=1$ 时, 算法获得了完整的Pareto最优前沿.

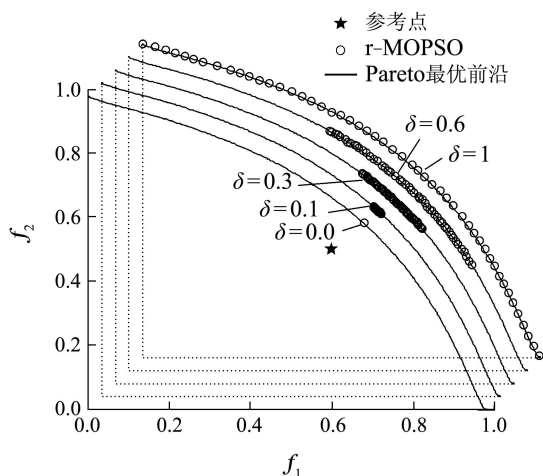


图2 不同 δ 值下FON函数的优化结果

Fig. 2 Effect of varying δ value on FON test function

因此决策者可通过 δ 值的选取控制所得Pareto最优前沿的分布广度: 如果决策者希望得到一组期望区域

附近的分布范围较广的解, 则可以选择较大的 δ 值; 如果希望得到一组分布范围较小的解, 则选择较小的 δ 值.

2) 权重取值分析.

图3展示了权重向量分别为(0.25, 0.75), (0.5, 0.5)和(0.75, 0.25)时的优化结果. 参考点选为(0.6, 0.5), 在图中用五角星表示, δ 为0.4.

从图中可以看出, 权重向量选取的不同会导致所得解分布区域的不同, 例如当权重向量为(0.5, 0.5)时, 在两个目标之间不存在偏好; 当权重向量为(0.25, 0.75)时, 目标 f_2 的重要性高于 f_1 , 所得解对于目标 f_2 的优化程度高于 f_1 ; 而当权重向量为(0.75, 0.25)时, 目标 f_1 的重要性高于 f_2 , 所得解在目标 f_1 上获得了更好的优化效果. 因此, 可以通过权重向量值的设定体现决策者的偏好或决策者所认为的目标的重要性.

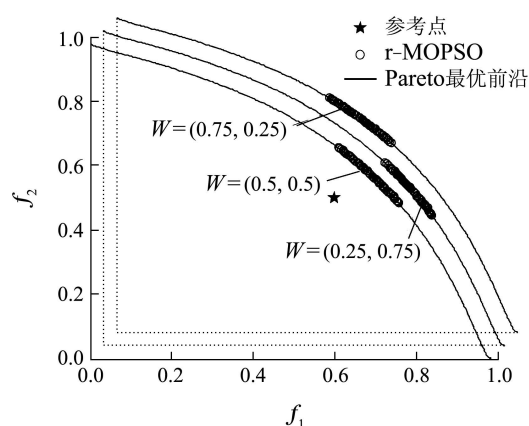


图3 不同权重值下FON函数的优化结果

Fig. 3 Effect of varying weights on FON test function

4.2 δ 取值对优化结果的影响(Effect of managing the δ value)

选用2目标ZDT3测试函数研究阈值取值对优化结果的影响. ZDT3的Pareto前沿包含5个非连续的凸曲线, 常被用来测试算法所得Pareto前沿的分布性. 参数设置如下: 粒子群规模为50, 外部存储器容量为30, 最大迭代次数为100, 惯性权重为0.6, 学习因子皆为2.

根据 δ 取值方式的不同, r -MOPSO算法的两种形式分别记为 r -MOPSO_l和 r -MOPSO_n, 前者 δ 采用线性减小的取值方式, 后者采用如图1所示非线性减小方式. 参考点选为(0.4, 0), δ 为0.2.

图4展示了上述两种形式算法获得的优化结果. 从图4中可以看出, r -MOPSO_n获得了逼近真实Pareto前沿的最优解集, 而 r -MOPSO_l陷入了局部最优. 正如第3.2节所讨论的, 引导算法向搜索空间的某个区域搜索可能会导致解的多样性的丢失, 从而造成早熟收敛, 而本文算法对 δ 取值方式的改进有效地补救了这一点.

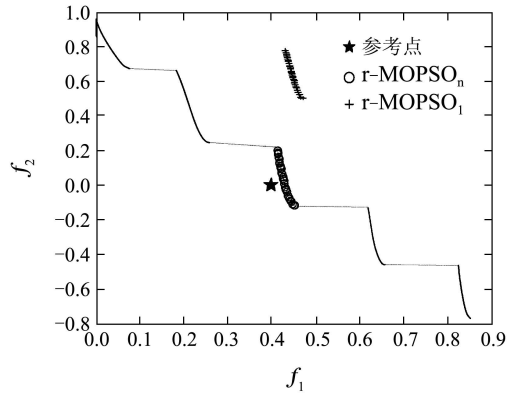
图4 δ 取值方式对ZDT3函数优化结果的影响Fig. 4 The positive effect of managing δ on ZDT3

表1 比较算法的参数设置

Table 1 Parameter configurations of the three algorithms for comparison

	种群规模	外部存储器容量	最大迭代次数	其他参数
	ZDT1: 50	ZDT1: 20	ZDT1: 100	
r-MOPSO	3目标DTLZ2: 50	3目标DTLZ2: 20	3目标DTLZ2: 100	惯性权重为0.6, 学习因子皆为2, 变异参数为0.5
	5目标DTLZ2: 100	5目标DTLZ2: 30	5目标DTLZ2: 200	
	10目标DTLZ2: 200	10目标DTLZ2: 50	10目标DTLZ2: 300	
	ZDT1: 50		ZDT1: 100	
r-NSGA-II	3目标DTLZ2: 50	—	3目标DTLZ2: 100	关于模拟二进制交叉(simulated binary crossover, SBX)的分布系数为10, 关于Polynomial变异的分布系数为20, 交叉概率为0.9, 变异概率为1/n
	5目标DTLZ2: 100		5目标DTLZ2: 200	
	10目标DTLZ2: 200		10目标DTLZ2: 300	
MDEPSO	—	—	—	收缩因子为0.7298, 交叉比率为0.2, 比例因子为0.4, 分布参数为0.1

4.3.1 性能测度(Performance metrics)

采用如下3个性能测度比较上述3种算法的性能:

1) C_r 测度. 拓广C测度^[19], 以评价算法的收敛性, 其定义如下:

$$C_r(A, B) = \frac{|\{b \in B | \exists a \in A, a \prec_r b \vee a = b\}|}{|B|}. \quad (14)$$

$C_r(A, B) = 1$ 表示集合B中的解完全被集合A中的解支配; $C_r(A, B) = 0$ 表示集合B中没有解被集合A中的解支配. $C_r(A, B)$ 越大, 表示集合B的受支配程度越大; 反之则越小.

2) SP 测度^[20]. 用来评价解集的多样性. SP 值越小, 非支配解分布越均匀; $SP = 0$ 表示解完全均匀地分布在目标空间.

3) 离参考点的距离测度^[21]. 用来量度解集围绕参考点的分布. 对于不同的算法所得的解集, 计算其中每个解与参考点的欧氏距离, 将最大值与最小值之差划分为10个区间; 分别计算每种算法对应每

4.3 与其他算法的性能比较 (Comparison with other algorithms)

将r-MOPSO与r-NSGA-II^[5]以及基于距离测度的多目标粒子群优化算法MDEPSO^[11]比较, 进一步验证其有效性.

MDEPSO算法利用基于决策者偏好的距离测度求解高维多目标优化问题, 粒子根据与参考点的距离更新其位置和速度.

采用3种算法分别优化2目标ZDT1函数、3目标DTLZ2函数、5目标DTLZ2函数和10目标DTLZ2函数. 3种算法的参数设置如表1所示.

个区间的解的个数占对应解集的比例; 对应较小区间值的解的个数越多, 对应算法所得解集越接近参考点.

4.3.2 比较结果与分析 (Comparison results and analysis)

图5和图6分别展示了3种算法优化3目标和10目标DTLZ2时所得的Pareto最优前沿, 表2和表3分别给出了 C_r 测度和 SP 测度的值, 而图7和图8展示了优化3目标和10目标DTLZ2时所得Pareto最优解离参考点的距离测度.

可以看出, 相对于其他两种算法, r-MOPSO在目标个数较少(例如为2)时的优势不明显, 但当目标个数增多(例如为10)时, r-MOPSO算法性能的优越性得以体现. 以10目标DTLZ2函数为例, 具体分析如下:

1) 相对于r-NSGA-II算法, 两种算法皆能获得接近真实Pareto最优前沿的解, 但r-MOPSO算法所得结果的收敛性和多样性均优于r-NSGA-II算法; 同

时,实验中观察到,当阈值取值相同时,r-MOPSO算法能以较快的速度向参考点附近的范围搜索。

2) 目标函数个数增多时,MDEPSO算法的收敛性以及解的多样性均明显下降。此外,从图6可以看出,MDEPSO所得解出现了聚集现象,可能的原因是MDEPSO算法仅将与参考点的距离作为评价粒子的适应度函数,因而搜索方向集中在参考点附近的区域,导致粒子飞向拥挤的区域。

3) 相对于其他两种算法,r-MOPSO获得了更靠近参考点的最优解集。目标个数为3时,如图7所示,r-MOPSO算法超过70%的解落在了离参考点距离前10%的区间内。目标个数为10时,如图8所示,r-MOPSO超过90%的解都落在了离参考点距离前10%的范围内。

4) 为了进一步证实上述结论,分别计算每种算法所得解集的 $\sum_{i=1}^{10} f_i^2$ 值。对于r-MOPSO算法,该值在

[1.002, 1.008]内,说明所得解十分逼近真实的Pareto前沿;对于r-NSGA-II算法,该值在[1.129, 1.384]内,说明其所得解较接近真实的Pareto前沿;而对于MDEPSO算法,该值的变化范围为[11.6407, 15.8662],显然该算法未能收敛到全局Pareto最优前沿。这是因为,一方面MDEPSO算法仅仅利用与参考点的距离进行粒子的选择与更新,从而很多受支配粒子被保留在了群体中,一定程度上阻碍了算法的寻优过程。另一方面,在优化的初始阶段引导算法向搜索空间的某个区域搜索降低了群体的多样性,从而导致算法陷入局部最优;相反地,r-MOPSO算法没有在搜索初期限制搜索区域,而是通过调整阈值 δ 的取值引导算法逐步向参考点搜索,从而维护了解的多样性。同时,基于拥挤距离测度的外部存储器更新以及领导粒子选择策略进一步保证了群体多样性。

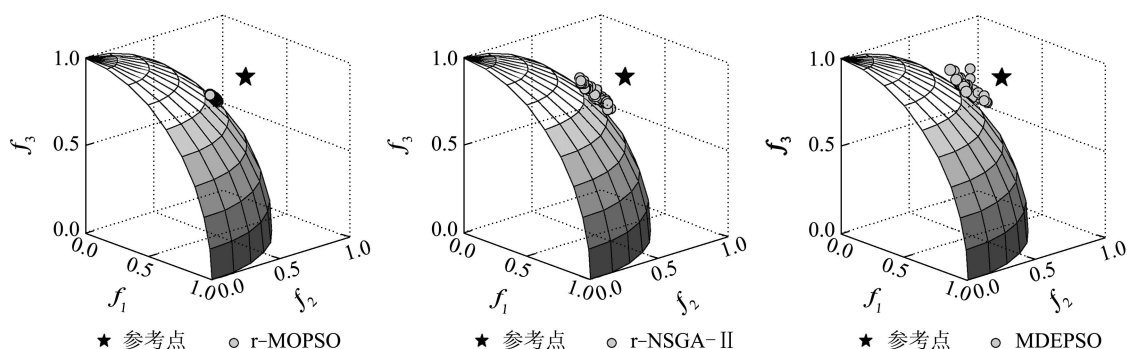


图5 3种算法优化3目标DTLZ2所得解

Fig. 5 Solutions obtained by three algorithms on 3-objective DTLZ2

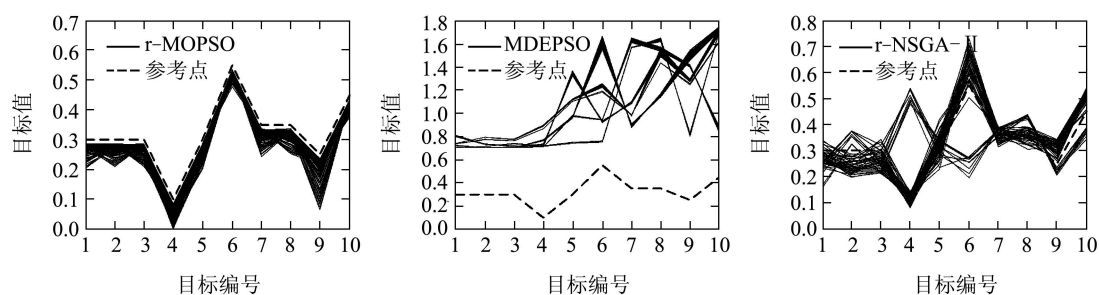


图6 3种算法优化10目标DTLZ2所得解

Fig. 6 Solutions obtained by three algorithms on 10-objective DTLZ2

表2 3种算法优化不同函数所得解的 C_r 测度值

Table 2 C_r indicator values of solutions obtained by three algorithms

测试函数	目标个数	$C_r(\text{r-MOPSO, r-NSGA-II}) / C_r(\text{r-NSGA-II, r-MOPSO})$	$C_r(\text{r-MOPSO, MDEPSO}) / C_r(\text{MDEPSO, r-MOPSO})$
ZDT1	2	0.2813/0.2078	0.2248/0.1874
	3	0.2136/0.1127	0.2297/0.1030
DTLZ2	5	0.4512/0.1045	0.7501/0.0065
	10	0.6737/0.0625	1/0

表3 3种算法优化不同函数所得解的SP测度值

Table 3 SP-indicator values of solutions obtained by three algorithms

测试函数	目标个数	SP(r-MOPSO)	SP(r-NSGA-II)	SP(MDEPSO)
ZDT1	2	0.0038	0.0107	0.0069
	3	0.0911	0.1985	0.3217
DTLZ2	5	0.1033	0.4454	0.7047
	10	0.1470	0.2372	1.0023

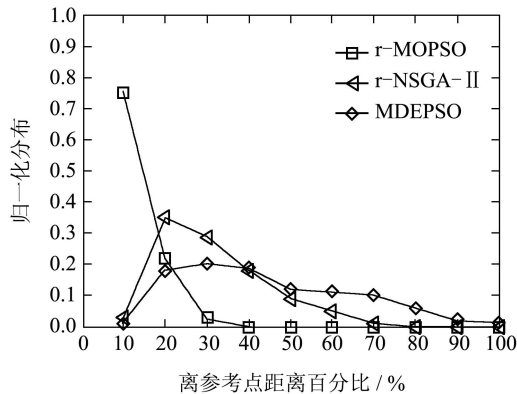


图7 3种算法优化3目标DTLZ2所得解围绕参考点的分布

Fig. 7 Distribution over the reference point for three algorithms on tri-objective DTLZ2

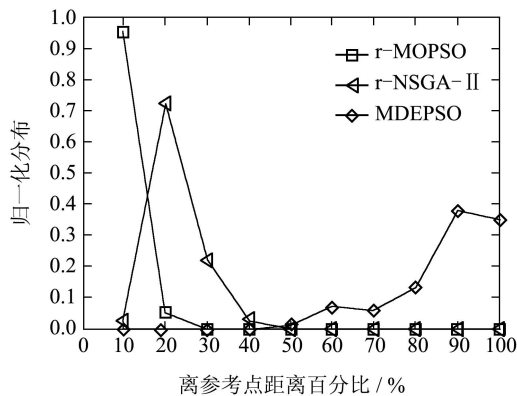


图8 3种算法优化10目标DTLZ2所得解围绕参考点的分布

Fig. 8 Distribution over the reference point for three algorithms on 10-objective DTLZ2

5 结论(Conclusions)

针对高维多目标优化问题, 本文提出了一种基于r支配的多目标粒子群优化算法r-MOPSO. r支配关系可以严格区分任意两个解, 采用其进行个体的比较与排序在很大程度上增加了算法的选择压力, 保证了算法的搜索能力不受目标个数增加的影响, 同时引导算法向决策者偏好的区域搜索. 为了弥补由此可能导致的群体多样性的丧失, 避免算法陷入局部最优, 采用非线性减小的非r支配阈值的取值方式; 在进行外部存储器更新以及领导粒子选择时,

不仅考虑目标空间的拥挤距离, 同时利用决策空间的拥挤距离. 选择多个标准测试函数作为测试对象, 并与r-NSGA-II以及MDEPSO算法进行比较, 结果表明所提算法在求解高维多目标问题上更为有效. 进一步的研究工作包括探究决策空间和目标空间拥挤距离之间可能存在的相关性, 以及设计动态种群规模的算法以减小计算成本.

参考文献(References):

- [1] ISHIBUCHI H, TSUKAMOTO N, NOJIMA Y. Evolutionary many-objective optimization: a short review [C] // *Proceedings of 2008 IEEE Congress on Evolutionary Computation*. Hong Kong: IEEE, 2008: 2424 – 2431.
- [2] 孔维健, 丁进良, 柴天佑. 高维多目标进化算法研究综述 [J]. *控制与决策*, 2010, 25(3): 321 – 326.
(KONG Weijian, DING Jingliang, CHAI Tianyou. Survey on large-dimensional multi-objective evolutionary algorithms [J]. *Control and Decision*, 2010, 25(3): 321 – 326.)
- [3] SCHUTZE O, LARA A, COELLO C C A. On the influence of the number of objectives on the hardness of a multiobjective optimization problem [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(4): 444 – 455.
- [4] DEB K, KUMAR A. Interactive evolutionary multi-objective optimization and decision-making using reference direction method [C] // *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. London: ACM, 2007: 781 – 788.
- [5] SAID L B, BECHIKH S, GHEDIRA K. The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making [J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(5): 801 – 818.
- [6] 巩敦卫, 季新芳. 融入偏好的区间高维多目标集合进化优化方法 [J]. *控制理论与应用*, 2013, 30(11): 1369 – 1383.
(GONG Dunwei, JI Xinfang. Optimizing interval higher-dimensional multi-objective problems using set-based evolutionary algorithms incorporated with preferences [J]. *Control Theory & Applications*, 2013, 30(11): 1369 – 1383.)
- [7] THIELE L, MIETTINEN K, KORHONEN P J, et al. A preference-based evolutionary algorithm for multi-objective optimization [J]. *Evolutionary Computation*, 2009, 17(3): 411 – 436.
- [8] KENNEDY J, EBERHART R C. Particle swarm optimization [C] // *Proceedings of the 4th IEEE International Conference on Neural Networks*. Piscataway, NJ: IEEE, 1995: 1942 – 1948.
- [9] ZHANG E Z, WU Y F, CHEN Q W. A practical approach for solving multi-objective reliability redundancy allocation problems using extended bare-bones particle swarm optimization [J]. *Reliability Engineering & System Safety*, 2014, 127: 65 – 76.

- [10] 李倩, 宫俊, 唐加福. 多目标粒子群算法在交叉培训规划中的应用 [J]. 控制理论与应用, 2013, 30(1): 17 – 22.
(LI Qian, GONG Jun, TANG Jiafu. Multi-objective particle swarm optimization algorithm for cross-training programming [J]. *Control Theory & Applications*, 2013, 30(1): 17 – 22.)
- [11] WICKRAMASINGHE U K, LI X. Using a distance metric to guide pso algorithms for many-objective optimization [C] // *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. Montreal, Quebec, Canada: ACM, 2009: 667 – 674.
- [12] CARVALHO A B, POZO A. Measuring the convergence and diversity of CDAS multi-objective particle swarm optimization algorithms: a study of many-objective problems [J]. *Neurocomputing*, 2012, 75(1): 43 – 51.
- [13] HIRANO H, YOSHIKAWA T. A study on two-step search based on PSO to improve convergence and diversity for many-objective optimization problems [C] // *Proceedings of 2013 IEEE Congress on Evolutionary Computation*. Cancun, Mexico: IEEE, 2013: 1854 – 1859.
- [14] BRITTO A, POZO A. I-MOPSO: a suitable PSO algorithm for many-objective optimization [C] // *2012 Brazilian Symposium on Neural Networks*. Brazil: IEEE, 2012: 166 – 171.
- [15] DEB K, SUNDAR J, UDAYA B R N, et al. Reference point based multi-objective optimization using evolutionary algorithms [J]. *International Journal of Computational Intelligence Research*, 2006, 2(3): 273 – 286.
- [16] CHEN J, XIN B, PENG Z, et al. Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization [J]. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2009, 39(3): 680 – 691.
- [17] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 256 – 279.
- [18] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [19] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: empirical results [J]. *Evolutionary Computation*, 2000, 8(2): 173 – 195.
- [20] SCHOTT J R. *Fault tolerant design using single and multicriteria genetic algorithm optimization* [D]. Cambridge: Massachusetts Institute of Technology, 1995.
- [21] BRITTO A, POZO A. Using reference points to update the archive of MOPSO algorithms in many-objective optimization [J]. *Neurocomputing*, 2014, 127: 78 – 87.

作者简介:

章恩泽 (1989–), 女, 博士研究生, 目前研究方向为智能优化算法,

E-mail: yzzez8986@gmail.com;

陈庆伟 (1963–), 男, 教授, 博士生导师, 目前研究方向为智能优

化算法、智能控制等, E-mail: cqw1002@sina.com.cn.