

基于网格排序的多目标粒子群优化算法

李 笠 王万良 徐新黎 李伟琨
(浙江工业大学计算机科学与技术学院 杭州 310023)
(liixac@gmail.com)

Multi-Objective Particle Swarm Optimization Based on Grid Ranking

Li Li, Wang Wanliang, Xu Xinli, and Li Weikun
(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023)

Abstract In multi-objective evolutionary algorithms, the majority of researches are Pareto-based. However, the efficiency of Pareto optimality in objective space will deteriorate when there are numerous weak dominance relations. Aiming at this problem, this paper presents a framework of grid-based ranking. By integrating gird strategy, which features both convergence and distribution, with the particle swarm optimization (PSO), we propose a novel grid-based ranking multi-objective particle swarm optimization (MOPSO). Unlike the strategy of Pareto-based dominance which conducts a pairwise comparison between individuals, the grid-based ranking mechanism combines the individual dominance information in the entire solution space, and takes advantage of this information to sort. As a result, we gain the merits of the relationship between individuals in the population effectively and efficiently. By incorporating the distance between particles and approximate optimal front, we reinforce the judgement of the merits of the relationship among particles in the solution space. The experimental assessment indicates that the proposed method in this paper has relative advantages in both convergence and distribution. On this basis, we discuss the influence of grid partition on efficiency in terms of the distribution of ranks over the process of evolutionary, which verifies the efficiency of the algorithm from the other aspect.

Key words multi-objective optimization; evolutionary algorithm; particle swarm optimization (PSO); grid; ranking

摘 要 在多目标进化算法中,近年的研究倾向于基于 Pareto 支配的最优化方法.针对传统的基于 Pareto 支配在排序效率上过低的问题,提出了一种基于网格排序的框架,利用网格同时表征收敛性与分布性的特性,结合粒子群算法,提出了一种基于网格排序的多目标粒子群优化算法.与个体两两进行比较的基于 Pareto 支配的策略不同,基于网格排序的机制融合了整个解空间中个体的占优信息,并利用占优信息进行排序,从而高效地得到个体在种群中的优劣关系;结合粒子到近似最优边界的距离,进一步加强了粒子在解空间中优劣关系的判别.对比实验分析表明:所提算法不论是在收敛性还是分布性上都具有较好的优势.在此基础上,讨论了网格划分数对算法效率的影响,从另一方面验证了算法的效率.

关键词 多目标优化;进化算法;粒子群算法;网格;排序

中图法分类号 TP18

粒子群优化算法 (particle swarm optimization, PSO) 是一种模拟鸟群觅食的群智能优化算法, 由 Kennedy 等人^[1]提出, 通过追随解空间中 2 个最优值 (个体历史最优 $pBest$ 、全局最优 $gBest$) 进行寻优, 由于其参数简洁、易于实现, 在优化^[2]、调度^[3]以及一些参数整定^[4]、图形图像^[5]等方面都取得了满意的效果. 但是在研究与工程实践中, 存在多个需要同时考虑的问题, 即多目标优化问题 (multi-objective problem, MOP)^[6-7], 此类问题由于各目标间存在相互约束的关系, 通常的线性规划等常规数值优化方法难以求解. 于是, 将粒子群优化算法应用到多目标优化成为目前的一个研究热点. Coello 等人^[8]第 1 次将粒子群算法扩展到多目标, 利用 Pareto 支配的思想, 提出了 MOPSO 算法; 此后, 许多学者对此进行了研究^[9-10].

但是在将 PSO 扩展到多目标优化中, 面临着如下问题与挑战:

1) 在单目标优化中, 个体的适应值能简单而唯一地被确定; 而在多目标优化问题中, 存在多个互相制约的目标, 个体的适应值不能通过简单的比较适应值大小进行确定, 因此, 如何判断个体的优劣是多目标优化中基础且关键的问题, 直接影响粒子群算法中的 $pBest$, $gBest$ 的选取, 进而影响整个算法的收敛.

2) 单目标优化问题由于其明确的数值关系, 能得到一个唯一的最优解; 而在多目标优化中, 由于多目标问题的特性决定了其很难得到单个最优解, 往往存在一组解集, 如何获得一组更加多样化且包含关键信息的解集, 以便给决策人员提供更加完整的参考是存在的另一个挑战.

在个体优劣的判断中 (挑战 1)), 目前主流的研究思路是基于 Pareto 支配, 通过比较各个目标上的数值大小, 得到个体的支配关系, 出现了许多优秀的多目标优化算法. 比如 Deb 等人^[11]利用 non-dominated sorting 提出了 NSGA-II, Zitzler 等人^[12]利用强度 Pareto 支配的思想提出了 SPEA2. 但是基于 Pareto 的方法在遇到非凸、多峰以及解空间中存在许多弱支配关系个体时, 效率会急剧下降^[13]. 于是, 有学者提出了一些解决方法, 主要有 3 类:

1) 松散的 Pareto 支配. 通过扩大支配空间来改善 Pareto 支配的不足, 此概念由 Laumanns 等人^[14]提出; Batista 等人^[15]改进了 L-dominance 并提出了 cone L-dominance; Zou 等人^[16]提出了 L-dominance, 其实质也是属于 Pareto 支配.

2) 非 Pareto 支配的方法. Weighted sum^[17]、weighted Tchebycheff^[18]、基于分解的方法^[19]等.

3) 基于聚合函数的方法, 利用构造的聚合函数 (aggregation function) 对个体优劣进行判别, 主要有基于 Max-min 适应值法^[20]等.

在解的多样性保持方面 (挑战 2), 主要有: 小生境方法, 模拟自然界生物的“物以类聚”^[21]来维持种群的分布性; 利用个体之间相异或相似度的聚类方法^[22]、利用个体的密度信息包括聚集密度法^[23]以及信息熵^[24]等. 信息熵是衡量分布的混乱程度或分散程度的一种度量, 分布越分散, 信息熵就越大; 分布越集中, 信息熵就越小. 采用信息熵的办法可以在一定程度上维持群体的多样性. 但是从熵的定义可以看出, 此种方法是从群体整体出发, 缺乏对群体内部个体之间关系的刻画, 对于种群进化过程中的多样性与分布性的调控不是很有效. 聚集密度主要是计算种群中个体两两之间的欧氏距离, 利用个体的密度信息来维持种群的多样性.

基于网格的方法在多目标优化算法当中有较好的应用. Leong 等人^[25]通过构建网格来维持解的多样性. Knowles 等人^[26]采用自适应网格来存放得到的非支配向量. Yang 等人^[27]利用网格技术应用于进化算法求解高维多目标优化问题. 但是, 这些网格方法本质都是以解空间中的个体为对象进行计算, 不一定能保证算法的收敛, 从而影响算法的效果^[28].

本文利用网格同时表征收敛性与分布性的特性, 通过坐标映射, 利用对个体在网格坐标系中占优情况进行排序, 结合个体与近似最优边界的欧氏距离, 提出了一种新的 $gBest$, $pBest$ 选取策略, 结合粒子群优化算法, 提出了一种基于网格排序的多目标粒子优化算法.

1 相关工作

1.1 多目标优化问题

一个典型的多目标优化问题包含多个目标同时优化, 不失一般性, 对于一个最小化的多目标优化问题有表达式:

$$\begin{aligned} \min_{\mathbf{X} \in \Omega} \mathbf{F}(\mathbf{X}) &= (f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_M(\mathbf{X}))^T, \\ \mathbf{X} &= (X_1, X_2, \dots, X_n)^T, \\ \mathbf{F}: \Omega &\rightarrow \mathbb{R}^M, \\ f_i: \Omega &\rightarrow \mathbb{R}^n. \end{aligned} \quad (1)$$

其中, $i=1, 2, \dots, M$, $\mathbf{F}(\mathbf{X})$ 是目标函数; $\mathbf{X}$ 为决策向量, Ω 为可行解集; $\mathbb{R}^n$ 为决策空间, $\mathbb{R}^M$ 为目标空间. $f: \mathbb{R}^n \rightarrow \mathbb{R}^M$ 为目标映射函数.

1.2 粒子群优化算法

在粒子群算法中,每个粒子代表1个潜在的解,粒子在解空间中根据个体历史最优粒子与全局最优粒子的飞行方向通过更新迭代逐渐靠近最优解.

将粒子群优化算法应用于多目标优化问题具有一定优势:

1) 与遗传算法等进化算法类似,粒子群算法也是通过规模较大的个体并行地在解空间上对非劣解进行搜索;

2) 由于粒子群是跟随自身最优粒子与全局最优粒子的方向,具有一定“记忆”功能,因此,具有较高的运行速度与计算效率.一类多目标粒子优化算法如图1所示:

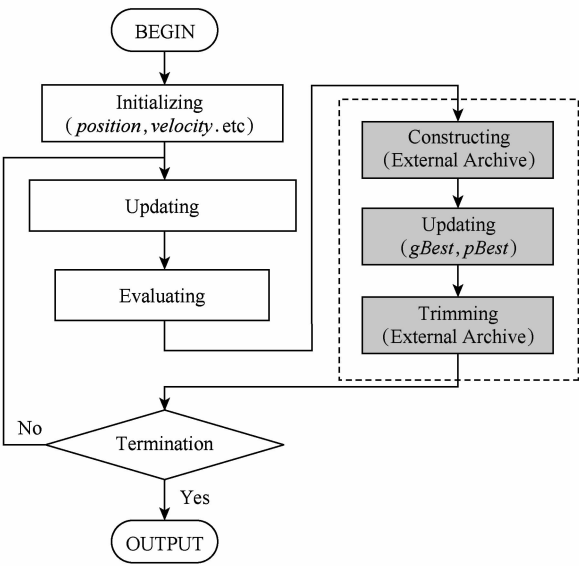


Fig. 1 A framework of MOPSO

图1 一类多目标粒子群优化算法框架

从图1中可以看出,多目标粒子群算法相对于单目标的粒子群算法来说,多出了外部档案集的构造与维护,另外在 $gBest$ 与 $pBest$ 的选取上面具有完全不同的机制.由于粒子群固有的跟踪特性,如果直接套用原来单目标粒子群算法的最优粒子选择机制,将会使粒子跟踪非支配解,从而使得算法收敛于非劣解的局部范围而难以到达真实最优边界.因此在将粒子群算法应用于多目标优化问题时,必须要解决全局最优位置与自身最优位置的选取问题.另一方面,对于外部档案集来说,拥挤、无序的非劣解使得算法的输出极不均匀,影响了实际应用的效果.基于以上分析,本文将从最优粒子选取机制、外部档案集的更新维护等方面展开.

2 基于网格排序的 MOPSO 算法

2.1 网格坐标与网格排序

2.1.1 网格坐标系

在多目标优化问题中,空间中的个体经历了从决策空间 $\Omega \subseteq \mathbb{R}^n$ 到目标空间 $\Pi \subseteq \mathbb{R}^r$ 的转换,如一个多目标优化问题的解 $X^* \in P^*$ (决策空间) 或者 $Y^* = \min f(X^*)$ (目标空间).图2展示了一个典型的空

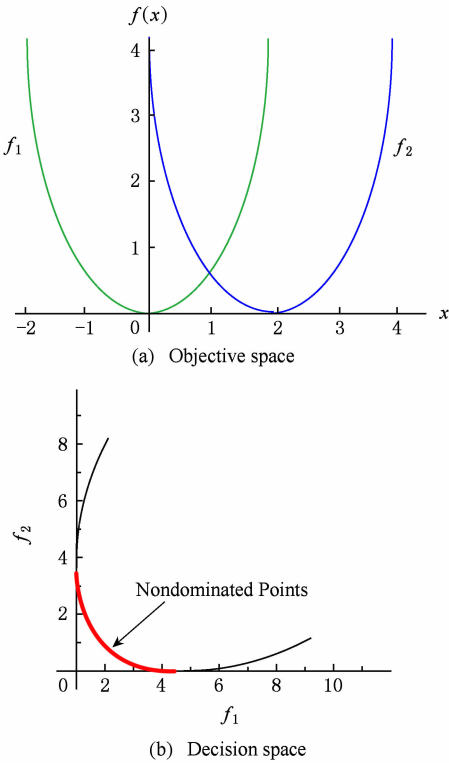


Fig. 2 A typical space-shifting in MOP

图2 多目标优化中一个典型的空

间转换.在决策空间中,利用各种方法(占优、支配),通过更新迭代求解.不管是在目标空间还是决策空间,都是基于笛卡儿坐标系(Cartesian coordinate system).本文利用网格的方法,将决策空间进行划分,同时将多目标优化问题的空间从笛卡儿坐标系映射到网格坐标系,将个体的数值坐标映射为网格坐标.下面对此进行描述.

定义1. 网格坐标的上下边界.设在决策空间中,第 m 个目标上函数的最小值、最大值分别为 $\min f_m(\mathbf{X}), \max f_m(\mathbf{X})$, 则在第 m 个目标上网格上下边界 u_m, l_m 为

$$u_m = \max f_m(\mathbf{X}) + \frac{\max f_m(\mathbf{X}) - \min f_m(\mathbf{X})}{2 \times c}, \quad (2)$$

$$l_m = \min f_m(\mathbf{X}) - \frac{\max f_m(\mathbf{X}) - \min f_m(\mathbf{X})}{2 \times c}, \quad (3)$$

其中, c 为网格坐标的划分数, 这里根据种群大小来决定取值.

利用定义 1 的上下边界, 结合个体在决策空间中的坐标可得到个体在网格坐标系中的坐标.

定义 2. 网格坐标. 设 $f_m(\mathbf{X})$ 为个体 X_i 在第 m

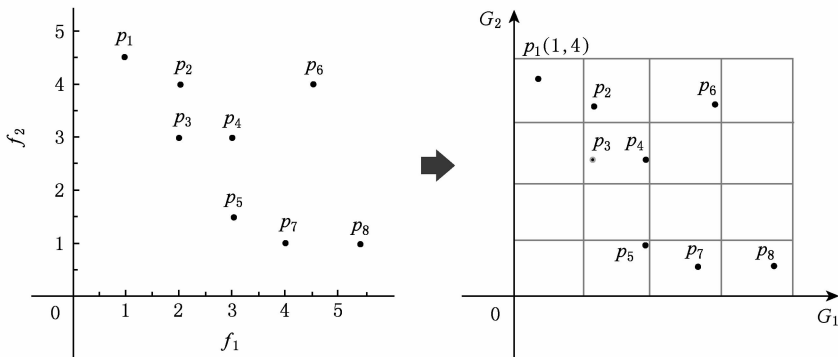


Fig. 3 A case of mapping from decision space to grid space

图 3 决策空间到网格空间的映射实例

从图 3 可以看出, 与笛卡儿坐标系类似, 网格坐标系可以反映个体的分布情况; 另一方面, 在网格坐标系中个体的密度信息同时被反映出来. 这是将坐标系转为网格的重要因素.

2.1.2 网格排序

如引言所述, 多目标优化问题的一个重点与难点是解空间中个体优劣的判断, 由于进化算法的寻优是通过各个体的更新迭代进行, 个体的优劣直接影响算法的效果. 为此, 本文借鉴 Pareto 支配的概念, 在网格坐标系中, 通过构造整体的累加函数对个体的支配关系进行排序, 得到整个解空间中个体的优劣关系. 先介绍 Pareto 支配的定义.

定义 3. Pareto 支配. 对于任意 2 个个体 $\mathbf{x}, \mathbf{y} \in f$ 以及对应的目标向量 $\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{y}) \in \mathbb{R}^M$, $\mathbf{x}$ 支配 $\mathbf{y}$ ($\mathbf{x} < \mathbf{y}$) 当且仅当 $\forall i \in \{1, 2, \dots, M\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ 且 $\exists j \in \{1, 2, \dots, M\}, f_j(\mathbf{x}) < f_j(\mathbf{y})$.

从定义 3 可以看出, Pareto 支配是通过判断个体在各目标上函数值大小进而判断支配性. 受此启发, 本文通过对解空间中个体优于其他个体的情况进行累加, 得到个体的全局网格排序.

定义 4. 全局网格排序 (global grid ranking, GGR). 定义个体的全局网格排序为个体在网格坐标下, 各目标上个体优于解空间内其他个体次数的和为

$$GGR(X_i) \triangleq \sum_{\substack{m=1 \\ i, j \in Pop, i \neq j}}^M find(G_m(X_i) < G_m(X_j)), \quad (5)$$

个目标上的函数值, 对应的网格坐标为

$$G_m(x) = \left\lceil \frac{f_m(\mathbf{X}) - l_m}{d_m} \right\rceil, \quad (4)$$

其中, $d_m = \frac{u_m - l_m}{c}$ 为第 m 个目标上网格的大小.

$\lceil \cdot \rceil$ 为向上取整函数. 图 3 描述了从决策空间到网格空间的映射.

其中, $GGR(X_i)$ 为个体 X_i 的全局网格排序, M 为目标个数, $G_m(X_i)$ 为个体 X_i 在目标 m 上的网格坐标, $find(\cdot)$ 函数为找出满足条件 $(\cdot)$ 的个数. 从式 (5) 可以看出, 个体的全局网格排序值越大, 支配的个体越多. 例如在图 3 中, 个体 p_5 的 GGR 值为 8, 支配了其他剩余个体.

在排序的过程中会遇到全局网格排序值相同的情况, 我们通过接下来的策略进行解决.

2.2 分布性保持策略

多目标优化问题的目标之一是希望得到的解尽可能多且均匀分布于最优边界, 解的分布性也是衡量算法优劣的一个方面^[29].

如前所述, 网格坐标不仅能反映个体在解空间中的分布情况, 还能通过网格内个体的数量来判断个体的聚集情况; 从优化角度来说, 既能反映算法的收敛性, 又能反映解分布的多样性. 在网格坐标系中, 如果网格中会遇到个体在同一个网格中, 说明这些个体密度值较大, 分布性较差. 本文通过定义混合网格距离衡量此情况下个体的优劣.

定义 5. 混合网格距离 (hybrid grid distance, HGD). 定义个体到所在网格的最小边界点的欧氏距离作为网格混合网格距离来判断同一网格内个体的优劣:

$$HGD(X_i) = \sqrt{\sum_{m=1}^M (f_m(X_i) - (l_m + (G_m(X_i) - 1) \times d_m))^2}, \quad (6)$$

其中, $f_m(X_i)$ 为个体 X_i 的实际函数值, $G_m(X_i)$ 为个体 X_i 的网格坐标. 个体的混合网格距离越小, 说明离实际最优边界越近^①, 在排序值相同的情况下应当被优先选取. 在图 4 中, 个体 p_3, p_4 在同一网格内, 其排序值相同, 但是从图 4 中可以明显看出, p_3 的混合网格距离小于 p_4 , 说明 p_3 个体离真实最优边界近, 在这种情况下 p_4 应该被排除.

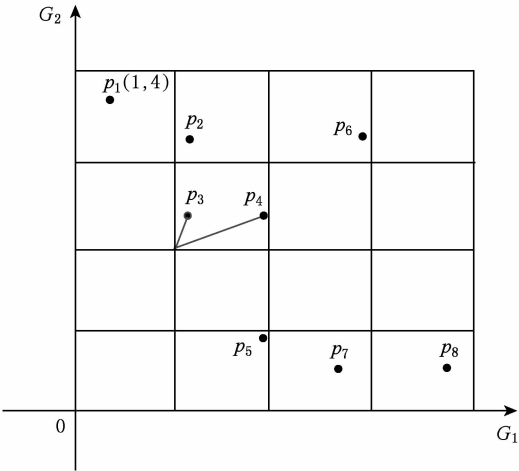


Fig. 4 A case of hybrid grid distance
图 4 一个混合网格距离实例

2.3 最优个体选取及外部档案集维护

2.3.1 最优个体选取

如引言所述, $pBest, gBest$ 是粒子群优化算法中 2 个非常重要的参数, 决定着整个种群的飞行方向, 直接影响整个算法的收敛效果. 对于 $pBest$ 的选择较简单, 主流的研究都是基于 Pareto 支配关系^[30], 在当前粒子位置与个体历史最优位置中选取非支配的粒子作为 $pBest$, 如果互不支配, 则在当中随机选取; $gBest$ 的选取则相对复杂, 许多学者也对此进行了研究, 主要有 3 类方法:

- 1) 随机选择^[8]. 对于存储于外部档案集的非支配个体来说, 随机选取一个个体作为群体的 $gBest$ 是最简单的方法, 容易出现在粒子集中的区域选择概率大, 不利于粒子在最优边界的分布, 降低了群体的多样性.
- 2) 聚集密度法^[31-32]. 计算当前粒子与外部档案集中非劣解的欧氏距离, 选取其中距离最小的粒子作为 $gBest$.
- 3) Mostaghim^[33] 利用构造的函数 $sigma$ 计算各粒子的值. 计算该粒子的 $sigma$ 值与外部档案集所有成员 $sigma$ 值之间的距离, 选择距离最小的外

部档案集中的成员作为该粒子的 $gBest$. 当解集中的成员过于拥挤时, $sigma$ 方法会增加算法的选择压力.

本文从粒子的全局网格排序、混合网格距离出发, 提出一种新的简洁高效的 $gBest, pBest$ 计算方法. 从 2.1 节、2.2 节的分析可以看出, 对于具有最大全局网格排序值的个体来说, 其支配解空间中其他剩余个体将此个体选为 $gBest$ 能使算法获得较好的收敛性; 如果多个个体同时具有最大网格排序值, 则利用定义 5 的混合距离选取距离最优边界最近的个体作为算法的 $gBest$.

对于 $pBest$ 的选取相对简单, 比较当前粒子与历史最优粒子的全局网格排序, 选取排序值较大者作为 $pBest$. 与 $gBest$ 选取类似, 如果两者排序值相同, 则利用混合距离筛选.

依据“全局网格排序为主, 混合网格密度为辅”与最优粒子 ($gBest, pBest$) 选取策略, 基于网格排序的多目标粒子群算法具有与单目标粒子群算法媲美的高效率, 只需判断单个排序值与密度信息, 算法收敛速度快且能获得分布性较好的最优解集.

2.3.2 外部档案集的维护

保存与维护非支配解是多目标粒子群算法的重要部分, 一般处理方法是建立一个外部档案集用来存放获得的非劣解. 当非劣解集的数量达到规定值时, 需要对其进行维护, 一方面提高算法搜索效率, 另一方面维护 $gBest$ 的多样性, 从而为种群的进化提供更大的搜索空间. 为粒子选择合适的 $gBest$ 能增加外部档案集的多样性, 两者互相促进, 获得更加均匀的最优边界集.

本文将外部档案集大小设定为 N , N 为种群大小, 先利用混合网格距离 (式 (6)) 排除掉同一网格中的其他粒子, 再利用 2.2 节的混合网格距离与 2.1.2 节的粒子全局网格排序信息 (定义 3), 每次取剩余个体降序排列的前 50% ($N/2$) 加入外部档案集. 由于外部档案集中整个群体的分布已发生改变, 所以每次需重新计算综合排序值 (定义 3), 再根据得到的综合排序值排序, 取得 N , 得到整个算法的最优边界值.

2.4 算法流程

根据标准粒子群优化算法的框架流程, 结合全局余量排序策略, GrRMOPSO 算法描述如下:

① 对于最小化的优化问题, 真实最优边界靠近坐标原点, 因此到网格左下角的距离能近似地反映个体到真实最优边界的距离.

算法 1. GrRMOPSO.

① 初始化:种群大小 N 、目标个数 M ;
② for 种群中的粒子 i
 随机产生粒子的速度与位置;
 网格空间映射;
 评估;
end for
③ 更新外部档案集;
④ for $t=1$ to 最大迭代次数
 for 种群中的粒子 i
 选出含多个粒子的网格,利用式(6)排除掉网格中较劣的个体;
 计算粒子的全局网格排序(式(5));
 选取 $gBest, pBest$;
 更新个体的位置与速度;
 取剩余个体降序排列的前 $50\%(N/2)$ 加入外部档案集;
 end for
 for 外部档案集中每个粒子 j
 选出含多个粒子的网格,利用式(6)排除掉网格中较劣的个体;
 计算粒子的全局网格排序(式(5));
 利用 2.3 节策略得到外部档案集;
 end for
 更新外部档案集;
 $t \leftarrow t+1$;
end for

2.5 算法复杂度分析

算法的时间复杂度表现为算法的执行时间随着输入规模的增长而增长的数量级大小,在很大程度上反映出一个算法的优劣程度.在本文所提的基于网格排序的多目标粒子群算法中,设种群规模为 N 、目标个数为 M ,主要步骤的时间复杂度如表 1 所示:

Table 1 Computational Complexity about GrRMOPSO

表 1 GrRMOPSO 时间复杂度分析

Scenario	Time Complexity
Global Grid Ranking	$O(MN^2)$
Selection of $pBest$	$O(N)$
Selection of $gBest$	$O(N)$
Updating the <i>velocity</i> and <i>position</i>	$O(MN)$
External Archive Maintenance	$O(MN^2)$

可以看出,算法的复杂度主要在排序与外部档案集的维护计算过程中,整个算法的复杂度为 $O(MN^2)$,与 NAGA-II^[11],MOPSO^[8] 等优秀的多目标算法一致.

3 实验对比及分析

3.1 测试函数

为了验证算法的有效性,本节选取了 4 个代表性的多目标优化算法(NSGA-II^[11], σ MOPSO^[33], cd MOPSO^[34], D^2 MOPSO^[35]),并在一些标准测试函数上进行对比实验,包括 KUR^[36],ZDT{1,2}^[37],DTLZ{1,2,5}^[38].选取的这些测试函数涵盖了各种多目标优化问题,包括线性、非凸函数等,采用的对比算法的参数以及函数分别如表 2、表 3 所示.在比较实验中,各对比算法的种群大小均设置为 $N=100$,外部档案的最大容量均设为 $E=100$,最大迭代次数为 $T_{\max}=300$.

Table 2 Parameters for Candidates

表 2 5 种对比算法采用的参数

NSGA-II	σ MOPSO	cd MOPSO	D^2 MOPSO	GrRMOPSO
	$w=0.4,$ $P_c=1.0,$ $P_m=1/n$	$w=0.4,$ $c_1=2,$ $c_2=2,$ $R_m=0.03$	$w=0.4,$ $c_1=2,$ $c_2=2$	$w=0.4,$ $c_1=2,$ $c_2=2,$ $c=10$

3.2 评价标准

通常来说,对一个算法的评估主要从收敛性与解的多样性 2 方面考虑.本文采用逆世代距离(inverted generational distance, IGD)^[39]与间距(spacing, S)^[40]对算法进行评价.在这 2 个指标的基础上,本文采用错误率(error ratio, ER)^[41]对得到的解的质量进行评估.

1) 逆世代距离 IGD .其表示为当前搜索到的非支配解同 Pareto 前端的距离:

$$IGD(P, P^*) \triangleq \frac{\sum_{i=1}^{|P^*|} dist(P_i^*, P)}{|P^*|}, \quad (7)$$

其中, P^* 为真实 Pareto 解集中的一部分, P 为算法得到的最优解集, $|P^*|$ 为解集的个体数目, $dist(P_i^*, P)$ 为 $P_i^* (P_i^* \in P^*)$ 到 P 中距离最近个体的欧氏距离.从式(7)中可以看出,对于 P^* ,采样越多,越能接近覆盖算法整个真实最优边界,也能在一定程度上反映算法的分布性能. IGD 的值越小,说明搜索到解的收敛性与分布性越好.

Table 3 Benchmark Functions

表 3 测试函数

Instance	Expression	Dimension	Objective	Domain	Properties
KUR	$\begin{cases} f_1(x) = \sum_{i=1}^2 (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})) \\ f_2(x) = \sum_{i=1}^3 (x_i ^{0.8} + 5 \sin(x_i)^3) \end{cases}$	3	2	$-5 \leq x_i \leq 5$ ($i=1, 2, \dots, m$)	Disconnected PF
ZDT1	$\begin{cases} f_1(x) = x_1, f_2(x) = g(1 - \sqrt{f_1/g}) \\ g(x) = 1 + 9 \sum_{i=2}^m \frac{x_i}{(m-1)} \end{cases}$	30	2	$0 \leq x_i \leq 1$	Convex PF
ZDT2	$\begin{cases} f_1(x) = x_1, f_2(x) = g(1 - (f_1/g)^2) \\ g(x) = 1 + 9 \sum_{i=2}^m \frac{x_i}{(m-1)} \end{cases}$	30	2	$0 \leq x_i \leq 1$	Non-convex PF
DTLZ1	$\begin{cases} \min f_1(x) = 0.5x_1x_2 \cdots x_{m-1}[1 + g(x_m)] \\ \min f_2(x) = 0.5x_1x_2 \cdots (1 - x_{m-1})[1 + g(x_m)] \\ \vdots \\ \min f_{m-1}(x) = 0.5x_1(1 - x_2)[1 + g(x_m)] \\ \min f_m(x) = 0.5(1 - x_1)[1 + g(x_m)] \\ g(x_m) = 100[x_m + \sum_{x_i \in x_m} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))] \end{cases}$	10	3	$0 \leq x_i \leq 1$ ($i=1, 2, \dots, m$)	Linear PF
DTLZ2	$\begin{cases} \min f_1(x) = [1 + g(x_m)] \cos\left(x_1 \frac{\pi}{2}\right) \cdots \cos\left(x_{m-2} \frac{\pi}{2}\right) \cos\left(x_{m-1} \frac{\pi}{2}\right) \\ \min f_2(x) = [1 + g(x_m)] \cos\left(x_1 \frac{\pi}{2}\right) \cdots \cos\left(x_{m-2} \frac{\pi}{2}\right) \sin\left(x_{m-1} \frac{\pi}{2}\right) \\ \min f_3(x) = [1 + g(x_m)] \cos\left(x_1 \frac{\pi}{2}\right) \cdots \sin\left(x_{m-2} \frac{\pi}{2}\right) \\ \vdots \\ \min f_m(x) = [1 + g(x_m)] \sin\left(x_1 \frac{\pi}{2}\right) \\ g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2 \end{cases}$	10	3	$0 \leq x_i \leq 1$ ($i=1, 2, \dots, m$)	Concave
DTLZ5	$\begin{cases} \min f_1(x) = [1 + g(x_m)] \cos\left(\theta_1 \frac{\pi}{2}\right) \cdots \cos\left(\theta_{m-2} \frac{\pi}{2}\right) \cos\left(\theta_{m-1} \frac{\pi}{2}\right) \\ \min f_2(x) = [1 + g(x_m)] \cos\left(\theta_1 \frac{\pi}{2}\right) \cdots \cos\left(\theta_{m-2} \frac{\pi}{2}\right) \sin\left(\theta_{m-1} \frac{\pi}{2}\right) \\ \min f_3(x) = [1 + g(x_m)] \cos\left(\theta_1 \frac{\pi}{2}\right) \cdots \sin\left(\theta_{m-2} \frac{\pi}{2}\right) \\ \vdots \\ \min f_m(x) = [1 + g(x_m)] \sin\left(\theta_1 \frac{\pi}{2}\right) \\ g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2, \theta_i = \frac{1 + 2g(x_m)x_i}{2(1 + g(x_m))} \end{cases}$	10	3	$0 \leq x_i \leq 1$ ($i=1, 2, \dots, m$)	Concave

2) 间距 S . 描述解向量的分布性, 通过与收敛到实际 Pareto 前端的比较从而对算法搜索到的非支配解的分布程度进行度量, 计算方法为

$$S \triangleq \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}, \tag{8}$$

其中, $\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i, d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|), i, j = 1, 2, \dots, n, \bar{d}$ 是 d_i 的平均数, n 为 Pareto 解的个数, S 的值越小越好, 如果 $S=0$, 说明所求解集在解空间中是等间距分布.

3) 错误率 ER . 描述算法搜索到的非支配解的错误率, 计算方法为

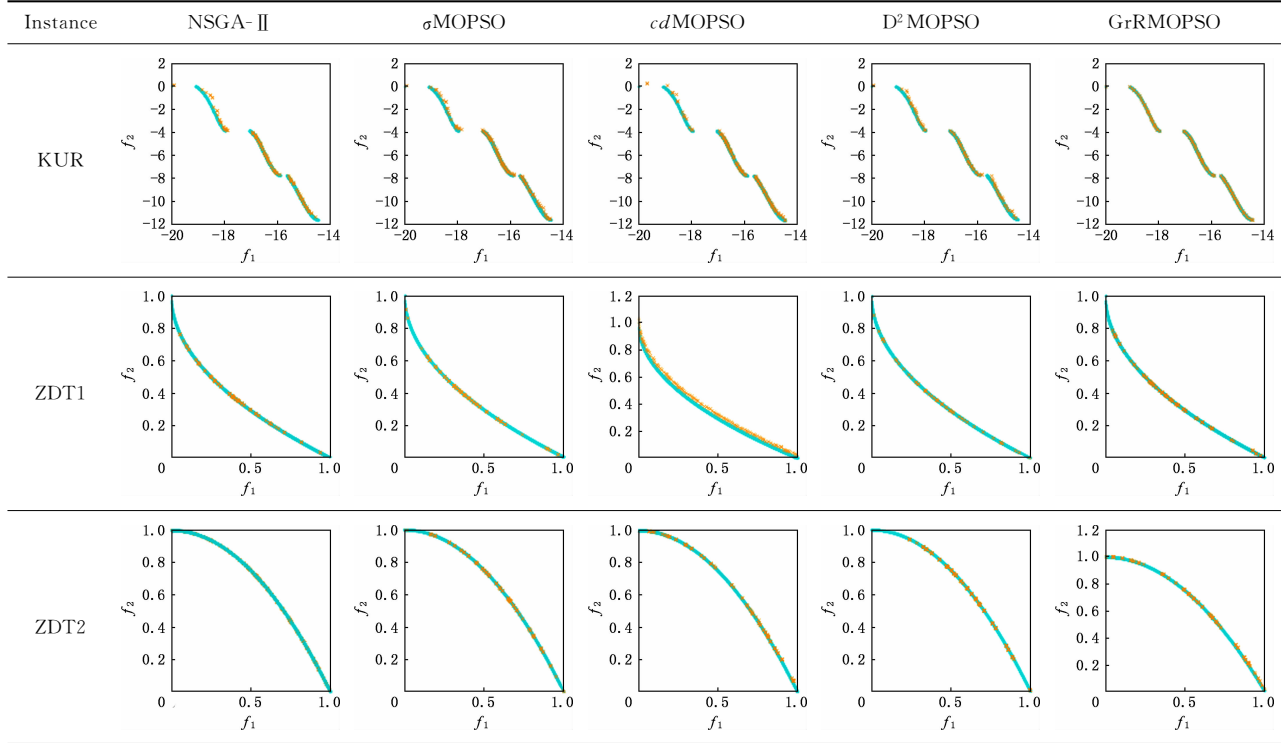
$$ER = \frac{1}{n} \sum_{i=1}^n e_i. \tag{9}$$

对于式(9), 如果第 i 个解是实际 Pareto 前端集合内的元素, 则 $e_i=0$, 否则 $e_i=1$.

3.3 实验结果及分析

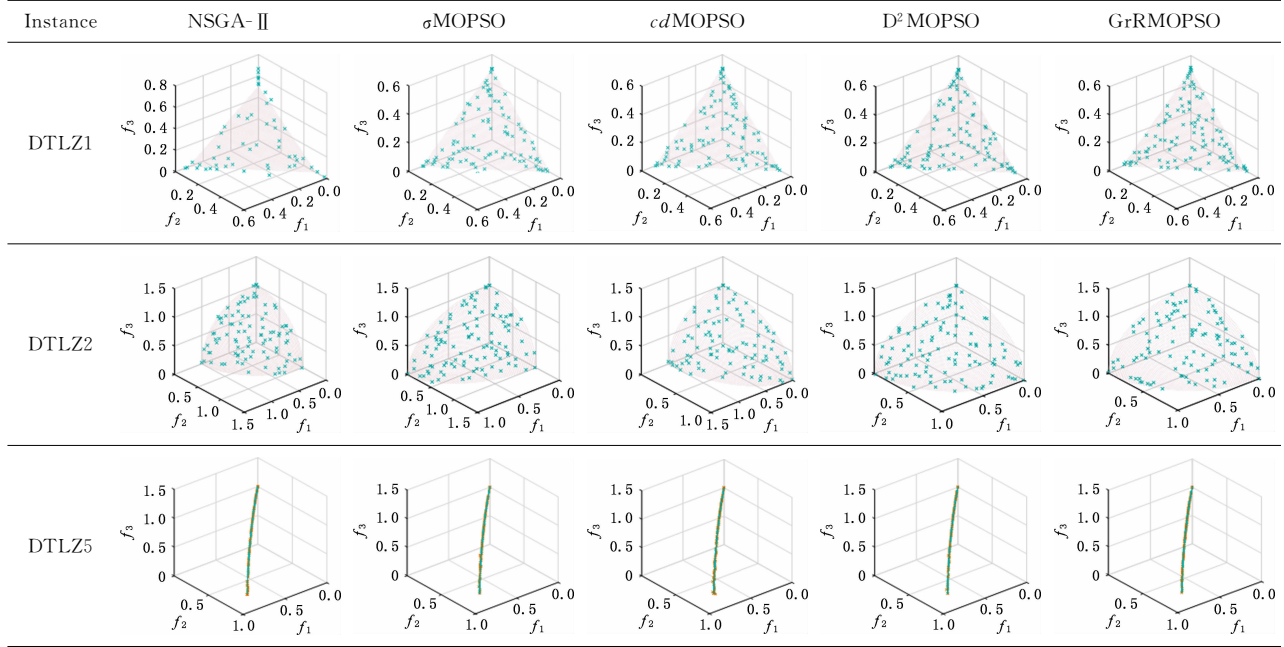
在进行详细的统计数据分析之前, 对各算法得到的解进行形象化展示是非常有意义的. 表 4、表 5 展示了各对比算法得到的非劣解集与真实 Pareto 前端的对比.

Table 4 Non-Inferior Solutions Obtained by Candidates on Bi-Objective Functions
表 4 5 种对比算法在双目标测试函数上的非劣解集



• True Pareto front; * Instance

Table 5 Non-Inferior Solutions Obtained by Candidates on Tri-Objective Functions
表 5 5 种对比算法在 3 目标测试函数上的非劣解集



• True Pareto front; * Instance

从表 4、表 5 中可以看出,各算法在简单的双目标问题上都能取得较好的效果,比较接近真实 Pareto 前端,说明这些算法能轻松应对简单的优化问题,特别是在 KUR 测试函数上,各对比算法都取

得了分布较均匀的解;在 3 目标优化问题上,对比算法效果稍微有点区别,NSGA-II 在 DTLZ1 问题上有一部分解偏离了真实 Pareto 前端。从总体上看,所提算法不论是在收敛性还是解的分布性上都具有

优势.

下面从统计数据的角度进一步说明. 为了减少随机性的影响, 各算法独立运行 30 次. 图 5 以盒图

(boxplot) 的形式展示了各算法在 *IGD* 指标上的对比情况, 表 6、表 7 展示了各算法在 *Spacing* 指标、*ER* 指标上的运行情况.

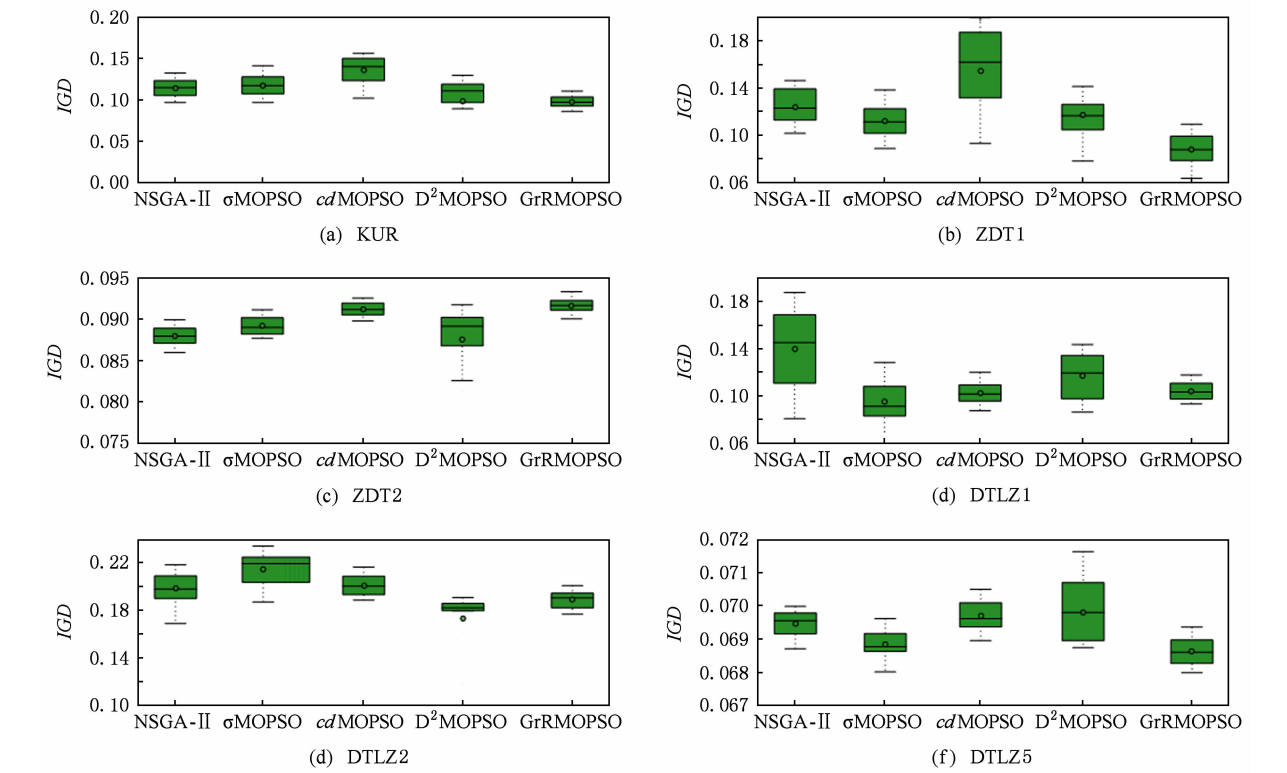


Fig. 5 IGD metric results on test suites

图 5 5 种算法在不同测试函数的 *IGD* 指标

Table 6 ER Metric Results

表 6 5 种算法上的 ER 指标

Instance	NSGA-II		σ MOPSO		cdMOPSO		D ² MOPSO		GrRMOPSO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
KUR	1.72E-01	6.77E-02	8.33E-02	2.98E-03	3.42E-02	1.19E-03	8.76E-03	1.07E-04	2.11E-03	1.34E-04
ZDT1	9.70E-02	3.22E-02	5.63E-02	1.27E-03	6.39E-01	2.02E-01	6.11E-02	2.43E-03	7.76E-02	2.76E-02
ZDT2	5.22E-02	3.10E-03	8.94E-02	4.30E-02	3.18E-02	9.10E-03	7.23E-03	2.06E-03	7.12E-03	2.33E-03
DTLZ1	3.27E-01	8.90E-02	3.90E-02	1.45E-02	8.01E-02	3.89E-02	9.45E-02	5.33E-03	1.22E-01	6.39E-03
DTLZ2	9.42E-02	2.27E-02	8.70E-02	4.12E-02	9.02E-02	3.20E-02	3.21E-03	1.22E-03	3.74E-03	1.37E-03
DTLZ5	5.08E-02	2.19E-02	4.53E-02	1.28E-02	5.22E-02	2.32E-02	5.36E-02	3.12E-02	4.29E-02	1.19E-02

Table 7 Spacing Metric Results

表 7 5 种算法上的 Spacing 指标

Instance	NSGA-II		σ MOPSO		cdMOPSO		D ² MOPSO		GrRMOPSO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
KUR	8.32E-02	2.19E-02	7.62E-02	1.33E-02	8.67E-02	3.03E-02	7.45E-02	2.55E-02	4.10E-02	8.20E-03
ZDT1	3.88E-02	1.71E-02	2.98E-02	1.03E-02	7.92E-01	5.40E-01	2.49E-02	8.23E-03	3.40E-02	1.09E-02
ZDT2	4.30E-03	8.11E-04	9.10E-03	3.41E-03	2.29E-02	9.54E-03	3.54E-02	8.29E-03	5.01E-03	9.20E-04
DTLZ1	1.98E-01	7.81E-02	2.94E-02	9.28E-03	1.53E-02	7.81E-03	3.09E-02	7.55E-03	2.89E-02	8.73E-03
DTLZ2	1.17E-01	7.78E-02	1.25E-01	8.93E-02	2.09E-01	8.41E-03	4.39E-02	1.28E-02	3.87E-02	9.44E-03
DTLZ5	5.46E-03	9.22E-04	5.32E-03	7.21E-04	8.97E-03	9.94E-04	5.91E-03	9.17E-04	4.47E-03	8.42E-04

从这些统计数据可以看出,GrRMOPSO 在大部分指标上占有优势,说明了所提算法的有效性.

3.4 网格划分数分析

在 GrRMOPSO 中存在一个需要用户调整的参数网格划分数 c , c 的大小对算法存在一定影响,如果 c 值过大,解空间划分过细,不能体现网格划分的

优势; c 值过小的话同一网格内粒子过多,会给后续的选择带来较大压力. 因此,本节讨论 c 值对算法性能的影响.

在 KUR 测试函数上采用 IGD 指标,验证 c 值从 5~20 之间的取值对算法性能的影响(其他参数保持不变),如图 6 所示:

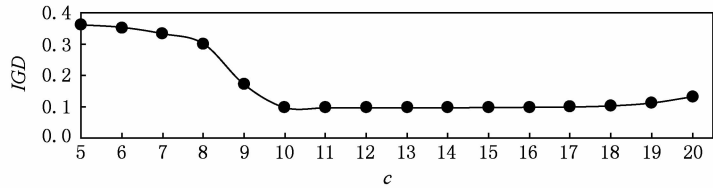


Fig. 6 IGD values with different numbers of c on KUR

图 6 不同网格划分数 c 在 KUR 函数上对应的 IGD 值

从图 6 中可以看出,在网格划分数小于 9 的时候对算法效果影响较大,即较小的划分数 c 会影响算法的性能,因为过多的粒子聚集在同一个网格内,给算法后续的选择带来较大压力.

另一方面,GrRMOPSO 主要是利用全局网格排序来判断个体优劣进而引导算法寻优,不同的网格划分数 c 影响了粒子的网格坐标,进而影响着算法的排序. 本节特定划分数 c 选取 4 个不同值分别为 5,10,15,20,采用 KUR 作为测试函数,将排序形象化地进行展示.

为了保持对比的统一性,减少随机性的误差,在不同 c 值情况下,我们利用程序随机产生一个初始种群,保存并固定为不同 c 值测试上的初始种群,利用粒子在解空间中的分布结合其排序值,得到不同迭代情况下排序策略对种群的引导情况,实验结果如图 7 所示.

从图 7 中可以看出,当网格划分数过小($c=5$),解空间中会存在过多相同排序值从而影响排序的效率,从另一方面验证了网格划分数的影响. 另外,在 $c=10$ 与 $c=20$ 之间, c 对排序值影响较小.

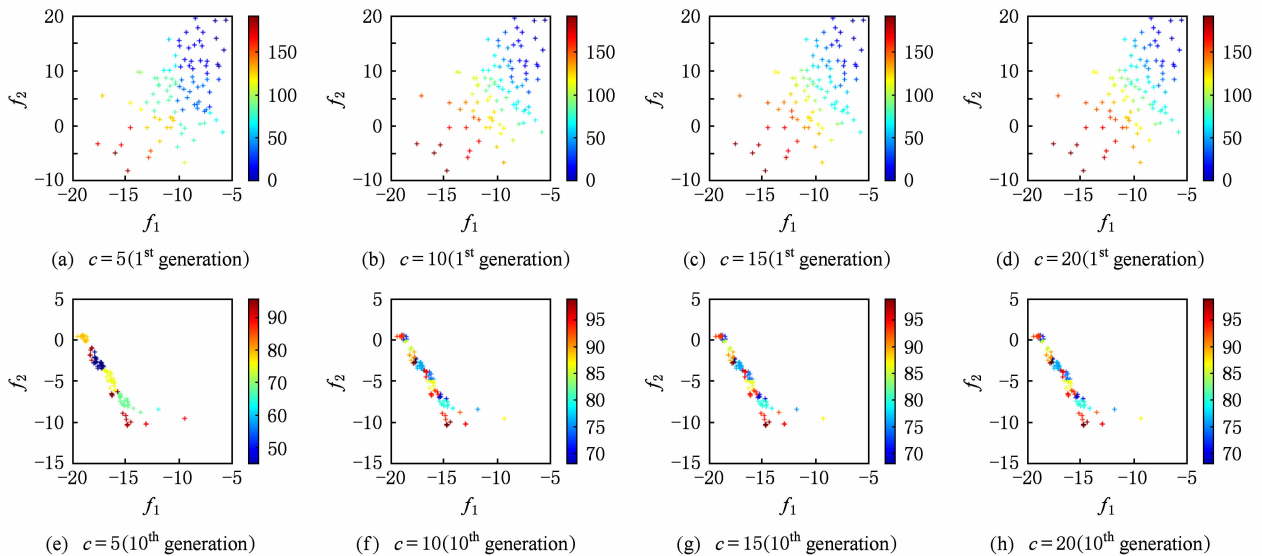


Fig. 7 A sketch map on ranking with different c (different colors represent different sort values)

图 7 不同网格划分数 c 对排序的影响示意图(不同颜色代表不同的排序值)

4 结束语

本文从网格坐标出发,将决策空间进行映射,利用网格的同时表现收敛性与分布性的特点,利用个

体对解空间中其他剩余粒子的占优情况,结合整个解空间中粒子的分布信息对粒子进行排序;提出了一种基于网格的距离计算方式,用于处理相同网格内多个粒子的情况;在此基础上,提出了一种基于网格排序的多目标粒子群优化算法. 通过分析与实验

可以得出:

1) 通过全局网格排序能够较好地得到粒子的排序信息,进而引导算法收敛;

2) 网格划分数对算法效果有一定影响,过小的网格划分不利于算法的排序操作.

当前的基于网格排序的框架只是应用在低维(2维、3维)优化问题以及粒子群算法并取得了较好的效果,下一步的研究计划将此框架扩展到高维多目标优化问题,毕竟高维优化问题对 Pareto 排序更为敏感.另外,将基于网格排序的机制应用于其他优秀的进化算法也是下一步即将开展的工作.

参 考 文 献

- [1] Kennedy J, Eberhart R. Particle swarm optimization [C] // Proc of IEEE Int Conf on Neural Networks. Piscataway, NJ: IEEE, 1995: 1942-1948
- [2] Li Jie, Zhang Junqi, Jiang Changjun, et al. Composite particle swarm optimizer with historical memory for function optimization [J]. IEEE Trans on Cybernetics, 2015, 45(10): 2350-2363
- [3] Shou Yongyi, Li Ying, Lai Changtao. Hybrid particle swarm optimization for preemptive resource-constrained project scheduling [J]. Neurocomputing, 2015, 148: 122-128
- [4] Chiou Juingshian, Tsai Shunhung, Liu Mingtang. A PSO-based adaptive fuzzy PID-controllers [J]. Simulation Modelling Practice and Theory, 2012, 26: 49-59
- [5] Zhao Yulei, Guo Baolong, Wu Xianxiang, et al. Image reconstruction algorithm for ECT based on dual particle swarm collaborative optimization [J]. Journal of Computer Research and Development, 2014, 51(9): 2094-2100 (in Chinese)
(赵玉磊, 郭宝龙, 吴宪祥, 等. 基于双粒子群协同优化的ECT图像重建算法[J]. 计算机研究与发展, 2014, 51(9): 2094-2100)
- [6] Dujardin Y, Vanderpooten D, Boillot F. A multi-objective interactive system for adaptive traffic control [J]. European Journal of Operational Research, 2015, 244(2): 601-610
- [7] Lee K-B, Kim J-H. Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots [J]. IEEE Trans on Evolutionary Computation, 2013, 17(6): 755-766
- [8] Coello C A C, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization [C] //Proc of the 2002 Congress on Evolutionary Computation. Piscataway, NJ: IEEE, 2002: 1051-1056
- [9] Xie Chengwang, Zou Xiufen, Xia Xuewen, et al. A multi-objective particle swarm optimization algorithm integrating multiply strategies [J]. Acta Electronica Sinica, 2015, 43(8): 1538-1544 (in Chinese)
(谢承旺, 邹秀芬, 夏学文, 等. 一种多策略融合的多目标粒子群优化算法[J]. 电子学报, 2015, 43(8): 1538-1544)
- [10] Mao Chengying, Yu Xinxin, Xue Yunzhi. Algorithm design and empirical analysis for particle swarm optimization-based test data generation [J]. Journal of Computer Research and Development, 2014, 51(4): 824-837 (in Chinese)
(毛澄映, 喻新欣, 薛云志. 基于粒子群优化的测试数据生成及其实证分析[J]. 计算机研究与发展, 2015, 51(4): 824-837)
- [11] Coello C A C, Lechuga M S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197
- [12] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization [C] //Proc of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Berlin: Springer, 2001: 95-100
- [13] Deb K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems [J]. Evolutionary Computation, 1999, 7(3): 205-230
- [14] Laumanns M, Thiele L, Deb K, et al. Combining convergence and diversity in evolutionary multiobjective optimization [J]. Evolutionary Computation, 2002, 10(3): 263-282
- [15] Batista L S, Campelo F, Guimarães F G, et al. Pareto cone ϵ -dominance: Improving convergence and diversity in multiobjective evolutionary algorithms [C] //Proc of Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2011: 76-90
- [16] Zou Xiufen, Chen Yu, Liu Mingzhou, et al. A new evolutionary algorithm for solving many-objective optimization problems [J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2008, 38(5): 1402-1412
- [17] Marler R T, Arora J S. The weighted sum method for multi-objective optimization: New insights [J]. Structural and Multidisciplinary Optimization, 2010, 41(6): 853-862
- [18] Zhang Qingfu, Li Hui. MOEA/D: A multiobjective evolutionary algorithm based on decomposition [J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731
- [19] Asafuddoula M, Ray T, Sarker R. A decomposition based evolutionary algorithm for many objective optimization [J]. IEEE Trans on Evolutionary Computation, 2015, 19(3): 445-460
- [20] Balling R, Wilson S. The maxi-min fitness function for multi-objective evolutionary computation: Application to city planning [C] //Proc the 2001 Annual Conf on Genetic and Evolutionary Computation (GECCO'2001). New York: ACM, 2001: 1079-1084
- [21] Wei Lingyun, Zhao Mei. A niche hybrid genetic algorithm for global optimization of continuous multimodal functions [J]. Applied Mathematics and Computation, 2005, 160(3): 649-661
- [22] Pulido G T, Coello C A C. Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer [C] //Proc of the 2004 Annual Conf on Genetic and Evolutionary Computation (GECCO'2004). Berlin: Springer, 2004: 225-237

- [23] Luo Biao, Zheng Jinhua, Xie Jiongliang, et al. Dynamic crowding distance? A new diversity maintenance strategy for MOEAs [C] //Proc of the 4th Int Conf on Natural Computation. Piscataway, NJ: IEEE, 2008; 580-585
- [24] Wang Yaonan, Wu Lianhong, Yuan Xiaofang. Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure [J]. *Soft Computing*, 2010, 14(3): 193-209
- [25] Leong W F, Yen G G. PSO-based multiobjective optimization with dynamic population size and adaptive local archives [J]. *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, 38(5): 1270-1293
- [26] Knowles J, Corne D. Properties of an adaptive archiving algorithm for storing nondominated vectors [J]. *IEEE Trans on Evolutionary Computation*, 2003, 7(2): 100-116
- [27] Yang Shengxiang, Li Miqing, Liu Xiaohui, et al. A grid-based evolutionary algorithm for many-objective optimization [J]. *IEEE Trans on Evolutionary Computation*, 2013, 17(5): 721-736
- [28] Li Bingdong, Li Jinlong, Tang Ke, et al. Many-objective evolutionary algorithms: A survey [J]. *ACM Computing Surveys (CSUR)*, 2015, 48(1): 1-13
- [29] Adra S F, Fleming P J. Diversity management in evolutionary many-objective optimization [J]. *IEEE Trans on Evolutionary Computation*, 2011, 15(2): 183-195
- [30] Alvarez-Benitez J E, Everson R M, Fieldsend J E. A MOPSO algorithm based exclusively on pareto dominance concepts [C] //Proc of Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2005; 459-473
- [31] Raquel C R, Naval Jr P C. An effective use of crowding distance in multiobjective particle swarm optimization [C] //Proc of the 7th Annual Conf on Genetic and Evolutionary Computation. New York: ACM, 2005; 257-264
- [32] Dai Cai, Wang Yuping, Ye Miao. A new multi-objective particle swarm optimization algorithm based on decomposition [J]. *Information Sciences*, 2015, 325: 541-557
- [33] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO) [C] //Proc of 2003 IEEE Swarm Intelligence Symp. Piscataway, NJ: IEEE, 2003; 26-33
- [34] Raquel C R, Naval Jr P C. An effective use of crowding distance in multiobjective particle swarm optimization [C] //Proc of the 7th Annual Conf on Genetic and Evolutionary Computation. New York: ACM, 2005; 257-264
- [35] Al Moubayed N, Petrovski A, McCall J. D²MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces [J]. *Evolutionary Computation*, 2014, 22(1): 47-77
- [36] Kursawe F. A variant of evolution strategies for vector optimization [M]. Berlin: Springer, 1991; 193-197
- [37] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results [J]. *Evolutionary Computation*, 2000, 8(2): 173-195
- [38] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems [C] //Proc of 2002 IEEE Congress on Evolutionary Computation. Piscataway, NJ: IEEE, 2002; 825-830
- [39] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multiobjective optimizers: an analysis and review [J]. *IEEE Trans on Evolutionary Computation*, 2003, 7(2): 117-132
- [40] Schott JR. Fault tolerant design using single and multicriteria genetic algorithm optimization [D]. Boston: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995
- [41] Van Veldhuizen DA. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations [D]. Ohio: Department of Electric Computer Engineer, Air Force Institute of Technology, 1999



Li Li, born in 1986. PhD candidate of Zhejiang University of Technology, Hangzhou, Zhejiang, China. Student member of CCF. His main current research interests include evolutionary computation, scheduling optimization, and multi-objective optimization.



Wang Wanliang, born in 1957. Professor and PhD supervisor in Zhejiang University of Technology, Hangzhou, Zhejiang, China. His main research interests include intelligent computing, scheduling optimization.



Xu Xinli, born in 1977. PhD. Associate professor in Zhejiang University of Technology, Hangzhou, Zhejiang, China. Her main research interests include intelligent computing, scheduling optimization, and wireless sensor networks (xxl@zjut.edu.cn).



Li Weikun, born in 1990. Master candidate of Zhejiang University of Technology, Hangzhou, Zhejiang, China. Student member of CCF. His main research interests include evolutionary computation and multi-objective optimization (lw_k_zjut@hotmail.com).