

Particle Swarm Optimization Algorithm Based on Membrane System

Nan Song, Taowei Chen, Yimin Yu, Kun Zhao

Information Institute, Yunnan University of Finance and Economics, Kunming Yunnan
Email: 348491433@qq.com

Received: Jul. 5th, 2019; accepted: Jul. 19th, 2019; published: Jul. 26th, 2019

Abstract

Due to the particle swarm algorithm has the characteristics of easy understanding and implementation, it is widely concerned in the field of science and engineering, but its search strategy is relatively simple, making it difficult for the algorithm to obtain the Pareto frontier and easy to fall into local optimal and infinite iteration. There are still some deficiencies in the aspect of global search and convergence. Inspired by the P system theory, this paper proposes a multi-objective particle swarm optimization algorithm based on the membrane system framework to solve the multi-objective optimization problem. According to the hierarchical structure, objects and rules of P system, the particle swarm algorithm is used to implement the parallel search strategy in the base film. The non-dominated sorting and congestion distance mechanism are used in the surface film to improve the convergence speed of the algorithm, and the evolution rule is used to maintain the diversity of solution. The simulation experiment uses the KUR, ZDT series and DTLZ series test functions to test the MOPSO algorithm and compare it with other multi-objective optimization algorithms, including MOPSO, dmOPSO, SMPSO, MMOPSO, MOEA/D, SPEA2, PESA2, NSGAI. The experimental results show that the new algorithm can guarantee the diversity of the solution, is fast convergence and close to the real Pareto frontier.

Keywords

Particle Swarm Optimization, Membrane System, Multi-Objective Optimization, Pareto Frontier, Non-Dominated Sorting

基于膜系统的粒子群算法

宋楠, 陈韬伟, 余益民, 赵昆

云南财经大学, 信息学院, 云南 昆明
Email: 348491433@qq.com

收稿日期: 2019年7月5日; 录用日期: 2019年7月19日; 发布日期: 2019年7月26日

摘要

由于粒子群算法具有易于理解实现等特点,受到科学与工程领域的广泛关注,但其搜索策略较为单一,使得算法很难获得Pareto前沿且容易陷入局部最优和无限迭代,在全局搜索和收敛性方面还有一定的不足。受P系统理论启发,本文提出了一种基于膜系统框架的多目标粒子群优化算法用于解决多目标优化问题。根据P系统的层次结构、对象和规则,在基本膜中采用粒子群算法实现并行搜寻策略,在表层膜中采用非支配排序和拥挤距离机制来提高算法收敛速度,并通过进化规则以保持解的多样性。仿真实验采用KUR、ZDT系列和DTLZ系列测试函数对MPSO算法进行测试,并与其他多目标优化算法进行对比,包括MOPSO、dMOPSO、SMPSO、MMOPSO、MOEA/D、SPEA2、PESA2、NSGAI。实验结果表明:新算法能够保证解的多样性,快速收敛并接近于真实的Pareto前沿。

关键词

粒子群算法, 膜系统, 多目标优化, Pareto前沿, 非支配排序

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

现实中许多工程和科学问题都存在多个彼此相互冲突的问题,如:相同材料的强度和重量,商品的成本和质量。如何求得这些问题的最优解是工程和学术领域的焦点问题。和简单的单目标优化问题不同,多目标问题最大特点是在提高某些目标的性能时,会降低其它某些目标性能,同时求得所有目标的最优解是不可能的。多目标问题就是不同目标间进行协调,使每个目标性能尽可能达到最优。但这会导致数量较多的最优解,甚至达到无穷多个[1]。

解决多目标优化问题的原始方法通常会将问题转换成单目标优化,采用数学规划方式进行求解,每次只能得到一种权重情况下的解。但对于多目标优化的目标函数和约束函数可能是非线性、不可微或不连续的,故采用传统的数学规划方法效率较低,且对于权重和目标给定的次序较敏感[2]。由于粒子群算法概念简单、易于实现,越来越多的学者尝试运用将粒子群算法与其它优化算法融合解决多目标优化算法问题。Li最早将粒子群优化(Particle Swarm Optimization, PSO)与NSGA2结合用于解决多目标优化问题[3]; Coello等人在PSO中采用自适应网格的方法,提出了经典的MOPSO算法[4];张提出了一个基于粒子群优化膜计算理论的云计算资源调度算法[5];Liu等提出了将遗传机制引入到膜系统的膜算法,该求解ZDT优化问题表现出了较好的求解性能[6];Parsopoulos和Vrahatis提出了一种目标聚合的多目标粒子群算法,采用固定的和自适应的权重组合将多目标优化问题转换为单目标优化问题,然后采用基于粒子群算法进行求解,但每运行一次算法只能获得一个Pareto最优解[7];Hu和Eberhart提出了一种动态邻近法来求解多目标优化算法,采用动态邻近策略来为粒子选取最优经验,但一次只能对一个目标寻优,其本质还是用一维的方法来处理多目标[8];Dou在膜系统上提出了一种新的粒子群优化算法用于膜系统的特征选择,利用P系统的层次结构和消息机制,将粒子群算法作为子算法引入所有基本膜中[9]。

与其它多目标优化算法相比,MOPSO有着收敛速度快的特点,但其在解均匀性和并行性方面表现较差。由于膜结构具有良好的并行性,并且结构简单,易于与其他进化算法相融合等特点,部分学者尝

试将膜计算理论与粒子群算法相结合,但其基本应用于单目标问题,无法直接应用于多目标优化问题。本文将膜计算理论与粒子群算法相结合,提出膜粒子群算法,充分利用膜的分裂及规则,在基本膜中采用粒子群算法实现并行搜寻策略,在表层膜中采用非支配排序和拥挤距离机制来提高算法收敛速度,并通过进化规则以保持解的多样性。

为验证 MPSO 算法的有效性和可行性,在仿真实验中采用 KUR、ZDT 系列与 DTLZ 系列目标函数进行测试。实验结果表明:该算法所得的非支配解集不易陷入局部最优,快速收敛并接近于真实的 Pareto 前沿,在解集的多样性与算法的收敛性和准确性等方面优于或部分优于其他算法。

2. 相关概念

2.1. 多目标优化问题

自然界中存在着大量多个目标彼此冲突的优化问题,这类问题被称为多目标优化问题。利用目标函数可以从数学角度对其算法优化性能进行衡量,因此,一般多目标优化问题(MOP)可被描述如下:

$$\begin{aligned} \min \quad & f(x) = [f_1(x), f_2(x), \dots, f_M(x)] \\ \text{st.} \quad & g(x) = [g_1(x), g_2(x), \dots, g_J(x)] \geq 0 \\ & h(x) = [h_1(x), h_2(x), \dots, h_K(x)] = 0 \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, \dots, N \end{aligned} \quad (1)$$

(1)式中 $X = (x_1, x_2, \dots, x_N)^T$ 为 N 维决策向量,且取值在 $x_i^{(L)} \leq x_i \leq x_i^{(U)}$ 之间; $f(x)$ 定义了 M 个目标函数;(1)中定义包含了 J 个不等式约束 $g(x)$ 与 K 个等式约束 $h(x)$,约束函数 $g(x)$ 与 $h(x)$ 共同确定决策向量中 X 的可行域。

定义 1 可行解:对任意解向量 $x \in \theta$,若满足(1)式中的约束条件 $g(x)$ 与 $h(x)$,则称为可行解,否则为不可行解。

定义 2 Pareto 支配:对于决策变量 $a = [a_1, a_2, \dots, a_n]^T$ 和 $b = [b_1, b_2, \dots, b_n]^T$,当且仅当 $\forall i \in \{1, 2, \dots, m\}, \exists j \in \{1, 2, \dots, m\}$,使得 $f_i(a) \leq f_i(b)$ 成立,称 a 支配 b ,记为 $b \prec a$ 。若它们之间不存在 Pareto 支配关系,则称它们为非支配。

定义 3 Pareto 最优解:如果满足 $X^* = \{x \in R^n \mid \neg \exists x' \prec x\}$,则 X^* 为 Pareto 最优解集或非支配解集,也可写为 PS^* , R^n 为解集的决策空间。

定义 4 Pareto 前沿: Pareto 最优解集对应应在目标函数空间上的映射集合,即 $PF^* = \{f(x^*) \mid x^* \in PS^*\}$,映射后的集合称为 Pareto 前沿。

2.2. 粒子群算法

粒子群优化算法最早源于模拟生物群体的社会行为,是一类基于群体智能的随机优化算法,由 Kenney 和 Eberhart 于 1995 年提出[10]。算法将模拟鸟群中每一只鸟称为一个“粒子”,它们以一定的速度和方向飞行,根据自身和同伴的飞行经验动态调整,鸟群所寻找的食物就是我们希望求得问题的最优解。鸟群中的鸟(即粒子)第 $t+1$ 次飞行的速度和飞行后的位置根据如下公式确定:

$$\begin{aligned} V_{t+1} &= \omega * V_t + c_1 * \text{rand}() * (pBest - X_t) + c_2 * \text{Rand}() * (gBest - X_t) \\ X_{t+1} &= X_t + V_{t+1} \end{aligned} \quad (2)$$

V_t 和 V_{t+1} 分别是第 t 和第 $t+1$ 次飞行的速度; X_t 和 X_{t+1} 分别是经过第 t 和 $t+1$ 次飞行后粒子落在的位置; ω 为惯性权重(inertia weight),描述粒子的惯性对当前速度的影响,其大小用来平衡局部和全局

的搜索能力； c_1 和 c_2 为个体最优和全局最优的学习因子； $rand()$ 和 $Rand()$ 是 $[0,1]$ 上相互独立的随机数； $pBest$ 是该粒子历次飞行中“最好”的位置； $gBest$ 是种群中“最好”的粒子位置。经过多次迭代，种群中的粒子会逐渐向“更好”的位置飞行，最终求得最优解。

2.3. 膜系统

作为自然计算的新分支，膜计算(或称为P系统)由Păun于1998年在芬兰图尔库计算机中心的研究报告中提出，正式论文于2000年见刊发表[11]。这类计算模型被称为膜系统或P系统，共分为三种类型：细胞型P系统[12]、组织型P系统[13]和神经型P系统[14]。P系统由分层的膜结构，对象的多重集和规则组成，具有分布式、极大并行性、非确定性和可扩展性等特点，且易于与其他进化算法相结合[15]。本文采用具有分层结构的细胞型P系统，其结构如图1所示。

图1中的膜结构由6层膜组成，最外层膜称为表层膜，它将细胞内部结构与外界环境隔开。不包含其他膜结构的称为基本膜，每个膜所包围的部分称为区域。多目标对象用大小写字母表示，常见的进化规则类型有多重集重写规则(例如 $d \rightarrow dy, y \rightarrow c$)和运输规则等[16]。

根据图1可将1) V 是目标对象的字母表，其包含的元素为字符对象；

2) $T \subseteq V$ 是输出字母表；

3) $C \subseteq V$ 是催化剂；

4) μ 是一个由 m 个膜组成的膜结构，各个膜及其所围的区域用标号集 H 表示， $H = \{1, 2, 3, \dots, m\}$ ，其中 m 称为膜系统 Π 的度；

5) $\omega_i \in V^* (1 \leq i \leq m)$ 表示膜结构 μ 中区域 i 包含的对象多重集；

6) $R_i (1 \leq i \leq m)$ 是关于膜系统 Π 和区域 μ 的进化规则的有限集；

7) 膜结构可等价表示为： $\Pi = \{V, T, C, \mu, \omega_1, \omega_2, \dots, \omega_m, R_1, R_2, \dots, R_m\}$ 。

P系统是从活细胞在内部结构处理化合物的方式中抽象出来的一种计算模型[17]。在膜结构所定义的区域中，目标对象用字符表示，根据给定的规则实现进化。因此膜系统作为一个建模框架可以很好的与其他进化算法相结合，用来构建一个用于求解近似优化的多目标进化算法，具有易于理解、可扩展性和可编程性、固定分块性和处理离散数据的能力等优点[18]。

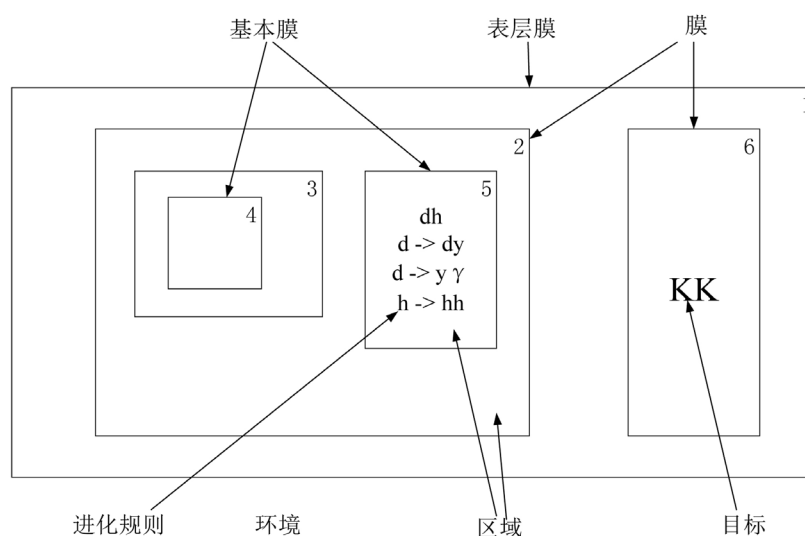


Figure 1. Cell type P system structure

图1. 细胞型P系统结构

3. MPSO

受膜系统功能和在分层结构中处理化合物的方式启发, 结合粒子群算法本文提出了一种基于膜系统框架的多目标粒子群算法(a multi-object particle swarm optimization algorithm based on the framework of membrane system, MPSO)。MPSO 中采用具有层次结构的细胞型 P 系统, 用字符对象表示优化问题的可行解, 由字符对象组成的解集构成多重集。利用类细胞 P 系统理论, 规则包括受 PSO 启发的进化规则、P 系统的进化规则和置换规则。根据膜系统的分层结构, 利用粒子群优化的个体最优和全局最优概念, 在基本膜中采用粒子群算法实现多搜寻策略并行更新解, 通过进化规则保持搜索解的多样性。在表层膜中, 利用外部档案使用非支配解集和拥挤距离机制提高非支配解的多样性和算法收敛速度, 使解集近似于 Pareto 前沿。

拥挤距离是判断外部档案中粒子(字符对象)与相邻个体间远近的指标, 如式(3)所示。拥挤距离越大说明该个体与其他个体越分散。

$$d_i = \sum_{m=1}^n \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{\max} - f_m^{\min}} \quad (3)$$

式(4)中, n 表示目标函数的个数, d_i 表示第 i 个字符对象在种群中的拥挤距离, f_m^{\max} 和 f_m^{\min} 分别表示种群中第 m 个目标取得的最大值和最小值, f_m^{i+1} 和 f_m^{i-1} 个字符对象在第 m 维两侧最临近点的目标函数值, 其中 $f_m^{i-1} \leq f_m^{i+1}$ 。

基于膜系统的粒子群算法的具体步骤如图 2 所示。

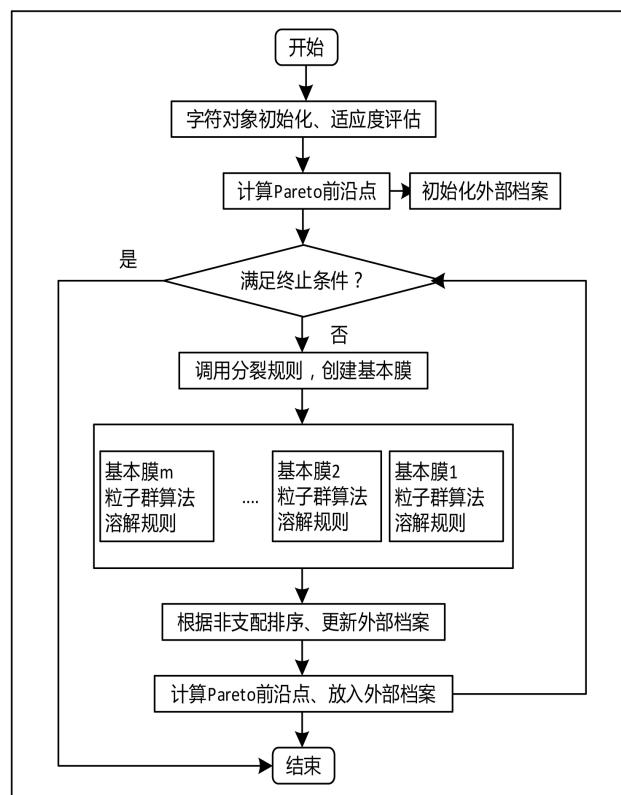


Figure 2. Algorithm flowchart
图 2. 算法流程图

步骤 1. 初始化及适应度评估。在表层膜内生成 N 个字符，每个字符包含 D 维变量，并在满足多目标优化问题约束的前提下，采用 10 进制的编码方式依次对 N 个字符进行初始化，其描述公式如公式(4)所示。

$$S_{i,j} = (S_{\max,j} - S_{\min,j}) * rand() + S_{\max,j} \quad (4)$$

其中 $S_{i,j}$ 表示第 i 个的第 j 维目标对象 ($N \geq i \geq 1, D \geq j \geq 1$)， N 为目标对象数目(粒子的数目)， $S_{\min,j}$ 与 $S_{\max,j}$ 分别表示矩阵 S 中的最小值和最大值，利用函数 $rand()$ 随机生成 0 至 1 之间的数字。

步骤 2. 根据适应度求出当前字符中 Pareto 前沿点，时间复杂度为 $O(N^2)$ 。

步骤 3. 初始化外部档案， $[]_0 \rightarrow [[]_1, []_2, \dots, []_m]_0$ 操作是在表层膜内调用分裂规则创建 m 个基本膜，利用 NSGA-II 中的非支配排序机制，将字符对象根据公式(5)分派给 m 个基本膜。

$$\begin{aligned} \omega &= sort(\omega) \\ \omega &= \{\omega_1, \omega_2, \dots, \omega_m\} \\ n &= \frac{sizeof(\omega)}{m} \end{aligned} \quad (5)$$

ω 为字符对象解集，函数 $sort()$ 利用非支配排序返回一系列解 ω 并在解集 ω 中使用通信规则 (w, in) ， n 表示每个基本膜中字符对象的数量。

步骤 4. 在基本膜内独立执行粒子群算法。各个基本膜内以最先存入外部档案内的 Pareto 前沿点为种群“最优”个体，运用公式(2)和(3)计算并更新个体速度和位置，再重新计算适应度。

步骤 5. 溶解。完成各自的粒子群算法后，调用基本膜内的溶解规则从而使各个基本膜破裂，将新产生的字符(个体)重新释放到另一基本膜内。直至所有基本膜都被溶解，并根据通信规则 (w, out) 将新产生的字符放入表层膜内。

步骤 6. 计算前沿点，存入外部档案。若 Pareto 前沿点数量小于预设数值 R ，则直接将所有点存入外部档案中。若 Pareto 前沿点数量大于预设数值，根据公式(4)计算所有 Pareto 前沿点的拥挤距离，从拥挤距离最小的点开始逐一删除，直至备选存入外部档案的 Pareto 前沿点数量与预设数值相等，在将这些前沿点存放在外部档案中。

步骤 7. 计算非支配排序，更新外部档案。判断外部档案字符数量是否超出限制，若超出限制，重新计算档案内所有字符的拥挤距离和非支配的等级，找出非支配等级小的字符。若存在两个字符对象非支配等级相等，需比较两个字符对象的拥挤距离，并选取拥挤距离大的一方，重复操作直至外部档案内字符数量与预设值相等。在表层膜中使用公式(5)并利用重写规则 $u \rightarrow v$ 更新外部档案中的字符对象。

步骤 8. 迭代。设置最大迭代次数作为终止条件，判断当前状态是否满足结束循环条件，若不满足，则继续执行。

步骤 9. 输出外部档案内的所有字符，即 Pareto 前沿点。

4. 仿真实验

4.1. 实验设置

为验证算法性能，采用 ZDT 系列与 DTLZ 系列函数进行测试，并与其他代表性几个最具代表性的多目标优化算法(包括 MOPSO、SPEA2、PESA2)进行对比。MPSO 算法中的学习因子(或加速度系数) c_1 与 c_2 设定为 2，二维目标的测试函数 KUR、ZDT 中的外部档案容量为 100，三维目标的测试函数 DTLZ 中的外部档案容量为 200；不同的多目标优化算法对二维目标测试函数迭代 10,000 次，对三维目标测试函数

DTLZ 迭代 100,000 次；独立运行各个算法 30 次，计算 IGD 的均值；基本膜的数量为 10 个。

4.2. 二维目标实验结果分析

对每种算法分别进行 30 次近似 Pareto 前沿计算，使用 IGD (inverted generational distance, 反转世代距离)度量算法获取的近似 Pareto 前沿与真实 Pareto 前沿之间的距离，评估算法性能。实验结果如图 3 所示。

对比观察 KUR 测试函数下各算法求得的近似 Pareto 前沿点，可以看出新算法与 SPEA2、PESA2 结果基本相近，前沿点分布较均匀，不管是收敛速度还是质量都优于传统 MOPSO 算法。

对比观察 ZDT1 和 ZDT2 测试函数下各算法求得的近似 Pareto 前沿点(图 4 与图 5)，可以看出新算法和 MOPSO 算法的收敛情况明显优于 SPEA2 和 PESA2 算法。但与 MOPSO 相比，新算法的近似 Pareto 前沿点分布更加均匀。

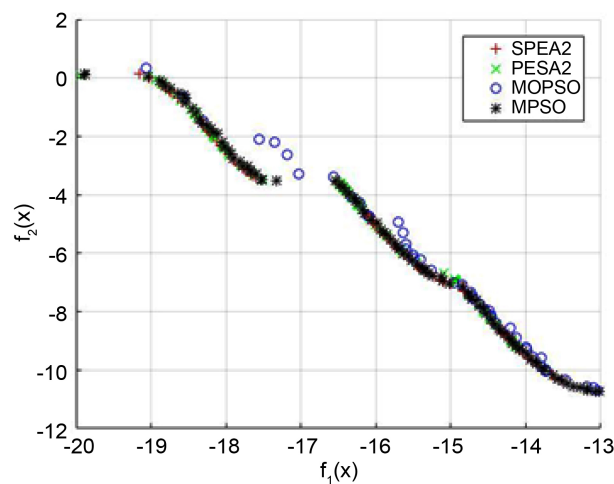


Figure 3. The approximate Pareto frontier distribution point map under KUR

图 3. 在 KUR 下求得近似 Pareto 前沿分布点图

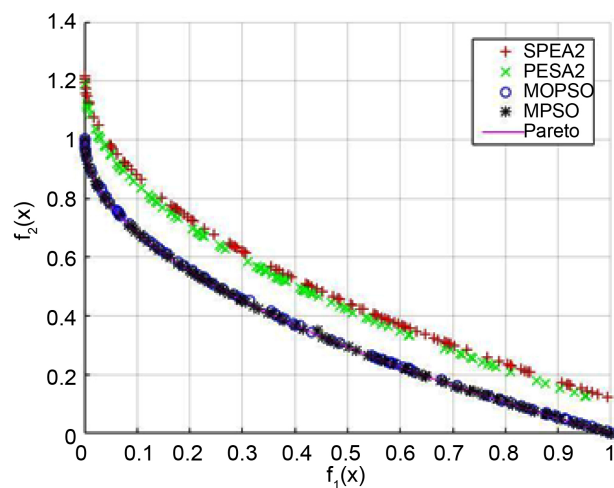


Figure 4. The approximate Pareto frontier distribution point map under ZDT1

图 4. 在 ZDT1 下求得近似 Pareto 前沿分布点图

对比观察 ZDT3 和 ZDT6 测试函数下各算法求得的近似 Pareto 前沿点(图 6 与图 7), 可以看出各算法结果基本相近。在 ZDT3 测试函数下, 近似 Pareto 前沿点在段新算法要稍优于其他三种算法。

IGD 数值的高低能够反映算法的收敛性与多样性, 故根据表 1 所示的 IGD 性能数据, 可以看出 MPSO 对比于其他优化算法在 KUR、ZDT3、ZDT6 测试函数上相对占优, 说明 MPSO 具有较好的收敛性、多样性以及稳定性。

4.3. 三维目标实验结果分析

为进一步检验算法对多目标优化的能力, 将 MPSO 与其他五种基于 PSO 的多目标优化算在测试函数上进行对比分析, 实验结果如表 2 所示。IGD 数值的方差(Std.)和均值(Mean)反映了算法的稳定性与准确性, 根据表 2 所示, MOEA/D 在 ZDT1、ZDT5、ZDT5、ZDT6 测试函数上 IGD 数值最小。总体而言, MPSO 相对于 dMOPSO、SMPSO、MMOPSO、NSGAI 具有较好的稳定性与准确性, 尤其是在 ZDT1、

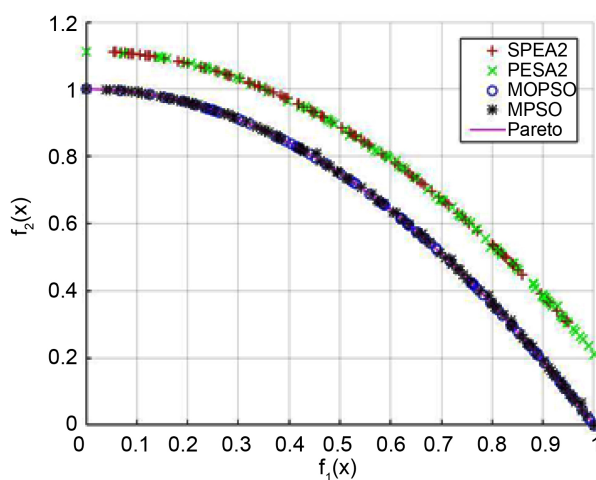


Figure 5. The approximate Pareto frontier distribution point map under ZDT2

图 5. 在 ZDT2 下求得近似 Pareto 前沿分布点图

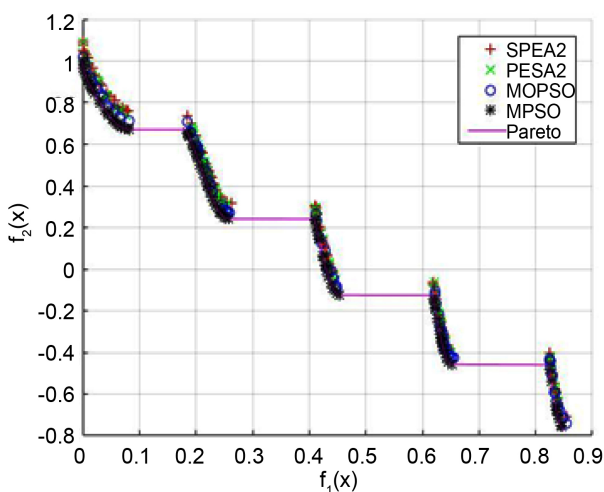


Figure 6. The approximate Pareto frontier distribution point map under ZDT3

图 6. 在 ZDT3 下求得近似 Pareto 前沿分布点图

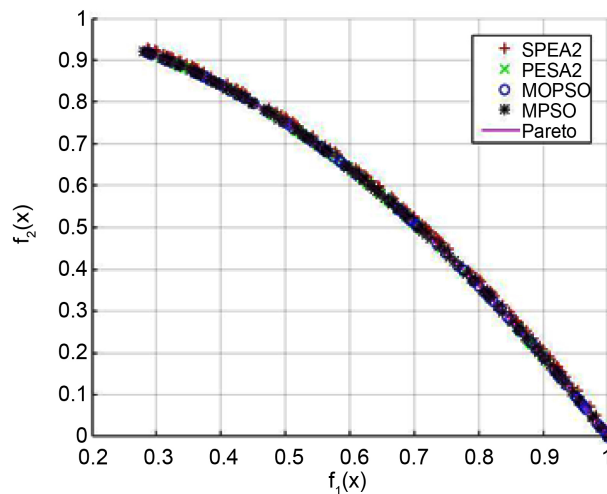


Figure 7. The approximate Pareto frontier distribution point map under ZDT6

图 7. 在 ZDT6 下求得近似 Pareto 前沿分布点图

Table 1. IGD performance statistics of different algorithms on five test functions

表 1. 不同算法在 5 个测试函数上的 IGD 性能统计

	MPSO		MOPSO		SPEA2		PESA2	
	平均值	方差	平均值	方差	平均值	方差	平均值	方差
KUR	5.96E-02	2.25E-01	2.25E+00	4.50E+00	5.71E-02	3.46E-02	6.12E-02	2.75E-03
ZDT1	4.84E-02	1.34E-01	5.63E-02	2.10E+00	1.31E-01	1.03E-01	1.05E-01	1.47E-01
ZDT2	4.96E-01	9.75E+02	6.23E-02	4.10E-01	1.52E-01	5.12E-01	1.41E-01	3.51E-01
ZDT3	5.96E-02	2.50E-01	3.79E-01	9.50E-02	4.50E+00	3.54E-01	4.50E+00	2.51E-01
ZDT6	4.36E-02	2.15E-01	5.29E-02	6.50E-01	5.68E-02	5.51E-01	5.41E-02	2.51E-01

Table 2. IGD performance statistics of different algorithms on 7 test functions

表 2. 不同算法在 7 个测试函数上的 IGD 性能统计

测试函数		MPSO	dMOPSO	SMPSO	MMOPSO	MOEA/D	NSGAI
DTLZ1	Mean	1.44e-04	5.79e00	2.08e00	1.66e-03	1.37e-04	1.85e-02
	Std.	2.35e-05	7.67e-01	2.57e-01	2.78e-03	3.78e-06	2.67e-03
DTLZ2	Mean	6.23e-04	1.98e-02	1.06-03	7.22e-04	1.63e-04	4.88e-02
	Std.	4.56e-06	2.45e-04	5.42e-04	4.24e-05	5.76e-05	1.35e-04
DTLZ3	Mean	1.43e-03	3.03e+01	3.75e00	5.86e-04	1.78e-04	4.92e-02
	Std.	2.98e-05	4.16e-01	6.29e-01	2.38e-06	4.55e-05	3.27e-03
DTLZ4	Mean	4.41e-04	7.61e-03	3.02e-04	7.33e-04	9.45e-01	4.83e-02
	Std.	3.56e-04	4.65e-03	7.24e-04	1.78e-03	3.02e-01	3.32e-05
DTLZ5	Mean	3.15e-05	2.58e-03	5.18e-05	6.54e-05	2.16e-02	2.85e-03
	Std.	7.98e-06	1.98e-01	3.87e-05	3.75e-06	3.45e-04	1.27e-03
DTLZ6	Mean	2.35e-06	4.25e-03	3.17e-06	3.58e-06	2.26e-02	2.97e-03
	Std.	3.24e-06	3.78e-04	2.54e-06	3.28e-06	3.05e-04	1.05e-05
DTLZ7	Mean	4.23e-04	1.33e-03	1.76e-03	1.28e-03	1.03e-01	5.63e-02
	Std.	1.98e-06	2.45e-04	9.24e-04	2.56e-02	3.91e-04	1.27e-02

ZDT2、ZDT3、ZDT6 测试函数上。因此说明，PSO 算法具有较好的平衡全局和局部搜索能力。

5. 结论

为解决实际生活中的多目标优化问题，本文提出了一种基于膜系统框架的多目标粒子群优化算法。根据膜系统的分层结构，利用粒子群优化的个体最优和全局最优概念，在基本膜上并行更新解，在表层膜中，利用外部档案使用非支配解集和拥挤距离机制提高非支配解的多样性。本文将该算法与其他几种优化算法在 KUR、ZDT 系列和 DTLZ 系列测试函数进行测试，实验结果表明：基于膜系统的粒子群算法具备收敛速度快、近似 Pareto 前沿点分布均匀等特点。因此说明新算法在求解多目标优化问题方面是可行且有效的。

参考文献

- [1] Deb, K. (2001) Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester.
- [2] 宁伟康. 进化多目标优化算法研究及其应用[D]: [博士学位论文]. 西安: 西安电子科技大学, 2018.
- [3] Li, X. (2003) A Non-Dominated Sorting Particle Swarm Optimizer for Multi-Objective Optimization. *Genetic and Evolutionary Computation, Genetic and Evolutionary Computation Conference*, Chicago, 12-16 July 2003, 37-48. https://doi.org/10.1007/3-540-45105-6_4
- [4] Coello, C.A.C., Pulido, G.T. and Lechuga, M.S. (2004) Handling Multiple Objectives with Particle Swarm Optimization. IEEE Press, Piscataway. <https://doi.org/10.1109/TEVC.2004.826067>
- [5] Zhang, Q. and Li, R. (2013) Cloud Resource Scheduling Based On Improved Particle Swarm Optimization Algorithm by Membrane Computing. *Computer Engineering and Applications*, **49**, 40-44.
- [6] Liu, C., Han, M. and Wang, X. (2011) A Multi-Objective Evolutionary Algorithm Based on Membrane Systems. *The 4th International Workshop on Advanced Computational Intelligence*, Wuhan, 103-109. <https://doi.org/10.1109/IWACI.2011.6159983>
- [7] Parsopoulos, K.E. and Vrahatis, M.N. (2005) Unified Particle Swarm Optimization for Tackling Operations Research Problems. *Swarm Intelligence Symposium*, Pasadena, 8-10 June 2005, 53-59.
- [8] Hu, X. and Eberhart, R.C. (2002) Multi-Objective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. *Congress on Evolutionary Computation*, Honolulu, 12-17 May 2002, 1677-1681.
- [9] Dou, Z. and Gao, L. (2012) Feature Selection in Conditional Random Fields Using a Membrane Particle Swarm Optimizer. *Journal of Xidian University*, **39**, 107-112.
- [10] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, Perth, 27 November-1 December 1995, Vol. 4, 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [11] Paun, G., Rozenberg, G. and Salomaa, A. (2009) Handbook of Membrane Computing. Oxford University Press, Oxford. <https://doi.org/10.1007/978-3-642-11467-0>
- [12] 韩丽莎. 类细胞 P 系统在划分聚类中的研究[D]: [硕士学位论文]. 济南: 山东师范大学, 2015.
- [13] 葛平平. 可满足性问题的改进型类组织 P 系统的求解研究[D]: [硕士学位论文]. 淮南: 安徽理工大学, 2015.
- [14] 王康. 细胞型和神经型 P 系统的应用问题研究[D]: [硕士学位论文]. 成都: 西华大学, 2011.
- [15] 张葛祥, 潘林强. 自然计算的新分支——膜计算[J]. *计算机学报*, 2010, 33(2): 208-214.
- [16] Pan, L.Q., Wang, J. and Hoogboom, H.J. (2011) Spiking Neural P Systems with Astrocytes. *Neural Computation*, **24**, 805-825. https://doi.org/10.1162/NECO_a_00238
- [17] Ye, L. and Guo, P. (2016) Design a Membrane System for Solving Linear Equations. *Journal of Computational and Theoretical Nanoscience*, **13**, 1561-1568. <https://doi.org/10.1166/jctn.2016.5081>
- [18] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. *软件学报*, 2009, 20(2): 271-289.

知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询。
2. 通过知网首页 <http://cnki.net/>顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org