基于 QUBO 模型的多维背包问题

王琪玮*，**杨洪礼**

山东科技大学数学与系统科学学院，山东青岛 266590

摘要: 多维背包问题是经典的组合优化问题，目标是找到一组最优的选择方案，以满足所有的约束条件。传统多维背包问题的求解算法普遍存在计算速度慢、计算量随问题维度的增加而呈现指数级增长等缺点。针对以上问题，提出 QUBO（二次无约束二进制优化）模型，将多维背包问题表示为一个二次无约束二进制优化问题。使用二进制变量表示多维背包问题的目标函数，通过惩罚项将约束条件以二次项的形式添加到目标函数中，进一步将目标函数转换为 QUBO 形式，选择量子退火算法找到 QUBO 问题的最优解。通过开源 Python 库 PyQUBO 创建相应模型，在 D-Wave 平台用量子退火算法求解。结果表明，QUBO 模型有较强的数学表达能力使问题更具有结构性，并适用于大规模问题，处理含有大量变量和约束条件的多维背包问题。

关键词: 多维背包问题; QUBO 模型; 二进制; 无约束; 量子退火法

DOI: 10.57237/j.cst.2023.03.005

Multidimensional Knapsack Problem Based on QUBO Model

Wang Qiwei*, **Yang Hongli**

College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China

Abstract: Multidimensional knapsack problem is a classical combinatorial optimization problem, the goal is to find a set of optimal options to satisfy all the constraints. The traditional algorithms for solving multidimensional knapsack problems generally have some disadvantages, such as slow computation speed and exponential increase of computation with the increase of problem dimension. To solve these problems, a QUBO (quadratic unconstrained binary optimization) model is proposed, and the multidimensional knapsack problem is expressed as a quadratic unconstrained binary optimization problem. Binary variables are used to represent the objective function of the multidimensional knapsack problem. The constraints are added to the objective function in the form of quadratic terms by means of penalty terms. The objective function is further transformed into QUBO form. The model is created by PyQUBO, an open source Python library, and solved by quantum annealing algorithm on D-Wave platform. The results show that the QUBO model has a strong ability of mathematical expression, which makes the problem more structured, and is suitable for large-scale problems, dealing with multidimensional knapsack problems with a lot of variables and constraints.

Keywords: Multidimensional Backpack Problem; QUBO Model; Binary; No Constraints; Quantum Annealing Algorithm

*通信作者: 王琪玮, 2938035936@qq.com

收稿日期: 2023-08-22; 接受日期: 2023-09-14; 在线出版日期: 2023-09-19 http://www.computscitech.com

1 引言

多维背包问题（Multidimensional Knapsack Problem, MKP）是一种经典的组合优化 NP-hard 问题，在项目投资和成本预算等领域被广泛应用[1]。求解多维背包问题的方法主要分为精确算法和启发式算法[2]两类。其中，精确算法包括动态规划法[3]、分支定界法[4]、贪心算法[5]、穷举法[6]等，启发式算法包括蚁群算法[7]、遗传算法[8]、粒子群算法[9]、模拟退火算法[10]。国内外学者在求解背包问题时提出了许多改进算法，近年来具有代表性的相关研究如下所述。

Luis Fernando Mingo López 等人[11]针对多维背包问题提出了一种混合遗传启发式算法，通过遗传算子将粒子群用于二进制变量，使用随机变异/交叉运算并添加罚函数来解决特殊情况，增强算法全局搜索能力。钱淑渠等人[12]在挖掘生物免疫系统的学习、记忆及识别功能上提出一种克隆修复免疫算法，采用贪婪修补方法提高抗体群的可行度，通过环境识别算子以及对相似环境初始群的产生方式加速算法跟踪相似环境的速度，仿真结果表明提出的算法具有解决高维组合优化问题的优越性，跟踪环境的速度较快。FENG 等人[13]提出一种基于自我学习的二元蛾搜索算法，使每个个体向比自身更好的个体学习，提高了算法的搜索能力。马立肖等人[14]采用双种群协同进化，以差异演化算法为主体的马克-鲍德温混合差异演化算法，在演化过程中分别引入拉马克进化和鲍德温效应 2 种局部搜索算子，引导种群进化方向，仿真实验结果表明，该算法求解精度高，收敛速度快。José García 和 Carlos Maureira [15]在量子布谷鸟搜索算法的基础上引入 k-最近邻技术，k-最近邻技术用于指导解决方案的移动，数值实验表明该算法减少执行时间同时提高解决方案的质量。Hamza Onoruoiza 和 Abubakar Bala [16]提出了一种基于化学反应原理的元启发式算法，称为化学反应优化算法，利用一个新颖的、乐观的邻域搜索算子和一个贪婪的修复和优化算子来修复无效的解决方案和改进可行的解决方案。罗亚波等人[17]提出了一种牵制平衡算法，通过模拟生态系统中各物种间相互依存、牵制，最终达到动态平衡的自然机制，提高了算法的收敛性。

尽管背包问题研究广度和深度在不断延伸，相关研究的模型建立、算法设计和实际应用也取得了众多成就，但这些算法对于物品数量和约束条件增多的大规模组合优化问题，仍存在求解难度和时间复杂度呈指数级增加等不足。为进一步求解多维背包问题，本文引入 QUBO 模型，通过 QUBO 转换将多维背包问题映射成一个叫哈密顿算符（Hamiltonian）的能量表达式，用于桥接经典计算和量子计算，在 D-Wave 平台用量子退火法求得最优解[18]。

QUBO 模型已经成为量子计算领域的基础，称为量子退火和富士通的数字退火。通过这些联系，QUBO 模型是 D-Wave Systems 开发的量子计算机和 IBM 开发的神经形态计算机进行实验的核心[19]。

D-Wave 平台是第一个基于量子退火的实际商业系统，其核心是量子处理单元（QPU），它将量子退火原理应用于复杂的组合优化问题[20]。

2 多维背包问题

多维背包问题可描述为：给定一个背包和 n 个物品，每个物品 i 具有多个维度或属性。在所有物品中选择一组满足所有约束条件的子集，使得背包中物品的总价值最大。该问题用数学模型[21]可表示为如下：

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

$$\mathrm{s.t.} \sum_{i=1}^{n} a_{ij} x_i \le b_j, j = 0,1,\dots,m \tag{2}$$

$$x_i \in \{0,1\}, i = 0,1,\dots,n \tag{3}$$

其中，公式(1)为放入背包中所有物品的总价值；n 为物品的数量；m 为背包的维度；v_i为物品 i 的价值；a_{ij}为物品 i 所对应背包 j 维度资源的占用；x_i为决策变量，x_i=1 时表示物品 i 被选中，x_i=0 时表示物品 i 未被选中；b_j为背包 j 维度资源的限制。

3 QUBO 模型

多维背包问题具有较高的抽象性和复杂性，面对多个约束条件以及大量数据时，给求得最优解带来了较大的困难，从而引入 QUBO 模型，使约束问题转为无约束问题，进而可在量子计算机运行。QUBO 模型表示的成本函数是由线性部分和二次项部分组成的二元变量的二次函数，设x_i是第 i 个二进制变量，x_j是第 j 个二进制变量，其公式化[22]为：

$$H_{QUBO}(x) = \sum_{i\in n} a_i x_i + \sum_{i,j\in n} b_{ij} x_i x_j\, , x_i \in \{0,1\} \tag{4}$$

其中a_i和b_{ij}为任意实数，n 为二进制变量个数的集合，对于任意的 i 和 j，b_{ij}=b_{ji}。

QUBO 可以用矩阵形式表示，使Q_{ij}是一个$n \times n$型的矩阵，其元素由下式给出：

$$Q_{ij} = \begin{cases} a_i, i = j, \forall i \in n \\ b_{ij}, \forall i, j \in n \text{ and } i < j \\ 0, otherwise \end{cases} \tag{5}$$

通过使用Q_{ij}和排列二进制变量生成的列向量x，将$H_{QUBO}(x)$重写为：

$$H_{QUBO}(x) = \sum_{i \le j} Q_{ij} x_i x_j = x^{\mathrm{T}} Q x \tag{6}$$

其中x^{T}表示为向量x的转置。

除了对决策变量的 0/1 限制外，QUBO 是一个无约束模型，所有问题数据都包含在 Q 矩阵中。这些特性使得 QUBO 模型作为组合优化问题的建模框架特别有吸引力，为经典约束表示提供了一种新的替代方案。

4 问题描述及 QUBO 模型的转化

多维背包问题可描述为：给定一个背包和 n 个物品，其中，公式(1)为放入背包中所有物品的总价值；n 为物品的数量；m 为背包的维度；v_i为物品 i 的价值；a_{ij}为物品 i 所对应背包 j 维度资源的占用；x_i为决策变量，x_i=1 时表示物品 i 被选中，x_i=0 时表示物品 i 未被选中；b_j为背包 j 维度资源的限制。

4.1 问题描述

假设有 n 个物品，表示为x_i，（i=0,…, n-1），每个物品有四个属性，分别为价值，物品长、宽、高，表示为v_i，l_i，w_i，h_i，给定一个长宽高固定的背包，即背包长为 L，宽为 W，高为 H，选取物品放入背包中，最大化背包中物品的价值之和，同时使背包中物品的长度之和必须小于背包的长度，背包中物品的宽度之和必须小于背包的宽度，背包中物品的宽度之和必须小于背包的宽度。用数学模型表示为如下：

目标函数：

$$\max \sum_{i=0}^{n-1} v_i x_i \tag{7}$$

约束条件：

$$\sum_{i=0}^{n-1} l_i x_i \le \mathrm{L} \tag{8}$$

$$\sum_{i=0}^{n-1} w_i x_i \le \mathrm{W} \tag{9}$$

$$\sum_{i=0}^{n-1} h_i x_i \le \mathrm{H} \tag{10}$$

4.2 QUBO 模型的转化

通过 QUBO 模型的转化将组合优化问题转化为二进制无约束的二次函数，使其与量子计算结合，利用量子力学叠加和纠缠的特性[23]寻找 QUBO 问题的全局最优解，解决经典计算难以处理的复杂优化问题。

步骤 1 定义二进制变量x_i表示是否选择物品 i 放入背包中，表示如下：

$$x_i = \begin{cases} 1\text{，物品}i\text{被选择} \\ 0\text{，物品}i\text{未被选择} \end{cases} \tag{11}$$

步骤 2 在 QUBO 矩阵中，我们只能将相等视为一种惩罚方法。因此，为了表达 QUBO 矩阵，除了公式(11)中定义的常用二进制变量外，还定义了一组松弛变量以使不等式(8) (9) (10)成为相等约束。通过日志编码使用松弛变量[24]，不等式约束由以下等式表示：

$$\sum_{i=0}^{n-1} l_i x_i + y1 = \mathrm{L} \tag{12}$$

$$\sum_{i=0}^{n-1} w_i x_i + y2 = \mathrm{W} \tag{13}$$

$$\sum_{i=0}^{n-1} h_i x_i + y3 = \mathrm{H} \tag{14}$$

松弛变量具体如下：

$$\mathrm{y1} = \sum_{i=0}^{[\log_2(\mathrm{L}-1)]} 2^i y1_i, y_i \in \{0,1\}, \forall i \in \{0,1,2,\dots,[\log_2(\mathrm{L}-1)]\} \tag{15}$$

$$\mathrm{y2} = \sum_{i=0}^{[\log_2(\mathrm{W}-1)]} 2^i y2_i, y_i \in \{0,1\}, \forall i \in \{0,1,2,\dots,[\log_2(\mathrm{W}-1)]\} \tag{16}$$

$$\mathrm{y3} = \sum_{i=0}^{[\log_2(\mathrm{H}-1)]} 2^i y3_i, y_i \in \{0,1\}, \forall i \in \{0,1,2,\dots,[\log_2(\mathrm{H}-1)]\} \tag{17}$$

通过日志编码引入松弛变量将不等式转换成等式，经由二进制展开，以创建 QUBO 约束系统 Ax=b 的形式。

步骤 3 将约束条件转换为二次惩罚以添加到目标函数，使约束二次优化模型转为等效的无约束 QUBO 模型。在哈密顿量H_C中编码这些约束，其中 P 为惩罚因子，表示如下：

$$H_C = \mathrm{P}\left(\mathrm{L} - \sum_{i=0}^{n-1} l_i x_i - \mathrm{y1}\right)^2 + \mathrm{P}\left(\mathrm{W} - \sum_{i=0}^{n-1} w_i x_i - \mathrm{y2}\right)^2 + \mathrm{P}\left(\mathrm{H} - \sum_{i=0}^{n-1} h_i x_i - \mathrm{y3}\right)^2 \tag{18}$$

步骤 4 将二次惩罚添加到目标函数，任意取惩罚因子 P 的值。

$$\min H=-\sum_{i=0}^{n-1} v_i x_i+H_C=-\sum_{i=0}^{n-1} v_i x_i{}^2+\mathrm{P}\left(\mathrm{L}-\sum_{i=0}^{n-1} l_i x_i-\mathrm{y1}\right)^{\mathrm{T}}\left(\mathrm{L}-\sum_{i=0}^{n-1} l_i x_i-\mathrm{y1}\right)+\mathrm{P}\left(\mathrm{W}-\sum_{i=0}^{n-1} w_i x_i-\mathrm{y2}\right)^{\mathrm{T}}\left(\mathrm{L}-\sum_{i=0}^{n-1} l_i x_i-\mathrm{y1}\right)+\mathrm{P}\left(\mathrm{H}-\sum_{i=0}^{n-1} h_i x_i-\mathrm{y3}\right)^{\mathrm{T}}\left(\mathrm{H}-\sum_{i=0}^{n-1} h_i x_i-\mathrm{y3}\right)=x^{\mathrm{T}}Dx+x^{\mathrm{T}}Ex+x^{\mathrm{T}}Fx+x^{\mathrm{T}}Gx+\mathrm{C}=x^{\mathrm{T}}Qx \tag{19}$$

由于x_i为只可取 0,1 值得二进制变量，故$x_i=x_i{}^2$，由于常数值不影响二进制变量x_i的取值，故去掉常数 C，得到 QUBO 模型$\min x^TQx$，x 为二进制变量。

5 数值实验

实例 1 当背包长为 15，宽为 7，高为 7，n 为 3，即有三件物品可选择时，详细数据如表 1 所示。

表 1 物品数为 3 时数据。

物品	属性			
	价值	长	宽	高
0	5	3	3	5
1	2	5	2	1
2	3	8	4	3

表达式 H 可扩展为如下：

$$\left(\mathrm{L}-\sum_{i=0}^{n-1} l_i x_i-\mathrm{y1}\right)^2=(15-3x_0-5x_1-8x_2-y1_0-2y1_1-4y1_2-8y1_3)^2=225-81x_0{}^2-125x_1{}^2-176x_2{}^2-29y1_0{}^2-56y1_1{}^2-104y1_2{}^2-176y1_3{}^2+30x_0x_1+48x_0x_2+6x_0y1_0+12x_0y1_1+24x_0y1_2+48x_0y1_3+80x_1x_2+10x_1y1_0+20x_1y1_1+40x_1y1_2+80x_1y1_3+16x_2y1_0+32x_2y1_1+64x_2y1_2+128x_2y1_3+4y1_0y1_1+8y1_0y1_2+16y1_0y1_3+16y1_1y1_2+32y1_1y1_3+64y1_2y1_3 \tag{20}$$

$$\left(\mathrm{W}-\sum_{i=0}^{n-1} w_i x_i-\mathrm{y2}\right)^2=(7-3x_0-2x_1-4x_2-y2_0-2y2_1-4y2_2)^2=49-33x_0{}^2-24x_1{}^2-40x_2{}^2-13y2_0{}^2-24y2_1{}^2-40y2_2{}^2+12x_0x_1+24x_0x_2+6x_0y2_0+12x_0y2_1+24x_0y2_2+16x_1x_2+4x_1y2_0+8x_1y2_1+16x_1y2_2+8x_2y2_0+16x_2y2_1+32x_2y2_2+4y2_0y2_1+8y2_0y2_2+16y2_1y2_2 \tag{21}$$

$$\left(\mathrm{H}-\sum_{i=0}^{n-1} h_i x_i-\mathrm{y3}\right)^2=(7-5x_0-1x_1-3x_2-y3_0-2y3_1-4y3_2)^2=49-45x_0{}^2-13x_1{}^2-33x_2{}^2-13y3_0{}^2-24y3_1{}^2-40y3_2{}^2+10x_0x_1+30x_0x_2+10x_0y3_0+20x_0y3_1+40x_0y3_2+6x_1x_2+2x_1y3_0+4x_1y3_1+8x_1y3_2+6x_2y3_0+12x_2y3_1+24x_2y3_2+4y3_0y3_1+8y3_0y3_2+16y3_1y3_2 \tag{22}$$

$$-\sum_{i=0}^{n-1} v_i x_i{}^2=-5x_0{}^2-2x_1{}^2-3x_2{}^2 \tag{23}$$

当惩罚因子 P 取值为 2 时，哈密顿量 H 展开式如下表示：

$$H=646-323x_0{}^2-326x_1{}^2-501x_2{}^2-58y1_0{}^2-112y1_1{}^2-208y1_2{}^2-352y1_3{}^2-26y2_0{}^2-48y2_1{}^2-80y2_2{}^2-26y3_0{}^2-48y3_1{}^2-80y3_2{}^2+104x_0x_1+204x_0x_2+12x_0y1_0+24x_0y1_1+48x_0y1_2+96x_0y1_3+12x_0y2_0+24x_0y2_1+48x_0y2_2+20x_0y3_0+40x_0y3_1+80x_0y3_2+204x_1x_2++20x_1y1_0+40x_1y1_1+80x_1y1_2+160x_1y1_3+8x_1y2_0+16x_1y2_1+32x_1y2_2+4x_1y3_0+8x_1y3_1+16x_1y3_2+32x_2y1_0+64x_2y1_1+128x_2y1_2+256x_2y1_3+16x_2y2_0+32x_2y2_1+64x_2y2_2+12x_2y3_0+24x_2y3_1+48x_2y3_2+8y1_0y1_1+16y1_0y1_2+32y1_0y1_3+32y1_1y1_2+64y1_1y1_3+128y1_2y1_3+8y2_0y2_1+16y2_0y2_2+32y2_1y2_2+8y3_0y3_1+16y3_0y3_2+32y3_1y3_2 \tag{24}$$

哈密顿量 H 矩阵 Q 由字典形式表示为如下：

{'x[0]': -323.0, 'x[1]': -326.0, 'x[2]': -501.0, 'y3[0]': -26.0, 'y2[2]': -80.0, 'y2[1]': -48.0, 'y1[3]': -352.0, 'y1[1]': -112.0, 'y3[2]': -80.0, 'y3[1]': -48.0, 'y1[0]': -58.0, 'y1[2]': -208.0, 'y2[0]': -26.0}, {('x[1]', 'x[0]'): 104.0, ('x[2]', 'x[0]'): 204.0, ('x[2]', 'x[1]'): 204.0, ('y3[0]', 'x[0]'): 20.0, ('y3[0]', 'x[1]'): 4.0, ('y3[0]', 'x[2]'): 12.0, ('y2[2]', 'x[0]'): 48.0, ('y2[2]', 'x[1]'): 32.0, ('y2[2]', 'x[2]'): 64.0, ('y2[1]', 'x[0]'): 24.0, ('y2[1]', 'x[1]'): 16.0, ('y2[1]', 'x[2]'): 32.0, ('y2[1]', 'y2[2]'): 32.0, ('y1[3]', 'x[0]'): 96.0, ('y1[3]', 'x[1]'): 160.0, ('y1[3]', 'x[2]'): 256.0, ('y1[1]', 'x[0]'): 24.0, ('y1[1]', 'x[1]'): 40.0, ('y1[1]', 'x[2]'): 64.0, ('y1[1]', 'y1[3]'): 64.0, ('y3[2]', 'x[0]'): 80.0, ('y3[2]', 'x[1]'): 16.0, ('y3[2]', 'x[2]'): 48.0, ('y3[2]', 'y3[0]'): 16.0, ('y3[1]', 'x[0]'): 40.0, ('y3[1]', 'x[1]'): 8.0, ('y3[1]', 'x[2]'): 24.0, ('y3[1]', 'y3[0]'): 8.0, ('y3[1]', 'y3[2]'): 32.0, ('y1[0]', 'x[0]'): 12.0, ('y1[0]', 'x[1]'): 20.0, ('y1[0]', 'x[2]'): 32.0, ('y1[0]', 'y1[3]'): 32.0, ('y1[0]', 'y1[1]'): 8.0, ('y1[2]', 'x[0]'): 48.0, ('y1[2]', 'x[1]'): 80.0, ('y1[2]', 'x[2]'): 128.0, ('y1[2]', 'y1[3]'): 128.0, ('y1[2]', 'y1[1]'): 32.0, ('y1[2]', 'y1[0]'): 16.0, ('y2[0]', 'x[0]'): 12.0, ('y2[0]', 'x[1]'): 8.0, ('y2[0]', 'x[2]'): 16.0, ('y2[0]', 'y2[2]'): 16.0, ('y2[0]', 'y2[1]'): 8.0}

在 D-Wave 平台用量子退火法求得最优解为：{x[0]: 1, x[1]: 1, x[2]: 0 }，即选择物品 0，物品 1 放入背包中，总价值为 7。

实例 2 当背包长为 20，宽为 20，高为 25，n 为 19，即有 19 件物品可选择时，详细数据如表 2 所示。

表 2 物品数为 19 时数据。

物品	属性			
	价值	长	宽	高
0	1	3	2	5
1	2	4	2	1
2	3	1	4	3
3	4	3	4	2
4	5	3	5	4
5	1	6	3	4
6	8	3	5	6
7	35	4	9	2
8	6	5	4	4
9	5	3	8	9
10	3	5	8	6
11	8	3	5	4
12	2	4	6	3
13	9	6	4	7
14	4	2	8	3
15	3	6	8	2
16	9	7	3	2
17	5	8	5	2
18	6	7	1	3

哈密顿量 H 矩阵 Q 由字典形式表示为如下：

{'x[5]': -1997.0, 'y3[2]': -736.0, 'x[16]': -1761.0, 'x[15]': -2227.0, 'x[17]': -2113.0, 'x[14]': -1896.0, 'y2[2]': -576.0,……, ('y1[2]', 'y1[3]'): 256.0, ('y1[2]', 'y1[0]'): 32.0, ('y1[2]', 'y1[1]'): 64.0, ('y1[2]', 'x[18]'): 224.0, ('y1[2]', 'x[12]'): 128.0, ('y1[2]', 'x[1]'): 128.0, ('y1[2]', 'y1[4]'): 160.0}

在 D-Wave 平台用量子退火法求得最优解为：{ 'x[0]': 0, 'x[1]': 0, 'x[2]': 0, 'x[3]': 0, 'x[4]': 0, 'x[5]': 0, 'x[6]': 0, 'x[7]': 1, 'x[8]': 0, 'x[9]': 0, 'x[10]': 0, 'x[11]': 1, 'x[12]': 0, 'x[13]': 1 'x[14]': 0, 'x[15]': 0, 'x[16]': 0, 'x[17]': 0, 'x[18]': 1}，即选择物品 7，物品 11，物品 13，物品 18 放入背包中，总价值为 58。

由此可合理推断 QUBO 模型适用于大量数据的计算。

6 结束语

本文使用 QUBO 模型将有多个约束条件的背包问题转换为二次无约束二进制问题，使其可在量子计算机上运行，并行计算来加速搜索过程并提高效率，同时在搜索过程中逃离局部最优解并继续进行全局搜索[25]。数值实验表明 QUBO 模型有利于解决大规模多约束条件的背包问题。在未来的研究中，拟将 QUBO 模型应用于更多实际问题中，以进一步测试其可行性。

参考文献

[1] Genserik L L Reniers, Kenneth Sörensen. An Approach for Optimal Allocation of Safety Resources: Using the Knapsack Problem to Take Aggregated Cost-Efficient Preventive Measures [J]. Risk Analysis, 2013, 33(11): 2056-2067.

[2] 陈建荣. 求解 0-1 背包问题的改进二进制捕鱼算法 [J]. 计算机技术与发展, 2023, 33(5): 187-193.

[3] Li Junbao, Chu Shuchuan, Yang J S P. Discriminant Pattern Classification [J]. Journal of Digital Information Management, 2008, 6(2): 203-207.

[4] LI V C, LIANG Y C, CHANG H F. Solving the multidimensional knapsack problems with generalized upper bound constraints by the adaptive memory projection method [J]. Computers & Operations Research, 2012, 39(9): 2111-2121.

[5] 孙佳宁，马海龙，张立臣，李鹏. 求解 0-1 背包问题的融合贪心策略的回溯算法 [J]. 计算机技术与发展, 2022, 32(02): 190-195.

[6] 乔丽娟，徐岩. 基于 0-1 背包问题的综合性实验研究 [J]. 电子技术, 2018, 47(11): 15-17.

[7] 熊伟清，魏平，王小权. 蚁群算法求解多维 0/1 背包问题 [J]. 计算机工程与科学, 2006, 28(10): 78-79.

[8] Abdellah Rezoug, Mohamed Bader-El-Den&Dalila Boughaci. Guided genetic algorithm for the multidimensional knapsack problem [J]. Memetic Computing, 2018, 10(1): 29-42.

[9] 李枝勇，马良，张惠珍. 求解 0/1 背包问题的自适应元胞粒子群算法 [J]. 计算机工程, 2014, 40(10): 198-203.

[10] 许小勇. 基于改进的模拟退火算法求解 0/1 背包问题 [J]. 海南大学学报：自然科学版, 2008, 26(4): 356-358.

[11] Luis Fernando Mingo López, Nuria Gómez Blas, Alberto Arteta Albert. Multidimensional knapsack problem optimization using a binary particle swarm model with genetic operations [J]. Soft Computing, 2018, 22: 2567-2582.

[12] 钱淑渠，武慧虹，林妤．求解高维动态背包问题的克隆修复免 [J]. 计算机工程, 2017, 43(9): 220-227.

[13] FENG Y H, WANG G G. A binary moth search algorithm based on self-learning for multidimensional knapsack problem [J]. Future Generation Computer Systems, 2022, 126: 48-64.

[14] 马立肖，赵占芳．一种求解背包问题的混合差异演化算法 [J]. 计算机工程, 2012, 38(7): 164-167.

[15] José García, Carlos Maureira. A KNN quantum cuckoo search algorithm applied to the multidimensional knapsack problem [J]. Applied Soft Computing, 2021, 102.

[16] Hamza Onoruoiza, Abubakar Bala. An improved chemical reaction optimisation algorithm for the 0-1 knapsack problem [J]. International Journal of Bio-Inspired Computation, 2022, 19(4): 253-266.

[17] 罗亚波，滕红玺．求解 0-1 背包问题的牵制平衡算法 [J]. 工业工程, 2023, 26(3): 116-123.

[18] Siddharth Jain. Solving the Traveling Salesman Problem on the D-Wave Quantum Computer [J]. Frontiers in Physics, 2021, 9.

[19] Fred Glover, Gary Kochenberger, Rick Hennig, Yu Du. Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models [J]. Annals of Operations Research, 2022, 314(1): 141-183.

[20] Christos Papalitsas, Theodore Andronikos, Konstantinos Giannakis, Georgia Theocharopoulou, Sofia Fanarioti. A QUBO Model for the Traveling Salesman Problem with Time Windows [J]. Algorithms 2019, 2022, 12(11): 224.

[21] 潘大志，蒋妍，刘雅文．求解多维背包问题的双决策交互差异算法 [J]. 计算机工程, 2023, 49(7): 21-33.

[22] Mashiyat Zaman, Kotaro Tanahashi, Shu Tanaka. PyQUBO: Python Library for Mapping Combinatorial Optimization Problems to QUBO Form [J]. IEEE Transactions on Computers, 2021, 71(4): 838-850.

[23] Peter J. Bussey. Modern quantum mechanic [J]. Contemporary Physics, 2021, 62(1): 53-54.

[24] Andrew Lucas. Ising formulations of many NP problems [J]. Frontiers in Physics, 2014, 2.

[25] V. Mehta, F. Jin, K. Michielsen, H. De Raedt. On the hardness of quadratic unconstrained binary optimization problems [J]. Frontiers in Physics, 2022, 10.