

## 0/1多维Knapsack问题的遗传算法

Shalin Shah

[sshah100@jhu.edu](mailto:sshah100@jhu.edu)

[shah.shalin@gmail.com](mailto:shah.shalin@gmail.com)

约翰霍普金斯大学

0/1多维knapsack问题是具有m个约束条件的0/1 knapsack问题，这使得它很难用传统的方法如动态编程或分支和约束算法来解决。我们提出了一个用Java和C++代码的多维背包问题的遗传算法，能够在很短的计算时间内解决公开的实例。我们的算法使用迭代计算的拉格朗日乘数作为约束权重来增强多维结包问题的贪婪算法，并在遗传算法的贪婪交叉中使用该信息。该算法使用其他几个超参数，可以在代码中设置，以控制收敛性。我们的算法改进了Chu和Biasley的算法，因为它能更快地收敛到最佳或接近最佳的解决方案。

关键词。

多维结包问题 遗传算法 实用率 贪婪算法

### 1. 简介

使用分支和约束或动态编程来解决多维knapsack问题是很困难的。由于存在多个约束条件，也很难获得一个好的近似解决方案，如贪婪算法。然而，可以将贪婪算法作为遗传算法的一部分，我们的结果表明，它的效果非常好。我们的算法不仅能够超过贪婪的估计，而且对于大多数问题实例，它能够找到最佳解决方案。我们的算法与[2]相似，后者使用贪婪的交叉来解决0/1结包问题。由于多维背包问题有多个约束条件，我们使用迭代计算的拉格朗日乘数给每个约束条件分配一个权重。这与[1]中使用代用乘数的方法类似。不同的是，我们在贪婪的交叉中使用乘数，这是高度建设性的，可以更快地找到最佳解决方案。

### 2. 问题和背景

0/1多维结包问题是具有m个约束条件的0/1结包问题，这使得它很难用传统方法解决。0/1多维结包问题可以表述为。给定n个对象，每个对象都有一个值 $v_i$ ，m个约束条件（或knapsack），每个约束条件都有一个能力约束 $c_j$ ，最大化值，使m个约束条件都得到满足。这m个约束条件中的每一个都有i个权重与之相关。这使得它成为一个一般的0/1整数编程问题。

最大化。  $\sum_{i=1}^n x_i v_i$

这样一来。  $\sum_{i=1}^n x_i w_{i,j} \leq c_j, j: 1 \dots m$

$x_i \in \{0,1\}$

我们的算法产生的初始种群选择一个对象的概率为0.5。在一个有n个对象的问题中，平均有n/2个对象被选中

。这个概率越高，算法收敛的速度就越快；但是。

这个概率越高，算法围绕贪婪估计收敛的机会就越大。这种生成初始种群的方式在种群中引入了很多无效的解决方案（噪音）。(初始群体生成的策略在[5]中讨论)。为了补偿无效的解决方案，我们研究了使用一个高度建设性的贪婪交叉。贪婪的交叉从父母那里获取具有最佳效用比的对象，并构建一个子代，使其始终是一个有效的解决方案。

我们使用拉格朗日乘法器来增加多维结包问题的效用比，根据以下步骤。

1. 对于每个对象和每个约束（对于该对象），权重（约束）值与相应的拉格朗日乘数 $l_j$ 相乘，得到这些值的总和。
2. 然后将步骤1中得到的数值除以约束条件的数量（可选步骤）。
3. 然后，得到价值（利润）和步骤2中得到的价值的比率，这就是该对象的利润-权重比率。

$$ratio_i = v_i / ((\sum_{j=1}^m l_j * w_{i,j}) / m)$$

其中 $l_j$ 是第 $j$ 个拉格朗日乘数， $m$ 是约束条件的数量。贪婪交叉法只是从两个亲本中按比率不递增的顺序抽取对象，并构建一个子代，使其满足所有约束条件。

### 3. 软件框架

我们的代码是用Java和C++编写的。任何JDK编译器和gcc都可以工作。我们在本文中使用的基准实例可以在[3]中找到。我们的代码可在[4]中找到。也可参见[8]。该代码要求在可执行文件所在的目录下有一个data.DAT 文件。改变 Constants.java 以增加或减少世代 的数量（Constants.GENERATIONS）。改变 Constants.java 中的数据文件名，使用一些其他的 Weing/Weish/Sento 文件。如果需要，将 GeneticAlgorithm.java中的DataProcessor改为ORLIB。

### 4. 实施和实证结果

我们在公开可用的实例中运行我们的算法。一些结果显示如下。更多的结果可以在我们的git库中找到[4]。我们的算法能够完全解决大多数实例，达到全局最优。

表-1：我们的算法适用于一些基准实例[3]。

实例	$m$	$n$	完全解 决	时间( 平均)	% 差距
森托1	30	60	13\20	4.8秒	0
森藤2	30	60	20\20	0.2秒	0
魏恩1	2	28	11\20	3.5秒	0

伟英5	2	28	20\20	0.6秒	0
伟英7	2	105	1\20	27.4秒	0
伟英8	2	105	11\20	2秒	0
炜仕05	5	30	20\20	0.02秒	0
炜炜10	5	50	16\20	0.1秒	0
炜炜15	5	60	16\20	1.2秒	0
炜仕20	5	70	10\20	10秒	0
炜仕25	5	80	10\20	1.9秒	0
炜炜30	5	90	13\20	3.6秒	0

$m$ 是约束的数量,  $n$ 是对象的数量。

## 5. 说明性的例子

我们的算法可以应用于任何0/1的整数编程问题, 而且效用比对于大多数类型的不等式约束来说是足够通用的。我们还没有尝试过在平等约束条件下运行我们的算法, 尽管它应该是微不足道的。我们的算法是快速的, 可以很快找到最优解。在我们的git资源库[4][8]中显示了有更多对象的实例。

## 6. 结论

传统的进化算法更适用于没有特定领域知识的问题。对于有部分领域知识的问题, 使用这种领域知识的遗传算法更有可能成功, 结果清楚地表明了这一点。一个好的搜索算法应该是全局性的, 并引入启发式算法来为算法提供建设性的指导。我们引入了一种新的贪婪交叉技术; 它构成了我们遗传算法的核心。如表1所示, 我们的算法能够在很短的时间内解决所有的实例, 达到最佳状态。有些问题如Weing7就比较难。未来的工作可能是在更大的实例上运行该算法, 这些实例的最优解是可用的。我们的算法是可并行的, 未来的工作可能是在Apache Spark或Map-Reduce上实现该算法。

## 参考文献

- [1] Chu, Paul C., and John E. Beasley."多维knapsack问题的遗传算法"。《启发式算法杂志》4.1 (1998) : 63-86。
- [2] Shah, Shalin."Genetic Algorithm for a class of Knapsack Problems." *arXiv preprint arXiv:1903.03494* (2019) 。
- [3] <http://people.brunel.ac.uk/~mastijb/jeb/orlib/files/mknapsack2.txt>
- [4] <https://github.com/shah314/gamultiknapsack>
- [5] "遗传算法初始种群生成方法的蒙特卡洛研究", R. Hill, 第31届冬季模拟会议论文集。仿真--通往未来的桥梁--第一卷, 1999年, 543-547 (1999年)

[6] "通过模拟退火进行优化", S. Kirkpatrick, 《科学》, 第4598号, 1983年5月13日, 第220卷, 4598, 671-680 (1983)。

[7] "与进化算法一起使用的理论和数值约束处理技术。A Survey of the State of the Art", C. Coello, Computer Methods in Applied Mechanics and Engineering, 191 (11-12), 1245-1287, January 2002 (2002)

[8] Shah, Shalin."GKNAP:用于解决多维结包问题的Java和C++包。"《开源软件杂志》4.42 (2019年) : 1756。

表2 - 代码元数据

AA A	代码元数据描述	<a href="https://github.com/shah314/gamultiknapsack">https://github.com/shah314/gamultiknapsack</a>
C1	当前代码版本	V1.7
C2	该代码版本使用的代码/资源库的永久链接	<a href="https://github.com/shah314/gamultiknapsack">https://github.com/shah314/gamultiknapsack</a>
C3	法律代码许可证	MIT许可
C4	使用的代码版本系统	指南针
C5	使用的软件代码语言	Java 和 C++
C6	编译要求、操作环境和依赖性	<p><b>Java实现</b></p> <p>我使用JDK 1.8对代码进行了测试，但任何JDK都应该工作正常。如果代码不能编译，请开一个问题。编译Java代码，然后运行GeneticAlgorithm。</p> <pre>javac *.java</pre> <p>java GeneticAlgorithm 文件名格式</p> <p>(文件名包含weing或orlib格式的实例) (格式是weing或orlib)。</p> <p>例子。</p> <pre>java GeneticAlgorithm data.DAT weing</pre> <p><b>C++实现</b></p> <p>该代码在使用gcc版本8的Mac上测试，使用homebrew下载。如果代码不能编译，请打开一个问题。编译C++代码，然后运行可执行文件。</p> <pre>g++ cmultiknapsack.cpp</pre> <p>./a.out 文件名格式</p> <p>(文件名包含weing或orlib格式的实例) (格式是weing或orlib)。</p> <p>例子。</p> <pre>./a.out data.DAT weing</pre> <p>(请删除data.DAT中的所有评论和其他无关的文字)。</p>
C7	如果有的话 链接到开发者文件/手册	<a href="https://github.com/shah314/gamultiknapsack">https://github.com/shah314/gamultiknapsack</a> 见[8]。
C8	支持问题的电子邮件	<a href="mailto:shah.shalin@gmail.com">shah.shalin@gmail.com</a>

