**基于多目标整数规划的**

**互联网购书决策问题求解**

**摘要**

随着电子商务行业的快速崛起，如何对各类经济问题做出快速决策，对于个人及企业有着重要意义。对于最优网购问题，在不同约束条件下最大化各类收益，并在环境及条件变化的情况下做出快速决策是本文需要解决的关键问题。

针对问题一，本文首先将购书数目作为决策变量，构建**多目标多变量的整数规划模型**。考虑到两个目标具有主次关系，通过赋予两个目标不同的权值将其转化为单目标规划模型，保证决策者在购买书籍数量最多的情况下尽可能多花费。由于该模型的解空间较为庞大，本文先通过**自适应遗传算法**迭代求出局部最优解，再将其作为**分支限界法**求解过程中的初始上界，进行剪枝以减小解空间的规模。最终获得了**购书数目13本，实际花费250元**的最优购书策略。

针对问题二，本文通过对决策目标的综合考虑，将两人合购书籍的问题转化为0-1整数规划模型。考虑到两人可能会采取共同购买和各自购买两种不同的策略，本文通过定义约束条件将两种购书策略下的模型统一，并通过两次最优化求解的过程解决了该多目标整数规划问题。最终得到两人**分别花费137.88元，127.596元，总计花费263.476元**的最优购书策略。

针对问题三，本文将不同网店促销力度作为灵敏度因子，对上述模型进行**灵敏度分析**。首先通过单独分析各网店促销力度的变化对决策的影响，依据使决策发生明显变化的灵敏度因子临界值和模型对不同因子的敏感性，确定后续分析过程中其取值个数及范围。为使决策者能在不同程度的促销力度下做出快速反应，本文还将各灵敏度因子组合，得到多网店促销力度下的**购书策略对比图**。最终，对不同策略进行定性分析，总结模型在不同网店促销力度下的购书策略变化规律。

本文引入新的概念，并建立逐层递进的模型，有效的解决了针对购书的多目标整数规划问题。同时，本文的决策模型具有较好的鲁棒性，根据实际情况更改约束条件及目标函数即可复用。

关键词：多目标整数规划模型 自适应遗传算法 分支限界法 灵敏度分析

**目录**

[一．问题重述 3](#_Toc70833413)

[1.1 问题背景 3](#_Toc70833414)

[1.2 问题提出 3](#_Toc70833415)

[二．问题分析 4](#_Toc70833416)

[三．模型假设 5](#_Toc70833417)

[四．符号说明 6](#_Toc70833418)

[五．问题一模型的建立与求解 7](#_Toc70833419)

[5.1 模型一的建立 7](#_Toc70833420)

[5.2 自适应遗传算法 9](#_Toc70833421)

[5.2.1 遗传算法简介 9](#_Toc70833422)

[5.2.2 遗传算法模型建立 10](#_Toc70833423)

[5.2.3 遗传算法模型的求解 11](#_Toc70833424)

[5.3 分支限界法求解 11](#_Toc70833425)

[5.4 模型求解 12](#_Toc70833426)

[六．问题二模型的建立与求解 13](#_Toc70833427)

[6.1 模型二的建立 13](#_Toc70833428)

[6.2 模型求解 15](#_Toc70833429)

[6.2.1 Lingo求解针对最小花销的整数规划模型 15](#_Toc70833430)

[6.2.2 Lingo求解在最小花销下的针对最低花销差的整数规划模型 16](#_Toc70833431)

[6.2.3 最终解答 16](#_Toc70833432)

[七．问题三模型的建立与求解 17](#_Toc70833433)

[7.1模型三的建立 17](#_Toc70833434)

[7.1.1 单网店促销力度参数改变模型建立 17](#_Toc70833435)

[7.2模型求解 18](#_Toc70833436)

[7.2.1 单网店促销力度参数改变模型求解 18](#_Toc70833437)

[7.2.2 多活动促销力度参数改变适应模型求解 24](#_Toc70833438)

[7.2.3 多网店促销力度改变模型灵敏度定性分析 28](#_Toc70833439)

[八．模型的评价与推广 29](#_Toc70833440)

[8.1 模型的评价 29](#_Toc70833441)

[8.1.1 模型的优点 29](#_Toc70833442)

[8.1.2 模型的缺点 29](#_Toc70833443)

[8.2 模型的推广 30](#_Toc70833444)

[参考文献 31](#_Toc70833445)

[附录 32](#_Toc70833446)

# 一．问题重述

## 1.1 问题背景

随着网络时代的发展和电子商务行业的壮大，网络购物凭借其简便快捷、购买体验良好等优势深受人们喜爱。此外，在网上购物还能够享受到线下购物所不具备的一些额外优惠。目前，网上购书已经逐渐取代了线下购书，成为越来越多人的购物选择。面对不同网店各式各样的优惠促销活动，通过决策规划制定合理的购买方案往往能够为消费者最大化优惠活动所带来的利益，以更小的花销购买到所需要的商品。此外，对于这一类互联网购物问题的研究还能够为销售商预测消费者可能采取的购物策略，从而修改活动促销方式并更改经营策略，以达到利润最大化。

## 1.2 问题提出

问题一：已知5家不同网店12种书目的价格和各网点包括满减、折扣、免运费、购物券等多种促销方式，需要在满足各类书购买至少两本的要求下，确定将所持有的250元尽可能花掉且购得尽可能多的书籍的购书策略。

问题二：在问题一的基础上，需要为两个舍友在各有持有250元且共享所购得的书籍，为其指定最经济且购得将书籍全部买下来的购书策略。

问题三：不同网店在活动期间可能通过加大促销力度吸引顾客购书，问题二中的两个舍友购书方案可能会因此发生改变。在问题二的求解基础上，通过分析不同促销力度对购书方案的影响，综合考虑以给出新的购书策略，并结合购书方案变化情况进行总结分析。

# 二．问题分析

对问题一，本文首先设置一系列整数决策变量及目标函数，将每种类型的书籍的购买数量作为模型的主要约束，从而构建满足约束下的多目标整数规划模型。考虑到小张会在保证书籍购买数量最多的前提下尽量花完预算金额，因此本文赋予两个目标不同的权值，将多目标整数规划模型转化为单目标规划模型。由于该问题参数过多导致解空间庞大，需要在最优化求解之前考虑通过遗传算法等启发式算法求解出局部最优解，以减小解空间数量，接下来再采用分支限界法等求解该问题。

对于问题二，由于两人需要使用最经济的方式购入图书并共享购得的图书， 通过分析可知只需购买特定书目的图书一本，因此可以构建出0-1规划模型，并将购齐所有书籍作为主要约束；此外，考虑到最经济的购书策略可能有多种，因此本文将两人的花销差值作为次要的优化目标。由于目标函数多个目标的权值参数无法确定，在两个目标中需要优先考虑总开销最小的情况下，本文将多目标规划转化为两个不同的单目标规划模型。首先寻找总花销最小的全局最优解，然后将该解作为第二个新模型的约束，得到满足总花销最小条件下下花销差值最小的解。

对于问题三，由于问题二构建的模型对于不同网店促销力度的灵敏程度各不相同。因此在进行灵敏度分析时，本文先分析不同网店的促销力度对购书方案的影响，寻找使购书策略发生变化的关键折扣程度；通过记录不同网店活动关键折扣力度，将不同的购书方案分至不同的折扣力度区间中。在实际情况中不同的网店在活动中很有可能以不同程度提高活动促销力度，因此为使得两位舍友快速反应，需要通过组合不同的促销力度给出不同情况下的购书策略，并对模型灵敏度进行分析总结。

# 三．模型假设

1.假设每家网店所有书籍库存充足；

2.假设快递费不会临时改变且配送人力始终充足；

3.假设全网读书日书券无法在本次订单中使用，但可以在其后的订单中使用；

4.假设除了各网店的促销之外不存在任何其他促销手段；

5.假设各网店的优惠政策不会在一次购买过程中发生改变；

6.假设各网点的促销力度加大并不会出现新的促销类型；

7.假设网店的书目质量、信用度不会对购物策略产生影响；

8.假设购书时的预算固定，不会发生变化。

# 四．符号说明

|  |  |
| --- | --- |
| 符号 | 意义 |
|  | 编号为的网店 |
|  | 从第家网店购买第种书目的数量 |
|  | 第家网店第种书目的价格 |
|  | 在第家网店购买的书目原价 |
|  | 第位室友在第家书店购买的书目原价 |
|  | 在第家网店购买的书目实际花费的总价 |
|  | 第位室友在第家书店购买的书目实际花费的总价 |
|  | 从网店S4所购买的最便宜的书目的价格 |
|  | 理学书所购买的总数量 |
|  | 工学书所购买的总数量 |
|  | 管理科学书所购买的总数量 |
|  | 个体生存能力权值 |
|  | 惩罚因子 |
|  | 第位室友从家网店购买第种书目的数量 |
|  | 第位室友购书的总花销 |
|  | 不同网店促销力度的灵敏度因子 |
|  | 购书策略改变率 |

注：本文中凡出现均表示第家网店，凡出现均表示第种书目，凡出现字母均表示第位室友

# 五．问题一模型的建立与求解

## 5.1 模型一的建立

通过对于问题一的分析可以看出，我们需要求解出在满足约束条件下的最优网购策略，也即需要建立多整数规划模型。

通过对所给的数据进行分析，本文中设表示从第家网店购买第种数目的数量。通过定量分析，我们可以确定任何一种书的购买总件数不可能大于9本，即：

|  |  |  |
| --- | --- | --- |
|  |  | （1） |

且三种书所购买的数量有以下关系：

|  |  |  |
| --- | --- | --- |
|  |  | （2） |

此外，需要注意到第4种书目仅在网店S4和S5出售，第8种书目仅在网店S2、S3和S4出售，因此有以下约束关系：

|  |  |  |
| --- | --- | --- |
|  |  | （3） |

记为第家网店中第种书目的价格，则在第家网店购买的书目原价为：

|  |  |  |
| --- | --- | --- |
|  |  | （4） |

此外，由于网店存在促销活动且购书不满60需要负担5元的快递费，因此实际所花费的价格与书店原价存在差异。

针对于不同网店的促销活动，我们有着以下的关系：

|  |  |  |
| --- | --- | --- |
|  |  | （5） |
|  |  | （6） |
|  |  | （7） |
|  |  | （8） |

其中表示从网店S4购买的最便宜的书目的价格。

|  |  |  |
| --- | --- | --- |
|  |  | （9） |

需要满足所花费的总金额不高于所持有的金额：

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

由于需要购买理学、工学和管理科学三种书目各至少两本，因此需要满足以下三个约束：

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

由于我们的策略为在满足约束条件下，优先选择购买尽可能多的书籍，其次则考虑尽可能将钱用尽。于是综合式（1）-（11），有如下的规划模型：

|  |  |  |
| --- | --- | --- |
|  |  | （12） |



## 5.2 自适应遗传算法

通过对时间复杂度的分析，我们会发现直接通过暴力DFS法求解上述规划约束模型，会因为巨大的解空间而无法在预期的时间内求解出最优解。因此，我们考虑先将目标放在求解出局部最优解，然后再通过其他策略从局部最优解向全局最优解的方向前进。

针对需要求解出局部最优解的需求，我们考虑采用启发式算法中的遗传算法。

### 5.2.1 遗传算法简介

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种模拟自然进化过程搜索相对最优解的过程。其基本步骤如下所示：

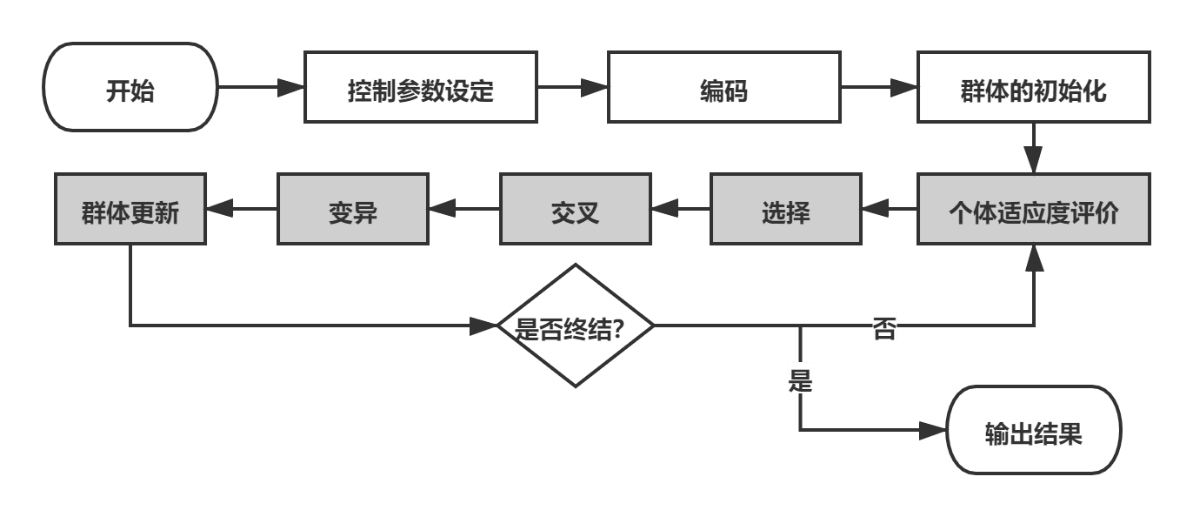


图1 遗传算法流程图

### 5.2.2 遗传算法模型建立

在遗传算法中，编码的意思就是把搜索空间解的表示映射成遗传空间解的表示。通常而言，编码有实数型编码和二进制编码两种。针对于本问题而言，由于购买书籍的数量满足式(1)，因此需要采用实数型编码方式，即将从不同网店购买的不同书目的购买数量编码成长度为60的染色体。如下图所示：

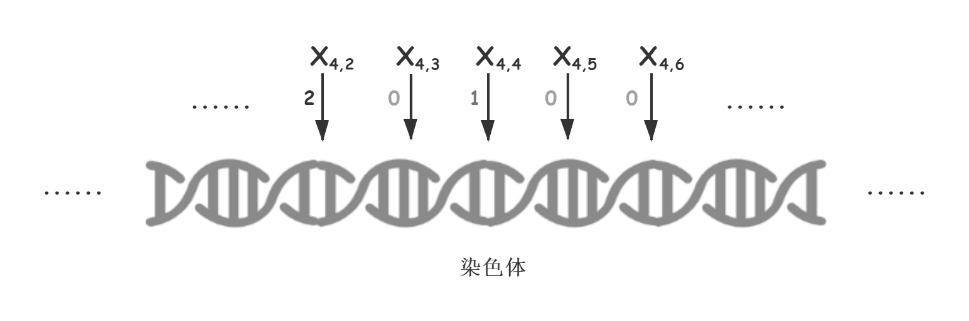


图2 染色体编码方式示意图

针对于初始种群的产生，我们通过随机数产生采取不同购书策略，并保证其满足约束条件。

在设计适应度函数时，除了保证目标函数达到最大以外，还需要考虑上述所定义的约束条件。为了防止单纯将不满足约束条件的个体所剔除可能导致的种群过于单一，我们在此令表示在任意一代群体中的干涉量值，构造适应度函数为：

|  |  |  |
| --- | --- | --- |
|  |  | （13） |

其中：是权值，是惩罚因子，。

针对于交叉选择和变异过程，传统的遗传算法由于采用固定的控制参数，导致了收敛速度相对比较慢且容易出现种群早熟现象，或很容易陷入局部最优解，失去遗传算法固有的优点。因此这里我们选用自适应遗传算法。即当种群适应度比较集中时，使交叉概率和变异概率增大；当种群适应度比较分散时，使和减小。即和的定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | （14） |
|  |  | （15） |

式中：为种群的适应度，为种群的最大适应度，为交叉双方是适应度较大的适应度，为个体的适应度，为常数。

当最优个体的适应度达到给定的阈值，或者最优个体的适应度和群体适应度不再上升时，或者迭代次数达到预设的代数时，算法终止。在此我们选取迭代次数为10000次。

### 5.2.3 遗传算法模型的求解

通过迭代，我们获取到了一组相对较优解，如下所示：

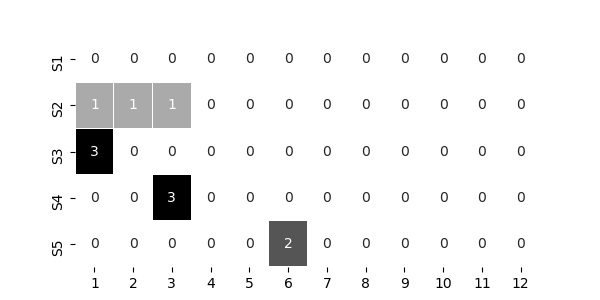


图3 遗传算法迭代局部最优解

将该组解其作为后续求最优解过程中的初始解，即可大大降低求解过程中可能存在的无效迭代，以加快运算速度。

## 5.3 分支限界法求解

在搜索算法中，分支限界法是可以减少解空间树的一种常见方法。在该方法中，每一个活结点只有一次机会成为扩展结点。活结点一旦成为扩展结点，就一次性产生其所有儿子结点。在这些儿子结点中，导致不可行解或导致非最优解的儿子结点被舍弃，其余儿子结点被加入活结点表。

与单纯的深度优先搜索或者回溯法不同，分支限界法会确定一个合理的限界函数，并根据限界函数确定目标函数的上界。接下来，按照广度优先搜索的策略遍历问题的解空间，在某一分支上依次搜索该结点的所有孩子结点，并分别估算这些孩子结点的目标函数的可能取值。如果某一孩子结点的目标值低于上界，则将其丢弃（其产生的解不会比当前解更好），否则加入到待处理表中。

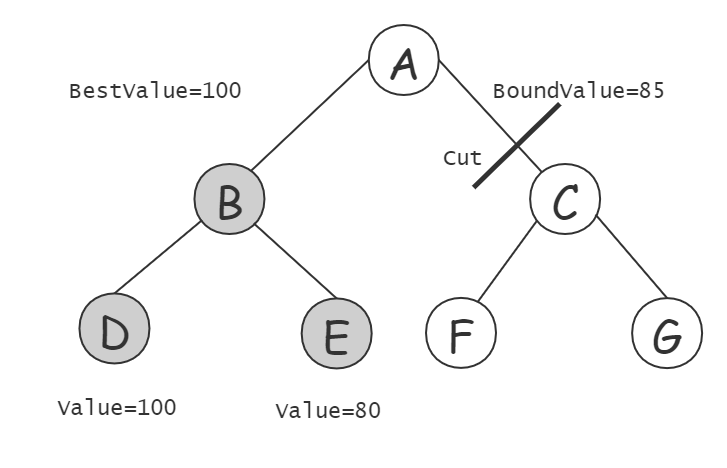


图4分支限界原理；当搜索到节点C时，

由于上界100大于预估上界85，故发生剪枝

我们将运用遗传算法所求出的局部最优解作为初始最优解，即可剪枝去除掉绝大多数的分支，从而大大降低了迭代次数。

为了提高效率，在此我们采用Lingo编写上述约束条件及目标函数，代码见附录代码2。选择策略为分支限界法(Branch & Bound)，我们求出了全局最优解。

## 5.4 模型求解

通过先采用自适应遗传算法求解出局部最优解，然后利用附录2部分的Lingo整数线性规划将其作为初始解进而求出全局最优解，我们得到问题一的解答如下所示：

表1问题一购物策略表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 书目 | A | B1 | B2 | B1+B2 | C1 | C2 | C3 | C1+C2+C3 | D | E | F | G |
| 网店S2 | 2 | 2 | 0 | / | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| 网店S4 | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

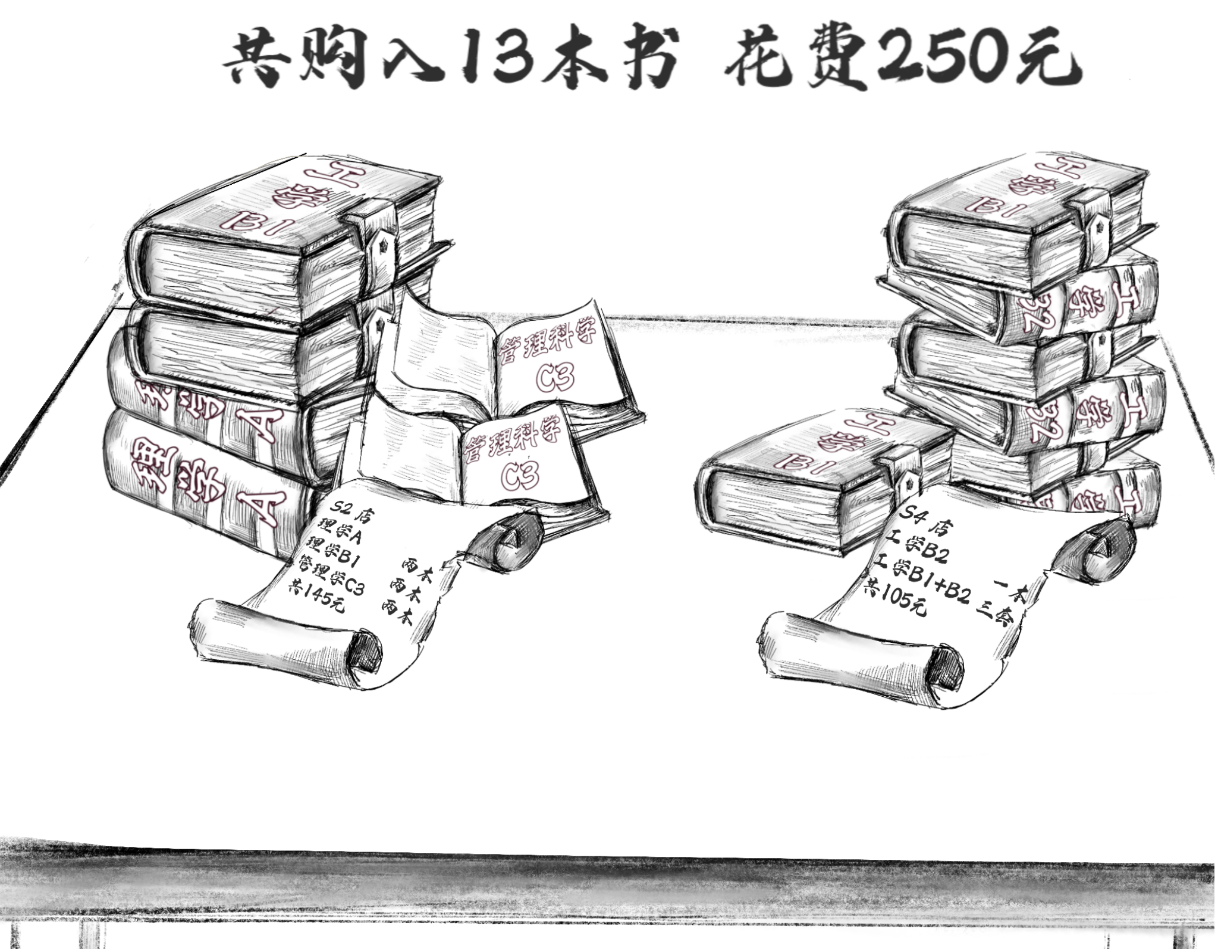


图5 问题一购物策略示意图

按照上述购书策略，小张可将250元全部花掉，并且购得了包括工学、理学和管理科学在内的共13本图书。

# 六．问题二模型的建立与求解

## 6.1 模型二的建立

本问题要求两人各拿250元购买书籍，并共享购得的图书，以最经济即花销最小的方式购入所有图书。对此他们有两种策略：第一种策略是两人合并钱款统一购买，从而最大限度的享受优惠；第二种策略是两人分开购买，最后共享所购得的图书。由于对两种策略分开求解过于繁琐，并且在考虑两人合并钱款购买时需要考虑到在同意书店分两次购买从而最大限度的利用优惠的情况，如两位室友在购买时可能会利用在S2书店购买所得的优惠券继续在S2书店购买，直接利用问题一的模型求解则会忽略掉这样的情况，导致可能忽略掉最优解。

因此本文将两人合并钱款一同购买等效为两人分别购买的情形，若两人分别在同一家书店购买书籍，则可以看作是两人合并钱款后为了获取更大优惠在同一家书店分两次购买；若两人合并钱款购买时，在一家书店最多只购买一次，此种情况同样可以用两人分别购买的情形刻画。

现令第一位室友的实际花销为，第二位室友的实际花销为。则需要满足下列约束关系：

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

由于我们将两种策略均按照分别购买后共享图书考虑，因此本文定义表示第位室友在第家书店购买第种书目的数量。由于小张的两位舍友希望用最经济即花销最小的方式把所有种类的书都买下来，因此他们不会重复购买同一书目的书籍，决策变量均为0-1整型变量，即：

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

对于该决策问题，首先要保证两位室友需要将所有书籍都买下来，即满足下述约束条件：

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

则两位室友在不同书店购买书目，使用优惠活动之前的原价为：

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

对于两位室友在每家书店购买书目的实际价格满足的约束条件和式（5）-（9）相同。综上，可以得出两位室友的实际总花销为：

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

本题要求两位室友以花销最小的方式购买所有书籍，因此目标函数为:

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

由于我们无法知道该最优解实际为两人合资购买还是分开购买后共享图书，因此对于上述目标函数求得最小值后，如果两人均有花销，则我们可以在两人的总花销最小的基础上尽量使两人的花销之差最小，即第二个目标函数:

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

根据以上分析可以得出两人购入书籍的多目标0-1整数规划模型：



## 6.2 模型求解

### 6.2.1 Lingo求解针对最小花销的整数规划模型

在建立了上述约束模型的基础上，我们通过Lingo求解该问题，得出两人在保证购入全部书籍下的最小花销为263.476元，其中两位室友的花销分别为106元和153.476元。

根据该结果可知两位室友分开购买书籍或者合资购买时在某家书店购买两次比合资且在书店仅购买一次更加划算。除此之外，根据此结果可得两人不同购书方案下总花销最小为263.476元，且为全局最优解。由于两人在该最低花销下的购书方案解可能不止一个，考虑实际情况，两人分开购买的花销相差越小越好。

### 6.2.2 Lingo求解在最小花销下的针对最低花销差的整数规划模型

由于已知两人购书的最小花销，为保证求两人最小花销差时总花销仍然为最小值，本文将6.2.1求得的最低花销作为新的约束条件，即：

|  |  |  |
| --- | --- | --- |
|  |  | （23） |

仍使用附录3的Lingo求解两人花销差的最小值，计算后可得两人的实际花销分别为137.88元和125.596元，总花销仍为263.476元。可知该解同时满足总花销最小和二人花销的差值最小的条件。

### 6.2.3 最终解答

根据运行结果，整理出如下图所示的购书策略清单：

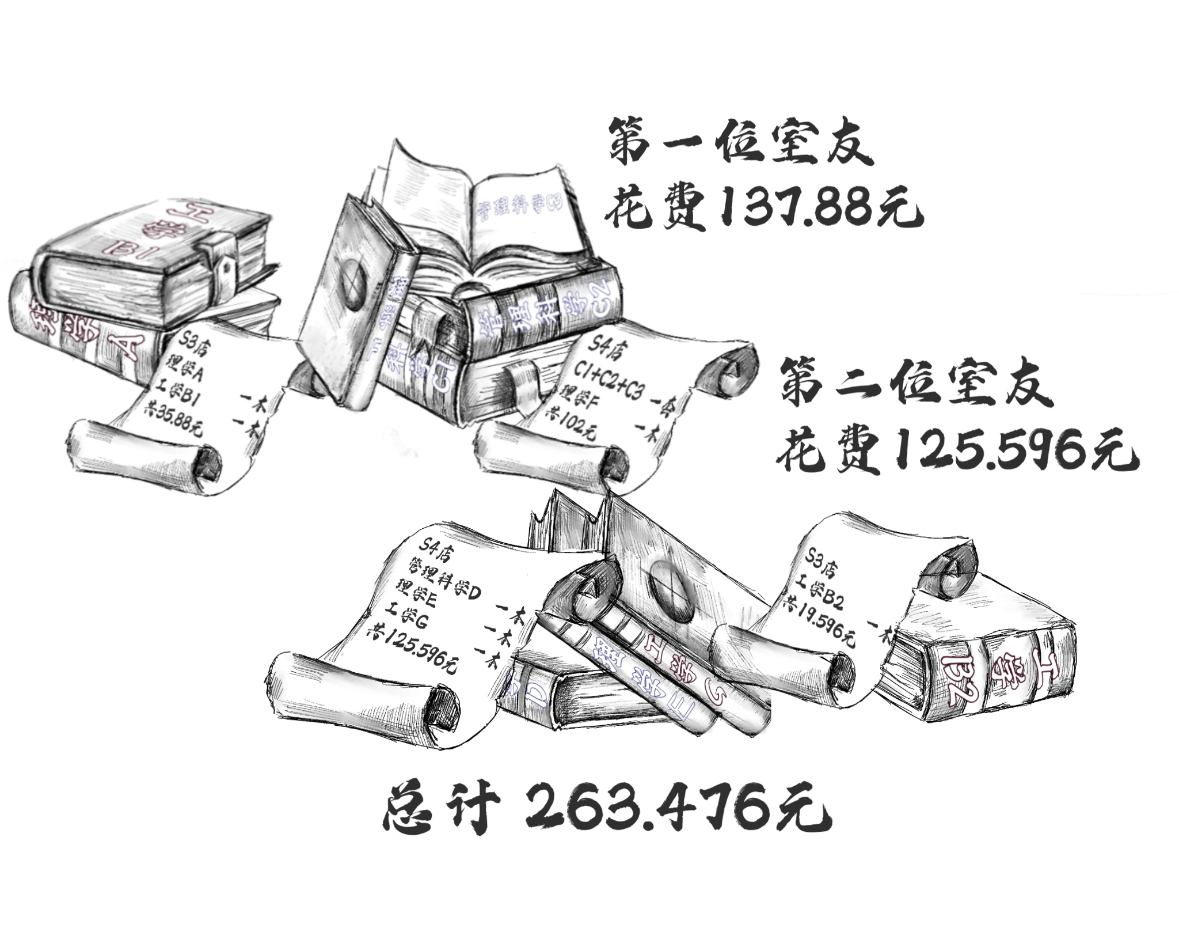


图6 问题二购书策略清单

其中，两位室友的购书策略如下:

表2 第一位室友购书策略

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 书目 | A | B1 | B2 | B1+B2 | C1 | C2 | C3 | C1+C2+C3 | D | E | F | G |
| 网店S3 | 1 | 1 | 0 | / | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 网店S4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

表3 第二位室友购书策略

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 书目 | A | B1 | B2 | B1+B2 | C1 | C2 | C3 | C1+C2+C3 | D | E | F | G |
| 网店S3 | 0 | 0 | 1 | / | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 网店S4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

可见两人分别花费137.88元和125.596元，使仅使用263.476元刚好购得了所有书籍并且两人的花费仅相差12.284元。

# 七．问题三模型的建立与求解

## 7.1模型三的建立

### 7.1.1 单网店促销力度参数改变模型建立

通过对问题3的分析可知，该问题是对上述问题2建立模型的参数进行灵敏度分析。因为实际情况中不同卖家可能根据售书情况调整活动促销力度，因此设定本问题中灵敏度因子分别为不同网店促销力度。

结合实际网店售书活动的促销力度数据考虑，网店促销力度参数的取值范围应是离散的，且考虑题目中所述促销力度稍大于预期而设定网店促销力度参数上界不超过原网店促销力度的两倍。

为通过购书策略对比图研究不同网店促销力度参数对模型购书策略的影响力，设定购书策略改变率为：

通过对活动促销力度参数设定调整，对问题2的模型进行反复带参求解，如下：

7.1.2 多网店促销力度参数改变模型建立

通过分析上述单网店促销力度参数改变对模型购书策略的影响，得到不同网店促销力度参数在设定离散范围内对模型具有显著影响的参数值，并依据模型对不同网店促销力度的灵敏度重新设定不同网店促销力度参数的个数与范围。最后根据不同的灵敏度因子组合重复对模型进行求解得到不同情况下的购书策略并分析结果。

## 7.2模型求解

### 7.2.1 单网店促销力度参数改变模型求解

单网店促销力度模型求解过程流程图如下：

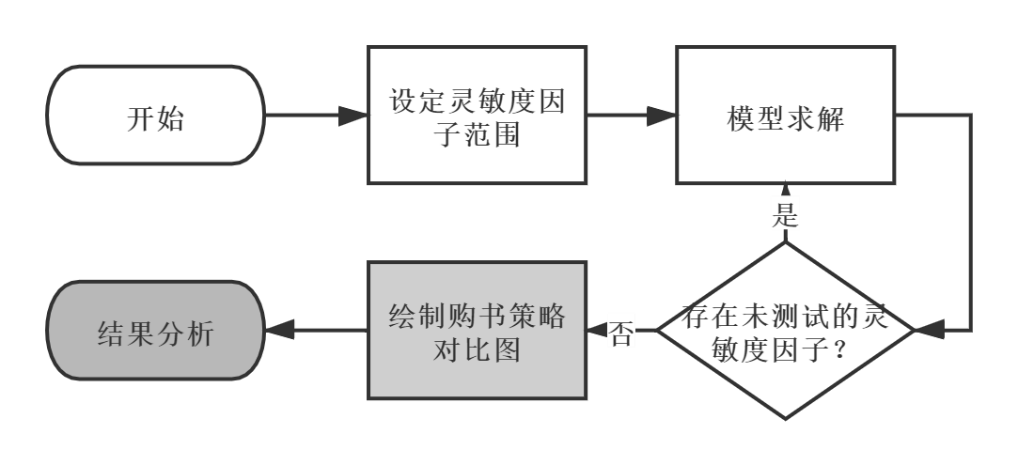


图7 单网店促销力度模型求解流程图

● 当只有S1网店促销力度发生不同程度的改变时，设参数范围对应如下：

将S1网店促销力度参数的不同组合带入上述建立的求解模型，得到不同组购书策略，将模型总结为购书策略对比图，对比图中纵坐标表示不同网店，横坐标为不同网店的书籍列表，图中黑格和灰格分别表示两位室友的选购图书策略，如下所示：

表4 灵敏度因子购书策略对比图

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |

分析数据发现S1网店的活动促销力度并不会影响购书策略发生变化，购书策略改变率，所以模型对于S1网店促销力度参数是稳定的。

● 当只有S2网店促销力度发生不同程度的改变时，设参数范围对应如下：

将不同S2网店促销力度参数带入上述建立的求解模型，得到不同组购书策略，将模型总结为购书策略对比图，如下所示：

表5 灵敏度因子购书策略对比图

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

分析购书策略对比图数据发现S2网店促销力度在不同阶段会对模型的购书策略产生影响。

表6 灵敏度因子购书策略分析表

|  |  |  |
| --- | --- | --- |
| 参数 |  |  |
| 购书策略对比图 |  |  |
|  |  | |
| 分析 | 在S2网店的赠送书券面额大于等于21元时，购书策略倾向于一个室友在S2网店购书达到活动参与最低要求，然后通过赠送的书券在S3网店购置剩余图书，另一个室友维持在S3与S4网店购置剩余图书。 | |
| 敏感度 | 模型对S2网店灵敏度因子**较敏感** | |

● 当只有S3网店促销力度发生不同程度的改变时，设参数范围对应如下：

将不同S3网店促销力度参数带入上述建立的求解模型，得到不同组购书策略，将模型总结为购书策略对比图，如下所示：

表7 灵敏度因子购书策略对比图

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |

分析数据发现S3网店的活动促销力度对模型的购书策略影响很大。

表8 灵敏度因子购书策略分析表

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 |  |  |  |
| 购书策略对比图 |  |  |  |
|  |  |  |  |
| 分析 | 两个室友分别在S4网店参与活动购买书籍，剩余书籍在S3网店购买 | 一个室友在S4网店购买书籍，其余在S3网店购买；另一个室友在S3网店购置书籍 | 两个室友全部在S3网店购置所需书籍 |
| 敏感度 | 模型对S3网店灵敏度因子**很敏感** | | |

● 当只有S4网店促销力度发生不同程度的改变时，鉴于S4网店促销活动形式，参数代表参与S4网店活动的最低消费金额，定参数范围对应如下：

将不同S4网店促销力度参数带入上述建立的求解模型，得到不同组购书策略，将模型总结为购书策略对比图，如下所示：

表9 灵敏度因子购书策略对比图

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |
|  |

分析数据发现S4网店的活动促销力度对模型的购书策略同样影响很大。

表10 灵敏度因子购书策略分析表

|  |  |  |
| --- | --- | --- |
| 参数 |  |  |
| 购书策略对比图 |  |  |
|  |  | |
| 分析 | 当活动参与限制调整为120元之下时，购书策略发生变化，其中两位室友虽然仍然集中于S3，S4网店购书，但参与活动限制的降低导致在不同网店购书的种类发生改变。 | |
| 敏感度 | 模型对S4网店灵敏度因子**敏感** | |

综上所述分析，两位室友的购书策略在各网店促销力度都不高的情况下更容易受到S3，S4网店促销活动的影响，S2网店促销力度达到一定程度也会影响购书策略。最后，基于以上对单网店促销力度参数改变模型的求解，本文推导出实际情况中多活动促销力度参数对模型购书策略的改变。

### 7.2.2 多活动促销力度参数改变适应模型求解

在上述求解过程中，通过购书策略的变化得到关键的促销力度参数，结合分析过程中模型对不同网店活动促销力度参数的敏感度不同，我们确定不同网店促销力度参数的个数与变化范围为：

表11 灵敏度因子取值表

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 敏感性 | 个数 | 范围 |
|  | 不敏感 |  |  |
|  | 不敏感 |  |  |
|  | 较敏感 |  |  |
|  | 很敏感 |  |  |
|  | 敏感 |  |  |

将不同促销力度参数组合带入上述建立的求解模型，得到不同的购书策略，实现流程图如下：

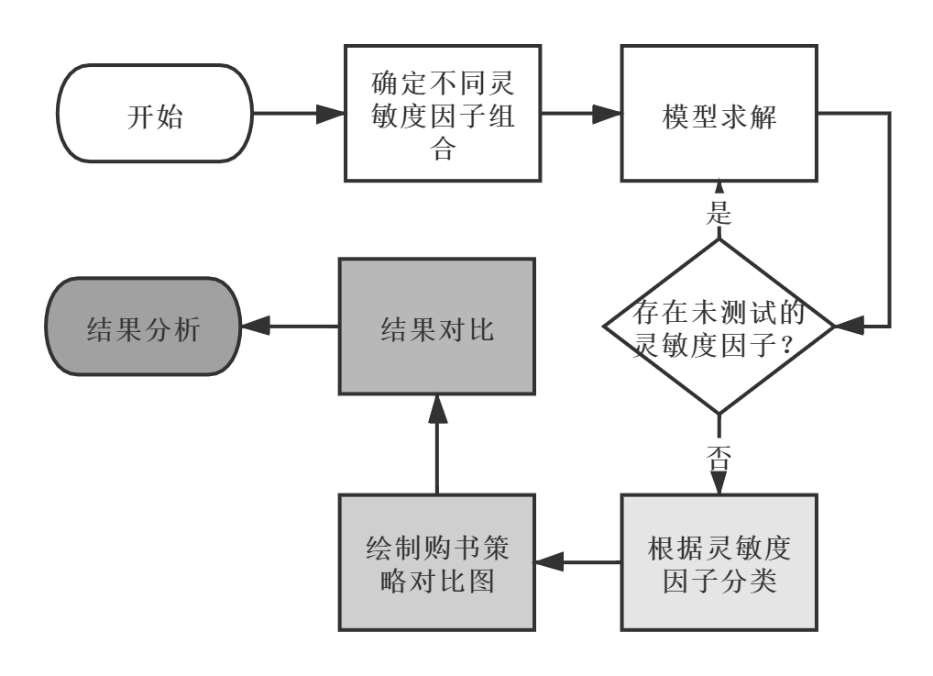


图8 多网店促销力度模型求解流程图

因模型对S3，S4网店促销力度参数变化相对更敏感，所以分别以不同的S2网店灵敏度因子建立S3、S4网店灵敏度因子对模型购书决策的共同影响，得到的不同购书策略对比图如下：

● S2网店促销方式：18元优惠券的情况

表12 时多灵敏度因子共同影响表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 125 | 118 | 113 | 105 |
| 0.88 |  |  |  |  |
| 0.85 |  |  |  |  |
| 0.82 |  |  |  |  |
| 0.79 |  |  |  |  |
| 0.76 |  |  |  |  |
| 0.73 |  |  |  |  |

● S2网店促销方式：21元优惠券的情况

表13 时多灵敏度因子共同影响表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 125 | 118 | 113 | 105 |
| 0.88 |  |  |  |  |
| 0.85 |  |  |  |  |
| 0.82 |  |  |  |  |
| 0.79 |  |  |  |  |
| 0.76 |  |  |  |  |
| 0.73 |  |  |  |  |

● S2网店促销方式：24元优惠券的情况

表14 时多灵敏度因子共同影响表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 125 | 118 | 113 | 105 |
| 0.88 |  |  |  |  |
| 0.85 |  |  |  |  |
| 0.82 |  |  |  |  |
| 0.79 |  |  |  |  |
| 0.76 |  |  |  |  |
| 0.73 |  |  |  |  |

通过以上不同灵敏度因子的组合，两位室友可以根据网店实际促销情况快速做出反应，更换当前选购策略以更经济的方式购书。

### 多网店促销力度改变模型灵敏度定性分析

**1.各网店促销力度参数在边界值时具有较强表现力与制约力**

在多活动促销力度参数下购书策略数据中，S3、S4网店活动促销力度参数达到边界值时发挥作用几乎等同于单网店活动促销力度参数的影响力度，但是两个参数相互之间会产生制约作用，最后抵消一定对模型的影响。

**2.S2网店促销力度参数在初期表现力较强**

通过上述模型求解分析可知，当S2网店促销力度参数达到21时，该灵敏度因子才对模型购书策略具有明显影响。通过分析多活动促销力度参数下购书策略数据，本文发现在S3、S4网店灵敏度因子更改不大时，S2网店灵敏度因子明显影响模型购书策略，在促销力度涨幅不大的初期具有良好的表现能力。

**3.S3网店活动促销参数对模型影响力略大于S4网店活动促销参数**

通过分析每张多活动促销力度参数下购书策略数据图，本文发现在S3、S4网店都有较大促销力度时，模型购书策略倾向于更多在S3网店完成购书，但是模型也会考虑室友中的一人选择S4网店促销活动完成购书目标。

综合上述灵敏度分析，可以在不同网店促销阶段总结如下购书策略方向：

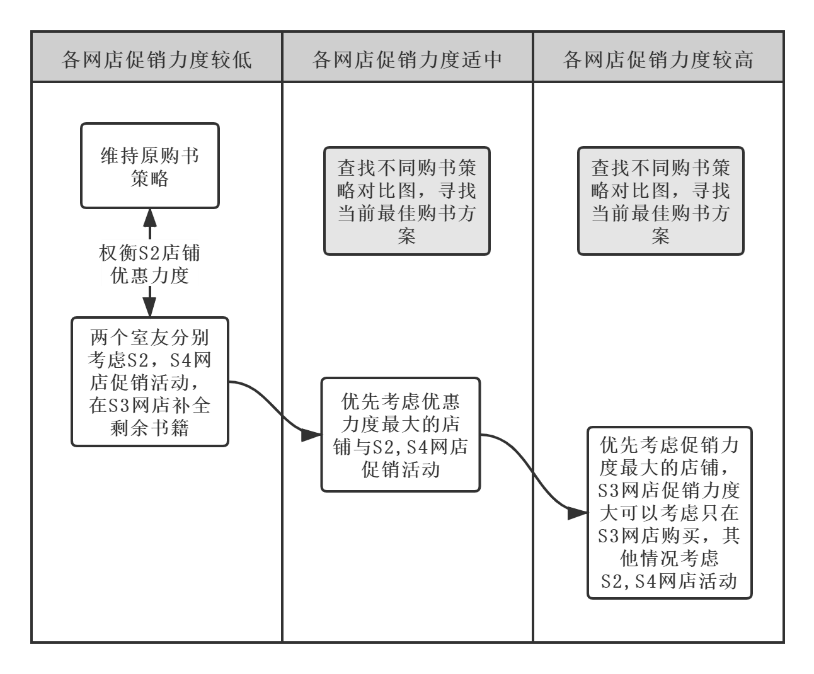


图9 不同网店促销阶段购书策略方向

# 八．模型的评价与推广

## 8.1 模型的评价

### 8.1.1 模型的优点

● 模型一（整数规划模型）：

○ 模型综合考虑了所有可能存在的购书过程中可能存在的所有优惠情况。

○ 模型在求解过程中运用生物学上的遗传的概念，使用概率机制进行迭代，也能有效的获得最佳个体。然后再以此作为求解全局最优解过程中的初始参数，可以大大降低解空间的大小。

○ 模型可复用性高，能比较有效的解决同类型问题。

* 模型二（0-1规划模型）：

○ 在模型一的基础上根据约束条件进行了改进，对两人同时参与购书时的策略提供了理论依据。

○ 该模型也具有较好的鲁棒性，可在实际生活中作为多人购物时策略手段。

* 模型三（模型灵敏度分析）：

○ 在进行模型灵敏度分析时，考虑实际情况下多家网店活动促销力度都发生改变，并给出模型对不同灵敏度因子的敏感性。

○ 对不同活动促销力度组合下，本文给出新的模型购书策略，通过定量分析帮室友对变化做出快速反应。

○ 通过多活动促销力度参数下购书策略数据进行定性分析，总结购书规律与购书策略方向。

### 8.1.2 模型的缺点

模型一：该模型在建立约束时默认每家网店购买一次，然后计算所获得的优惠。针对于本问题模型具有较好的效果，但对于多次购买可能产生更优策略的情况未予以考虑。

模型二：该模型是直接将约束条件和目标函数作为Lingo求解的参数，可能会出现因为参数过多而导致运行效率低这一情况。

模型三：灵敏度分析过程中为适应问题时空局限性，仅选取灵敏度因子的关键参数值进行组合分析，所获得购书策略数据量相对较少。

此外，前两个模型均只考虑费用因素对于最优购物方案选择的影响，并没有考虑不同网店的书目质量、信用度等因素对购书策略所产生的影响，具有一定的局限性。

## 8.2 模型的推广

本文不仅在网购方面选择最优购物方案时可以使用，在日常生活中很多大型的市场购物也可以参考本文中的模型来制定较为合适的采购方案。同样对于销售商而言，他们在制定促销方案后，可以通过本文的模型进行先前期求解，预测购买者可能选取的最优购物方案，从而根据利益再对促销方案进行修改，以达到利润最大化。

由于模型的求解思路具有一般性，因此可以根据实际情况增加其他约束。例如，针对于多人同时购物的决策，根据需求增加变量即可复用。

# 参考文献

[[1]王倩,李风军.改进的自适应遗传算法及应用[J/OL].重庆师范大学学报(自然科学版):1-8[2021-05-02].http://kns.cnki.net/kcms/detail/50.1165.N.20210407.1457.016.html.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CAPJ&dbname=CAPJLAST&filename=CQSF2021040600I&v=0fwNwVwypzXalVh4cd%25mmd2BP%25mmd2B7cO6yKSyReSpPmBe4DC6F3Z9Dg4e1PPQQaLKdwIK4kt)

[[2]王茂萍,潘大志,冯世强,张琴.贪心核加速动态规划算法精确求解适用范围[J].软件导刊,2020,19(08):54-59.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFDLAST2020&filename=RJDK202008013&v=V81bKPUIWb%25mmd2BzjuOqSWqKI4pEJwJvnc1VSjbFznvyvmwHjrG5X2KwRB4myQ1Gjhc9)

[[3]陈天. 基于决策变量分类的大规模多目标进化算法研究[D].湘潭大学,2020.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CMFD&dbname=CMFD202101&filename=1020407706.nh&v=%25mmd2BMfps6owL05iQsFkE17Z84I1oW21SXxFGE2Z8vEuJ7UOTFm55QjBO8DxExf%25mmd2FceMQ)

[[4]牛晓霞.基于Lingo软件的分枝定界法的验证求解优化设计[J].现代商业,2014(35):273-274.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFDLAST2015&filename=XDBY201435150&v=6uE5c84lexegQBNDGuls6C2BWc%25mmd2BGDjmoveJOlyO0%25mmd2BSl4QpRfpICwPLeOckRfW2Zf)

[[5]高娟,徐裕生,卫龙飞.基于自适应遗传算法的一类非线性规划问题的新算法[J].河北工业科技,2009,26(06):461-463.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFD2009&filename=HBGY200906001&v=mRNqyVUxiK8w3ZvZXvaN5J5NN0uFdMKpPQgIh7xR3fY1IhLXuv4IvDKFtOx54PRo)

[[6]王京辉,乔卫民.线性整数规划的分支限界解法及其MATLAB实现[J].计算机工程,2005(06):20-21+24.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFD2005&filename=JSJC200506007&v=09IGWp%25mmd2Fz2NpULlzwS70P%25mmd2BX0h7mh9VzjdQS%25mmd2Fbkwue03IlXrk1XX65q0uyvIq16wdK)

[[7]庞留勇,黄伟亮.线性规划多变量系数变化的灵敏度分析[J].天中学刊,2005(05):8-9.](https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFD2005&filename=TZXK200505003&v=MDCToEmUEKm8GzfwDMojDPZv%25mmd2Bbd9gLprLFXuu%25mmd2B3XXQERzUdb6Z%25mmd2FwadetQ5vr1Ag9)

# 附录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 程序编号 | 1 | 程序说明 | 遗传算法求局部最优解部分代码 | 编程语言 | Python |
| class **NGA**:      population=[]      dimension=1      bestPos=worstPos=0      mutationProb=10      crossoverProb=90      maxIterTime=10000      evalFunc=None      arfa =1.0      popu=2      def **\_\_init\_\_**(self,popuNum, dimension,evalFunc,crossoverProb=10,mutationProb=90,maxIterTime=1000):          for i in **range**(popuNum):              oneInd=Individual()              oneInd.eval=evalFunc(oneInd.chromsome)              self.population.append(oneInd)  **print**("初始种群",oneInd.eval)            self.crossoverProb=crossoverProb          self.mutationProb=mutationProb          self.maxIterTime=maxIterTime          self.evalFunc=evalFunc          self.popu=popuNum         self.dimension=dimension  *#评估函数*  def **evaluateFunc**(x):      ganshe=0 *#定义干涉量*  *#约束1:部分书籍无法购买*      if x[15]+x[16]+x[17]+x[35]+x[39]>0:          ganshe-=10000  *#约束2:每种书买两本*      if x[0]+x[1]+x[2]+x[3]+x[4]+x[45]+x[46]+x[47]+x[48]+x[49]+x[50]+x[51]+x[52]+x[53]+x[54]<2:          ganshe+=10000      if x[5]+x[6]+x[7]+x[8]+x[9]+x[10]+x[11]+x[12]+x[13]+x[14]+2\*(x[15]+x[16]+x[17]+x[18]+x[19])+x[55]+\          x[56]+x[57]+x[58]+x[59]<2:          ganshe+=10000      if x[20]+x[21]+x[22]+x[23]+x[24]+x[25]+x[26]+x[27]+x[28]+x[29]+x[30]+x[31]+x[32]+x[33]+x[34]+\          3\*(x[35]+x[36]+x[37]+x[38]+x[39])+x[40]+x[41]+x[42]+x[43]+x[44]<2:          ganshe+=10000      s1=25\*x[0]+19.5\*x[5]+22\*x[10]+36.1\*x[20]+32.5\*x[25]+33.6\*x[30]+40\*x[40]+55\*x[45]+30\*x[50]+48.5\*x[55];      s2=20\*x[1]+19.5\*x[6]+22.5\*x[11]+39\*x[21]+36\*x[26]+33\*x[31]+100\*x[36]+38.8\*x[41]+55.5\*x[46]+32.5\*x[51]+51\*x[56];      s3=19\*x[2]+20\*x[7]+21.3\*x[12]+35.8\*x[22]+34\*x[27]+32\*x[32]+99.9\*x[37]+42\*x[42]+54\*x[47]+29.5\*x[52]+49.3\*x[57];      s4=22\*x[3]+21\*x[8]+20\*x[13]+40\*x[18]+35.4\*x[23]+35.4\*x[28]+35\*x[33]+102\*x[38]+43.5\*x[43]+56\*x[48]+31\*x[53]+50\*x[58];      s5=23\*x[4]+19\*x[9]+21.5\*x[14]+40\*x[19]+38.2\*x[24]+35.5\*x[29]+34\*x[34]+46\*x[44]+52\*x[49]+33\*x[54]+48.9\*x[59];      z1=0      if s1<=1:          z1=0      elif s1<60:          z1=s1+5      elif s1<100:          z1=s1      elif s1<200:          z1=s1-10      else:          z1=s1-25      z2=0      if s2<=1:          z2=0      elif s2<60:          z2=s2+5      elif s2<100:          z2=s2      elif s2<200:          z2=s2-15      else:          z2=s2-30      z3=0.92\*s3      z4=0      if s4<=1:          z4=0      elif s4<60:          z4=s4+5      elif s4<130:          z4=s4      else:          chepaest=0 *#最便宜的一本书*          if x[13]+x[18]>0:              cheapest=20          elif x[8]>0:              cheapest=21          elif x[3]>0:              cheapest=22          elif x[53]>0:              cheapest=31          elif x[33]+x[38]>0:              cheapest=35          elif x[28]>0:              cheapest=35.4          elif x[23]>0:              cheapest=35.4          elif x[43]>0:              cheapest=43.5          elif x[58]>0:              cheapest=50          elif x[48]>0:              cheapest=56          z4=s4-cheapest      z5=0      if s5<=1:          z5=0      elif s5<60:          z5=s5+5      else:          z5=s5  *#计算总花销*      w=z1+z2+z3+z4+z5  *#约束:总花销需要不超过250*      if w>250:          ganshe+=1000  *#计算书本总数*      book=0      for i in **range**(60):          book+=x[i]      book+=x[18]+x[19]+2\*(x[36]+x[37]+x[38])  *#目标函数*      return 100000\*book+w-ganshe | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 程序编号 | 2 | 程序说明 | 求解问题一整数规划模型 | 编程语言 | Lingo |
| model:  max=1000\*book+w;  @gin(x1);@gin(x2);@gin(x3);@gin(x4);  @gin(x5);@gin(x6);@gin(x7);@gin(x8);  @gin(x9);@gin(x10);@gin(x11);@gin(x12);  @gin(x13);@gin(x14);@gin(x15);@gin(x16);  @gin(x17);@gin(x18);@gin(x19);@gin(x20);  @gin(x21);@gin(x22);@gin(x23);@gin(x24);  @gin(x25);@gin(x26);@gin(x27);@gin(x28);  @gin(x29);@gin(x30);@gin(x31);@gin(x32);  @gin(x33);@gin(x34);@gin(x35);@gin(x36);  @gin(x37);@gin(x38);@gin(x39);@gin(x40);  @gin(x41);@gin(x42);@gin(x43);@gin(x44);  @gin(x45);@gin(x46);@gin(x47);@gin(x48);  @gin(x49);@gin(x50);@gin(x51);@gin(x52);  @gin(x53);@gin(x54);@gin(x55);@gin(x56);  @gin(x57);@gin(x58);@gin(x59);@gin(x60);  x1<=10;x1>=0;x2<=10;x2>=0;  x3<=10;x3>=0;x4<=10;x4>=0;  x5<=10;x5>=0;x6<=10;x6>=0;  x7<=10;x7>=0;x8<=10;x8>=0;  x9<=10;x9>=0;x10<=10;x10>=0;  x11<=10;x11>=0;x12<=10;x12>=0;  x13<=10;x13>=0;x14<=10;x14>=0;  x15<=10;x15>=0;x16=0;x17=0;  x18=0;x19<=10;x19>=0;x20<=10;  x20>=0;x21<=10;x21>=0;x22<=10;  x22>=0;x23<=10;x23>=0;x24<=10;  x24>=0;x25<=10;x25>=0;x26<=10;  x26>=0;x27<=10;x27>=0;x28<=10;  x28>=0;x29<=10;x29>=0;x30<=10;  x30>=0;x31<=10;x31>=0;x32<=10;  x32>=0;x33<=10;x33>=0;x34<=10;  x34>=0;x35<=10;x35>=0;x36=0;  x37<=10;x37>=0;x38<=10;x38>=0;  x39<=10;x39>=0;x40=0;x41<=10;  x41>=0;x42<=10;x42>=0;x43<=10;  x43>=0;x44<=10;x44>=0;x45<=10;  x45>=0;x46<=10;x46>=0;x47<=10;  x47>=0;x48<=10;x48>=0;x49<=10;  x49>=0;x50<=10;x50>=0;x51<=10;  x51>=0;x52<=10;x52>=0;x53<=10;  x53>=0;x54<=10;x54>=0;x55<=10;  x55>=0;x56<=10;x56>=0;x57<=10;  x57>=0;x58<=10;x58>=0;x59<=10;  x59>=0;x60<=10;x60>=0;  x1+x2+x3+x4+x5+x46+x47+x48+x49+x50+x51+x52+x53+x54+x55>=2;  x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+2\*(x16+x17+x18+x19+x20)+x56+x57+x58+x59+x60>=2;  x21+x22+x23+x24+x25+x26+x27+x28+x29+x30+x31+x32+x33+x34+x35+3\*(x36+x37+x38+x39+x40)+x41+x42+x43+x44+x45>=2;  s1=25\*x1+19.5\*x6+22\*x11+36.1\*x21+32.5\*x26+33.6\*x31+40\*x41+55\*x46+30\*x51+48.5\*x56;  s2=20\*x2+19.5\*x7+22.5\*x12+39\*x22+36\*x27+33\*x32+100\*x37+38.8\*x42+55.5\*x47+32.5\*x52+51\*x57;  s3=19\*x3+20\*x8+21.3\*x13+35.8\*x23+34\*x28+32\*x33+99.9\*x38+42\*x43+54\*x48+29.5\*x53+49.3\*x58;  s4=22\*x4+21\*x9+20\*x14+40\*x19+35.4\*x24+35.4\*x29+35\*x34+102\*x39+43.5\*x44+56\*x49+31\*x54+50\*x59;  s5=23\*x5+19\*x10+21.5\*x15+40\*x20+38.2\*x25+35.5\*x30+34\*x35+46\*x45+52\*x50+33\*x55+48.9\*x60;  z1=@if(s1#le#1,0,@if(s1#lt#60,s1+5,@if(s1#lt#100,s1,@if(s1#lt#200,s1-10,s1-25))));  z2=@if(s2#le#1,0,@if(s2#lt#60,s2+5,@if(s2#lt#100,s2,@if(s2#lt#200,s2-15,s2-30))));  z3=s3\*0.92;  p=@if((x14+x19)#gt#0,20,@if(x9#gt#0,21,@if(x4#gt#0,22,@if(x54#gt#0,31,@if((x34+x39)#gt#0,35,@if(x29#gt#0,35.4,@if(x24#gt#0,35.4,@if(x44#gt#0,43.5,@if(x59#gt#0,50,@if(x49#gt#0,56,0))))))))));  z4=@if(s4#le#1,0,@if(s4#lt#60,s4+5,@if(s4#lt#130,s4,s4-p)));  z5=@if(s5#le#1,0,@if(s5#lt#60,s5+5,s5));  w=z1+z2+z3+z4+z5;  w<=250;  book=x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+2\*x16+2\*x17+2\*x18+2\*x19+2\*x20+x21+x22+x23+x24+x25+x26+x27+x28+x29+x30  +x31+x32+x33+x34+x35+3\*x36+3\*x37+3\*x38+3\*x39+3\*x40+x41+x42+x43+x44+x45+x46+x47+x48+x49+x50+x51+x52+x53+x54+x55+x56+x57+x58+x59+x60;  init:  x3=1;x6=0;x13=0;x31=0;  x33=0;x1=0;x2=0;x4=1;  x5=0;x7=3;x8=0;x9=1;  x10=0;x11=0;x12=0;x14=5;  x15=0;x16=0;x17=0;x18=0;  x19=0;x20=0;x21=0;x22=0;  x23=0;x24=0;x25=0;x26=0;  x27=0;x28=0;x29=0;x30=0;  x32=2;x34=0;x35=0;x36=0;  x37=0;x38=0;x39=0;x40=0;  x41=0;x42=0;x43=0;x44=0;  x45=0;x46=0;x47=0;x48=0;  x49=0;x50=0;x51=0;x52=0;  x53=0;x54=0;x55=0;x56=0;  x57=0;x58=0;x59=0;x60=0;  endinit  end | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 程序编号 | 3 | 程序说明 | 求解问题二的0-1规划模型 | 编程语言 | Lingo |
| model:  data:  s1Discount1=@pointer(1);  s1Discount2=@pointer(2);  s2Discount=@pointer(3);  s3Discount=@pointer(4);  s4Discount=@pointer(5);  enddata  [OBJ] min=2\*(w1+w2)+@abs(w1-w2);  @bin(x1);@bin(x2);@bin(x3);@bin(x4);  @bin(x5);@bin(x6);@bin(x7);@bin(x8);  @bin(x9);@bin(x10);@bin(x11);@bin(x12);  @bin(x13);@bin(x14);@bin(x15);@bin(x16);  @bin(x17);@bin(x18);@bin(x19);@bin(x20);  @bin(x21);@bin(x22);@bin(x23);@bin(x24);  @bin(x25);@bin(x26);@bin(x27);@bin(x28);  @bin(x29);@bin(x30);@bin(x31);@bin(x32);  @bin(x33);@bin(x34);@bin(x35);@bin(x36);  @bin(x37);@bin(x38);@bin(x39);@bin(x40);  @bin(x41);@bin(x42);@bin(x43);@bin(x44);  @bin(x45);@bin(x46);@bin(x47);@bin(x48);  @bin(x49);@bin(x50);@bin(x51);@bin(x52);  @bin(x53);@bin(x54);@bin(x55);@bin(x56);  @bin(x57);@bin(x58);@bin(x59);@bin(x60);  @bin(y1);@bin(y2);@bin(y3);@bin(y4);  @bin(y5);@bin(y6);@bin(y7);@bin(y8);  @bin(y9);@bin(y10);@bin(y11);@bin(y12);  @bin(y13);@bin(y14);@bin(y15);@bin(y16);  @bin(y17);@bin(y18);@bin(y19);@bin(y20);  @bin(y21);@bin(y22);@bin(y23);@bin(y24);  @bin(y25);@bin(y26);@bin(y27);@bin(y28);  @bin(y29);@bin(y30);@bin(y31);@bin(y32);  @bin(y33);@bin(y34);@bin(y35);@bin(y36);  @bin(y37);@bin(y38);@bin(y39);@bin(y40);  @bin(y41);@bin(y42);@bin(y43);@bin(y44);  @bin(y45);@bin(y46);@bin(y47);@bin(y48);  @bin(y49);@bin(y50);@bin(y51);@bin(y52);  @bin(y53);@bin(y54);@bin(y55);@bin(y56);  @bin(y57);@bin(y58);@bin(y59);@bin(y60);  x1+x2+x3+x4+x5+y1+y2+y3+y4+y5=1;  x6+x7+x8+x9+x10+x16+x17+x18+x19+x20+y6+y7+y8+y9+y10+y16+y17+y18+y19+y20=1;  x11+x12+x13+x14+x15+x16+x17+x18+x19+x20+y11+y12+y13+y14+y15+y16+y17+y18+y19+y20=1;  x21+x22+x23+x24+x25+y21+y22+y23+y24+y25+x36+x37+x38+x39+x40+y36+y37+y38+y39+y40=1;  x26+x27+x28+x29+x30+y26+y27+y28+y29+y30+x36+x37+x38+x39+x40+y36+y37+y38+y39+y40=1;  x31+x32+x33+x34+x35+y31+y32+y33+y34+y35+x36+x37+x38+x39+x40+y36+y37+y38+y39+y40=1;  x41+x42+x43+x44+x45+y41+y42+y43+y44+y45=1;  x46+x47+x48+x49+x50+y46+y47+y48+y49+y50=1;  x51+x52+x53+x54+x55+y51+y52+y53+y54+y55=1;  x56+x57+x58+x59+x60+y56+y57+y58+y59+y60=1;  fee=5;  freeLimit=60;  s1=25\*x1+19.5\*x6+22\*x11+36.1\*x21+32.5\*x26+33.6\*x31+40\*x41+55\*x46+30\*x51+48.5\*x56;  s2=20\*x2+19.5\*x7+22.5\*x12+39\*x22+36\*x27+33\*x32+100\*x37+38.8\*x42+55.5\*x47+32.5\*x52+51\*x57;  s3=19\*x3+20\*x8+21.3\*x13+35.8\*x23+34\*x28+32\*x33+99.9\*x38+42\*x43+54\*x48+29.5\*x53+49.3\*x58;  s4=22\*x4+21\*x9+20\*x14+40\*x19+35.4\*x24+35.4\*x29+35\*x34+102\*x39+43.5\*x44+56\*x49+31\*x54+50\*x59;  s5=23\*x5+19\*x10+21.5\*x15+40\*x20+38.2\*x25+35.5\*x30+34\*x35+46\*x45+52\*x50+33\*x55+48.9\*x60;  z1=@if(s1#le#1,0,@if(s1#lt#freeLimit,s1+fee,@if(s1#lt#100,s1,@if(s1#lt#200,s1-s1Discount1,s1-s1Discount2))));  z2=@if(s2#le#1,0,@if(s2#lt#freeLimit,s2+fee,@if(s2#lt#100,s2,@if(s2#lt#200,s2-s2Discount,s2-2\*s2Discount))));  z3=s3\*s3Discount;  p1=@if((x14+x19)#gt#0,20,@if(x9#gt#0,21,@if(x4#gt#0,22,@if(x54#gt#0,31,@if((x34+x39)#gt#0,35,@if(x29#gt#0,35.4,@if(x24#gt#0,35.4,@if(x44#gt#0,43.5,@if(x59#gt#0,50,@if(x49#gt#0,56,0))))))))));  z4=@if(s4#le#1,0,@if(s4#lt#freeLimit,s4+fee,@if(s4#lt#s4Discount,s4,s4-p1)));  z5=@if(s5#le#1,0,@if(s5#lt#freeLimit,s5+fee,s5));  w1=z1+z2+z3+z4+z5;  w1<=250;  book1=x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+2\*x16+2\*x17+2\*x18+2\*x19+2\*x20+x21+x22+x23+x24+x25+x26+x27+x28+x29+x30  +x31+x32+x33+x34+x35+3\*x36+3\*x37+3\*x38+3\*x39+3\*x40+x41+x42+x43+x44+x45+x46+x47+x48+x49+x50+x51+x52+x53+x54+x55+x56+x57+x58+x59+x60;  sy1=25\*y1+19.5\*y6+22\*y11+36.1\*y21+32.5\*y26+33.6\*y31+40\*y41+55\*y46+30\*y51+48.5\*y56;  sy2=20\*y2+19.5\*y7+22.5\*y12+39\*y22+36\*y27+33\*y32+100\*y37+38.8\*y42+55.5\*y47+32.5\*y52+51\*y57;  sy3=19\*y3+20\*y8+21.3\*y13+35.8\*y23+34\*y28+32\*y33+99.9\*y38+42\*y43+54\*y48+29.5\*y53+49.3\*y58;  sy4=22\*y4+21\*y9+20\*y14+40\*y19+35.4\*y24+35.4\*y29+35\*y34+102\*y39+43.5\*y44+56\*y49+31\*y54+50\*y59;  sy5=23\*y5+19\*y10+21.5\*y15+40\*y20+38.2\*y25+35.5\*y30+34\*y35+46\*y45+52\*y50+33\*y55+48.9\*y60;  zy1=@if(sy1#le#1,0,@if(sy1#lt#freeLimit,sy1+fee,@if(sy1#lt#100,sy1,@if(sy1#lt#200,sy1-s1Discount1,sy1-s1Discount2))));  zy2=@if(sy2#le#1,0,@if(sy2#lt#freeLimit,sy2+fee,@if(sy2#lt#100,sy2,@if(sy2#lt#200,sy2-s2Discount,sy2-2\*s2Discount))));  zy3=sy3\*s3Discount;  p2=@if((y14+y19)#gt#0,20,@if(y9#gt#0,21,@if(y4#gt#0,22,@if(y54#gt#0,31,@if((y34+y39)#gt#0,35,@if(y29#gt#0,35.4,@if(y24#gt#0,35.4,@if(y44#gt#0,43.5,@if(y59#gt#0,50,@if(y49#gt#0,56,0))))))))));  zy4=@if(sy4#le#1,0,@if(sy4#lt#freeLimit,sy4+fee,@if(sy4#lt#s4Discount,sy4,sy4-p2)));  zy5=@if(sy5#le#1,0,@if(sy5#lt#freeLimit,sy5+fee,sy5));  w2=zy1+zy2+zy3+zy4+zy5;  w2<=250;  book2=y1+y2+y3+y4+y5+y6+y7+y8+y9+y10+y11+y12+y13+y14+y15+2\*y16+2\*y17+2\*y18+2\*y19+2\*y20+y21+y22+y23+y24+y25+y26+y27+y28+y29+y30  +y31+y32+y33+y34+y35+3\*y36+3\*y37+3\*y38+3\*y39+3\*y40+y41+y42+y43+y44+y45+y46+y47+y48+y49+y50+y51+y52+y53+y54+y55+y56+y57+y58+y59+y60;  data:  @pointer(6)=@status();  enddata  end | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 程序编号 | 4 | 程序说明 | 通过Python代码调用Lingo计算 | 编程语言 | Python |
| import sys  from pyLingo import \*  def **book**(s1Discount1,s1Discount2,s2Discount,s3Discount,s4Discount):  *#create Lingo enviroment object*      pEnv = lingo.pyLScreateEnvLng()      if pEnv is None:  **print**("cannot create LINGO environment!")  **exit**(1)  *#open LINGO's log file*      name='\_'+str(s1Discount1)+'\_'+str(s1Discount2)+'\_'+str(s2Discount)+'\_'+str(s3Discount)+'\_'+str(s4Discount)      cur\_name='book'+name+'.log'      errorcode = lingo.pyLSopenLogFileLng(pEnv,cur\_name)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#pass memory transfer pointers to LINGO*  *#define pnPointersNow*      pnPointersNow = N.array([0],dtype=N.int32)  *#@POINTER(1)*      s1Discount1\_cur = N.array([s1Discount1],dtype=N.double)      errorcode = lingo.pyLSsetDouPointerLng(pEnv, s1Discount1\_cur, pnPointersNow)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#@POINTER(2)*      s1Discount2\_cur = N.array([s1Discount2],dtype=N.double)      errorcode = lingo.pyLSsetDouPointerLng(pEnv, s1Discount2\_cur, pnPointersNow)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#@POINTER(3)*      s2Discount\_cur = N.array([s2Discount],dtype=N.double)      errorcode = lingo.pyLSsetDouPointerLng(pEnv, s2Discount\_cur, pnPointersNow)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#@POINTER(4)*      s3Discount\_cur = N.array([s3Discount],dtype=N.double)      errorcode = lingo.pyLSsetDouPointerLng(pEnv, s3Discount\_cur, pnPointersNow)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#@POINTER(5)*      s4Discount\_cur = N.array([s4Discount],dtype=N.double)      errorcode = lingo.pyLSsetDouPointerLng(pEnv, s4Discount\_cur, pnPointersNow)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#@POINTER(6)*      Status = N.array([-1.0],dtype=N.double)      errorcode = lingo.pyLSsetDouPointerLng(pEnv, Status, pnPointersNow)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#Run the script*      cScript = "SET ECHOIN 1 \n TAKE book.lng \n GO \n QUIT \n"      errorcode = lingo.pyLSexecuteScriptLng(pEnv, cScript)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)  *#Close the log file*      errorcode = lingo.pyLScloseLogFileLng(pEnv)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **print**("errorcode = ", errorcode)  **exit**(1)      if Status[0] == const.LS\_STATUS\_GLOBAL\_LNG:  **print**("\nGlobal optimum found!")      elif Status[0] == const.LS\_STATUS\_LOCAL\_LNG:  **print**("\nLocal optimum found!")      else:  **print**("\nSolution is non-optimal\n")  *#delete Lingo enviroment object*      errorcode = lingo.pyLSdeleteEnvLng(pEnv)      if errorcode != const.LSERR\_NO\_ERROR\_LNG:  **exit**(1)    if \_\_name\_\_ == '\_\_main\_\_':      rate=[0.1,0.2,0.3,0.4,0.5]      discount\_float\_2=[18,21,24] *#s2店铺浮动折扣*      discount\_float\_3=[0.88,0.85,0.82,0.79,0.76,0.73]  *#s3店铺浮动折扣*      boundary\_4=[125,118,113,105] *#s4边界浮动*      for i in discount\_float\_2:          for j in discount\_float\_3:              for k in boundary\_4:                  book(16,40,i,j,k)                  sys.stdin.read(1) | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 程序编号 | 5 | 程序说明 | 根据Lingo计算结果绘制热力图 | 编程语言 | Python |
| import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  import pandas as pd  import re  s2=[18,21,24] *#s2店铺浮动折扣*  s3=[0.88,0.85,0.82,0.79,0.76,0.73]  *#s3店铺浮动折扣*  s4=[125,118,113,105] *#s4边界浮动*  res=""  f, (ax1) = plt.subplots(figsize = (6,3),nrows=1)  for i in **range**(**len**(s3)):      for j in **range**(**len**(s4)):          with **open**('book\_16\_40\_18\_'+str(s3[i])+'\_'+str(s4[j])+'.log',"r") as fp:              res=""              res+=fp.read()              fp.close()              x=np.zeros(60,dtype=int)  *#正则表达式读取*              for k in **range**(1,61):                  pattern=re.search(r'X{}.\*?(\d)'.format(k),res,re.S)                  x[k-1]=pattern.group(1)  *#匹配Y*              for k in **range**(1,61):                  pattern=re.search(r'Y{}.\*?(\d)'.format(k),res,re.S)                  if int(pattern.group(1))!=0:                      x[k-1]=2              x=x.reshape((12,5)).T              cmap = sns.cubehelix\_palette(start = 1.5, rot = 3, gamma=0.8, as\_cmap = True)              sns.heatmap(pd.DataFrame(x,columns=[i for i in **range**(1,13)],index=['S1','S2','S3','S4','S5']),                          linewidths = 0.1, ax = ax1, vmax=2, vmin=0, cmap='gray\_r',cbar=False)                plt.savefig('pic//'+str(i+1)+','+str(j+1)+'.jpg') | | | | | |