

基于贪婪离散类电磁机制算法求解背包问题

王建龙*, 孙合明

(河海大学 理学院, 南京 211100)

(* 通信作者电子邮箱 cheneyw@163.com)

摘要:针对基本类电磁机制算法不能够有效解决离散型的背包问题,提出了一种贪婪离散类电磁机制算法。首先,提出一种交叉操作;然后,利用提出的交叉操作对基本类电磁机制算法中的合力计算公式和粒子移动方法进行修改,使其能够适用于离散型问题;最后,引入贪婪算法的机制来处理经过类电磁机制算法迭代得到的解,使这些解满足背包问题的约束条件。通过对3个经典的背包测试问题进行的测试结果表明:该算法可以解决离散型的背包问题,并且具有较优的求解性能。

关键词:类电磁机制算法;背包问题;离散;约束条件;贪婪算法

中图分类号:TP301.6 **文献标志码:**A

Solution to knapsack problem based on greedy discrete electromagnetism-like mechanism algorithm

WANG Jianlong*, SUN Heming

(College of Sciences, Hohai University, Nanjing Jiangsu 211100, China)

Abstract: The basic Electromagnetism-like Mechanism (EM) algorithm cannot solve the discrete problems like knapsack problem. A new algorithm called Greedy Discrete Electromagnetism-like Mechanism (GDEM) algorithm was proposed. Firstly, a cross-operating was proposed; then the operating was used to modify the methods of the force calculation and particles movement in the basic EM algorithm; at last, greedy algorithm was introduced to process the constraint condition. Based on the three classical test knapsack problems, the results have shown that GDEM algorithm can solve knapsack problems efficiently.

Key words: Electromagnetism-like Mechanism (EM) algorithm; Knapsack Problem (KP); discrete; constraint condition; greedy algorithm

0 引言

类电磁机制 (Electromagnetism-like Mechanism, EM) 算法^[1-2]是一种新型的基于种群随机的全局优化算法。类电磁机制算法与遗传算法类似,其将问题的一个解比作一个带电粒子,在吸引-排斥机制的作用下,每个粒子都排斥较差的粒子,同时被较优的粒子吸引,从而找到全局最优的粒子。随着国内外学者对类电磁机制算法的深入研究,该算法的性能得到了提高,同时被很好地应用到了函数优化、资源管理、工程技术等领域^[3-9]。但针对离散型问题的研究还比较少,并且算法的性能也不够理想。

背包问题 (Knapsack Problem, KP) 是一种组合优化的 NP 完全问题^[10],解决好背包问题具有一定的实际意义。与背包问题类似的问题出现在了商业、组合数学、计算机复杂性理论、密码学和应用数学等领域。对于小规模背包问题,完全枚举法、分支定界法等一些传统方法可以高效地求解;而对于一些大规模的背包问题,传统优化算法出现了计算量大、求解结果不理想等问题。智能优化算法被应用到背包问题中,很好地解决了这样的问题^[11-14]。受此启发,本文提出一种贪婪离散类电磁机制算法,该算法不仅提出了一种交叉操作,可以较好地解决基本类电磁机制算法中的合力计算和粒子移动公式不能离散化的问题,还引入了贪婪算法,有效地处理了背包

问题中存在的约束条件。

1 相关问题

1.1 类电磁机制算法的基本原理

类电磁机制算法根据电磁理论中的吸引排斥机制,把每一个可行解当作一个带电的粒子,各粒子所带的电荷量由目标函数值决定,电荷量的大小影响该粒子对其他粒子的吸引或排斥作用,这样种群中所有的粒子就可以按照既定的法则进行更新,直至满足停止条件。如式(1)所示的函数优化问题:

$$\max f(X) \quad (1)$$

$$\text{s. t. } X \in [L, U]$$

$$[L, U] = \{X \in \mathbf{R}^n \mid l_k \leq x_k \leq u_k, k = 1, 2, \dots, n\}$$

需要求目标函数 $f(X)$ 在可行域中求最大值,其中 X 是决策变量。类电磁算法的求解过程主要有如下四步。

步骤1 初始化。从可行域中随机选取 m 个粒子 ($X_1, X_2, \dots, X_i, \dots, X_m$) 作为问题的初始解,并计算这个粒子的目标函数值 $f(X_i)$,找出目标函数值最大的粒子,记为最优粒子 X_{best} 。

步骤2 局部搜索。对每个粒子周围的位置进行随机搜索,如果找到一个粒子比当前粒子的目标函数值大,则更新当前粒子,否则当前粒子的位置保持不变。对种群中所有粒子进行局部搜索后,更新最优粒子 X_{best} 。

收稿日期:2013-04-07;修回日期:2013-05-23。

作者简介:王建龙(1988-),男,江苏兴化人,硕士研究生,主要研究方向:智能算法、运筹学;孙合明(1970-),男,山东诸城人,副教授,博士,主要研究方向:智能算法、人工智能。

步骤3 计算合力。首先根据种群中每个粒子 X_i 的目标函数值来计算该粒子所带的电荷量 q_i , 计算方法如式(2)所示:

$$q_i = \exp \left[-n \frac{f(X_i) - f(X_{best})}{\sum_{k=1}^m (f(X_k) - f(X_{best}))} \right];$$

$$i = 1, 2, \dots, m \quad (2)$$

然后根据所求得的电荷量,来计算粒子 X_i 所受的合力 F_i , 计算方法如式(3)所示:

$$F_i = \begin{cases} \sum_{j \neq i}^m (X_j - X_i) \frac{q_i q_j}{\|X_i - X_j\|^2}, & f(X_j) > f(X_i) \\ \sum_{j \neq i}^m (X_i - X_j) \frac{q_i q_j}{\|X_i - X_j\|^2}, & f(X_j) \leq f(X_i) \end{cases} \quad (3)$$

由此可看出,目标函数值较大的粒子对目标函数值较小的粒子具有吸引作用;反之,则产生排斥作用。

步骤4 移动粒子。粒子 X_i 将沿着合力 F_i 的方向移动,移动方法如式(4)所示:

$$X_i = X_i + \lambda \frac{F_i}{\|F_i\|} (RNG); \quad i = 1, 2, \dots, m \quad (4)$$

其中:步长 λ 是一个随机数, $\lambda \in (0, 1)$; RNG 是一个向量,其分(向)量表示对应的朝上边界 u_k 或者下边界 l_k 移动的可行步长。

经过以上四个步骤,所有的粒子完成一次迭代。如果满足停止条件,则停止迭代,输出最优解;否则,返回步骤1继续迭代。

1.2 背包问题

背包问题具体可以描述为有 n 只物品,物品 k 的重量为 w_k , 价值为 p_k ($k = 1, 2, \dots, n$)。现有一个背包容量为 C , 需从 n 只物品中选取一些物品装入背包。在所选物品总重量不超过背包容量的前提下,使得装入背包的物品的总价值最大。具体数学模型如下:

$$\max \text{value}(X) = \sum_{k=1}^n p_k x_k \quad (5)$$

$$\text{s. t.} \quad \sum_{k=1}^n w_k x_k \leq C \quad (6)$$

其中: $X = (x_1, x_2, \dots, x_k, \dots, x_n)$; x_k 只有两个值(分别是0和1), $x_k = 0$ 表示物品 k 未被选中, $x_k = 1$ 表示物品 k 被选中。

1.3 贪婪算法

贪婪算法是一种能够在某种标准下选取最优解的分级处理方法,它总是选择在当前看来是最优的解,所以用贪婪算法求得的解只能看作是近似最优解。例如在上述背包问题,可以直接利用贪婪算法进行求解。计算所有物品的价值密度 p_k/w_k , 并按照价值密度进行降序排列,得到一个队列 $r[k]$, 其中 $r[1]$ 表示价值密度最大的物品的序号, $r[2]$ 次之,以此类推。按照队列 $r[k]$ 中的序号,依次选择物品装入背包,直到背包装满为止,这样就能求得上述背包问题的一个近似最优解。

2 贪婪离散类电磁机制算法

2.1 基本原理

基本类电磁机制算法适用于连续型的优化问题,为了把类电磁机制算法应用到离散型的优化问题中,本文对基本类电磁机制算法进行修改,并且加入贪婪算法的机制,来处理离

散型问题的约束条件。

2.1.1 初始化

离散型优化问题一般采用二进制编码,即每个解向量的分量为0或1。在 GDEM 算法中,本文采用混沌机制对初始粒子进行赋值,方法如式(7)所示:

$$X_i^k = \begin{cases} 0, & \text{rand}() < P_{\text{initial}} \\ 1, & \text{rand}() \geq P_{\text{initial}} \end{cases} \quad (7)$$

其中: X_i^k 为粒子 i ($i = 1, 2, \dots, m$) 第 k ($k = 1, 2, \dots, n$) 维分量的取值, P_{initial} 是一个常数。另外接下来本文中如果无特别说明, m 都为种群规模, n 都为问题的维数, $\text{rand}()$ 是区间 $(0, 1)$ 上的一个随机数。

2.1.2 计算合力

基本类电磁机制算法中,合力确定了粒子间的吸引或排斥作用,其计算方法如式(3)所示。这样的计算方法不适用于离散型的优化问题,为了解决这样的问题,本文定义了一个新的运算——交叉操作,记为 \otimes 。假设 X_1 和 X_2 是种群中两个粒子,即长度相同的两个向量,它们的分量为0或1。 $X_1 \otimes X_2$ 的计算方法如下:

```
For k = 1 to m
  If (rand() < q1/(q1+q2))
    NEW(k) = X1^k
  Else
    NEW(k) = X2^k
  End If
End For
Return NEW
```

其中: q_1 和 q_2 是根据式(2)计算出的两个粒子的电荷量, NEW 是 X_1 和 X_2 交叉操作后产生的新向量。从上述过程可以看出,粒子 i 所带电荷量越大,其编码信息越可能被保留在新产生的向量中。在 GDEM 算法中,本文把种群中每个粒子 i 与其他所有粒子进行交叉操作,将最后得到的向量作为粒子 i 的合力,即移动方向。算法的伪代码描述如下:

```
For i = 1 to m
  F_i = X_i
  For j = 1 to m
    If (i ≠ j)
      F_i = F_i ⊗ X_j
    End If
  End For
End For
Return F
```

2.1.3 移动粒子

基本类电磁机制算法根据粒子的合力对粒子进行移动,移动方法如式(4)所示。在 GDEM 算法中,本文修改了合力的计算方法,那么粒子的移动方法也要进行修改。本文同样采用交叉操作对粒子进行移动。在移动粒子前,本文加入了变异操作来取代基本类电磁机制算法中的局部搜索,这样有利于保持种群的多样性,避免出现局部收敛。粒子 X_i 变异后得到粒子 $temp$, 如果粒子 $temp$ 较粒子 X_i 更好,则直接更新粒子, $X_i = temp$; 否则采用上述的交叉操作更新粒子 X_i 。算法的伪代码描述如下:

```
For i = 1 to m
  temp = X_i, t = [rand() * n]
  Flip temp(t)
  If (value(temp)) > value(X_i)
    X_i = temp
```

```
Else
     $X_i = X_i \otimes F_i$ 
End If
```

```
End For
```

其中: $t = [\text{rand}() * n]$ 是对中括号内的值向上取整; Flip 为转换数值操作,即 0 变成 1,1 变成 0; $\text{value}()$ 是根据式(5)算出的装入背包的物品的总价值。

2.1.4 约束条件的处理

在非线性约束优化问题中,有多种处理约束条件的方法,最常用的就是利用罚函数法,把有约束的优化问题转变为无约束的优化问题进行求解。在处理较小规模的问题时,罚函数法可以快速求得最优解,但在处理高维的非线性约束优化问题时,罚函数法出现了求解效率低下、迭代时间过长的问題。在 GDEM 算法中,本文采用贪婪算法的机制来处理式(6)。假设 $X = (x_1, x_2, \dots, x_n)$ 是非正常编码,即粒子 X 不满足式(6), $Y = (y_1, y_2, \dots, y_n)$ 是修正后的正常编码。首先将粒子 X 中所有 $x_k = 1$ 的物品按其价值密度 p_k/w_k 进行降序排列,得到一个队列 $r[k]$,其中 $r[1]$ 为价值密度最大的物品的序号, $r[2]$ 次之,以此类推。然后对非正常编码进行修正。

2.2 算法流程

```
k = 1, sum = wr[1]
While (sum ≤ C) Do
    yr[1] = 1;
    k = k + 1;
    sum = sum + wq[k]
End While
For j = k To n
    yr[j] = 0
End For
Return Y = (y1, y2, ..., yn)
```

本文提出的 GDEM 算法求解背包问题的主要步骤如下。

步骤 1 初始化。确定问题的维数 n , 种群的规模 m , 最大迭代次数 MAXITER , 常数 P_{initial} , 背包容量 C 。根据 2.1.1 节的混沌机制进行初始化,确定初始种群,并根据式(5)计算种群中每个粒子 X_i 的目标函数值,找出目标函数值最大的最优粒子,记为 X_{best} 。

步骤 2 计算合力。首先根据式(2)计算种群中每个粒子 X_i 的电荷量 q_i ,再利用 2.1.2 节中的交叉操作计算粒子所受的合力 F_i 。

步骤 3 移动粒子。根据 2.1.3 节中的方法,先对粒子进行变异操作,再利用交叉操作移动粒子。

步骤 4 修正粒子。根据 2.1.4 节中的约束条件的处理方法,对非正常编码的粒子进行修正。

步骤 5 停止条件。检查是否满足停止条件,如果满足,则停止迭代,输出最优解;否则,返回步骤 2,继续迭代。

3 实验测试

为了验证算法的有效性,本文选取多个实例对 GDEM 算法进行测试,并将结果与其他智能算法以及其他改进类电磁机制算法进行比较。本文所有测试采用 Matlab R2009a 编程,计算机 CPU 为 Intel Core 2 2.66 GHz, RAM 为 2 GB。

3.1 测试用例

下面给出 3 个常用的测试用例,用来测试 GDEM 算法的性能。

实例 1 物品的价值集 $P = [220, 208, 198, 192, 180, 180, 165, 162, 160, 158, 155, 130, 125, 122, 120, 118, 115, 110, 105,$

$101, 100, 100, 98, 96, 95, 90, 88, 82, 80, 77, 75, 73, 72, 70, 69, 66, 65, 63, 60, 58, 56, 50, 30, 20, 15, 10, 8, 5, 3, 1]$, 物品的重量集 $W = [80, 82, 85, 70, 72, 70, 66, 50, 55, 25, 50, 55, 40, 48, 50, 32, 22, 60, 30, 32, 40, 38, 35, 32, 25, 28, 30, 22, 50, 30, 45, 30, 60, 50, 20, 65, 20, 25, 30, 10, 20, 25, 15, 10, 10, 10, 4, 4, 2, 1]$, 背包的最大容量 $C = 1000$, 问题的规模 $n = 50$ 。

实例 2 物品的价值集 $P = [597, 596, 593, 586, 581, 568, 567, 560, 549, 548, 547, 529, 529, 527, 520, 491, 482, 478, 475, 475, 466, 462, 459, 458, 454, 451, 449, 443, 442, 421, 410, 409, 395, 394, 390, 377, 375, 366, 361, 347, 334, 322, 315, 313, 311, 309, 296, 295, 294, 289, 285, 279, 277, 276, 272, 248, 246, 245, 238, 237, 232, 231, 230, 225, 192, 184, 183, 176, 174, 171, 169, 165, 165, 154, 153, 150, 149, 147, 143, 140, 138, 134, 132, 127, 124, 123, 114, 111, 104, 86, 74, 63, 62, 58, 55, 48, 27, 22, 12, 6]$, 物品的重量集 $W = [54, 183, 106, 82, 30, 58, 71, 166, 117, 190, 90, 191, 205, 128, 110, 89, 63, 6, 140, 86, 30, 91, 156, 31, 70, 199, 142, 98, 178, 16, 140, 31, 24, 197, 101, 73, 169, 73, 92, 159, 71, 102, 144, 151, 27, 131, 209, 164, 177, 177, 129, 146, 17, 53, 164, 146, 43, 170, 180, 171, 130, 183, 5, 113, 207, 57, 13, 16, 20, 63, 12, 24, 9, 42, 6, 109, 170, 108, 46, 69, 43, 175, 81, 5, 34, 146, 148, 114, 160, 174, 156, 82, 47, 126, 102, 83, 58, 34, 21, 14]$, 背包的最大容量 $C = 6718$, 问题的规模 $n = 100$ 。

实例 3 物品重量 w_k 是区间 $[1, 10]$ 上的随机数, e_k 是区间 $[0, 5]$ 上的随机数, 物品价值 $p_k = w_k + 5$, 背包容量 $C = 0.5 \sum_{k=1}^n w_k$, 问题规模 n 分别取 100, 250 和 500 来进行测试。

3.2 GDEM 算法与其他算法性能比较

用实例 1 和实例 2 分别进行测试,其中常数 $P_{\text{initial}} = 0.5$, 种群规模 $m = 20$, 最大迭代次数 $\text{MAXITER} = 1000$ 。每个算法都进行 10 次求解,取最大值为算法最后的求解结果,并与文献[16]中的离散粒子群(Discrete Particle Swarm Optimization, DPSO)算法、模式替代进化算法(Schema Replaced Evolutionary Algorithm, SREA)进行比较。具体情况如表 1 所示, DPSO 算法求解实例 1 的结果中,最优解编码为 49 位,所以其结果不作为参考依据。

表 1 三种算法的求解实例 1 和实例 2 结果对比

| 实例 | SREA | | DPSO 算法 | | GDEM 算法 | |
|------|-----------|-----------|-----------|-----------|-----------|-----------|
| | 最优解 编码 | 价值/ 容量 | 最优解 编码 | 价值/ 容量 | 最优解 编码 | 价值/ 容量 |
| 实例 1 | 110101011 | | 110100111 | | 110101011 | |
| | 110110110 | | 110100110 | | 110110110 | |
| | 110111111 | 3 103/ | 110111111 | 3 112/ | 110111111 | 3 103/ |
| | 101000010 | 1 000 | 111000010 | 1 000 | 101000010 | 1 000 |
| | 100110000 | | 110100000 | | 100110000 | |
| 实例 2 | 01000 | | 1010 | | 01000 | |
| | 111111111 | | 111111111 | | 111111111 | |
| | 111111111 | | 111111111 | | 111111111 | |
| | 111111111 | | 111111111 | | 111111111 | |
| | 111111111 | | 111111111 | | 111111111 | |
| | 111111111 | | 111111111 | | 111111111 | |
| | 101111111 | 26 559/ | 101111111 | 26 559/ | 101111111 | 26 711/ |
| | 101000101 | 6 717 | 101000101 | 6 717 | 011000101 | 6 715 |
| | 101101111 | | 101101111 | | 101101111 | |
| | 111000111 | | 111000111 | | 111000111 | |
| | 011100000 | | 011100000 | | 011100000 | |
| | 000000000 | | 000000000 | | 000000000 | |
| | 1 | | 1 | | 1 | |

图 1~2 为实例 1 和实例 2 的求解过程图。

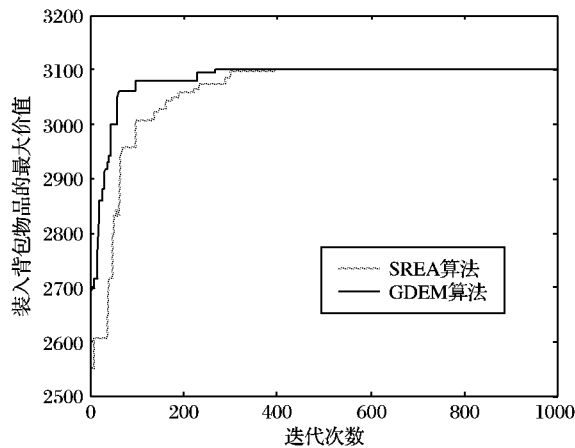


图 1 两种算法求解实例 1 的过程

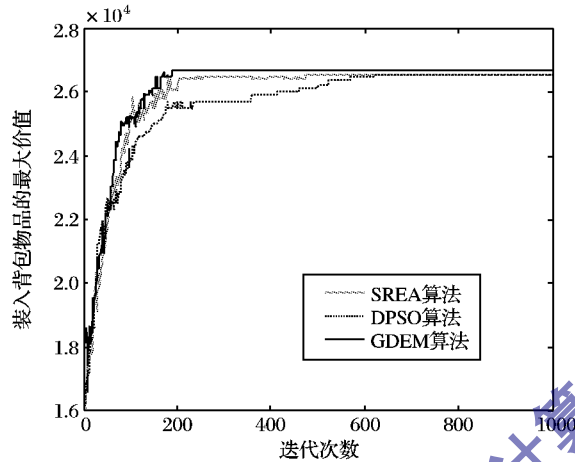


图 2 三种算法求解实例 2 的过程

从表 1 可以看出,本文提出的 GDEM 算法可以解决实例 1~2 所述的两个典型的背包问题。并且,GDEM 算法在求解问题规模较大的实例 2 时,求解结果要优于 SREA 和 DPSO 算法,这也说明了 GDEM 算法求解大规模的离散型的背包问题具有较优异的性能。同时从图 1~2 也可以看出,GDEM 算法求得最后最优解的速度明显快于其他两个算法。

3.3 GDEM 算法与其他改进 EM 算法的性能比较

目前对于基本类电磁机制算法的离散化改进较少,典型的有离散类电磁机制算法解决装配序列问题^[7],离散类电磁机制算法解决旅行商(Travelling Salesman Problem, TSP)问题^[8],还有量子类电磁机制(Quantum-inspired Electromagnetism-like Mechanism,QEM)算法解决 0/1 背包问题^[9]等。本节利用实例 1~3 对 GDEM 算法进行测试,并与文献[9]中的 QEM 算法进行对比。本次实验最大迭代次数 MAXITER = 1000,种群规模 m = 20,实例 3 的问题规模分别取 100,250 和 500。首先对实例 1 和实例 2 分别进行 5 次求解,记录每一次的物品价值和对应使用的背包容量,具体情况见表 2~3。然后对实例 3 中每一种群体规模都进行 10 次求解,记录最大值、平均值以及最小值,具体情况见表 3。两种算法求解实例 3 的过程如图 3~5 所示。

由表 2~3 可看出,GDEM 算法在求解实例 1 和实例 2 时,结果好于 QEM 算法,并且性能比较稳定。由表 4 以及图 3~5 可以看出,当问题规模为 100 时 GDEM 算法所求得

最大值为 502.8,优于 QEM 所求得的 499.9,同时 GDEM 算法所求解的平均值和最劣值都要优于 QEM 算法。问题规模为 250 和 500 时,情况与上述相同。可见 GDEM 算法是一种有效的求解离散型背包问题的算法,并且算法性能要优于先前所提出的离散型的类电磁机制算法。

表 2 两种算法求解实例 1 的结果对比

| 求解 序号 | QEM 算法 | | GDEM 算法 | |
|----------|--------|-------|---------|-------|
| | 物品价值 | 使用容量 | 物品价值 | 使用容量 |
| 1 | 3 103 | 1 000 | 3 103 | 1 000 |
| 2 | 3 095 | 996 | 3 103 | 1 000 |
| 3 | 3 100 | 1 000 | 3 103 | 1 000 |
| 4 | 3 103 | 1 000 | 3 100 | 1 000 |
| 5 | 3 103 | 1 000 | 3 103 | 1 000 |

表 3 两种算法求解实例 2 的结果对比

| 求解 序号 | QEM 算法 | | GDEM 算法 | |
|----------|--------|-------|---------|-------|
| | 物品价值 | 使用容量 | 物品价值 | 使用容量 |
| 1 | 26 682 | 6 748 | 26 706 | 6 716 |
| 2 | 26 688 | 6 688 | 26 700 | 6 709 |
| 3 | 26 559 | 6 717 | 26 711 | 6 715 |
| 4 | 26 694 | 6 708 | 26 706 | 6 716 |
| 5 | 26 559 | 6 717 | 26 706 | 6 716 |

表 4 两种算法求解实例 3 的结果对比

| 问题 规模 | QEM 算法 | | | GDEM 算法 | | |
|----------|---------|---------|---------|---------|---------|---------|
| | 最大值 | 平均值 | 最小值 | 最大值 | 平均值 | 最小值 |
| 100 | 499.9 | 479.6 | 436.7 | 502.8 | 482.3 | 452.6 |
| 250 | 1 201.5 | 1 127.5 | 1 103.2 | 1 212.4 | 1 163.2 | 1 121.9 |
| 500 | 2 291.5 | 2 213.9 | 2 186.4 | 2 300.8 | 2 249.2 | 2 203.6 |

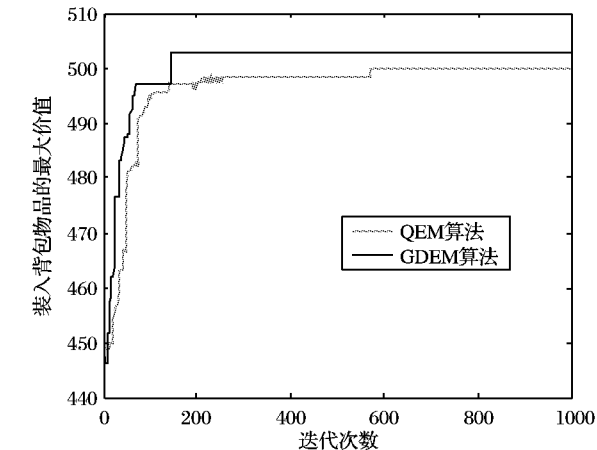


图 3 两种算法求解实例 3 的过程(问题规模为 100)

4 结语

为了利用类电磁机制算法解决离散型的背包问题,本文提出了一种交叉操作,对基本类电磁机制算法的合力计算方法以及粒子移动方法进行修改,并将贪婪算法中的选择机制引入到算法中,对背包问题中的约束条件进行处理,提出了一种贪婪离散类电磁机制算法。通过对 3 个常用的背包问题的求解测试,验证了 GDEM 算法的有效性;同时将 GDEM 算法的求解结果与其他智能算法进行比较,对比结果说明了 GDEM 算法具有较优异的性能。

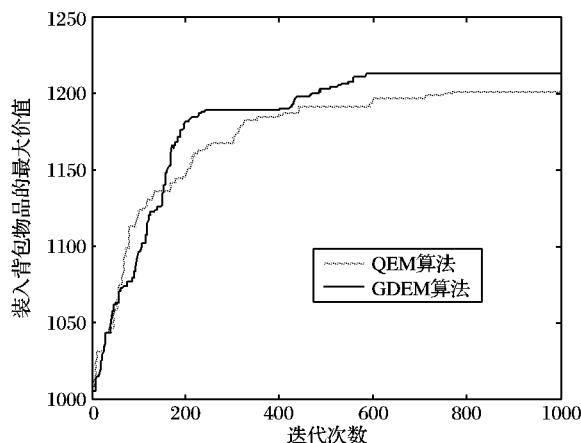


图4 两种算法求解实例3的过程(问题规模为250)

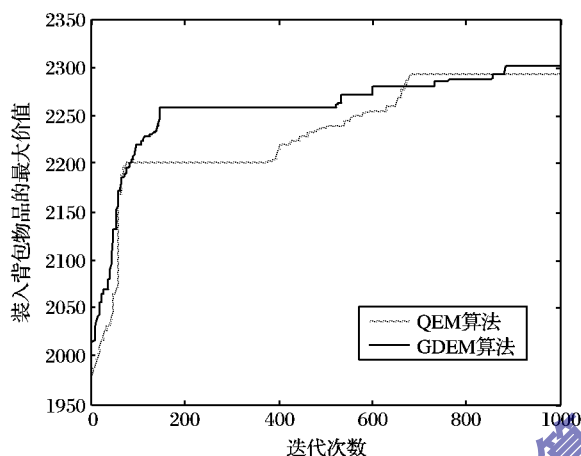


图5 两种算法求解实例3的过程(问题规模为500)

参考文献:

- [1] BIRBIL S I, FANG S C. An electromagnetism-like mechanism for global optimization[J]. *Journal of Global Optimization*, 2003, 25(3): 263–282.
- [2] BIEBIL S I. Stochastic global optimization techniques[D]. Raleigh: North Carolina State University, Department of Industrial Engineering, 2002.
- [3] 单玉乐, 曾建潮, 谭瑛. 一种改进的无局部搜索的类电磁机制算法[J]. *太原科技大学学报*, 2010, 31(6): 437–440.
- [4] 李如琦, 李芝荣, 凌武能, 等. 基于类电磁机制算法的配电网重构[J]. *电力系统保护与控制*, 2012, 40(14): 116–120.
- [5] 张智晟, 龚文杰, 段晓燕, 等. 类电磁机制算法在水电站厂内经济运行中的应用研究[J]. *电工电能新技术*, 2011, 30(4): 17–20.
- [6] 张科, 曹平. 复杂边坡非圆弧滑动面求解的类电磁机制算法[J]. *中南大学学报: 自然科学版*, 2011, 42(10): 3125–3130.
- [7] 孙禄, 张春江, 高亮, 等. 基于离散类电磁机制算法的装配序列规划[J]. *机械科学与技术*, 2012, 31(3): 353–358.
- [8] NIKBAKHS A, MOHSEN G A, REZA T. A discrete binary version of the electromagnetism-like heuristic for solving traveling salesman problem [C]// *Advanced Intelligent Computing Theories and Applications: with Aspects of Artificial Intelligence*. Berlin: Springer-Verlag, 2008: 123–130.
- [9] CHOU Y, CHANG C, CHIU C, et al. Classical and quantum-inspired electromagnetism-like mechanism for solving 0/1 knapsack problems[C]// *Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics*. Piscataway, NJ: IEEE Press, 2010: 3211–3218.
- [10] GAREY M R, JOHNSON D S. Computers and intractability: a guide to the theory of NP-completeness[M]. San Francisco: W. H. Freeman, 1979.
- [11] 吕晓峰, 张勇亮, 马羚. 一种求解0-1背包问题的改进遗传算法[J]. *计算机工程与应用*, 2011, 47(34): 44–46.
- [12] 何小锋, 马良. 求解0-1背包问题的量子蚁群算法[J]. *计算机工程与应用*, 2011, 47(16): 29–31.
- [13] 库向阳, 朱命昊, 赵亚敏. 求解0/1背包问题的改进人工鱼群算法研究[J]. *计算机工程与应用*, 2011, 47(21): 43–46.
- [14] 马丰宁, 谢龙, 郑重. 求解背包问题的基因属性保留遗传算法[J]. *天津大学学报*, 2010, 43(11): 1020–1024.
- [15] 汪定伟, 王俊伟, 王洪峰, 等. 智能优化方法[M]. 北京: 高等教育出版社, 2007.
- [16] 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.

(上接第2489页)

这个问题), 所以要想解决这个问题, 必须要修改 Hadoop 核心, 解决思路是: 每个 mapper 向带有编号的 HDFS block 写内容, 最后通过“汇总”这些 HDFS block 编号, 利用 Hadoop 元数据^[12]在逻辑上再形成一个 HDFS 大文件。

参考文献:

- [1] WHITE T. Hadoop: the definitive guide [M]. 影印版. 3版. 南京: 东南大学出版社, 2011.
- [2] ZIKOPOLOS P C, EATON C, DEROOS D, et al. Understanding big data: analytics for enterprise class hadoop and streaming data [M]. New York: McGraw-Hill, 2012.
- [3] HIDALGO C A. How to transform big data into knowledge[EB/OL]. [2013-02-10]. <http://enterprise.huawei.com/en/about/e-journal/ict/detail/hw-195167.htm>.
- [4] ABOUZEID A, BAJDA-PAWLIKOWSKI K, ABADI D J, et al. HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads[C]// *Proceedings of the 35th International Conference on Very Large Data Bases*. New York: ACM Press, 2009: 733–743.
- [5] 覃雄派, 王会举, 杜小勇, 等. 大数据分析——RRDBMS 与 MapReduce 的竞争与共生[J]. *软件学报*, 2012, 23(1): 32–45.
- [6] 王珊, 王会举, 覃雄派, 等. 架构大数据: 挑战、现状与展望[J]. *计算机学报*, 2011, 34(10): 1741–1752.
- [7] Apache Sqoop Project. Sqoop User Guide (V1.4.3) [EB/OL]. [2013-02-10]. <http://sqoop.apache.org/docs/1.4.3/SqoopUserGuide.html>.
- [8] HWANG K, FOX G, DONGARRA J. Distributed and cloud computing: from parallel processing to the Internet of things[M]. 北京: 机械工业出版社, 2012.
- [9] TANENBAUM A S, VAN-STEEN M. Distributed systems principles and paradigms[M]. 北京: 清华大学出版社, 2012.
- [10] 周志明. 深入理解 Java 虚拟机[M]. 北京: 机械工业出版社, 2011.
- [11] Apache Ambari Project. Ambari user guide [EB/OL]. [2013-02-10]. <http://incubator.apache.org/ambari/1.2.2/installing-hadoop-using-ambari/content/index.html>.
- [12] 文艾, 王磊. Hadoop 分布式文件系统深度实践[M]. 北京: 清华大学出版社, 2012.