

具有学习机制的离散差分演化算法

周雅兰¹ 朱耀辉² 张 军¹

(广东商学院信息学院 广州 510320)¹ (中山大学信息科学与技术学院 广州 510006)²

摘 要 如何将差分演化算法应用于离散领域是目前该领域的一个热点研究问题。用分布估计算法对搜索空间中优质解的分布进行建模,然后根据建立的模型抽样产生新解,利用分布估计这种全局信息学习的机制,提出具有学习机制的离散差分演化算法并用于求解多维背包问题。实验结果表明,提出的算法具有良好的性能。

关键词 离散差分演化算法,学习机制,分布估计算法,多维背包问题

中图分类号 TP18

文献标识码 A

Discrete Differential Evolution with Learning Mechanism

ZHOU Ya-lan¹ ZHU Yao-hui² ZHANG Jun¹

(School of Information, Guangdong University of Business Studies, Guangzhou 510320, China)¹

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China)²

Abstract How to apply differential evolution to discrete field is currently a hot research problem in this area. Estimation of distribution algorithms build a probability model which characterizes the distribution of the current promising solutions in the search space and generates new solutions according to the model. Using the global learning of estimation of distribution algorithms, a discrete differential evolution with learning mechanism was proposed for multidimensional knack problem. Simulation results show that the proposed algorithm has good performance.

Keywords Discrete differential evolution, Learning mechanism, Estimation of distribution algorithm, Multidimensional knack problem

1 前言

差分演化(Differential Evolution, DE)最初提出时是一种实数编码的进化算法^[1,2],用于在连续空间求解全局优化问题,近年来在这方面取得了较好的效果,见文献[3-5]。但是在离散空间领域差分演化算法还没有较好的研究成果,因此,将差分演化算法应用于离散领域是该领域目前的一个热点研究方向。

分布估计算法(Estimation of Distribution Algorithm, EDA)^[6-8]是最近提出的进化算法,具有全局学习能力。不同于传统的进化算法,EDA既没有交叉操作也没有变异操作,取而代之的是建模和抽样操作,即首先选择种群中较优的个体,然后对这些较优个体在解空间中的分布建立概率模型(即进行全局学习),最后从这个概率模型中抽样产生新解。

本文将分布估计算法的全局学习思想引入到离散差分演化算法中,形成具有学习机制的离散差分演化算法,在多维背包问题(Multidimensional Knack Problem, MKP)上的求解结果表明具有学习机制的离散差分演化算法具有良好的性能。

2 离散差分演化算法

2.1 标准差分演化算法

DE是一种有效的演化算法^[1,2],它主要通过变异、交叉和选择这3种操作的反复迭代来完成寻优过程。

设在 D 维空间,DE种群的向量个数为 N ,第 t 代的向量个体 i 表示为 $X_i(t) = (x_{i1}(t), \dots, x_{id}(t), \dots, x_{iD}(t))$, $i=1, 2, \dots, N$ 。

首先进行变异操作,产生变异向量。随机初始化种群后,给每一个目标向量 $X_i(t)$ 产生一个变异向量 $M_i(t+1) = (m_{i1}(t+1), \dots, m_{id}(t+1), \dots, m_{iD}(t+1))$,变异向量是DE的关键组件,它是通过把两个不同向量的差异(也称为差分向量)加权后与第3个向量求和而形成的,用式(1)表示:

$$M_i(t+1) = X_{i1}(t) + F(X_{i2}(t) - X_{i3}(t)) \quad (1)$$

式中, $i1, i2, i3 \in \{1, 2, \dots, N\}$ 是三个不相等的整数,参数 F 为缩放因子,用于控制差分向量的放大程度,参数 F 的正确取值有利于维持算法局部开采和全局探索能力之间的平衡。

接着进行交叉操作,产生中间向量。通过目标向量与变异向量进行交叉来产生中间向量 $Trial_i(t+1) = (trial_{i1}(t+1)$

到稿日期:2010-08-25 返修日期:2010-12-11 本文受国家自然科学基金(60905038, 60873198),广东省自然科学基金项目(1015103200 1000001),广东高校优秀青年创新人才培养计划(育苗工程)项目(LYM09083),广东商学院科研创新团队建设计划,广东商学院校级科研项目(08BS52001)资助。

周雅兰(1979—),女,博士,讲师,CCF会员,主要研究方向为人工智能与数据挖掘,E-mail:zhouyulan@gdcc.edu.cn;朱耀辉(1984—),男,硕士,主要研究方向为人工智能;张军(1966—),男,博士,教授,主要研究方向为信息安全。

1), ..., trial_{id}(t+1), ..., trial_{id}(t+1)), 而常用的 DE 交叉方法用二项式交叉表示为:

$$trial_{id}(t+1) = \begin{cases} m_{id}, & \text{if rand}() < CR \text{ or } d=r \\ x_{id}, & \text{otherwise} \end{cases} \quad (2)$$

式中, CR 是交叉率, 决定中间向量来自于变异向量的概率; $r \in \{1, 2, \dots, D\}$ 。

最后进行选择操作, 产生下一代向量。标准 DE 采用贪婪选择的思想, 从中间向量 $Trial_i(t+1)$ 和目标向量 $X_i(t)$ 之间选择一个较好的作为下一代向量 $X_i(t+1)$ 。

2.2 离散差分演化算法

目前已经有一些离散差分演化算法, 但是它们大部分是将离散空间问题映射或转换到连续空间求解, 见文献[9-14]。2007 年, Gong 和 Tuson 提出了一种二进制编码差分演化算法, 简称为 BDE(Binary DE)^[15], 这种方法在离散空间将求解的问题编码, 然后利用 DE 算法直接求解。BDE 中有两种二进制编码方法, 分别是“所有变量是一维”(All Decision Variables as a Single Dimension)和“每个变量是一维”(Each Decision Variable as a Single Dimension)。两种编码的区别是产生变异向量的方法不同, 而交叉、选择操作都与标准 DE 的相同, 下面描述这两种编码产生变异向量的过程^[15]:

在“所有变量是一维”这种编码方法中, 变异策略依据式(1), 首先随机选择 3 个个体 X_{i1}, X_{i2}, X_{i3} , 然后用异或操作计算个体 X_{i2} 和 X_{i3} 的海明距离 dis , 接着计算度量距离(scaled distance) $d' = F \times dis$ 。例如, 如果 $dis = 6, F = 0.3$, 则 $d' = 1.8$, 表示有 80% 的概率选择 X_{i1} 中的任意两个维翻转产生变异向量, 有 20% 的概率选择 X_{i1} 中的任意一个维翻转产生变异向量。这种确定 X_{i1} 中变异维个数的方法表示如下:

$$D_m = \begin{cases} \text{int}(d') + 1, & \text{if } \text{rand}() < d' - \text{int}(d') \\ \text{int}(d'), & \text{otherwise} \end{cases} \quad (3)$$

式中, $\text{int}()$ 是向下取整函数, $\text{rand}()$ 是在 $[0, 1]$ 区间生成的一个随机数, D_m 是求得的 X_{i1} 中可任意翻转维的个数。

在“每个变量是一维”这种编码方式中, 变异策略同样也是依据式(1), 首先随机选定 3 个个体 X_{i1}, X_{i2}, X_{i3} , 对 X_{i2} 和 X_{i3} 的各维分别进行异或操作, 如果某维的异或操作结果为 1 且 $\text{rand}() < F$, 则 X_{i1} 对应的该维翻转。通过这种翻转 X_{i1} 特定维的方式产生变异向量。

从上面的描述可见, “所有变量是一维”和“每个变量是一维”这两种编码方法在变异时翻转维的限制条件不同, 前者可以改变个体任意维产生变异向量, 而后者则只能改变特定的维来产生变异向量。

3 带学习机制的离散差分演化算法

3.1 分布估计算法(EDA)

分布估计算法主要有 3 步操作^[6-8]: 选择、建模和抽样。设 $\text{Pop}(t)$ 是第 t 代的种群, EDA 按照如下迭代方式工作:

第一步 选择。从 $\text{Pop}(t)$ 中选择 M 个个体形成父种群 $Q(t)$;

第二步 建模。向 $Q(t)$ 中的个体学习, 抽取统计信息建立概率模型 P ;

第三步 抽样。从构建的概率模型 P 中抽样产生新解。

EDA 的概率模型可以分为变量无关、双变量相关和多变

量相关 3 种。变量无关模型是最简单的一种^[7], 但被广泛应用, 其比较有代表性的是用于单变量边缘分布算法(Univariate Marginal Distribution Algorithm, UMDA)、基于群体的增量学习算法、紧凑遗传算法和带 guided mutation 的进化算法中。本文也采用变量无关模型来建模求解 0-1 组合优化问题。

3.2 BDE-EDA

受文献[16]启发, 我们将 EDA 的全局学习思想^[6-8]引入到 Gong 和 Tuson 提出的 BDE 中^[15], 形成具有学习机制的离散差分演化算法(BDE-EDA), 算法的过程如下:

- 第一步 初始化, 随机生成 N 个个体, 初始化概率向量 P , 设 $i = 0$, 迭代次数 $generation = 0$;
- 第二步 $i = i + 1$, 从 N 个个体中随机选出 3 个不同的个体;
- 第三步 按照 BDE 中的方法执行变异操作;
- 第四步 按照式(2)执行交叉操作;
- 第五步 执行 DE 的选择操作;
- 第六步 如果 $i < N$ 则继续第二步, 否则转向第八步;
- 第七步 如果 $++generation \geq startingGen$ 且 $generation \% interval = 0$, 执行第八步; 否则转向第十二步;
- 第八步 计算种群的平均适应度值;
- 第九步 选出适应度值在平均适应度值之上的个体作为优势种群, 根据式(4)更新概率向量 P ;
- 第十步 从概率向量 P 中抽样产生 M 个新个体;
- 第十一步 从 $N + M$ 个个体中选出 N 个个体放入下一代种群;
- 第十二步 如果没有达到最大迭代次数则转向第二步; 否则算法终止, 输出最优解。

上述算法中, 第一步的初始化概率向量 P , 用来统计个体每一维出现 1 的个体数量, 根据概率模型 $p_d = \sum_{i=1}^N x_{id} / N$ 计算出一个初始概率向量 $P = (p_1, p_2, \dots, p_d, \dots, p_D)$, 其中 p_i 为第 i 维取值为 1 的概率, $1 - p_i$ 表示第 i 维取值为 0 的概率。

算法的第三步可以根据具体求解问题选择“所有变量是一维”和“每个变量是一维”两种编码方法中的一种来执行变异操作, 本文的仿真实验中选择“所有变量是一维”这种编码方法来求解多维背包问题。

第七步到第十步是学习策略的具体实现, 其中第七步是对使用学习策略的时机进行判断, 包括开始使用学习策略的时间 $startingGen$ 和使用学习策略的间隔代数 $interval$ ^[16]。在算法初始阶段, 种群质量通常不高, 还不适合学习, 因此提出的算法在经过 $startingGen$ 次迭代后才开始学习。为了降低算法的运行时间, 设置了 $interval$ 参数, $interval$ 的值不能设置得太大, 否则可能会因为种群多样性下降得太快而使学习策略失效。第九步是为了保证算法的质量, 只向适应度值在种群平均适应度值之上的个体学习。在每一次迭代中, 都按照 UMDA^[6,7] 的概率更新规则更新概率向量 P , 选择适应度值最高的 M 个个体作为优势种群 D_i^s , 由优势种群构建概率模型, 估计联合概率分布, 用方程表示如下:

$$p_d(t) = p(x_{id}(t) | D_i^s) = \prod_{d=1}^D p_d(t) \\ = \prod_{d=1}^D \frac{\sum_{X_i(t) \in D_i^s} x_{id}(t)}{M} \quad (4)$$

第十一步可以看成是一种选择操作, 从 N 个个体中选出最差的 M 个个体用新产生的 M 个个体替代, 通常 $M < N$ 。

从上述算法描述的过程可以看出学习机制是统计种群中一些优秀的个体,然后向这些优秀的个体学习,这种考虑全局的学习机制能让算法收敛到更有可能达到全局最优状态的区域内,从而使算法更加有效。

4 仿真实验

为了显示引入学习机制后 BDE 算法的良好性能,将 BDE-EDA 与 BDE 算法求解 MKP 的实验结果进行了比较,程序用 C 语言实现,在笔记本电脑 (Core2 T6600 2.2GHz

2.19GHz, RAM 2G) 上运行。为了进一步显示 BDE-EDA 算法的相对性能,还与求解 MDP 的 TCNN (transiently chaotic neural network) [17], 以及 Zhou 等人 [18] 最近在 TCNN 中引入一个简单的启发式策略 (heuristic strategy, HS) 后形成的 TCNN-HS 的实验结果进行了比较。本文的实验数据来自于 OR-LIBRARY (<http://people.brunel.ac.uk/~mastijb/jeb/orlib/files/>) 中的 16 组测试实例,在表 1 的第 1 列和第 2 列分别列出了这 16 组测试实例的名称和已知最优值。下面首先描述用于仿真实验的多维背包问题,然后是实验结果的比较。

表 1 BDE-EDA, BDE, TCNN 和 TCNN-HS 算法的实验结果比较

问题实例	已知最优解	BDE-EDA		BDE		TCNN		TCNN-HS	
		最好解	平均值	最好解	平均解	最好解	平均解	最好解	平均解
WEISH03	4115	4115	4115.00	4115	4084.63	4115	4018.1	4115	4108.3
WEISH06	5557	5557	5557.00	5542	5496.23	5557	5473.5	5557	5545.0
WEISH07	5567	5567	5567.00	5550	5525.00	5567	5460.9	5567	5567.0
WEISH08	5605	5605	5605.00	5574	5533.77	5605	5505.9	5605	5604.8
WEISH11	5643	5643	5643.00	5643	5629.07	5643	5291.7	5643	5642.2
WEISH13	6159	6159	6159.00	6084	6065.90	6159	5962.8	6159	6132.6
WEISH14	6954	6954	6954.00	6897	6848.27	6954	6817.6	6954	6954.0
WEISH16	7289	7289	7286.33	7273	7214.67	7289	7200.7	7289	7281.3
WEISH18	9580	9580	9563.77	9443	9409.03	9540	9413.7	9580	9535.5
WEISH19	7698	7698	7698.00	7645	7590.30	7683	7495.8	7698	7698.0
WEISH20	9450	9450	9450.00	9410	9402.17	9445	9311.8	9450	9450.0
WEISH22	8947	8947	8945.10	8868	8723.30	8947	8798.8	8947	8931.7
WEISH23	8344	8344	8344.00	8185	8076.27	8280	8122.8	8344	8341.1
WEISH26	9584	9584	9584.00	9327	9273.77	9584	9449.1	9584	9579.7
WEISH28	9492	9492	9492.00	9214	9117.90	9438	9303.9	9492	9492.0
WEISH29	9410	9410	9410.00	9203	9093.73	9410	9172.0	9410	9373.5

4.1 多维背包问题 (MKP)

MKP 就是从 n 个物体中选出一组物体装入到 m 个背包中,使得所装的物品总价值最大 [19]。设正实数 p_j, ω_j, c_i 分别表示物体 j 的价值、体积和背包 i 的容量,最大化 $f(x) = \sum_{j=1}^n p_j x_j$, 约束条件为 $\sum_{j=1}^n \omega_j x_j \leq c_i$, 其中 $i=1, 2, \dots, m, x_j \in \{0, 1\}, j=1, 2, \dots, n, x_i=1$ 表示物品 i 被装入背包, $x_i=0$ 表示物品 i 不被装入背包。一个定义完整的 MKP 还应满足 $\omega_{ij} < c_i$ 且 $\sum_{j=1}^n \omega_{ij} > c_i$ 。

MKP 是一个典型的组合优化问题,也被证明是一个 NP 难问题,在实际中有着广泛的应用。基于传统方法求解 MKP 的不足,近年来,许多研究者将进化算法如模拟退火、禁忌搜索、遗传算法和蚁群算法等用于求解 MKP, 见文献 [20, 21]。

为了满足约束条件加快算法求解过程,通常在元启发式算法中引入了修补因子把不满足 MKP 限制条件的个体修补成合法的个体,本文在 BDE-EDA 与 BDE 算法的变异、交叉操作之后,选择操作之前采用 Li 等人提出的基于贪婪方法的修补操作 (check-and-repair operator), 其过程描述如下 [22]:

- 第一步 计算每一个物品在每一个背包中的价值密度 $\delta_{ij} = p_i \cdot c_j / \omega_{ij}$;
 - 第二步 计算每个物品的最低价值密度 $\delta_i = \min \{ p_i \cdot c_j / \omega_{ij} \}$;
 - 第三步 按照 δ_i 的值对所有物品排序;
 - 第四步 从物品集中删除 δ_i 值最小的物品,例如把该物品对应的基因位由 1 变成 0;
 - 第五步 重复第四步直到找到一个合法的解。
- 从上述过程可以看出,针对背包问题的性质,该操作采用

贪婪算法的思想来尽可能保留单位价值量大的物品。

4.2 实验结果

实验中 BDE 算法的种群规模为 100, 最大迭代次数为 50, 变异因子为 0.5, 交叉因子为 0.9。BDE-EDA 算法中的 startGen=10, interval=2, 其余参数与 BDE 算法的设置一样。表 1 是 BDE-EDA, BDE, TCNN 和 TCNN-HS 算法在 16 个测试用例运行 50 次时得到的最好解和平均解。表 2 是 BDE-EDA, BDE 和 TCNN-HS 3 种算法的平均运行时间。

表 2 3 种算法的平均运行时间 (秒)

问题实例	BDE-EDA	BDE	TCNN-HS
WEISH03	0.020	0.009	0.12
WEISH06	0.022	0.010	0.17
WEISH07	0.108	0.028	0.17
WEISH08	0.125	0.030	0.17
WEISH11	0.030	0.016	0.30
WEISH13	0.031	0.018	0.29
WEISH14	0.019	0.036	0.41
WEISH16	0.021	0.036	0.38
WEISH18	0.045	0.023	0.47
WEISH19	0.039	0.022	0.55
WEISH20	0.044	0.024	0.50
WEISH22	0.052	0.024	0.70
WEISH23	0.054	0.024	0.70
WEISH26	0.064	0.028	0.87
WEISH28	0.069	0.028	0.86
WEISH29	0.067	0.027	0.87

从表 1 可以看出, BDE-EDA 算法和 TCNN-HS 算法在全部 16 组问题实例上均能求得已知最优解, TCNN 能在 11 组问题实例上求得已知最优解, 而 BDE 算法只能在 WEISH03 和 WEISH11 这 2 组问题实例上获得已知最优值。在平均解 (下转第 249 页)

- [5] 丛晓青. 一种动态描述逻辑及其应用[D]. 北京: 中国科学院研究生院, 2008
- [6] Description Logic Website[OL]. <http://dl.kr.org/>
- [7] Baader F, Nutt W. Basic Description Logics[M]. Cambridge University Press, 2003; 47-100
- [8] Wolter F, Zakharyashev M. Dynamic description logics [R]. CSLI Publications, 2000
- [9] 黄河. 语义 Web 中知识服务的研究[D]. 北京: 中国科学院研究生院, 2006
- [10] 董明楷. 面向智能主体的动态描述逻辑研究[D]. 北京: 中国科学院研究生院, 2003

- [11] Baader F, Sattler U. An Overview of Tableau Algorithms for Description Logics[J]. *Studia Logica*, 2001, 69; 5-40
- [12] Chau K W. An ontology-based knowledge management system for flow and water quality modeling[J]. *Advances in Engineering Software*, 2007, 38; 172-781
- [13] Alex B, Ronald J B. Conceptual modeling with description logics [M]. Cambridge University Press, 2003; 359-381
- [14] 吴修国, 曾广周, 许崇敬. 基于描述逻辑的目标研究[J]. *计算机科学*, 2008, 35(7); 142-144

(上接第 227 页)

方面, BDE-EDA 算法和 TCNN-HS 算法在 5 个问题实例上均能获得与已知最优解相同的平均解, 而在其他的 11 个问题实例上 BDE-EDA 算法能获得比 TCNN-HS 算法更好的平均解, BDE 和 TCNN 在所有问题实例上获得的平均解均比 BDE-EDA 的差。可以说 BDE-EDA 算法具有比 BDE 和 TCNN 算法更好的性能, 并且具有比 TCNN-HS 算法更稳定的性能。表 1 中各算法求得的与已知最优解相同的最好解和平均解用加粗字体标出。

从表 2 可以看出, BDE-EDA 算法的运行时间比 BDE 稍长, 这是因为 BDE-EDA 算法要花费时间向优质个体学习建立概率模型, 但是二者的时间差距不是很明显。TCNN-HS 算法的运行时间最长, 而且随着问题规模的增加算法的速度变得越来越慢。

综上所述, 从解的质量和运行时间来看, BDE-EDA 的运行虽然略慢于 BDE, 但是它的最优解和平均解的质量明显比 BDE 要好。而 TCNN-HS 算法虽然能取得与 BDE-EDA 相当的最优解, 但是在稳定性以及运行时间上比不上 BDE-EDA 算法。因此, 可以说 BDE-EDA 算法是一个更稳定、有更好寻优能力且运行效率较高的算法。

结束语 提出了具有学习机制的离散差分演化算法, 并用于求解多维背包问题, 实验结果表明提出的算法具有良好的性能。未来的研究工作是将该算法与其他机制相结合来进一步提高性能, 例如结合模拟退火等算法, 并将算法推广到实际应用中。

参 考 文 献

- [1] Price K. Differential Evolution V S. The Functions of the 2nd ICEO[C]// *Proceedings of IEEE International Conference on Evolutionary Computation*. Indianapolis, USA, 1997; 153-157
- [2] Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces [R]. International Computer Science Institute, Berkley, 1995
- [3] 陈荣元, 林立宇, 王四春, 等. 数据同化框架下基于差分进化的遥感图像融合[J]. *自动化学报*, 2010, 36(3); 392-398
- [4] 王凌, 黄付卓, 李灵坡. 基于混合双种群差分进化的电力系统经济负荷分配[J]. *控制与决策*, 2009, 24(8); 1156-1160
- [5] 李太勇, 唐常杰, 吴江, 等. 基于差分进化基因表达式编程的全局函数优化[J]. *计算机科学*, 2009, 36(11); 140-142
- [6] 周树德, 孙增圻. 分布估计算法综述[J]. *自动化学报*, 2007, 33(2); 113-124
- [7] Mühlenbein H. The equation for response to selection and its use for prediction[J]. *Evolutionary Computation*, 1997, 5(3); 303-346

- [8] Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning[R]. CMU-CS-94-163. School of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 1994
- [9] 胡蓉, 钱斌. 一种求解随机有限缓冲区流水线调度的混合差分进化算法[J]. *自动化学报*, 2009, 35(12); 1580-1586
- [10] 吴亮红, 王耀南, 陈正龙. 求解混合整数非线性规划问题的改进差分进化算法[J]. *小型微型计算机系统*, 2007, 28(4); 666-669
- [11] Qian B, Wang L, Huang D X, Wang X. Scheduling multi-objective job shops using a memetic algorithm based on differential evolution[J]. *International Journal of Advanced Manufacturing Technology*, 2008, 35(9/10); 1014-1027
- [12] Al-Anzi F S, Allahverdi A. A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times[J]. *European Journal of Operational Research*, 2007, 182; 80-94
- [13] Engelbrecht A P, Pampara G. Binary differential evolution strategies[C]// *IEEE Congress on evolutionary computation*, 2007; 1942-1947
- [14] Onwubolu G, Davendra D. Scheduling flow shops using differential evolution algorithm[J]. *European Journal of Operational Research*, 2006, 171; 674-692
- [15] Gong T, Tuson A L. Differential evolution for binary encoding [J]. *Soft Computing in Industrial Applications*, 2007, ASC 39; 251-262
- [16] Chang P C, Chen S H, Fan C Y, et al. Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems[J]. *Applied Mathematics and Computation*, 2008, 205; 550-561
- [17] Wang B Y, Dong H, He Z Y. A chaotic annealing neural network with gain sharpening and its application to the 0/1 knapsack problem[J]. *Neural Processing Letters*, 1999; 243-247
- [18] Zhou Y, Kuang Z H, Wang J H. A chaotic neural network combined heuristic strategy for multidimensional knapsack problem [C]// *ISICA 2008, LNCS 5370*. 2008; 715-722
- [19] Kong M, Tian P, Kao Y. A new ant colony optimization algorithm for the multidimensional knapsack problem[J]. *Computers and Operations Research*, 2008, 35(8); 2672-2683
- [20] 徐青鹤, 刘士荣, 吕强. 基于蚁群混沌行为的离散粒子群算法及其应用[J]. *计算机科学*, 2010, 37(5); 178-180
- [21] 曾智, 杨小帆, 陈静, 等. 求解多维 0-1 背包问题的一种改进的遗传算法[J]. *计算机科学*, 2006, 33(7); 220-223
- [22] Li H, Jiao Y C, Zhang L, et al. Genetic algorithm based on the orthogonal design for multidimensional knapsack problems[C]// *ICNC 2006. LNCS. Vol. 4221*, Springer, Heidelberg, 2006; 696-705