第 27 卷　第 1 期　　华侨大学学报(自然科学版)　　Vol.27　No.1
2006 年 1 月　　Journal of Huaqiao University (Natural Science)　　Jan. 2006

文章编号　1000-5013(2006)01-0016-04

求解 0-1 背包问题的混合遗传算法

宋海洲　　魏旭真

(华侨大学数学系，福建 泉州 362021)

摘要　对于 0-1 背包问题设计一种价值密度，并在此基础上提出求解 0-1 背包问题的混合遗传算法. 经大量数值实验比较该方法与传统方法及简单遗传算法，结果表明算法能有效求解 0-1 背包问题.

关键词　背包问题，不可行解，贪心法，遗传算法

中图分类号　TP 301.6　　**文献标识码**　A

假设我们要从多种物品(一般称为项目)中选择几种物品，装满背包. 若有 n 个不同的项目，对于项目 j，其重量为 w_j，价值为 p_j，b 是背包承受的最大重量. 背包问题就是要在不超过背包承重的前提下，使装入背包价值最大. 设 x_i 为 0-1 变量，若项目 i 被选入，则 $x_i = 0$. 背包问题的数学模型[1]为

$$\max \sum_{i=1}^{n} x_i p_i, \tag{1}$$

受约束于 $\sum_{i=1}^{n} w_i x_i \le b$, $x_i = 0$ 或 1，$i = 1, 2, \cdots, n$. 目前，求解背包问题的主要方法有递归回溯法、贪心法和遗值算法[2~5]. 由于背包问题是 NP 问题(没有多项式时间复杂度时间算法的问题)，因此在项目 n 很大时使用递归回溯法求解并不现实，而利用贪心法求解有时效果也不理想. 在简单遗传算法求解背包问题中，人们通常利用基于惩罚法来解决不可行性和背包承重利用不足可行解的处理. 但是，随着求解规模(n)的增大，利用惩罚法效果并不理想. 针对上述不足，本文提出一种基于贪心法的混合遗传算法(HGA). 大量计算试验表明，该算法能够较好地处理不可行性和背包承重利用不足的可行解问题，从而提高求解的速度和精度.

1　基于贪心法的混合遗传算法

1.1　编码方式

采用二进制编码，n 个项目的背包问题的解可以表示为二进制串 $x = (x_1, \cdots, x_i, \cdots, x_n)$，$x_i = 0$ 或 1. $x_i = 1$ 表示第 i 个项目被选入背包；$x_i = 0$ 表示第 i 个项目不被选入背包. 例如，10 个项目的背包问题中，(0,1,0,1,0,0,0,0,1,0) 表示项目 2，4 和 9 被选入背包，而其他项目则不选入背包的解.

1.2　适应值函数

适应值函数 $f(x) = \sum_{i=1}^{n} x_i p_i$.

1.3　可行解的处理

初始种群和遗传操作(交叉和变异算子)产生的解，可以分为可行解和不可行解两类. 对于可行解，存在着背包承重利用不足的问题. 人们一般是通过惩罚函数对背包承重利用不足的可行解进行惩罚，以使个体朝充分利用背包的承重方向进化，从而达到加快遗传算法收敛速度的作用. 本文受贪心法思想的启发，设计了一种背包承重利用不足的可行解处理新方法，即设 $x = (x_1, \cdots, x_i, \cdots, x_n)$ 是一个背包承重利用不足的可行解. 我们采用 2 个步骤将 x 修正为充分利用背包承重的可行解 x'. (1) 对 x 中还没有背

收稿日期　2005-06-24

作者简介　宋海洲(1971-)，男，副教授，主要从事数学模型的研究. E-mail: hzsong@hqu.edu.cn

基金项目　福建省自然科学基金资助项目(20511028)

的项目,按价值-重量比降序排列. (2) 按上述排列的先后次序依次加入项目,直到背包不能装为止. 这样就形成了充分利用背包承重的可行解 x'.

1.4　不可行解的处理

不可行解的产生主要有以下 2 种情况. (1) 初始种群构造时产生不可行解的染色体编码结构. (2) 遗传操作(如交叉和变异算子)导致不可行解的染色体编码结构. 针对这些不可行解,可以采用 2 个步骤进行修复. (1) 对 x 中已经背的项目,按价值-重量比升序排列. (2) 按上述排列的先后次序依次减掉项目(即令相应项目对应的 x_i 为 0),直到形成可行解为止.

1.5　初始种群的产生

先用完全随机的方法产生初始群体 $p=(p_1, p_2, \cdots, p_{\mathrm{popsize}})$,popsize 为种群规模. 然后,用上述修正程序将 p 中的可行解修正完善;将 p 中的不可行解修复,得到初始种群 $p'=(p'_1, p'_2, \cdots, p'_{\mathrm{popsize}})$. 最后用贪心法产生一个近似最优解,将 p 中的适应度最差的个体替换掉,得到初始种群. 这样产生的初始种群,既无不可行解,又具有较高的质量.

1.6　遗传操作

(1) 选择. 采用轮盘赌选择方法. 先计算出群体中各个个体所对应的适应度总和 $\sum_{i=1}^{n} f_i$,再计算出每一个个体的相对适应度大小 $f_i / \sum_{i=1}^{n} f_i$. 它即为每一个个体遗传到下一代群体中的概率,每一个概率的值组成一个区域,全部的概率和为 1. 最后再产生一个 0 到 1 之间的随机数,依据该随机数出现在上述的哪个概率区域来确定各个个体被选中的次数. 采用精英法则,强行将上一代的最优个体直接进入下一代. 这样,每进化一代,下一代的最优个体一定不劣于上一代. (2) 交叉. 交叉采用"优-优交叉"和"差-差交叉"策略. 采用均匀交叉操作,设 S_1, S_2 是两个待交叉的父个体,均匀交叉操作方法有 3 个步骤. (a) 随机产生与父个体等长的两个 0-1 掩码(掩码中的片段表明了哪个父个体向新个体提供变量值),利用掩码形成两个新个体. (b) 用上面的修正(或修复)程序将新个体修正(或修复). (c) 将两个修正(或修复)后的新个体与两个父个体放在一起选择,选择两个适应值较高的个体作为子代. (3) 变异. 变异采用位变异和均匀变异操作,均匀变异率一般取 0.001～0.010,可以随着项目的个数增加而增加. 均匀变异操作有 2 个步骤. (1) 随机的产生与父个体等长的一个 0-1 掩码(掩码中的片段表明了父个体的位是否变异),利用掩码形成一个新个体. (2) 用上面的修正(或修复)程序将新个体修正(或修复).

1.9　算法流程

混合遗传算法流程图,如图 1 所示.

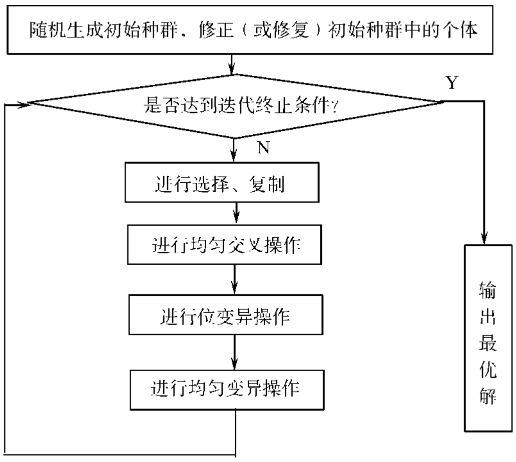

图 1　混合遗传算法(HGA)流程图

2　仿真实验

(1) 对小规模(20 个项目)的背包问题,进行递归回溯法、贪心法和混合遗传算法(HGA)实验. 其中,HGA 的参数设置为:均匀交叉率 0.9,位变异率 0.1,均匀变异率 0.01,终止迭代次数 100 代. 20 个项目的背包问题为

$$\max \sum_{i=1}^{20} x_i p_i, \tag{2}$$

受约束于 $\sum_{i=1}^{20} w_i x_i \leq b = r \sum_{i=1}^{20} w_i$, $x_i = 0$ 或 1, $i = 1, 2, \cdots, 20$. 其中, r 分别取 0.01, 0.05, 0.10, 0.20, …, 0.90, 0.95 和 0.99; w, p 为随机整数取 1～100 的 20 维向量分量. 对于同一个背包问题,贪心法和 HGA 各做 10 轮实验,结果如表 1 所示($r = b / \sum_{i=1}^{n} w_i$,下同). 从实验结果可以看出,对规模比较小(20 个项目)的背包问题,贪心法已经有不错的效果,但是 HGA 的效果更好. HGA 不仅收敛率相当高(收敛率在 98 %以上),而且收敛速度也是相当快的,平均遗传代数为 16.65 代就可得到最优解. (2) 对规模为 n

(n个项目)的背包问题,进行递归回溯法、贪心法和混合遗传算法(HGA)实验. 其中HGA的参数设置

表1　3种算法比较

n	r	递归回溯法得到的最优值	贪心法得到的近似最优值的平均值	HGA 得到的近似最优值的平均值	HGA 运行 10 轮中收敛到最优解的轮数	HGA 平均遗传代数
20	0.01	71.9	71.9	71.9	10	1.80
20	0.10	338.7	330.0	338.7	10	11.30
20	0.30	566.2	563.5	566.1	9	23.80
20	0.50	810.1	800.4	810.1	10	41.50
20	0.70	894.7	885.9	894.3	9	20.30
20	0.90	1 035.5	1 032.2	1 035.5	10	15.40
20	0.99	1 062.1	1 062.0	1 062.1	10	3.00
平　均					9.8	16.65

为:均匀交叉率 0.9,位变异率 0.1,均匀变异率 0.01,终止迭代次数 100 代. n 个项目的背包问题为

$$\max \sum_{i=1}^{n} x_i p_i , \tag{3}$$

受约束于 $\sum_{i=1}^{n} w_i x_i \leq b = 0.08 \sum_{i=1}^{n} w_i$, $x_i = 0$ 或 1, $i = 1, 2, \cdots, 20$. n 分别取 20,30,⋯,90 和 100;w,p 为随机整数取 1～100 的 n 维向量分量. 对同一个背包问题,贪心法和 HGA 各做 10 轮实验,结果如表 2 所示. 从实验结果可以看出,对规模 n 比较大的背包问题,递归回溯法未能得到结果(采用 Matlab 编程,运行速度稍慢),而贪心法和 HGA 都能得到结果. 但是 HGA 的精度都要比贪心法的精度高,而且 HGA 的收敛速度是相当快的,平均遗传代数为15.70代就得到近似最优解. (3) 对规模为 n(n个项目)的背包

表2　递归回溯法和贪心法与 HGA 实验结果比较

n	r	递归回溯法得到的最优值	贪心法得到的近似最优值的平均值	HGA 得到的近似最优值的平均值	HGA 的平均遗传代数
20	0.08	267.6	259.2	267.6	23.90
40	0.08	未得到结果	600.3	610.4	15.00
60	0.08	未得到结果	998.7	1002.6	11.10
80	0.08	未得到结果	1 319.3	1 321.3	15.10
100	0.08	未得到结果	1 590.7	1 598.1	21.90
平　均					15.70

问题,进行简单遗传算法(SGA)和混合遗传算法(HGA)比较实验,结果如表 3 所示. 每组数据运行相同的问题10轮,其中SGA和HGA两个算法的参数设置为:均匀交叉率0.9,位变异率0.1,均匀变异率

表3　SGA 和 HGA 实验比较结果

n	20 [3]	30	40	50
r	0.8	0.8	0.8	0.8
递归回溯法	1 024.0	未得到结果	未得到结果	未得到结果
贪心法	1 018.0	1 317.0	2 092.0	2 575.0
HGA 最好结果	1 024.0	1 322.0	2 096.0	2 585.0
HGA 平均结果	1 024.0	1 322.0	2 095.6	2 585.0
HGA 最坏结果	1 024.0	1 322.0	2 092.0	2 585.0
HGA 最大遗传次数	57.0	16.0	27.0	24.0
HGA 平均遗传次数	19.0	10.2	17.0	14.2
HGA 最小遗传次数	10	7	10	8
SGA 最好结果	1 024.0	1 322.0	2 096.0	2 585.0
SGA 平均结果	1 024.0	1 318.0	2 093.2	2 581.7
SGA 最坏结果	1 024.0	1 317.0	2 092.0	2 578.0
SGA 最大遗传次数	100	100	100	100
SGA 平均遗传次数	100	100	100	100
SGA 最小遗传次数	100	100	100	100

续表

n	60	70	80	90	100
r	0.8	0.8	0.8	0.8	0.8
递归回溯法	未得到结果	未得到结果	未得到结果	未得到结果	未得到结果
贪心法	3 227.0	3 116.0	3 932.0	4 485.0	5 186.0
HGA 最好结果	3 233.0	3 120.0	3 944.0	4 493.0	5 194.0
HGA 平均结果	3 232.5	3 120.0	3 944.0	4 493.0	5 193.5
HGA 最坏结果	3 232.0	3 120.0	3 944.0	4 493.0	5 189.0
HGA 最大遗传次数	59.0	23.0	20.0	45.0	41.0
HGA 平均遗传次数	29.6	16.0	16.3	19.1	21.2
HGA 最小遗传次数	13	9	11	12	10
SGA 最好结果	3 233.0	3 119.0	3 944.0	4 491.0	5 186.0
SGA 平均结果	3 231.4	3 113.1	3 933.8	4 480.5	5 170.4
SGA 最坏结果	3 228.0	3 107.0	3 922.0	4 469.0	5 154.0
SGA 最大遗传次数	100	100	100	100	100
SGA 平均遗传次数	100	100	100	100	100
SGA 最小遗传次数	100	100	100	100	100

0.01 ,终止迭代次数 100 代. n 个项目的背包问题为

$$\max \sum_{i=1}^{n} x_i p_i \tag{4}$$

受约束于 $\sum_{i=1}^{n} w_i x_i \leq b = 0.08 \sum_{i=1}^{n} w_i$, $x_i = 0$ 或 1 , $i = 1, 2, \cdots, 20$. n 分别取 20 ,30 , …,90 和 100 ; w, p 为随机整数取 1～100 的 n 维向量分量. 对同一个背包问题 SGA 和 HGA 各做 10 轮实验. 从实验结果可以看出 , HGA 不仅精度比 SGA 高(HGA 的最差精度都比 SGA 的平均精度要高 ,而 SGA 的精度有时比贪心法还差) ,而且 HGA 的收敛速度要比 SGA 的快(HGA 平均遗传代数为 16.65 代就得到近似最优解 ,而 SGA 则需要 100 代) .

3　结束语

本文将具有较强搜索能力的 SGA 与传统的贪心算法相结合 ,提出一种增强 SGA 搜索导向的混合遗传算法. 它加快了算法的搜索速度 ,又提高了算法的精度 ,而且还能够有效地克服传统方法易陷入局部最优的缺点. 实验表明 ,本文算法是求解 0-1 背包问题的有效算法.

参　考　文　献

1　康立山 ,谢　云 ,尤矢勇 ,等. 非数值并行算法 ——模拟退火法[M]. 北京 :科学出版社 ,2000. 9～12
2　郭晓晖. 遗传算法在求解背包问题中的应用[J]. 大连铁道学院学报 ,2001 ,22(3) :32～35
3　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
4　李　娟 ,方　平 ,周　明. 一种求解背包问题的混合遗传算法[J]. 南昌航空工业学院学报 ,1998 ,12(3) :31～35
5　王小平 ,曹立明. 遗传算法 ——理论 ,应用与软件实现[M]. 西安 :西安交通大学出版社 ,2002. 136～143

A Hybrid Genetic Algorithm for Solving 0-1 Knapsack Problem

Song Haizhou　　Wei Xuzhen
(Department of Mathematics , Huaqiao University , 362021 , Quanzhou , China)

Abstract　A value density is designed for 0-1 knapsack problem and on this basis , a bybrid genetic algorithm is proposed for solving 0-1 knapsack problem. Compared with conventional method and single genetic algorithm on the basis of large quantity of numerical experiments , as shown by experimental results , this algorithm is very effective.

Keywords　knapsack problem , infeasible solution , greedy method , genetic algorithm