

基于大规模变量分解的多目标粒子群优化算法研究

邱飞岳^{1),2)} 莫雷平¹⁾ 江波²⁾ 王丽萍³⁾

¹⁾(浙江工业大学信息工程学院 杭州 310023)

²⁾(浙江工业大学现代教育技术研究所 杭州 310023)

³⁾(浙江工业大学信息智能与决策优化研究所 杭州 310023)

摘 要 含有大规模变量的多目标优化问题是目前多目标进化算法领域的研究重点,多目标粒子群优化方法具有收敛性良好、计算简单和参数设置少等优点,但随着优化问题决策变量的增多,“变量维度”成为了瓶颈.针对上述问题,文中提出的变量随机分解策略,增加关联变量分配到同组的概率,使得算法更好的保留变量间的关联性,并将合作协同进化框架融合到算法中,提出了基于大规模变量分解的多目标粒子群优化算法(CCMOPSO).将该算法在经典标准测试函数 ZDT1、ZDT2、ZDT3、DTLZ1、DTLZ2 变量扩展后进行仿真对比实验,采用加法二进制 ϵ 指标和超体积指标(HV)对算法收敛性和多样性进行对比分析,实验结果表明,在解决大规模变量的多目标函数中,变量维度越高,该算法比经典多目标算法 MOPSO、NSGA-II、MOEA/D 以及 GDE3 越具有更好的多样性与收敛性,同时使得计算复杂度明显降低.

关键词 粒子群优化;大规模变量;随机分解;合作协同;全局优化

中图法分类号 TP391 **DOI号** 10.11897/SP.J.1016.2016.02598

Multi-Objective Particle Swarm Optimization Algorithm Using Large Scale Variable Decomposition

QIU Fei-Yue^{1),2)} MO Lei-Ping¹⁾ JIANG Bo²⁾ WANG Li-Ping³⁾

¹⁾(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

²⁾(Institute of Modern Educational Technology, Zhejiang University of Technology, Hangzhou 310023)

³⁾(Institute of Information Intelligence and Decision Optimization, Zhejiang University of Technology, Hangzhou 310023)

Abstract The multi-objective optimization of problems with large scale variable has become a focus in multi-objective evolutionary algorithm research field. Multi-objective particle swarm optimization algorithm is of better convergence, easier calculation and less parameter settings, yet “variable dimensionality” will be triggered as strategic variables increase. To solve the problem, this paper proposes random variable decomposition strategy, i. e. to promote the possibility of distributing associated variables into one group by random variable decomposition on the basis of variable groups, so as to realize better maintenance in association between variable groups. CCMOPSO is proposed through the integration of cooperative co-evolution evolutionary frame into the large scale variable decomposition. Comparative simulation experiment is conducted after the variable extension on typical standard functions of ZDT1, ZDT2, ZDT3, DTLZ1 and DTLZ2. Comparison between convergence and diversity of the algorithm with the binary addition index ϵ and hyper-volume indicator (HV), shows this algorithm is of better diversity, convergence and

收稿日期:2015-01-26;在线出版日期:2015-10-29. 本课题得到国家自然科学基金面上项目(61472366,61379077,61503340)、浙江省自然科学基金(LZ13F020002,LY13F030010,LY17F020022)资助. 邱飞岳,男,1965年生,博士,教授,博士生导师,主要研究领域为智能控制、优化理论与方法、学习科学与媒体技术. E-mail: qfy@zjut.edu.cn. 莫雷平,男,1990年生,硕士研究生,主要研究方向为信息处理与决策优化. 江波,男,1985年生,博士,副教授,主要研究方向为多目标进化算法及应用. 王丽萍,女,1964年生,博士,教授,博士生导师,主要研究领域为信息处理、决策优化、商务智能. E-mail: wlp@zjut.edu.cn.

easiness in multi-objective function with large scale variable than MOPSO, NSGA-II, MOEA/D and GDE3, and computational complexity is decreased.

Keywords particle swarm optimization; large scale variable; random decomposition; cooperative co-evolution; global optimization

1 引言

多目标进化算法(Multi-Objective Evolutionary Algorithm, MOEA)利用进化算法求解多目标优化问题,它不仅具有良好的全局最优性,而且单次运行能够提供多个候选解.其中,多目标粒子群算法(Multi-Objective Particle Swarm Optimization, MOPSO)以其良好的收敛性、简单的计算和较少的参数设置,被众多学者关注.

但随着实际应用问题的复杂化,多目标粒子群优化算法已经难以满足实际应用问题的需要,许多实际应用问题中含有成百上千个变量.如商品回收中逆向物流网络的设计,目标为最小化供应链中的物流成本以及缺货或库存成本,因此制造商在一定区域内将设计几十个回收点和处理点,由此产生上千个决策变量.这对多目标粒子群优化算法遭遇的“维度灾难”是一种考验,因为其优化性能随着变量的增加而显著下降.为了解决这一瓶颈问题,现有的学者试图从变量分解的角度进行研究.变量分解方法首先将大规模变量分解为若干个维度适中的变量组,然后将每组变量视为一个子种群,再通过子种群间的合作协同实现问题求解.这种“分而治之”的思想即合作协同进化(Cooperative Co-evolution, CC)^[1]显著地提高了现有进化算法在大规模变量优化问题上的求解效率,因而备受关注.

根据目前已有的研究成果,进化算法中的变量分解方法分为4类:(1)固定分解.这类方法将变量分解为固定的几个分组,为每个变量组分配一个子种群,再通过子种群间的合作协同进行搜索.此类方法实际上是合作协同进化(Cooperative Co-evolution)方法^[1]的原型.其主要代表有 Potter 和 De Jong,在最早提出的合作协同进化算法中,人们就为每个变量分配一个种群.而在多目标优化中,Keeratitittumrong 等人^[2]率先提出为每个变量分配一个子种群.文献^[3-4]中所提出的进化多目标优化方法也同样使用了类似的分解机制;(2)随机分解.这类方法是在没有任何先验信息的情况下,通过将变量进行随机分组来实现问题的分解.这种思路在合作协同进化方法中也受到了广泛关注^[5-6].如 Yang 和 Yao 等人^[7]

率先提出的随机分组(Random Grouping)策略;基于高频率随机分组的分解策略^[8];Li 和 Yao^[9]进一步提出了基于动态高频率随机分组的分解策略.最近 Antonio 和 Coello Coello^[10]将随机分组策略用于求解含有大规模变量的多目标优化问题,但是他们对多目标粒子群算法没有进行改进;(3)扰动分解.这类方法是在给定关联变量间所要满足的条件下,不断地扰动变量来发现关联变量.此类方法在合作协同进化方法研究中也受到了关注.比如,基于变量相关系数的分解策略^[11]、基于最优个体选择的变量交互学习(Variable Interaction Learning)^[12]、基于统计概率的变量交互学习^[13]以及最近提出的基于差分分组的变量交互学习方法^[14],但其计算复杂度显著增加;(4)关联自适应分解.这类方法先从遗传算法染色体的二进制编码水平上发现基因间的关联,再为包含关联变量的个体赋予较高的选择概率^[15].该类方法与扰动方法的不同之处是直接进化过程中来实现变量分解^[16].

现有的变量分解方法大多采用固定分组的模式,且缺乏发现变量相关性的有效办法,况且变量分解方法大多集中在单目标优化问题上.本文引入变量随机分解策略,研究了变量分解的关联性,并将合作协同框架融合多目标粒子群算法(MOPSO),提出了基于变量分解的多目标粒子群优化算法(CCMOPSO),该算法与经典算法 NSGA-II、MOEA/D、GDE3 在测试函数 ZDT1、ZDT2、ZDT3、DTLZ1、DTLZ2 中进行仿真实验和性能测试对比.

2 背景知识

2.1 多目标优化概念

针对多目标优化问题,我们可以假定一个含有 n 维决策变量, m 维目标变量的多目标优化问题^[17]可以概述为

$$\begin{cases} \max y = (f_1(x), f_2(x), \dots, f_n(x))^T \\ \text{s. t. } g_i(x) \leq 0, i = 1, 2, \dots, p \\ h_j(x) \leq 0, j = 1, 2, \dots, q \end{cases} \quad (1)$$

式(1)中, $f_n(x)$ ($k=1, \dots, n$) 为目标函数, $g_i(x)$ ($i=1, \dots, p$) 和 $h_j(x)$ ($i=1, \dots, q$) 分别为目标函数所满足的约束.

$x = (x_1, x_2, \dots, x_n) \in X, y = (y_1, y_2, \dots, y_n) \in Y$.

定义 1(可行解集合)^[18]. 对于一个 $x \in X$, 如果符合约束条件 $g_i(x) (i=1, \dots, p)$ 和 $h_j(x) (i=1, \dots, q)$, 则称 x 为可行解, 由 X 中的所有可行解组成的集合称为可行解集合, 记为 X_f 且 $X_f \subseteq X$.

定义 2(Pareto 占优)^[18]. 假设 $x_A, x_B \in X_f$ 是多目标优化问题中的可行解, 则称与 x_B 相比, x_A Pareto 支配 x_B , 当且仅当

$$\begin{aligned} & \forall i=1, 2, \dots, m, \\ & f_i(x_A) \leq f_i(x_B) \wedge \exists j=1, 2, \dots, m, \\ & f_j(x_A) < f_j(x_B) \end{aligned}$$

记为 $x_A \succ x_B$, 也称 x_A 占优 x_B .

定义 3(Pareto 最优解)^[18]. 一个解 $x^* \in X_f$ 称为 Pareto 最优解, 当且仅当满足以下条件:

$$\neg \exists x \in X_f: x \succ x^* .$$

2.2 多目标粒子群优化算法

粒子群优化算法 (Particle Swarm Optimization, PSO)^[19] 是由 Kennedy 和 Eberhart 于 1995 年提出的一种受鸟群觅食和社会交互启发的新型智能优化算法. PSO 利用粒子个体的个体认知和社会交互来引导群体收敛到潜在的全局最优区域. 最早提出的粒子群算法通常用式(2)~(3)更新粒子状态.

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (2)$$

$$x_{id} = x_{id} + v_{id} \quad (3)$$

式(2)和式(3)中: c_1 和 c_2 为正常数, 我们称之为学习因子; r_1 和 r_2 是介于 $[0, 1]$ 间的随机数; $v_{id} \in [-V_{\max}, V_{\max}]$, V_{\max} 是粒子飞行的最大速度, 表示个体每代更新的最大步长.

式(2)中, v_{id} 是指个体的当前速度对自身下一代速度的影响. $(p_{id} - x_{id})$ 表示的是个体认知能力 (个体历史最优位置) 对粒子学习步长的影响; $(p_{gd} - x_{id})$ 是指个体的社会交互能力对学习步长的影响, 即当前种群中的全局最优位置对粒子的影响. 粒子个体的飞行方式如图 1 所示.

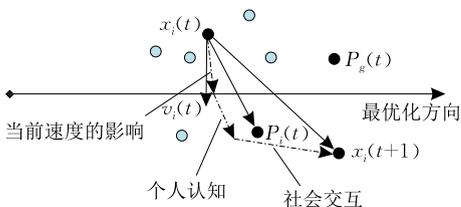


图 1 粒子个体的“飞行”方式

在粒子群优化算法中, 引导粒子的选择对算法的收敛性起着重要的作用. 将其引入多目标优化领域时, 由于多目标优化问题的解是一群最优粒子, 因此引导粒子的选择变得复杂. 如何保证这群粒子均匀分布并从中选择全局最优个体, 从而引导粒子群的“飞

行”, 也是多目标粒子群优化算法的关键. 现已提出的多目标粒子群优化算法主要是通过引入精英保留策略, 即设置外部种群, 用于存放历代获得的非支配解, 利用小生境技术、拥挤距离法、Sigma 方法等控制外部种群中粒子的分布. 另一方面, 在不损失算法收敛性的同时, 引入新的变异策略, 以一定概率对当代种群进行变异, 从而增加粒子群的多样性.

2.3 合作协同进化框架

进化算法中对于大规模变量的处理机制始于 Potter 和 De Jong^[1] 在 1995 年提出的“分而治之” (divide and conquer) 策略, 又称为合作协同进化 (Cooperative Co-evolution, CC). Potter 和 De Jong 提出的合作协同进化策略与传统的进化算法的最大不同在于: 它先通过将一个完整的 n 维变量分解为 m 个 s 维 ($n=m \times s$) 的子变量集合 (如图 2 所示), 而后使用一定数量的种群在每个子变量集合的可行域中进行搜索, 再将在每个子空间上搜索到的局部解结合成为一个完整解. 现有研究表明这种“分而治之”的思想在处理大变量函数时具有良好的效果.

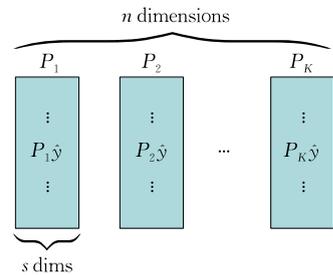


图 2 合作协同进化中的变量分解示意

3 基于变量分解的多目标粒子群优化算法 (CCMOPSO)

3.1 变量随机分解策略

综述国内外研究现状, 目前有关多目标优化中的变量分解研究成果并不多见. 尽管早期已有合作协同框架的“分而治之”的思想, 且在解决大规模变量优化问题时有较好的效果, 但是有文献提出, 当变量之间不是相互独立时, 使用合作协同框架的效果并不理想, 主要原因为变量分解是固定的, 而在绝大多数的大规模变量多目标优化问题中, 变量之间往往存在着内在的关联. 固定的变量分解会导致这些内在关联信息大量丢失. 本文所给出的变量随机分解方法, 目的是通过挖掘变量间的内在关联信息, 使关联变量尽可能的被归在同一组中.

本文提出的一种变量随机分解策略, 如图 3 所示. 将 D 维的决策向量组随机分配到 S 个子元素, 这里使用随机分组的目的是增加关联变量分到同一组

的概率,每个子元素的维度为 m ,满足 $D=m \times S$.将 S 个子元素设成每个有 NP 个粒子(个体)的子种群.

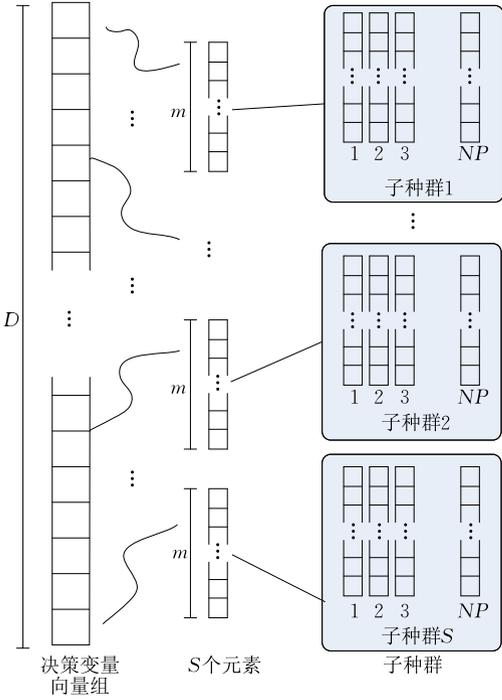


图 3 变量随机分解过程

3.2 算法框架描述

为了更好地描述种群间的协同方法,现设 y_i 为粒子 i 当前的个体最优位置, \hat{y} 为各子种群当前的全局最优位置,所有粒子的速度和位置 v_i 和 x_i 分别按式(2)、(3)进行更新.

各子种群间个体的速度和位置更新方法如下:

先对其进行适应度评估, $b(j, z)$ 为适应度函数, 定义为

$$b(j, z) = (P_1 \hat{y}, \dots, P_{j-1} \hat{y}, z, P_{j+1} \hat{y}, \dots, P_S \hat{y}) \quad (4)$$

其中, $z = P_j x_i$ 是子种群 j 中的粒子 i 当前搜索到的位置. $z = P_j y_i$ 表示为子种群 j 中粒子 i 的个体最优位置. $z = P_j \hat{y}$ 表示子种群 j 搜索到的全局最优位置.

由得到的适应度代入,可得到目标函数 $f_1(b(j, z))$, $f_2(b(j, z))$, 对其按照 NSGA-II 中的非支配排序方法对其进行非支配排序,随机选择非支配排序等级最高的粒子中的一个作为子种群 j 的最优位置.以子种群 1 为例,种群间的协同过程如图 4 所示.算法框图如图 5 所示,子种群 1 到 S 用多目标粒子群优化算法进行优化,对各子种群的解进行非支配排序,若满足终止条件,则将各个子种群的最优非支配解作为输出结果.若不满足终止条件,则进入下一轮迭代,将父代子种群的最优非支配解集用于子代的协同.

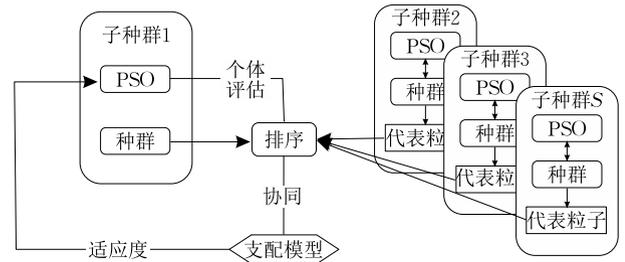


图 4 子种群间的协同

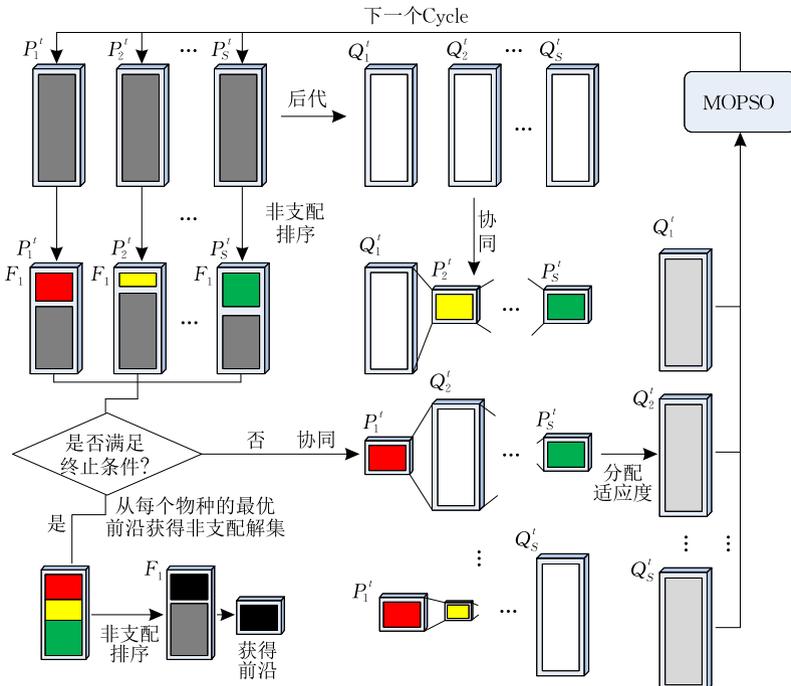


图 5 算法流程

3.3 算法流程

算法流程如图 6 所示.

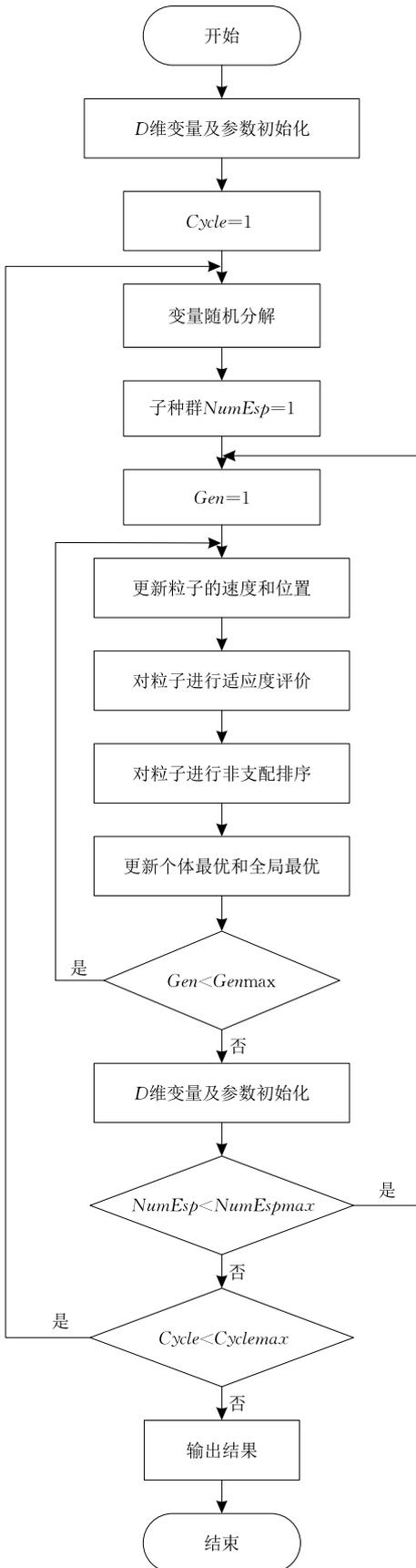


图 6 CCMOPSO 算法流程

步骤 1. 初始化 D 维变量, 每个子种群粒子个数为 NP , 初始化粒子的速度和位置, 计算粒子的适应度值, 对粒子进行非支配排序, 随机选择非支配等级最高的粒子中的一个作为 $gbest$, 设定最大周期 $Cyclemax$, 子种群数量 $NumEspmax$, 种群最大迭代次数 $Genmax$;

步骤 2. $Cycle=1$, 周期开始执行, 对 D 维变量随机分解成 $NumEspmax$ 个子种群, 每个子种群粒子个数为 NP ;

步骤 3. $NumEsp=1$, 对子种群中的第 1 个进行迭代处理;

步骤 4. $Gen=1$, 对子种群进行迭代, 以粒子当前位置为个体最优位置, 步骤 1 中的 $gbest$ 为全局最优位置, 按照式(2)和式(3)更新粒子的速度和位置, 用式(4)对粒子进行适应度评价, 按照 NSGA-II 中的非支配排序方法对粒子进行排序, 随机选出非支配等级最高粒子中的一个作为 $gbest$, 更新个体最优和全局最优, 进入下一次的迭代, $Gen=Gen+1$;

步骤 5. 满足 $Gen=Genmax$, 进行外部种群维护, 则 $NumEsp=NumEsp+1$, 进入下一个子种群处理, 下一个子种群处理中, 将上一个子种群中得到的 $gbest$ 作为全局最优进行更新, 其他操作和步骤 4 相同;

步骤 6. 当 $NumEsp=NumEspmax$ 时, 重新进行变量随机分解, 重复执行步骤 3、4、5;

步骤 7. 一般情况下, 当满足 $Cycle=Cyclemax$ 时, 则输出结果, 程序结束.

该算法的程序框架如算法 1 所示.

算法 1. CCMOPSO 算法程序框架.

REQUIRE: $NP, Cycles, Gmax, NumEsp$

ENSURE: $SolutionSet$

$Pobs \leftarrow Populations(NP, NumEsp)$

$Initialize\ Species(Pobs)$

FOR $j \leftarrow 1$ Cycles DO

FOR $i \leftarrow 1$ NumEsp DO

FOR $k \leftarrow 1$ Gmax DO

MOPSO($Pobs[i]$)

END FOR

END FOR

END FOR

$SOLUTIONSET \leftarrow Obtain\ Non-Dominated\ Set(Pobs)$

RETURN $SolutionSet$

4 仿真实验及结果分析

本文选择常用二维测试函数 ZDT1、ZDT2、ZDT3 和高维目标测试函数 DTLZ1、DTLZ2 (如表 1), 选取 NSGA-II、MOEA/D 和 GDE3 算法, 与

CCMOPSO 算法进行对比. 性能评价采用统计前沿, 加法二进制指标 ϵ , 超体积指标 (HV), 算法运行时间. 各算法对每个测试函数分别独立运行 20 次. 选取 NSGA-II、MOEA/D 以及 GDE3 算法进行对比的目的为: (1) NSGA-II^[20] 算法是多目标优化的经典算法; (2) MOEA/D^[21] 是目前处理大变量优化问题的一种有效算法; (3) GDE3(差分进化算法)^[22]

经过多代改进, 是近年来提出的并被证明是解决大规模优化问题的有效算法. 本文以 Coello Coello 的多目标粒子群优化算法为基础, 采用变量随机分解策略, 融合 CC 框架, 目的在于解决大规模变量问题的“维度灾难”. 经过实验测试发现, MOPSO 算法无法处理大规模变量问题(当测试函数变量个数超过 100 时, 无法获得 Pareto 前沿).

表 1 测试函数

函数名称	目标函数表达式	约束范围	特性
ZDT1	$f_1(x) = x_1; f_2(x) = g(1 - \sqrt{f_1/g}); g(x) = 1 + 9 \sum_{i=2}^n x_i / (n-1)$	$0 \leq x_i \leq 1;$ $i = 1, 2, \dots, n$	连续凸
ZDT2	$f_1(x) = x_1; f_2(x) = g(1 - (f_1/g)^2); g(x) = 1 + 9 \sum_{i=2}^n x_i / (n-1)$	$0 \leq x_i \leq 1;$ $i = 1, 2, \dots, n$	连续凹
ZDT3	$f_1(x) = x_1; f_2(x) = g(1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1));$ $g(x) = 1 + 9 \sum_{i=2}^n x_i / (n-1)$	$0 \leq x_i \leq 1;$ $i = 1, 2, \dots, n$	不连续
DTLZ1	$f_1(x) = \frac{1}{2} x_1 x_2 \dots x_{M-1} (1 + g(x_M))$ $f_2(x) = \frac{1}{2} x_1 x_2 \dots (1 - x_{M-1}) (1 + g(x_M))$... $f_{M-1}(x) = \frac{1}{2} x_1 (1 - x_2) (1 + g(x_M))$ $f_M(x) = \frac{1}{2} (1 - x_1) (1 + g(x_M))$ $g(x_M) = x_M + \sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))$	$0 \leq x_i \leq 1;$ $i = 1, 2, \dots, n$	$\sum_{i=1}^M f_i^2 = 0.5$
DTLZ2	$f_1(x) = (1 + g(x_M)) \cos(x_1 \pi/2) \dots \cos(x_{M-1} \pi/2)$ $f_2(x) = (1 + g(x_M)) \cos(x_1 \pi/2) \dots \sin(x_{M-1} \pi/2)$... $f_M(x) = (1 + g(x_M)) \sin(x_1 \pi/2)$ $g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2$	$0 \leq x_i \leq 1;$ $i = 1, 2, \dots, n$	$\sum_{i=1}^M f_i^2 = 1$

文中所有仿真的实验平台为 Intel(R) Core(TM) 2Duo P8600, 2.0GB RAM, 实验环境为 MATLAB 7.12.0. ZDT1、ZDT2、ZDT3 算法的种群大小为 200, DTLZ1、DTLZ2 算法的种群大小为 500, 分别将测试函数 ZDT1~ZDT3 的变量个数分别扩展为 100、300、500、1000. 将 DTLZ1、DTLZ2 的变量扩展为 200、500, 目标维度 M 设为 3. 对应的迭代次数为 500, 分别独立运行 20 次. CCMOPSO 中的变异概率为 0.1, 学习因子 c_1 、 c_2 为 1.495, 惯性权重为 0.729, 这里我们选取分组大小为 50 进行仿真实验.

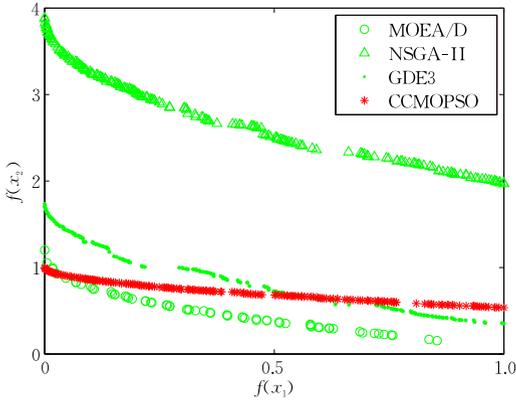
4.1 Pareto 前沿对比分析

4.1.1 二维目标测试函数

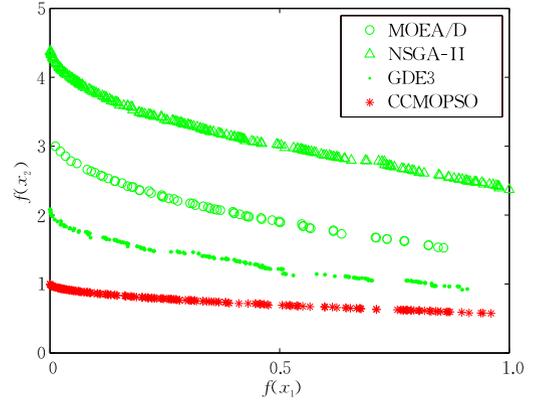
20 次实验得到的统计 Pareto 前沿对比图如图 7~图 9 所示. ZDT1 测试函数的 Pareto 前沿如图 7 所示. 其中图 7(a)、(b)、(c)、(d) 分别为变量个数为 100, 300, 500, 1000 的 NSGA-II、MOEA/D、GDE3、CCMOPSO 的 Pareto 前沿对比图, 其中

CCMOPSO 变量分组为 $S=50$. 同理, ZDT2 测试函数的 Pareto 前沿对比如图 8 所示, ZDT3 测试函数的 Pareto 前沿对比如图 9 所示.

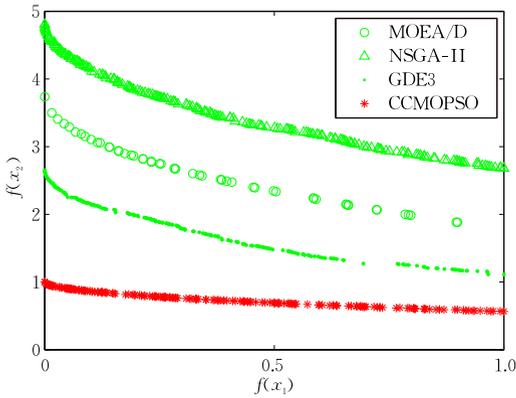
从图 7~图 9 的 Pareto 前沿对比图中可以看出, 通过 CCMOPSO 得到的 Pareto 在变量个数为 100、300、500、1000 时均明显优于 NSGA-II. 从图 7(a)、图 8(a)、图 9(a) 可以看出, 当变量个数为 100 时, 通过 CCMOPSO 所得到的 Pareto 前沿在多样性上优于 MOEA/D, 但是在收敛性上比 MOEA/D 差, 两者各有优劣; CCMOPSO 与 GDE3 相比, ZDT1 和 ZDT2 测试函数的 Pareto 前沿收敛性和多样性略差, ZDT3 的 Pareto 前沿略好. 但是随着变量数目的增加, 如图 7(b)~(d)、图 8(b)~(d) 以及图 9(b)~(d) 所示, MOEA/D 和 GDE3 的 Pareto 前沿明显恶化, 通过 CCMOPSO 得到的 Pareto 前沿在解集的收敛性和多样性方面明显优于 MOEA/D 和 GDE3. 除此之外, 传统 MOPSO 算法在大变量情况下难以得到 Pareto 前沿.



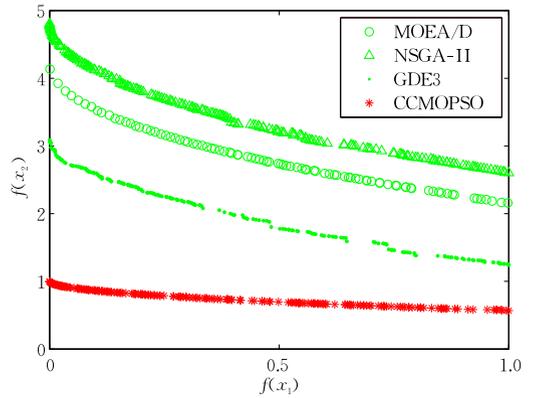
(a) ZDT1(100变量)



(b) ZDT2(300变量)

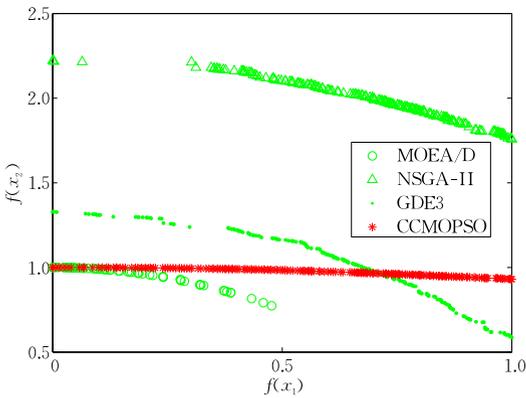


(c) ZDT3(500变量)

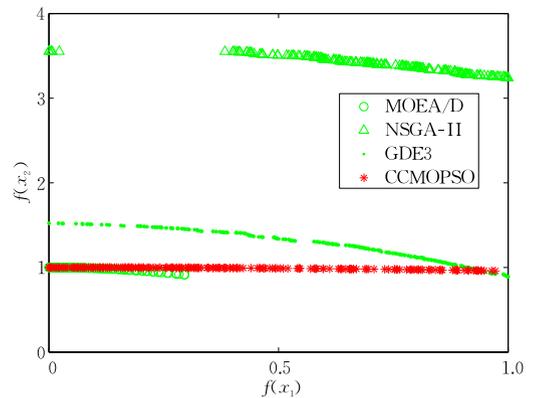


(d) ZDT4(1000变量)

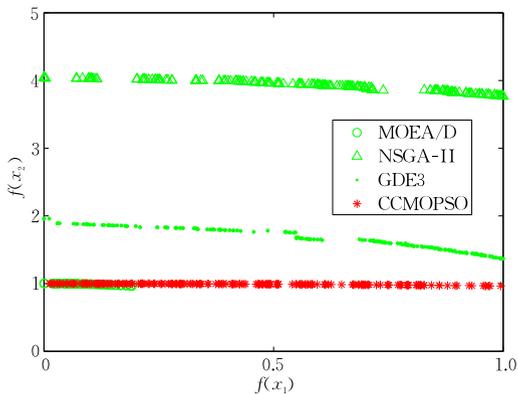
图 7 测试函数 ZDT1 的 Pareto 前沿对比



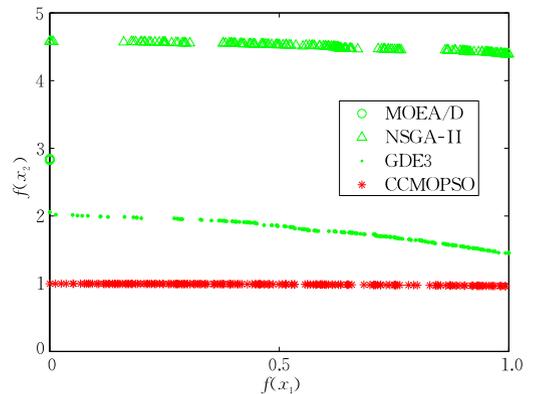
(a) ZDT2(100变量)



(b) ZDT2(300变量)



(c) ZDT2(500变量)



(d) ZDT2(1000变量)

图 8 测试函数 ZDT2 的 Pareto 前沿对比

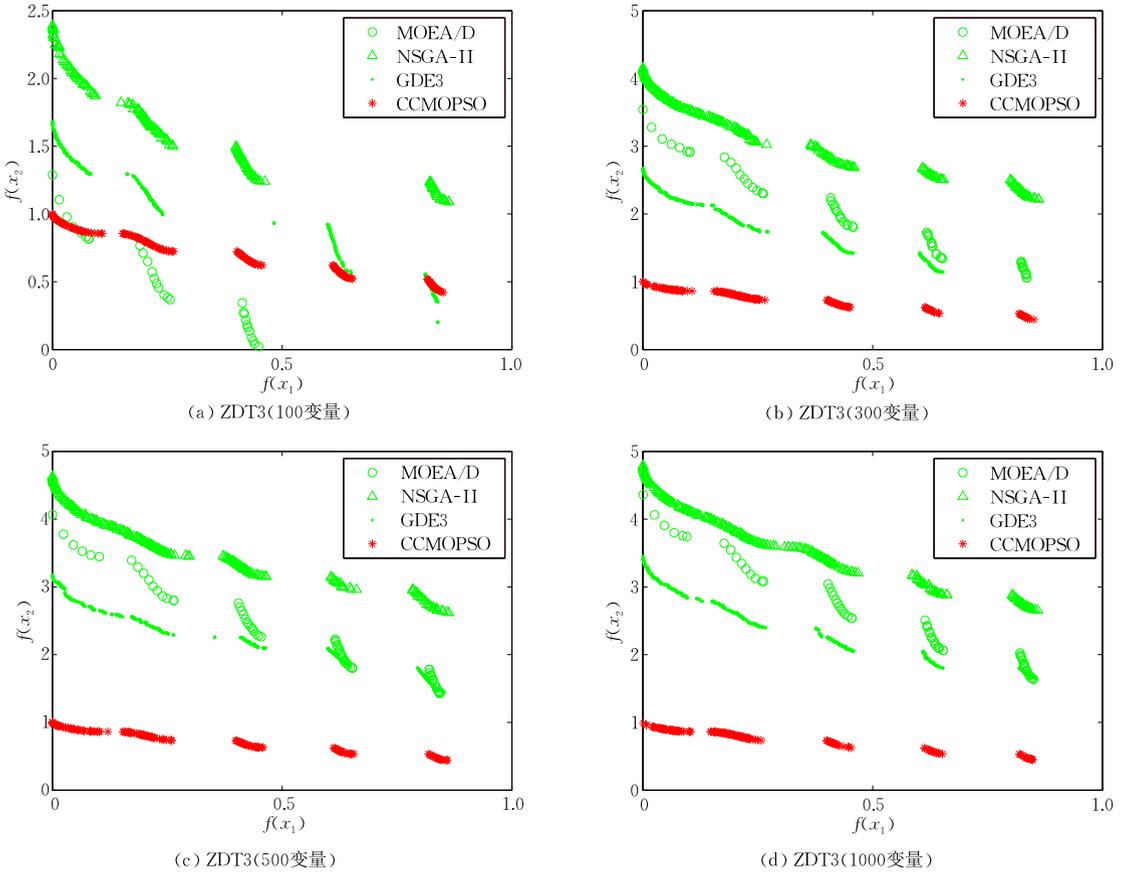


图 9 测试函数 ZDT3 的 Pareto 前沿对比

4.1.2 三维目标测试函数

20 次实验得到的统计 Pareto 前沿对比图如下所示. DTLZ1 测试函数的 Pareto 前沿如图 10 所示. 其中图 10(a)、(b)、(c)、(d) 分别为变量个数为 200 时 NSGA-II、MOEA/D、GDE3、CCMOPSO 所求 Pareto 前沿对比图, 图 10(e)、(f)、(g)、(h) 分别为变量个数为 500 时 NSGA-II、MOEA/D、GDE3、CCMOPSO 所求的 Pareto 前沿对比图, 其中 CCMOPSO 变量分组为 $S=50$. 同理, DTLZ2 测试函数的 Pareto 前沿对比如图 11 所示. 从图 10、图 11 的 Pareto 前沿对比图中可以看出, 当目标维度增加后, 通过 CCMOPSO 得到的 Pareto 前沿的多样性和收敛性在 DTLZ1、DTLZ2 变量个数为 200, 500 时均明显优于 NSGA-II、MOEA/D 以及 GDE3.

综上所述, 从 Pareto 前沿的角度分析可以得出: 在处理大规模变量的多目标优化问题时, CCMOPSO 相对于 MOPSO 有了明显的提升, 而且比算法 NSGA-II 和 MOEA/D 以及 GDE3 的效果更好. 为了更客观的说明该算法的性能, 本文将通过加法二进制 ϵ 指标值、超体积指标值 (HV) 和算法运行时间进一步说明.

4.2 算法性能分析

4.2.1 ϵ 指标值对比

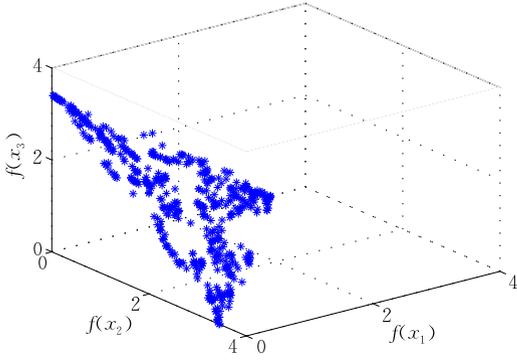
ϵ 指标^[23]是由 Zitzler 等人提出的一种用于评价解集收敛性能的指标, 该指标使用任意给出的两个优化问题的解集 A 和 B 来判断解集的收敛性, 其中

$$I_A = I_{\epsilon+}(A, B) = \text{Inf}_{\epsilon \in \mathfrak{R}} \{ \forall z^2 \in B, \exists z^1 \in A: z^1 <_{\epsilon+} z^2 \},$$

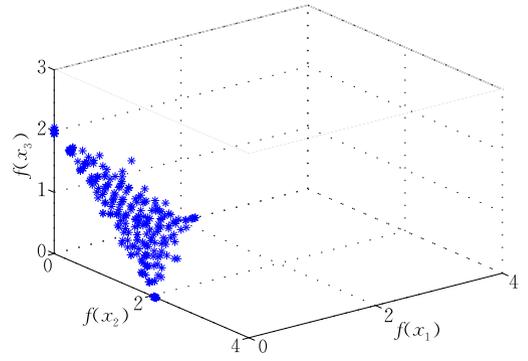
$$I_B = I_{\epsilon+}(B, A) = \text{Inf}_{\epsilon \in \mathfrak{R}} \{ \forall z^2 \in A, \exists z^1 \in B: z^1 <_{\epsilon+} z^2 \}.$$

以最小化为例, 以上公式中, 对于 $\epsilon > 0$ 当且仅当 $\forall 1 \leq i \leq n$ 时, 有 $z_i^1 \leq \epsilon + z_i^2$, 则 $z^1 <_{\epsilon+} z^2$ ($z^1 \in$ 占优 z^2). 对于输出对 $(I_A \leq 0, I_B > 0)$, 表示 A 严格优于 B ; $(I_A > 0, I_B > 0)$ 表示 A 与 B 无法相互比较, 但如果 $I_A < I_B$, 则可以判断 A 弱优于 B ; 同理可得, 如果有 $(I_A < I_B \leq 0)$, 则可以认为 A 弱优于 B .

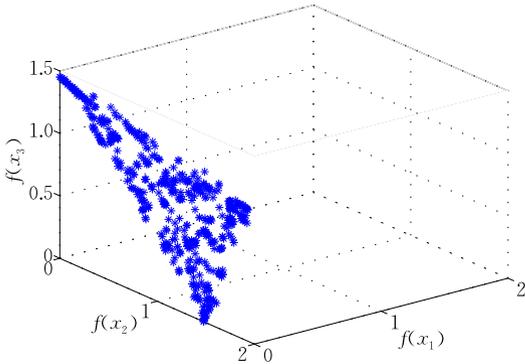
在对比实验中, 本文用字母 C 、 N 、 M 和 G 来分别表示 4 种算法 CCMOPSO、NSGA-II、MOEA/D、GDE3 所产生的解集. 加法二进制 ϵ 指标值如表 2, 在 CCMOPSO 与 NSGA-II 的对比中, 所有测试函数的 $I(C, N) < 0$, 且 $I(N, C) > 0$, 这说明在所有情况下, CCMOPSO 所求解集均严格优于 NSGA-II. 在 CCMOPSO 与 MOEA/D 的对比中, 当 ZDT1、



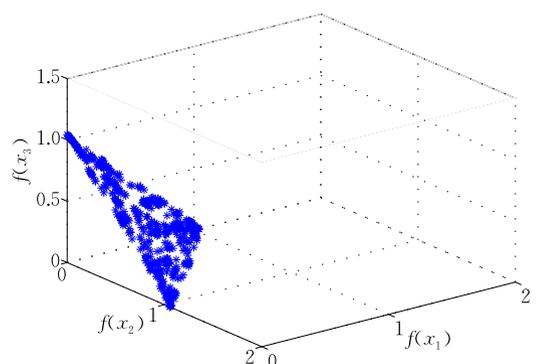
(a) DTLZ1(200变量) USING NSGA-II



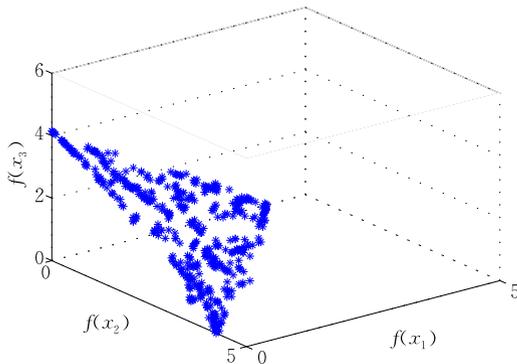
(b) DTLZ1(200变量) USING MOEA/D



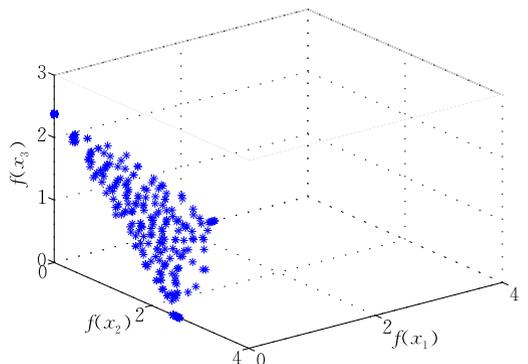
(c) DTLZ1(200变量) USING GDE3



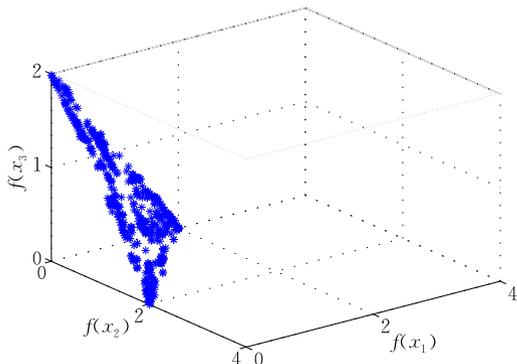
(d) DTLZ1(200变量) USING CCMOPSO



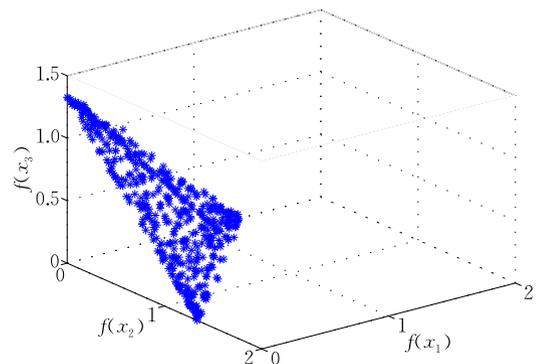
(e) DTLZ1(500变量) USING NSGA-II



(f) DTLZ1(500变量) USING MOEA/D

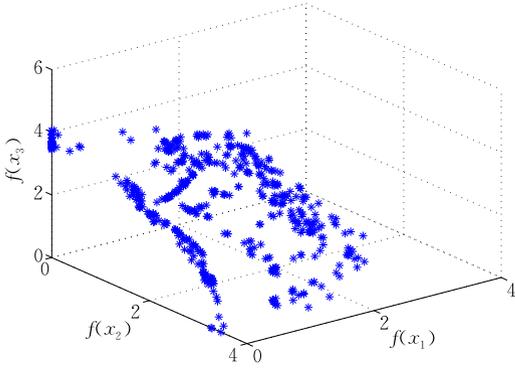


(g) DTLZ1(500变量) USING GDE3

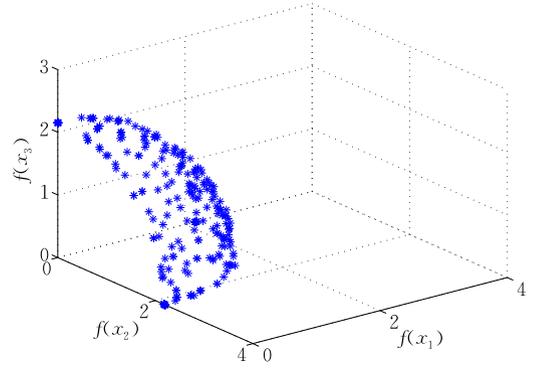


(h) DTLZ1(500变量) USING CCMOPSO

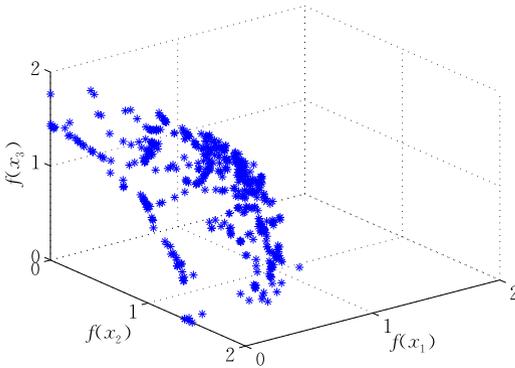
图 10 测试函数 DTLZ1 的 Pareto 前沿对比



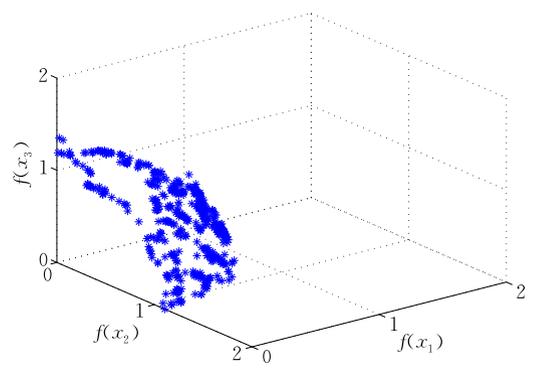
(a) DTLZ2(200变量) USING NSGA-II



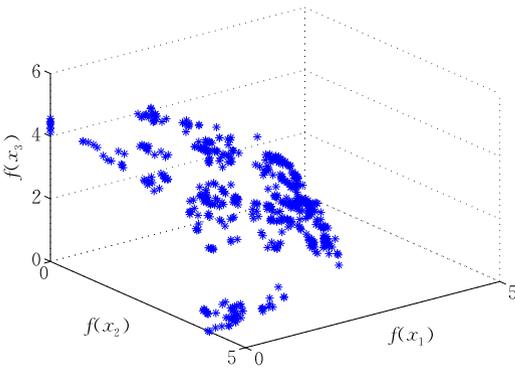
(b) DTLZ2(200变量) USING MOEA/D



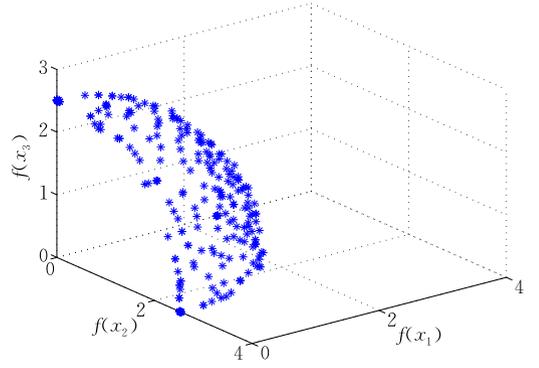
(c) DTLZ2(200变量) USING GDE3



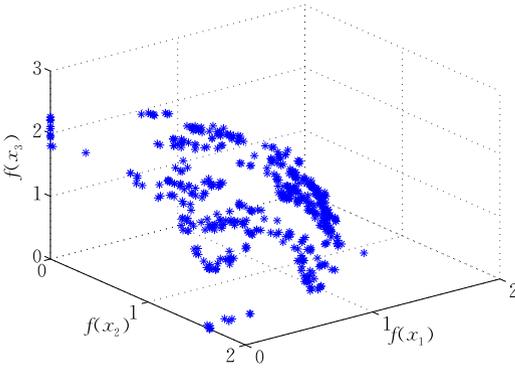
(d) DTLZ2(200变量) USING CCMOPSO



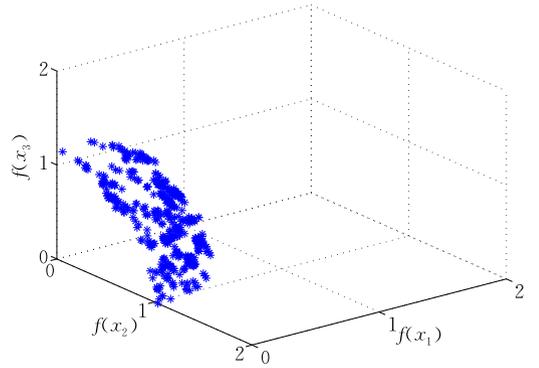
(e) DTLZ2(500变量) USING NSGA-II



(f) DTLZ2(500变量) USING MOEA/D



(g) DTLZ2(500变量) USING GDE3



(h) DTLZ2(500变量) USING CCMOPSO

图 11 测试函数 DTLZ2 的 Pareto 前沿对比

ZDT2、ZDT3 的变量个数为 100 时, $I(C, M) > 0$, $I(M, C) < 0$, 这说明当变量个数为 100 时, MOEA/D 优于 CCMOPSO. 但是当 ZDT1、ZDT3 变量个数为 300、500、1000 时, $I(C, M) < 0$, $I(M, C) > 0$, 这说明 CCMOPSO 严格优于 MOEA/D, 另外, 当 ZDT2 变量个数为 300、500 时, $I(C, M) < 0$, $I(M, C) < 0$, $I(C, M) < I(M, C)$, 这说明当 ZDT2 变量个数增加

时, CCMOPSO 优于 MOEA/D. 当目标维度为 3 时, $I(C, M) < 0$, 且 $I(M, C) > 0$, 这说明在 DTLZ1 和 DTLZ2 问题下, CCMOPSO 所求解集均严格优于 MOEA/D. 同理, 在 CCMOPSO 与 GDE3 的对比中, ZDT1、ZDT3 变量为 100 和 ZDT2 变量为 100、300 的情况下, CCMOPSO 弱优于 GDE3, 其他情况均严格优于 GDE3.

表 2 加法二进制指标值

ϵ 指标值	CCMOPSO 与 NSGA-II		CCMOPSO 与 MOEA/D		CCMOPSO 与 GDE3		
	$I(C, N)$	$I(N, C)$	$I(C, M)$	$I(M, C)$	$I(C, G)$	$I(G, C)$	
ZDT1	100	-0.990	0.998	0.076	-0.293	-0.160	-0.114
	300	-1.000	1.370	-0.709	0.694	-0.393	0.292
	500	-1.000	1.678	-0.889	0.896	-0.512	0.436
	1000	-1.000	1.601	-0.998	1.157	-0.665	0.537
ZDT2	100	-0.866	0.761	0.103	-0.193	-0.257	-0.143
	300	-1.000	2.237	-0.151	-0.081	-0.407	-0.014
	500	-0.987	2.763	-0.101	-0.032	-0.633	0.395
	1000	-0.999	3.389	-0.027	1.828	-0.708	0.461
ZDT3	100	-0.513	0.391	0.215	-0.399	-0.209	-0.021
	300	-0.861	1.217	-0.621	0.406	-0.532	0.406
	500	-0.862	1.618	-0.813	0.632	-0.795	0.649
	1000	-0.851	1.692	-0.825	0.766	-0.815	0.782
DTLZ1	200	-0.822	0.820	-0.329	0.385	-0.141	0.159
	500	-1.002	1.044	-0.321	0.376	-0.220	0.235
DTLZ2	200	-1.461	1.259	-0.489	0.785	-0.314	0.209
	500	-2.010	1.881	-0.530	1.126	-0.631	0.512

综上所述, 由加法二进制 ϵ 指标值的对比结果可以得出: CCMOPSO 在大规模变量优化问题上产生的解集要优于 NSGA-II、MOEA/D、GDE3 产生的解集.

4.2.2 超体积指标值(HV)对比

超体积是指被非支配解集覆盖的目标空间区域大小. 超体积度量方法, 也被称为 Lebesgue 测度, 在理论上具有良好的数学性质, 即在所有的一元测度中, 它是一个能够判定非支配解集 X 不比另一个非支配解集 Y 差的方法^[24].

对于一个前沿上的解集 P_{front} , 超体积的计算公式为

$$HV(P_{\text{front}}) = \Delta \left(\bigcup_{p \in P_{\text{front}}} \{x \mid p \prec x \prec x_{\text{ref}}\} \right).$$

上式中, Δ 即 Lebesgue 的测度, x_{ref} 为参照点. 对于两目标优化问题, HV 是坐标区域的面积; 对于三目标优化问题, HV 是三维空间构成的体积; 对于大于三个目标的优化问题, HV 表示为超体积值. HV 的值越大, 解集的质量越高.

为了同时评估算法的收敛性和多样性, 本文采用超体积指标 (HV) 对 CCMOPSO、NSGA-II、

MOEA/D 以及 GDE3 所求解集进行对比. 算法分别独立运行 20 次, 得到的超体积对比结果如图 12~图 15 所示. 由于 MOPSO 算法难以产生 Pareto 前沿, 故认为超体积指标为无穷小.

图 12 为分别对测试函数 ZDT1 在 100、300、500、1000 变量大小情况下, 用 NSGA-II、MOEA/D、GDE3 和 CCMOPSO 独立运行 20 次, 得到的超体积指标值对比图. 与 ZDT1 相同, 图 13 和图 14 分别是测试函数 ZDT2 和 ZDT3 的超体积指标值对比图. 图 15 为分别对测试函数 DTLZ1 和 DTLZ2 在 200、500 变量大小情况下得到的超体积指标值对比图. 表 3 是 20 次仿真结果所产生解集的超体积指标平均值. 对比结果显示: 通过变量随机分解策略, CCMOPSO 算法所得解集的超体积指标值均优于 NSGA-II. 当变量个数为 100 时, MOEA/D 求得的解集的超体积指标略优于 CCMOPSO, GDE3 求得的解集的超体积指标略差于 CCMOPSO, 但随着变量增加到 300、500、1000 时, MOEA/D 所得解集的超体积指标值明显减小, 而 CCMOPSO 所求解集的超体积指标值仍保持较大值, 明显优于 MOEA/D 和 GDE3 所求解集.

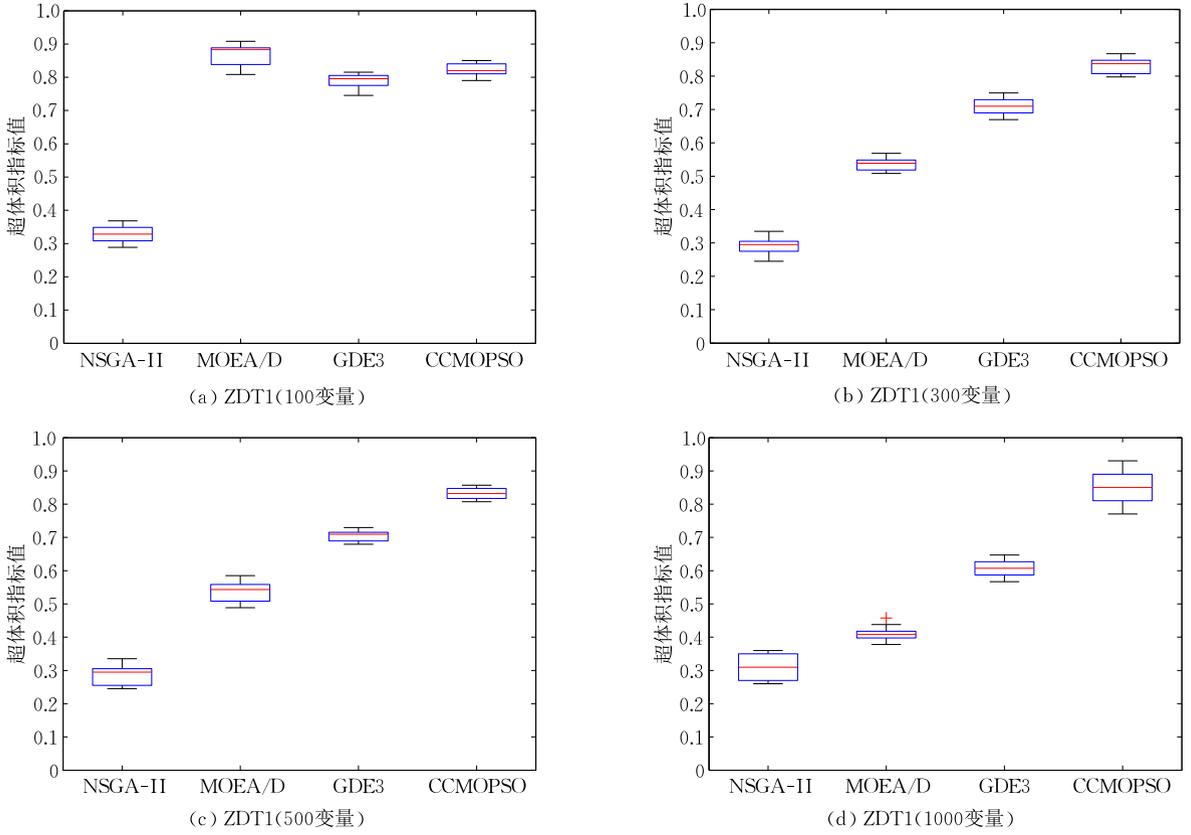


图 12 测试函数 ZDT1 用 NSGA-II、MOEA/D、GDE3 和 CCMOPSO 算法得到的超体积指标对比

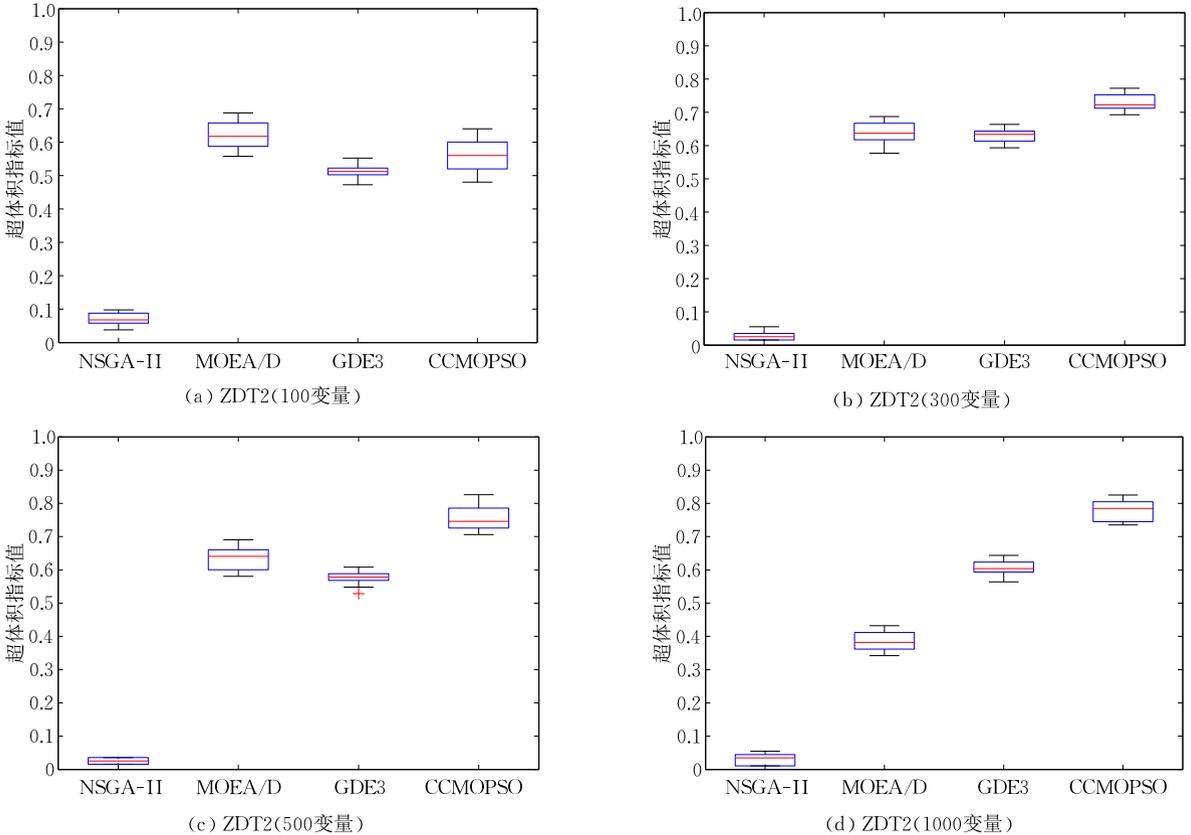
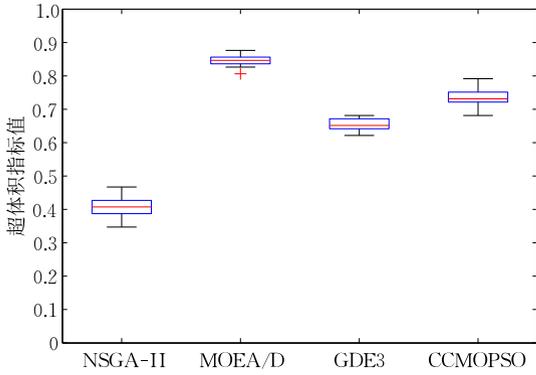
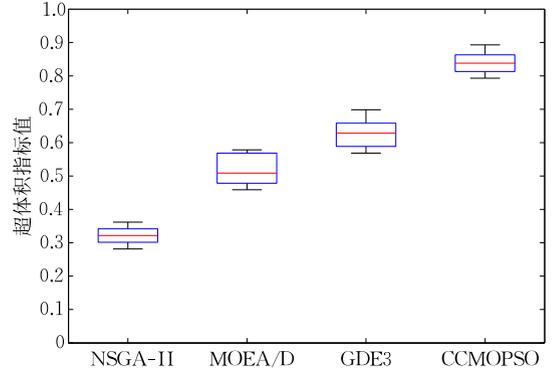


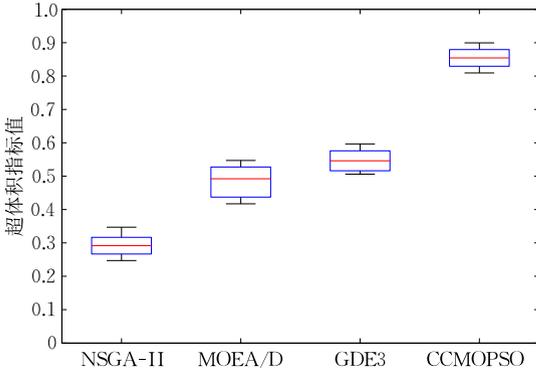
图 13 测试函数 ZDT2 用 NSGA-II、MOEA/D、GDE3 和 CCMOPSO 算法得到的超体积指标对比



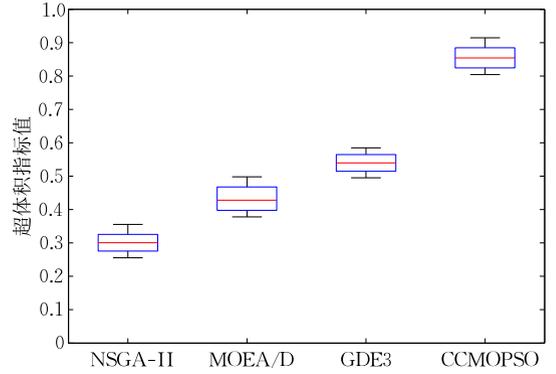
(a) ZDT3(100变量)



(b) ZDT3(300变量)

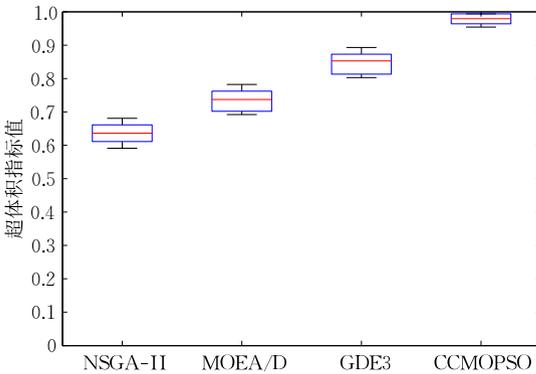


(c) ZDT3(500变量)

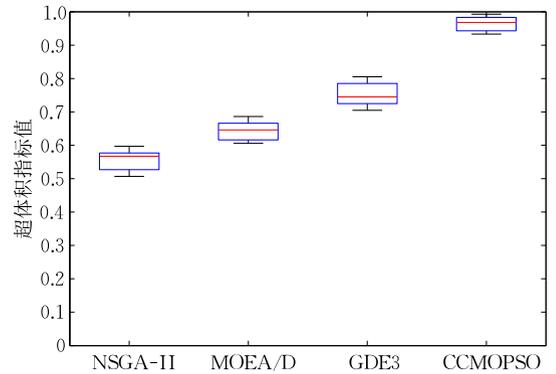


(d) ZDT3(1000变量)

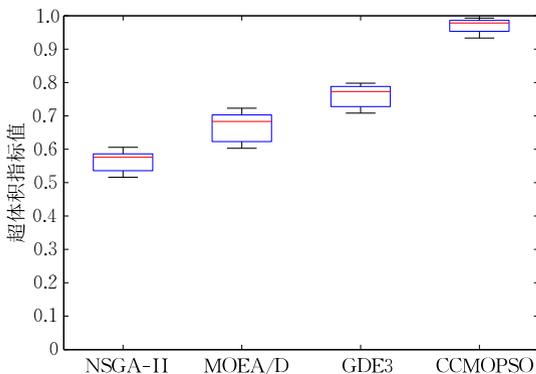
图 14 测试函数 ZDT3 用 NSGA-II、MOEA/D、GDE3 和 CCMOPSO 算法得到的超体积指标对比



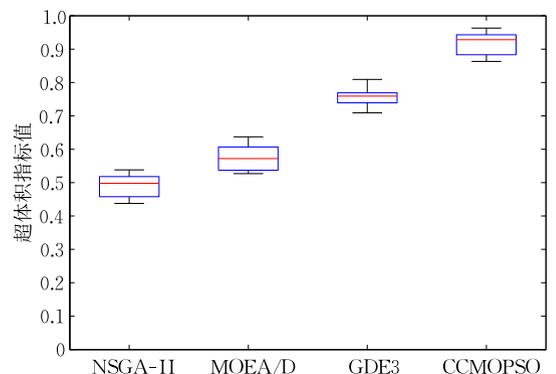
(a) DTLZ1(200变量)



(b) DTLZ1(500变量)



(c) DTLZ2(200变量)



(d) DTLZ2(500变量)

图 15 测试函数 DTLZ1/DTLZ2 用 NSGA-II、MOEA/D、GDE3 和 CCMOPSO 算法得到的超体积指标对比

表 3 超体积指标平均值

超体积指标平均值	CCMOPSO	NSGA-II	MOEA/D	GDE3	
ZDT1	100	0.820	0.328	0.888	0.795
	300	0.837	0.295	0.539	0.709
	500	0.852	0.298	0.488	0.671
	1000	0.850	0.309	0.607	0.407
ZDT2	100	0.560	0.068	0.618	0.512
	300	0.723	0.025	0.637	0.633
	500	0.746	0.026	0.661	0.578
	1000	0.850	0.310	0.408	0.603
ZDT3	100	0.731	0.407	0.886	0.651
	300	0.843	0.322	0.529	0.628
	500	0.859	0.297	0.477	0.546
	1000	0.864	0.305	0.544	0.437
DTLZ1	200	0.994	0.681	0.782	0.853
	500	0.993	0.567	0.656	0.775
DTLZ2	200	0.986	0.576	0.683	0.768
	500	0.923	0.498	0.587	0.759

综上所述,由超体积指标值的对比结果可以得出,NSGA-II、MOEA/D、GDE3 大规模变量全局优化问题上产生的解集比 CCMOPSO 所产生的解集差。

4.2.3 算法运行时间

衡量一个算法优劣的重要指标为算法的运行效率。为了比较 4 种算法的运行效率,本文统计了 20 次 4 种算法在变量为 100、300、500、1000 时,ZDT1、ZDT2、ZDT3、DTLZ1、DTLZ2 测试函数运行至 500 代所耗费的时间的平均值(单位:s),如表 4 所示。

表 4 算法运行时间

算法运行时间/s	CCMOPSO	NSGA-II	MOEA/D	GDE3	
ZDT1	100	55.7	87.3	60.9	58.4
	300	97.6	157.1	113.2	108.5
	500	108.3	212.6	151.7	134.9
	1000	124.8	405.2	284.6	237.1
ZDT2	100	53.9	83.4	58.6	60.5
	300	90.5	146.8	102.8	98.3
	500	104.2	210.9	156.1	142.7
	1000	118.3	391.5	274.6	251.8
ZDT3	100	67.1	93.2	78.9	75.9
	300	102.8	159.5	121.7	112.5
	500	108.4	230.1	179.2	164.0
	1000	132.9	447.2	307.4	287.1
DTLZ1	200	164.2	331.7	285.4	243.9
	500	207.5	464.2	387.1	322.0
DTLZ2	200	177.4	368.6	327.2	286.2
	500	231.1	516.4	439.5	364.5

综上所述,NSGA-II 算法的运行时间大于其他 3 种算法的运行时间。随着优化问题中决策变量数目的增加,NSGA-II、MOEA/D、GDE3 的运行时间呈递增趋势,而 CCMOPSO 的运行时间相对保持平稳,决策变量数目的增加对算法的运行效率影响较小。其原因是 CCMOPSO 采用了变量分解策略,当

变量数目增加时,平均每一组的变量增加数目相对要少,因此非支配排序所消耗的时间影响较小。综上所述,CCMOPSO 相比 NSGA-II、MOEA/D 和 GDE3,算法复杂度明显降低。

4.2.4 对比实验结果

从 Pareto 前沿、加法二进制 ϵ 指标、超体积指标值(HV)、算法运行时间的对比结果表明:引入变量随机分解策略的 CCMOPSO 在求解含有大规模变量的优化问题的求解精度以及算法的运行效率整体上均优于 NSGA-II、MOEA/D、GDE3,随着变量维度的增加,优势更加明显。MOPSO 在大规模变量优化问题的求解上难以收敛到最优前沿,且求解精度差于 CCMOPSO。综上,CCMOPSO 算法在解决大规模变量的多目标函数中,变量维度越高,它比多目标算法 MOPSO、NSGA-II、MOEA/D 以及 GDE3 具有越好的多样性与收敛性,同时使得计算复杂度明显降低。

5 结 论

本文提出的变量随机分解策略,将合作协同进化框架融合于多目标粒子群优化算法(MOPSO),提出了基于变量分解的多目标粒子群优化算法(CCMOPSO)。通过对标准测试函数 ZDT1、ZDT2、ZDT3、DTLZ1、DTLZ2 的变量个数扩展后进行的仿真实验,与多目标进化算法 NSGA-II、MOEA/D、GDE3 进行对比,用加法二进制 ϵ 指标、超体积指标(HV)和算法运行时间对算法性能进行分析,实验结果表明:CCMOPSO 相比 NSGA-II、MOEA/D 和 GDE3,在求解精度和运行效率上都有显著提升。但实验表明,当目标函数为多模态时,还是会存在传统粒子群优化算法易陷入局部最优的缺陷,该算法在求解此类问题上仍有待进一步深入研究。

参 考 文 献

- [1] Potter M A, De Jong K A. A cooperative co-evolutionary approach to function optimization//Davidor Y, Schwefel H-P, Männer R eds. Parallel Problem Solving from Nature, PPSN III. LNCS 866. Springer Berlin Heidelberg, Germany, 1994: 249-257
- [2] Keerativuttitumrong N, Chaiyaratana N, Varavithya V. Multi-objective co-operative co-evolutionary genetic algorithm //Guervós J J M, Adamidis P, Beyer H-G, et al, eds. Parallel Problem Solving from Nature, PPSN VII. LNCS 2439. Springer Berlin Heidelberg, Germany, 2002: 288-297

- [3] Iorio A W, Li X. A cooperative co-evolutionary multi-objective algorithm using non-dominated sorting//Deb K ed. Genetic and Evolutionary Computation—GECCO 2004. LNCS 3102. Seattle, USA, 2004: 537-548
- [4] Tan K C, Yang Y J, Goh C K. A distributed cooperative co-evolutionary algorithm for multi-objective optimization. IEEE Transactions on Evolutionary Computation, 2006, 10(5): 527-549
- [5] Mahdavi S, Shiri M. Cooperative co-evolution with a new decomposition method for large-scale optimization//Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC). Beijing, China, 2014: 1285-1292
- [6] Ye S, Dai G. A hybrid adaptive co-evolutionary differential evolution algorithm for large-scale optimization//Proceedings of the 2014 IEEE Congress on Evolutionary Computation. Beijing, China, 2014: 1277-1284
- [7] Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative co-evolution. Information Sciences, 2008, 178(15): 2985-2999
- [8] Omidvar M N, Li X, Yang Z, et al. Cooperative co-evolution for large scale optimization through more frequent random grouping//Proceedings of the 2010 IEEE Congress on Evolutionary Computation. Barcelona, Spain, 2010: 1-8
- [9] Li X, Yao X. Cooperatively coevolving particle swarms for large scale optimization. IEEE Transactions on Evolutionary Computation, 2012, 16(2): 210-224
- [10] Antonio L M, Coello Coello C A. Use of cooperative co-evolution for solving large scale multi-objective optimization problems//Proceedings of the 2013 IEEE Congress on Evolutionary Computation. Cancun, Mexico, 2013: 2758-2765
- [11] Ray T, Yao X. A cooperative co-evolutionary algorithm with correlation based adaptive variable partitioning//Proceedings of the 2009 IEEE Congress on Evolutionary Computation. Trondheim, Norway, 2009: 983-989
- [12] Chen W, Weise T, Yang Z, et al. Large-scale global optimization using cooperative co-evolution with variable interaction learning//Schaefer R, Cotta C, Kołodziej J, Rudolph G eds. Parallel Problem Solving from Nature, PPSN XI. LNCS 6239. Springer Berlin Heidelberg, Germany, 2010: 300-309
- [13] Sun L, Yoshida S, Cheng X, et al. A cooperative particle swarm optimizer with statistical variable interdependence learning. Information Sciences, 2012, 186(1): 20-39
- [14] Omidvar M N, Li X, Mei Y, et al. Cooperative co-evolution with differential grouping for large scale optimization. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 378-393
- [15] Fan J, Wang J, Han M. Cooperative co-evolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods. IEEE Transactions on Fuzzy Systems, 2014, 22(4): 829-839
- [16] Smith J, Fogarty T C. An adaptive poly-parental recombination strategy//Fogarty T C ed. Evolutionary Computing. LNCS 993. Springer Berlin Heidelberg, Germany, 1995: 48-61
- [17] Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. Chichester: John Wiley & Sons, 2001
- [18] Gong Mao-Guo, Jiao Li-Cheng, Yang Dong-Dong, et al. Research on evolutionary multi-objective optimization algorithms. Journal of Software, 2009, 20(2): 271-289 (in Chinese)
(公茂果, 焦李成, 杨咚咚等. 进化多目标优化算法研究. 软件学报, 2009, 20(2): 271-289)
- [19] Kennedy J, Eberhart R. Particle swarm optimization//Proceedings of the 1995 IEEE International Conference on Neural Networks Proceedings. Piscataway, USA, 1995: 1942-1948
- [20] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197
- [21] Zhang Q, Li H. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731
- [22] Kukkonen S, Lampinen J. GDE3: The third evolution step of generalized differential evolution//Proceedings of the 2006 IEEE Congress on Evolutionary Computation. California, USA, 2006, 1: 443-450
- [23] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multi-objective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation, 2003, 7(2): 117-132
- [24] Beume N, Fonseca C M. On the complexity of computing the hyper-volume indicator. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 1075-1082



QIU Fei-Yue, born in 1965, Ph. D., professor, Ph.D. supervisor. His research interests include intelligent control, optimization theory and method, learning science and media technology.

MO Lei-Ping, born in 1990, M. S. candidate. His research interests include information processing and decision optimization.

JIANG Bo, born in 1985, Ph. D., associate professor. His research interest is multi-objective evolutionary algorithms and application.

WANG Li-Ping, born in 1964, Ph. D., professor, Ph. D. supervisor. Her current research interests include information processing, decision optimization, business intelligence.

Background

The multi-objective optimization of large scale variable optimization has been applied in the practical work and greatly changed our lives. Nowadays solving the multi-objective optimization with hundred and thousand variables has become one of the hot research topics. This paper proposes Multi-Objective Particle Swarm Optimization Algorithm Using Large Scale Variable Decomposition. The algorithm proposed is of better diversity, convergence and easiness in multi-objective function with large scale variable than MOPSO, NSGA-II, MOEA/D and GDE3.

This research is supported by the National Natural Science Foundation of China under Grant Nos. 61472366,

61379077 and 61503340, and the Zhejiang Provincial Natural Science Foundation of China under Grant Nos. LZ13F020002, LY13F030010, and LY17F020022. Our project aims to provide better performance in large scale variable optimization. Our group has been working on the preferences-based many objective optimization using bipolar preferences dominance model. New algorithms have been applied into practical work and many high-quality research papers were published in key journals. This paper focuses on researching a new algorithm based on large scale variable decomposition by integration cooperative co-evolution framework.