文章编号: 1001-0920 (2016) 07-1178-07 DOI: 10.13195/j.kzyjc.2015.0690

求解多维背包问题的蚁群-拉格朗日松弛混合优化算法

任志刚, 赵松云, 黄姗姗, 梁永胜

(西安交通大学电子与信息工程学院, 西安 710049)

摘 要: 针对多维背包问题 (MKP) NP-hard、约束强的特点, 提出一种高效的蚁群-拉格朗日松弛 (LR) 混合优化算法. 该算法以蚁群优化 (ACO) 为基本框架, 并基于 LR 对偶信息定义了一种 MKP 效用指标. ACO 使得整体算法具有全局搜索能力, 所设计的效用指标将 MKP 的优化目标与约束条件有机地融合在一起. 该指标一方面可以用来定义 MKP 核问题, 降低问题规模; 另一方面, 可以用作 ACO 的启发因子, 引导算法在有希望的解区域中强化搜索. 在大量标准算例上的测试结果表明, 所提出算法的鲁棒性较好; 与其他已有算法相比, 在求解质量和求解效率方面均具有很强的竞争力.

关键词: 多维背包问题; 蚁群优化; 拉格朗日松弛; 核问题

中图分类号: TP18 **文献标志码:** A

Hybrid optimization algorithm of ant colony optimization and Lagrangian relaxation for solving multidimensional knapsack problem

REN Zhi-gang, ZHAO Song-yun, HUANG Shan-shan, LIANG Yong-sheng

(School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. Correspondent: REN Zhi-gang, E-mail: renzg@mail.xjtu.edu.cn)

Abstract: A hybrid optimization algorithm that integrates ant colony optimization(ACO) with Lagrangian relaxation(LR) is proposed for solving NP-hard and strongly constrained multidimensional knapsack problems(MKP). This algorithm takes ACO as the basic framework and defines a novel utility index for MKP based on LR dual information. ACO endows the algorithm with global search ability, and the designed utility index organically combines the optimization object and the constraint conditions of MKP together. Benefiting from this characteristic, the utility index is used to define the core problem for MKP on the one hand, with the aim of reducing the problem scale. On the other hand, it can be used as the heuristic factor of ACO, directing the algorithm to intensively search those promising solution areas. Simulation results on a large number of benchmark instances show that the proposed algorithm is of strong robustness. Compared with existing algorithms, it is also highly competitive in terms of solution quality and efficiency.

Keywords: multidimensional knapsack problem; ant colony optimization; Lagrangian relaxation; core problem

0 引 言

多维背包问题 (MKP) 是运筹学领域一经典的组合优化问题, 被证明是一种 NP-hard 问题, 其求解目标是在满足各项资源约束的前提下, 从候选对象集中找出可以使目标价值达到最大的对象子集[1]. MKP 具有广泛的工程背景, 可以用来描述资本预算、资源分配、分布式数据库处理等实际问题[2], 因此, 研究 MKP 的求解算法具有重要的理论意义和实际价值.

MKP 的求解算法分为精确算法和启发式算法. 精确算法主要为分支定界法[3]和动态规划法[4], 这类方法虽然可以求得最优解, 但计算复杂度大, 仅适于规模较小的问题; 启发式算法并不立足于寻找问题的最优解, 而是期望在可接受时间内获得近优解. 目前, 有效求解 MKP 的启发式算法主要是各种智能优化算法, 包括遗传算法[1]、禁忌搜索算法[5]、分布估计算法[6]、果蝇优化算法[7]、和声搜索算法[8]和蚁群优化 (ACO) 算法[9-10]等. 其中, ACO 算法凭借其适应性强、收敛速度快等优势吸引了众多学者的关注和研究.

现有成果在应用智能优化算法求解 MKP 时通常以算法为主导, 主要研究 MKP 解的编码方式和更

收稿日期: 2015-06-01; **修回日期:** 2015-10-25.

基金项目: 国家自然科学基金项目(61105126); 中国博士后科学基金项目(2014M560784).

作者简介: 任志刚(1982–), 男, 副教授, 博士, 从事智能计算与机器学习、复杂系统的建模优化与控制等研究; 赵松云(1978–), 男, 硕士生, 从事智能计算、组合约束优化的研究.

新机制, 使其符合所选算法的模型要求, 而对MKP自身内在信息的利用程度较低. 具体到ACO算法, 现有文献侧重于在解的表示与构造层面研究信息素与MKP解的关联方式、候选解构造规则和信息素更新规则[9], 取得了较好的求解结果, 证实了应用ACO求解MKP的可行性. 然而, 这些算法机械地套用ACO的标准过程, 对MKP自身信息的利用仅体现在一些简单直观的启发信息上, 没有充分考虑MKP的约束特点, 这就限制了其性能的进一步提高.

近年来, 利用传统数学规划方法中的松弛技术提取MKP的内在结构信息, 并用以指导智能优化算法的搜索过程[10-12], 是MKP求解算法的一个发展趋势; 另一方面, 随着实际工程问题的规模不断增大, 对高维MKP的求解算法的需求日益增强. 围绕上述两点, 本文提出一种蚁群-拉格朗日松弛(LR)混合优化算法, 简称为AL-MKP. 与常规算法不同的是, 该算法并不直接求解原始的MKP, 而是首先利用LR[13]对偶信息为MKP对象定义一种效用指标, 然后根据该指标构建一低维的MKP核问题, 最后利用ACO进行求解. 新定义的效用指标可以进一步用作ACO的启发因子, 引导算法在有希望的解区域中重点搜索, 提高搜索效率. 大量标准算例上的测试结果表明: 所提出算法的鲁棒性较好; 与其他已有算法相比, 在求解质量和效率方面均具有很强的竞争力.

1　MKP及其LR模型的数学描述

MKP本质上是0/1规划问题, 其数学模型为

$$\max P(X)=\sum_{j=1}^{n}p_jx_j. \tag{1}$$

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

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

其中: n为对象个数; p_j为对象j的价值; x_j为决策变量, 当对象j被选择时, $x_j=1$, 否则$x_j=0$; m为资源种类数; b_i为第i种资源的提供量; r_{ij}为对象j对资源i的需求量; 式(2)通常称为MKP的容量约束. MKP的求解目标是, 在满足m个容量约束的前提下, 从n个对象中找出可以最大化总价值的对象子集, 若$m=1$, 则MKP转化为经典的0/1背包问题. 一个定义良好的MKP通常满足p_j、b_i和r_{ij}均非负, 且$r_{ij}\leqslant b_i, \sum_{j=1}^{n}r_{ij}>b_i$.

在采用启发式算法求解MKP时, 需要综合考虑对象价值和容量约束的影响, 本文提出的AL-MKP算法利用LR实现这一目的. 对于给定的拉格朗日乘子向量$\Lambda=(\lambda_1,\lambda_2,\cdots,\lambda_m)\in\boldsymbol{R}_+^m$, MKP的LR形式为

$$L(\Lambda)=\max\Big[\sum_{j=1}^{n}p_jx_j+\sum_{i=1}^{m}\lambda_i\Big(b_i-\sum_{j=1}^{n}r_{ij}x_j\Big)\Big]=\max\sum_{j=1}^{n}v_j(\Lambda)x_j+\sum_{i=1}^{m}\lambda_ib_i; \tag{4}$$

$$\text{s.t. }x_j\in\{0,1\},\ j=1,2,\cdots,n. \tag{5}$$

其中$v_j(\Lambda)=p_j-\sum_{i=1}^{m}\lambda_ir_{ij}$为对象$j$的拉格朗日价值(LV). 记式(4)和(5)所示LR问题的最优解为$X_L^*(\Lambda)$, 可得

$$x_{Lj}^*(\Lambda)=\begin{cases}1, & v_j(\Lambda)\geqslant 0;\\ 0, & \text{otherwise.}\end{cases} \tag{6}$$

对于任意的$\Lambda\in\boldsymbol{R}_+^m$, 相应的$L(\Lambda)$均为原MKP的最优解提供了一个上界. 为了获得最紧上界, 定义MKP的拉格朗日对偶问题, 具体为

$$\min\{L(\Lambda)|\Lambda\in\boldsymbol{R}_+^m\}. \tag{7}$$

该问题以拉格朗日乘子向量Λ为优化变量, 其求解方法比较成熟, 一种常用方法是利用次梯度算法[14]求取近优解$L(\Lambda^*)$.

对于近优的拉格朗日乘子向量Λ^*, 相应的LR问题的最优解$X_L^*(\Lambda^*)$虽然不一定是原MKP的最优解, 但通过局部调整可以转化为后者[13], 因此提供了许多有用信息: 1) LR问题的最优解与原MKP的最优解所包含的对象个数相差不多; 2) 二者具有许多相同对象, 且这些对象的LV值$v(\Lambda^*)$都偏大. 这同时也表明LV可以用来衡量MKP对象的综合效用, 因此本文将其作为对象的效用指标.

2　AL-MKP算法

2.1　MKP核问题

MKP中各对象的效用一般是大小不等的. 大量仿真实验表明, 效用值很大的对象一般都包含在最优解中, 而效用值很小的对象基本不会出现在最优解中. 那么, 求解MKP的核心难点在于如何确定那些具有一般效用值的对象是否包含在最优解中. 本文将由这些难以确定的对象构成的低维MKP定义为原MKP的核问题. 核问题的思想最早应用于0/1背包问题, 并产生了一些成功的求解算法[15]. 目前, 有关MKP核问题的研究还很少见, 本文将提出一种根据对象的LV确定MKP核问题的方法.

首先将所有对象按照$v(\Lambda^*)$进行降序排列, 得到的对象序列记为$\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$; 找到第$z$个对

象, 使得

$$\begin{cases} \boldsymbol{v}_j(\Lambda^*) \geqslant 0,\ \forall j \in \{1,2,\cdots,z\}; \\ \boldsymbol{v}_j(\Lambda^*) < 0,\ \forall j \in \{z+1,z+2,\cdots,n\}. \end{cases} \tag{8}$$

其中 $\boldsymbol{v}_j$ 表示排序后的第 j 个对象的LV值. 可见, $\boldsymbol{X}$ 中的前 z 个对象构成了LR问题的最优解 $X_L^*(\Lambda^*)$, 假设 $X_L^*(\Lambda^*)$ 的可信度为 $\gamma\,(0 \leqslant \gamma \leqslant 1)$, 即认为 $\boldsymbol{X}$ 中的前 $\lfloor z\gamma \rfloor$ 个不违反约束的对象(称为可信对象, 用集合 $\boldsymbol{T}$ 表示)肯定包含在原MKP的最优解中, 而后续 $\lceil z(1-\gamma) \rceil$ 个对象是否被包含则有待确定. 将这 $\lceil z(1-\gamma) \rceil$ 个难以确定的对象沿序列 $\boldsymbol{X}$ 扩展一倍, 将由此得到的 $2\lceil z(1-\gamma) \rceil$ 个对象构成的低维MKP定义为原MKP的核问题, 即

$$\max \sum_{j \in \boldsymbol{C}} \boldsymbol{p}_j \boldsymbol{x}_j; \tag{9}$$

$$\text{s.t.} \sum_{j \in \boldsymbol{C}} \boldsymbol{r}_{ij} \boldsymbol{x}_j \leqslant b_i - \tilde{r}_i,\ i = 1,2,\cdots,m, \tag{10}$$

$$\boldsymbol{x}_j \in \{0,1\},\ j \in \boldsymbol{C}. \tag{11}$$

其中: $\boldsymbol{C} = \{\lfloor z\gamma \rfloor + 1,\cdots,z + \lceil z(1-\gamma) \rceil\}$ 为核心对象集合, $\tilde{r}_i = \sum_{j=1}^{\lfloor z\gamma \rfloor} \boldsymbol{r}_{ij}$ 为由可信对象消耗的第 i 种资源量. 从核问题的定义可知, $\boldsymbol{X}$ 中的最后 $n-z-\lceil z(1-\gamma) \rceil$ 个对象被认为肯定不会出现在原MKP的最优解中, 这些对象称为不可信对象. 图1以规模为 100×5 的标准MKP算例为例, 绘制了排序后对象的LV变化曲线, 并对相应的核问题进行了说明.

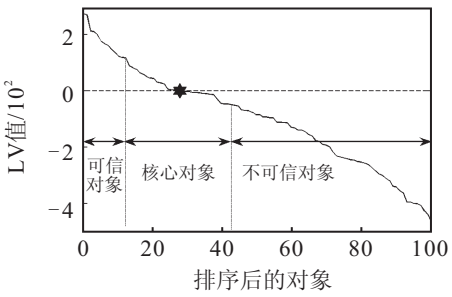

图 1 对象的LV曲线与核问题

2.2 ACO求解策略

2.2.1 ACO的基本思想

ACO是通过模拟自然界中蚁群的觅食行为而发展起来的一种智能优化算法[16], 目前已成功应用于许多组合优化问题[17]. 在采用ACO求解某问题时, 人工蚂蚁根据解的基本组成单元上所关联的信息素值和启发因子值逐步构造候选解. 当所有蚂蚁都生成一个完整的候选解后, ACO在较好解的组成单元上释放一定量的信息素, 以引导蚂蚁在后续搜索过程中重点探索相应的有希望的解区域, 最终找到理想解. 由此可知, 在为MKP设计ACO求解策略时, 需要确定信息素的关联方式、启发因子、候选解的构造规则和信息素的更新规则. 尽管AL-MKP算法利用ACO求解降维后的MKP核问题, 但为了方便描述, 后文仍以MKP原问题为例来说明ACO求解策略.

2.2.2 信息素关联方式与候选解构造规则

由式(1)～(3)可知, MKP是子集选择问题. 在对这类问题应用ACO时, 通常可将候选对象看作基本解单元, 并为每个对象 j 赋予一个信息素因子 τ_j $(\tau_j > 0)$ 和一个启发因子 $\eta_j\,(\eta_j > 0)$, 分别用来记录期望将对象 j 选入候选解的经验信息和先验信息[16].

在构造候选解时, 每只蚂蚁都从一个空解出发, 然后不断地根据下式所示的概率选择规则, 以概率 $P_t(j)$ 从可行候选对象集合 F_t 中选择一对象 j, 并添加到当前解中, 直到不存在可行候选对象:

$$P_t(j) = \begin{cases} \dfrac{\tau_j \eta_j^{\beta}}{\sum\limits_{i \in F_t} \tau_i \eta_i^{\beta}},\ j \in F_t; \\ 0,\ \text{otherwise}. \end{cases} \tag{12}$$

其中参数 $\beta\,(\beta > 0)$ 用来调节启发因子相对于信息素因子的作用力度.

2.2.3 启发因子

启发因子对ACO的性能具有重要影响. 由前文可知, 对象LV值 $v(\Lambda^*)$ 将对象价值与资源需求有机地集成在一起, 可以综合衡量对象的效用, 因此可以用作ACO的启发因子. 然而, 某些对象的LV值可能为负, 需要进行标称化处理, 最终确定的启发因子为

$$\eta_j = v_j(\Lambda^*) + \sigma, \tag{13}$$

其中 $\sigma\,(\sigma > 0)$ 为修正因子, 通常可设为

$$\sigma = 1.5|\min\{v_j(\Lambda^*)|j = 1,2,\cdots,n\}|.$$

2.2.4 信息素更新规则

每次迭代中, 当所有蚂蚁都生成一个完整的候选解后, 需要对信息素进行更新. 首先将所有信息素都均匀地衰减一部分, 以使蚂蚁忘记部分搜索经验; 然后对属于当前迭代最优解 X^{ib} 的对象奖励一些信息素, 以引导蚂蚁在后续迭代中重点搜索这些优解的邻域. 具体规则为

$$\tau_j \leftarrow \rho\tau_j + \Delta\tau_j; \tag{14}$$

$$\Delta\tau(j) = \begin{cases} \dfrac{1}{L(\Lambda^*) - P(X^{\text{ib}})},\ x_j^{\text{ib}} = 1; \\ 0,\ \text{otherwise}. \end{cases} \tag{15}$$

其中: 参数 $\rho\,(0 < \rho < 1)$ 为信息素留存系数, $\Delta\tau_j$ 为对象 j 上信息素的增加量. 如果对象 j 包含在解 X^{ib} 中, 则 $\Delta\tau_j$ 与差值 $[L(\Lambda^*) - P(X^{\text{ib}})]$ 成反比. $L(\Lambda^*)$ 是通过求解MKP的拉格朗日对偶问题而获得的最紧上

界, 因此$[L(\Lambda^*) - P(X^{\rm ib})]$恰当地反应了解$X^{\rm ib}$的质量. 该差值越小, 表明$X^{\rm ib}$的质量越高. 此外, 利用最大-最小蚂蚁系统[18]的思想, 对信息素的最小值进行限制, 以尽可能防止算法陷入局部优解, 即

$$\tau_j \leftarrow \max\{\tau_j, \tau_{\min}\}, \tag{16}$$

$$\tau_{\min} = \frac{\varepsilon}{(1-\rho)[L(\Lambda^*) - P(X^{\rm ib})]}, \tag{17}$$

其中参数ε $(0 < \varepsilon < 1)$为比例系数.

2.3 局部搜索策略

为进一步提高蚂蚁的搜索精度, 一种惯常做法是对蚂蚁构造的每个候选解进行局部搜索. AL-MKP 算法中所用局部搜索策略的伪代码如下, 其中心思想是在保证候选解可行的前提下, 通过采用高价值的对象替换低价值的对象来提高解质量.

输入: X, s, N_{ls};

输出: $isImproved$, X// $isImproved$记录是否有改进.

1) 设置$X' \leftarrow X$, $isImproved \leftarrow$ false, $l \leftarrow 0$;

2) 从X'中随机选择s个变量x_j, 并设$x_j \leftarrow \bar{x}_j$; //"翻转"操作

3) 计算$r_i = \sum_{j=1}^{n} r_{ij}x_j$, $i = 1, 2, \cdots, m$, 设$j \leftarrow n$;

4) For $i = 1, 2, \cdots, m$, do //"删除"操作

5)　While $r_i > b_i$, do

6)　　If $\bar{x}_j = 1$, then

7)　　$\bar{x}_j \leftarrow 0$, $r_i \leftarrow r_i - \bar{r}_{ij}$, $i = 1, 2, \cdots, m$;

8)　　$j \leftarrow j - 1$;

9) For $j = 1, 2, \cdots, n$, do//"增加"操作

10)　If $\bar{x}_j = 0$ and $r_i + \bar{r}_{ij} \leqslant b_i$, $i = 1, 2, \cdots, m$, then

11)　　设置$\bar{x}_i \leftarrow 1$, $r_i \leftarrow r_i + \bar{r}_{ij}$, $i = 1, 2, \cdots, m$;

12) If $P(X') \geqslant P(X)$, then //判断是否有改进

13)　设置$X \leftarrow X'$, $isImproved \leftarrow$ true

14) Else

15)　设置$X' \leftarrow X$;

16) 设置$l \leftarrow l + 1$;

17) If $l < N_{ls}$, then

18)　转至 2);

19) Else

20)　返回$isImproved$, X.

对于每个待处理的候选解, 首先执行对象"翻转"操作, 以扩展搜索空间; 然后执行"删除"操作, 以保证当前候选解的可行性; 最后执行"增加"操作, 以尽可能改进候选解的性能[1]. 2) 中的s表示对每个候选解"翻转"的位数, s值越大, 局部搜索的范围越大, 但计算量也随之增大. 17) 中的N_{ls}表示对每个候选解进行局部搜索的次数.

2.4 AL-MKP 算法的实现

AL-MKP 算法的具体实现步骤如下.

Step 1: 求解原 MKP 的拉格朗日对偶问题, 获得各对象的 LV 值.

Step 2: 设置可信度参数γ, 根据各对象的 LV 值构建 MKP 核问题.

Step 3: 设置 ACO 参数 (包括β、ρ、ε、蚂蚁个数$M'_{\rm ant}$和迭代次数$N'_{\rm ant}$) 和局部搜索参数 (s、N_{ls}).

Step 4: 初始化信息素因子, 计算启发因子.

Step 5: 每只蚂蚁由式 (12) 构建一个候选解.

Step 6: 根据 2.3 节伪代码对每个候选解进行局部搜索.

Step 7: 统计蚁群获得的当前迭代最优解$X^{\rm ib}$和全局最优解$X^{\rm gb}$.

Step 8: 利用$X^{\rm ib}$并由式 (14) ~ (17) 更新信息素.

Step 9: 判断是否达到预设的迭代次数或时间限制, 若达到, 则继续下一步, 否则转至 Step 5.

Step 10: 将获得的核问题的全局最优解$X^{\rm gb}$与原问题的可信对象集$\boldsymbol{T}$进行合并, 作为原问题的优解进行输出.

2.5 AL-MKP 算法的复杂度

对于$n \times m$维的 MKP 问题, 若相应的 LR 问题的最优解中包含的对象个数为z (一般情况下, $z \ll n$), 则 MKP 自身的可行解中平均包含的对象个数也近似为z. 蚂蚁根据式 (12) 构造这样一个可行解的平均计算时间复杂度为$O(mnz)$. 如果直接采用 ACO 求解原 MKP 问题, 则计算复杂度为$O(N_{\rm ant}M_{\rm ant}mnz)$. 由第 2.1 节可知, 根据本文所提出方法构造出的 MKP 核问题的维数为$2z(1-\gamma) \times m$, 那么 ACO 的求解复杂度变为

$$O(2N'_{\rm ant}M'_{\rm ant}mz(1-\gamma)z') = O(N'_{\rm ant}M'_{\rm ant}mzz'),$$

其中z'表示核问题的可行解中平均包含的对象个数, 满足$z' \ll z$. 随着问题规模的显著下降, 所需投入的蚂蚁个数和迭代次数也大幅减少, 即$M'_{\rm ant} < M_{\rm ant}$, $N'_{\rm ant} < N_{\rm ant}$. 综合来看, 在提取核问题后, 算法的总体计算量将大大减小.

3 实验与分析

为了评估 AL-MKP 算法的性能, 采用两组标准算例对其进行测试. 第 1 组算例来自 ORLIB 库[1], 命名规则为 mkp*n.m-xy*, 其中*xy*表示具有相同维数的算例的标号. 第 2 组算例来自 http://hces.bus.olemiss.edu/tools.html, 命名规则为 Mk_gk*n.m-xy*, 其中包含了 3 个

对象个数大于1 000的大规模算例. 实验采用的AL-MKP算法代码是在Visual C++ 6.0集成环境中编写的, 运行环境为Windows XP操作系统、Intel i5-2300处理器(2.8 GHz). 除特别说明之外, 在实验中, 对每个算例均独立测试25次, 每次测试允许的最大迭代次数为2 000.

3.1 参数设置及其影响

AL-MKP算法涉及的主要参数包括: 启发因子的重要度指数β、信息素留存系数ρ、信息素下界的比例系数ε、蚂蚁个数$M'_{\rm ant}$、局部搜索策略中的"翻转"位数s、局部搜索次数N_{ls}、LR问题的最优解的可信度γ. 对于ACO的有关参数, 目前还没有系统性的设置方法, 因此选择文献中常用的实验方法确定这些参数[16,18], 最终设置为

$$\beta = 5.0,\ \rho = 0.99,\ \varepsilon = 0.005$$

$$M'_{\rm ant} = 50,\ s = 4,\ N_{ls} = 50.$$

本文重点考察可信度γ对AL-MKP的性能影响. 由γ的物理意义可知, γ值越大, 核问题的规模越小, AL-MKP算法的求解效率越高. 但这样可能导致将原本不属于最优解的对象强制包含在最优解中, 从而降低求解质量. 图2和图3以算例mkp500.5-00为例, 给出了当γ由0变化到1时, AL-MKP算法性能的变化情况.

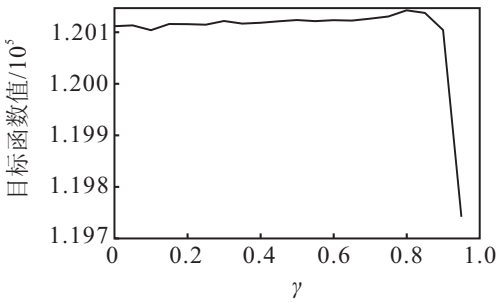

图2 AL-MKP的求解质量随γ的变化情况

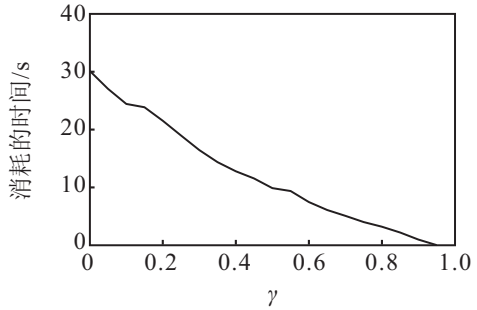

图3 AL-MKP的求解效率随γ的变化情况

由图2可见: 随着γ值的增大, 算法的求解质量逐渐变好; 但当γ超过某一阈值后, 求解质量急剧下降. 由图3可见, AL-MKP算法所消耗的计算时间随着γ的增大而减少. 这些观测结果与前面的定性分析是一致的.

根据大量算例上的测试结果给出γ的经验设置规则

$$\gamma = \min\left\{0.8, 0.4 + {\rm sgn}(m < 25) \cdot \frac{n}{50} \cdot 0.05\right\}.$$

当约束个数较少($m < 25$)时, γ值随对象个数的增多而增大, 最大不超过0.8; 当约束个数较多($m \geqslant 25$)时, 取$\gamma = 0.4$. 尽管该规则并不是最优选择, 但适用于绝大多数测试算例.

3.2 与其他算法的比较

为了综合衡量AL-MKP算法的性能, 将其与两种已有算法进行比较:

一种是NP+BAS+LP算法[10], 该算法将ACO与嵌套分割算法(NP)、线性规划(LP)松弛方法相结合, 综合利用了ACO的快速寻优能力、NP的全局优化能力和LP提供的上界信息, 是目前求解MKP的性能最优的ACO算法之一;

另一种是bFOA算法[7], 该算法将二进制果蝇优化算法与局部搜索策略相结合, 是新近提出的一种性能优越的MKP求解算法.

表1和表2分别给出了AL-MKP与NP+BAS+LP、bFOA算法的比较结果. 两种已有算法的求解结果直接摘自相应的文献. 对于AL-MKP算法, 除了相应的解性能指标外, 还给出了核问题的维数、首次获得最终优解的平均时间和完成所有迭代消耗的平均时间.

由表1可见, 对于被测的20个算例, AL-MKP算法在每次测试中均可以获得其中12个算例的最优解. 从平均解性能来看, 该算法在5个算例上的表现优于NP+BAS+LP算法, 在11个算例上的表现与后者相当. 尽管实现两个算法的硬件平台略有不同, 但从"完成时间"项的巨大差异可以得知, AL-MKP算法的求解效率高于NP+BAS+LP算法. 此外, AL-MKP算法的"首达时间"远小于"完成时间", 由此可知, AL-MKP算法获得同等解质量所需要的总体迭代次数可以进一步减少.

由表2可见, 对于被测的11个算例, AL-MKP算法在其中8个算例上的表现优于bFOA算法. 特别地, 对于算例Mk_gk*150.25*-03, AL-MKP算法可以获得优于已知最优解(5 650)的新解(5 651). 从"标准差"项来看, AL-MKP算法在不同规模问题上的表现较为一致; 除了算例Mk_gk*100.15*-01外, 相应的标准差均小于bFOA算法. 这些结果表明, AL-MKP算法具有较强的鲁棒性.

表 1　AL-MKP 与 NP+BAS+LP 的比较

算 例	已知最优解	NP+BAS+LP			AL-MKP					
		最优解	平均解	完成时间/s	问题维数	最优解	平均解	标准差	首达时间/s	完成时间/s
mkp*100.5*-00	24 381	24 381	24 381.0	15.5	32×5	**24 381**	**24 381.0**	0.000 0	0.004 4	0.966 8
mkp*100.5*-02	23 551	23 551	23 551.0	14.7	32×5	**23 551**	**23 551.0**	0.000 0	0.022 6	1.039 4
mkp*100.5*-04	23 991	23 991	23 984.6	16.9	36×5	***23 991***	***23 991.0***	0.000 0	0.036 9	1.226 9
mkp*100.5*-06	25 591	25 591	25 591.0	10.9	38×5	**25 591**	**25 591.0**	0.000 0	0.004 9	1.276 2
mkp*100.5*-08	24 216	24 216	24 216.0	12.4	32×5	**24 216**	**24 216.0**	0.000 0	0.068 1	0.906 9
mkp*100.5*-10	42 757	42 757	42 757.0	13.0	56×5	**42 757**	**42 757.0**	0.000 0	0.051 8	2.551 3
mkp*100.5*-12	41 968	41 968	41 961.9	16.2	60×5	***41 968***	***41 967.1***	0.276 9	0.344 3	2.773 7
mkp*100.5*-14	42 218	42 218	42 218.0	13.0	60×5	**42 218**	**42 218.0**	0.000 0	0.006 8	2.681 2
mkp*100.5*-16	42 009	42 009	42 009.0	15.7	62×5	**42 009**	**42 009.0**	0.000 0	0.004 4	2.876 5
mkp*100.5*-18	43 441	43 441	43 441.0	13.0	58×5	**43 441**	**43 441.0**	0.000 0	0.014 4	2.660 1
mkp*500.5*-00	120 148	120 148	120 147.3	–	58×5	***120 148***	***120 148.0***	0.000 0	1.556 3	3.460 6
mkp*500.5*-02	121 131	121 131	121 130.0	–	62×5	121 131	121 128.8	3.555 3	0.651 9	2.810 1
mkp*500.5*-04	122 319	122 319	122 319.0	–	58×5	**122 319**	**122 319.0**	0.000 0	0.052 0	3.153 2
mkp*500.5*-06	119 127	119 127	119 120.2	–	60×5	**119 127**	**119 120.2**	4.023 7	1.335 6	3.441 3
mkp*500.5*-08	121 575	121 575	121 571.8	–	58×5	***121 575***	***121 574.4***	0.916 5	1.446 2	3.437 5
mkp*500.5*-10	218 428	218 428	218 426.1	–	110×5	218 428	218 422.8	1.854 7	2.243 1	5.8130
mkp*500.5*-12	217 534	217 534	217 534.0	–	104×5	217 534	217 529.3	2.427 6	0.817 6	5.501 9
mkp*500.5*-14	218 966	218 966	218 962.2	–	106×5	***218 966***	***218 962.4***	1.227 5	0.166 7	5.712 5
mkp*500.5*-16	219 989	219 989	219 989.0	–	132×5	219 989	219 987.1	0.400 0	0.476 1	5.935 0
mkp*500.5*-18	216 976	216 976	216 976.0	–	104×5	**216 976**	**216 976.0**	0.000 0	1.305 2	5.631 2

表 2　AL-MKP 与 bFOA的比较

算 例	已知最优解	bFOA			AL-MKP					
		最小偏差	平均偏差	标准差	问题维数	最小偏差	平均偏差	标准差	首达时间/s	完成时间/s
Mk_gk*100.15*-01	3 766	0.239 0	0.305 3	0.033 2	56×15	***0.000 0***	***0.076 5***	0.044 9	1.099 4	3.409 4
Mk_gk*100.25*-02	3 958	0.000 0	0.198 3	0.230 6	60×25	***0.000 0***	***0.000 0***	0.000 0	0.502 4	3.606 2
Mk_gk*150.25*-03	5 650	0.070 8	0.158 4	0.094 6	82×25	***−0.017 7***	***0.029 0***	0.017 6	1.810 0	5.584 0
Mk_gk*150.50*-04	5 764	0.052 0	0.194 3	0.356 7	78×50	***0.052 0***	***0.143 0***	0.023 6	2.590 5	7.633 7
Mk_gk*200.25*-05	7 557	0.052 9	0.119 8	0.078 7	98×25	***0.026 5***	***0.070 9***	0.029 8	3.096 4	7.091 2
Mk_gk*200.50*-06	7 672	0.117 3	0.204 6	0.121 4	84×50	0.156 4	0.231 0	0.027 3	5.075 8	9.410 6
Mk_gk*500.25*-07	19 215	0.052 0	0.082 2	0.047 2	306×25	0.119 7	0.139 7	0.007 9	27.205 8	35.733 8
Mk_gk*500.50*-08	18 801	0.085 1	0.140 2	0.126 7	324×50	0.446 8	0.475 1	0.013 0	24.861 8	43.276 8
Mk_gk*1 500.25*-09	58 085	2.174 4	2.281 6	0.926 7	944×25	***0.086 1***	***0.125 8***	0.009 3	159.527 5	250.701 3
Mk_gk*1 500.50*-10	57 292	1.743 7	1.790 5	0.414 8	956×50	***0.315 9***	***0.360 7***	0.012 9	167.815 0	274.446 8
Mk_gk*2 500.100*-11	95 231	1.503 7	1.573 8	0.578 0	1 520×100	***0.409 5***	***0.419 9***	0.005 8	525.728 8	765.640 6

4　结　论

本文针对MKP问题NP-hard、约束强的特点, 提出了一种高效的蚁群-拉格朗日松弛混合优化算法. 首先根据拉格朗日松弛对偶信息, 提出一种MKP效用指标; 在此基础上, 给出了MKP核问题的定义. 核问题的引入大大减小了问题规模, 从而缩小了蚁群的搜索空间; 所提出的效用指标可以间接用作蚁群的启发因子, 引导算法在有希望的解区域中强化搜索; 所设计的局部搜索策略进一步增强了算法的精细搜索能力. 通过在大量中高规模算例上的测试, 验证了所提出算法在求解质量和求解效率方面的优势.

参考文献(**References**)

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

[2] Kellerer H, Pferschy U, Pisinger D. Knapsack problems[M]. Berlin: Springer, 2004: 449-482.

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

[5] Vasquez M, Vimont Y. Improved results on the 0-1 multidimensional knapsack problem[J]. European J of Operational Research, 2005, 165(1): 70-81.

[6] 王凌, 王圣尧, 方晨. 一种求解多维背包问题的混合分布估计算法[J]. 控制与决策, 2011, 26(8): 1121-1125.
(Wang L, Wang S Y, Fang C. A hybird distribution estimation algorithm for solving multidimensional knapsack problem[J]. Control and Decision, 2011, 26(8): 1121-1125.)

[7] Wang L, Zheng X L, Wang S Y. A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem[J]. Knowledge-Based Systems, 2013, 48(1): 17-23.

[8] 欧阳海滨, 高立群, 孔祥勇, 等. 一种求解0-1背包问题的二进制修正和声搜索算法[J]. 控制与决策, 2014, 29(7):

1174-1180.
(OuYang H B, Gao L Q, Kong X Y, et al. A binary modified harmony search algorithm for 0-1 knapsack problem[J]. Control and Decision, 2014, 29(7): 1174-1180.)

[9] 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, 31(5): 810-819.
(Yu X C, Zhang T W. An improved ant algorithm for multidimensional knapsack problem[J]. Chinese J of Computers, 2008, 31(5): 810-819.)

[10] Al-Shihabi S, Ólafssonb S. A hybrid of nested partition, binary ant system, and linear programming for the multidimensional knapsack problem[J]. Computers & Operations Research, 2010, 37(2): 247-255.

[11] Puchinger J, Raidl G R, Pferschy U. The multidimensional knapsack problem: Structure and algorithms[J] Informs J on Computing, 2010, 22(2): 250-265.

[12] Martins J P, Fonseca C M, Delbem A C B. On the performance of linkage-tree genetic algorithms for the multidimensional knapsack problem[J]. Neurocomputing, 2014, 146(1): 17-29.

[13] Wolsey L A. Integer Programming[M]. New York: Wiley-Interscience, 1998: 167-181.

[14] Fisher M L. The Lagrangian relaxation method for solving integer programming problems[J]. Management Science, 2004, 50(12): 1861-1871.

[15] Pisinger, D. Core problems in knapsack algorithms[J]. Operations Research, 1999, 47(4): 570-575.

[16] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.

[17] 张晓霞, 唐立新. 一种新的求解MMKP问题的ACO&PR算法[J]. 控制与决策, 2009, 24(5): 729-733.
(Zhang X X, Tang L X. A new ACO&PR algorithm for multiple-choice multidimensional knapsack problem[J]. Control and Decision, 2009, 24(5): 729-733.)

[18] Stützle T, Hoos H H. Max-min ant system[J]. Future Generation Computer Systems, 2000, 16(8): 889-914.

(责任编辑：郑晓蕾)