基于蚁群混沌行为的离散粒子群算法及其应用

徐青鹤 刘士荣 吕 强
(杭州电子科技大学自动化研究所 杭州 310018)

摘 要 考虑蚁群算法与粒子群算法的各自特点,在粒子群算法的基础上借鉴蚁群算法的信息素机制,对粒子群算法的速度位置更新公式重新定义,提出了一种基于蚁群混沌行为的离散粒子群算法,并将其应用到背包问题中。实验结果表明,该算法可以得到较优解。

关键词 信息素机制,混沌,离散粒子群,背包问题
中图法分类号 TP18 **文献标识码** A

Discrete Particle Swarm Optimization Based on Chaotic Ant Behavior and its Application

XU Qing-he LIU Shi-rong LV Qiang
(Institute of Automation, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract Considering their own characteristics of ant colony algorithm and particle swarm optimization algorithm, the update equations of the speed and position of particles were redefined on the basis of PSO algorithm. A discrete particle swarm optimization algorithm based on chaotic ant behavior was proposed using the idea of pheromone refresh mechanism of ant colony algorithm for reference. Knapsack problem was used to test the performance of the algorithm. Compared with other algorithms, the results of the experiment show that the proposed algorithm can result in better profits.

Keywords Pheromone mechanism, Chaotic, Discrete particle swarm optimization, Knapsack problem

粒子群算法(Particle Swarm Optimization, PSO)由 Eberhart 和 Kennedy 于 1995 年提出[1],它源于对鸟类捕食行为的研究,是一种进化优化算法,最初主要应用于连续优化问题。此后,为了解决组合优化问题,Kennedy 等[2]在 1997 年又提出了一种离散二进制 PSO 算法。目前有大量的研究人员在研究 PSO 算法,但大多数研究都针对连续优化问题[3-5],只有少数学者针对一些具体问题,提出了一些离散粒子群算法[6,7]。

0-1 背包问题是一类经典的组合优化 NP 完全问题[8],对该问题求解方法的研究无论是理论上还是在实践中都具有一定的意义,如资源配置、装载问题、投资决策等均可建模为背包问题。对于一些小规模的背包问题,传统的方法(如隐枚举法、分支定界法等)可以快速得到精确的最优值,但在求解大规模的背包问题上传统的方法就存在计算量大、迭代时间过长、求解效果不佳等弱点。此后,进化算法、遗传算法等仿生方法相继被用于解决背包问题,最近一些改进的进化算法及遗传算法能够较成功地求解背包问题[9,10]。在粒子群算法方面,最近有学者将量子理论应用到粒子群算法中,提出了离散化量子群算法,该算法对背包问题也能求解到较满意的解[11]。

针对二进制离散化问题,本文提出了基于蚁群混沌行为的离散粒子群算法。算法借鉴了蚁群算法的信息素共享机制,对粒子群的初始化进行混沌操作,并在一定条件下进行重新初始化。相关实验表明,该算法能成功解决 0-1 背包问题,并且性能较优。

1 相关问题描述及实现

1.1 背包问题描述

背包问题可以描述为有 m 个物品,其质量为 ω_i,价值为 p_i,$i\in\{1,\cdots,m\}$,在给定的背包容量 C 条件下,要求所选取的物品价值最大,其数学描述如下:

$$\max f(X)=\sum_{i=1}^{m}p_i x_i \tag{1}$$

$$s.t.\ \sum_{i=1}^{m}\omega_i x_i\leqslant C \tag{2}$$

式中,$X=\{x_1,x_2,\cdots,x_m\}$,$x_i=0$ 或者 $x_i=1$。当 $x_i=1$ 时,表示第 i 个物品被选中;$x_i=0$ 时,表示第 i 个物品未被选中。

1.2 基本粒子群算法原理

PSO 算法中的每个个体都称为一个粒子。每个粒子模仿鸟的寻食行为,通过跟踪两个"极值"来搜索解空间的最优值:一个是每个粒子当前已搜索到的最优值,称为个体极值;另一个是整个群体当前已搜索到的最优值,称为全局极值。其速度位置更新公式如下:

$$v_i^{k+1}=\omega v_i^k+c_1\xi(p_i^k-x_i^k)+c_2\eta(p_g^k-x_i^k) \tag{3}$$

$$x_i^{k+1}=x_i^k+v_i^{k+1} \tag{4}$$

式中,$v_i=[v_{i1},v_{i2},\cdots,v_{in}]$是粒子 i 的速度,代表了粒子 i 现

到稿日期:2009-06-08 返修日期:2009-08-17 本文受国家自然科学基金资助项目(60675043),浙江省科技计划项目(2007C21051),杭州电子科技大学科研启动基金项目(KYS09150543)资助。

徐青鹤(1984—),男,硕士生,主要研究方向为智能优化算法,E-mail:xuqinghe1984411@126.com;刘士荣(1951—),男,博士生导师,主要研究方向为复杂系统建模、控制与优化、智能系统与智能机器人等;吕 强(1977—),男,讲师,主要研究方向为群体智能技术、机器人的导航与控制。

在位置与其下一步目标位置之间的距离；$x_i=[x_{i1},x_{i2},\cdots,x_{in}]$是粒子 i 的目前位置；p_i 是迄今搜索到的个体最优解；p_g 是整个群体迄今搜索到的最优解；ω 是保持原来速度的系数，称为惯性权重；c_1 是粒子跟踪自己历史最优值的权重系数，它表示粒子自身的认识，称为认知常数；c_2 是粒子跟踪群体最优值的权重系数，它表示粒子对整个群体知识的认识，称为社会知识常数；ξ，η 是[0,1]区间内均匀分布的随机数。粒子群算法就是依据粒子自身速度进行惯性运动并对粒子自身的行为进行思考，同时还参与群体信息共享与相互合作，从而在粒子群中寻找最佳位置。这 3 部分的相互作用与制约决定了算法的寻优性能。

2 基于蚁群混沌行为的离散粒子群算法原理及实现

2.1 基于蚁群混沌行为的离散粒子群算法原理

二进制离散粒子群算法与基本粒子群算法不同之处在于每个粒子的每一位只有两个状态，分别是 0 态和 1 态。粒子的每一位的状态由当前状态变换到下一个状态，只有 4 种可能情况，分别为从 0 态变换到 0 态、从 0 态变换到 1 态、从 1 态变换到 0 态、从 1 态变换到 1 态。粒子每一位状态的变化将根据所有粒子在该位共享的一个叫做信息素矩阵 P 里所存放的信息素来决定。举例说明：如第 i 个粒子的第 j 位此时状态是 0 态，历史最优值在第 j 位的状态为 0 态，全局最优粒子此时状态为 1 态，那么粒子由当前状态 0 态变为历史最优状态 0 态所产生的信息素是 v_{00}^j。同样，粒子由当前状态 0 态变为全局最优状态 1 态所产生的信息素为 v_{01}^j。信息素的多少决定着粒子下一时刻的状态，其基本原理如图 1 所示。

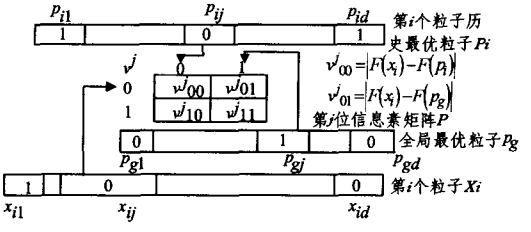

图 1 基本原理图

生物学家 Cole 发现整个蚁群行为是一个周期行为，然而单个蚂蚁的行为却是混沌行为[12]。本文利用单个蚂蚁的混沌行为构造了混沌二进制序列，用于初始化粒子群。具体操作如下：

根据门限函数：

$$\eta_l(x)=\begin{cases}0, & x<l\\ 1, & x>l\end{cases} \tag{5}$$

可由 Logisic 映射迭代 d 次，得到一个对应的 d 位长度的二进制混沌序列，其中 l 为常数。对于 Logisic 映射，l 取 0.5。同时，为了增强 DPSO 算法的全局搜索能力，在每次迭代中，粒子群以 k 的概率重新初始化，表示为：

$$P_i=R_i(P_i),\ \text{rand}()<k \tag{6}$$

式中，P_i 表示第 i 个粒子，$R_i(P_i)$表示对 P_i 重新初始化，$rand()$ 为[0,1]之间的随机数，k 为常数，本文取 $k=0.05$。

本文借鉴蚁群算法中信息素共享机制重新选择了粒子群速度更新公式

$$v(x_i(j),p_i(j),j)^{t+1}=\rho\times v(x_i(j),p_i(j),j)^t+(1-\rho)(F(p_i)-F(x_i)) \tag{7}$$

$$v(x_i(j),p_g(j),j)^{t+1}=\rho\times v(x_i(j),p_i(j),j)^t+(1-\rho)(F(p_g)-F(x_i)) \tag{8}$$

式中，$j=1,\cdots,d$ 是粒子的长度(维数)；i 表示第 i 个粒子，$i=1,\cdots,n$；x_i 表示当前的粒子位置，p_i 表示粒子的历史最优位置，p_g 表示粒子群体的最优位置；$x_i(j)$表示第 i 个粒子的第 j 位的值，$p_i(j)$表示第 i 个粒子的历史最优第 j 位的值，$p_g(j)$表示全局最优的第 j 位的值；F 是粒子的适应度函数，t 表示迭代次数；ρ 表示信息素的持久率，$1-\rho$ 表示信息素蒸发率，ρ 值越小，则信息素蒸发得越快，过去所保留的信息就会更快地被忘记，本文 ρ 取 0.8。v_j^{t+1}是在 $t+1$ 次迭代时第 j 个位置的速度信息素矩阵，采用一个粒子接一个粒子的速度信息素更新方式。

这一做法一方面使连续的粒子群算法离散化，另一方面使粒子根据自己的信息来判断飞行方向，还可以参考其他粒子留下来的信息，从而达到种群中所有粒子进行充分的信息交流。

本文采用文献[11]一样的方法对解进行修正，具体如下：

```
Procedure repair P(t)
Begin
  Knapsack-overfilled = false
  If ∑_{i=1}^{m} ω_i x_i^t > C then
    Knapsack-overfilled = true
  While (knapsack-overfilled) do
    Select the jth item from the knapsack randomly
    x_j^t = 0
  If ∑_{i=1}^{m} ω_i x_i^t ≤ C then
    Knapsack-overfilled = false
  End
  While (~knapsack-overfilled) do
    Select the jth item from the knapsack randomly
    x_k^t = 1
  If ∑_{i=1}^{m} ω_i x_i^t > C then
    Knapsack-overfilled = true
  End
    x_k^t = 0
End
```

2.2 算法流程

本文提出了基于蚁群混沌行为的离散粒子群算法，其具体实现步骤如下。

Step 1 初始化粒子的总数 n、最大迭代次数 k 和粒子的长度 d，初始化速度信息素矩阵，并且构造全体粒子。

Step 2 计算粒子的适应度，找出粒子的全局最优粒子和历史最优粒子，并对粒子的解进行修正。

Step 3 根据式(7)—式(8)，更新速度信息素矩阵。

Step 4 看是否满足停止条件(最大迭代次数)。若满足停止条件，则输出最优值；否则转到 Step2 继续运行，直到满足条件为止。

3 实验测试

下面给出 3 个 KP 问题实例，其具体表示如下。

实例 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]，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]，背包的最大容量 C=1000，规模为 d=50。

实例 2　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]，物品的价值集 P=[597,596,593,586,581,568,567,560,549,548,547,529,529,527,520,491,482,478,475,475,466,462,459,458,454,451,449,443,442,421,410,409,395,394,390,377,375,366,361,347,334,322,315,313,311,309,296,295,294,289,285,279,277,276,272,248,246,245,238,237,232,231,230,225,192,184,183,176,174,171,169,165,165,154,153,150,149,147,143,140,138,134,132,127,124,123,114,111,104,89,74,63,62,58,55,48,27,22,12,6]，背包的最大容量 C=6718，规模 d=100。

实例 3　物品的重量 $\omega_i \in [1,10]$ 随机选取，价值 $p_i=\omega_i+l_i$，其中 $l_i \in [0,5]$ 随机选取，背包的容量 $C=0.5\sum_{i=1}^{m}\omega_i$。考虑 3 个背包问题，它们具有不同的物品数量，分别取 100,250,500。

这些实例是被广为引用的著名的 KP 问题实例，用来测试算法的性能。

3.1　DPSO 算法与其他仿生算法的性能比较

本节利用实例 1 及实例 2 对 DPSO 算法性能进行测试，并将测试结果与其他仿生算法所得到的结果进行对比。

对实例 1、实例 2 重复计算 10 次，得到的最大价值与 GPSGA[9]，SREA[10] 的相比情况如表 1 所列。

表 1　3 种方法所求结果的比较

算法	GPSGA		SREA		DPSO	
实例	最优解代码	价值/容量	最优解代码	价值/容量	最优解代码	价值/容量
实例 1	1101010111101 1011011011111 1101000010100 11000001000	3103/1000	1101010111101 1011011011111 1101000010100 11000001000	3103/1000	1101001111101 0011011011111 1111000010110 1000001010	3112/1000
实例 2	—	—	1111111111111 1111111111111 1111111111111 1111111011111 1110100010110 1101111111000 1110111000000 000000001	26559/6717	1111111111111 1111111111111 1111111111111 1111111011111 1110100010110 1101111111000 1110111000000 000000001	26559/6717

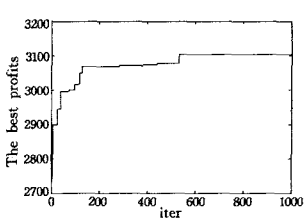

图 2　实例 1 最大价值随迭代的变化

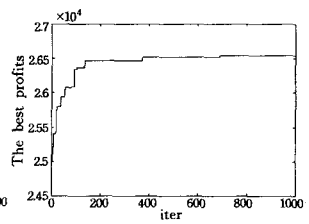

图 3　实例 2 最大价值随迭代的变化

由表 1 可知，相对于改进的遗传算法和进化算法，本文提出的离散粒子群算法也能成功地求解上述背包问题，甚至对实例 1 还能得到更好的解，充分说明了本方法的有效性。图 2 为取最大价值时最大价值随迭代次数的变化图，从中也可以看出本算法的有效性。当然从图 1 也发现求得最大价值时迭代次数有点偏长。

3.2　DPSO 算法与其他改进的离散粒子群算法的性能比较

本节利用实例 3 对 DPSO 算法性能进行测试，并将测试结果与基于其他方式改进的离散粒子群算法所得到的结果进行对比。本次实验统一取迭代次数 1000 次、粒子数 20。重复计算 10 次得到的最大价值与离散量子群算法[11]得到的结果比较如表 2 所列，重复计算 10 次得到的每次价值与取得最大价值时随迭代次数变化的情况如图 4、图 5、图 6 所示。

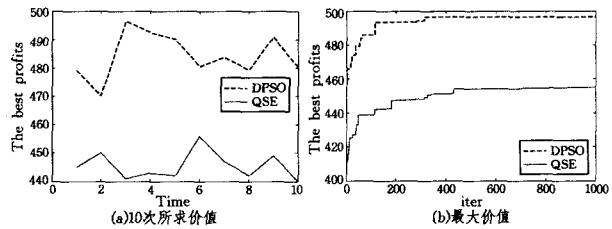

图 4　当物品为 100 个时两种离散粒子算法的比较

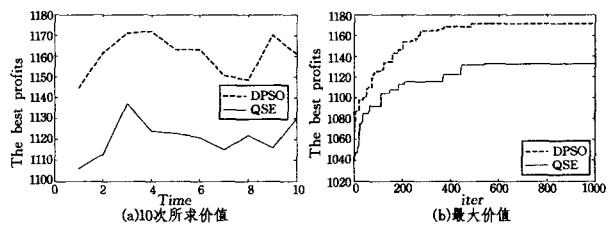

图 5　当物品为 250 个时两种离散粒子算法的比较

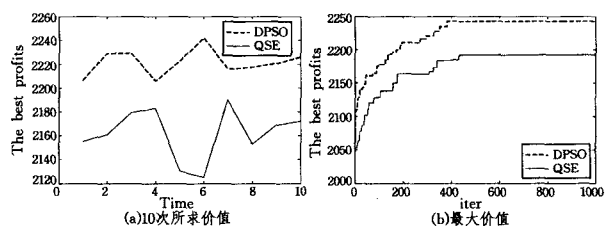

图 6　当物品为 500 个时两种离散粒子算法的比较

由表 2 及图 4、图 5、图 6 知，当物品数为 100 时，本文方法得到的最大价值为 496.8714，最小价值为 470.2210，最小价值也比 QSE 最大价值大；在物品数为 250 时，本文方法得到的最大价值为 1172.1，最小价值为 1144.4，也大于 QSE 最大价值 1137.2；在物品数为 500 时，本文方法得到的最大价值为 2241.9，最小价值为 2205.8，也大于 QSE 的最大价值 2190。可见，DPSO 算法性能较 QSE 方法好。

表 2　2 种离散粒子群算法所求结果的比较

物品数	100	250	500
方法	最大价值	最大价值	最大价值
DPSO	496.8714	1172.1	2241.9
QSE	455.96	1137.2	2190

3.3　DPSO 算法与传统方法的性能比较

对实例 3 重复计算 10 次得到的结果与传统的一些求解背包问题方法[11]相比，其结果如表 3 所列，其中 HILLKS 表

（下转第 286 页）

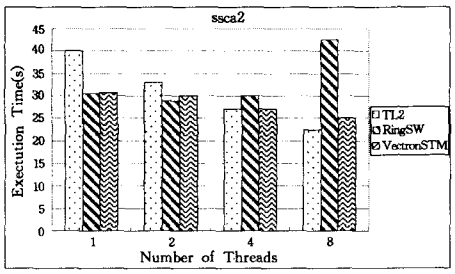

图 4　ssca2 使用 TL2，RingSW 和 VectorSTM 的执行结果

结束语　本文提出了一种采用向量时钟的不需要原子操作的 STM 实现算法 VectorSTM。VectorSTM 能够保证共享变量私有化的安全，而且实验数据表明 VectorSTM 有较好的可扩展性，在一些问题上还具有性能优势。但是 VectorSTM 性能受到误判率影响很严重，下一步我们将探索如何有效降低误判率，进一步提高 VectorSTM 的性能。

参 考 文 献

[1] Hammond L, Carlstrom B D, Wong V, et al. Programming with *transactional coherence and consistency*(TCC)[J]. ACM SIGPLAN Notices, 2004, 39(11): 1-13

[2] University of Rochester Department of Computer Science. Rochester Software Transactional Memory[EB/OL]. http://www.cs.rochester.edu/research/synchronization/rstm/

[3] Dice D, ShalevN O, Shavit. Transactional Locking II[C]// Proc. of the 20th Intl. Symp. on Distributed Computing. Berlin: Springer, 2006: 194-208

[4] Rajwar R, Lai M H K. Virtualizing Transactional Memory[C]// Proc. of the 32nd Annual Intl. Symp. on Computer Architecture (ISCA). Los Alamitos, CA: IEEE, 2005: 494-505

[5] Spear M F, MichaelC M M, von Praun. RingSTM: scalable transactions with a single atomic instruction[C]// 20th ACM Symposium on Parallelism in Algorithms and Architectures. New York: ACM, 2008: 275-284

[6] Blustein J, El-Maazawi A. Bloom Filters-A Tutorial, Analysis, and Survey[R]. Halifax, NS: Dalhousie University, 2002: 1-31

[7] Spear M F, Marathe V J, Dalessandro L, et al. Privatization Techniques for Software Transactional Memory[R]. TR 915. Dept. of Computer Science, Univ. of Rochester, Feb. 2007

[8] Guerraoui R, Herlihy M, Pochon B. Polymorphic contention management[C]// 19th International Symposium on Distributed Computing. 2005

[9] Minh C C, Chung J, Kozyrakis C, et al. STAMP: Stanford transactional applications for multi-processing[C]// IEEE International Symposium on Workload Characterization. Los Alamitos, CA: IEEE, 2008: 35-46

(上接第 180 页)

示爬上法，SAKS 表示模拟退火法，TABUKS 表示禁忌搜索法。

由表 3 可知，本文提出的离散粒子群算法对于较大规模的背包问题其搜索效果要比传统的一些方法好得多。

表 3　DPSO 与传统方法所求结果的比较

物品数	100	250	500
方法	最大价值	最大价值	最大价值
DPSO	496.8714	1172.1	2241.9
HILLKS	412.74	1035.9	2032.3
SAKS	415.27	1045.8	2048.8
TABUKS	429.47	1111.8	2120.1

结束语　本文借鉴蚁群算法的信息素机制对粒子进行更新，应用蚂蚁的混沌行为来初始化粒子群，并且在一定的条件下重新初始化，得到一种基于蚂蚁混沌行为的离散粒子群算法；将其应用到背包问题中，并将结果与改进的遗传算法、改进的进化算法、改进的粒子群算法及传统一些求解背包问题算法相比较。结果显示，本方法能得到较好的实验结果。

参 考 文 献

[1] Kennedy J, Eberhart R C. Particle Swarm Optimization[C]// Proceedings of the IEEE International Conference on Neural Networks. Perth, Aust: IEEE Piscataway, 1995: 1942-1948

[2] Kennedy J, Eberhart R C. A Discrete Binary Version of Particle Swarm Algorithm[C]// Proceedings of the 1997 Conference on System, Man, and Cybernetics. Piscataway: IEEE Press, 1997: 4104-4108

[3] Shi Y, Eberhart R C. Fuzzy Adaptive Particle Swarm Optimization[C]// Proceedings of the IEEE Conference on Evolutionary Computation, Soul: IEEE, 2001: 101-106

[4] Noel M M, Jannett T C. Simulation of a new hybrid particle swarm optimization algorithm[C]// Proc. of the 36th Southeastern Symposium on System Theory. Atlanta: IEEE, 2004: 150-153

[5] 贾东立，张家树. 基于混沌变异的小生境粒子群算法[J]. 控制与决策，2007，22(1)：117-120

[6] 钟一文，杨建刚，宁正元. 求解 TSP 问题的离散粒子群算法[J]. 系统工程理论与实践，2006，6：88-94

[7] Kashan A H, Karimi B. A discrete particle swarm optimization algorithm for scheduling parallel machines [J]. Computers & Industrial Engineering (2008), doi: 10.1016/j.cie.2008.05.007

[8] Garey M R, Johnson D S. Computers and intractability: A guide to the theory of NP-completeness[M]. San Francisco: Freeman W. H and Co, 1979

[9] 张铃，张钹. 佳点集遗传算法[J]. 计算机学报，2001，24(9)：917-922

[10] Li K S, Jia Y Z, Zhang W S, et al. A new method for solving 0/1 knapsack problem based on evolutionary algorithm with schema replaced[C]// Proceedings of the IEEE International Conference on Automation and Logistics. Qingdao, 2008: 2569-2571

[11] Wang Y, Feng X Y, Huang Y X, et al. A novel quantum swarm evolutionary algorithm and its applications[J]. Neurocoputing, 2007, 70: 633-640

[12] Cole B J. Is animal behavior chaotic? Evidence from the activity of ants[J]. Proceedings of The Royal Society of London Series B-Biological Sciences, 1991, 244: 253-259