

一种求解多维背包问题的和声优化搜索算法

拓守恒

(陕西理工学院 数学与计算机科学学院, 陕西 汉中 723000)

摘 要: 提出了一种解决多维背包问题的和声优化搜索算法. 算法采用 n 进制编码初始化和声记忆库, 采用自适应的和声保留概率、音调调节概率和音调调节步长产生新解, 通过修正算子修正不可行解, 以保证解满足约束条件, 然后利用非劣解集更新算法优化最优前端, 使其扩大覆盖率, 保证均匀性. 通过 20 次随机实验, 结果表明, 该算法能够有效的克服早熟收敛, 能够保持种群多样性和求解精度, 具有解决复杂多维背包问题的能力.

关键词: 和声搜索算法; 多维背包问题; n 进制编码

中图分类号: TP181

文献标识码: A

文章编号: 1000-7180(2012)06-0076-05

A Harmony Search Algorithm for Resolving Multi-Dimensional Knapsack Problems

TUO Shou-heng

(School of Maths & Computer Science, Shaanxi University of Technology, Hanzhong 723000 China)

Abstract: A multi-band encoding harmony search algorithm is proposed for MKP in this work. First, harmonic memory is initialized by multi-band encoding. Adaptive harmony memory consideration rate, pitch adjusting rate and band width are used to produce new solution. The infeasible solution is modified by modified algorithm for satisfy the constraint condition. Then, a pareto-optimal-set optimization algorithm is used to enlarge coverage rate and uniformity. Finally, 20 independent experiments illustrate that the proposed algorithm has some advantages in convergence velocity, population diversity and solution precision compared with traditionally algorithm.

Key words: harmony search algorithm; multi-dimensional knapsack problems; multi-band encoding

1 引言

多维背包问题 (multidimensional knapsack problem, MKP) 是一种 NP 难的多目标组合优化问题. 在项目投资决策、项目管理、资源分配、物流运输、网络资源分配等问题有广泛的应用. MKP 通过给出所有实体的价值和重量(成本、代价), 在满足所有约束的条件下, 从中选择出 n 个互不相交的子集, 使其各子集内物品的的价值之和最大. 针对 MKP 问题, 研究者提出了很多不同的求解算法, 目前, 研究的热点主要通过生物进化算法(遗传算法 genetic algorithms GA、粒子群优化算法 Particle swarm

optimization PSO、蚁群算法 Ant colony optimization ACO 等算法)和多目标优化策略进行求解.

上述算法对对多维背包问题基本都采用二进制编码, 对于具有 n 个背包的问题, 解 x 为 $n * m$ 的二进制编码矩阵, 这样采用生物进化算法时, 当种群个体数为 NP 时, 需要 NP 个大小为 $n * m$ 二进制编码矩阵 x , 在进化过程中, 进行变异、交叉等算法时, 时间复杂度至少为 $O(n * m * NP)$, 这样对于一个复杂的多维背包问题, 随着背包个数, 物品个数的增加, 算法的运行时间会大幅增加, 收敛速度降低. 本文提出一种基于和声搜索算法的多维背包问题求解算法, 算法采用 n 进制编码, 每个个体采用 1 维的 n

收稿日期: 2011-10-25; 修回日期: 2011-12-01

基金项目: 国家高技术研究发展计划(“八六三”计划)(2008AA01A303); 国家自然科学基金项目(81160183)

进制串来表示, m 件物品通过模拟音乐师调整乐器音调来选择背包装入^[1]。

2 问题描述

2.1 多维背包问题模型

多维背包问题模型如下: $\max y = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$,
 $i = 1, 2, \dots, n$;

$$\text{s. t. : } \sum_{j=1}^m w_{ij} \varphi(i, x_j) \leq W_i,$$

$x = (x_1, x_2, \dots, x_m) \in \{1, 2, \dots, n\}^m, (i = 1, 2, \dots, n, n \text{ 是背包数量})$;

$$\text{其中, } \varphi(i, x_j) = \begin{cases} 1, & x_j = i \\ 0, & \text{其他} \end{cases}, (i = 1, 2, \dots, n), x_j$$

$= i$ 表示物品 j 被装入背包 i 中了;

$$f_i(x) = \sum_{j=1}^m v_{ij} \varphi(i, x_j), (i = 1, 2, \dots, n); v_{ij} \text{ 表}$$

示物品 j 装入背包 i 时的价值. w_{ij} 表示物品 j 装入背包 i 时的重量(代价), W_i 是背包 i 的容量(最大可承受重量值)。

2.2 Pareto 多目标优化算法

由于多维背包问题是多目标优化问题, 各目标是让每个背包装中物品的价值之和最大, 但各个背包又彼此冲突, 相互制约, 需要在各目标之间进行协调权衡和折中处理, 使得所有目标尽可能达到最优. 最优解不再是在给定约束条件下使得所有目标最大的解, 而是 Pareto 最优解集. 下面是有关 Pareto 解集的描述^[2]。

(1) Pareto 支配: 解 x 支配解 $y: (x > y)$, 当且仅当

$$\forall i \in \{1, 2, \dots, n\}, f_i(x) \geq f_i(y)$$

$$\exists i \in \{1, 2, \dots, n\}, f_i(x) > f_i(y)$$

(2) Pareto 最优: 解 x 是 Pareto 最优解当且仅当 $\neg \exists y: y > x$.

(3) Pareto 最优解集: 所有最优解的集合 $Pt = \{x | \neg \exists y > x\}$.

(4) Pareto 最优前端: 所有 Pareto 最优解对应的目标函数值所形成的区域 P_F :

$$P_F = \{f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \mid x \in Pt\}$$

和单目标优化算法不同, 多目标优化算法的目标是: (1) 获得一组数量尽可能大的非劣解; (2) 这组解要接近问题的全局 Pareto 最优前端; (3) 这组解要尽可能均匀的分布在最优前端. 为了实现上述目

标, 相应的有 3 种算法性能评价指标: 覆盖率指标, 收敛性指标, 均匀性指标。

3 和声搜索算法

和声搜索算法 (Harmony Search Algorithm: HSA)^[3] 是 Geem 等人提出的一种新的启发式搜索算法, 算法模拟音乐师创作音乐时调整乐队中各乐器音调的过程, 音乐师通过调整各个乐器的音调使得音乐达到一种最美妙的和声状态. HSA 通过类比音乐和最优化问题的相似性而提出. 目前, 该方法已在组合优化^[4]、多目标优化^[5]等问题中得到了广泛应用. 和声搜索算法的基本步骤如下:

Step1 设置参数. (1) 决策变量个数 m . (2) 各决策变量的取值范围. (3) 和声记忆库的大小 N . (4) 和声记忆的保留概率. (5) 音调调节概率. (6) 最大搜索代数 T .

Step2 初始化和声记忆库(种群).

Step3 利用下列音符产生规则新解.

(1) 记忆中的和声音符. 产生随机数 $r_1 = \text{rand}(0, 1)$, 如果 $r_1 < \text{HMCR}$, 则在和声记忆库内选取乐器的音符. 在音符空间内随机产生音符分量.

(2) 用接近记忆的音符替换. 产生随机数 $r_2 = \text{rand}(0, 1)$, 如果 $r_2 < \text{PAR}$, 利用通过微调分量部分音符, 产生扰动.

Step4 更新和声记忆库. 如果新产生的解好于记忆库内最差解, 将其替换掉.

Step5 终止条件判断. 如果满足终止条件则结束. 否则转至 Step3.

4 基于和声搜索算法的多维背包问题求解算法

4.1 初始化和声记忆库

(1) 计算物品的单位价值比: $k(i, j) = \frac{\text{value}(i, j)}{\text{weight}(i, j)} (i = 1, 2, \dots, n; j = 1, 2, \dots, m)$, 并根据价值比降序排序.

(2) 随机初始化和声记忆库

定义和声矢量 $(x^1, x^2, \dots, x^m)$ 为 $[LB, UB]$ 内的整数, 本文中 $LB = 0, UB = n$; x^j 表示物品 j 装入背包的编号, $x^j = 0$ 表示物品 j 不放入任何背包, $x^j = k (k = 1, 2, \dots, UB)$ 表示物品 j 装入背包 k 中. 具体算法如下:

算法 1: 和声记忆库初始化算法

For $i = 1: N$

For $j=1:m$

$r = \text{select from } LB_j \text{ to } UB_j; \% \text{随机选取背包}$

if $w_i + \text{weight}(r, j) \leq W(r)$ then

$x_i^j = r;$

$w_i = w_i + \text{weight}(r, j);$

$$HM = \begin{bmatrix} (x_1^1 & x_1^2 & \cdots & x_1^m) & (w_1^1, w_1^2, \cdots, w_1^n) & (v_1^1, v_1^2, \cdots, v_1^n) \\ (x_2^1 & x_2^2 & \cdots & x_2^m) & (w_2^1, w_2^2, \cdots, w_2^n) & (v_2^1, v_2^2, \cdots, v_2^n) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ (x_N^1 & x_N^2 & \cdots & x_N^m) & (w_N^1, w_N^2, \cdots, w_N^n) & (v_N^1, v_N^2, \cdots, v_N^n) \end{bmatrix}.$$

式中, m 是物品总数量, $x_i^j \in \{0, 1, 2, \cdots, n\}$, (n 是背包个数). 表示将物品 j 装入背包 x_i^j 中, 如果 $x_i^j = 0$ 表示, 物品 j 没能装入任何背包. w_i 是背包 k 中物品的重量, v_i 是背包 k 中物品的价值总和.

4.2 自适应新和声产生算法

新和声的产生基于现有记忆、微调扰动和随机选择三种规则. 算法描述如下:

算法 2: 自适应新和声产生算法

For $j=1:n$

If $\text{rand}(0,1) < \text{HMCR}$ $x_{\text{new}}^j = x_r^j, r \in [1, N]$

If $\text{rand}(0,1) < \text{PAR}$ // 随机扰动

$x_{\text{new}}^j = \text{Roulette_select_From}(HM(x_j))$

EndIf

Else

$x_{\text{new}}^j = \text{Select from } [LB, UB] // \text{随机产生}$

EndIf

EndFor

其中参数 HMCR 和 PAR 根据下列公式进行自适应音符选择:

(1) $\text{HMCR} = e^{-\frac{(s-u)^2}{2\sigma^2}}, s = t/T, u \in [0.8, 1.5], \delta \in [0.7, 2]$. HMCR 选择概率变化曲线如图 1 所示.

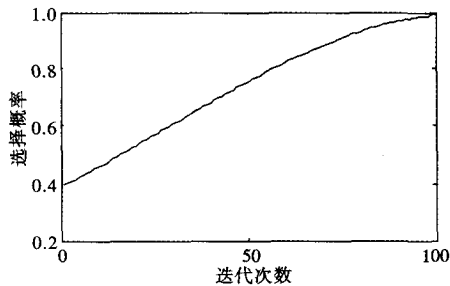


图 1 HMCR 选择概率变化曲线

(2) $\text{PAR} = \text{PAR}_{\min} + \frac{t}{T} (\text{PAR}_{\max} - \text{PAR}_{\min})$.

4.3 修正算法

由于经过三种规则产生的新和声, 其对应解的部分背包内物品重量可能会超过其最大容量, 部分

$v_i = v_i + \text{value}(r, j);$

else $x_i^j = 0;$

End

End

背包可能会很空, 因此需要对背包进行修正. 修正时, 如果超重, 优先去掉单位价值比小的物品. 如果背包没有装满, 从未被装包的物品中选择价值比重最大的物品进入背包, 直至不能再装^[6-7].

算法 3: 修正算法

假如: $x_{\text{new}} = (x_1, x_2, \cdots, x_m)$ 是需要修正的和声矢量(个体编码串).

For $r=1:n$

If $\text{weight}(r) > W(r)$ %第 i 个背包超重

$\text{index} = \text{getMinValue}(r); \% \text{获得价值重量比最小物品的位置.}$

$x_{\text{new}}(\text{index}) = 0; \% \text{物品 index 退出背包.}$

End

%检测是否有物品能够进入背包 r

For $i=1:m$ % m 是物品个数

If $\text{weight}(r) + w(r, i) \leq W(r) \ \& \ ks(i) = 0$ Then

$x_{\text{new}}(i) = r; \text{end IF}$

End For

End For

4.4 更新和声记忆库

如果和声记忆库中存在解 x_i 受 x_{new} 支配, 则用 x_{new} 替换 x_i .

For $i=1:N$

If $f(x_i) < f(x_{\text{new}})$ then $x_i = x_{\text{new}}$

EndFor

4.5 基于 NSGA2 和 SPEA2 的非劣解集的选择

本文结合 SPEA2 和 NSGA2 算法, 通过对非劣解集的更新进行多目标优化. 算法描述如下:

算法 4: 非劣解集更新算法

设 HM 是和声记忆库, A 是非劣解集, Na 是 A 中非劣解个数的最大值.

For $t=1:T$

(1) 计算和声记忆库 HM 和 A 中个体的适应值.

(2) 将和声记忆库 HM_t 中的非劣解拷贝到 A_t 中.

(3) 确定 $A_{t+1} = \{x^i | x^i \in HM_t \cup A_t \text{ 非劣}\}$, 如果 $|A_{t+1}| > Na$, 则修剪 A_{t+1} ; 如果 $|A_{t+1}| < Na$, 则将 HM_t 和 A_t 中的受支配解加入到 A_{t+1} 中, 直到的 $|A_{t+1}| = Na$.

End

修剪策略是根据文献中的空间均匀性度量指标^[8]进行, 定义如下:

$$S = \frac{\sqrt{\frac{1}{n_p} \sum_{i=1}^{n_p} (d_i - \bar{d})^2}}{\bar{d}}, \text{ 其中 } \bar{d} = \frac{\sum_{i=1}^{n_p} d_i}{n_p}, n_p \text{ 是}$$

目前非劣解集 A_t 中解的数量, d_i 为第 i 个解与 A_t 中最近解的距离。

$$d_i = \min_j \sqrt{\sum_{k=1}^n (f_k(x_i) - f_k(x_j))^2}, i \neq j, i, j = 1, 2, \dots, n_p,$$

在每次的拥挤修剪中, 将 d_i 从大到小排序, 选取 d_i 较大的前 n_{pf} 个非劣解作为最优非劣解集。

4.6 基于和声搜索算法的多维背包问题求解算法

算法5: 基于和声搜索算法的多维背包问题求解算法

Step1: 设置参数: 设定最大进化代数 T , 和声记忆库大小 N , 设定和声记忆的保留概率 $HMCR$, 音调调节概率 PAR 和和声矢量的下限和上限 $[LB, UB]$ 等。

Step2: 根据算法1进行初始化和声记忆库 HM 。

Step3: 利用自适应和声产生算法产生新和声; 利用修正算法对个体进行修正操作, 使其满足约束条件; 采用更新算法更新和声记忆库。

Step4: 根据算法4更新非劣解集, 使其尽可能满足覆盖率、收敛性和均匀性等指标要求。

Step5: 判断是否满足终止条件, 如果否转至 Step3, 否则结束。

5 实验仿真

为了测试本文算法的有效性, 数据选取 Zitzler 的主页 <http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/> 的测试数据, 物品个数分别为 100, 250 和 500, 背包数为 2 和 3. 并和经典的多目标优化算法 SPEA2, NSGA2 进行比较. 测试中, 设置最大进化代数为 $T=500$; 和声记忆库大小为 100, 非劣解集规模为 250; 每种算法独立运行 20 次. 获得最优非劣解集的分布情况通过

图2、图3、图4和图5进行了描述。

实验结果表明, 对于 2 背包 250 和 500 物品问

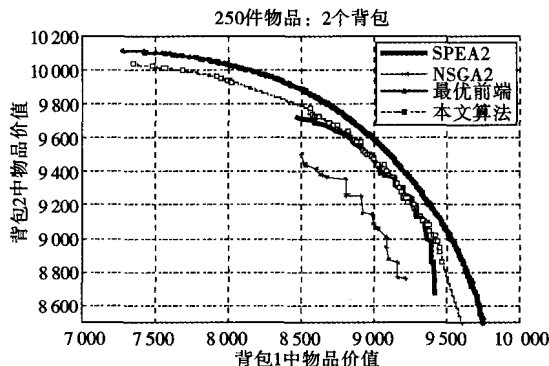


图2 2 背包 250 件物品时算法的最优非劣解集

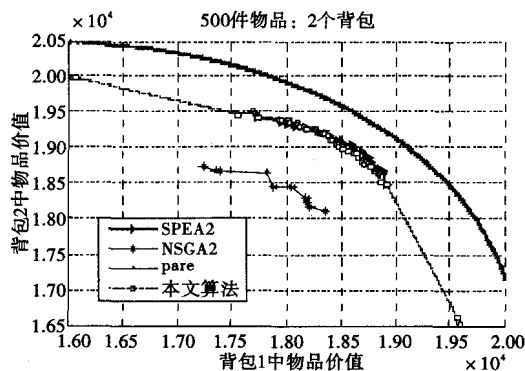
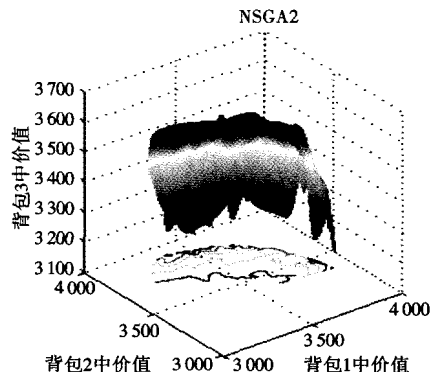
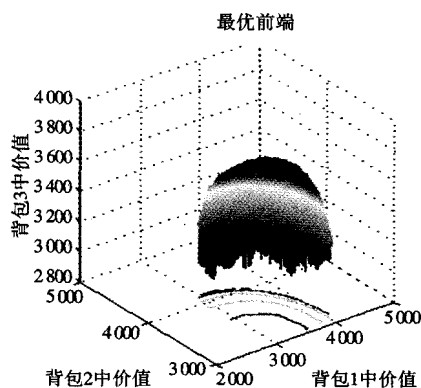


图3 2 背包 500 件物品时算法的最优非劣解集



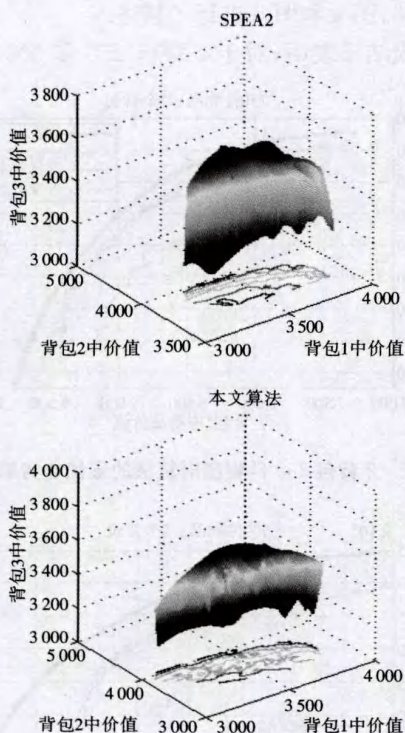


图4 3背包100件物品,获得最优 Pareto 解集分布

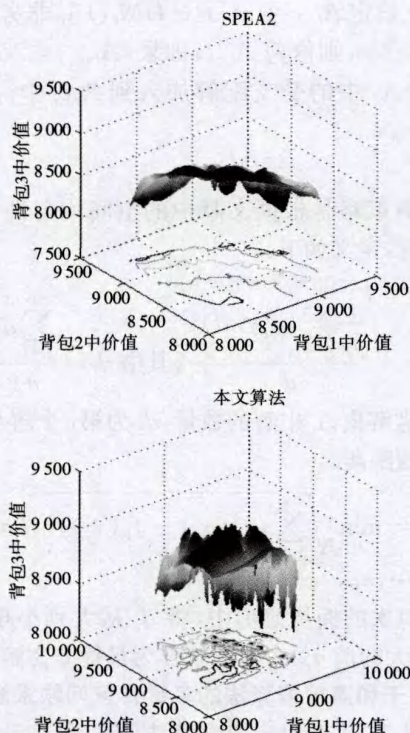
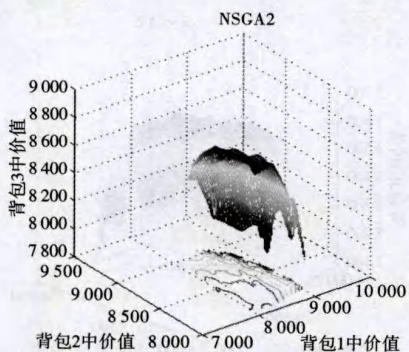


图5 3背包250件物品最优 Pareto 解集分布情况

题,本文算法在收敛性方面和 SPEA2 算法相当,在覆盖率和均匀性都优于 NSGA2 和 SPEA2 算法.对于 3 背包问题,本文算法在收敛性、覆盖率和均匀性都明显优于 NSGA2 和 SPEA2 算法,而 NSGA2 和 SPEA2 算法存在一些局部最优.

由图 5 中 3 背包 250 件物品(该问题最优前端未知)的优化结果可以发现,本文算法在收敛性和覆盖率方面和 NSGA2 和 SPEA2 算法相比具有优势,但均匀性不够好,存在最优解集局部集中性,需要对此进行改进.总体来看,本文算法不会随着问题物品数量和目标数的增加而显著降低性能.能够快速的收敛于最优前端,具有应用价值.



6 结束语

本文采用 n 进制编码策略,利用和声搜索算法的三种和声调解规则进行物品装包,用修正算法对背包进行修正,通过基于 Pareto 的最优前端更新算法更新非劣解集.测试结果表明,算法在最优解集的覆盖率、均匀性和收敛性方面都具有一定的优势和应用能力.

参考文献:

- [1] 李玲玮,张建军,张涛,等.面向任务的拼修策略问题及求解算法[J].系统工程理论与实践,2009(7):97-104.
- [2] 杜巍,李树苗,陈煜聪.一种求解多维背包问题的小世界算法[J].西安交通大学学报,2009(2):10-14.
- [3] 刘勇,马良.元胞微粒群算法及其在多维背包问题中的应用[J].管理科学学报,2011(1):86-96.
- [4] 喻学才,张田文.多维背包问题的一个蚁群优化算法[J].计算机学报,2008(5):810-819.
- [5] 刘毅,宋玉阶.多维背包问题的 DNA 计算[J].生物数学学报,2008(1):180-186.
- [6] 雷德明,严新平,多目标智能优化算法及其应用[M].北京:科学出版社,2009.

(下转第 86 页)

参考文献:

- [1] 施晓红, 龙志强. 雷达训练模拟设计中的信号仿真方法研究 [J]. 系统仿真学报, 2008, 20(7): 1716.
- [2] 刘峰, 龙腾. 一种半实物雷达仿真系统硬件体系结构设计 [J]. 系统仿真学报, 2006, 18(S2): 643.
- [3] Mark A Richards. Fundamentals of radar signal processing [M]. New York: McGraw-Hill, 2005: 225.
- [4] 马晓岩, 向家彬. 雷达信号处理 [M]. 湖南: 湖南科学技术出版社, 1998: 163.
- [5] 刘朝晖, 韩月秋. 用 FPGA 实现 FFT 的研究 [J]. 北京理工大学学报, 1999, 19(2): 234.
- [6] Zhang Yongquan, Wang Xiqin. Algorithm and new structure for radar sliding MTD (moving target detec-

tion) [C]//The record of the IEEE International Radar Conference. Arlington, Virginia, 2001: 223-225.

作者简介:

陈勇 男, (1983—), 博士研究生. 研究方向为实时仿真、分布实时仿真等.

姚新宇 男, (1971—), 副教授, 硕士生导师. 研究方向为分布实时仿真、仿真系统集成等.

潘玉林 男, (1979—), 博士研究生. 研究方向为实时仿真、分布实时仿真等.

唐小凤 女, (1981—), 硕士研究生. 研究方向为图像处理、虚拟现实等.

(上接第 80 页)

- [7] Geem Z, Kim J, Loganathan G. A new heuristic optimization algorithm: harmony search [J]. Simulation, 2001, 76(2): 60-68.
- [8] Schott J R. Fault tolerant design using single and multi-criteria genetic algorithms [D]. Boston: Massachusetts Institute of Technology, 1995.

作者简介:

拓守恒 男, (1978—), 硕士, 讲师. 研究方向为进化计算、人工智能、神经网络等.