

引用格式:王之仓,李和成.采用多项式变异策略和分解方法的多目标进化算法[J].微电子学与计算机,2021,38(1):96-100.
[WANG Z C, LI H C. Multi-objective evolutionary algorithm using decomposition method and polynomial mutation operator [J]. Microelectronics & Computer, 2021, 38(1): 96-100.]

采用多项式变异策略和分解方法的多目标进化算法

王之仓^{1,3}, 李和成²

(1 青海师范大学 计算机学院, 青海 西宁 810008; 2 青海师范大学 数学与统计学院,
青海 西宁 810008; 3 西安邮电大学 自动化学院, 陕西 西安 710021)

摘 要: MOEA/D-M2M 算法将一个多目标优化问题同时转换为若干个多目标优化子问题, 分别求得这些子问题的 Pareto 解, 最终得到原多目标优化问题的 Pareto 解, 保证了种群的多样性, 比 MOEA/D 具有更好的算法性能. 多项式变异算子具有强化局部搜索和加强收敛的作用, 但是在多目标进化算法中应用多项式变异算子的成果不多. 将多项式变异算子和 MOEA/D-M2M 中提出的多目标优化问题的新的分解方法相结合, 提出了一种新的采用多项式变异策略和分解方法的多目标进化算法(MOEA/PmD). 考虑到非均匀变异算子通过动态调整步长获得自适应性的特点, 尝试通过将非均匀变异算子替换 MOEA/PmD 中的多项式变异算子而构造采用非均匀变异算子和分解方法的多目标进化算法(MOEA/NumD). 实验表明 MOEA/PmD 算法比 MOEA/NumD 算法和 MOEA/D-M2M 算法具有更好的性能.

关键词: 分解方法; 多目标进化算法; 多项式变异算子; 非均匀变异算子

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2021)01-0095-06

Multi-objective evolutionary algorithm using decomposition method and polynomial mutation operator

WANG Zhi-cang^{1,3}, Li He-cheng²

(1 School of Computer Science, Qinghai Normal University, Xining 810008, China;

2 School of Mathematics and Statistics, Qinghai Normal University, Xining 810008, China;

3 School of Automation, Xi'an University of Post & Telecommunications, Xi'an 710121, China)

Abstract: The algorithm of MOEA/D-M2M can transform multi-objective optimization problem into a number of multi-objective optimization sub-problems and obtains the Pareto solutions of these sub-problems separately, and finally obtains the Pareto solution of the multi-objective optimization problem, which ensures the diversity of population and has better algorithm performance than MOEA/D. Polynomial mutation operator has the function of strengthening local search and convergence, but there is less work on the application of the mutation operator in the multi-objective evolutionary algorithm. A new multi-objective evolution algorithm using decomposition methods and polynomial mutation operators (MOEA/PmD) is proposed by combining the polynomial mutation operator and the new decomposition method of multi-objective optimization problem proposed in MOEA/D-M2M. Non-uniform mutation operator obtains the adaptability by dynamically adjusting the step size, an attempt is done to replace the polynomial mutation operator in MOEA/PmD with the non-uniform mutation operator, and multi-objective evolutionary algorithm using non-uniform mutation operator and decomposition method (MOEA/NumD) is devised. Experiments show MOEA/PmD algorithm has better performance than

收稿日期: 2020-05-26; 修回日期: 2020-06-01

基金项目: 国家自然科学基金(61966030); 教育部春晖计划项目(Z2017052); 青海省自然科学基金(2018-ZJ-901); 西安市科技局计划项目(2020KJRC0109)

MOEA/NumD and MOEA/D-M2M.

Key words: Multi-objective optimization; evolutionary algorithm; decomposition method; polynomial mutation

1 引言

在控制工程^[1]、机器人^[2-3]、软件开发^[4]、工程设计^[5]、生产调度^[6]等诸多工程领域中,存在大量需要同时考虑对若干个目标进行优化的问题,而且这些目标之间是相互冲突的,这种问题被称为多目标优化问题(Multi-Objective Optimization Problem, MOP).

生物进化是一个不断优化的过程,在不断的变化过程中增加自身的适应性.进化算法^[7-8]通过模拟自然界的生物进化过程来对问题进行求解,鲁棒、并行和高效的特点使得该算法有了长足的发展和广泛的应用.近二十余年来,各种求解多目标优化问题的新算法及应用不断涌现^[9-13].MOEA/D-M2M^[11]算法将一个多目标优化问题均匀分解为若干个子问题,在求解多目标优化问题时比 MOEA/D^[10]算法具有更加优异的性能.

变异算子能够加强进化算法的局部搜索能力,在算法运行的后期可以加速向最优解收敛并保持解的多样性.因此,进化算法的收敛性、鲁棒性、全局搜索性能等受限于变异操作的性能.研究表明^[14-15]变异算子在进化算法中起着非常重要的作用.结合多项式变异策略与基于分解的多目标进化算法,本文提出采用多项式变异策略和分解方法的多目标进化算法(Multi-Objective Evolution Algorithm Using Polynomial Mutation Strategy and Decomposition Methods, MOEA/PmD),鉴于非均匀变异算子具有动态调整步长获得自适应性的特点,本文也尝试将非均匀变异算子和分解方法相结合,提出采用非均匀变异算子和分解方法的多目标进化算法(Multi-Objective Evolution Algorithm Using Non-uniform Mutation Operator and Decomposition Method, MOEA/NumD),然后在 ZDT 系列、DTLZ1-4 等测试问题上比较了 MOEA/PmD、MOEA/NumD、MOEA/D-M2M 等算法.实验表明,采用本文提出的 MOEA/PmD 算法拥有更好的性能.

2 采用多项式变异策略和分解方法的多目标进化算法

2.1 多项式变异算子

在进化算法中,变异算子是按照一定的变异概

率 pm 随机在个体上执行变异运算.本文提出的 MOEA/PmD 算法采用了多项式变异算子,如式(1)所示.

$$v'_k = v_k + \delta \cdot (u_k - l_k) \quad (1)$$

式中, $\delta =$

$$\begin{cases} [2u + (1-2u)(1-\delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & \text{if } u \leq 0.5 \\ 1 - [2(1-u) + 2(u-0.5)(1-\delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & \text{if } u \geq 0.5 \end{cases} \quad (2)$$

式(2)中, v_k 表示一个父个体, $\delta_1 = (v_k - l_k)/(u_k - l_k)$, $\delta_2 = (u_k - v_k)/(u_k - l_k)$, u 是一个 $[0, 1]$ 区间的随机数, η_m 是分布指数.

2.2 越界处理

在执行交叉和变异操作时,可能会出现新生成个体的某些维度上的数值超出取值范围的情形,此时需要及时调整到取值范围,本文采用的策略是用随机生成对应维度上取值范围内的值替代超出取值范围的值.个体 v 的第 k 维取值的调整方法,如式(3.3)所示.

$$v'_k = \begin{cases} v'_k & \text{if } l_k \leq v'_k \leq u_k \\ l_k + \text{rand} * (u_k - l_k) & \text{otherwise} \end{cases} \quad (3)$$

2.3 采用多项式变异策略和分解方法的的多目标进化算法

基于分解方法的多目标进化算法不但能用于求解多目标优化问题,也适用于求解超多目标优化问题. MOEA/D-M2M 算法将一个多目标优化问题转换为若干个子空间上的若干个凸优化问题,避免了采用基于支配的选择算子的多目标进化算法丢失非凸搜索区域并减慢搜索速度的问题.多项式变异算子能够增强局部搜索的能力,在后期既能加速收敛,又能维持多样性.基于此,本文提出了基于多项式变异策略和分解方法的新的多目标进化算法(MOEA/PmD),如算法 1 所示.

算法 1: 采用多项式变异策略和分解方法的多目标进化算法 MOEA/PmD

Step1: 预处理. 生成均匀的方向向量, 给定个体分组数 $classnum$, 确定每个分组的个体数目.

Step2: 初始化. 给定最大迭代次数 G_{max} , 当前迭代次数 $G = 0$. 多项式变异中, 分布指数 η_m 取值为 1.

Step3:产生初始种群,计算种群中每个个体的适应度值并对种群进行分组。

Step4:判断算法是否满足结束条件,如果是,则结束迭代并输出结果,否则,转步骤 5。

Step5:交叉操作. 对于种群中的每个个体 P_1 ,按照一定的概率执行交叉操作,生成新的个体 chdC. 本算法采用 MOEA/D-M2M 算法的交叉算子. 如果新生成的个体 chdC 取值有超出范围的,则按照式(3)进行越界处理。

Step6:变异操作. 对交叉运算得到的新个体 chdC 按照式(1)进行多项式变异操作,生成新的个体 chdM. 如果新生成的个体 chdM 取值有超出范围的,则按照式(3)式进行越界处理。

Step7:用所有新生成的个体构建中间种群,计算中间种群中每个新生个体的适应度值。

Step8:合并原种群和中间种群,根据步骤 1 确定的每个分组的中心和每个分组的个体数目,从合并种群中选择个体进行分组,产生新一代个体种群 pop. 转步骤 4。

3 实验结果与分析

3.1 测试函数

为了评估所提算法的性能,将其结果与由 MOEA/PmD、MOEA/NumD 和 MOEA/D-M2M 获得的结果进行比较. 采用 9 个测试函数,包含了 ZDT1—ZDT4、ZDT6,以及 DTLZ1—DTLZ4,采用的决策变量个数为 10,采用逆世代距离(Inverted Generational Distance, IGD)和超体积指标(Hypervolume, HV)作为性能评价指标,测试了三种算法的性能。

3.2 参数设定

对于每个测试函数,每个算法独立运行 10 次算法,实验参数设定如下:

均采用实数编码,运行 ZDT 系列测试函数时种群规模为 100,运行 DTLZ1—DTLZ4 测试函数时种群规模为 300. 分组(子种群)个数为 10. 多项式变异算子中分布指数 $\eta_m=11$. 运行 ZDT 系列测试函数时适应度函数值的计算次数为 600,运行 DTLZ 系列测试函数时适应度函数值的计算次数为 1 000。

3.3 实验分析

图 1 给出了分别用 MOEA/PmD 算法、MOEA/NumD 算法和 MOEA/D-M2M 算法运行 10 次 ZDT 系列测试函数后得到的最大超体积的 Pareto 前沿,可以看到在 ZDT1、ZDT3 和 ZDT4 等

测试函数上,本文提出的 MOEAD/PmD 算法获得了分布更广的 Pareto 前沿。

表 1 给出了三种算法在 ZDT 系列测试函数上运行后得到的逆世代距离 IGD,可以看出 MOEA/PmD 在 ZDT 系列测试函数上都获得了更优的最大、最小和平均值,表明上 MOEA/PmD 算法的收敛性更好,而在 ZDT1、ZDT2、ZDT4 和 ZDT6 上得到了最小的标准差,表明 MOEA/PmD 算法具有更好的鲁棒性。

表 2 给出了三种算法在 ZDT 系列测试函数上运行后得到的超体积,可以看出 MOEAD/PmD 在 ZDT1、ZDT3 和 ZDT4 上具有明显的优势,在 ZDT2 上三种算法都得到的超体积最大值是相等的,而 MOEA/PmD 得到的最小值和平均值比 MOEA/M2M 略小,在 ZDT6 上 MOEA/PmD 和 MOEA/D-M2M 一起得到了相等的最小值,而且 MOEA/PmD 得到的 Pareto 解集具有更好的标准差,只是 MOEA/PmD 得到的最大值和平均值上比 MOEA/D-M2M 略小。

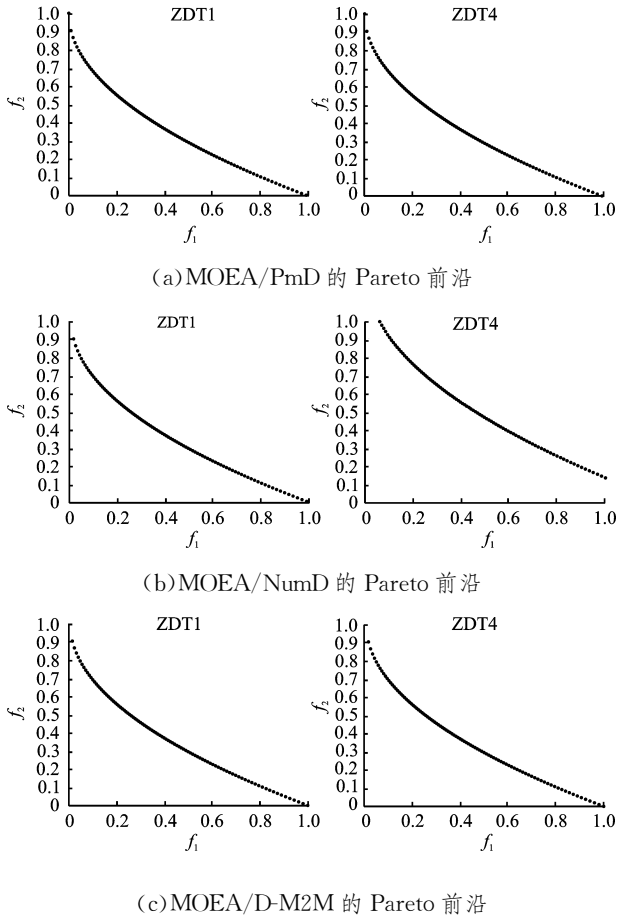


图 1 MOEAD/PmD、MOEA/NumD 和 MOEA/D-M2M 在测试问题 ZDT1 和 ZDT4 得到的 Pareto 前沿

表 1 MOEA/PmD、MOEA/NumD 和 MOEA/D-M2M 在 ZDT 系列测试函数上的逆世代距离 IGD

测试函数	算法	最大值	最小值	平均值	标准差
ZDT1	MOEA/ PmD	0.004 9	0.004 9	0.004 90	8.67E-19
	MOEA/NumD	0.009 1	0.006 0	0.006 49	8.73E-04
	MOEA/D-M2M	0.006 3	0.006 1	0.006 21	5.39E-05
ZDT2	MOEA/ PmD	0.003 8	0.003 8	0.003 80	0.00E+00
	MOEA/NumD	0.004 0	0.003 9	0.003 91	3.00E-05
	MOEA/D-M2M	0.003 9	0.003 8	0.003 83	4.58E-05
ZDT3	MOEA/ PmD	0.534 0	0.361 5	0.470 57	6.31E-02
	MOEA/NumD	0.595 6	0.588 4	0.593 16	2.39E-03
	MOEA/D-M2M	0.595 7	0.505 3	0.569 56	3.23E-02
ZDT4	MOEA/ PmD	0.004 9	0.004 9	0.004 9	8.70E-19
	MOEA/NumD	0.523 5	0.121 7	0.477 15	1.19E-01
	MOEA/D-M2M	0.006 3	0.006 2	0.006 22	4.00E-05
ZDT6	MOEA/ PmD	0.003 3	0.003 2	0.003 23	4.58E-05
	MOEA/NumD	0.042 0	0.003 2	0.008 88	1.17E-02
	MOEA/D-M2M	0.003 3	0.003 2	0.003 23	4.58E-05

表 2 MOEA/PmD、MOEA/NumD 和 MOEA/D-M2M 在 ZDT 系列测试函数上的超体积 HV

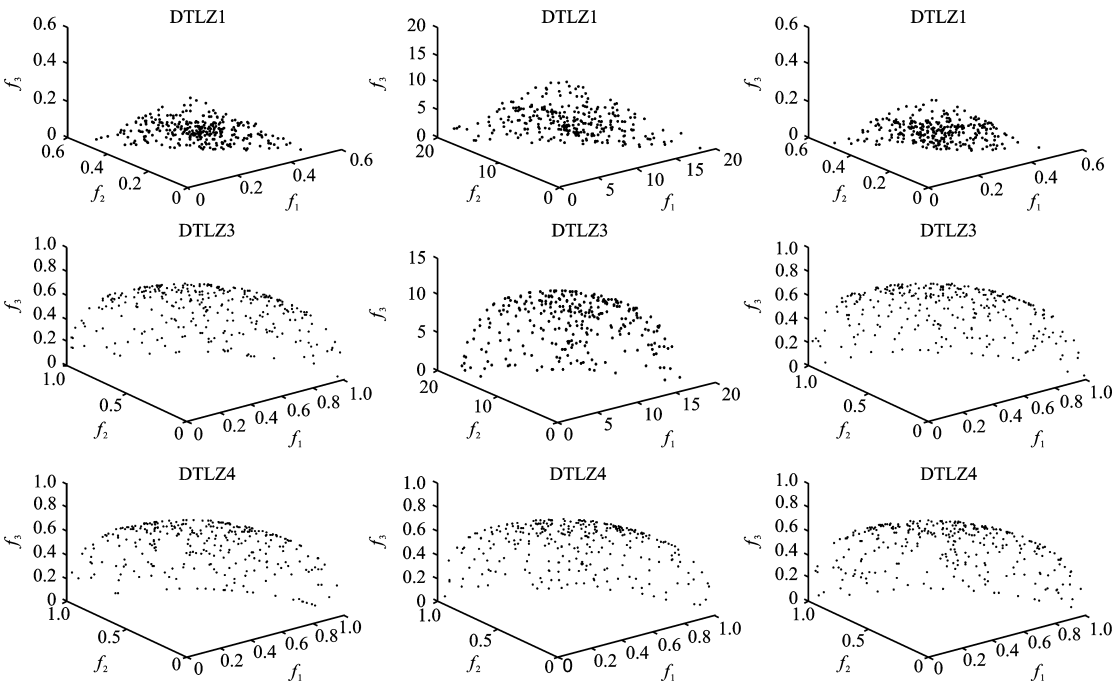
测试函数	算法	最大值	最小值	平均值	标准差
ZDT1	MOEA/ PmD	0.661 5	0.661 4	0.661 48	4.00E-05
	MOEA/NumD	0.661 5	0.653 1	0.660 20	2.44E-03
	MOEA/D-M2M	0.661 5	0.661 4	0.661 47	4.58E-05
ZDT2	MOEA/ PmD	0.328 4	0.328 1	0.328 29	8.31E-05
	MOEA/NumD	0.328 4	0.327 8	0.328 25	1.80E-04
	MOEA/D-M2M	0.328 4	0.328 3	0.328 39	3.00E-05
ZDT3	MOEA/ PmD	1.627	1.613 2	1.623 02	4.80E -03
	MOEA/NumD	1.610 5	1.606 1	1.609 00	1.68E-03
	MOEA/D-M2M	1.610 9	1.610 6	1.610 69	1.22E-04
ZDT4	MOEA/ PmD	0.661 6	0.661 4	0.661 5	4.47E-05
	MOEA/NumD	0.493 8	0.102 4	0.145 3	1.16E-01
	MOEA/D-M2M	0.661 5	0.661 4	0.661 49	3.00E-05
ZDT6	MOEA/ PmD	0.282 2	0.281 5	0.281 95	2.06E-04
	MOEA/NumD	0.282 2	0.228 6	0.273 41	1.60E-02
	MOEA/D-M2M	0.282 3	0.281 5	0.282 1	2.32E-04

图 2 给出了 MOEA/PmD 算法、MOEA/NumD 算法和 MOEA/D-M2M 算法分别在测试函数 DTLZ1—DTLZ4 上运行 10 次后得到的具有最大超体积的解集的 Pareto 界面,可以看到 DTLZ1、DTLZ3 和 DTLZ4 上,本文提出的 MOEAD/PmD 算法获得了更好地的 Pareto 界面,而 MOEA/NumD 算法在 DTLZ1 和 DTLZ3 上获得的 Pareto 界面显然没有收敛到理想的范围。

表 3 给出了三种算法在测试函数 DTLZ1—DTLZ4 上运行 10 次后得到的逆世代距离 IGD 的最大值、最小值、平均值和标准差,可以看出 MOEA/PmD 在 DTLZ1、DTLZ3 和 DTLZ4 得到的

IGD 小于 MOEAD/NumD 和 MOEA/D- M2M,而在 DTLZ2 上 MOEA/D- M2M 得到 IGD 更小。

表 4 给出了三种算法在测试函数 DTLZ1—DTLZ4 上运行 10 次后得到的超体积 HV 的最大值、最小值、平均值和标准差,可以看出 MOEAD/PmD 在 DTLZ1—DTLZ4 上都具有明显的优势,仅仅在 DTLZ3 上 MOEAD/PmD 得到比 MOEA/D-M2M 稍小的超体积最大值,但是其在 DTLZ3 得到的体积最小值和平均值比 MOEA/D-M2M 更大。与此同时,MOEAD/PmD 在 DTLZ1、DTLZ3 和 DTLZ4 上得到的超体积的标准差最小,也表明了该算法的鲁棒性。



(a)MOEA/PmD 的 Pareto 前沿 (b)MOEA/NumD 的 Pareto 前沿 (c)MOEA/D-M2M 的 Pareto 前沿

图 2 MOEA/PmD、MOEA/D-M2M 和 MOEA/NumD 在测试问题 DTLZ1-DTLZ4 上得到的 Pareto 前沿

表 3 MOEA/PmD、MOEA/NumD 和 MOEA/D-M2M 在测试函数 DTLZ1-4 上的逆世代距离 IGD

测试函数	算法	最大值	最小值	平均值	标准差
DTLZ 1	MOEA/PmD	0.028 0	0.018 6	0.022 41	3.08E-03
	MOEA/NumD	0.038 0	0.018 5	0.028 08	4.76E-03
	MOEA/D-M2M	0.037 2	0.018 3	0.022 69	5.18E-03
DTLZ 2	MOEA/PmD	0.047 8	0.038 5	0.041 68	2.64E-03
	MOEA/NumD	0.043 4	0.039 5	0.041 29	1.18E-03
	MOEA/D-M2M	0.042 6	0.039 5	0.040 74	9.98E-04
DTLZ 3	MOEA/PmD	0.058 3	0.042 5	0.049 91	5.44E-03
	MOEA/NumD	0.112 3	0.045 9	0.076 43	2.13E-02
	MOEA/D-M2M	0.164 5	0.040 5	0.067 25	4.05E-02
DTLZ 4	MOEA/PmD	0.071 8	0.036 3	0.055 25	1.02E-02
	MOEA/NumD	0.099 1	0.038 2	0.063 85	2.06E-02
	MOEA/D-M2M	0.077 2	0.043 4	0.055 61	1.14E-02

表 4 MOEA/PmD、MOEA/NumD 和 MOEA/D-M2M 在测试函数 DTLZ1-4 上的超体积 HV

测试函数	算法	最大值	最小值	平均值	标准差
DTLZ 1	MOEA/PmD	0.094 4	0.089 7	0.092 99	1.39E-03
	MOEA/NumD	0.093 9	0.086 4	0.090 00	1.82E-03
	MOEA/D-M2M	0.094 4	0.087 9	0.093 12	1.94E-03
DTLZ 2	MOEA/PmD	0.432 5	0.427 6	0.429 83	1.30E-03
	MOEA/NumD	0.431 4	0.423 9	0.429 10	1.99E-03
	MOEA/D-M2M	0.432 0	0.427 6	0.429 78	1.25E-03
DTLZ 3	MOEA/PmD	0.423 0	0.376 4	0.399 18	1.62E-02
	MOEA/NumD	0.407 4	0.288 9	0.347 18	3.86E-02
	MOEA/D-M2M	0.425 1	0.270 0	0.383 59	5.37E-02
DTLZ 4	MOEA/PmD	0.412 8	0.408 3	0.410 55	1.21E-03
	MOEA/NumD	0.407 9	0.400 6	0.404 48	2.44E-03
	MOEA/D-M2M	0.411 5	0.406 9	0.409 83	1.32E-03

4 结束语

本文将多项式变异算子和 MOEA/D-M2M 算法提出的新的分解方法相结合,提出一种新的采用多项式变异策略和分解方法的多目标进化算法.从实验结果可以看出,本文提出的算法优于 MOEA/D-M2M.本文也尝试提出采用非均匀变异算子和分解方法的多目标进化算法(MOEA/NumD 算法),通过实验发现,MOEA/NumD 在某些测试问题上能得到好的实验结果,但是总体效果不如 MOEA/PmD.下一步计划是将本文提出的算法拓展到求解约束多目标优化问题中.

参考文献:

- [1] QIAN S Q, YE Y Q, LIU Y M, et al. An improved binary differential evolution algorithm for optimizing PWM control laws of power inverters[J]. Optimization and Engineering, 2018, 19(2): 271-296. DOI: 10.1007/s11081-017-9354-5.
- [2] MORADI B. Multi-objective mobile robot path planning problem through learnable evolution model[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2019, 31(2): 325-348. DOI: 10.1080/0952813X.2018.1549107.
- [3] RAHUL K. Robot mission planning using Co-evolutionary optimization[J]. Robotica, 2020, 38(3): 512-530. DOI: 10.1017/S026357471900081X.
- [4] 樊田田, 许蕾, 陈林. 基于多目标优化算法 NSGA-II 推荐相似缺陷报告[J]. 计算机学报, 2018, 42(10): 2175-2189. DOI: 10.11897/SP.J.1016.2019.02175. FAN T T, XU L, CHEN L. Recommending similar bug reports based on multi-targets optimization algorithm NSGA-II[J]. Chinese Journal of Computers, 2018, 42(10): 2175-2189. DOI: 10.11897/SP.J.1016.2019.02175.
- [5] JANGIR P, TRIVEDI N. Non-dominated sorting moth flame optimizer: a novel multi-objective optimization algorithm for solving engineering design problems[J]. Engineering Technology Open Access Journal, 2018, 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579.
- [6] LIN L, GEN M. Hybrid evolutionary optimisation with learning for production scheduling: state-of-the-art survey on algorithms and applications[J]. International Journal of Production Research, 2018, 56(1-2): 193-223. DOI: 10.1080/00207543.2018.1437288.
- [7] HOLLAND J H. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan Press, 1975.
- [8] TRIPATHY A. Numerical optimization of computer models[J]. Journal of the Operational Research Society, 1982, 33(12): 1166. DOI: 10.1057/jors.1982.238.
- [9] 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. DOI: 10.1109/4235.996017.
- [10] ZHANG Q F, LI H. MOEA/D: a multi-objective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731. DOI: 10.1109/TEVC.2007.892759.
- [11] LIU H L, GU F Q, ZHANG Q F. Decomposition of a multiobjective optimization problem into a number of simple multiobjectivesubproblems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 450-455. DOI: 10.1109/TEVC.2013.2281533.
- [12] WANG Y P, LIU H Y, WEI F, et al. Cooperative co-evolution with formula-based variable grouping for large-scale global optimization[J]. Evolutionary Computation, 2018, 26(4): 569-596. DOI:10.1162/evco_a_00214.
- [13] LI H C, WANG Z C. An evolutionary algorithm using parameter space searching for interval linear fractional bilevel programming problems[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2016, 30(4): 1659011. DOI: 10.1142/S0218001416590114.
- [14] WANG H B, WANG J, ZHEN XX, et al. Oriented multi-mutation strategy in a many-objective evolutionary algorithm[J]. Information Sciences, 2019, 478: 391-407. DOI:10.1016/j.ins.2018.11.042.
- [15] ZHANG Q, ZOU D X, DUAN N, et al. An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem[J]. Applied Soft Computing, 2019, 78: 641-669. DOI: 10.1016/j.asoc.2019.03.019.

作者简介:

王之仓 男,(1974-),博士,教授.研究方向为进化计算、信息安全. E_mail:zhcwang@xupt.edu.cn.

李和成 男,(1973-),博士,教授,博士生导师.研究方向为计算智能、最优化理论与方法.