DOI: 10.3901/JME.2019.12.222

膜计算多粒子群算法*

陈东宁^{1,2} 王跃颖^{1,2} 姚成玉³ 刘一丹^{1,2} 吕世君^{1,2}

- (1. 燕山大学河北省重型机械流体动力传输与控制重点实验室 秦皇岛 066004;
2. 先进锻压成形技术与科学教育部重点实验室(燕山大学) 秦皇岛 066004;
3. 燕山大学河北省工业计算机控制工程重点实验室 秦皇岛 066004)

摘要: 针对粒子群(Particle swarm optimization, PSO)算法进化后期收敛速度较慢,易陷入局部极值点,精度较差等不足,提出膜计算多粒子群(MC-MPSO)算法。在该算法中,将原始 PSO、标准 PSO、中值导向粒子群(MPSO)、扩展粒子群(EPSON)、多作用力粒子群(MFPSO)、两阶段作用力粒子群(TFPSO)等六种具有不同优点的粒子群算法分别放入六个基本膜内,提出 MC-MPSO 算法的膜间交流与粒子更新机制,在进化前期,各粒子群算法按自身机制进行搜索寻优,即各基本膜各自进化来充分发挥各基本膜内算法的优点;在进化后期,各基本膜内算法与比自身更好的表层膜内最优解粒子交流,各表层膜逐步吞并搜索能力较差的基本膜,而最适合问题优化求解的基本膜长大并按照表层膜输出,使 MC-MPSO 算法集成了基本膜内六种粒子群算法的各自优势,并具有适应不同类型优化求解问题的寻优能力。通过与基本膜内六种粒子群算法的测试对比,与遗传算法、鱼群算法及其他基于膜计算的粒子群算法的比较,证明了 MC-MPSO 算法具有更好的寻优能力和适用性。最后,将 MC-MPSO 算法应用于串联和桥式系统可靠性优化问题,验证了所提算法的有效性。

关键词: 粒子群算法;膜计算;MC-MPSO 算法;可靠性优化

中图分类号: TP18; TB114

Membrane Computing Multi Particle Swarm Optimization(MC-MPSO) Algorithm

CHEN Dongning^{1,2} WANG Yueying^{1,2} YAO Chengyu³ LIU Yidan^{1,2} LÜ Shijun^{1,2}

- (1. Hebei Provincial Key Laboratory of Heavy Machinery Fluid Power Transmission and Control, Yanshan University, Qinhuangdao 066004;
2. Key Laboratory of Advanced Forging & Stamping Technology and Science(Yanshan University), Ministry of Education of China, Qinhuangdao 066004;
3. Key Laboratory of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004)

Abstract: Membrane computing multi particle swarm optimization(MC-MPSO) algorithm is proposed to overcome of particle swarm optimization(PSO) algorithm the defections of easy getting trapped in a local optimum, slow convergent speed and low accuracy in the later evolution process. In MC-MPSO algorithm, original PSO, standard PSO, median-oriented PSO(MPSO), extended PSO(EPSON), multi force PSO(MFPSO), two-stage force PSO(TFPSO) with different advantages of particle swarm algorithm are put into six membranes respectively. The communication among membranes and update mechanisms of particles are proposed in MC-MPSO. First of all, elementary membrane grow up according to their own searching mechanism with the advantages of each PSO algorithm. Secondly, six algorithms in the membranes exchange optimally with better membrane, and the surface membrane gradually swallow the membranes of poor searching ability. Then the membranes which can solve the problems properly grow up and the best membrane export through surface membrane. The MC-MPSO algorithm integrates the advantages of the six particle swarm optimization algorithms, and has the ability to adapt to different types of optimization problems. By comparing with the test of six algorithms in the membranes, the comparison of genetic algorithm, fish swarm algorithm and other particle swarm optimization algorithms based on membrane computing, the results show that the MC-MPSO algorithm has better search capability of optimal

* 国家自然科学基金(51405426, 51675460)、中国博士后科学基金(2017M621101)和河北省自然科学基金(E2016203306)资助项目。
20180515 收到初稿, 20181116 收到修改稿

solution and wide applicability. Finally, the MC-MPSO algorithm is applied in the reliability optimization of series and bridge systems. The effectiveness of the proposed algorithm is verified.

Key words: PSO algorithm; membrane computing; MC-MPSO algorithm; reliability optimization

0 前言

粒子群(Particle swarm optimization, PSO)算法是一种进化计算技术, 1995 年由 EBERHART 和 KENNEDY 提出^[1]。相对于其他的群智能优化算法, 粒子群算法具有结构简单、调整参数少、鲁棒性好等优点, 被广泛应用于各类工程优化问题^[2-7]。但粒子群算法进化后期收敛速度慢、易陷入局部最优、精度较差, 因而自粒子群算法被提出以来一直被国内外学者们研究改进^[8-9]。

粒子群算法是基于模拟鸟群个体之间的交互关系提出的, 因而, 粒子间的交互关系直接影响算法的性能, 它体现在粒子间作用力的设定上, 很多学者从调节惯性权重、调节引力和调节斥力三个方面对粒子间作用力规则进行改进。EBERHART 和 SHI 最早提出了线性递减的动态惯性权重, 调节作用于粒子间的作用力, 提高粒子群算法的性能, 即为标准 PSO 算法^[10]; 文献[11-12]分别提出了一种参考混沌理论的自适应惯性权重, 通过惯性权重调节粒子间的作用力来提升算法性能; 文献[12]的中值导向粒子群(Median-oriented PSO, MPSO)算法用中位数的粒子平均化粒子群间引力, 提高了算法收敛速度; 扩展粒子群(Extended PSO, EPSO)算法同时扩展粒子间的引力和斥力, 减少算法陷入局部的可能, 增强收敛解的稳定性^[13]。在此基础上, 笔者提出了多作用力粒子群(Multi force PSO, MFPSO)算法, 在不同搜索阶段采用不同的作用力规则, 提高了算法寻优性能^[14]; 为进一步提高搜索精度, 降低寻优时间, 又提出了两阶段作用力粒子群(Two-stage force PSO, TFPSO)算法^[15]。上述算法从不同侧面对粒子群算法进行了改进, 但又各有优势和不足, 有必要博采众长、集成改进算法的优势, 使算法同时适合解单峰、多峰等不同优化问题并具有更好的寻优能力。

膜计算是欧洲科学院院士 Păun 在 1998 年提出的^[16], 也被称为 P 系统^[17]。膜计算是从多种生物膜的功能和结构中抽象出的一些最基本的特征而构成的计算模型, 其三大关键组成部分为: 膜的层次结构、表示对象的多重集和进化规则集, 是具有极大并行性、分布式和非确定性的计算模型^[18-19]。目前膜计算分别与遗传算法^[20]、鱼群算法^[21]等算法结合以提高算法性能。膜计算与粒子群算法结合也取得

了一定进展, 主要有: 文献[22]提出了基于单层膜结构的 P 系统粒子群算法, 各基本膜内用标准 PSO 算法; 文献[23]将标准 PSO 算法放入五个膜内, 三个参照膜进行全局搜索, 以搜索最优解所在区域, 一个主膜用于最优解所在区域确定后的精细化搜索, 一个表层膜用于吸收主膜排出的粒子, 进而通过膜区域之间信息交流寻找最优解。现有膜计算与粒子群结合的算法, 采用的膜间交流形式和膜内粒子群算法较单一, 搜索机制、寻优能力仍有欠缺, 因而有必要进一步研究综合寻优能力更强的膜计算与粒子群算法结合的算法。

为此, 本文将经典 PSO、改进 PSO 等六种有代表性的粒子群算法放入六个膜内, 利用膜计算的分布并行性, 提出膜计算多粒子群(Membrane computing multi PSO, MC-MPSO)算法; 然后, 通过测试函数并与其他算法进行比较, 证明 MC-MPSO 算法对不同类型优化问题的适用性; 最后, 将所提算法应用于可靠性优化问题, 验证了 MC-MPSO 算法的有效性。

1 膜计算多粒子群算法的提出

膜计算多粒子群算法的思想是: 将六种具有不同优点的粒子群算法(原始 PSO、标准 PSO、MPSO、EPSO、MFPSO 和 TFPSO)分别放入六个基本膜内, 在算法进化前期, 各粒子群算法按照自身机制搜索, 即各基本膜各自进化来充分发挥各基本膜内算法的优势; 在算法进化后期, 各基本膜内算法与比自身更好的膜内最优解粒子交流, 通过载体综合各个基本膜共同完成寻优任务, 表层膜逐步排除搜索能力差的膜, 而最适合解问题优化求解的膜长大并输出。

1.1 膜间交流机制

如图 1 所示, MC-MPSO 算法中将具有不同优点的六种粒子群算法分别放入六个基本膜内, 包括: 适合解对称函数问题且运行时间较短的原始 PSO 算法^[1]和标准 PSO 算法^[10], 适合解复杂非线性多峰函数问题的 MPSO 算法^[12], 适合解具有多个局部极小点的多峰问题的 EPSO 算法^[13], 适合解复杂多模态问题的 MFPSO 算法^[14], 以及能提高复杂多峰问题收敛精度的 TFPSO 算法^[15]。原始 PSO 能提高算法的随机性和多样性; 标准 PSO 随着迭代次数变化的惯性权重, 提高整体运算效果; 引用 MPSO 算法以利用其较高的寻优收敛速度; EPSO 算法收敛解

的稳定性较好;MFPSO 算法在解决种群维数较高的问题上具有很大优势;TFPSO 算法按照进化停滞次数阈值进行区分两个阶段,深化算法的局部精细化搜索,提高收敛速度。

图 1 MC-MPSO 算法的膜组成示意图

MC-MPSO 算法在搜索前期,各粒子群算法按照自身机制进行搜索寻优,即各基本膜充分发挥膜内算法的优势;在搜索后期,每个细胞膜在细胞中并不是独立存在,而是与其他细胞膜相互借鉴优势,以便更好地发展自己。用全连接拓扑结构,将各基本膜两两相连,各膜内粒子均能共享整个群体的信息。全连接拓扑结构可以充分利用多个膜的信息,但是计算时间较长。另外,如果一个膜一直同时受到几个膜的能量影响则不能有效吸收每个膜的能量,即粒子一直同时受到多个方向的力作用,其合力会消弱单个力的优势。为此,对全连接型拓扑结构进行改进,提出膜间载体交流的质数拓扑结构,通过表 1 所示的质数设置方法对全连接拓扑结构进行连接取舍,以选取符合条件的膜来进行交流。

表 1 质数设置方法

质数分布	基本膜 1	基本膜 2	基本膜 3	基本膜 4	基本膜 5	基本膜 6
基本膜 1	—	7	5	3	2	11
基本膜 2	11	—	7	5	3	2
基本膜 3	2	11	—	7	5	3
基本膜 4	3	2	11	—	7	5
基本膜 5	5	3	2	11	—	7
基本膜 6	7	5	3	2	11	—

质数拓扑结构膜间交互方式,计算效率优于全连接方式,同时,各基本膜在迭代搜索时,有时与一个基本膜进行交流,有时与多个基本膜进行交流,相对于只与一个膜交流或是一直与多个膜交流,进化速度加快,体现了动态拓扑的思想。

在算法进化后期,满足表 1 的拓扑质数条件的两膜进行交流,应用表 1 质数设置表将同时来的能量选择性吸收。同时,满足单个膜进化停滞次数时也进行膜间交流。采用智能的动态切换条件,当单

个基本膜自身搜索机制运行可能比受到其他膜的交流效果更好,或在几次交流后自身搜索机制更占优势,为此引入算法进化停滞次数 g 为阈值,将该阈值作为膜间是否交流的切换依据,即基本膜内按照自身搜索机制已经连续达到 g 代数且再无更新时,则与其他膜传过来的载体进行交流。针对所提的 MC-MPSO 算法单个膜可能出现的停滞现象,用其他膜来影响它。MC-MPSO 算法强化了群体内部个体间的协同与合作。

以两个不同膜 k 和膜 h 间、两个不同粒子 i 和粒子 j 间交流为例,MC-MPSO 算法的进化规则为:①选择规则,算法各基本膜选择膜内当代适应值最好的粒子迭代到下一步运算,并在算法后期对膜间交流次数进行排序,选择被交流次数少的膜以便淘汰;②胞外分泌规则 $[i]_k \rightarrow [j]_h$,在算法后期淘汰表层膜将达到催化剂要求的膜;③入规则 $i[k] \rightarrow [j]_k$,即 i 被复制成 j ,进入膜内;④出规则 $[i]_k \rightarrow j[k]$,即 i 被复制成 j ,输出膜;⑤膜溶解规则 $[i]_k \rightarrow j$,在 MC-MPSO 算法中表示基本膜被溶解,膜内粒子直接作为结果输出;⑥对运输规则 $(i, \text{in}; j, \text{out})$,即在同一时间 i 进入膜, j 退出膜;⑦ $Qi \rightarrow Qj$ 为催化剂的应用,在催化剂 Q 作用下 i 进化成 j ,而 Q 本身不发生变化。

当载体运输的最优粒子比本膜内最优解好且满足规则条件时,载体内粒子 j 对即将比较的膜内粒子 i 进行能量 e 转移,粒子能量 e 转移公式为

$$e = p_{jd}(t) - x_{id}(t) \quad (1)$$

式中, $x_{id}(t)$ 、 $p_{jd}(t)$ 分别为粒子 i 、个体最优粒子 j 第 d 维在第 t 代的位置。

MC-MPSO 算法中,两个不同膜 k 和膜 h 间的载体为载体 z_{kh} 或 z_{hk} , z_{kh} 表示膜 k 对膜 h 的载体, z_{hk} 表示膜 h 对膜 k 的载体,其中 $k, h = 1, 2, \dots, 6$ 。膜间的载体运转如式(2)~(4)所示

$$V_{z_{kh}} = s_k c_{kh} r_{kh} [p_{kid}(t) - x_{hid}(t)] \quad (2)$$

$$V_{z_{hk}} = s_k c_{kh} r_{kh} [p_{hid}(t) - x_{kid}(t)] \quad (3)$$

$$V_h = \sum_{k \neq h} V_{z_{kh}} \quad (4)$$

式中, $V_{z_{kh}}$ 、 $V_{z_{hk}}$ 分别为膜 k 通过载体对膜 h 的能量影响和膜 h 通过载体对膜 k 的能量影响; V_h 为基本膜 h 内受到的膜间力; $p_{kid}(t)$ 、 $p_{hid}(t)$ 分别为膜 k 、 h 内粒子的第 d 维在第 t 代的个体历史最优位置; $x_{kid}(t)$ 、 $x_{hid}(t)$ 分别为膜 k 、 h 内粒子 i 的第 d 维在第 t 代的位置; s_k 为膜间交流因子,若载体携带粒子比膜内最优粒子优则 s_k 为 1,否则为 0; c_{kh} 为加

速常数; r_{kh} 为(0,1)区间的随机数。

给每个膜分配一个记忆因子,各基本膜间的每次交流都被记录,每次能量转移将相应被吸收的膜对应的记忆因子累加一次,当后期膜间交流次数达到催化剂要求(即迭代次数分别为 t_1 、 t_2 、 t_3 、 t_4 、 t_5)时,对记忆因子排序,排序最小的值对应的膜进行表层膜的胞外分泌规则,直至剩余最后一个基本膜。

1.2 膜内粒子更新机制

膜1内原始PSO算法的速度和位置更新公式如式(5)、(6)所示

$$v_{1id}(t+1) = v_{1id}(t) + c_1 r_1 [p_{1id}(t) - x_{1id}(t)] + c_2 r_2 [p_{1gd}(t) - x_{1id}(t)] + c_3 V_1 \quad (5)$$

$$x_{1id}(t+1) = x_{1id}(t) + v_{1id}(t+1) \quad (6)$$

式中, c_1 、 c_2 、 c_3 为加速常数; r_1 、 r_2 为(0,1)区间的随机数; $v_{1id}(t)$ 、 $v_{1id}(t+1)$ 分别为膜1内粒子 i 第 d 维在第 t 、 $t+1$ 代的速度; $p_{1gd}(t)$ 为膜1内第 d 维在第 t 代全局历史最优粒子的位置。

膜2内标准PSO算法的位置更新公式与膜1的位置更新公式相同,速度更新公式如式(7)所示

$$v_{2id}(t+1) = w_1 v_{2id}(t) + c_1 r_1 [p_{2id}(t) - x_{2id}(t)] + c_2 r_2 [p_{2gd}(t) - x_{2id}(t)] + c_3 V_2 \quad (7)$$

式中, $v_{2id}(t)$ 、 $v_{2id}(t+1)$ 分别为膜2内粒子 i 第 d 维在第 t 、 $t+1$ 代的速度; w_1 为惯性权重,

$w_1 = w_{\min} \left(\frac{w_{\max}}{w_{\min}} \right)^{\left(1 - \frac{t}{t_{\max}}\right)}$, $w_{\min}$ 、 $w_{\max}$ 分别为最小、最大惯性权重, $t_{\max}$ 为最大迭代次数; $p_{2gd}(t)$ 为膜2内第 d 维在第 t 代全局历史最优粒子的位置。

膜3内MPSO算法引入加速因子 $a_i(t)$ 以提高全局搜索能力

$$a_i(t) = \frac{A_i(t)}{\sum_{j=1}^N A_j(t)} \quad (8)$$

$$A_i(t) = \frac{fit_i(t) - \max(fit_i(t))}{\text{median}(fit_i(t)) - \max(fit_i(t))} \quad (9)$$

式中, $fit_i(t)$ 为粒子 i 的适应值; $\max(fit_i(t))$ 、 $\text{median}(fit_i(t))$ 分别为 $fit_i(t)$ 的最大值和中值。

MPSO算法的速度和位置的更新公式如下

$$v_{3id}(t+1) = v_{3id}(t) + a_i(t) [r_1 (p_{3id}(t) - p_{md}(t) - x_{3id}(t)) + r_2 (p_{3gd}(t) - p_{md}(t) - x_{3id}(t))] + c_3 V_3 \quad (10)$$

$$x_{3id}(t+1) = x_{3id}(t) + v_{3id}(t+1) + \frac{1}{2} [r_1 (p_{3id}(t) - x_{3id}(t)) + r_2 (p_{3gd}(t) - x_{3id}(t))] \quad (11)$$

式中, $v_{3id}(t)$ 、 $v_{3id}(t+1)$ 分别为膜3内粒子 i 第 d

维在第 t 、 $t+1$ 代的速度; $p_{md}(t)$ 为粒子的第 d 维在第 t 代的中间位置; $p_{3gd}(t)$ 为膜3内第 d 维在第 t 代全局历史最优粒子的位置。

膜4内EPSO算法的位置更新公式与膜1的位置更新公式相同,速度更新公式如式(12)所示

$$v_{4id}(t+1) = w_2 v_{4id}(t) + \sum_{j \in B(i)} c_j r_1 [p_{4jd}(t) - x_{4id}(t)] - \sum_{j \in W(i)} c_j r_2 [p_{4jd}(t) - x_{4id}(t)] + c_3 V_4 \quad (12)$$

式中, $v_{4id}(t)$ 、 $v_{4id}(t+1)$ 分别为膜4内粒子 i 第 d 维在第 t 、 $t+1$ 代的速度; c_j 为加速常数; w_2 为惯性权重, $w_2 = w_{\max} - \frac{t}{t_{\max}} (w_{\max} - w_{\min})$; $B(i)$ 为存放比当前粒子 i 适应值好的个体最优粒子的集合; $W(i)$ 为存放比当前粒子 i 适应值差的个体最优粒子的集合。

膜5内MFPSO算法的位置更新公式与膜1的位置更新公式相同,速度更新公式如式(13)所示

$$v_{5id}(t+1) = w_2 v_{5id}(t) + \alpha \sum_{j \in N} c_j r_1 [x_{5jd}(t) - x_{5id}(t)] + \beta \left(\sum_{j \in B(i)} c_j r_2 [p_{5jd}(t) - x_{5id}(t)] + \sum_{j \in W(i)} c_j r_2 [x_{5id}(t) - p_{5jd}(t)] \right) + \gamma c_g r_3 [p_{5gd}(t) - x_{5id}(t)] + c_3 V_5 \quad (13)$$

式中, $v_{5id}(t)$ 、 $v_{5id}(t+1)$ 分别为膜5内粒子 i 第 d 维在第 t 、 $t+1$ 代的速度; r_3 为(0,1)区间的随机数; c_g 为加速常数; $p_{5gd}(t)$ 为膜5内第 d 维在第 t 代全局历史最优粒子的位置; α 、 β 、 γ 分别为前、中、后3个时期搜索的切换系数,按照式(14)~(16)求得

$$\alpha = \begin{cases} 1 & t < l_1 \\ 0 & \text{其他} \end{cases} \quad (14)$$

$$\beta = \begin{cases} 1 & l_1 \leq t < l_2 \\ 0 & \text{其他} \end{cases} \quad (15)$$

$$\gamma = \begin{cases} 1 & t \geq l_2 \\ 0 & \text{其他} \end{cases} \quad (16)$$

式中, l_1 为前期向中期搜索切换时的切换因子,达到此迭代次数由前期向中期过渡; l_2 为中期向后期搜索切换时的切换因子,达到此迭代次数由中期向后期过渡。

膜6内TFPSO算法的位置更新公式与膜1的位置更新公式相同,速度更新公式如式(17)所示

$$v_{6id}(t+1) = w_2 v_{6id}(t) + \mu \left\{ \sum_{j \in B(i)} c_j r_1 [p_{6jd}(t) - x_{6id}(t)] + \sum_{j \in W(i)} c_g r_1 [x_{6id}(t) - p_{6jd}(t)] \right\} + \theta \{ c_1 r_2 [p_{6id}(t) - x_{6id}(t)] + c_2 r_3 [p_{6gd}(t) - x_{6id}(t)] \} + c_3 V_6 \quad (17)$$

式中, $v_{6id}(t)$ 、 $v_{6id}(t+1)$ 分别为膜6内粒子 i 第 d

维在第 t 、 $t+1$ 代的速度; $p_{6gd}(t)$ 为膜 6 内第 d 维在第 t 代全局历史最优粒子的位置; μ 、 θ 分别为前、后期切换系数, 由式(18)、(19)求得

$$\mu = \begin{cases} 1 & t < l_{\max} \\ 0 & \text{其他} \end{cases} \quad (18)$$

$$\theta = \begin{cases} 1 & t \geq l_{\max} \\ 0 & \text{其他} \end{cases} \quad (19)$$

式中, $l_{\max}$ 为算法进化的停滞次数, 当 t 达到 $l_{\max}$ 值时, 进化由前期转换为后期。

1.3 算法执行步骤

MC-MPSO 算法的流程如图 2 所示, 算法实现的具体步骤如下。

图2 MC-MPSO 算法流程

(1) 首先设定膜结构为六个基本膜外包围一个表层膜, 初始化每个基本膜中的粒子。

(2) 迭代初期, 各基本膜按自身机制进行优化, 选择当代最好的粒子复制, 并按出规则进行记录。

(3) 通过测试, 各算法运行 t_0 代时已将算法优势体现比较完全, 因而从 t_0 代开始, 各基本膜分别选择膜内最好粒子交给各自复制移民载体, 各载体满足拓扑质数条件与动态切换条件时送至对应膜

内, 若比当前膜内本代最优粒子好, 则按入规则把载体内粒子放入膜内进行吸收; 否则, 载体将携带的粒子吞噬并将对方膜内满足条件且比携带粒子好的本代最优粒子运回, 放入本膜内更新。

(4) 各膜间的每次交流执行入规则都会被记录, 当后期膜间交流次数满足催化剂要求时, 对记忆因子进行排序, 将最小记忆因子对应的膜进行表层膜的胞外分泌, 直至剩余一个最适合求解优化问题的膜, 当达到终止条件即迭代次数为 $t_{\max}$ 时, 将该基本膜内粒子溶解至表层膜中输出。

2 算法测试及对比

为验证 MC-MPSO 算法的寻优性能, 采用表 2

表2 Benchmark 函数

Benchmark 函数	范围
$f_1(x) = \sum_{i=1}^n x_i^2$ $\min f_1(x) = f_1(0, 0, \dots, 0) = 0$	[-100, 100]
$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ $\min f_2(x) = f_2(1, 1, \dots, 1) = 0$	[-50, 50]
$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1$ $\min f_3(x) = f_3(0, 0, \dots, 0) = 0$	[-600, 600]
$f_4(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e$ $\min f_4(x) = f_4(0, 0, \dots, 0) = 0$	[-32, 32]
$f_5(x) = \sum_{i=1}^n [x_i^2 - 10 \cos 2\pi x_i + 10]$ $\min f_5(x) = f_5(0, 0, \dots, 0) = 0$	[-5.12, 5.12]
$f_6(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ $y_i = 1 + \frac{x_i + 1}{4}, \min f_6(x) = f_6(-1, -1, \dots, -1) = 0$	[-50, 50]
$f_7(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ $\min f_7(x) = f_7(1, 1, \dots, 1) = 0$	[-50, 50]
$f_8(x) = -\sum_{i=1}^n (x_i \sin \sqrt{x_i})$ $\min f_8(x) = f_8(420.9687, 420.9687, \dots, 420.9687) = -418.9829n$	[-500, 500]

所示的常被用于评价优化算法性能的 8 个 Benchmark 函数、表 3 所示的 3 个二维测试函数，并且与遗传算法、鱼群算法、以及粒子群与膜计算相结合的其他算法进行对比，进而通过分析算法寻优精度与运行时间来检验所提 MC-MPSO 算法的寻优性能。

表 3 二维测试函数

二维测试函数	范围
$f_9(x, y) = -\cos x \cos y \cdot \exp^{-(x-\pi)^2 + (y-\pi)^2}$ $\min f_9(x, y) = f_9(\pi, \pi) = -1$	$-100 \leq x, y \leq 100$
$f_{10}(x, y) = (x^2 + y^2)^{0.25} [\sin^2(50(x^2 + y^2)^{0.1}) + 1.0]$ $\min f_{10}(x, y) = f_{10}(0, 0) = 0$	$-10 \leq x, y \leq 10$
$f_{11}(x, y) = -0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2}$ $\min f_{11}(x, y) = f_{11}(0, 0) = -1$	$-10 \leq x, y \leq 10$

上述 11 个函数三维图如图 3 所示。

图 3 11 个测试函数三维图

2.1 算法寻优能力测试

为了检验 MC-MPSO 算法的性能，选取原始 PSO 算法、标准 PSO 算法、MPSO 算法、EPSO 算法、MFPSO 算法、TFPSO 算法和 MC-MPSO 算法对上述测试函数进行测试对比。参数设置见表 4，MC-MPSO 算法各基本膜内的粒子数为 8。各算法运行 10 次的平均值优化结果曲线如图 4 所示。寻优结果如表 5 所示。

表 4 参数设置

参数	原始标准 PSO	标准 PSO	MPSO	EPSO	MFPSO	TFPSO	MC-MPSO
加速常数	1.49	1.49	1.49	1.49	1.49	1.49	1.49
c_1, c_2, c_3, c_f, c_g	1.49	1.49	1.49	1.49	1.49	1.49	1.49
最小惯性权重 $w_{\min}$	—	0.4	0.4	0.4	0.4	0.4	0.4
最大惯性权重 $w_{\max}$	—	0.9	0.9	0.9	0.9	0.9	0.9
切换因子 l_1	—	—	—	—	50	—	—
切换因子 l_2	—	—	—	—	350	—	—
膜内停滞次数阈值 g	—	—	—	—	—	—	8
进化停滞次数阈值 $l_{\max}$	—	—	—	—	—	200	200
最大迭代次数 $t_{\max}$	500	500	500	500	500	500	500
种群规模 N	48	48	48	48	48	48	48
种群维数 n	10	10	10	10	10	10	10

(c) $f_3(x)$ 优化结果曲线 (g) $f_7(x)$ 优化结果曲线 (d) $f_4(x)$ 优化结果曲线 (h) $f_8(x)$ 优化结果曲线 (e) $f_5(x)$ 优化结果曲线 (i) $f_9(x)$ 优化结果曲线 (f) $f_6(x)$ 优化结果曲线 (j) $f_{10}(x)$ 优化结果曲线

图4 函数优化结果曲线

由图 4 各算法寻优曲线可以看出, 函数中的 $f_2(x)$ 、 $f_5(x)$ 、 $f_6(x)$ 中, 原始 PSO 的寻优精度最低; 函数 $f_7(x)$ 、 $f_8(x)$ 中, 原始 PSO 的寻优精度能力处于中间; 函数 $f_1(x)$ 、 $f_3(x)$ 、 $f_4(x)$ 中, 原始 PSO 的寻优精

度最高。同理, 其他 5 个基本膜内的标准 PSO、MPSO、EPSO、MFPSO、TFPSO 算法分别针对某一类函数寻优效果好。而 MC-MPSO 算法对于这 11 个测试函数都表现出了较好的寻优能力。

由表 5 测试函数的结果看出, MC-MPSO 算法整体寻优精度最高; 由标准差来看, MC-MPSO 算法的波动性较小; 从运行时间看, MC-MPSO 算法略次于原始 PSO 算法、标准 PSO 算法、MPSO 算法, 但明显优于 EPSO 算法、MFPSO 算法、TFPSO 算法。综合分析算法的寻优精度、寻优速度和鲁棒性, MC-MPSO 算法优于其他六种算法, 因而, MC-MPSO 算法适应求解优化问题的范围更广泛。

2.2 与其他算法对比

2.2.1 与遗传算法对比

通过优化式(20), 与文献[24]遗传算法对比

$$f(x) = \frac{\sin(10\pi x)}{x} \quad 1 \leq x \leq 2 \quad (20)$$

表5 7种算法的寻优结果

算法	测试指标	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$	$f_{11}(x)$
原始 PSO	平均值	0*	2.5×10^4	0*	$8.9 \times 10^{-16*}$	9.5×10^{-1}	2.3×10^{-5}	2.0×10^{-1}	-3.5×10^3	-9.3×10^{-1}	0*	-1*
	标准差	0*	4.2×10^2	0*	0*	3.4×10^{-1}	2.9×10^{-5}	1.4×10^{-1}	3.6×10^2	2.2×10^{-1}	0*	0*
	时间/s	0.68	1.04	0.69	0.87	0.53*	0.65	0.67*	0.68*	0.17	0.16*	0.15*
标准 PSO	平均值	0*	5.5×10^2	0*	$8.9 \times 10^{-16*}$	0*	4.3×10^{-5}	2.4×10^{-1}	-3.2×10^3	-9.9×10^{-1}	0*	-1*
	标准差	0*	7.9×10^4	0*	0*	0*	4.7×10^{-5}	1.3×10^{-1}	6.0×10^2	5.1×10^{-5}	0*	0*
	时间/s	0.46*	0.89*	0.69*	0.85*	0.54	0.65*	0.77	0.73	0.17*	0.17	0.15
MPSO	平均值	4.0×10^{-15}	8.9	2.0×10^{-9}	1.3×10^{-6}	1.2×10^{-10}	3.1×10^{-8}	0.4	-2.1×10^3	-8.8×10^{-1}	2.8×10^{-16}	-1*
	标准差	2.5×10^{-15}	0.3×10^{-1}	3.2×10^{-9}	1.5×10^{-6}	1.5×10^{-10}	3.6×10^{-8}	0.3	3.3×10^2	1.5×10^{-1}	2.5×10^{-16}	0*
	时间/s	0.81	1.40	0.98	1.15	0.86	0.93	1.18	0.87	0.26	0.23	0.24
EPSO	平均值	8.7×10^{-8}	0.2×10^{-1}	5.9×10^{-4}	0.1×10^{-1}	3.2×10^{-4}	3.3×10^{-6}	2.5×10^{-4}	$-4.2 \times 10^{3*}$	-1*	0.8×10^{-2}	-1*
	标准差	1.7×10^{-7}	0.1×10^{-1}	8.3×10^{-4}	0.9×10^{-2}	6.1×10^{-4}	1.9×10^{-5}	4.6×10^{-4}	4.2×10^{-4}	1.1×10^{-5}	0.5×10^{-2}	2.7×10^{-9}
	时间/s	10.26	12.42	10.87	11.20	10.72	11.01	13.17	10.79	0.79	0.79	0.78
MFPSO	平均值	2.9×10^{-35}	$1.7 \times 10^{-5*}$	2.0×10^{-12}	1.9×10^{-5}	1.6×10^{-13}	2.8×10^{-28}	$1.1 \times 10^{-10*}$	$-4.2 \times 10^{3*}$	-1*	1.1×10^{-3}	-1*
	标准差	9.3×10^{-35}	1.3×10^{-5}	6.5×10^{-12}	4.8×10^{-5}	5.1×10^{-13}	3.5×10^{-28}	$3.3 \times 10^{-10*}$	$9.6 \times 10^{-13*}$	1.2×10^{-6}	2.2×10^{-3}	0*
	时间/s	19.88	24.73	21.86	21.52	20.53	21.06	25.20	19.39	1.38	1.42	1.46
TFPSO	平均值	1.3×10^{-7}	1.1×10^{-2}	3.6×10^{-4}	0.5×10^{-2}	6.8×10^{-4}	1.5×10^{-8}	6.0×10^{-6}	$-4.2 \times 10^{3*}$	-1*	0.7×10^{-2}	-1*
	标准差	3.2×10^{-7}	4.0×10^{-2}	5.0×10^{-4}	0.5×10^{-2}	1.7×10^{-3}	1.1×10^{-5}	1.9×10^{-5}	9.1×10^{-4}	4.6×10^{-5}	0.4×10^{-2}	3.4×10^{-9}
	时间/s	10.70	13.11	11.70	11.83	11.34	11.48	13.49	10.18	0.97	0.96	0.91
MC-MPSO	平均值	0*	2.9×10^{-5}	0*	$8.9 \times 10^{-16*}$	0*	$3.1 \times 10^{-30*}$	9.0×10^{-7}	$-4.8 \times 10^{3*}$	-1*	0*	-1*
	标准差	0*	$6.3 \times 10^{-6*}$	0*	0*	0*	$5.5 \times 10^{-30*}$	2.8×10^{-6}	4.9×10^{-4}	$5.7 \times 10^{-11*}$	0*	0*
	时间/s	2.06	2.98	2.47	2.70	2.21	2.41	3.03	2.21	0.55	0.47	0.46

注: 标有“*”的数值为最优值。

取 MC-MPSO 算法和遗传算法的最大进化代数均为 50, 每个膜内粒子数为 2, 进化停滞次数阈值 $l_{\max}$ 取 20, 其他参数与表 4 参数一致。对比曲线如图 5 所示, 结果见表 6。

表6 MC-MPSO 算法与遗传算法对比

算法	最优值	运行时间/s
遗传算法	-0.87	0.25
MC-MPSO 算法	-0.87	0.06

图 5 MC-MPSO 算法与遗传算法的对比曲线

由图 5 和表 6 可见, 两种算法最优值相同, 但 MC-MPSO 算法计算时间明显少于遗传算法。

2.2.2 与鱼群算法对比

通过优化式(21), 与文献[24]鱼群算法对比

$$f(x, y) = \frac{\sin x}{x} \frac{\sin y}{y} \quad -10 \leq x \leq 10 \quad y \leq 10 \quad (21)$$

取 MC-MPSO 算法和鱼群算法的最大进化代数均为 50, 每个膜内粒子数为 2, 进化停滞次数阈值 $l_{\max}$ 取 20, 其他参数与表 4 参数一致。对比曲线如图 6 所示, 结果见表 7。

图 6 MC-MPSO 算法与鱼群算法的对比曲线

表 7 MC-MPSO 算法与鱼群算法对比

算法	最优值	运行时间/s
鱼群算法	1	3.10
MC-MPSO	1	0.09

由图 6 和表 7 可以看出, 本文所提的 MC-MPSO 算法比鱼群算法收敛速度快, 在 MC-MPSO 算法迭代刚开始时就已搜索到最优值附近, MC-MPSO 算法运行时间明显少于鱼群算法。

2.2.3 与其他基于膜改进的粒子群算法对比

与文献[25]进行对比, PSOPS 算法和 HPSOPS 算法的种群规模为 30、最大迭代次数为 50, 各算法运行 30 次, 将本文所提的 MC-MPSO 算法与文献所提算法对比。运用测试函数 $f_2(x)$, 得到算法对比

数据见表 8。

表 8 MC-MPSO 算法与 PSOPS 算法、HPSOPS 算法的比较

算法	PSOPS 算法	HPSOPS 算法	MC-MPSO 算法
平均值	47.53	15.68	2.61×10^{-5}
平均时间/s	4.29	2.94	2.87

可以看出, MC-MPSO 算法搜索求解优于文献[25], 且运行时间也最短。

3 可靠性优化应用

串联系统和桥式系统是两种检验可靠性优化效果的常用系统, 下面对这两种系统进行可靠性优化, 通过算法对比来验证 MC-MPSO 算法的适应性。

3.1 串联系统

针对 5 级串联系统^[26], 利用原始 PSO、标准 PSO、MPSO、EPSO、MFPSP、TFPSO 和 MC-MPSO 算法分别进行可靠性优化, 各算法种群维数为 5, 各算法的参数选取见表 4, 优化结果如图 7 和表 9 所示。

图 7 串联系统可靠性优化曲线

表 9 串联系统可靠性优化结果

优化参数	原始 PSO 算法	标准 PSO 算法	MPSO 算法	EPSO 算法	MFPSP 算法	TFPSO 算法	MC-MPSO 算法
$P(x_1)$	0.160	0.160	0.172	0.151	0.163	0.145	0.163
$P(x_2)$	0.153	0.160	0.116	0.132	0.148	0.145	0.146
$P(x_3)$	0.127	0.110	0.183	0.132	0.114	0.145	0.130
$P(x_4)$	0.155	0.167	0.144	0.170	0.155	0.145	0.160
$P(x_5)$	0.119	0.121	0.129	0.132	0.137	0.145	0.112
C	174.558	174.797	174.920	174.851	175.000	174.829	175.000
$P(T)$	0.537	0.540	0.554	0.539	0.539	0.544	0.535
时间/s	0.014	0.013	0.017	0.011	0.015	0.012	0.002

3.2 桥式系统

针对桥式系统^[26], 利用原始 PSO、标准 PSO、MPSO、EPSO、MFPSP、TFPSO 和 MC-MPSO 算

法分别对该模型进行可靠性优化, 算法参数选取同上, 得到对应的优化结果如图 8 和表 10 所示。

图 8 桥式系统可靠性优化曲线

表 10 桥式系统可靠性优化结果

优化参数	原始 PSO	标准 PSO	MPSO	EPSO	MFPSO	TFPSO	MC-MPSO
$P(x_1)$	0.152	0.158	0.119	0.135	0.135	0.126	0.149
$P(x_2)$	0.102	0.100	0.126	0.135	0.135	0.126	0.112
$P(x_3)$	0.234	0.207	0.300	0.135	0.135	0.270	0.213
$P(x_4)$	0.151	0.158	0.167	0.135	0.135	0.126	0.150
$P(x_5)$	0.102	0.100	0.109	0.135	0.135	0.126	0.095
C	173.691	173.752	167.001	174.951	175.000	174.807	175.000
$P(T)$	0.036	0.036	0.041	0.040	0.040	0.038	0.036
时间/s	0.005	0.005	0.003	0.011	0.020	0.012	0.001

可见, MC-MPSO 算法的寻优能力均优于原始 PSO、标准 PSO、MPSO、EPSO、MFPSO、TFPSO、以及文献[26]混合 μ PSO 算法, 并且运行时间也远小于其他算法。

4 结论

(1) 针对单一一种粒子群算法在求解不同优化问题的适应性不足, 提出了膜计算多粒子群算法, 并提出基于质数拓扑结构的膜间载体交流机制, 提高了算法的搜索精度和计算效率。

(2) 将所提出的 MC-MPSO 算法进行了算法测试与对比, 并应用于串联系统、桥式系统的可靠性优化问题, 验证了 MC-MPSO 算法对于解决优化问题的可行性。

参 考 文 献

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//IEEE International Conference on Neural Networks, November 27-December 1, 1995, Perth, WA, Australia, IEEE, 1995: 1942-1948.
- [2] OUYANG H, GAO L, KONG X, et al. Hybrid harmony

search particle swarm optimization with global dimension selection[J]. Information Sciences, 2016, 6(346-347): 318-337.

- [3] 谢延敏, 孙新强, 田银, 等. 基于改进粒子群算法和小波神经网络的高强钢扭曲回弹工艺参数优化[J]. 机械工程学报, 2016, 52(19): 162-167.
- XIE Yanmin, SUN Xinqiang, TIAN Yin, et al. Optimization of parameters in twist springback process for high-strength sheets based on improved particle swarm optimization algorithm and wavelet neural network[J]. Journal of Mechanical Engineering, 2016, 52(19): 162-167.
- [4] 姚成玉, 王斌, 陈东宁, 等. 混合粒子交互微粒群算法[J]. 机械工程学报, 2015, 51(6): 198-207.
- YAO Chengyu, WANG bin, CHEN Dongning, et al. Hybrid-particle interaction particle swarm optimization algorithm[J]. Journal of Mechanical Engineering, 2015, 51(6): 198-207.
- [5] TANG M, XIN Y. Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization[J]. Computer Networks, 2016, 100: 1-11.
- [6] 王小巧, 刘明周, 葛茂根, 等. 基于混合粒子群算法的复杂机械产品装配质量控制阈优化方法[J]. 机械工程学报, 2016, 52(1): 130-138.
- WANG Xiaoqiao, LIU Mingzhou, GE Maogen, et al. Online control threshold optimization for complex mechanical products assembly process based on hybrid genetic particle swarm optimization[J]. Journal of Mechanical Engineering, 2016, 52(1): 130-138.
- [7] CHEN Dongning, ZHANG Ruixing, YAO Chengyu, et al. Dynamic topology multi force particle swarm optimization algorithm and its application[J]. Chinese Journal of Mechanical Engineering, 2016, 29(1): 124-135.
- [8] 周蓉, 沈维蕾, 刘明周, 等. 带时间窗装卸一体化车辆路径问题的混合离散粒子群优化算法[J]. 中国机械工程, 2016, 27(4): 494-502.
- ZHOU Rong, SHEN Weilei, LIU Mingzhou, et al. A hybrid discrete particle swarm optimization algorithm for vehicle routing problem with time windows and simultaneous pickup and deliver[J]. China Mechanical Engineering, 2016, 27(4): 494-502.
- [9] NGO T T, SADOLLAH A, KIM J K. A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems[J]. Journal of Computational Science, 2016, 13: 68-82.
- [10] EBERHART R C, SHI Y. Particle swarm optimization: developments, applications and resources[C]//IEEE

- Congress on Evolutionary Computation, May 27-30, 2001, Seoul, Korea, IEEE, 2001: 81-86.
- [11] 安鹏. 基于混沌理论和自适应惯性权重的 PSO 算法优化[J]. 吉林大学学报(理学版), 2015, 53(6): 1224-1228.
- AN Peng. Optimization of PSO algorithm based on chaotic theory[J]. Journal of Jilin University(Science Edition), 2015, 53(6): 1224-1228.
- [12] BEHESHTI Z, SHAMSUDDIN S M, HASAN S. MPSO: Median-oriented particle swarm optimization[J]. Applied Mathematics and Computation, 2013, 219(11): 5817-5836.
- [13] 莫思敏, 曾建潮, 谢丽萍. 扩展的微粒群算法[J]. 控制理论与应用, 2012, 29(6): 811-816.
- MO Simin, ZENG Jianchao, XIE Liping. Extended particle-swarm optimization algorithm[J]. Control Theory & Applications, 2012, 29(6): 811-816.
- [14] 陈东宁, 张瑞星, 姚成玉, 等. 求解液压阀块加工车间调度的多作用力微粒群算法[J]. 中国机械工程, 2015, 26(3): 369-378.
- CHEN Dongning, ZHANG Ruixing, YAO Chengyu, et al. A mfpso algorithm for solving hydraulic manifold processing shop scheduling[J]. China Mechanical Engineering, 2015, 26(3): 369-378.
- [15] YAO Chengyu, TAN Xueyun, CHEN Dongning, et al. Two-stage force particle swarm optimization algorithm and application in hydraulic system reliability optimization[C]// International Conference on Fluid Power and Mechatronics, August 5-7, 2015, Harbin, IEEE, 2015: 1291-1296.
- [16] PĂUN G, ROZENBERG G, SALOMAA A. The oxford handbook of membrane computing[M]. New York: Oxford University Press, 2010.
- [17] SINGH G, DEEP K, NAGAR A K. Cell-like P-systems based on rules of particle swarm optimization[J]. Applied Mathematics and Computation, 2014, 246: 546-560.
- [18] ZHANG G, GHEORGHE M, PAN L, et al. Evolutionary membrane computing: a comprehensive survey and new results[J]. Information Sciences, 2014, 279: 528-551.
- [19] 黄亮. 膜计算优化方法研究[D]. 杭州: 浙江大学, 2007.
- HUANG Liang. Research on membrane computing optimization methods[D]. Hangzhou: Zhejiang University, 2007.
- [20] 欧柱. 细胞型膜计算模型遗传优化设计研究[D]. 成都: 西南交通大学, 2013.
- OU Zhu. Genetic optimization design of cell-like membrane computing models[D]. Chengdu: Southwest Jiaotong University, 2013.
- [21] 廖孝勇, 李尚键, 孙棣华, 等. 一种基于膜计算的改进人工鱼群算法[J]. 小型微型计算机系统, 2014, 35(5): 1142-1146.
- LIAO Xiaoyong, LI Shangjian, SUN Dihua, et al. An improved artificial fish swarm algorithm based on membrane computing[J]. Journal of Chinese Computer Systems. 2014, 35(5): 1142-1146.
- [22] 杜强, 向来生, 刘希玉. 基于 P 系统的粒子群优化算法[J]. 计算机应用研究, 2013, 30(8): 2269-2272.
- DU Qiang, XIANG Laisheng, LIU Xiyu. P system based particle swarm optimization algorithm[J]. Application Research of Computers, 2013, 30(8): 2269-2272.
- [23] ZHOU Fen, ZHANG Gexiang, RONG Haina, et al. A particle swarm optimization based on P systems[C]// International Conference on Natural Computation, August 10-12, 2010, Yantai, IEEE, 2010: 3003-3007.
- [24] 史峰, 王辉, 郁磊, 等. MATLAB 智能算法 30 个案例分析[M]. 北京: 北京航空航天大学出版社, 2011.
- SHI Feng, WANG Hui, YU Lei, et al. 30 cases of MATLAB intelligent algorithm[M]. Beijing: Beihang University Press, 2011.
- [25] 周芬. 粒子群膜算法及其应用研究[D]. 成都: 西南交通大学, 2011.
- ZHOU Fen. Particle swarm optimization membrane algorithm and its application research[D]. Chengdu: Southwest Jiaotong University, 2011.
- [26] 陈东宁, 姚成玉. 基于 T-S 故障树和混合 μ PSO 算法的可靠性优化方法[J]. 中国机械工程, 2013, 24(18): 2415-2420.
- CHEN Dongning, YAO Chengyu. Reliability optimization method based on T-S fault tree and hybrid μ PSO algorithm[J]. China Mechanical Engineering, 2013, 24(18): 2415-2420.

作者简介: 陈东宁(通信作者), 女, 1978 年出生, 博士, 副教授。主要研究方向为可靠性分析及优化。

E-mail: dncchen@ysu.edu.cn

王跃颖, 女, 1989 年出生, 硕士研究生。主要研究方向为群智能优化算法及应用。

姚成玉, 男, 1975 年出生, 博士后, 教授。主要研究方向为系统可靠性及故障诊断。

刘一丹, 女, 1992 年出生, 硕士研究生。主要研究方向为群智能优化。

吕世君, 男, 1975 年出生, 博士研究生, 讲师。主要研究方向为系统可靠性分析。