Journal of System Simulation

Volume 29 | Issue 12 Article 25

6-6-2020

Distributed Estimation Algorithm for Multi-dimensional Multi-choice Knapsack Problem

Tan Yang
1. College of Computer Science, Human Normal University, Changsha 410081, China;;2. Human Radio and Television University, Changsha 410004, China;

Liu Zhang
2. Human Radio and Television University, Changsha 410004, China;

Zhou Hong
2. Human Radio and Television University, Changsha 410004, China;

Distributed Estimation Algorithm for Multi-dimensional Multi-choice Knapsack Problem

Abstract

Abstract: As it is difficult to realize local optimization of the Multidimensional Multiple-choice Knapsack Problem (MMKP), the Estimation of Distribution Algorithms (EDA) is applied to optimize the MMKP. *In order to improve the local optimization ability of EDA, value weight factors of items for selection are built to improve the EDA initial model and probabilistic model updating methods. The impact of the extreme effects on the algorithm optimization process is balanced to overcome the defect that the local optimization ability of the traditional EDA is weak. A new non-feasible solution repair mechanism is adopted to maintain the facilitation of machine learning methods for the probabilistic model and improve the global optimization ability of the improved algorithm.* Experimental results show that this algorithm can effectively optimize the MMKP and its performance is much better than traditional optimization algorithms.

Keywords

heuristics, multidimensional multiple-choice knapsack, value weight, estimation of distribution, optimality

Recommended Citation

Tan Yang, Liu Zhang, Zhou Hong. Distributed Estimation Algorithm for Multi-dimensional Multi-choice Knapsack Problem[J]. Journal of System Simulation, 2017, 29(12): 3123-3131.

第 29 卷第 12 期 系统仿真学报© Vol. 29 No. 12
2017 年 12 月 Journal of System Simulation Dec., 2017

一种求解多维多选择背包问题的分布估计算法

谭阳[1,2]，刘章[2]，周虹[2]
(1. 湖南师范大学计算机科学学院，长沙 410081；2. 湖南广播电视大学，长沙 410004)

摘要：针对多维多选择背包问题(MMKP)局部难以优化的特点，提出将分布估计算法(EDA)应用于优化 MMKP 问题。*为了提升 EDA 优化局部的能力，以构建待选物品价值权重因子的方式来改进 EDA 的初始模型和概率模型更新方法；并平衡了极值效应对算法寻优过程的影响，克服了传统 EDA 局部优化能力不强的缺陷.同时采用新的非可行解的修复机制，维护了机器学习法对概率模型的促进作用，提高了改进算法的全局优化能力。*实验结果表明,该算法能够有效地优化 MMKP 问题，其性能高于传统的优化算法。

关键词：启发式；多维多选择背包；价值权重；分布估计；优化
中图分类号：TP391 文献标识码：A 文章编号：1004-731X (2017) 12-3123-09
DOI: 10.16182/j.issn1004731x.joss.201712025

Distributed Estimation Algorithm for Multi-dimensional Multi-choice Knapsack Problem

Tan Yang[1,2], *Liu Zhang*[2], *Zhou Hong*[2]
(1. College of Computer Science, Human Normal University, Changsha 410081, China;
2. Human Radio and Television University, Changsha 410004, China.)

Abstract: As it is difficult to realize local optimization of the Multidimensional Multiple-choice Knapsack Problem (MMKP), the Estimation of Distribution Algorithms (EDA) is applied to optimize the MMKP. *In order to improve the local optimization ability of EDA, value weight factors of items for selection are built to improve the EDA initial model and probabilistic model updating methods. The impact of the extreme effects on the algorithm optimization process is balanced to overcome the defect that the local optimization ability of the traditional EDA is weak. A new non-feasible solution repair mechanism is adopted to maintain the facilitation of machine learning methods for the probabilistic model and improve the global optimization ability of the improved algorithm.* Experimental results show that this algorithm can effectively optimize the MMKP and its performance is much better than traditional optimization algorithms.

Keywords: heuristics; multidimensional multiple-choice knapsack; value weight; estimation of distribution; optimality

引言

多维多选择背包问题(multidimensional multi-choice knapsack problem，MMKP)是经典的 NP 难题问题，但是因其在货物装载、资源分配、材料分割、项目选择、分布式计算资源调度等方面具有广泛的实际应用价值[1-2]，所以一直是具有重要工程意义的研究方向之一。早期求解 MMKP 的方法主要采用精确求解法如动态规划、递归法、回溯法、分支限界法等，但是此类方法对于大规模 MMKP 而言，其计算量和存储量是难以承受的。近年来随

收稿日期：2015-11-16 修回日期：2016-03-25;
基金项目：国家自然科学基金(10971060)，湖南省教育厅重点项目(10A074)，湖南省高校科研项目(14C0781，15C0928)；
作者简介：谭阳(1979-)，男，湖南望城，硕士，副教授，研究方向为智能计算、数据挖掘。

着智能优化算法的提出，在求解 MMKP 上为人们提供了新的思路，如 Manicassamy 等在文献[3]中提出采用遗传算法的方式来对 MMKP 进行求解，并在此基础上通过抑制冲突基因位和环境诱导基因表达的方式来提高算法的收敛速度，以此获得了一种自适应调节优化 MMKP 的有效方法。文献[4]中 Chen 等人则是通过减小问题维度的线性规划的方式来优化 MMKP 问题，通过设立一组固定和一组可变的规则来获取给定问题的线性松弛信息，在对中小型 MMKP 的优化上取得了极快的计算速度，并且只需要很小的计算空间。文献[5]中 Khalili 等人首先通过采用模糊的方法将 MMKP 的约束分为收益和开销两个基本项，再通过简单的绑定枚举和目标规划的方式来解决 MMKP。由此可见具备启发性质的人工智能算法在求解 MMKP 时，不仅能有效的避免算法陷入局部最优解还能显著提高求解的速度。

分布估计算法(Estimation of Distribution Algorithm，EDA)是一种基于统计学原理的随机优化算法，最初由 Baluja 在 1994 年所提出[6]。标准 EDA 是通过概率模型及其更新来描述空间中解的分布情况，并对宏观解空间进行全局优化来引导种群的进化。相较其他类型的人工智能算法而言，标准 EDA 表现出较强的整体的搜索性能，但对于问题的局部优化能力有限，特别是在约束条件较多的情况下表现更为明显。因此，在现有的国内外研究文献中多见于将 EDA 应用于 0-1 背包问题及多维背包问题等约束条件较少的优化问题上；尚未发现将 EDA 应用于 MMKP 优化的研究。

为了改善 EDA 的局部优化性能，使之能有效的对 MMKP 问题进行优化，本文通过将 MMKP 的约束条件映射至待选物品之上，并将其视为待选物品的基本特性。其次将待选物品的基本特性归纳为“利润”与“成本”两项，通过计算物品利润与成本的比值以生成物品的价值权重，并以此为基础提出一种以物品价值权重值为因子的分布估计算法(EDA/w)。在对 MMKP 的优化过程中以物品的价值权重因子作为先验描述和 EDA/w 优化的引导，并且对 EDA/w 在寻优过程中所获得的非可行解进行修复，将修复后的非可行解与可行解一同映射至概率分布模型之中，以此提高 EDA/w 对 MMKP 的寻优精度和速度，克服了标准 EDA 因局部寻优能力不足不能有效优化 MMKP 的问题。

1 基本理论

1.1 对 MMKP 的数学描述

在 MMKP 中不仅需要考虑多种资源的约束条件，而且还需考虑物品选择条件限制的问题。不失一般性，MMKP 的数学模型如下：给定 m 种资源，并将需要放入背包中的物品分为互斥的 n 类，且类的集合为 $I=\{1,\cdots,n\}$，物品集合为 $J=[J_1,\cdots J_i\cdots,J_n]$，其中每类物品 $J_i(i\in I)$，$J_i=\{1,\cdots,n_i\}$中有 n_i 个不同的物品。

$$(\text{MMKP})\ Max \quad Z=\sum_{i=1}^{n}\sum_{j=1}^{n_i}c_{ij}x_{ij}$$

$$s.t.\sum_{i=1}^{n}\sum_{j=1}^{n_i}w_{ij}^{k}x_{ij}\leqslant C^{k},\quad \forall k=1,\cdots,m \tag{1}$$

$$\sum_{j=1}^{n_i}x_{ij}=1,\quad \forall i=1,\cdots,n \tag{2}$$

$$x_{ij}\in\{0,1\},\quad \forall i=1,\cdots,n\quad j=1,\cdots,n_i \tag{3}$$

式中：C^k 为资源 k 的最大值;x_{ij} 表示第 i 类物品中的第 j 个物品，若该物品被选取则 x_{ij}=1。c_{ij} 为第 J_i 类物品 $j(j\in J_i)$的价值(利润)，其所需消耗的资源(成本)k 为 $w_{ij}^{k}(k\in M,M=\{1,\cdots,m\})$。MMKP 需要在每类物品 $i(i\in I)$中选择一个物品 $j(j\in J_i)$放入相应的背包之中，并且满足该背包的资源约束，同时还要使得物品总收益最大。为了映射出每类物品与所选择物品在不同背包中的对应关系，所有可行解的表达均为一个 $m\times\sum_{i=1}^{n}J_i$ 阶的 0-1 矩阵。例如：一个物品种类为 4，每类物品数量为 3，背包数目为 2 的 MMKP 的解如图 1 所示。

DOI: 10.16182/j.issn1004731x.joss.201712025

M \ n	← J_1 →			← J_2 →			← J_3 →			← J_4 →		
	1	2	3	1	2	3	1	2	3	1	2	3
m_1	0	1	0	0	0	1	1	0	0	0	1	0
m_2	1	0	0	0	1	0	0	1	0	0	0	1

图 1 MMKP 问题的 0-1 矩阵解
Fig.1 0-1 matrix for MMKP problem

1.2 标准 EDA 算法

分布估计算法首先需要构造一个描述解空间的概率模型，之后再通过对种群的评估，从中选择优秀的个体集合，然后采用统计概率的机器学习方法根据优秀个体的分布情况来构造概率模型；最后由这个概率模型随机采样产生新的种群，如此反复迭代，从而实现种群的进化，直至满足终止条件。

1.3 对标准 EDA 应用于 MMKP 的分析

为了能够充分了解标准 EDA 对于 MMKP 问题优化的特点，并针对标准 EDA 的不足进行改进。因此，需要构造一个简单的应用于 MMKP 的标准 EDA。不失一般性，若一个物品种类为 n，背包数为 m 的 MMKP，其所有物品总数为：N，$\left(N=\sum_{i=1}^{n} J_i \right)$ 依据标准 EDA 的算法需要建立规模为 $m\times N$ 的概率矩阵 $p(x)$。

$$p(x)=\begin{bmatrix} p(x_{1,1}) & p(x_{1,2}) & \cdots & p(x_{1,N}) \\ p(x_{2,1}) & p(x_{2,2}) & \cdots & p(x_{2,N}) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_{m,1}) & p(x_{m,2}) & \cdots & p(x_{m,N}) \end{bmatrix} \tag{4}$$

以矩阵 $p(x)$描述解空间中的概率分布模型，其中 $p(x_{i,j})$表示第 i 个背包选择第 j 个物品的概率。以 PBIL 机器学习法则[7]为例，通常会将初始概率 $p(x)$ 设置为以 0.5 为基准的均匀分布状态；随着算法的迭代，新个体的产生则由概率矩阵 $p(x)$的分布情况来产生，若给定 EDA 的种群规模为 S，PBIL 则通过计算所有个体的适应值，并从中选择适应值最高的前 s 个个体 $s<S$，以下式(5)来更新概率矩阵 $p(x)$。

$$p_{t+1}(x_{i,j})=(1-\alpha)p_t(x_{i,j})+\alpha\frac{1}{s}\sum_{l=1}^{s}x_{i,j}^{t} \tag{5}$$

式中：t 为算法当前进化的代数；$p_t(x)$为第 t 代时的概率；α，$(0<\alpha<1)$为机器学习的速率；$x_{i,j}^{t}$ 为第 t 代时第 i 个背包中选择的第 j 个物品。由上式(5)可以看出标准 EDA 主要是通过“概率分布模型 O”和“种群 P”这两类数据的协同作用来引导其进行搜索；O 是对上一代 P 进行的统计推导的结果，而 P 又是通过对本代 O 进行随机模拟而获得的。O 主要用来记忆问题变量的概率分布情况，而 P 则是通过引入竞争、演化机制来对 O 进行不断地修正。我们知道 MMKP 其实是背包问题的一种特殊情况，标准 EDA 作为一种基于完全随机过程的启发式算法，在优化 MMKP 的过程中所获得的结果会受多种条件的约束，因此产生有效解的概率过低，从而割裂了模型 O 与种群 P 之间的联系，使得概率分布模型陷入僵化，导致标准 EDA 对 MMKP 的优化效率低下。

2 结合价值权重因子的分布估计算法

为了提高人工智能算法的优化效率，Osorio 等人在文献[8]中指出通过单纯形的方法求得松弛解，并以此为基础往往能够找到更好的近似解甚至是精确解。因此，以简单的可行松弛解来作为优化导向，在加以区分的变量区域内进行搜索则能够提高优化算法对 MMKP 的优化效率。

2.1 价值权重因子

通过对 MMKP 问题的分析，发现可以将其约束条件归纳为“利润”与“成本”两大类别。若以“利润”和“成本”作为主要的条件来约束，那么在其他约束条件允许的范围内选择“利润”和“成本”比值最高的待选物品，即可快速获得 MMKP 的可行松弛解。物品的“利润”和“成本”比值(价值权重)可以通过物品的利润与资源消耗比 c_{ij}/w_{ij} 来度量，c_{ij}/w_{ij} 值越大则表明该物品的权重值越高。因此，以 MMKP 问题中的类型 $J_i(i\in I)$作为区分，划分所有待选的物品，并以下式(6)来计算所有不同类型中物品的价值权重，即获得该物品价值权重的因子值。

$$\beta_{ij}=\frac{(c_{ij}/w_{ij})}{\sum_{j=1}^{n_i}(c_{ij}/w_{ij})} \tag{6}$$

由式(6)可知，物品的价值权重 c_{ij}/w_{ij} 与其权重因子β_{ij}，$(0<\beta_{ij}<1)$值成正比。显然，在同等约束条件下选择β值较大的物品，所能获得的直观收益也较大。

2.2 对非可行解的修复

标准 EDA 在优化 MMKP 过程中仅通过自身的概率模型来生成新个体，这对于约束条件众多的 MMKP 问题来说获得可行解的概率过低，从而导致优化效率不高。但是，优化过程中所有产生的新个体都是对 MMKP 概率分布情况的映射；因此，非可行解也同样具备更新、修正分布模型的价值，为了能保证所有产生的新解均为可行解，需要建立非可行解的修复机制。通过对 MMKP 解的结构(图 1 所示)的分析，修复非可行解应以类 $J_i(i\in I)$为单位先行修复，之后再考虑不同背包 m_k, $(k=1,\cdots,m)$的资源约束问题。

2.2.1 “类型”修复算子

若修复非可行解的任意类 J_i(如图 2 所示)，首先需满足问题的约束：描述类 J_i 的矩阵 $m\times n_i$ 中的任意列只能存在 1 个被选项，且还需满足该矩阵中的“1”的总量小于等于 m。

n

	J_1	J_2	…	← J_i →				…	J_n
M	…	…	…	1	2	…	n_i	…	…
m_1	…	…	…	0	1	…	0	…	…
m_2	…	…	…	1	0	…	0	…	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m_m	…	…	…	0	0	…	1	…	…

图 2 对非可行解的类 J_i 的修复

Fig.2 Infeasible solutions for type J_i restoration

依次扫描 $m\times n_i$ 矩阵中的列，若某列中“1”的数量大于 1，则随机保留 1 个，其余置“0”；重复这一过程直至完成对所有列的扫描。统计该类矩阵中所有“1”的数量，若大于 m，则将矩阵中β值最小位置的“1”置“0”；重复这一过程直至“1”的数目等于 m。即完成对于类 J_i 的可行修复，并重复这一过程直至完成该解的所有类的可行修复。

2.2.2 “维度”修复算子

“维度”修复，是对单一背包进行可行修复。首先依次检查所有背包是否满足资源约束条件，若有第 k 个背包 m_k 不满足，则寻找该背包中β值最小的物品将其置“0”；重复这一过程直至背包 m_k 满足资源约束条件。以上述方法修复所有非可行背包，即完成一个非可行解的修复过程。

2.2.3 非可行解的修复流程

Step 1：以问题的约束条件来判断个体(解)是否为非可行解，若是则进行下一步，否则转 Step 4；

Step 2：扫描该非可行解的所有类(1~n)构成的矩阵，以“类型”修复算子进行修复；

Step 3：扫描该非可行解的所有背包(1~m)，若资源约束条件不满足，则以“维度”修复算子进行修复；

Step 4：判断是否满足终止条件，满足则终止修复程序；否则继续下一个个体(解)。

2.3 结合价值权重因子的分布估计算法(EDA/w)

文献[8]中指出标准 EDA 的效率对具体问题的概率分布模型 O_t 较为敏感。因此，设想先对待求解的 MMKP 快速地求解，以获得一个可行松弛解，并以该松弛解为基础建立初步的先验模型。该先验模型的作用是引导 EDA/w 的搜索方向以提高搜索效率。由前文可知，在条件约束范围内，通过不断选择当前β_{ij} 值最大的物品可以快速地获取当前待解 MMKP 的一个松弛可行解，该解虽粗糙但却是对待解 MMKP 问题的可行描述，并且可作为 EDA/w 优化的导向。因此，在求解 MMKP 的概率分布模型中导入价值权重因子，同时再将概率分布模型更新的方式由标准 EDA 的完全随机过程调整为具备指向性的随机过程，那么就使得 EDA/w 对

MMKP 的优化过程更具目的性，以提升 EDA/w 对待解 MMKP 的优化性能和寻优速度。因此，本文以物品的价值权重值β_{ij}作为 EDA/w 的分布模型的初始概率 $p_0(x)$，如下式(7)所示。

$$p_0(x)=\begin{bmatrix}\beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,N}\\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,N}\\ \vdots & \vdots & \ddots & \vdots\\ \beta_{m,1} & \beta_{m,2} & \cdots & \beta_{m,N}\end{bmatrix} \tag{7}$$

同时针对标准 EDA 局部搜索能力不足的问题，本文以下式(8)作为 EDA/w 概率分布模型的更新方法，其主要目的是：在确保优化算法能够对全局进行有效搜索的前提下，还能平衡不同个体间的竞争压力。

$$p_{t+1}(x_{i,j})=(1-\alpha)(1-\beta_{i,j})p_t(x_{i,j})+\alpha\frac{1}{s}\sum_{l=1}^{s}x_{i,j}^{t} \tag{8}$$

由式(8)可知，物品的价值权重因子值越高反而降低了其对概率模型更新的影响，以此减小由超级个体带来的极值效应，从而平衡个体间的差异，减少早熟收敛现象的发生。

2.4 EDA/w 的算法流程

综上所述，结合价值权重因子的分布估计算法流程如下：

Step 1：以式(4)计算所有物品的价值权重因子β_{ij}，并以此初始化概率向量 $p_0(x)$，t=0；

Step 2：以 $p_0(x)$进行随机采样生成 S 个个体，并转 Step 4；

Step 3：以 $p_0(x)$进行随机采样，生成 S–s 个个体；

Step 4：检测所有新生成个体，并对非可行个体进行修复；

Step 5：计算新个体的适应值，并依据适应值的大小对所有解进行降序排序，同时划分出的前 s 个最优个体；

Step 6：依据最优的 s 个个体的分布情况，以式(8)来更新向量概率 $p_t(x)$，t=t+1；

Step 7：判断算法是否符合终止条件，符合则算法结束并输出结果；否则转 Step 3。

通过在分布估计算法中引入价值权重因子可以较好的在算法初期凸显“性价比”较高个体，而保留机制使得在算法迭代过程中产生的优秀个体得以保留，从而保证算法不会发生退化。同时为了防止超级个体引发极值效应，在机器学习的环节中通过价值权重因子自身来平衡个体间的竞争关系，以使算法的搜索性能更为均衡。

3 实验仿真与分析

为了更好分析 EDA/w 的性能，本文引入了 EDA+，EDA+与 EDA 的区别在 EDA 基础上增加了本文所提出的对非可行解进行修复的流程。实验的测试平台为：Core i7 1.8GHz 的 CPU 和 4GB 的 RAM，系统为 Ubuntu Linux v12.04。测试计算中的实例来源于国际标准测试集 MMKPLIB[9]实例库中的经典算例。其中 Best 为算法在优化过程中找到的最优值，AVG 为算法优化过程的平均值，Tb 为首次搜索到算法最佳值所需要的最小迭代次数。为了保证一致性，EDA、EDA+和 EDA/w 均采用 C 语言编程实现。三种测试算法对实例 I01～I13 分别用 0 到 100 作为随机数种子独立优化 100 次，以获得的最优解来考察算法的优化质量，以每次寻优最佳的平均值来考察算法的鲁棒性，并记录下最优解所需的迭代次数。其中，测试算法的种群规模为 S=300，优势种群大小为 s=30，学习速率 α=0.05，最大迭代次数 t_{Max}=1 000。下表 1 为 EDA、EDA+和 EDA/w 对实例 I01~I13 的优化结果。

可以看出，通过加入非可行解修复算子后 EDA+相较 EDA 在寻优结果上有较大的提升，这表明 EDA+能够将算法迭代过程所获得的结果(可行与非可行)都映射至自身的概率模型中，提升了寻优的精度。通过加入价值权重因子，使得 EDA/w 在优化过程具备了更好的指向性，同时，价值权重因子还能够对 EDA/w 优化过程所产生的优势个体进行平衡，已避免算法陷入局部最优。表 1 中 EDA/w 较其他对比算法在最优解和平均优化值上都取得了更好的优化结果，在实例 I01～I13 的最优解上较 EDA 的平均改进幅度为 8.5%。同时，在

算法鲁棒性的指标(AVG)上 EDA/w 也有较大程度的提升，这也表明 EDA/w 对不同 MMKP 的敏感度更低，适应性更好。在首次最优值搜索迭代次数(Tb)上 EDA/w 略逊于 EDA，但通过对比 EDA+可以明显看出价值权重因子对寻优算法的引导作用。

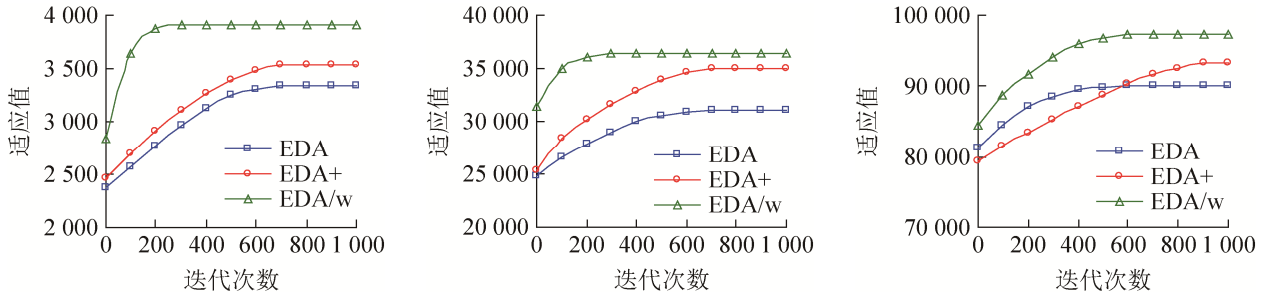
图 3 为 EDA、EDA+和 EDA/w 分别在实例 I05、I08 和 I13 上的优化情况，可以看出 EDA/w 在收敛的迭代次数上要少于 EDA 和 EDA+，这也表明 EDA/w 具备更高的收敛性。

文献[8]中提到算法种群规模的大小会对算法的寻优结果产生影响。为此本文针对实例 I05、I08、I13 在其他参数与前文一致的情况下，采用不同的种群规模的 EDA/w 对实例进行求解，计算测试结果如表 2 所示。

表 1　3 种对比算法对 I01～I13 的优化结果

Tab. 1　I01~I13 optimization expense for 3 algorithm

实例	EDA			EDA+			EDA/w		
	Best	AVG	Tb	Best	AVG	Tb	Best	AVG	Tb
I01	154	151	11	173	152	138	173	173	38
I02	344	327	47	359	336	274	364	364	98
I03	1 524	1 473	94	1 563	1 418	217	1 602	1 597	127
I04	3 384	3 173	157	3 375	3 110	158	3 597	3 514	41
I05	3 601	3 340	64	3 800.1	3 535	431	3 905.7	3 847	105
I06	4 467.2	4 387	143	4 623.4	4 126	354	4 799.3	4 528	124
I07	20 876	20 790	215	23 983	23 405	374	24 595	24 276	75
I08	31 408	31 084	394	35 797	34 970	437	36 873	36 460	77
I09	46 421	46 109	114	48 012	471 254	687	49 179	48 876	421
I10	55 663	53 442	311	59 879	58 132	542	61 480	61 147	146
I11	67 248	67 013	328	72 013	71 087	627	73 789	73 186	243
I12	81 026	78 647	157	84 046	82 363	873	86 091	85 279	449
I13	90 854	90 032	458	96 103	93 175	764	98 440	97 357	409



a:I05　b:I08　c:I13

图 3　三种算法对实例的优化比较

Fig.3　Instances optimization expense for 3 algorithm

表 2　不同种群规模下 EDA/w 的寻优情况

Tab. 2　Optimization of EDA/w under different population size

实例	*S*=80		*S*=150		*S*=200		*S*=300		*S*=400		*S*=500	
	Best	Tb	Best	Tb	Best	Tb	Best	Tb	Best	Tb	Best	Tb
I05	3 897	194	3 905.7	272	3 905.7	218	3 905.7	105	3 905.7	154	3 905.7	157
I08	35 782	654	36 434	357	36 848	204	36 873	77	36 873	214	36 873	194
I13	97 351	504	97 876	493	98 048	553	98 440	409	98 440	397	98 431	547

DOI: 10.16182/j.issn1004731x.joss.201712025

从表 2 中可以看出 EDA/w 随着种群规模的增大在寻优精度上得到了提升。但种群规模超过一定程度后反增加了算法首次搜索最优值(Tb)的开销。经分析，认为出现这种情况的原因是优势种群与算法种群规模之间的比例关系，比例过小会使得模型采样过于集中，容易导致算法陷入局部最优，比例过大会导致模型采样范围增大，算法难以进行有效的收敛。因此，依据实算经验优势种群为算法种群规模的 10%左右较为适宜。

下表 3 为 EDA/w 算法与其他算法优化结果的比较。其中实例 I01~I06 的最优解[10]为确定值，实例 I07~I13 的最优解未知，其已知最优解均为各优化算法的优化数据。表 3 中除 EDA/w 为实算数据外，其他对比算法的数据均来源于引用文献。其中，“-”表示文献未给出该信息；“*”表示通过本文算法寻优所获得的较对比算法更好的结果，并对最优数据进行了加粗处理。Moser 算法的优化结果直接取至于文献[11]中的数据，KLMA 算法的数据结果来源于文献[12]，MRLS 算法的数据结果来源于文献[13]，蚁群混合算法(ACO＆PR)算法的优化数据来源于文献[14]，CH 算法的优化数据来源于文献[15]，LNS 算法的优化数据来源于文献[16]，CHMW 算法的优化数据来源于文献[17]，改进的模糊人工蜂群算法(FABC)的优化数据来源于文献[18]，MACH 算法的优化数据来源于文献[19]，CPH 算法的优化数据来源于文献[20]。MOSER 与 KLMA 算法均为早期求解 MMKP 的经典的启发式算法，对其进行的对比研究较为广泛。ACO&PR 与 FABC 则代表国内学者在将其他类型的仿生算法应用于 MMKP 上的优化成果。MRLS、CH、LNS、CHMW、MACH 与 CPH 则代表近期国外学者们对 MMKP 优化问题的几种改进方法。

表 3 对比算法在实例 I01~I13 上的寻优情况
Tab. 3 I01~I13 optimization expense for Compare algorithm

实例	对比算法										
	MOSER	KLMA	MRLS	ACO&PR	CH	LNS	CHMW	FABC	MACH	CPH	EDA/w
I01	151	167	**173**	**173**	-	-	-	**173**	-	-	**173**
I02	291	354	**364**	**364**	-	-	-	**364**	-	-	**364**
I03	1 464	1 533	**1 602**	1 556	-	-	-	**1 602**	-	-	**1 602**
I04	3 375	3 437	**3 597**	3 452	-	-	-	**3 597**	-	-	**3 597**
I05	**3 905.7**	3 899.1	**3 905.7**	**3 905.7**	-	-	-	**3 905.7**	-	-	**3 905.7**
I06	4 115.2	4 799.3	**4 799.3**	**4 799.3**	-	-	-	**4 799.3**	-	-	**4 799.3**
I07	23 556	23 912	24 587	23 938	24 587	24 592	24 585	23 980.14	24 586	24 592	**24 595***
I08	35 373	35 979	36 877	35 997	**36 894**	36 886	36 885	36 005.56	36 888	36 885	36 873
I09	47 205	47 901	49 167	47 928	49 179	49 180	49 172	47 968	**49 180**	49 179	49 179
I10	58 648	59 811	61 437	59 846	61 464	61 469	61 460	59 879.34	**61 480**	61 464	**61 480**
I11	70 532	71 760	73 773	-	73 780	73 783	73 778	-	73 783	72 780	**73 789***
I12	82 377	84 141	86 069	-	86 081	86 097	86 077	-	**86 094**	86 081	86 091
I13	94 166	96 003	98 429	-	98 433	98 436	98 429	-	98 438	98 433	**98 440***

通过实验对比表明，EDA/w 对实例 I01~I06 都能够获得当前已知的最优解。在实例 I10 上与其他对比算法的优化结果持平。在实例 I07、I11 和 I13 上优于其他对比算法，获得了更好的解。就总体情况来说 EDA/w 在实例 I01~I13 的寻优上能够获得 10 次最优解，表现出较强的寻优能力。

表 4 为 EDA/w 与其他知名算法在实例 Inst01~Inst20 上优化情况的比较。其中 Bests 为优化算法在实例 Inst01~Inst20 上所能够获得最优结果的次数，以体现算法对不同实例的优化能力；Sum。为算法对所有实例优化数值的总和，以体现算法对实例优化的整体性能。

表 4 对比算法在实例 Inst01~Inst20 上的寻优情况
Tab. 4 Inst01~Inst20 ptimization expense for Compare algorithm

实例	对比算法						
	MRLS	CH	LNS	CHMW	MACH	CPH	EDA/w
Inst01	10 714	**10 738**	**10 738**	10 732	10 724	10 727	**10 738**
Inst02	**13 598**	**13 598**	**13 598**	**13 598**	**13 598**	**13 598**	**13 598**
Inst03	10 943	10 944	**10 955**	10 943	10 938	**10 955**	10 944
Inst04	14 429	14 442	14 445	14 445	**14 456**	14 452	14 447
Inst05	17 053	17 053	17 055	17 055	17 053	17 055	**17 059***
Inst06	16 823	16 827	16 835	16 823	16 832	16 823	**16 840***
Inst07	16 423	16 440	16 440	16 440	**16 442**	16 440	16 440
Inst08	17 506	17 510	**17 514**	17 505	17 508	17 510	17 507
Inst09	17 754	**17 761**	17 757	17 753	17 760	**17 761**	17 753
Inst10	19 314	**19 316**	19 305	19 306	19 311	**19 316**	**19 316**
Inst11	19 431	**19 441**	**19 441**	19 434	19 437	**19 441**	19 437
Inst12	21 730	21 732	**21 739**	21 738	21 738	21 732	21 738
Inst13	21 569	**21 577**	**21 577**	21 574	**21 577**	**21 577**	**21 577**
Inst14	32 869	**32 874**	32 873	32 869	32 872	32 872	32 872
Inst15	39 148	39 160	39 159	39 160	39 161	39 160	**39 162***
Inst16	43 354	43 362	43 364	43 363	**43 366**	43 362	43 362
Inst17	54 349	54 360	54 360	54 352	**54 363**	54 360	54 360
Inst18	60 456	60 464	60 463	60 463	**60 467**	60 460	**60 467**
Inst19	64 921	64 925	64 928	64 924	64 931	64 925	**64 932***
Inst20	75 603	75 612	**75 615**	75 609	75 614	75 612	75 614
Bests	1	7	8	1	7	6	**9**
Sum.	587 987	588 136	588 161	588 086	588 148	588 138	**588 163**

通过表 4 可以看出，EDA/w 分别在实例 Inst01、Inst02、Inst10、Inst13、Inst18 上获得了与其他算法相同的最优解，并且能够在实例 Inst05、Inst06、Inst15、Inst19 上获得了较其他对比算法更好的解。就总体情况而言，EDA/w 在对实例 I01~I13 及实例 Inst01~Inst20 上所获得的最优解次数以及优化值总和上均优于其他对比算法。这表明 EDA/w 是一种有效的 MMKP 优化算法，在对于 MMKP 的优化上表现出较强的竞争力。

4 结论

对于多维多选择背包问题(MMKP)优化的主要困难之处在于众多约束条件的限制，通常要求优化算法兼具很强的全局和局部搜索的能力。标准 EDA 算法是一种全局搜索能力很强的寻优算法，比较适宜于优化约束条件相对较少的问题，如 0-1 背包问题和多维背包问题。本文以物品利润与成本的比值来构建物品的价值权重因子，并将价值权重值引入到分布概率模型的构建和更新中；在此基础之上新算法还对 MMKP 解的构成进行了分析，并提出了一种非可行解的修复方法。此举使得改进后的算法在搜索过程中比标准 EDA 更具有目的性，克服了标准 EDA 局部搜索能力偏弱的缺陷，后续实验也验证了 EDA/w 能够有效的对 MMKP 进行优化，并且在总体的优化性能上有所提高。

参考文献：

[1] Shapiro J L. Diversity Loss in General Estimation of Distribution Algorithms[M]// Parallel Problem Solving from Nature-PPSN IX. Springer Berlin Heidelberg, 2006: 92-101.

[2] Zhou Q, Luo W. A Novel Multi-Population Genetic Algorithm for Multiple-Choice Multidimensional Knapsack Problems[J]. Lecture Notes in Computer

Science (S0302-9743), 2010, 63(82): 148-157.

[3] Cherfi N, Hifi M. A column generation method for the multiple-choice multi-dimensional knapsack problem [J]. Computational Optimization and Applications (S0926-6003), 2010, 46(1): 51-73.

[4] Taha Ghasemi, Mohammadreza Razzaz. Development of core to solve the Multidimensional Multiple choice Knapsack Problem[J]. Computers & Industrial Engineering (S0360-8352), 2011, 60(2): 349-360.

[5] Khalili-Damghani K, Nojavan M, Tavana M. Solving fuzzy Multidimensional Multiple-Choice Knapsack Problems: The multi-start Partial Bound Enumeration method versus the efficient epsilon-constraint method [J]. Applied Soft Computing (S1568-4946), 2013, 13(4): 1627-1638.

[6] Baluja S. Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive[J]. Learning Proceedings of IEEE Conference on Evolutionary Computation (S2210-6502), 1994, 2(4): 710-715.

[7] 高尚. 背包问题的分布估计算法[J]. 中南大学学报(自然科学版), 2013, 44(2): 165-168.
GAO Shang. Solving knapsack problem by estimation of distribution algorithm[J]. Journal of Central South University(Science and Technology). 2013, 44(2): 165-168.

[8] Osorio M A, Glover F, Hammer P. Cutting and Surrogate Constraint Analysis for Improved Multidimensional Knapsack Solutions[J]. Annals of Operations Research (S0254-5330), 2002, 117(1/2/3/4): 71-93.

[9] Hifi M, Michrafy M, Sbihi A. Heuristic algorithms for the multiple-choice multidimensional knapsack problem[J]. Journal of the Operational Research Society (S2194-6698), 2004, 55(12): 1323-1332.

[10] Sin C Ho, Michel Gendreau. Path relinking for the vehicle routing problem[J]. Journal of Heuristics (S1381-1231), 2006, 12(1/2): 55-72.

[11] Moser M, Jokanovic D P, Shiratori N. An Algorithm for the Multidimensional Multiple-Choice Knapsack Problem[J]. Ieice Transactions on Fundamentals of Electronics Communications & Computer Sciences (S0916-8508), 1997, 80(3): 582-589.

[12] Khan S, Li K F, Manning E G, et al. Solving the knapsack problem for adaptive multimedia systems[J]. Studia Informatica Universalis (S0867-1753), 2002, 2(1): 157-178.

[13] Hifi M, Michrafy M, Sbihi A. A Reactive Local Search-Based Algorithm for the Multiple-Choice Multi-Dimensional Knapsack Problem[J]. Computational Optimization and Applications (S0926-6003), 2006, 33(2/3): 271-285.

[14] 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 Xiaoxia, Tang Lixin. A new ACO&PR algorithm for multiple-choice multidimensional knapsack problem[J]. Control and Decision, 2009, 24(5): 729-733.

[15] Cherfi N, Hifi M. Hybrid algorithms for the Multiple-choice Multi-dimensional Knapsack Problem [J]. International Journal of Operational Research (S1109-2858), 2009, 5(1): 89-109.

[16] Hifi M, Wu L. An equivalent Model for Exactly Solving the Multiple-choice Multidimensional Knapsack Problem[J]. International Journal of Combinatorial Optimization Problems and Informatics (S2007-1558), 2012, 3(3): 43-58.

[17] Cr´evits I, Hanafi S, Mansi R, Wilbaut C. Iterative semi-continuous relaxation heuristics for the multiple-choice multidimensional knapsack problem[J]. Computers and Operations Research (S0305-0548), 2012, 39(1): 32-41.

[18] 柳寅, 马良, 黄钰. 模糊人工蜂群算法的多选择多维背包问题求解[J]. 运筹与管理, 2013, 22(5): 98-103.
Liu Yin, Ma Liang, Huang Yu. Fuzzy artificial bees colony algorithm for solving multi-choice multidimensional knapsack problem[J]. Operations Research & Management Science, 2013, 22(5): 98-103.

[19] Raïd Mansi, Cláudio Alves, J M Valério de Carvalho, et al. A hybrid heuristic for the multiple choice multidimensional knapsack problem[J]. Engineering Optimization (S0305-215X), 2013, 45(8): 983-1004.

[20] Shojaei H, Basten T, Geilen M, et al. A fast and scalable multi-dimensional multiple-choice knapsack heuristic[J]. ACM Transactions on Design Automation of Electronic Systems (S1084-4309), 2013, 18(4): 96-95.