学术论文

背包问题的混合粒子群优化算法

高　尚[1,2]，杨静宇[3]
（1. 江苏科技大学电子信息学院，江苏 镇江　212003；2. 苏州大学江苏省计算机信息处理技术重点实验室，江苏 苏州　215006；3. 南京理工大学计算机科学与技术系，南京　210094）

[摘要]　经典的粒子群是一个有效的寻找连续函数极值的方法，结合遗传算法的思想提出的混合粒子群算法来解决背包问题，经过比较测试，6 种混合粒子群算法的效果都比较好，特别交叉策略 A 和变异策略 C 的混合粒子群算法是最好的且简单有效的算法，并成功地运用在投资问题中。对于目前还没有好的解法的组合优化问题，很容易地修改此算法就可解决。

[关键词]　粒子群算法；背包问题；遗传算法；变异
[中图分类号]　TP301.6　　[文献标识码]　A　　[文章编号]　1009－1742（2006）11－0094－05

1　引言

背包问题是运筹学中一个典型的组合优化难题，有着广泛的应用背景，如货物装载问题，选址问题等。由于此问题比较简单典型，因此评价算法优劣常常以此问题作为的测试对象进行研究。背包问题属于 NP 问题，目前求解的方法有精确方法（如动态规划、递归法、回溯法、分支限界法等[1]），近似算法（如贪心法[1]，Lagrange 法等）以及智能优化算法（如模拟退火算法[2]、遗传算法[2]、遗传退火进化算法[3]、蚁群算法[4,5]）。精确方法虽然可以得到准确解，但时间复杂性与物品数目成指数关系。近似算法和智能优化算法虽然不一定得到准确解，但可得到比较有效解，并且时间复杂性比较低。笔者尝试采用粒子群优化算法解决此问题。

2　背包问题数学模型

背包问题的描述有许多种形式，最典型的是 0－1 背包问题。0－1 背包问题是指：给定 n 种物品和一个背包，物品 i 的质量是 c_i，其价值为 p_i，背包的质量容量为 C，如何选择物品，使得装入背包中物品的总价值最大。其数学模型为

$$\begin{aligned} &\max \sum_{i=1}^{n} p_i x_i , \\ &\text{s.t.} \sum_{i=1}^{n} c_i x_i \leqslant C \\ &x_i = 0,1(i = 0,1,\cdots,n) \end{aligned} \tag{1}$$

3　基本粒子群优化算法

粒子群优化（PSO，particle swarm optimization）算法是一种进化计算技术，最早是由 Kennedy 与 Eberhart 于 1995 年提出的[6]。源于对鸟群捕食的行为研究的 PSO 同遗传算法类似，是一种基于迭代的优化工具。系统初始化为一组随机解，通过迭代搜寻最优值。目前已广泛应用于函数优化，神经网络训练，模糊系统控制以及其他遗传算法的应用领域。目前已提出了多种 PSO 改进算法[6~9]，如自适应 PSO 算法、杂交 PSO 算法、协同 PSO 算法。笔者提出基于遗传算法思想的一种新的 PSO 算法来解决背包问题。

PSO 是模拟鸟群的捕食行为，设想这样一个场

[收稿日期]　2005－06－14；修回日期　2005－07－19
[作者简介]　高　尚（1972－），男，江苏镇江市人，博士，江苏科技大学副教授，主要从事人工智能等方面的研究

景：一群鸟在随机搜索食物。在这个区域里只有一块食物。所有的鸟都不知道食物在那里。但是他们知道当前的位置离食物还有多远，那么找到食物的最优策略是什么呢？最简单有效的就是搜寻目前离食物最近的鸟的周围区域。PSO 从这种模型中得到启示并用于解决优化问题。PSO 中，每个优化问题的解都是搜索空间中的一只鸟，称为“粒子”。所有的例子都有一个由被优化的函数决定的适应值，每个粒子还有一个速度决定他们飞翔的方向和距离，然后粒子们就追随当前的最优粒子在解空间中搜索。PSO 初始化为一群随机粒子（随机解），然后通过迭代找到最优解。在每一次迭代中，粒子通过跟踪两个极值来更新自己。一个是粒子本身所找到的最优解，称为个体极值 pbest；另一个极值是整个种群目前找到的最优解，称为全局极值 gbest，另外也可以不用整个种群而只是用其中一部分为粒子的邻居，那么在所有邻居中的极值就是局部极值。

在找到这 2 个最优值时，每个粒子根据如下的公式来更新自己的速度和新的位置：

$$\boldsymbol{v}_{k+1} = c_0\boldsymbol{v}_k + c_1(\mathbf{pbest}_k - \boldsymbol{x}_k) + c_2(\mathbf{gbest}_k - \boldsymbol{x}_k) \tag{2}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{v}_{k+1} \tag{3}$$

其中 $\boldsymbol{v}_k$ 是粒子的速度向量；$\boldsymbol{x}_k$ 是当前粒子的位置；$\mathbf{pbest}_k$ 粒子本身所找到的最优解的位置；$\mathbf{gbest}_k$ 整个种群目前找到的最优解的位置；c_0，c_1，c_2 表示群体认知系数，c_0 一般取介于（0，1）之间的随机数，c_1，c_2 取（0，2）之间的随机数。$\boldsymbol{v}_{k+1}$是 $\boldsymbol{v}_k$，$\mathbf{pbest}_k - \boldsymbol{x}_k$ 和 $\mathbf{gbest}_k - \boldsymbol{x}_k$ 矢量的和。在每一维粒子的速度都会被限制在一个最大速度 $\boldsymbol{v}_{\max}$（$\boldsymbol{v}_{\max} > 0$），如果某一维更新后的速度超过用户设定的 $\boldsymbol{v}_{\max}$，那么这一维的速度就被限定为 $\boldsymbol{v}_{\max}$，即若 $\boldsymbol{v}_k > \boldsymbol{v}_{\max}$ 时，$\boldsymbol{v}_k = \boldsymbol{v}_{\max}$ 或 $\boldsymbol{v}_k < -\boldsymbol{v}_{\max}$ 时，$\boldsymbol{v}_k = -\boldsymbol{v}_{\max}$。

4　解 0－1 背包问题的混合粒子群算法

首先把原约束方程作为罚函数项加入到原目标中，变成无约束的优化问题，即

$$\min f = -\sum_{i=1}^{n} p_i x_i + M\left\{\min\left\{0, \left[C - \sum_{i=1}^{n} c_i x_i\right]\right\}\right\}^2 \tag{4}$$

其中 M 为一充分大的正数。

背包问题的解用向量 $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 表示，粒子群算法的本质是利用个体极值信息和全局极值两个信息，来指导粒子下一步迭代位置。对于 0－1 背包问题，若按基本粒子群算法，其速度难于表达，故采用遗传算法[2]交叉操作的思想：式（1）中的 $c_0\boldsymbol{v}_k$ 项可以看作遗传算法的变异操作，式（1）中的 $c_1(\mathbf{pbest}_k - \boldsymbol{x}_k) + c_2(\boldsymbol{gbest}_k - \boldsymbol{x}_k)$ 项可以看作遗传算法的交叉操作，让当前解与个体极值和全局极值分别作交叉操作，产生的解为新的位置。交叉方法可以采用以下两种方法：

1）交叉策略 A

a. 两串 old1 和 old2 交叉，在第二个串 old2 中随机选择一个交叉区域；

b. 将 old1 的相应的交叉区域由 old2 交叉区域代替。

例如两父串为

old1 = 1 0 0 1 0 1 1 1 1 0 0 1，

old2 = 0 1 1 0 1 0 1 0 1 1 0 0。

交叉区域为 0 1 0 1 1，交叉后为

new1 = 1 0 0 1 0 0 1 0 1 1 0 1。

2）交叉策略 B

a. 随机产生 k 个 1 到 n 的整数 j_1，j_2，…，j_k；

b. 将 old1 的 j_1，j_2，…，j_k 的位置数值由 old2 相应的部分代替。

具体变异操作可以采用下面三种：

1）变异策略 A

a. 在解空间 $(x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 中随机选择一块区域，如 $(x_i, x_{i+1}, \cdots, x_j)^{\mathrm{T}}$；

b. $(x_i, x_{i+1}, \cdots, x_j)^{\mathrm{T}} \leftarrow (\bar{x}_i, \bar{x}_{i+1}, \cdots, \bar{x}_j)^{\mathrm{T}}$//取反运算。

如原来解为 1 0 0 1 0 1 1 1 1 0 0 1，随机产生区域为 1 1 0 0，则变异操作后的解为

1 0 0 1 0 1 1 0 0 1 1 1。

2）变异策略 B

a. 在解空间 $(x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 中随机选择一块区域，如 $(x_i, x_{i+1}, \cdots, x_j)^{\mathrm{T}}$；

b. $(x_i, x_{i+1}, \cdots, x_j)^{\mathrm{T}}$ 取随机值。

3）变异策略 C

a. 在解空间 $(x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 中随机选择一个 1 到 n 的整数 j；

b. $x_j \leftarrow \bar{x}_j$。

如原来解为1001011111001，随机产生整数为4，则变异操作后的解为1000011111001。

解0-1背包问题的混合粒子群算法hybrid_PSO如下：

1）设定粒子数 n_p，规定迭代次数 n_{max}，随机产生 n_p 个初始解 X_0；

2）根据当前位置根据式（4）计算适应值 l_0，设置当前适应值为个体极值 **plbest**，当前位置为个体极值位置 **pxbest**，根据各个粒子的个体极值 **plbest**，找出全局极值 **glbest** 和全局极值位置 **gxbest**；

3）While（迭代次数 < 规定迭代次数 n_{max}）do

4）for $j = 1 : n_p$

5）第 j 个粒子位置 $X_0(j)$ 与 **gxbest** 交叉得到 $X'_1(j)$；

6）$X'_1(j)$ 与 **pxbest** 交叉得到 $X_1(j)$；

7）对 $X_1(j)$ 进行变异操作；

8）根据当前位置计算适应值 l_1；

9）如果 $l_1(j) <$ **plbest**(j)，则 **pxbest**$(j) = X_1(j)$，**plbest**$(j) = l_1(j)$；

10）End

11）根据各个粒子的个体极值 **plbest**，找出全局极值 **glbest** 和全局极值位置 **gxbest**；

12）$X_0 \leftarrow X_1$；

13）End

14）输出全局极值 **glbest** 和全局极值位置 **gxbest**。

该粒子群算法的时间复杂性估算如下：以交叉时间和变异操作花费最多，变异操作的时间与交叉操作时间相近，里面循环需要作 $O(3n_p)$ 交叉（或变异）操作，外循环执行 n_{max} 次，所以时间复杂性大约为 $O(3n_p n_{max})$。

5 数值仿真与分析

5.1 各种策略比较

采用文献[4]的一个典型的背包问题数据，$n = 10$，$C = 269$ g，$\{p_1, p_2, \cdots, p_{10}\} = \{55, 10, 47, 5, 4, 50, 8, 61, 85, 87\}$ 元，$\{c_1, c_2, \cdots, c_{10}\} = \{95, 4, 60, 32, 23, 72, 80, 62, 65, 46\}$ g。混合粒子群算法分别2种交叉策略和3种变异策略，组合起来6种方法进行比较，各种算法各测试100次，最优目标值为295元，最优解为：$X^* = \{0, 1, 1, 1, 0, 0, 0, 1, 1, 1\}$。参数粒子数 $n_p = 10$，最大迭代次数 $n_{max} = 10$ 的结果如表1所示；若粒子数 $n_p = 20$，最大迭代次数 $n_{max} = 30$。蚁群算法参数如下：蚂蚁数20，信息素强度重要性 $\alpha = 1.5$，吸引度重要性 $\beta = 2$，遗留程度 $\rho = 0.9$，迭代次数为100。计算结果如表2所示。

表1 各种策略结果比较（$n_p = 10$，$n_{max} = 10$）

Table 1 Comparison results of strategies with $n_p = 10$ and $n_{max} = 10$

算法		平均值/元	最好解/元	最差解/元	最好解的次数
交叉策略A		261.29	295	201	6
交叉策略B		265.92	295	205	4
交叉策略A	变异策略A	280.91	295	237	15
	变异策略B	280.49	295	231	13
	变异策略C	285.40	295	246	28
交叉策略B	变异策略A	279.18	295	222	15
	变异策略B	282.87	295	232	14
	变异策略C	283.62	295	210	18

表2 各种策略结果比较（$n_p = 20$，$n_{max} = 30$）

Table 2 Comparison results of strategies with $n_p = 20$ and $n_{max} = 30$

算法		平均值/元	最好解/元	最差解/元	最好解的次数
蚁群算法		287	295	277	64
交叉策略A		275.62	295	265	12
交叉策略B		270.78	295	245	8
交叉策略A	变异策略A	275.52	295	245	19
	变异策略B	294.80	295	287	90
	变异策略C	294.98	295	294	98
交叉策略B	变异策略A	293.00	295	279	44
	变异策略B	293.1	295	255	47
	变异策略C	293.27	295	284	42

从表1和表2可以看出，只进行交叉操作的2个算法，效果比较差。效果最好的是交叉策略A和变异策略C的组合算法最好。并且交叉策略A比交叉策略B简单，变异策略C也比变异策略A和变异策略B简单。变异策略A和变异策略B，由于变异的位数多，增加了部分差的解，因此效果不如变异方法C。随着粒子数和最大迭代次数增加，计算效果变好，但其运算量也将增大。另外这里的

交叉操作与遗传算法的交叉操作不同，遗传算法随机选择 2 个解进行交叉，而粒子群交叉操作是对每个粒子进行交叉操作，并且与个体极值和全局极值进行交叉，考虑了优生的思想，因此效果比传统的遗传效果好。

5.2　与其他算法比较

针对背包问题，对递归算法、回溯方法和蚁群算法进行了比较，采用文献[5]的测试方法，数据随机产生，各 c_i 和 p_i 在 1～100 随机生成，物品种类取值为 5～20，背包的质量容量 C 为总质量的 80%。每个算法运行 100 次，统计出平均运行时间，结果如表 3 所示。从表 3 可以看出交叉策略 A 和变异策略 C 的粒子群算法运行的效率较高，尽管比文献[5]蚁群算法运行稍长点，但得到精确解的次数较高（见表 2）。

表 3　各种算法的平均运行时间比较

Table 3　Comparison of average run time of algorithms

n	5～9	10	11	12	13	14	15	16	17	18	19	20
回溯方法[5]/s	<0.001	0.240	2	3	7	16	33	42	88	190	393	620
蚁群算法[5]/ms	<1	<1	<1	<1	<1	<1	<1	<1	40	50	56	66
递归算法/ms	<60	60	70	110	130	150	180	230	300	460	640	1 060
交叉策略 A 和变异策略 C 的粒子群算法/ms	<50	50	65	78	84	96	100	120	130	170	180	200

5.3　应用实例

求解背包问题有着广泛的应用前景，在理论上，可以解决整数规划问题；在实践中，资源分配、投资问题、货物装运、预算控制、项目选择等实际问题均可归结为背包问题，其求解效率将直接影响社会生产力。下面举一应用在投资问题的实例，投资问题是指某企业将投资 n 种项目，项目 i 需投资 c_i 元，可得收益 p_i 元，总投资为 C 元，如何投资项目最合理？其数学模型就是式（1），实质就是背包问题。具体数据如下 $n=7$，$\{c_1, c_2, \cdots, c_7\} = \{3, 4, 3, 3, 15, 13, 16\}$ 元，$\{p_1, p_2, \cdots, p_7\} = \{12, 12, 9, 15, 90, 26, 112\}$ 元，$C=35$ 元。分别用蚁群算法、遗传退火算法和交叉策略 A 和变异策略 C 的粒子群算法运行，每个算法运行 100 次，结果如表 4 所示。粒子数 $n_p=20$，最大迭代次数 $n_{max}=30$。表 4 可见，交叉策略 A 和变异策略 C 的粒子群算法运行的效率较高。

表 4　各种算法结果比较

Table 4　Comparison results of algorithm

算法	平均值/元	最好解/元	最差解/元	最好解的次数
蚁群算法	210.42	217	202	84
遗传退火进化算法[3]	215.65	217	211	87
交叉策略 A 和变异策略 C 的粒子群算法	216.97	217	214	99

6　结论

提出的粒子群优化算法不仅可以解决背包问题，对于整数规划问题，对该算法作适当修改，也可适用。粒子群优化算法（PSO）是起源对简单社会系统的模拟，擅长连续问题的优化，笔者尝试采用粒子群算法解决离散优化问题。PSO 研究处于初期，还有许多问题值得研究，如算法的收敛性、理论依据等。但从当前的粒子群算法的应用效果[8,9]来看，这种模仿自然生物的新型系统的寻优思想无疑具有十分光明的前景，还有待于进一步展开。

参考文献

[1] 王晓东. 计算机算法设计与分析[M]. 北京：电子工业出版社，2001. 92～168

[2] 王　凌. 智能优化算法及其应用[M]. 北京：清华大学出版社，2001. 17～59

[3] 金慧敏，马　良. 遗传退火进化算法在背包问题中的应用[J]. 上海理工大学学报，2004，26(6)：561～564

[4] 马　良，王龙德. 背包问题的蚂蚁优化算法[J]. 计算机应用，2001，21(8)：4～5

[5] 于永新，张新荣. 基于蚁群系统的多选择背包问题优化算法[J]. 计算机工程，2003，29(20). 75～76，84

[6] Eberhart R C，Kennedy J. A new optimizer using particles swarm theory[A]. Proc Sixth International Symposium on

Micro Machine and Human Science[C]. Nagoya, Japan, 1995. 30 ~ 43

[7] Shi Y H, Eberhart R C. A modified particle swarm optimizer [A]. IEEE International Conference on Evolutionary Computation[C]. Anchorage, Alaska, 1998. 69 ~ 73

[8] 李爱国，覃 征，鲍复民，等. 粒子群优化算法[J]. 计算机工程与应用，2002，38(21)：1 ~ 3

[9] 杨 维，李歧强. 粒子群优化算法综述[J]. 中国工程科学，2004，6(5)：87 ~ 94

Solving Knapsack Problem by Hybrid Particle Swarm Optimization Algorithm

Gao Shang[1,2], Yang Jingyu[3]

(1. *School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu* 212003, *China*; 2. *Provincial Key Laboratory of Computer Information Processing Technology, Suzhou, Jiangsu* 215006, *China*; 3. *Department of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing* 210094, *China*)

[**Abstract**] The classical particle swarm optimization is a powerful method to find the minimum of a numerical function, on a continuous definition domain. The particle swarm optimization algorithm combining with the idea of the genetic algorithm is recommended to solve knapsack problem. All the 6 hybrid particle swarm optimization algorithms are proved effective. Especially the hybrid particle swarm optimization algorithm derived from across strategy A and mutation strategy C is a simple yet effective algorithm and it has been applied successfully to investment problem. It can easily be modified for any combinatorial problem for which there has been no good specialized algorithm.

[**Key words**] particle swarm algorithm; knapsack problem; genetic algorithm; mutation