第 26 卷 第 8 期
Vol. 26 No. 8

控 制 与 决 策
Control and Decision

2011 年 8 月
Aug. 2011

文章编号: 1001-0920 (2011) 08-1121-05

一种求解多维背包问题的混合分布估计算法

王 凌, 王圣尧, 方 晨

(清华大学 a. 清华信息科学与技术国家实验室，b. 自动化系，北京 100084)

摘 要: 针对多维背包问题 (MKP), 提出一种基于分布估计算法的混合求解算法. 该算法基于优势种群构建概率模型, 并基于概率模型采样产生新个体; 同时, 提出一种基于 MKP 问题信息的修复机制, 有效修复采样后种群中的不可行解. 另外, 设计了一种自适应的局部搜索操作, 以增强算法的局部搜索能力. 基于标准测试集的仿真结果和算法比较验证了所提出的混合算法的有效性和鲁棒性.

关键词: 多维背包问题；分布估计算法；概率模型；混合算法

中图分类号: TP18 **文献标识码:** A

A hybrid distribution estimation algorithm for solving multidimensional knapsack problem

WANG Ling, *WANG Sheng-yao*, *FANG Chen*

(a. Tsinghua National Laboratory for Information Science and Technology，b. Department of Automation，Tsinghua University，Beijing 100084，China. Correspondent：WANG Ling，E-mail：wangling@tsinghua.edu.cn)

Abstract: A hybrid algorithm based on estimation of distribution algorithm is proposed to solve multidimensional knapsack problem(MKP). In the algorithm, the probabilistic model is built with the superior population and new individuals are generated based on probability model. Meanwhile, a repair operator based on MKP specific knowledge is proposed to effectively repair infeasible solutions generated by sampling. In addition, an adaptive local search is designed to enhance the exploitation ability. Simulation results based on benchmark testing problems and comparisons demonstrate the effectiveness and robustness of the proposed hybrid algorithm.

Key words: multidimensional knapsack problem；estimation of distribution algorithm；probabilistic model；hybrid algorithm

1 引 言

多维背包问题 (MKP)[1]是典型的 NP 难问题, 对其研究具有重要的学术意义. MKP 具有广泛的工程背景, 很多实际问题都可描述为 MKP 模型, 例如库存压缩问题[2], 项目选择与货物装卸[3], 分布式计算系统中分配处理器和数据库[4]等. 早期求解 MKP 的方法主要集中于精确求解算法, 如分支定界法[3]和动态规划法[5]等. 一般而言, 精确求解算法的计算量和存储量均很大, 难以有效求解大规模 MKP. 随着智能优化算法的提出与发展, 近年来出现了许多基于进化算法的研究. 文献 [6] 采用遗传算法 (GA) 求解 MKP, 通过在标准 GA 中引入基于问题信息的操作提高了算法的性能, 同时降低了算法的计算复杂度; [7] 设计了一种二进制蚂蚁算法, 采用不同的信息素更新和初始化方法进行优化搜索; [8] 采用分散搜索算法求解 MKP, 通过采用集成式存储和相应的强化搜索机制提高了算法性能.

分布估计算法 (EDA)[9]是一种新颖的群体进化算法, 近年来得到了广泛的研究与发展. PBIL (population based incremental learning)[10]是最早的 EDA 模型. 按照模型的复杂程度和变量间相互作用关系, EDA 算法可分为变量无关 EDA, 双变量相关 EDA 及多变量相关 EDA. PBIL 与 UMDA (univariate marginal distribution algorithm), cGA (compact GA)[11] 等属于变量无关 EDA. 双变量相关 EDA 的典型代表有 MIMIC (mutual information maximization for input

收稿日期: 2010-05-05；**修回日期:** 2010-06-11.

基金项目: 国家自然科学基金项目(70871065, 60834004)；教育部新世纪优秀人才支持计划项目(NCET-10-0505)；高等学校博士学科点专项科研基金项目(20100002110014).

作者简介: 王凌(1972−), 男, 教授, 博士生导师, 从事智能优化调度理论与方法等研究；王圣尧(1988−), 男(满族), 博士生, 从事智能优化调度的研究.

clustering)[12], COMIT (combining optimizers with mutual information trees)[13], BMDA (bivariate marginal distribution algorithm)[14]等. 多变量相关EDA的代表性算法有FDA (factorized distribution algorithms)[15], ECGA (extended compact GA)[16]和BOA (Bayesian optimization algorithm)[17]. 目前, EDA已在多个领域得到了研究, 包括特征选择、癌症分类、模式匹配、神经网络修剪、护理调度、二次分配、机械结构设计等[18]. 然而现有文献显示, EDA在MKP上几乎没有得到研究, 仅文献[19]提出了一种松弛互补EDA. 实验结果表明, EDA性能优于遗传算法和混合遗传算法.

本文针对MKP的特点, 提出一种新概率模型更新机制, 设计了基于问题信息的修复机制和变步长局部搜索操作, 进而提出一种有效的混合EDA. 基于大量标准问题的仿真结果与算法比较共同验证了所提出算法的有效性及鲁棒性.

2 多维背包问题

本质上, MKP是一般0-1规划问题. 其数学模型描述如下:

$$\max \sum_{j=1}^{n} p_j x_j. \tag{1}$$

$$\text{s.t. } \sum_{j=1}^{n} r_{i,j} x_j \leqslant b_i,\ i = 1, 2, \cdots, m; \tag{2}$$

$$x_j \in \{0, 1\},\ j = 1, 2, \cdots, n. \tag{3}$$

其中: n为物体的数量, p_j为物体j的价值, m为资源的种类数, b_i为资源i的总量, $r_{i,j}$为物体j消耗资源i的量. MKP的目标是选出若干件物体, 使得在满足资源约束的前提下这些物体的总价值最大. 通常, 所有p_j, b_i和$r_{i,j}$均非负, 且$r_{i,j} \leqslant b_i$, $\sum_{j=1}^{n} r_{i,j} > b_i$.

3 分布估计算法

分布估计算法是一种新型基于概率模型的群体进化算法. 与传统GA不同, EDA不采用交叉和变异操作, 而是先通过一个概率模型描述候选解在空间的分布, 采用统计学习手段从群体宏观的角度建立一个描述解分布的概率模型; 然后对概率模型随机采样产生新的种群, 如此反复, 进而实现种群的进化. 标准EDA的算法流程如下:

Step 1: 初始化种群.

Step 2: 选择优势群体.

Step 3: 构建概率模型.

Step 4: 随机采样.

Step 5: 生成新群体.

Step 6: 判断终止条件是否满足. 若满足, 则输出优化结果; 否则, 转Step 2.

可见, 概率模型是EDA的核心, EDA通过概率模型及其更新来描述解分布以及种群的整体进化趋势. 根据具体问题的不同, 需设计合适的概率模型及其更新机制. 另外, 由于EDA过多侧重于解空间宏观的全局优化, 单纯EDA的局部搜索能力有限. 为了得到更好的优化性能, 下节将提出一种适合于MKP的混合EDA.

4 混合分布估计算法

针对MKP, 分别就算法设计中问题解的表达、概率模型及其更新方式、不可行解的修复以及局部搜索进行阐述, 最后给出混合EDA的算法流程.

4.1 解的表达

种群中每一个体对应MKP的一个解, 每一个体用一个长度为n的0-1字符串表示, 如图1所示. $x_i = 1$表示选择物体i放入背包中, $x_i = 0$表示物体i没有被选择.

x_1	x_2	$\cdots$	x_{i-1}	x_i	x_{i+1}	$\cdots$	x_{n-1}	x_n

图 1 问题解的表达

4.2 概率模型及更新机制

算法采用一个包含n个分量的概率向量$p(x) = [p(x_1), p(x_2), \cdots, p(x_n)]$表示解空间分布的概率模型, 其中$p(x_i)$表示$x_i$取值为1的概率. 开始时算法将$p(x)$置为$[0.5, 0.5, \cdots, 0.5]$, 表示初始状态为均匀分布. 在随后的每代进化中, 算法基于概率向量$p(x)$进行采样以产生新个体. 即对于每一待定的x_i, 产生一个0～1间的随机数α, 若$\alpha < p(x_i)$, 则$x_i = 1$; 反之, $x_i = 0$. 如此反复, 产生M个个体并计算其目标值, 选择目标值最高的N个个体($N < M$), 采用机器学习中的Hebb规则更新概率$p(x)$, 即

$$p_{l+1}(x) = (1 - \alpha) p_l(x) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_l^k. \tag{4}$$

其中: $p_l(x)$为第l代的概率向量, x_l^k为第l代选中的第k个个体, $\alpha (0 < \alpha < 1)$为学习速率.

4.3 非可行解修复机制

在EDA中, 根据概率模型采样产生新个体时并未考虑解的可行性, 因此有可能产生的新个体不可行, 此时必须采用修复操作来处理. 目前处理MKP广泛采用基于伪效用比率的修改机制, 其核心操作是代理对偶方法[20]. 由于该修复机制需先求出MKP对偶问题的最优解, 这在一定程度上加大了算法的计算量, 降低了算法的效率. 在此, 提出一种基于问题结构的智能修复机制, 无需求解原问题的对偶问题.

算法初始化是通过简单运算得到$m \times n$阶矩阵

Q 和辅助矩阵 F. 其中: 矩阵 Q 的元素 $q_{ij} = p_j/r_{i,j}$; 辅助矩阵 F 每一行存储 n 个物体标号的一个排列, 即 f_{ij} 表示矩阵 F 中第 i 行元素升序排列时第 j 位的物体标号. 具体而言, 算法按以下修复操作的伪代码进行修复:

1) 输入一个不可行解 $X = [x_1, x_2, \cdots, x_n]$;

2) 计算 $R = \sum_{j=1}^{n} r_{ij}x_j,\ i = 1, 2, \cdots, n$;

3) For $i = 1, 2, \cdots, m$, do {

4) $j = 1$;

5) While $R_i > b_i$, do {

6) If $x_{f_{i,j}} = 1$, then $x_{f_{i,j}} = 0$ and $R_i = R_i - r_{i,f_{i,j}}$;

7) End if

8) $j = j + 1$;

9) End while

10) End for

11) 找到约束 i^*, 使得 $\forall i, b_{i^*} - R_{i^*}$ 最小, 令 $j = n$;

12) While $R_i < b_i$ and $j > 0$, do {

13) If $x_{f_{i,j}} = 0$ and $R_i + r_{i,f_{i,j}} \leqslant b_i$, then $x_{f_{i,j}} = 1$ and $R_i = R_i + r_{i,f_{i,j}}$;

14) End if

15) $j = j - 1$;

16) End while.

实际上, q_{ij} 是一维背包问题的伪效用比. 对于解 X 不满足的某一维约束 i, 上述操作基于该约束的伪效用比进行修复, 直至满足约束为止; 然后通过资源剩余量最小的维度的伪效用比来平衡各种资源的剩余量. 另外, 由于每次只考虑一个维度约束, 上述修复过程的运算量并不大.

4.4　变步长局部搜索

为了增强 EDA 的搜索能力, 平衡全局和局部搜索, 这里针对 MKP 设计一种变步长局部搜索. 具体步骤如下:

Step 1: 随机选择 s 个值为 1 的 x_j 置 0.

Step 2: 随机选择 $s + 1$ 个值为 0 的 x_j 置 1.

Step 3: 判断新解是否可行. 若不可行, 则进行修复操作.

Step 4: 判断是否达到最大搜索步数. 若是, 则结束局部搜索; 否则, 转 Step 1.

上述类似于二进制 GA 的 "基因" 随机变异很重要, 可帮助对解进行随机调整, 在一定程度上避免算法陷入某些局部区域, 从而增大找到全局最优解的可能性. 实验表明, 在参数 s 取值合适的情况下, 这种局部搜索可大大提高算法的性能. 另外, 搜索次数 Y 表征了局部搜索的强度. 本文设计一种变步长的局部搜索机制, 即事先设定一个初始步长, 随着进化代数的增加, 逐步增大步长.

4.5　混合分布估计算法

根据上述设计, 下面给出求解 MKP 的混合 EDA 的流程:

Step 1: 初始化概率向量 $p(x)$.

Step 2: 对 $p(x)$ 进行随机采样, 产生 M 个个体.

Step 3: 修复不可行解, 计算适应值.

Step 4: 对种群中的最优个体进行局部搜索 (按需调整步长).

Step 5: 选出最好的 N 个个体, 并更新概率向量 $p(x)$.

Step 6: 判断终止条件是否满足. 若满足, 则输出优化结果; 否则, 转 Step 2.

可见, 该算法结合分布估计算法和局部搜索对解空间进行全局和局部搜索, 同时采用修复操作处理不可行解. 在进化初期, 算法基于 EDA 思想重点对解空间进行粗搜索, 定位好的区域; 在进化后期, 算法已通过对解空间的抽样和统计逐步缩小了搜索范围, 此时通过局部搜索能细致地搜索优良区域, 进而获得性能更好的解.

由概率向量 $p(x)$ 的定义知, 若某一位置元素 $p_i(x)$ 接近 0 或 1, 则第 i 个物体完全可确定是否被选中. 因此, 出于对概率向量收敛性和算法整体计算效率的考虑, 算法进化中, 若每一位都大于 0.99 或小于 0.01, 或总进化代数已达到 5 000 代, 则算法终止.

5　仿真测试和比较

选取国际上的标准测试问题 (测试集1)[19] 和 MK_gk 测试集 (测试集 2) 对 HEDA 算法进行性能测试. 测试集 1 包含 8 个问题 (物体个数分别为 15, 20, 28, 39, 60, 105), 文献 [19] 采用这些数据对他们提出的松弛互补 EDA 进行测试并与混合遗传算法进行了比较, 所得结果表明了 EDA 性能的优越性. 测试集 2, 即 MK_gk 测试集 (http://hces.bus.olemiss.edu/tools.html) 包含 11 个大规模问题, 物体个数从 100 到 2 500 不等. 在此先采用测试集 1 探讨本文提出的 HEDA 的性能, 并与文献 [19] 的松弛互补 EDA 进行比较; 然后采用测试集 2 考察 HEDA 求解大规模 MKP 的性能.

对于文献 [19] 采用的每个测试问题, HEDA 均独立运行 20 次, 统计平均性能 (AVG) 以及找到最优解的成功率 (SR), 基于此进行算法比较. 对于测试集 2 中的每个大规模 MKP, 统计 20 次独立实验所得解与已知最优解之间的最优相对偏差、平均相对偏差及相对偏差的方差, 基于此考察算法的优化质量和鲁棒性.

5.1 参数设置及其影响

HEDA参数设置如下: $N = 10$, $s = 2$, 初始步长$Y = 50$, 每进化20代Y增加1. 对于种群规模M和学习速率α, 进行如下考察, 探讨它们对HEDA性能的影响.

5.1.1 种群规模的影响

对于测试集1所有问题, 学习速率设为0.1, 种群规模分别取40, 80, 120, 160, 200. 相应HEDA的优化成功率如表1所示.

表 1 种群规模对 HEDA 成功率的影响 (测试集 1)

种群规模	40	80	120	160	200
成功率/%	64.375	76.875	81.250	83.750	85.000

对于测试集2所有问题, 学习速率设为0.1, 种群规模分别取200, 400, 600, 800, 1 000. 相应HEDA所得最优解与已知最优解的相对偏差如表2所示.

表 2 种群规模对 HEDA 最优相对偏差的影响 (测试集 2)

种群规模	200	400	600	800	1 000
相对偏差/%	0.30	0.21	0.18	0.18	0.16

由表1和表2可见, 增大种群规模可以提高HEDA的性能, 包括成功率和相对偏差. 其原因在于加大种群规模有利于算法更全面地搜索空间, 但必将增加计算量. 因此, 兼顾算法性能和效率, 种群规模应适当设置.

5.1.2 学习速率的影响

对于测试集1所有问题, 种群规模设为200, 学习速率分别取0.5, 0.1, 0.05, 0.01, 0.001. 相应HEDA的优化成功率如表3所示.

表 3 学习速率对 HEDA 成功率的影响 (测试集 1)

学习速率	0.001	0.01	0.1	0.5
成功率/%	100	88.75	85.00	58.13

对于测试集2所有问题, 种群规模设为1 000, 学习速率分别取0.5, 0.1, 0.05, 0.01. 相应HEDA所得最优解与已知最优解的相对偏差如表4所示.

表 4 学习速率对 HEDA 最优相对偏差的影响 (测试集 2)

学习速率	0.01	0.05	0.1	0.5
相对偏差/%	0.24	0.14	0.16	0.32

由表3和表4可见, 对于中小规模问题, 减小学习速率可以提高HEDA的性能. 其原因在于过大的学习速率会使种群快速趋向于当前最优解, 从而导致算法早熟收敛而陷入局部最优. 对于大规模问题, HEDA的最优相对偏差随着学习速率的加大先降后升. 学习速率过大的影响原因类似于中小规模问题; 而学习速率过小, 使得在搜索空间很大的情况下(大规模问题), 优势种群对整体种群的进化指导能力变差, 进而导致算法收敛缓慢且寻优能力下降.

基于上述算法参数对性能影响的讨论, 兼顾算法优化质量和效率, 对两个测试集合分别设置如下基准参数: 对于测试集1, 种群大小为200, 学习速率为0.001; 对于测试集2, 种群大小为1 000, 学习速率为0.05.

5.2 实验统计结果与比较

首先, 基于测试集1考察HEDA的性能, 并与文献[19]的互补松弛分布估计算法(RCEDA)进行比较. 这里, 对每个问题均独立运行100次, 统计平均结果和成功率, 所得结果如表5所示.

表 5 基于测试集 1 的统计结果和比较

问题	n, m	已知最优	RCEDA[19] AVG	RCEDA[19] SN	HEDA AVG	HEDA SN
Sento1	60, 30	7 772	7 771.5	95	**7 772**	**100**
Sento2	60, 30	8 722	8 720.9	93	**8 722**	**100**
Weing7	105, 2	1 095 445	1 095 422.6	90	**1 095 445**	**100**
Weing8	105, 2	624 319	624 316.8	80	**624 319**	**100**
Knap15	15, 10	4 015	4 012.7	95	**4 015**	**100**
Knap20	20, 10	6 120	6 117.9	97	**6 120**	**100**
Knap39	39, 5	10 618	10 615.3	95	**10 618**	**100**
Knap28	28, 10	12 400	12 389.4	98	**12 400**	**100**

从表5可见, HEDA在100次独立实验中均能够100%地找到最优解, 而RCEDA不能一致地找到最优解. 从平均性能上看, HEDA的性能也优于RCEDA. 因此, HEDA的优化质量和鲁棒性均好于松弛分布估计算法. 由文献[19]知, RCEDA优于混合遗传算法, 因此HEDA也优于混合遗传算法, 这是一种更有效的求解MKP的群体进化算法.

为了进一步考察HEDA的性能, 下面采用规模更大的测试集2进行测试, 每个测试问题独立运行20次, 统计结果如表6所示.

表 6 基于测试集 2 的 HEDA 性能统计

问题	n, m	最优值	最优偏差/%	平均偏差/%	偏差方差
MK_gk01	100, 15	3 766	0.265 5	0.342 5	0.042 8
MK_gk02	100, 25	3 958	0.075 8	0.196 6	0.077 8
MK_gk03	150, 25	5 650	0.106 2	0.200 9	0.045 1
MK_gk04	150, 50	5 764	0.086 7	0.212 5	0.045 3
MK_gk05	200, 25	7 557	0.132 3	0.217 7	0.034 7
MK_gk06	200, 50	7 672	0.156 4	0.241 1	0.039 2
MK_gk07	500, 25	19 215	0.109 3	0.168 6	0.024 9
MK_gk08	500, 50	18 801	0.170 2	0.220 7	0.022 0
MK_gk09	1 500, 25	58 085	0.117 1	0.133 6	0.008 7
MK_gk10	1 500, 50	57 292	0.132 7	0.159 0	0.011 9
MK_gk11	2 500, 100	95 231	0.190 1	0.216 7	0.013 6

由表6可见, HEDA找到最优解与已知最优解的相对偏差很小, 最差的最优相对偏差仅为0.265 5%, 最差平均相对偏差仅为0.342 5%. 因此, HEDA对大规模MKP的优化质量令人满意. 另外, 对于所有测试问题的最差偏差方差仅为0.077 8%, 可见HEDA对初始种群具有很强的鲁棒性.

总之, 本文提出的混合分布估计算法是求解多维背包问题的一种有效方法.

6 结 论

本文提出了一种有效解决多维背包问题的混合分布估计算法, 同时设计了一种不可行解的修复机制以及一种变步长的局部搜索方法. 基于标准测试问题的研究表明, HEDA无论对于中小规模MKP, 还是大规模MKP, 均具有优良的优化质量和鲁棒性, 而且HEDA的性能优于其他文献所提出的松弛分布估计算法和混合遗传算法, 是一种更有效的群体进化算法.

进一步的工作是针对MKP构造双变量或多变量相关的概率模型, 进而得到其他类型的分布估计算法, 并探讨EDA对其他类型组合优化问题的应用, 如生产调度问题等.

参考文献(References)

[1] Chu P, Beasley J. A genetic algorithm for the multidimensional knapsack problem[J]. J of Heuristics, 1998, 4(1): 63-86.

[2] Gilmore P C, Gomory R E. The theory and computation of knapsack functions[J]. Operation Research, 1966, 14(6): 1045-1074.

[3] Shih W. A branch and bound method for the multiconstraint zero-one knapsack problem[J]. J of the Operational Research Society, 1979, 30(4): 369-378.

[4] Gavish B, Pirkul H. Allocation of databases and processors in a distributed computing system[C]. Proc of the Int Conf on Management of Distributed Data Processing. Paris, 1982: 215-231.

[5] Toth P. Dynamic programming algorithms for the zero-one knapsack problem[J]. Computing, 1980, 25(1): 29-45.

[6] Chu P C, Beasley J E. A genetic algorithm for the multidimensional knapsack problem[J]. J of Heuristics, 1998, 4(1): 63-86.

[7] Kong M, Tian P, Kao Y C. A new ant colony optimization algorithm for the multidimensional knapsack problem[J]. Computers & Operations Research, 2008, 35(8): 2672-2683.

[8] Hanafi S, Wilbaut C. Scatter search for the 0-1 multidimensional knapsack problem[J]. J of Mathematical Modelling and Algorithms, 2008, 7(2): 143-159.

[9] Mühlenbein H, Paass G. From recombination of genes to the estimation of distributions I: Binary parameters[J]. Lecture Notes in Computer Science, 1996, 1141(1): 178-187.

[10] Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning[R]. Pittsburgh: CMU, 1994.

[11] Harik G R, Lobo F G, Goldberg D E. The compact genetic algorithm[C]. Proc of the IEEE Conf on Evolutionary Computation. Indianapolis, 1998: 523-528.

[12] Jordan M I, Lecun Y, Solla S A. Advances in neural information processing systems[M]. Cambridge: MIT Press, 1997: 1-20.

[13] Baluja S, Davies S. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space[C]. Proc of the 14th Int Conf on Machine Learning. San Francisco, 1997: 30-38.

[14] Benítez J M. Advances in soft computing: Engineering design and manufacturing[M]. London: Springer-Verlag, 1999: 3-30.

[15] Mühlenbein H, Mahnig T. Convergence theory and applications of the factorized distribution algorithm[J]. J of Computing and Information Technology, 1999, 7(1): 19-32.

[16] Harik G. Linkage learning via probabilistic modeling in the ECGA[R]. Illinois: UIUC, 1999.

[17] Pelikan M, Goldberg D E, Cantú-Paz E. BOA: The Bayesian optimization algorithm[C]. Proc of the Genetic and Evolutionary Computation. San Francisco, 1999: 525-532.

[18] Zhou S D, Sun Z Q. A survey on estimation of distribution algorithms[J]. Acta Automatica Sinica, 2007, 33(2): 113-124.

[19] Yang G Y, Ouyang Z M, Quan H Y. Relaxed complemental estimation of distribution algorithm for the multidimensional knapsack problem[J]. Computer Engineering and Applications, 2007, 43(12): 77-80.

[20] Pirkul H. A heuristic solution procedure for the multiconstraint zero-one knapsack problem[J]. Naval Research Logistics, 1987, 34(2): 161-172.