

差分进化帝王蝶优化算法求解 折扣{0-1}背包问题

冯艳红¹, 杨娟², 贺毅朝¹, 王改革³

(1. 河北地质大学信息工程学院, 河北石家庄 050031; 2. 凯理学院数学科学学院, 贵州凯里 556011;
3. 中国海洋大学信息科学与工程学院, 山东青岛 266100)

摘要: 帝王蝶优化算法(Monarch Butterfly Optimization, MBO)是一种新颖的群体智能算法, 自从提出就在实际优化问题上表现出很好的性能. 但是, 帝王蝶优化算法的迁移算子采用随机选择两个个体来生成新个体, 并没有记忆整个种群的最优解, 容易造成全局最优帝王蝶搜索经验的丢失. 根据 MBO 寻优过程的内在机制以及差分进化算法的变异算子能够利用个体间的差异信息, 将 MBO 分别与目前性能最优、应用范围最广的 7 种差分进化(Differential Evolution, DE)变异策略相结合, 实验验证了 7 种不同算法的性能. 基于性能最优的 DE/best/2/bin 变异模式, 提出了一种差分进化帝王蝶优化算法(Monarch Butterfly Optimization Algorithm with Differential Evolution, DEMBO), 使得算法能够记忆种群最优解并实现种群内部信息的充分共享, 达到既加快收敛速度又提高解的精度目的. 在 30 个典型折扣{0-1}背包问题(D{0-1}KP)实例上进行了一系列实验, 实验结果表明: (1) DEMBO 能够在时间复杂度不变的条件下, 显著提高算法的求解精度和收敛速度; (2) DEMBO 在求解所有 D{0-1}KP 实例时, 均能够获得一个近似比非常接近 1 的近似解.

关键词: 折扣{0-1}背包问题; 差分进化; 帝王蝶优化算法; 贪心修复策略; 近似比

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2018)06-1343-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.06.010

Monarch Butterfly Optimization Algorithm with Differential Evolution for the Discounted {0-1} Knapsack Problem

FENG Yan-hong¹, YANG Juan², HE Yi-chao¹, WANG Gai-ge³

(1. School of Information Engineering, Hebei GEO University, Shijiazhuang, Hebei 050031, China;

2. School of Mathematical Science, Kaili University, Kaili, Guizhou 556011, China;

3. College of Information Science and Engineering, Ocean University of China, Qingdao, Shandong 266100, China)

Abstract: Recently, inspired by the migratory behavior of monarch butterflies in nature, a swarm intelligence optimization algorithm, called monarch butterfly optimization algorithm (MBO), is proposed. Since MBO is proposed, it has good performances in various real-world optimization problems. However, migration operator of MBO selects randomly two individuals to generate new offspring, in which the useful search experience of global optimal individual is easily lost. Based on the intrinsic mechanism of the search process of MBO and the character of differential mutation operator, MBO is combined with 7 kinds of DE mutation strategies, respectively. Then a series of experiments are conducted to verify their performance. A DEMBO based on MBO and better differential evolution mutation strategy is presented, in which migration operator is replaced by differential mutation operator to enhance its global optimization ability. The over-all performance of DEMBO is verified and analyzed by 30 typical discounted {0-1} knapsack problem (D {0-1} KP) instances. The experimental results demonstrate that DEMBO can significantly improve the solution quality and convergence speed under the condition of not increasing the time complexity. Meanwhile, the approximation ratio of all the D {0-1} KP instances obtained by DEMBO is close to 1.0.

Key words: discounted {0-1} knapsack problem; differential evolution; monarch butterfly optimization algorithm; greedy repair strategy; approximate ratio

1 引言

帝王蝶优化 (Monarch Butterfly Optimization, MBO) 算法^[1]是王改革等人提出的一种较为新颖的群体智能算法,不同于基于生物群体的其他群体智能优化方法, MBO 模拟的是自然界中帝王蝶随着季节变化进行迁徙的行为. 已有的相关研究工作^[2,3]指出, MBO 在求解数值优化问题时,寻优性能与差分进化 (Differential Evolution, DE)^[4,5]等算法相比有一定的竞争力. 目前, MBO 及其改进形式已被应用到了实际优化问题求解中,如 0-1 背包问题^[1]、神经网络的训练^[2]等.

折扣 {0-1} 背包问题 (Discounted {0-1} Knapsack Problem, D{0-1} KP)^[6,7]最早由学者 Guldan B 首先提出,并给出了求解该问题的动态规划法;之后, Aiyong Rong^[8]等人对 D{0-1} KP 的核 (Core) 问题进行研究并结合动态规划法对问题求解;最近贺毅朝^[9]等人建立了 D{0-1} KP 的两个新的数学模型,并给出两种基于遗传算法对该问题求解的可行有效方法 (简记为 FirEGA 和 SecEGA);此外,贺毅朝^[11]等人对确切性算法和近似算法求解 D{0-1} KP 问题的性能进行比较,提出了一种二进制 PSO 算法求解该问题的新思路.

本文从 MBO 寻优过程的内在机制出发,注意到迁徙算子 (Migration Operator, MO) 并未体现出如 PSO 算法中的社会信息对于位置更新的贡献. 此外,子种群 1 中变异个体的生成仅仅考虑了父代种群中的一个个体,忽略了全局最优个体对种群搜索方向的引导作用,同时不利于种群多样性的保持. 大量实验结果表明,DE 算法的变异操作是 DE 算法中最关键的步骤,高度影响 DE 算法的性能. 目前性能最优且应用最广的 DE 变异策略包括 DE/best/ k , DE/current-to-rand/ k 和 DE/current-to-best/ k 等 ($k=1,2$). 因此,本文将 MBO 算法分别与 7 种典型差分变异策略结合,形成 7 种不同的混合 MBO 与差分变异的新算法,并通过实验验证了不同算法的性能,寻找到 MBO 与最优变异策略结合的新算法,即:差分进化帝王蝶优化算法 (Monarch Butterfly Optimization with Differential Evolution, DEMBO),新算法不仅记忆种群最优解并实现种群内部信息的充分共享,而且加快收敛速度、避免算法陷入局部最优. 为验证 DEMBO 算法的有效性,利用三类大规模 D{0-1} KP 实例^[6,8,9] (分别简记为 UDKP, WDKP 和 SDKP) 进行一系列实验. 实验结果表明,本文提出的 DEMBO 算法在求解 D{0-1} KP 问题上表现出良好的性能.

2 D{0-1} KP 的第一数学模型

D{0-1} KP 问题^[6,8]的提出源于商业领域中的“打折”思想. 该问题具有较高的理论和实际价值,在投资

决策、项目安排和预算控制等诸多领域都有实际应用.

D{0-1} KP 问题的简单形象描述是:给定 n 个项目 (物品) 组,任意一个项目组 i ($0 \leq i \leq n-1$) 均包含三个编号分别为 $3i, 3i+1$ 以及 $3i+2$ 的物品,其中物品 $3i$ 和物品 $3i+1$ 的价值系数分别为 p_{3i} 和 p_{3i+1} ,重量系数分别为 w_{3i} 和 w_{3i+1} ,物品 $3i+2$ 被认为是前两个物品结对组合而成,拥有折扣重量 $w_{3i+2} < w_{3i} + w_{3i+1}$,不变的总价值 $p_{3i+2} = p_{3i} + p_{3i+1}$. 同时 $w_{3i} < w_{3i+2}, w_{3i+1} < w_{3i+2}$,对于每一个项目组 i ($0 \leq i \leq n-1$),物品 $3i, 3i+1$ 以及 $3i+2$ 至多允许有一个物品被装入最大载重为 C 的背包. 问题是用 $3n$ 个物品的一部分去填充背包,在物品总重量不超过 C 的前提下使装入背包中的物品价值之和最大. 则 D{0-1} KP 的第一数学模型为:

$$\max f(X) = \max \sum_{i=0}^{n-1} (x_{3i}p_{3i} + x_{3i+1}p_{3i+1} + x_{3i+2}p_{3i+2}) \quad (1)$$

$$\text{s. t. } x_{3i} + x_{3i+1} + x_{3i+2} \leq 1, \quad \forall i \in [0, n-1] \quad (2)$$

$$\sum_{i=0}^{n-1} (x_{3i}w_{3i} + x_{3i+1}w_{3i+1} + x_{3i+2}w_{3i+2}) \leq C \quad (3)$$

$$x_{3i}, x_{3i+1}, x_{3i+2} \in \{0, 1\}, \quad \forall i \in [0, n-1] \quad (4)$$

其中,二元决策变量 x_j ($0 \leq j \leq 3n-1$) 表示物品 j 是否被装入背包. 如果 $x_j = 0$,说明物品 j 未被装入背包;否则,表示物品 j 被装入背包. 显然, D{0-1} KP 问题是包含 $n+1$ 个不等式约束、比经典 0-1 KP 更为复杂的组合优化问题.

3 帝王蝶优化算法的基本原理

在标准帝王蝶优化算法中,假设目标问题对应一个 d 维的解空间,随机初始化 N 个帝王蝶个体 $\mathbf{P} = (X_1, X_2, \dots, X_N)$ 分布在此解空间,第 i 个个体在解空间的位置可以表示为一个 d 维向量 $\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$,显然,每一个个体的位置都对应着目标问题的一个潜在解. 此外,帝王蝶优化算法按照适应度值将整个种群 P 分为子种群 SP1 (N_1 个个体) 和子种群 SP2 (N_2 个个体),迁移算子 (Migration Operator, MO) 和蝴蝶调整算子 (Butterfly Adjusting Operator, BAO) 分别作用于两个子种群.

在迁移算子中,先产生一个随机数 $r = \text{rand} * \text{peri}$,并且给出参数 p ,则产生新个体的公式为:

$$x_{i,k}^{t+1} = x_{j,k}^t \quad (5)$$

其中, $i=1, 2, \dots, N_1; k=1, 2, \dots, d; t$ 为当前进化代数; rand 是 $[0, 1]$ 区间服从均匀分布的随机数; peri 代表迁移周期; p 表示子种群 1 中帝王蝶所占比例. 如果 $r \leq p$, j 是在子种群 1 中随机选择的一个个体,否则,个体 j 随机选择于子种群 2.

在蝴蝶调整算子中,若 $\text{rand} \leq p$,则子种群 2 中个体按如下公式更新:

$$x_{i,k}^{t+1} = x_{\text{best},k}^t \quad (6)$$

其中, best 表示全局最优个体.

否则, 个体将采用如下公式进行位置更新:

$$x_{i,k}^{t+1} = x_{r1,k}^t \quad (7)$$

其中, $r1$ 是在子种群 2 中随机选择的一个个体.

在这种情况下, 继续判断. 若 $\text{rand} > \text{BAR}$, 对应用式 (6) 或式 (7) 的位置进一步更新为:

$$x_{i,k}^{t+1} = x_{i,k}^t + \alpha \times (dx_k - 0.5) \quad (8)$$

其中, dx 是个体 i 进行莱维行走的步长. α 表示权重系数, $\alpha = 1/t^2$.

考虑最小优化问题, 则帝王蝶优化算法的基本步骤如算法 1.

算法 1 MBO

- Step 1: 参数初始化. 令初始代数 $t = 0$, 设置最大迭代次数 MaxGen, 种群规模 N , 令子种群 1 规模 $N1$, 子种群 2 规模 $N2$. 设置蝴蝶调整率 BAR, 蝴蝶迁移周期 peri, 以及迁移率 p ;
- Step 2: 在待求解问题的搜索空间随机初始化帝王蝶种群;
- Step 3: 评估每个个体的适应度值;
- Step 4: 应用快速排序 (Quicksort) 算法对种群排序, 适应度值较小的 $N1$ 个个体组成子种群 1, 余下的 $N2$ 个个体形成子种群 2;
- Step 5: 利用 MO 算子更新子种群 1;
- Step 6: 利用 BAO 算子更新子种群 2;
- Step 7: 合并两个新生成的子种群为一个种群;
- Step 8: 判断是否满足终止条件. 若未满足, 则 $t = t + 1$, 转 Step 3.

4 差分进化帝王蝶优化算法求解 D{0-1} KP

4.1 差分进化

差分进化算法^[12]是美国学者 Storn 和 Price 为了模拟“优胜劣汰、适者生存”的自然进化法则于 1995 年提出的一种进化算法.

Price 和 Storn 先后提出了 10 余种不同的策略来生成差分向量以实现算法的变异操作. 本文选取 7 个目前使用最广、应用最为成功的变异策略, 分别简称为 DE1, DE2, ..., DE7, 其表示形式如下:

$$\text{DE1: } V_i = X_{r1} + F * (X_{r2} - X_{r3}) \quad (9)$$

$$\text{DE2: } V_i = X_{r1} + \lambda * (X_{r2} - X_{r3}) + F * (X_{r4} - X_{r5}) \quad (10)$$

$$\text{DE3: } V_i = X_{\text{best}} + F * (X_{r1} - X_{r2}) \quad (11)$$

$$\text{DE4: } V_i = X_{\text{best}} + \lambda * (X_{r1} - X_{r2}) + F * (X_{r3} - X_{r4}) \quad (12)$$

$$\text{DE5: } V_i = X_{r1} + \lambda * (X_{\text{best}} - X_{r2}) + F * (X_{r3} - X_{r4}) \quad (13)$$

$$\text{DE6: } V_i = X_i + \lambda * (X_{r1} - X_i) + F * (X_{r2} - X_{r3}) \quad (14)$$

$$\text{DE7: } V_i = X_i + \lambda * (X_{\text{best}} - X_i) + F * (X_{r2} - X_{r3}) \quad (15)$$

其中, X_{r1}, X_{r2}, X_{r3} 以及 X_{r4} 是在父代种群中随机选择的四个相异个体; X_{best} 表示种群当前最优个体; X_i 表示当前目标个体. $\lambda \in [0, 2]$, $F \in [0, 2]$ 称为缩放因子, 控制差分向量对 V_i 的影响. 由于在基本 MBO 的 MO 算子中, 新个体的产生并没有记忆整个种群的最优值, 同

时, 种群内部缺乏充分的信息共享, 因此, 本文在提出的 DEMBO 算法中替换 MO 算子中的式 (5) 为差分变异策略, 以达到通过种群内个体间的合作与竞争, 提高求解质量的目的, 称为 New Migration Operator (NMO).

4.2 混合编码方式和评估函数

基本 MBO 算法的 MO 算子和 BAO 算子对于连续问题的求解计算是封闭的, 而 D{0-1} KP 问题上的每一个可行解都对应于一个二进制向量, 因此, 直接应用基本 MBO 算法求解此类问题是不可行的.

本文针对 D{0-1} KP 问题的特点, 采用一种简单有效的混合编码机制^[12]. 我们将每一个候选解表示成一个二元组 $\langle X, Y \rangle$. 其中, $X \in [-a, a]^n$ 是一个实值向量, 构成求解问题的搜索空间, 本文中, 参数 a 取值为 5.0; $Y \in [0, 1]^n$ 是二进制向量, 表示问题的解空间. 为此, 需要构建从连续空间到离散空间的映射关系. 如式 (16) 所示:

$$h(x) = \begin{cases} 1, & \text{if } \text{sig}(x) \geq 0.5 \\ 0, & \text{else} \end{cases} \quad (16)$$

其中, $\text{sig}(x) = 1/(1 + e^{-x})$ 为 sigmoid 函数.

因此, 任意一个候选解 S 可由目标函数 f 评估.

$$f(S) = \sum_{i=0}^{3n-1} s_i p_i \quad (17)$$

其中, $S = \{s_0, s_1, \dots, s_{3n-1}\}$ 表示 D{0-1} KP 问题的一个候选解.

4.3 两阶段个体修复与优化策略

由式 (1) ~ (4) 可知, D{0-1} KP 问题是包含 $n + 1$ 个不等式约束条件的组合优化问题, 因此, 在求解该问题的过程中, 必然会产生不可行解. 应用演化算法处理不可行解的常用方法包括罚函数法, 修复法, 而基于贪心策略的个体修复方法是一种比罚函数法更加有效的解决 0-1 KP 问题不可行解的方法. 为此, 本文采用一种基于贪心策略的两阶段处理不可行解的算法^[9]: 不可行解修复算子 (Repair Operator, RO) 以及可行解优化算子 (Optimization Operator, OO). 算法详细流程参见文献 [9].

4.4 DEMBO 求解 D{0-1} KP 问题的基本框架

本文提出的 DEMBO 算法求解 D{0-1} KP 问题的主要步骤为: 首先在问题搜索空间随机初始化 N 个帝王蝶个体. 然后, 下面的过程重复进行直到终止条件到达: 适应度值较优的 $N1(N * p)$ 个个体组成子种群 1, 其余的 $N2(N - N1)$ 个个体组成子种群 2. 子种群 1 中的所有个体按照 NMO 算子完成位置更新, 子种群 2 中的所有个体按照 BAO 算子完成位置更新. 基于贪婪策略的两阶段 RO 算子与 OO 算子作用于所有个体, 即 RO 将所有的不可行解转换为可行解, OO 对所有的可行解进行进一步优化以期获得更高质量的解. 合并两个新生成的子种群. 最后, 输出 D{0-1} KP 问题的最优解.

DEMBO 算法求解 D{0-1} KP 问题流程描述如算法 2.

算法 2 DEMBO

- Step 1: 将 $3n$ 个物品项应用 Quicksort 按照价值-重量比率降序排列, 并将排序后物品项的下标依次存入数组 $H[0], H[1], \dots, H[3n-1]$;
- Step 2: 随机初始化规模为 N 的种群 $P(0) = (\langle X_1, Y_1 \rangle, \langle X_2, Y_2 \rangle, \dots, \langle X_N, Y_N \rangle)$, 设置当前代数 $t=1$, 最大迭代次数 MaxGen, 子种群 1 中帝王蝶比率 p , 蝴蝶调整率 BAR, 差分变异缩放因子 F . 计算种群中所有个体适应度值;
- Step 3: 划分整个种群为子种群 1 和子种群 2;
- Step 4: 将 NMO 算子作用于子种群 1;
- Step 5: 将 BAO 算子作用于子种群 2;
- Step 6: 对整个种群中所有个体执行 RO 算子和 OO 算子;
- Step 7: 应用式(17)评估种群中个体的适应度. 记录 $\langle X_{best}, Y_{best} \rangle$;
- Step 8: 重组两个子种群为一个种群;
- Step 9: 判断是否满足终止条件. 若未满足, 则 $t=t+1$, 转 Step 3;
- Step 10: 输出 $(Y_{best}, f(Y_{best}))$.

5 仿真结果分析

本节首先给出三类 D{0-1} KP 实例以及实验环境设置; 比较分析了不同差分变异策略对算法性能的影响; 然后, 比较分析 DEMBO、MBO、FirEGA 以及 SecEGA 求解三类大规模 D{0-1} KP 实例的综合性能; 接着, 平均收敛曲线图给出了 DEMBO 和 MBO 求解高维 UD-KP10、WDKP10 和 SDKP10 的实验结果分布以及收敛速度情况; 最后, 比较分析了四种算法求解三类大规模 D{0-1} KP 实例的基于最优值和平均值的近似比.

5.1 三类 D{0-1} KP 实例以及实验设置

本文中, 我们测试三类不同的 D{0-1} KP 实例^[9]: UD-KP, WDKP 和 SDKP, 每一类实例均包含 10 个规模分别为 300, 600, \dots , 3000 的实例, 分别称为 UD-KP1 ~ UD-KP10, WDKP1 ~ WDKP10 以及 SDKP1 ~ SDKP10.

所有实验在硬件配置为 Intel(R) Core(TM) i5-2415M CPU 2.30GHz、内存 4.00GB 的计算机上进行, 操作系统为 Microsoft Windows 7. 编程语言为 C++, 编译环境为 Visual C++ 6.0.

5.2 差分变异策略的选取

为了比较不同差分变异策略的效率, 我们将 7 种不同的差分变异策略分别与 MBO 结合, 得到 7 种不同的算法, 分别简记为 DE1MBO, DE2MBO, \dots , DE7MBO. 本文选取中等规模的不同类型的三个实例 (UDKP5, WDKP5 和 SDKP5) 来考察不同变异策略对算法性能的影响. 表 1 给出了 7 种算法求解 3 个实例的实验结果. 表 2 给出对 7 种算法的求解结果排名 (基于最优值). 其中粗体表示该算法获得的值更优.

为了消除随机性, 7 种算法在 3 个测试实例上均进行 30 次独立运行实验. 算法的种群规模均为 50, 迭代次数取值为实例规模. 算法的参数设置为: 迁移率 $p = 5/12$, 迁移周期 $peri = 1.2$, 蝴蝶调整率 $BAR = 5/12$, 差分变异缩放因子 $\lambda = F = 0.7$.

从表 1 和表 2 的实验结果可得到如下结论: DE4MBO 和 DE6MBO 在 3 个不同类型的测试实例上, 平均求解性能要好于其他算法. 其中, DE4MBO 在求解

表 1 7 种算法求解 UDKP5, WDKP5 和 SDKP5 的实验结果

		DE1MBO	DE2MBO	DE3MBO	DE4MBO	DE5MBO	DE6MBO	DE7MBO
UDKP5	Best	400614	385803	407821	419743	402309	415404	404367
	Mean	396265	181878	400393	414176	397445	410760	400171
	Worst	391434	378023	393543	410077	391787	403988	394610
WDKP5	Best	418991	410751	418854	424079	418697	422994	416531
	Mean	411378	403862	412739	419378	411964	417021	413208
	Worst	407701	399508	409771	416231	407807	412044	410613
SDKP5	Best	440780	437477	448420	443966	441618	456528	448837
	Mean	439201	436599	445411	442180	439364	454970	445864
	Worst	438014	435842	443065	440919	438499	453504	441030

表 2 7 种算法基于最优值的排名

	DE1MBO	DE2MBO	DE3MBO	DE4MBO	DE5MBO	DE6MBO	DE7MBO
UDKP5	6	7	3	1	5	2	4
WDKP5	3	7	4	1	5	2	6
SDKP5	6	7	3	4	5	1	2
Mean Rank	5	7	3.33	2	5	1.67	4

UDKP5 和 WDKP5 时,表现了最好的性能优势;DE6MBO 在求解 SDKP5 时,在最优值、平均值和最差值方面优于其它 6 种算法.此外,7 种算法在求解 UDKP5, WDKP5 和 SDKP5 时,根据其获得最优值的平均排名,可得性能排名为:

$$\text{DE6MBO} > \text{DE4MBO} > \text{DE3MBO} > \text{DE7MBO} > \\ \text{DE1MBO} = \text{DE5MBO} > \text{DE2MBO}$$

注意到,DE6MBO 和 DE4MBO 利用了两个差分向量,有利用保持种群的多样性,并且选择种群当前全局最优个体或者当前目标个体参与变异,有利于优秀基因的保留.因此,本文选择 DE4MBO 求解 30 个 DKP 实例,并将 DE4MBO 简记为 DEMBO.

5.3 性能比较与分析

为了验证本文提出的新算法 DEMBO 的有效性,将算法与文献[9]中的 FirEGA、SecEGA 以及原始 MBO 进行对比实验. FirEGA 以及 SecEGA 实验数据来源于文献[9]. 本文采用 Best, Worst, Mean, ARB, 以及 ARM 五个评价指标. Best, Worst, Mean 分别表示 MBO 和 DEMBO 在独立运行 30 次过程中获得的最好值、最差值以及数学期望值. ARB 和 ARM 分别定义为:

$$\text{ARB} = \text{Opt}/\text{Best} \text{ 和 } \text{ARM} = \text{Opt}/\text{Mean}.$$

其中,Opt 是通过动态规划法计算得出的实例最优值.

表 3 ~ 表 5 给出了 MBO、DEMBO、FirEGA 和 SecEGA 求解三类 D{0-1} KP 实例的计算结果. 其中粗体表示该算法获得的值更优.

表 3 MBO 和 DEMBO 求解 UDKP1 ~ UDKP10 的实验结果

		UDKP1	UDKP2	UDKP3	UDKP4	UDKP5	UDKP6	UDKP7	UDKP8	UDKP9	UDKP10
FirEGA	Best	80101	152969	244291	319680	403908	483350	564656	590237	652354	708744
	Mean	79325	151045	241061	316503	399525	478779	559815	584264	646592	703947
	Worst	78499	149732	239114	313141	396937	474558	555763	580258	642965	700702
SecEGA	Best	77974	148042	230485	306358	375619	447231	531192	560932	619444	689248
	Mean	76808	146310	225232	301700	371688	442556	523809	555100	615990	684872
	Worst	75624	144113	222118	299059	368445	438762	517579	545509	609077	662742
MBO	Best	80258	152472	238773	309230	385528	464213	582142	559297	618930	679796
	Mean	77174	147698	235683	301856	380589	459064	566332	551184	609164	664673
	Worst	74118	143564	231681	295719	376373	451324	549228	543788	598246	653391
DEMBO	Best	82402	160150	256683	334812	419743	508631	604087	615289	692628	755892
	Mean	79985	158155	252888	326817	414176	503308	599894	609888	682665	745821
	Worst	77881	155642	249151	321536	410077	498064	593232	601094	659413	718396

表 4 MBO 和 DEMBO 求解 WDKP1 ~ WDKP10 的实验结果

		WDKP1	WDKP2	WDKP3	WDKP4	WDKP5	WDKP6	WDKP7	WDKP8	WDKP9	WDKP10
FirEGA	Best	82722	137712	254234	314107	426783	463870	544059	574201	647012	677359
	Mean	82539	137225	253294	312343	424384	460750	541505	571594	644298	673776
	Worst	82454	136983	252909	310665	421584	455201	535551	565119	639241	660332
SecEGA	Best	82664	137712	254315	314227	426783	463435	544776	573862	648402	677118
	Mean	81118	135951	247826	308628	421415	461779	537821	567507	643444	672912
	Worst	80284	134490	236444	293697	391633	446741	501094	527204	589018	610343
MBO	Best	81147	135400	247724	302877	407791	439850	515569	545440	615941	639652
	Mean	80436	133340	245264	298674	402903	437224	511173	540800	609577	634521
	Worst	79440	130441	242049	294426	400007	431430	503549	534371	605719	629571
DEMBO	Best	82520	136093	246774	313736	424079	461642	542708	572206	646790	675618
	Mean	81699	135560	245802	312518	419378	460191	541455	570735	644372	674348
	Worst	80852	134924	244779	311294	416231	458633	539452	568086	636559	672031

表 5 MBO 和 DEMBO 求解 SDKP1 ~ SDKP10 的实验结果

		SDKP1	SDKP2	SDKP3	SDKP4	SDKP5	SDKP6	SDKP7	SDKP8	SDKP9	SDKP10
FirEGA	Best	93316	159116	235372	336369	451184	459236	607200	661104	728443	755189
	Mean	93192	158936	235204	335844	447335	458746	602797	659844	727364	752931
	Worst	93064	158798	235015	335524	444252	458427	600496	659120	726872	749879
SecEGA	Best	93009	159107	235474	336597	444748	458509	598648	662465	730036	756662
	Mean	91684	156557	231287	330437	435933	453973	592672	653459	726324	750716
	Worst	90256	154241	224872	318638	415923	430286	571469	610664	671623	697520
MBO	Best	93082	155956	229812	323785	443991	437841	591175	631136	691902	713077
	Mean	92021	153919	227082	320911	440484	433284	587621	624802	685393	709285
	Worst	90824	151047	224873	317396	436863	429364	584882	620375	679861	701457
DEMBO	Best	93248	159499	234686	335634	443966	458469	610956	656040	728168	753242
	Mean	92723	158589	233625	333513	442180	454943	607002	651069	725138	751871
	Worst	92238	156882	232553	331432	440919	452385	602995	644127	722374	749553

从表 3 可以看出, DEMBO 在求解 UDKP1 ~ UDKP10 时表现出最好的性能. 从表 4 和表 5 可以看出, DEMBO 在求解 WDKP 和 SDKP 实例问题时, 表现出比 FirEGA 略差的性能.

从表 3 ~ 表 5 可以看出, 对于所有的 30 个 D{0-1} KP 实例, DEMBO 算法在三个评价指标 Best, Mean 以及 Worst 上获得的结果要明显优于 MBO 算法. 对于三个高维的 UDKP10, WDKP10 以及 SDKP10, 搜索空间的范围为 2^{3000} , 求解难度较大, 而 DEMBO 的最差值甚至优于 MBO 的最优值和数学期望值, 这充分表明差分变异策略能够增加种群多样性, 提升 DEMBO 的求解性能.

为了进一步更加直观地比较 MBO 和 DEMBO 的平均求解性能, 验证差分变异策略对 DEMBO 收敛速度的影响, 图 1 ~ 图 3 给出两种算法独立运行 30 次求解高维实例 UDKP10, WDKP10 和 SDKP10 的平均收敛曲线. 从这些收敛曲线图可以明显看出, DEMBO 在三个高维 D{0-1} KP 实例上的收敛速度要远远快于 MBO, 同时收敛精度也远远好于 MBO. 显然, MBO 在求解高维 D{0-1} KP 实例时, 从进化开始就陷入了局部最优. 比较而言, DEMBO 生成的新个体能够保留部分个体和全局最优个体的有益信息, 加大了在搜索空间中处于较好

位置的概率, 加速算法收敛.

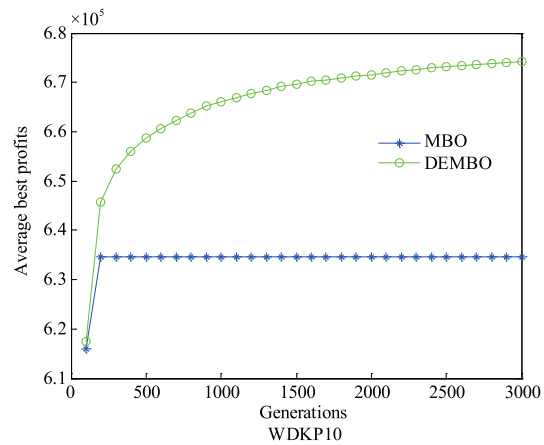


图2 MBO和DEMBO求解WDKP10的平均收敛曲线

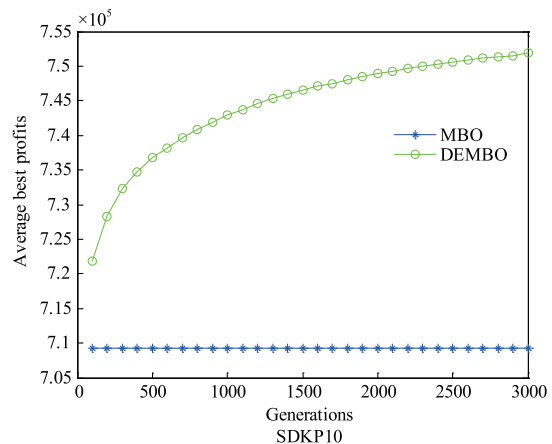


图3 MBO和DEMBO求解SDKP10的平均收敛曲线

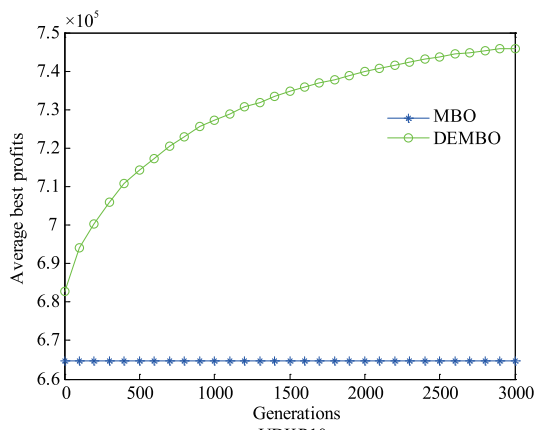


图1 MBO和DEMBO求解UDKP10的平均收敛曲线

近似算法的近似比是衡量近似算法求得的实例 I 的近似值与最优值之间的接近程度, 近似算法的近似比不会小于 1. 显然, 近似算法的近似比越大, 它求得的近似解就越差. 为此, 图 4 ~ 图 6 给出了 MBO、DEMBO、FirEGA 以及 SecEGA 求解 10 个 UDKP 实例, 10 个 WDKP 实例以及 10 个 SDKP 实例的 ARB、ARM 比较图. 其

中, Approximation ratio on the best value 对应 ARB, Approximation ratio on the mean value 对应 ARM.

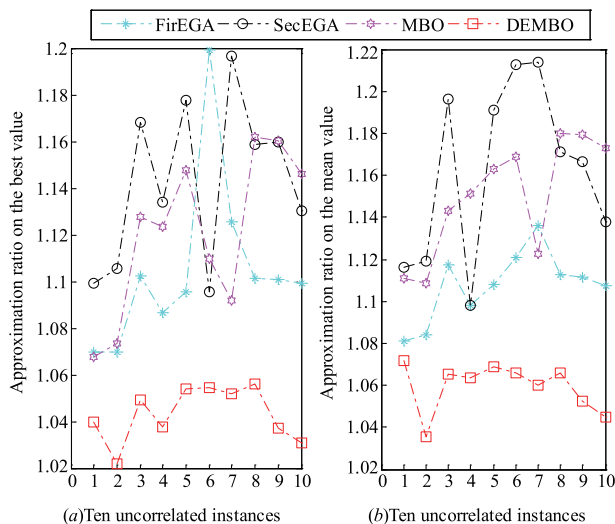


图4 四种算法求解UDKP实例的ARB和ARM比较图

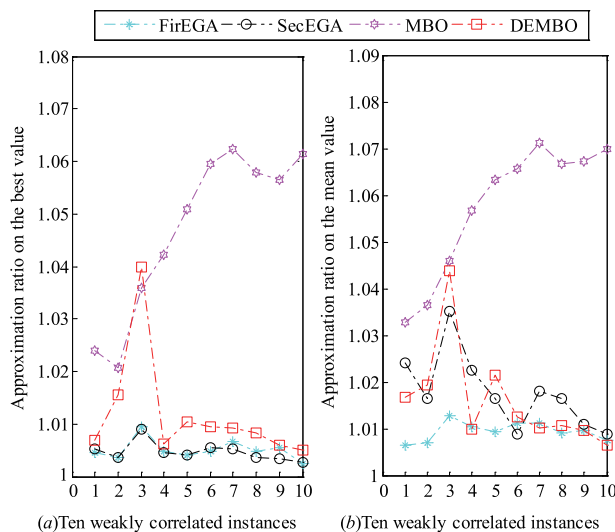


图5 四种算法求解WDKP实例的ARB和ARM比较图

从图 4 可以看出: DEMBO 在求解 UDKP1 ~ UDKP10时, ARB 的值小于 1.06, ARM 的值小于 1.08.

从图 5 可以看出: 对于所有的 10 个 WDKP 实例, FirEGA 和 SecEGA 的 ARB 值相差无几, 并且全部小于 1.01, DEMBO 的 ARB 值范围为 [1.01, 1.04], MBO 的 ARB 值范围为 [1.02, 1.07]. 此外, 随着问题维度的增加, FirEGA 和 SecEGA 的 ARB 值分布几乎表现为一条直线, 而 MBO 和 DEMBO 的 ARB 值曲线均有上升趋势. 说明对于 WDKP 实例, 问题复杂度增加, 从而导致 MBO 和 DEMBO 的性能稍有下降, 而 FirEGA 和 SecEGA 的性能保持稳定. 四种算法的 ARB 值的比较结果为: SecEGA ≤ FirEGA < DEMBO < MBO, 从四种算法的 ARM 值来看, FirEGA 性能最优, SecEGA 次之, DEMBO 排名

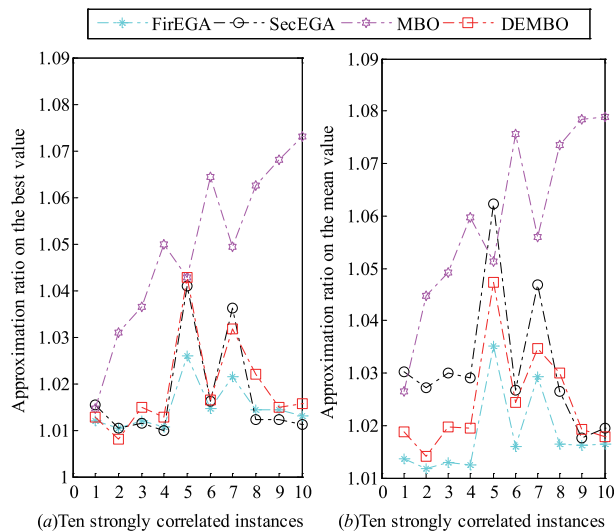


图6 四种算法求解SDKP实例的ARB和ARM比较图

第三, MBO 表现出最差的性能.

从图 6 可以看出: 对于所有的 10 个 SDKP 实例, 四种算法表现出如图 5 相似的性能. 对于 SDKP1-SDKP4 以及 SDKP8-SDKP10, FirEGA 以及 SecEGA 的 ARB 值非常接近于 1.01, DEMBO 的 ARB 值范围为 [1.01, 1.05], MBO 的 ARB 值范围为 [1.02, 1.07].

6 结语

在 MBO 算法中, MO 算子采用随机从两个子种群中选择个体重组生成新的个体, 全局最优个体信息并未保留, 易导致搜索经验丢失, 陷入局部最优. 为此, 本文提出了一种采用差分变异的方法生成子种群 1 中的新个体. 首先将 MBO 与 7 种经典的 DE 变异策略结合, 形成 7 种基于 MBO 的新算法, 并实验验证了不同变异策略对算法性能的影响, 选取了最优的 DE/best/2/bin 模式作为新的变异算子 (NMO), 利用四个帝王蝶个体之间的差异以及种群当前最优个体来生成第三个个体, 比 MBO 算法的 MO 算子更有效, 能增加种群多样性, 进而增加找到全局最优解的概率. 在三类物品价值与重量存在相关性的 30 个典型 D{0-1} KP 实例上进行了一系列的比较实验, 实验结果表明: 与基本 MBO 算法相比, 本文提出的 DEMBO 算法能在算法时间复杂度不变的前提下, 在提高解的精度和加快收敛速度方面都有明显改善.

参考文献

[1] Feng Y, Wang G G, Deb S, et al. Solving 0-1 knapsack problem by a novel binary monarch butterfly optimization [J]. Neural Computing & Applications, 2017, 28(7): 1619 - 1634.
 [2] Faris H, Aljarah I, Mirjalili S. Improved monarch butterfly

- optimization for unconstrained global search and neural network training[J]. Applied Intelligence, 2018, 48(2):445 – 464.
- [3] Chen S, Chen R, Gao J. A monarch butterfly optimization for the dynamic vehicle routing problem[J]. Algorithms, 2017, 10(3):107.
- [4] 刘会超, 吴志健. 基于旋转学习机制的差分演化算法[J]. 电子学报, 2015, 43(10):2040 – 2046.
Wu Hui-chao, Wu zhi-jian. Differential evolution algorithm using rotation-based learning[J]. Acta Electronica Sinica, 2015, 43(10):2040 – 2046. (in Chinese)
- [5] 阎大海, 李元香, 龚文引, 等. 一种求解约束优化问题的自适应差分进化算法[J]. 电子学报, 2016, 44(10):2535 – 2542.
Xia Da-hai, Li Yuan-xiang, Gong Wen-yin, et al. An adaptive differential evolution algorithm for constrained optimization problems [J]. Acta Electronica Sinica, 2016, 44(10):2535 – 2542. (in Chinese)
- [6] Zhu H, He Y, Wang X, et al. Discrete differential evolutions for the discounted {0-1} knapsack problem[J]. International Journal of Bio-inspired Computation, 2017, 10(4):219 – 238
- [7] Guldan B. Heuristic and Exact Algorithms for Discounted Knapsack Problems [D]. Bavaria, Germany: University of Erlangen-Nürnberg, 2007.
- [8] Rong A, Figueira J R, Klamroth K. Dynamic programming based algorithms for the discounted {0-1} knapsack problem[J]. Applied Mathematics & Computation, 2012, 218(12):6921 – 6933.
- [9] 贺毅朝, 王熙照, 李文斌, 等. 基于遗传算法求解折扣{0-1}背包问题的研究[J]. 计算机学报, 2016, 39(12).
He Yi-chao, Wang Xi-zhao, Li wen-bin, et al. Research on genetic algorithms for the discounted {0-1} knapsack problem[J]. Chinese Journal of Computers, 2016, 39(12). (in Chinese)
- [10] He YC, Wang XZ, He YL, et al. Exact and approximate algorithms for discounted {0-1} knapsack problem[J]. Information Sciences, 2016, 369:634 – 647.
- [11] 魏文红, 周建龙, 陶铭, 等. 一种基于反向学习的约束差分进化算法[J]. 电子学报, 2016, 44(2):426 – 436.
Wei Wen-hong, Zhou Jian-long, Tao Ming, et al. Constrained differential evolution using opposition-based learning[J]. Acta Electronica Sinica, 2016, 44(2):426 – 436. (in Chinese)
- [12] 贺毅朝, 王熙照, 寇应展. 一种具有混合编码的二进制差分演化算法[J]. 计算机研究与发展, 2007, 44(9):1476 – 1484.
He Y C, Wang X Z, Kou Y Z. A binary differential evolution algorithm with hybrid encoding[J]. Journal of Computer Research and Development, 2007, 44(9):1476 – 1484. (in Chinese)

作者简介



冯艳红 女, 1978 年生, 河北卢龙人, 河北地质大学信息工程学院副教授, 研究方向: 进化计算、群体智能及机器学习。
E-mail: qinfyh@163.com



杨娟 女, 1982 年生, 湖南邵阳人, 凯里学院数学科学学院副教授, 研究方向: 数学物理方程、近似算法。
E-mail: hnyangjuan1982@126.com

贺毅朝 男, 1969 年生, 河北晋州人, 硕士, 教授, 计算机学会 (CCF) 高级会员, 研究方向为进化算法理论与应用、算法设计与分析、计算复杂性理论和 Group Testing 理论。
E-mail: heyichao19@163.com

王改革 (通信作者) 男, 1984 年生, 山东临沂人, 博士, 研究方向为进化计算、群体智能及大数据优化。
E-mail: gaigewang@gmail.com