

基于收敛速度和多样性的 多目标粒子群种群规模优化设计

韩红桂^{1,2}, 武淑君^{1,2}

(1. 北京工业大学信息学部, 北京 100124; 2. 计算智能与智能系统北京市重点实验室, 北京 100124)

摘 要: 针对多目标粒子群优化算法种群规模难以确定的问题, 文中提出了一种基于收敛速度和多样性的多目标粒子群优化 (Convergence speed and Diversity-based Multi-Objective Particle Swarm Optimization, CD-MOPSO) 算法. 首先, 利用优化过程的收敛速度和多样性指标构造种群规模适应度函数, 完成了种群规模与优化性能关系的描述; 其次, 基于适应度函数设计了一种种群规模自适应调整方法, 实现了种群规模的动态调整; 最后, 将提出的 CD-MOPSO 在基准优化问题 ZDT 上测试并应用于城市管网优化, 实验结果显示 CD-MOPSO 能够根据求解问题自动调整种群规模, 与 NSGA-II、MOPSO、SPEA2 和 EMDS-MOPSO 相比具有更快的收敛速度和更好的优化结果.

关键词: 多目标粒子群优化算法; 种群规模; 自适应调整方法; 动态调整; 适应度函数; 收敛速度; 多样性; 基准测试函数; 城市管网优化

中图分类号: TP18

文献标识码: A

文章编号: 0372-2112 (2018)09-2263-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2018.09.031

Design of Population Size for Multi-objective Particle Swarm Optimization Algorithm Based on the Convergence Speed and Diversity

HAN Hong-gui^{1,2}, WU Shu-jun^{1,2}

(1. Department of Information, Beijing University of Technology, Beijing 100124, China;

2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China)

Abstract: To determine the population size of multi-objective particle swarm optimization algorithm (MOPSO), an improved MOPSO, based on the convergence speed and diversity, named CD-MOPSO, is proposed. Firstly, the fitness function of population size, which is developed by the convergence speed and diversity during the evolutionary process, is used to describe the relationship between the population size and the performance of MOPSO. Secondly, according to the fitness function, an adaptive adjustment method is designed to update the population size of MOPSO dynamically. Finally, the proposed CD-MOPSO is tested on the ZDT benchmark optimization problems and applied to a real optimization problem of urban pipe networks. The experimental results show that the proposed CD-MOPSO can adjust the population size automatically according to the problem, compared with the performance of NSGA, MOPSO, SPEA2 and EMDS-MOPSO, CD-MOPSO has faster convergence speed with better optimization results.

Key words: multi-objective particle swarm optimization algorithm (MOPSO); population size; adaptive adjustment method; dynamic adjustment; fitness function; convergence speed; diversity; benchmark test functions; urban pipe networks optimization

1 引言

随着社会的不断发展, 越来越复杂的优化问题涌现出来, 这些优化问题由多个相互冲突的目标构成-多

目标优化问题 (Multi-Objective Optimization Problem, MOP)^[1]. 传统的优化方法在求解 MOP 上存在计算复杂度高、计算时间长等问题^[2], 而多目标粒子群优化算法 (Multi-Objective Particle Swarm Optimization, MOPSO) 具

有流程简单、收敛速度快、搜索效率高等特点,更适用于求解 MOP^[3].

近年来, Fereshteh 等人提出了一种基于拥挤距离的 MOPSO (Multi-Objective Particle Swarm Optimization based on Crowding Distance, MOPSO-CD)^[4], 该算法通过引入一种突变操作来改善非支配解的多样性. Daneshyari 等人引入文化框架的概念提出了一种变飞行参数的 MOPSO, 与其他优化算法相比该算法有更好优化结果^[5]. 以上改进型 MOPSO 在解决 MOP 上性能有所提高, 然而这些算法的种群规模在进化过程中始终是固定的. 进化过程中算法的种群规模和参数调整共同影响 MOPSO 的性能^[6], 在 MOPSO 参数调节上, 众多学者做了大量研究, 而针对 MOPSO 种群规模的研究很少, 因此, 如何在进化过程中动态调整算法的种群规模仍是 MOPSO 的一个瓶颈.

为了解决优化算法中种群规模难以确定的问题, 许多学者针对不同的优化算法进行了初步尝试. 例如, Koumoussis 等人设计了一种基于锯齿波的变种群规模的遗传算法 (Sawtooth Genetic Algorithm, Saw-Tooth GA)^[7], Brest 等人提出了一种自适应调整种群规模的差分进化算法^[8], Liu 等人基于最优解设计了一种新的蚁群优化算法^[9], Chen 等人提出了一种基于多样性的粒子群优化算法^[10], 以上四种算法根据不同优化算法的特点设计出不同的种群规模调整方法. 然而, 围绕 MOPSO 种群规模设计问题的研究甚少, Leong 等人提出了一种动态 MOPSO (Dynamic population multiple-swarm MOPSO, DMOPSO)^[11], 该算法能够动态调整种群规模, 但是需要额外设置参数. 因此, 如何设计出有效地种群规模自适应调整方法仍是 MOPSO 的一个重要问题.

基于以上分析, 文中设计了一种基于收敛速度和多样性的 MOPSO (Convergence speed and Diversity-based Multi-Objective Particle Swarm Optimization, CD-MOPSO), 该算法结合收敛速度和多样性对种群规模和优化性能之间的关系进行了描述, 给出了 MOPSO 的种群规模适应度函数, 基于适应度函数设计了一种自适应的种群规模调整方法, 提高了 MOPSO 的性能. 在标准测试函数和城市管网优化上的测试结果证明了该算法的有效性.

2 MOPSO

2.1 多目标优化问题 (MOP)

MOP 需要同时优化两个或两个以上的目标函数, 而且这些目标经常发生冲突, 最小化一个 m 维目标的 MOP 数学描述如下^[12,13]:

$$\min F(x) = [f_1(x), f_2(x), \dots, f_m(x)] \quad (1)$$

$$\text{s.t. } g_i(x) \geq 0, i = 1, 2, \dots, p \quad (2)$$

$$h_j(x) = 0, j = 1, 2, \dots, q \quad (3)$$

式中, x 为决策变量, $F(x)$ 为优化目标, $f_i(x)$ 为第 i 个目标函数, m 为目标函数的个数; $g_i(x)$ 为第 i 个不等式约束, p 为不等式约束的个数; $h_j(x)$ 为第 j 个等式约束, q 为等式约束的个数.

2.2 MOPSO

MOPSO 是一种基于粒子之间信息共享的启发式算法^[14], 种群中的每个粒子代表待优化问题的一个潜在解, 算法通过粒子在解空间内追随最优粒子的行为迭代搜索最优解^[15].

粒子的位置和速度更新公式如下:

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

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 (p_{i,d}(t) - x_{i,d}(t)) + c_2 r_2 (g_d(t) - x_{i,d}(t)) \quad (5)$$

式中, t 为进化代数, $d \in D$ 为搜索空间的第 d 维, ω 为惯性权重, c_1 和 c_2 为学习因子, r_1 和 r_2 为 $[0, 1]$ 之间均匀分布的随机数; $x_{i,d}(t)$ 为第 t 次迭代时粒子的位置信息, $v_{i,d}(t)$ 为第 t 次迭代时粒子的速度信息, $p_{i,d}(t)$ 为第 t 次迭代时粒子的历史最优位置, $g_d(t)$ 为第 t 次迭代时的全局最优位置.

3 CD-MOPSO

文中提出的 CD-MOPSO 首先基于迭代过程中的收敛速度和多样性对算法的种群规模与优化性能之间的关系进行了描述, 获得了种群规模适应度函数; 其次根据种群规模适应度函数设计了一种自适应调节算法种群规模的方法.

3.1 算法种群规模与优化性能关系分析

3.1.1 收敛速度指标

收敛速度指标是描述算法优化性能的重要指标之一. 文中根据算法迭代过程中收敛速度的变化完成了算法种群规模与优化性能关系的描述.

定义 1 (收敛速度) 算法连续三次迭代产生的 Pareto 前沿之间的最小欧式距离的平均值的差值称为算法的收敛速度. 计算公式为:

$$S_{B_{i+1}} = \frac{\sqrt{\sum_{i=1}^{n_i} d_i^2}}{n_{i+1}} \quad (6)$$

$$S = S_{B_{i+1}} - S_{B_i} \quad (7)$$

式中, n_{i+1} 为算法第 $i+1$ 次迭代产生的非支配解的个数, B_{i-1}, B_i, B_{i+1} 为算法连续三次迭代产生的 Pareto 前沿, d_i 为前沿 B_{i+1} 中第 i 个解到前沿 B_i 的最小欧式距离, $S_{B_{i+1}}$ 为前沿 B_{i+1} 与前沿 B_i 之间最小的欧式距离的平均值.

文中提出的收敛速度指标通过计算前沿之间欧式距离的差值判断算法的收敛速度. 因此, 该指标需要连

续三次迭代产生的 Pareto 前沿来计算. 若 $S > 0$, 则说明当前次迭代算法的收敛速度相对较快, 算法的种群规模较小; 反之, 说明当前次迭代算法的收敛速度相对较慢, 算法的种群规模较大.

3.1.2 多样性指标

多样性指标是描述算法优化性能的重要组成部分. 文中通过迭代过程中多样性值的变化给出了非支配解的分布情况与种群规模的关系.

定义 2 (多样性) 算法产生的 Pareto 前沿中非支配解的分布情况称为算法的多样性. 计算公式为:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (8)$$

式中, d_i 为算法求得的第 i 个非支配解到与它最近的非支配解之间的欧式距离, \bar{d} 为 d_i 的平均值. 多样性指标反映算法产生的 Pareto 前沿中非支配解的分布情况, 粒子在解空间内分布的越均匀, SP 的值越小, 说明算法的多样性越好^[16,17].

算法多样性变化的数学描述为:

$$C = SP_i - SP_{i-1} \quad (9)$$

式中, C 为算法种群多样性变化的指标, SP_i 为算法第 i 次迭代产生的 Pareto 前沿的种群多样性值.

若 $C > 0$, 说明当前次迭代算法的种群多样性相对较差, 算法的种群规模较小; 反之, 则说明算法的种群多样性相对较好, 种群规模较大.

3.2 种群规模适应度函数

基于种群规模与优化性能关系的描述, 文中设计了一种种群规模适应度函数, 描述为:

$$S > 0 \& C > 0 \quad (10)$$

$$S \leq 0 \& C < 0 \quad (11)$$

$$S > 0 \& C < 0 \quad (12)$$

$$S \leq 0 \& C > 0 \quad (13)$$

种群规模适应度函数描述如表 1 所示.

文中通过种群规模适应度函数来判断算法的种群规模大小. 当算法连续五次满足式 (12), 则算法停止计算该适应度函数, 由于算法朝好的方向进化, 因此理论上不存在式 (13) 的情况. 该方法不仅为调整算法种群规模提供了依据, 同时尽可能减少不必要的计算步骤.

表 1 种群规模适应度函数分析

适应度函数	收敛速度	多样性	种群规模情况	执行策略
式 (10)	变快	变差	较小	增加种群规模
式 (11)	变慢	变好	较大	减少种群规模
式 (12)	变快	变差	大小合适	种群规模不变
式 (13)	变慢	变差	不考虑	不考虑

文中提出的 CD-MOPSO 是变种群规模的算法, 因

此无法进行准确的复杂度分析. 虽然种群规模适应度函数增加了算法的计算复杂度, 但与种群规模引起的变化相比该适应度函数对算法的计算复杂度影响不大.

3.3 种群规模自适应调整方法

基于种群规模适应度函数文中提出了一种自适应调整种群规模的方法. 该方法设计了种群规模增加和减少机制, 可根据不同的优化问题自动调整算法的种群规模.

3.3.1 种群规模增加机制

种群规模增加机制是为了增加算法中粒子的数目, 使搜索空间里有更多的粒子, 减少算法陷入局部最优的情况, 改善种群多样性.

种群规模增加机制为:

$$N(t+1) = N(t) + k_1 \times N_s(t) \quad (14)$$

式中, t 代表第 t 次迭代; k_1 为区间 $[0, 1]$ 之间的随机数; $N(t)$ 表示第 t 次迭代时算法的种群规模; $N_s(t)$ 为算法第 t 次迭代时产生的非支配解的个数.

种群规模增加机制增加的粒子为目标空间中非支配解对应的粒子. 当算法执行种群规模增加机制时, 从未被支配的粒子中随机选择部分粒子, 将这些粒子的位置信息赋给新增加的粒子, 从而提高新增加粒子的有效性.

3.3.2 种群规模减少机制

种群规模减少机制是为了减少算法中粒子的数目, 从而加快算法的收敛速度, 减少算法的计算复杂度.

种群规模减少机制为:

$$N(t+1) = N(t) - k_2 \times (N(t) - N_s(t)) \quad (15)$$

式中, k_2 为区间 $[0, 1]$ 之间的随机数.

种群规模减少机制减少的粒子为目标空间中被支配的解对应的粒子. 当算法执行种群规模减少机制时, 删除部分被支配的粒子, 从而加快算法的收敛速度, 在一定程度上改进算法的性能.

3.4 CD-MOPSO

基于以上分析, 文中提出的 CD-MOPSO 的具体流程如算法 1 所示.

算法 1 CD-MOPSO

输入: 随机初始化一个包含 N 个粒子的种群, 初始化粒子的位置 $x_i(0)$ 、速度 $v_i(0)$ 、惯性权重 w 、学习因子 c_1 和 c_2 . 将各粒子的初始位置设为当前历史最优位置 $p_i(0)$, 从初始外部档案中选取全局最优位置 $g(0)$;

1. 根据式 (4) 和式 (5) 更新粒子的位置 $x_i(t)$ 和速度 $v_i(t)$;
2. 评估每个粒子的适应度值, 进行非支配排序;
3. 更新外部档案;
4. 更新个体最优位置 $p_i(t)$ 、全局最优位置 $g(t)$;
5. 根据式 (6)、(7)、(8) 和式 (9) 计算 CS 和 CSP 值;
6. 根据式 (10)、(11)、(12) 和 (13) 判断当前次迭代算法应使用种群

规模增加或减少机制;

7. 根据式(14)或式(15)自动调整算法的种群规模.

4 仿真实验与分析

文中将 CD-MOPSO 在标准测试函数上测试并应用于双环管网优化,同时将该算法与其他优化算法的测试结果进行比较,实验结果验证了该算法的有效性.文中对比算法的参数均参照对应的参考文献.

4.1 标准测试函数

文中选用了标准测试函数 ZDT-1、ZDT-2、ZDT-3 和 ZDT-4 来测试 CD-MOPSO 的性能.将 CD-MOPSO 与 NSGA-II^[18]、固定种群规模的 MOPSO^[19]、SPEA2^[20] 以及 EMDS-MOPSO^[21] 的测试数据进行比较.五种优化算法的初始种群规模和最大迭代次数均设为 200,每种算法在测试函数上均独立运行 10 次.

4.1.1 实验结果及比较分析

从表 2 中可以看出 CD-MOPSO 可根据不同测试函数调整算法的种群规模;与 NSGA-II、MOPSO、SPEA2 和 EMDS-MOPSO 相比,该算法具有更好的多样性值(除 ZDT-1、ZDT-2 和 ZDT-3 函数时,固定种群规模的 MOPSO 有更好的 SP 值)和更好的收敛性值(除 ZDT-4 函数时,EMDS-MOPSO 有更好的 GD 值);此外,从表中可以看出该算法的平均运行时间比其他四种算法的运行时间短很多.

此外,为分析 CD-MOPSO 的性能与最大迭代次数的关系,文中将 CD-MOPSO 的最大迭代次数设为 200、500、1000,分别在四个基准测试函数上均独立运行 10 次,并将产生的结果进行对比,对比结果如表 3 所示.

从表 3 中的数据可以看出,当最大迭代次数变为 500 或 1000 时,CD-MOPSO 的性能(多样性-SP 值、收敛性-GD 值)与最大迭代次数为 200 时算法的性能相比无明显变化,但是算法的运行时间随最大迭代次数的增大而增加.

4.2 双环管网优化

城市给水管网的优化设计是在一定的约束条件下,通过平衡经济性、可靠性、有效性等多个目标之间的

冲突,寻求全局最优解^[22,23].

4.2.1 问题描述

文中选取了国际经典的双环管网^[24]来测试 CD-MOPSO 的性能,双环管网的物理模型如图 1 所示.

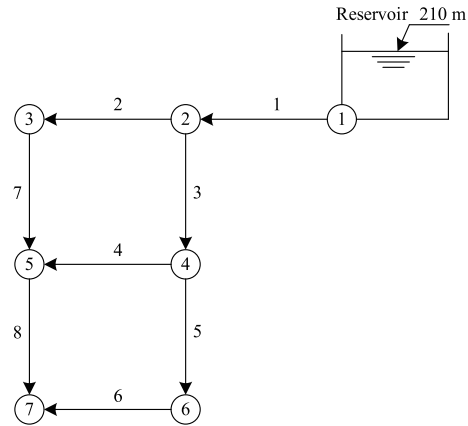


图1 双环管网

双环管网的函数表达式为:

$$\min f_1 = \sum_{j=1}^R (a + b d_j^\alpha) L_j, j = 1, 2, \dots, R \quad (16)$$

$$\min f_2 = \sum_{i=1}^I (I_{si} - \bar{I}_s) L_j, i = 1, 2, \dots, I \quad (17)$$

式中, f_1 为管网的经济性目标函数, R 为管道的总数, L_j 代表第 j 段管道的长度, d_j 代表第 j 段管道的直径, a 、 b 、 α 为统计常数与指数; f_2 为管网的可靠性目标函数, I 为管网系统节点数目, I_{si} 为第 i 个节点的富余水头值, \bar{I}_s 为节点富余水头的均值.

双环管网约束条件包含节点连续性方程、能量方程以及压降方程,其函数表达式为:

$$\text{s. t. } \mathbf{A}\mathbf{q} + \mathbf{Q} = \mathbf{0}, \mathbf{L}\mathbf{h} = \mathbf{0}, \mathbf{h} = \mathbf{s}\mathbf{q}^n \quad (18)$$

$$d_k \in \{d_1, d_2, \dots, d_z\} \quad (19)$$

式中, \mathbf{A} 为节点的衔接矩阵, \mathbf{q} 为管网系统中的管段流量矩阵, \mathbf{Q} 为管网系统中的节点流量; \mathbf{L} 为回路矩阵, \mathbf{h} 为管网系统中的管段水头损失矩阵, \mathbf{s} 为管网系统中的管段摩阻矩阵; z 为可选标准管径的总数目, $n = 1.852 \sim 2$, 根据采用水头公式不同而定.

表 2 五种优化算法对四种不同测试函数的测试结果

测试函数	优化算法	评价指标				
		N_0	N	SP	GD	T
ZDT-1	NSGA-II ^[18]	200	200	5.023E-01	4.572E-03	131.659
	MOPSO ^[19]	200	200	4.455E-03	7.434E-05	23.778
	SPEA2 ^[20]	200	200	6.236E-01	8.218E-03	24.005
	EMDS-MOPSO ^[21]	200	200	2.171E-01	6.845E-05	23.956
	CD-MOPSO	200	124	5.074E-03	6.352E-05	16.079

续表

测试函数	优化算法	评价指标				
		N_0	N	SP	GD	T
ZDT-2	NSGA-II ^[18]	200	200	7.067E-01	1.218E-02	177.461
	MOPSO ^[19]	200	200	4.134E-03	7.231E-05	37.646
	SPEA2 ^[20]	200	200	8.012E-01	6.146E-02	35.423
	EMDS-MOPSO ^[21]	200	200	2.086E-01	8.811E-05	32.335
	CD-MOPSO	200	146	4.257E-03	6.547E-05	26.196
ZDT-3	NSGA-II ^[18]	200	200	3.852E-01	2.741E-03	136.887
	MOPSO ^[19]	200	200	2.056E-02	8.859E-04	20.816
	SPEA2 ^[20]	200	200	4.584E-01	5.058E-03	32.337
	EMDS-MOPSO ^[21]	200	200	1.396E-01	3.361E-05	30.865
	CD-MOPSO	200	85	3.088E-02	5.633E-04	13.278
ZDT-4	NSGA-II ^[18]	200	200	9.772E-01	4.858E-01	77.065
	MOPSO ^[19]	200	200	5.572E-03	7.737E-04	20.108
	SPEA2 ^[20]	200	200	8.576E-01	1.877E+0	30.632
	EMDS-MOPSO ^[21]	200	200	2.066E-01	1.103E-04	28.658
	CD-MOPSO	200	91	4.551E-03	2.499E-04	13.717

表 3 不同最大迭代次数下 CD-MOPSO 的测试结果

测试函数	最大迭代次数	评价指标				
		N_0	N	SP	GD	T
ZDT-1	200	200	124	5.074E-03	6.352E-05	16.079
	500	200	112	5.216E-03	7.255E-05	25.663
	1000	200	120	4.84E-03	8.011E-05	40.345
ZDT-2	200	200	146	4.257E-03	6.547E-05	26.196
	500	200	140	4.457E-03	6.676E-05	45.376
	1000	200	150	4.195E-03	6.768E-05	55.185
ZDT-3	200	200	85	3.088E-02	5.633E-04	13.278
	500	200	79	4.386E-02	8.003E-04	33.898
	1000	200	84	1.207E-02	7.243E-04	42.844
ZDT-4	200	200	91	4.551E-03	2.499E-04	13.717
	500	200	89	3.741E-03	2.285E-04	39.147
	1000	200	105	4.741E-03	2.851E-04	46.769

4.2.2 实验结果

双环管网的优化实验中,算法的初始种群规模和最大迭代次数均设为 200,算法均独立运行 10 次.

从图 2 中可以看出 CD-MOPSO 和 MOPSO 得到的 Pareto 前沿更逼近真实前沿,且 CD-MOPSO 的种群规模远少于 MOPSO 的种群规模.

从表 4 中可以看出 NSGA-II、MOPSO、CD-MOPSO 在双环管网上测试得到的 SP 值分别为 2.206237E-05、7.022404E-07 和 4.312183E-07,说明 CD-MOPSO 在双

环管网优化上有更好的多样性值.

表 4 NSGA-II、MOPSO 和 CD-MOPSO 的 SP 值

多样性(SP)	NSGA-II ^[18]	MOPSO ^[19]	CD-MOPSO
最大值	7.865242E-05	8.652854E-08	5.645235E-07
最小值	2.206237E-05	7.022404E-07	4.312183E-07
平均值	6.768522E-05	7.365562E-07	4.562342E-07

根据以上分析,可以看出 CD-MOPSO 和 MOPSO 在双环管网上优化得到的 Pareto 解比 NSGA-II 优化得到

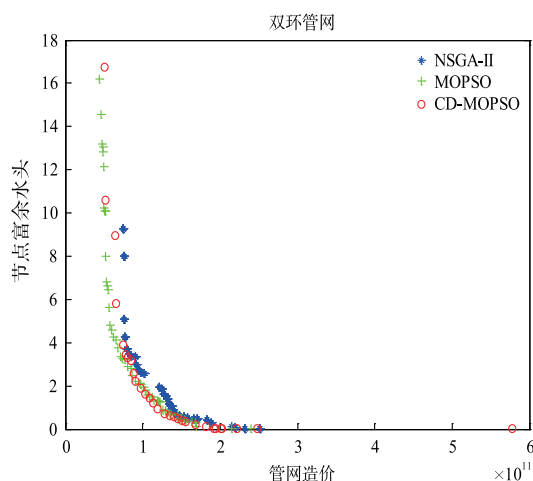


图2 NSGA-II、MOPSO、CD-MOPSO对双环管网的测试结果

的 Pareto 解有更好的适应度值,CD-MOPSO 与其他算法相比有更好的多样性值。

5 结论

针对 MOPSO 种群规模难以确定的问题,文中通过设计种群规模适应度函数和种群规模调整方法提出 CD-MOPSO。该算法可根据不同的 MOP 调整算法的种群规模,有效地平衡了算法收敛速度和多样性之间的矛盾,并且在标准测试函数和双环管网上的实验结果证明该算法可以加快算法的收敛速度,同时增强算法的性能。

参考文献

- [1] Xue B, Zhang M, Browne W N. Particle swarm optimization for feature selection in classification: A multi-objective approach[J]. IEEE Transactions on Cybernetics, 2013, 43(6): 1656 – 1671.
- [2] Von Lücken C, Barán B, Brizuela C. A survey on multi-objective evolutionary algorithms for many-objective problems[J]. Computational Optimization and Applications, 2014, 58(3): 707 – 756.
- [3] 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.
XIE Cheng-wang, ZOU Xiu-fen, XIA Xue-weng, et al. A multi-objective particle swarm optimization algorithm integrating multiply strategies[J]. Acta Electronica Sinica, 2015, 43(8): 1538 – 1544. (in Chinese)
- [4] Sheikholeslami F, Navimipour N J. Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance[J]. Swarm and Evolutionary Computation, 2017, 35(4): 53 – 64.
- [5] Daneshyari W, Yen G G. Cultural MOPSO: A cultural framework to adapt parameters of multiobjective particle swarm optimization[A]. IEEE Congress on Evolutionary Computation[C]. Hong Kong: IEEE, 2008. 1325 – 1332.
- [6] Leong W F, Yen G G. Dynamic population size in PSO-based multiobjective optimization[A]. IEEE Congress on Evolutionary Computation[C]. Canada: IEEE, 2006. 1718 – 1725.
- [7] Koumoussis V K, Katsaras C P. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(1): 19 – 28.
- [8] Brest J, Zamuda A, Boskovic B, et al. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction[A]. IEEE Congress on Evolutionary Computation[C]. Hong Kong: IEEE, 2008. 2032 – 2039.
- [9] Liu Y, Liu J, Li X, et al. A Self-Adaptive control strategy of population size for ant colony optimization algorithms[A]. International Conference in Swarm Intelligence[C]. Bali: Springer International Publishing, 2016. 443 – 450.
- [10] Chen D B, Zhao C X. Particle swarm optimization with adaptive population size and its application[J]. Applied Soft Computing, 2009, 9(1): 39 – 48.
- [11] Leong W F, Yen G G. PSO-based multiobjective optimization with dynamic population size and adaptive local archives[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008, 38(5): 1270 – 1293.
- [12] Marler R T, Arora J S. Survey of multi-objective optimization methods for engineering[J]. Structural and Multidisciplinary Optimization, 2004, 26(6): 369 – 395.
- [13] Durillo J J, Nebro A J. jMetal: A Java framework for multi-objective optimization[J]. Advances in Engineering Software, 2011, 42(10): 760 – 771.
- [14] Lalwani S, Singhal S, Kumar R, et al. A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm[J]. Transactions on Combinatorics, 2013, 2(1): 39 – 101.
- [15] Taherkhani M, Safabakhsh R. A novel stability-based adaptive inertia weight for particle swarm optimization[J]. Applied Soft Computing, 2016, 38: 281 – 295.
- [16] Lim D K, Woo D K, Yeo H K, et al. A novel surrogate-assisted multi-objective optimization algorithm for an electromagnetic machine design[J]. IEEE Transactions on Magnetics, 2015, 51(3): 1 – 4.
- [17] Li L, Wang W, Xu X. Multi-objective particle swarm optimization based on global margin ranking[J]. Information Sciences, 2017, 375: 30 – 47.

- [18] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6 (2): 182 – 197.
- [19] Tripathi P K, Bandyopadhyay S, Pal S K. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients [J]. Information Sciences, 2007, 177 (22): 5033 – 5049.
- [20] Bleuler S, Brack M, Thiele L, et al. Multiobjective genetic programming: Reducing bloat using SPEA2 [A]. IEEE Congress on Evolutionary Computation [C]. South Korea: IEEE, 2001. 536 – 543.
- [21] Fan S K S, Chang J M, Chuang Y C. A new multi-objective particle swarm optimizer using empirical movement and diversified search strategies [J]. Engineering Optimization, 2015, 47 (6): 750 – 770.
- [22] Prasad T D, Hong S H, Park N. Reliability based design of water distribution networks using multi-objective genetic algorithms [J]. KSCE Journal of Civil Engineering, 2003, 7 (3): 351 – 361.
- [23] Cisty M. Hybrid model for water distribution design [A]. IEEE Congress on Evolutionary Computation [C]. Barcelona: IEEE, 2010. 1 – 8.
- [24] Alperovits E, Shamir U. Design of optimal water distribution systems [J]. Water Resources Research, 1977, 13 (6): 885 – 900.

作者简介



韩红桂 男, 1983 年 08 月生于江苏泰州. 现为北京工业大学教授、博士生导师. 主要研究方向为城市污水处理过程智能优化控制.

E-mail: rechardhan@bjut.edu.cn



武淑君 女, 1995 年 01 月生于河南邓州. 现为北京工业大学信息学部硕士研究生. 主要研究方向为污水处理过程智能优化.

E-mail: wushujun@emails.bjut.edu.cn