

# 一类基于分式规划、DEA、博弈论的供应商选择、订单分配和议价模型研究<sup>1</sup>

李润泽<sup>a</sup>

[lirz@mail2.sysu.edu.cn](mailto:lirz@mail2.sysu.edu.cn)

中山大学数学学院

张景桐<sup>a b</sup>

[zhangjt39@mail2.sysu.edu.cn](mailto:zhangjt39@mail2.sysu.edu.cn)

中山大学管理学院

## 报告摘要

本文选取的论文利用直接建模法、数据包络分析（DEA）和 Nash 均衡建立了一个两阶段模型，为采购商进行供应商选择和订单分配、与供应商议价以确定最终价格提供了有效的解决方案，并通过数值实验说明模型合理性。本文在原论文的基础上，针对第一阶段模型提出了一种基于遗传算法和分式规划的算法，针对第二阶段模型提出了一种理论解法，并成功复现数值实验。本文对论文中涉及的数据包络分析（DEA）和 Nash 均衡等原理进行了数学原理上的说明与拓展，阐述了他们在原论文中的作用。此外，本文以该论文为主线，梳理了供应商选择与订单分配问题相关文献，发现层次分析法和目标规划都能够有效地从多维度对供应商进行评价，进而选出优质的供应商。最后，附录一阐述了代码包的使用事项。

关键词：供应商选择，订单分配，议价问题，分式规划，遗传算法、数据包络分析，Nash 均衡

---

<sup>1</sup> 基于“Supplier selection and order allocation using two-stage hybrid supply chain model and game-based order price”的报告，该论文在 2021 年刊登于 Operations research [\[1\]](#)

目录

1 引言 ..... 1

2 问题描述与模型..... 2

    2.1 问题描述 ..... 2

    2.2 模型建立 ..... 3

        2.2.1 模型假设与变量名 ..... 3

        2.2.2 阶段一： 供应商选择和订单分配模型..... 4

        2.2.3 阶段二： 供应商议价模型..... 7

3 模型构建及求解中的数学原理 ..... 8

    3.1 模型的求解原理 ..... 8

        3.1.1 第一阶段模型求解方法详述 ..... 9

            3.1.1.1 遗传算法过程详述 ..... 9

            3.1.1.2 原论文中使用的变量替换法 ..... 10

            3.1.1.3 用变量替换法处理目标函数 ..... 11

            3.1.1.4 非线性规划的求解 ..... 11

        3.1.2 第二阶段模型理论分析与解析解 ..... 13

    3.2 数据包络分析（DEA） ..... 14

        3.2.1 传统的 DEA 方法..... 14

        3.2.2 原论文中的 DEA 方法..... 16

        3.2.3 动态 DEA 方法..... 17

    3.3 Nash 均衡 ..... 18

4 数值实验 ..... 20

    4.1 数据来源、处理与展示..... 20

    4.2 阶段一的数值实验结果 ..... 21

        4.2.1 最优化供应链总开销的数值实验结果 ..... 21

        4.2.2 最优化供应商总效率的数值实验结果 ..... 22

        4.2.3 双目标综合优化结果 ..... 22

    4.3 阶段二的数值实验结果 ..... 23

        4.3.1 阶段二额外需要的参数选择说明 ..... 23

        4.3.2 阶段二的数值实验结果 ..... 24

        4.3.3 理论解和数值解的比较 ..... 25

    4.4 遗传算法结合分式规划的复现情况..... 25

4.4.1 两种不同的变量替换法求解结果 .....	25
4.4.2 将 $t$ 用 $qk$ 表示的必要性 .....	26
4.4.3 订单分配问题结果（分式规划、罚函数和梯度下降法） .....	27
4.4.4 供应商选择问题结果（遗传算法，以 4.4.3 中的最优化结果为适应度函数） ...	30
<b>5 供应商选择与订单分配问题综述 .....</b>	<b>31</b>
5.1 综述总览 .....	31
5.2 层次分析法的应用 .....	32
5.2.1 基本的层次分析法 .....	32
5.2.2 模糊集理论改进的层次分析法 .....	34
5.3 目标规划的应用 .....	35
5.3.1 目标的规划基本原理 .....	35
5.3.2 目标规划应用解决供应商问题 .....	36
<b>6 结论与待解决的问题 .....</b>	<b>37</b>
<b>参考文献 .....</b>	<b>38</b>
<b>附录一：程序说明 .....</b>	<b>39</b>
a. 文件夹 gams .....	39
b. 文件夹 matlab .....	39
c. 使用建议 .....	40
<b>附录二：遗传算法的完整求解结果 .....</b>	<b>40</b>

# 1 引言

供应链管理是运筹学的一个重要应用领域，具有较强的现实意义。现今的商业竞争不再是单个企业之间的竞争，而是供应链与供应链的协调竞争。下游的采购企业在进行采购时，会对上游的供应商进行严格把控与挑选。

原论文的研究与供应商的选择、订单分配和博弈议价相关，模型共分为两个阶段。第一阶段是采购商通过一种评价方法选取最优的一批供应商，并且合理地分配订单数量给对应的供应商；第二阶段是在第一阶段的基础上，即选定供应商之后，采购商与供应商协调采购价格使得买卖双方都能达到满意。

对于模型的建立，在第一阶段中，原论文使用直接建模的方法评估了供应链总开销，用数据包络分析（DEA）的方法评估了供应链的总效率，最后得到一个多目标非线性混合整数规划；在第二阶段中，原论文使用了博弈论中不对称 Nash 协商模型，构建了一个凸的非线性规划问题。本文除了对原论文中的模型建立进行了复述，还详细阐述了数据包络分析（DEA）和不对称 Nash 协商模型的数学原理、其在原论文中的应用以及一些更加深刻的运用方法（如动态 DEA）。

对于模型的求解，原论文直接使用了 lingo 求解器进行求解。本文首先使用了 lindoglobal 求解器对原论文数值实验进行复现，验证了原论文结果中大部分结果的正确性，发现了少数结果的不严谨之处。随后，针对第一阶段模型，本文站在分式规划的角度，提出了一种基于遗传算法、分式规划、罚函数和梯度下降的算法，并成功复现数值实验的结果（运行时间约两分四十秒）；针对第二阶段模型，本文提出了一种理论解法，并成功得到第二阶段模型的解析解，也成功复现了数值实验的结果

对于原论文中所提到问题的历史研究成果，本文在最后进行了综述性质的展示，列举了供应商选择与订单分配问题的历史研究方法和相关论文，并着重阐述了层次分析法和目标规划在相关问题中的应用。与原论文进行比较，本文发现层次分析法（AHP）和目标规划都能够有效地从多维度对供应商进行评价，并有效选择出优质的供应商

本文将按照如下顺序陈述：第二部分将对问题进行描述，并介绍原论文提出供应商选择、订单分配混合模型以及纳什博弈议价模型；第三部分将对论文使用的数学方法进行分析，从理论上详细讨论模型求解的算法设计、模型建立过程中数据包络分析（DEA）和博弈论的应用；第四部分将展示数值实验的结果，包括使用 lindoglobal 求解器的复现情况和使用自行设计的算法的复现情况；第五部分将对供应商选择和订单分配问题进行综述性陈述；第六部分将进行总结和阐述遗留问题。

## 2 问题描述与模型

本部分主要对原论文模型建立部分的内容进行总结和复述。

### 2.1 问题描述

原论文考虑一个由单一制造商（买方）和多个供应商（供应商）组成的买卖双方供应链模型，目的是选择一组合适的有效供应商并确定双方商定的价格。

模型分为两个阶段，在第一阶段中，买方根据市场研究和未来市场情况制定战略计划，确定年度需求目标。随后，买方寻找市场上可供选择的潜在供应商，并利用提出的混合模型对供应商们进行分析，站在效率最大化和成本最小化的角度，对供应商做出选择以及完成对供应商订单量的分配。

在第二阶段中，买方已经根据第一阶段的模型选择好所需的供应商，买卖双方进行价格谈判和讨价还价，以同时使卖方和买方的效用最大化，并最终降低买方的采购成本。在这个议价阶段，讨价还价模型是买方和卖方共同进行的。供应商试图通过观察对方的建议价格来平衡价格并签订合同。通过在买方预算约束和双方效用条件下实施该模型，买方将能够获得适当的价格，卖方将能够获得相应的利润。

图 1 对描述问题进行了总结，即研究的为买方根据需求选择一批供应商，并与供应商进行议价。

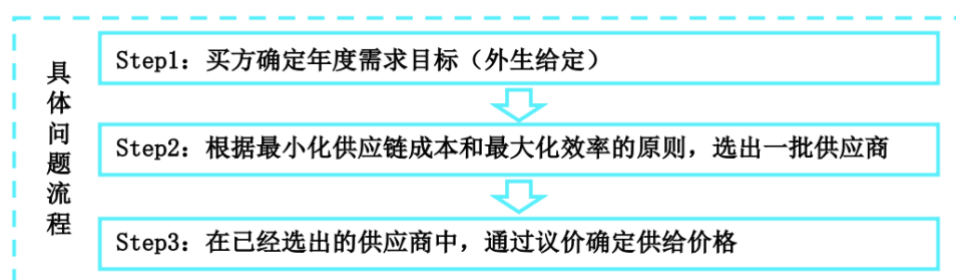


图 1 问题描述流程图

图 2 则展示了买方和卖方交互的方式，说明了第一阶段评估和选择供应商以及分配订单的过程，以及第二阶段在已经选择的供应商中通过讨价还价博弈定价的过程。

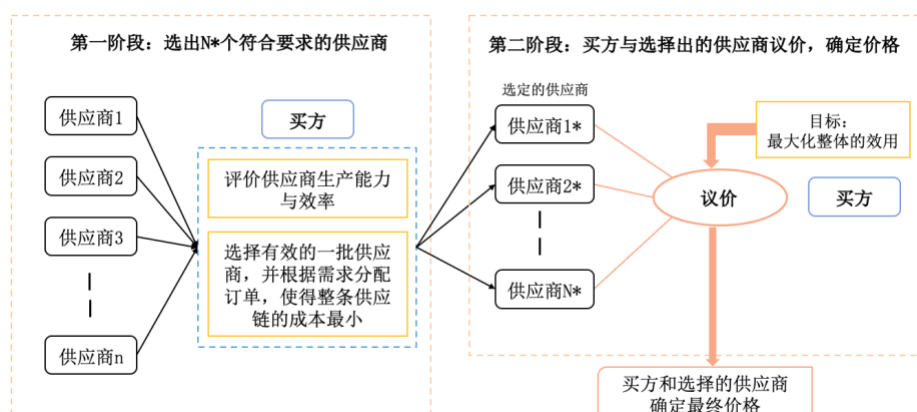


图 2 买方卖方交互图

## 2.2 模型建立

根据所提出的两阶段模型如图 3 所示，在第一阶段，提出了 MOMINLP 模型，它考虑了单个买方多供应商协调模型来选择有效的供应商，计算最优订货量，使供应链总成本最小化，使效率最大化。该混合模型同时进行供应商选择和订单分配。第二阶段，建立了买方价格与选定供应商价格之间的讨价还价模型。

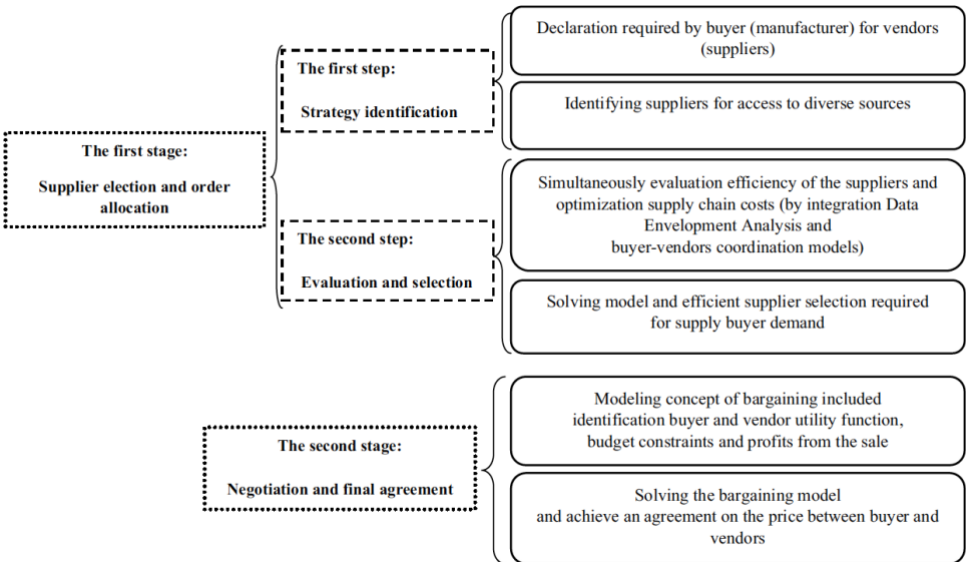


图 3 模型建立流程图

### 2.2.1 模型假设与变量名

本文建立如下假设：

假设	
1. 考虑由一个买方和多个供应商组成的两阶段供应链	
2. 买方和供应商不允许缺货	
3. 模型视为单个产品	
4. 年需求是已知的，并且随着时间的推移是恒定的	
5. 买方可从多个供应商处采购所需商品	
6. 考虑多周期模型	
7. 谈判阶段买方的年度预算有限	
8. 供应商年生产能力有限	
9. 建模开始时，没有选择供应商	
10. 持有成本与平均库存成比例	
下文中使用的符号以及他们的含义：	
第一阶段模型	第二阶段模型
指标	
供应商集合, $k = \{1, \dots, N\}$	供应商集合, $k = \{1, \dots, n\}$
候选供应商的输入集合 (DMUs), $i = \{1, \dots, m\}$	
候选供应商的输出集合 (DMU), $j = \{1, \dots, s\}$	

## 参数

$D$	买方的年度需求（外生给定，且不变）	$D$	买方的年度需求（外生给定，且不变）
$P_k$	第 $k$ 个供应商的年度生产率 $\sum_{k=1}^N P_k \geq D$	$P_k$	第 $k$ 个供应商的年度生产率 $\sum_{k=1}^N P_k \geq D$
$c_k$	第 $k$ 个供应商的单位价格	$c_k$	第 $k$ 个供应商的单位价格
$A_k$	第 $k$ 个供应商的下单价格	$A_k$	第 $k$ 个供应商的下单价格
$z_k$	第 $k$ 个供应商的变化成本（人力、材料等）	$z_k$	第 $k$ 个供应商的变化成本（人力、材料等）
$S_k$	第 $k$ 个供应商的建造成本	$S_k$	第 $k$ 个供应商的建造成本
$h_b$	买方在单位时间持有单位库存的成本	$h_b$	买方在单位时间持有单位库存的成本
$h_k$	第 $k$ 个供应商在单位时间持单位库存的成本	$h_k$	第 $k$ 个供应商在单位时间持单位库存的成本
$\varepsilon$	无穷小量	$q_k^*$	买方在每个周期需要在第 $k$ 个供应商处的购买数量定额（由模型一导出）
$I_{ik}$	第 $k$ 个供应商的第 $i$ 个投入	$q_k'^*$	基于买方的年度需求率，买方在每个周期需要在第 $k$ 个供应商处的购买数量定额（由模型一导出）
$O_{jk}$	第 $k$ 个供应商的第 $j$ 个产出	$B$	买方预算
$N$	买方角度下需要的供应商数量上线	$G_k$	第 $k$ 个供应商的最小利润期望
$q_k$	买方在每个周期需要在第 $k$ 个供应商处的购买数量	$c_k^*$	第 $k$ 个供应商通过博弈确认的单位商品价格
$q_k'$	基于买方的年度需求率，买方在每个周期需要在第 $k$ 个供应商处的购买数量定额（由模型一导出）	$U_B$	买方与供应商签订合同的成本
$y_k$	买方是否从第 $k$ 个供应商处购买的哑变量，购买设定为 1，不购买为 0	$U_k$	第 $k$ 个供应商的利润
$Q$	每个周期从所有供应商处下单的订单量		
$e_k$	第 $k$ 个供应商的无效率水平		
$V_{ki}$	第 $k$ 个供应商的第 $i$ 个投入的比重		
$U_{kj}$	第 $k$ 个供应商的第 $j$ 个产出的比重		

现在将供应商选择、订单分配以及定价问题抽象为数学模型

## 2.2.2 阶段一：供应商选择和订单分配模型

(1) 目的：供应商选择和订单分配

(2) 目标函数：a.最小化供应链的年度成本 $Z_1$ ；b.最大化所选供应商的效率 $Z_2$ ；c.最小化最终目标函数 $Z$ 。

### a. 最小化供应链的年度总成本 $Z_1$

供应链年度总成本（SCAC）为买方年度成本（BAC）和卖方年度成本（SSAC）之和；买方年度成本由购买成本（APC）、下单成本（AOC）以及库存成本（AIHC）构成，卖方成本由生产成本（ACP）、建造成本（SAC）和库存成本（ACIH）构成。

此处将介绍买方年度成本的由来，卖方年度成本的计算方式是类似的。

APC 取决于每单位商品的价格。由于时间范围内的周期数等于  $D/Q$ ，因此 APC 的公式为：

$$APC = \frac{D}{Q} \sum_{k=1}^n c_k q_k \quad (1)$$

在式(1)中，APC 等于所选供应商在不同时期基于采购数量的采购成本之和。此外，式

(2)表示供应链中的年度订购成本。

$$AOC = \frac{D}{Q} \sum_{k=1}^n A_k y_k \quad (2)$$

从式(2)中可以清楚地看出，如果买方决定从供应商 $k$ 订购，则在每个期间都会发生与该供应商相对应的固定订购成本。需要注意的是，在计算存货持有成本时，需要计算单位时间内的平均存货。单位时间的平均库存量是通过取每个周期的平均库存量除以周期长度（ $T$ ）得到的。从图 3 可以看出，在订货周期内，供应商 $k$ 的买方库存将从 $q_k$ 稳定地变为零，订货周期的长度为  $Q/D$ 。从供应商  $k$  收到的平均买方库存( $I_k$ )计算如下： $I_k = \frac{\frac{1}{2} \times q_k \times T_k}{T} = \frac{\frac{1}{2} \times q_k \times \frac{q_k}{D}}{\frac{Q}{D}} =$

$$\frac{q_k^2}{2Q}。$$

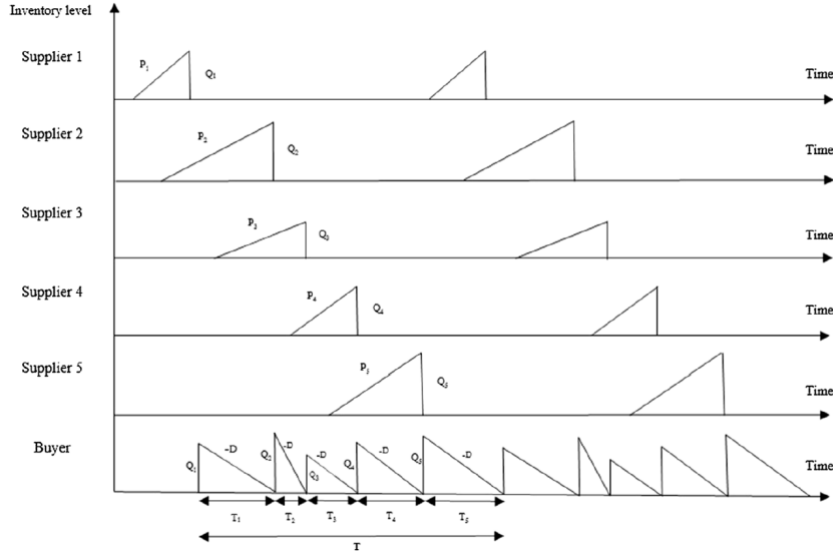


图 3 买方卖方的库存变化

根据买方单位时间的库存持有成本，从供应商 $k$ 收到的订单的 $AIHC_k$ 等于 $AIHC_k = h_b I_k = \frac{h_b}{2Q} q_k^2$ 。因此，买方从所有供应商收到的订单的 AIHC 可以写成关系式(3)。

$$AIHC = \sum_{k=1}^n AIHC_k = \sum_{k=1}^n h_b I_k = \frac{h_b}{2Q} \sum_{k=1}^n q_k^2 \quad (3)$$

计算买方的年成本如(4)：

$$BAC = APC + AOC + AIHC = \frac{D}{Q} [\sum_{k=1}^n (c_k q_k + A_k y_k)] + \frac{h_b}{2Q} \sum_{k=1}^n q_k^2 \quad (4)$$

总结起来，可以得到如下模型

$$SCAC = BAC + SSAC \quad (5)$$

$$BAC = APC + AOC + AIHC = \frac{D}{Q} [\sum_{k=1}^n (c_k q_k + A_k y_k)] + \frac{h_b}{2Q} \sum_{k=1}^n q_k^2$$

$$SSAC = ACP + SAC + ACIH = \frac{D}{Q} [\sum_{k=1}^n (z_k q_k + S_k y_k)] + \frac{D}{2Q} \sum_{k=1}^n \frac{h_k}{P_k} q_k^2 \quad (6)$$

$$\min \underbrace{Z_1}_{\text{供应链年度总成本}} = \underbrace{BAC}_{\text{买方年度成本}} + \underbrace{SSAC}_{\text{卖方年度成本}} \quad (7)$$



将具体的买方年度成本（BAC）和卖方年度成本（SSAC）代入，可得

$$\min \underbrace{Z_1}_{\text{供应链年度总成本}} = \frac{D}{Q} \left( \underbrace{\sum_{k=1}^n [(z_k + c_k)q_k]}_{\text{购买生产成本}} + \underbrace{(A_k + S_k)y_k}_{\text{下订单成本}} + \underbrace{\frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k^2}_{\text{库存总成本}} \right) \quad (8)$$

供应链总成本可以由上述公式表示，由购买生产成本、下订单成本和库存总成本构成，其中 $q_k$ 和 $y_k$ 是决策变量， $q_k$ 表示买方向第 $k$ 个供应商采购的订单量， $y_k$ 为0-1变量，表示是否向第 $k$ 个供应商进行采购

#### b. 最大化所选供应商的效率

通过 DEA 方法评估决策单元效率， $e_k$ 表示第 $k$ 家供应商的无效率值， $1 - e_k$ 则表示第 $k$ 家供应商的效率值

$$\max Z_2 = \sum_{k=1}^N (1 - e_k) \quad (9)$$

#### c. 最小化最终目标函数Z

$Z_1$ 、 $Z_2$ 的目标相悖，因此采用多目标规划的处理方法，将目标函数 $Z_1$ 和 $Z_2$ 进行类似归一化的处理，并乘上对应权重 $w_1$ 和 $w_2$ ，实际决策中可以根据决策者的意见调节目标函数  $Z$  中关于 $Z_1$ 、 $Z_2$  的权重。同时它是一个非线性规划模型。

$$\begin{aligned} \min Z &= w_1 \times \frac{Z_1 - Z_1^*}{Z_1^*} + w_2 \times \frac{Z_2 - Z_2^*}{Z_2^*} \\ &= w_1 \times \frac{\frac{D}{Q} (\sum_{k=1}^n [(z_k + c_k)q_k + (A_k + S_k)y_k + \frac{1}{2} (\frac{h_b}{D} + \frac{h_k}{P_k}) q_k^2]) - Z_1^*}{Z_1^*} \\ &\quad + w_2 \times \frac{\sum_{k=1}^N (1 - e_k) - Z_2^*}{Z_2^*} \end{aligned} \quad (10)$$

其中权重 $w_1$ 和 $w_2$ 满足 $w_1 + w_2 = 1$ ， $Z_1^*$ 和 $Z_2^*$ 为以 $Z_1$ 和 $Z_2$ 各自单独作为目标进行规划得最优值。

#### (3) 约束：

下面对每个约束的含义进行解释

$$Q = \sum_{k=1}^n q_k \quad (11)$$

$$q_k \leq \frac{y_k P_k}{D} Q, \forall k \quad (12)$$

约束(11)表示买方的总需求量与分配给每个供应商的单量之和相等，约束(12)表示分配给第 $k$ 个供应商的订单量不能够超过其最大生产能力。

$$q_k \geq \varepsilon y_k, \forall k \quad (13)$$

$$\sum_{i=1}^m V_{ki} I_{ik} = y_k, \forall k \quad (14)$$

$$\sum_{j=1}^s U_{kj} O_{jk} + e_k = y_k, \forall k \quad (15)$$

$$\sum_{i=1}^m U_{ki} O_{iz} - \sum_{j=1}^s V_{kj} I_{iz} \leq 0, \forall k; \forall z (k \neq z) \quad (16)$$

约束(13)中 $\varepsilon$ 为无穷小量,  $y_k$ 为 0-1 变量, 表示供应商 $k$ 是否被选择, 约束(13)是对 $y_k$ 的限制, 即分配订单 $q_k = 0$ 的供应商的 $y_k$ 不能为 1 (该供应商不被选到)。约束(14)至(16)则是采用 DEA 的方法对供应商 $k$ 的效率进行评价, 在后续 DEA 的数学原理中会着重分析。

$$\sum_{k=1}^n y_k \leq N \quad (17)$$

$$U_{kj} \geq \varepsilon y_k, \forall j, k \quad (18)$$

$$V_{ki} \geq \varepsilon y_k, \forall i, k \quad (19)$$

$$U_{kj} O_{jk} \leq y_k, \forall j, k \quad (20)$$

约束(17)限制选取的供应商个数不能超过 $N$ 个, 约束(18)至(19)表示第 $k$ 个供应商的投入输入和产出输出的评价权重是非负的。

$$y_k = 0, 1, \forall k \quad (21)$$

$$q_k \geq 0, \forall k \quad (22)$$

$$e_k \geq 0, \forall k \quad (23)$$

$$Q \geq \varepsilon \quad (24)$$

$$U_{kj}, V_{ki} \geq 0, \forall i, j, k \quad (25)$$

约束(21)表示是否选择供应商 $y_k$ 为 0-1 变量, 约束(22)和(23)保证订单量和效率是非负的, 约束(24)说明每个供应商的最大生产能力是大于 0 的。

### 2.2.3 阶段二：供应商议价模型

在协商环境下, 买方和卖方通过谈判定价, 最大化双方的利益。此时双方在决定价格时存在相反的利益, 又存在交易的动机, 因此会产生博弈。在通常情况下, 卖方会出让部分非常规利益给买方以获得更多的市场份额, 而买方则希望最优化成本, 这为讨价还价提供了可能。

(1) 目的: 定价

(2) 目标函数: 最大化博弈双方的效用 $Z_3$ , 根据对买方和卖方的议价能力调整模型( $w_B, w_k$ )

$$\max Z_3 = \max \left( \underbrace{B - U_B}_{\text{买方的效用}} \right)^{w_B} \prod_{k=1}^N \left( \underbrace{U_k - G_k}_{\text{第}k\text{个卖方的效用}} \right)^{w_k} \quad (26)$$

目标函数 $Z_3$ 以乘积的形式来衡量买卖双方总效用, 并通过 $w_B, w_k$ 来反应买方和卖方效用在总效用中的权重, 并且权重之和为常数 $C$ , 通常 $C$ 选取为 1, 即

$$w_B + \sum_{k=1}^N w_k = C \quad (26)$$

关于该目标函数的形式会在后续 Nash 均衡的数学原理进行具体阐述。

(3) 约束:

$$U_B = D \sum_{k=1}^N [c_k q_k^* + A_k b_k^* + \frac{1}{2} (\frac{h_b}{D}) q_k^* q_k^*] \quad (27)$$

约束(27)表示,  $U_B$ 为买方在采购中总的成本。

$$U_k = D \left( c_k q_k^* - \left[ z_k q_k^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{P_k} \right) q_k^* q_k^* \right] \right), k = 1, \dots, N \quad (28)$$

约束(28)表示,  $U_k$ 为第 $k$ 个供应商从销售中获得利润。

$$U_B \leq B \quad (29)$$

$$U_k \geq G_k, k = 1, \dots, N \quad (30)$$

$$U_B \geq \sum_{k=1}^N U_k \quad (31)$$

约束(29)表示, 买方采购总成本 $U_B$ 不能超过其最大资金约束 $B$ ; 约束(30)表示每个卖方从销售中获得利润 $U_k$ 不能低于其最低期望得到的利润 $G_k$ ; 约束(31)表示, 因为存在着库存和下单等成本, 使得买方在采购中总的成本 $U_B$ 大于所有卖方获利之和。

$$U_B \geq 0 \quad (32)$$

$$U_k > 0, k = 1, \dots, N \quad (33)$$

$$c_k > 0, k = 1 \dots N \quad (34)$$

约束(32)至(34)则是变量非零或大于零的约束。

### 3 模型构建及求解中的数学原理

本部分阐述模型构建中使用的方法——数据包络分析(DEA)和博弈论中的 Nash 均衡, 以及为了自行设计模型求解算法而涉及的非线性规划的数学原理。

#### 3.1 模型的求解原理

原论文中所有模型的求解都通过使用 lingo 求解器完成。为了实践课内外所学的优化方法, 本文针对本问题自行设计了求解算法。

第一阶段待求解的问题总结起来, 可以分为两步, 第一步是供应商的选择问题, 第二步是订单分配问题。

$$\min_{all \text{ var.}} Z = \min_{\{y_k\}} \min_{other \text{ var.}} Z$$

注意到如果供应商选择确定了, 那么订单分配问题将是一个连续型的非线性规划问题, 而对于纯整型但目标函数很复杂的订单分配问题, 遗传算法有着天然的优势。基于这个思想, 可以设计如下算法: 将二元变量序列  $\{y_k\}_{k=1}^N$  视为遗传算法中的染色体, 将订单分配问题的目标函数最优值作为染色体的适应度函数值, 然后迭代求解, 找到适应度最高的染色体。

订单分配问题是一个非线性规划, 更确切地说是一个分式规划。分式规划相较于一般非线性规划的特点是分母可能带来数值问题。例如在使用罚函数法或其他松弛约束的方法时, 如果惩罚力度不够, 将导致对分母不为 0 的约束 (一般分式规划都会有类似的约束, 除非分

母本身是一个非零函数)失效,进而导致求解出现错误。因此,处理分式规划的一种思路是使用变量替换的方法,将分式规划转化为一般的非线性规划。原论文中也使用了此方法,但考虑得并不全面,在 3.1.1.3 中本文将比较传统意义上和原论文中的变量替换法的区别。对于转换后的非线性规划,3.1.1.4 中将使用罚函数法和梯度下降法得到最优结果。

第二阶段待求解的问题是一个普通的非线性规划,与原论文不同的是,本文通过对模型进行理论分析,找到了模型的理论解,这将在 3.1.2 中详细介绍。

### 3.1.1 第一阶段模型求解方法详述

为了简明叙述过程,本节不考虑对供应商效率的优化问题而单纯地考虑供应链总成本的优化问题,即要求 $Z_1$ 的最优值 $Z_1^*$ 。因为前者无论是目标函数还是增加约束条件都是线性的,从求解难度上看不会造成任何影响。重新展示目标函数以及相关约束如下:

$$\min_{\text{供应链年度总成本}} Z_1 = \frac{D}{Q} \left( \sum_{k=1}^n \left[ \underbrace{(z_k + c_k)q_k}_{\text{购买生产成本}} + \underbrace{(A_k + S_k)y_k}_{\text{下订单成本}} + \underbrace{\frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k^2}_{\text{库存总成本}} \right] \right) \quad (35)$$

$$Q = \sum_{k=1}^n q_k \quad (11)$$

$$q_k \leq \frac{y_k P_k}{D} Q, \forall k \quad (12)$$

以及一些对变量非负性的要求。整体的算法的一个迭代过程可以总结为:1、生成初始染色体池,即随机选定一系列供应商;2、计算这些染色体的适应度,即在给定供应商的情况下求解订单分配问题使得供应链总开销最小;3、按照既定规则进行选择、交叉和变异。



图 4 阶段一模型求解算法总框架

#### 3.1.1.1 遗传算法过程详述

对供应商的选择,与自然选择中优胜劣汰的过程有一定程度的相似,因此使用遗传算法从直观上看是合适的。本问题中无需特意建立实际问题与染色体编码之间的联系,而自然地选择将 $\{y_k\}_{k=1}^N$ 作为染色体编码。如 $(1,0,0,0,0,1,1,0,1,0)$ 这么一个编码表示在十个供应商中选择第一个、第六个、第七个和第九个,而其他不选择。

对于每一个染色体,可以利用分式规划计算其适应度。注意到问题中对供应商选择的最大数量有限制 $N$ ,所以在计算适应度时,如果发现 $\sum_k y_k > N$ 可以直接将适应度设为 0。

对于遗传算法中的选择操作, 本文使用轮盘赌的方法。即计算染色体池中每条染色体的适应度在所有染色体适应度总和中所占比例, 并将这个比例作为选中该条染色体的概率, 然后进行随机选取至染色体池容量上限。比如染色体池中只有 $x, y$ 两条染色体, 计算他们的适应度为 $a, b$ 。那么按照 $(\frac{a}{a+b}, \frac{b}{a+b})$ 的概率分布在 $\{x, y\}$ 中选择两次, 其中选择 $x$ 的概率为 $\frac{a}{a+b}$ , 选择 $y$ 的概率为 $\frac{b}{a+b}$ 。

对于遗传算法中的交叉操作, 本文使用单点随机交叉。即将相邻的染色体两两配对, 对于每一对染色体, 他们有 $p_c$ 的概率进行交叉操作。过程是: 随机确定交叉点, 随后将交叉点后的部分染色体交换。如染色体池中有 $\{x_1, x_2, x_3, x_4\}$ 四条染色体, 先进行配对:  $\{(x_1, x_2), (x_3, x_4)\}$ 。对于染色体对 $(x_1, x_2)$ , 设他们为 $((0,0,0,0), (1,1,1,1))$ , 如果他们在第二个位点发生交叉, 那么交叉操作后的结果是 $((0,0,1,1), (1,1,0,0))$ 。

对于遗传算法中的变异操作, 本文使用基本位变异。即对于每一条染色体, 有 $p_m$ 的可能性发生变异: 随机指定一个变异点, 该点的数值由 0 变为 1 或由 1 变为 0。

经过这样三次操作后, 染色体池完成更新, 则可以重新计算染色体的适应度开始新一轮迭代。遗传算法只是实现整数规划的一种启发式算法, 除此之外还有更有理论保障的分支定界法和割平面法, 因此本文不将重心放于此, 而着重介绍适应度的计算方法, 即分式规划的求解。

### 3.1.1.2 原论文中使用的变量替换法

在正式介绍变量替换法前, 先阐述原论文中使用的方法。当染色体确定时,  $\{y_k\}_{k=1}^N$ 的值已经确定, 以 $J$ 表示 $\{y_k\}_{k=1}^N$ 中那些不为 0 的 $y_k$ 的下标构成的集合。模型可以化简为:

$$\min Z = \frac{D \sum_{k \in J} \left[ (z_k + c_k) q_k + (A_k + S_k) + \frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k^2 \right]}{\sum_{k \in J} q_k} \quad (36)$$

满足:

$$q_k \leq \frac{P_k}{D} \sum_k q_k, \text{ for all } k \in J \quad (37)$$

$$q_k > 0, \text{ for all } k \in J \quad (38)$$

为了处理分母, 设 $(\sum_{k \in J} q_k)^{-1} = t$ , 进行变量替换 $q'_k = q_k t$ , 则可以将模型转变为:

$$\min Z = D \sum_{k \in J} \left[ (z_k + c_k) q'_k + (A_k + S_k) t + \frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k q'_k \right] \quad (40)$$

满足:

$$q'_k \leq \frac{P_k}{D}, \text{ for all } k \in J \quad (41)$$

$$\sum_{k \in J} q'_k = 1 \quad (42)$$

以及非负性条件。可以发现, 进行变量替换后, 目标函数已不再是分式的形式, 转变为一般的非线性规划。但原论文在此处的考虑并不周全, 注意到变量替换后, 原变量 $q_k$ 并未被完全

消去。在约束条件中，作者简单地用 $\sum_{k \in J} q'_k = 1$ 替代 $q'_k = q_k t$ 来规避约束中的非线性情况，但这是不可取的。因为若如此，在新约束中对 $q_k$ 仅有非负性约束，在实际问题中，变量前的系数均为正数，那么在优化过程中可以直接将 $q_k$ 取为 0 而不会对约束和其他变量造成任何影响。数值实验的结果也证实了这一点。

如果需要对原论文的变量替换法进行补救，只能添加 $|J|$ 个约束条件 $q'_k = q_k t$ 。如此一来，不仅增加了不少的非线性约束，变量数也增加了很多，这并不令人满意。为了正确处理这个分式目标函数，下一节将介绍正规的变量替换方法。

### 3.1.1.3 用变量替换法处理目标函数

根据 I.M. Stancu-Minasian 的观点，如果用变量替换法处理一个分式规划 $\max q(x) = \frac{f(x)}{g(x)}$ ，需要找到一个（在定义域上）严格递增的函数 $\Phi_0: R \rightarrow R$ ，满足 $\Phi_0(t) > 0$ 对任意 $t > 0$ 成立。随后做变换 $F(y, t) = f(\frac{y}{t})\Phi_0(t)$ ， $G(y, t) = g(\frac{y}{t})\Phi_0(t)$ ， $y = tx$ 。对于约束也需要做一些变换，但由于本问题剩下的约束都是线性的，因此无需额外说明。变量替换后，需要求解的问题变为：

$$\max F(y, t)$$

满足：

$$\begin{aligned} G(y, t) &= \gamma \\ t &\geq 0 \end{aligned}$$

这里 $\gamma$ 需要用一些技术确定，本问题中 $\gamma$ 可以设为 1。当 $f(x)$ 和 $g(x)$ 是二次函数时， $\Phi_0(t)$ 一个通常的选择是 $\Phi_0(t) = t^2$ 。

具体到本问题，首先先对目标函数取倒数：

$$\max Z = \frac{\sum_{k \in J} q_k}{D \sum_{k \in J} \left[ (z_k + c_k)q_k + (A_k + S_k) + \frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k^2 \right]} \quad (43)$$

其次令 $\left( \sum_{k \in J} \left[ (z_k + c_k)q_k + (A_k + S_k) + \frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k^2 \right] \right)^{-1} = t^2$ ，并做变量替换 $q'_k = q_k t$

则问题变为：

$$\max Z = \sum_{k \in J} t q'_k \quad (44)$$

满足约束：

$$\sum_{k \in J} \left[ (z_k + c_k)q'_k t + (A_k + S_k)t^2 + \frac{1}{2} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k'^2 \right] = 1 \quad (45)$$

$$q'_k \leq \frac{P_k}{D} \sum_{k \in J} q'_k, \text{ for all } k \in J \quad (46)$$

以及一些非负性约束。

现在讨论一下此变量替换法和原论文中方法的优劣比较。可以发现，如果进行如此变量替换，替换后模型中已经不再出现原变量，这是变量替换法最基本应当达到的要求。使用本节的变换后，模型相较原来增加了一个变量和一个非线性约束，在接受范围内。

### 3.1.1.4 非线性规划的求解

为了求解上述非线性规划，罚函数法和梯度下降法是一种常见的手段。本节仅介绍理论上的算法，具体的参数选择问题，将在第四部分中叙述。

数值实验中发现，如果直接将约束作为惩罚项，将会为梯度下降法带来很大的困难。因为约束(45)的梯度非常大，这会导致梯度下降的主要方向变成“为了满足约束条件”的方向，并且迫使步长变得很短（约束(45)的凸性并不好，步长稍大将导致梯度爆炸），使得收敛变得很慢。但这是遗传算法的适应度函数，计算速度缓慢是不可取的。

柳暗花明之处在于，约束(45)恰好是关于 $t$ 的一个二次方程，于是可以将 $t$ 用 $q'_k$ 表示出来代入目标函数中：

$$t = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (47)$$

其中 $A = \sum_{k \in J} (A_k + S_k)$ ,  $B = \sum_{k \in J} (z_k + c_k) q'_k$ ,  $C = \frac{1}{2} \left( \sum_{k \in J} \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q_k'^2 \right) - 1$ 。注意到 $A$ 中不含 $q'_k$ ，于是这样操作后问题没有变回分式规划。

现在可以写出罚函数：

$$\min f = -t \sum_{k \in J} q'_k + \sigma \max\{C_i(x), 0\}^2 \quad (48)$$

其中 $C_i(x)$ 是线性的不等式约束，包括非负性约束。

$f$ 的梯度是易求的：

$$\nabla f = -\nabla t \sum_{k \in J} q'_k - t(1, 1, \dots, 1)_{|J|} + 2\sigma \max\{C_i(x), 0\} C_i \quad (49)$$

其中 $(1, 1, \dots, 1)_{|J|}$ 表示长度为 $|J|$ 元素均为1的向量，而 $C_i$ 表示第 $i$ 个线性不等式约束的系数构成的向量。此外，

$$\nabla t = \frac{-\nabla B + \frac{1}{2} \frac{(2B\nabla B - 4A\nabla C)}{\sqrt{B^2 - 4AC}}}{2A} \quad (50)$$

$$\nabla B = (z_k + c_k)_{k \in J} \quad (51)$$

$$\nabla C = \left( \left( \frac{h_b}{D} + \frac{h_k}{P_k} \right) q'_k \right)_{k \in J} \quad (52)$$

梯度已经全部算出，可以使用梯度下降法求解每一次迭代中罚函数的最小值，具体来说，可以使用下述算法：

---

#### 罚函数法算法

---

步骤一：取 $\sigma_1 > 0$ ，取允许误差 $\varepsilon > 0$ ，令 $k := 1$

步骤二：用梯度下降法求解无约束极值问题：

$$\min f = -t \sum_{k \in J} q'_k + \sigma_k \max\{C_i(x), 0\}^2$$

的最优解 $x^k = x(\sigma_k)$ 。

步骤三：检查约束条件是否满足：是否存在 $i$ 使得

$$C_i(x^k) > \varepsilon$$

若是，则取 $\sigma_{k+1} > \sigma_k$ 转到步骤二；否则迭代停止，得到最优解 $x^k$ 。

---

算法更具体的细节，如参数的选择等，将在数值实验部分 4.2.3 中叙述。

### 3.1.2 第二阶段模型理论分析与解析解

对于第二阶段的模型，在进行理论分析后，本文找到了模型的解析解。首先复述一下需要求解的模型：

$$\max Z_3 = (B - U_B) \prod_{k=1}^N (U_k - G_k) \quad (53)$$

满足如下约束：

$$U_B = D \sum_{k=1}^N [c_k q_k^* + A_k b_k^* + \frac{1}{2} (\frac{h_b}{D}) q_k^* q_k^*] \quad (27)$$

$$U_k = D \left( c_k q_k^* - \left[ z_k q_k^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{P_k} \right) q_k^* q_k^* \right] \right), k = 1, \dots, N \quad (28)$$

$$U_B \leq B \quad (29)$$

$$U_k \geq G_k, k = 1, \dots, N \quad (30)$$

$$U_B \geq \sum_{k=1}^N U_k \quad (31)$$

$$U_B \geq 0 \quad (32)$$

$$U_k > 0, k = 1, \dots, N \quad (33)$$

$$c_k > 0, k = 1 \dots N \quad (34)$$

注意到，站在实际的角度，约束  $U_B \geq \sum_{k=1}^N U_k$  是自然满足的，因为当给定具体参数可以分析参数正负性后，可以得知  $U_B$  相当于  $D \sum_{k=1}^N c_k q_k^*$  加上一个正数，而  $\sum_{k=1}^N U_k$  相当于  $D \sum_{k=1}^N c_k q_k^*$  加上一个负数。

留意  $Z_3$  的形式，它是  $N + 1$  个单项式的乘积，且这些单项式的和是一个常数（ $U_B$  中的  $c_k$  与  $\sum_{k=1}^N U_k$  中的  $c_k$  抵消了）。具体来说，这启发我们使用均值不等式：

$$\left[ (B - U_B) \prod_{k=1}^N (U_k - G_k) \right]^{\frac{1}{N+1}} \leq \frac{B - U_B + \sum_{k=1}^N U_k - G_k}{N + 1} \quad (54)$$

化简上式右端得：

$$RHS = \frac{B - D * \sum_{k=1}^N \left\{ \left[ A_k b_k^* + \frac{1}{2} \left( \frac{h_b}{D} \right) q_k^* q_k^* \right] + \left[ z_k q_k^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{P_k} \right) q_k^* q_k^* \right] + \frac{G_k}{D} \right\}}{N + 1} \quad (55)$$

确实是一个常数。

接着考察均值不等式等号成立的条件。注意到利用  $U_B \leq B, U_k \geq G_k, k = 1, \dots, N$  这两条约束，均值不等式要求的非负性自然满足。此外，均值不等式能取得等号当且仅当：

$$(B - U_B) = (U_k - G_k), k = 1, \dots, N$$

这里有  $N$  个方程，而且关于变量  $q_k$  都是线性的。方程的系数矩阵是可逆的，其具体形式如



下:

$$\begin{pmatrix} 2q_1^* & q_2^* & \cdots & q_N^* \\ q_1^* & 2q_2^* & \cdots & q_N^* \\ \vdots & \vdots & \ddots & \vdots \\ q_1^* & q_2^* & \cdots & 2q_N^* \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} \quad (56)$$

其中,

$$b_k = \frac{B + G_k}{D} + \left[ z_k q_k^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{P_k} \right) q_k^* q_k^* \right] - \sum_{k=1}^N \left[ A_k b_k^* + \frac{1}{2} \left( \frac{h_b}{D} \right) q_k^* q_k^* \right], k = 1, \dots, N \quad (57)$$

从上述方程中解出 $c_k$ , 即得到第二阶段的变量值。

## 3.2 数据包络分析 (DEA)

在 3.2 节中, 本文将循序渐进地对 DEA 方法进行分析, 首先介绍教材中传统的 DEA 方法, 以便读者对 DEA 有初步的了解, 再详细分析论文中使用的 DEA 方法, 最后则是介绍新出现的动态 DEA 方法。

### 3.2.1 传统的 DEA 方法

数据包络分析 (Data Envelopment Analysis, DEA) 是运筹学与管理科学交叉的一个新研究领域, 可以用来比较生产单位的效率。它是根据多项投入指标和多项产出指标, 利用线性规划对具有可比性的同类型单位进行相对有效性评价的一种数量分析方法。

下面将结合运筹学教材, 介绍最基本的 DEA 方法与思想。

设有 $n$ 个生产单位 (decision making units, DMU), 每个生产单位都有 $m$ 种输入, 以及有 $s$ 种输出。

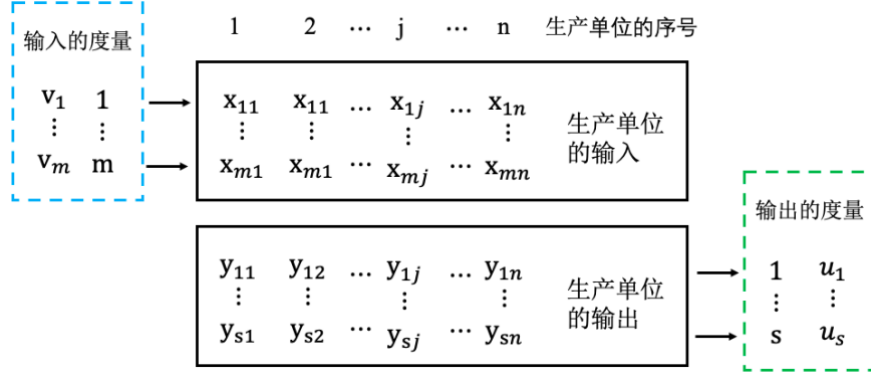


图 5 DEA 的流程

其中 $x_{ij}$ 为第 $j$ 个生产单位对第 $i$ 种输入的投入量,  $x_{ij} > 0$ ;  $y_{ij}$ 为第 $j$ 个生产单位对第 $i$ 种输出的产出量,  $y_{ij} > 0$ ;  $v_i$ 是第 $i$ 种输入的一种度量 (或者称权重),  $u_r$ 是第 $r$ 种输出的一种度量 (或者称权重)。

这里, 每个生产单位的输入向量  $X_j = (x_{1j}, x_{2j}, \dots, x_{mj})^T$  和输出向量  $Y_j = (y_{1j}, y_{2j}, \dots, y_{sj})^T$  为已知数据, 而输入的度量  $V = (v_1, v_2, \dots, v_m)^T$  和输出的度量  $U = (u_1, u_2, \dots, u_s)^T$  为决策变量。

对于第 $j$ 个生产单位有相应的效率评价指数

$$h_j = \frac{\sum_{r=1}^s u_r y_{rj}}{\underbrace{\sum_{i=1}^m v_i x_{ij}}_{\text{投入产出比}}}, \quad j = 1, 2, \dots, n \quad (58)$$

效率评价函数 $h_j$ 可以理解为第 $j$ 个生产单位的产出和（各项产出量乘对应的权重再求和）

$\sum_{r=1}^s u_r y_{rj}$ ，除以第 $j$ 个生产单位的投入和 $\sum_{i=1}^m v_i x_{ij}$ ，即 $h_j = \frac{\text{产出和}}{\text{投入和}}$ ，并且可以适当地选择度量系数 $V$ 和 $U$ ，使得 $h_j \leq 1, j = 1, 2, \dots, n$ ，即使得每个生产单元的效率评价都小于等于 1。

现在通过 DEA 方法来评价第 $k$ 个生产单元的效率。

$$\max h_k = \frac{\sum_{r=1}^s u_r y_{rk}}{\sum_{i=1}^m v_i x_{ik}} \quad (59)$$

$$\begin{cases} \frac{\sum_{r=1}^s u_r y_{rj}}{\sum_{i=1}^m v_i x_{ij}} \leq 1, j = 1, 2, \dots, n \\ V = (v_1, v_2, \dots, v_m)^T \geq 0 \\ U = (u_1, u_2, \dots, u_s)^T \geq 0 \end{cases} \quad (60)$$

在这个效率 $h_k$ 最优化模型中，约束条件分别表示每个生产单元的效率值小于等于 1，度量系数 $V$ 和 $U$ 非负。并且可以发现，度量系数 $V$ 和 $U$ 没有分量和为 1 的约束，这是因为， $V$ 和 $U$ 同时扩大某一倍数时，效率值 $h_k$ 不变。此外该分式规划可以通过变化转化为线性规划来解<sup>2</sup>。

如果该最优化模型能达到 $\max h_k = 1$ ，则说明第 $k$ 个生产单元是 DEA 有效的，对应的理解是能找到一组评价度量权重 $V$ 和 $U$ ，使得第 $k$ 个生产单元在这种评价方式下是有效率的；若 $h_k$ 达不到 1，则不认为第 $k$ 个生产单元是 DEA 有效的。

为了方便读者理解，下面给出一个例子。

有工厂 A、B 和 C，其生产输入为人力和矿石，其产出为铁和煤。

工厂		A	B	C
输入	人力	90	86	108
	矿石	64	99	99
输出	铁	252	282	294
	煤	223	287	317

表 1: DEA 的例子

上表的意思为工厂 A 用了 90 元的人力成本和 64 元的矿石成本，生产出了价值 252 元的铁和 223 元的煤；B 用了 86 元的人力成本和 99 元的矿石成本，生产出了价值 282 元的铁和 287 元的煤；C 用了 108 元的人力成本和 99 元的矿石成本，生产出了价值 294 元的铁和 317 元的煤

为了评价三家工厂相对效率值，采用 DEA 方法进行评价，通过 matlab 实现发现评价方式为输入度量 $V = (8, 4.3)^T$ 及输出度量 $U = (4, 0)^T$ 时，工厂 A 的效率值为 $h_A = 1$ ，工厂 B 的效率值为 $h_B = 1$ ，工厂 C 的效率值为 $h_C = 0.9003$ ，故在这种评价方式下，A 和 B 都是 DEA 有效的，而 C 的效率值不如 AB 二者。

<sup>2</sup> 同样可以使用变量替换法，取 $\Phi_0(t) = t$ 即可。

### 3.2.2 原论文中的 DEA 方法

在原论文中使用的 DEA 方法是传统方法的一种延伸修改，形式上虽有不同，但思想上基本一致，DEA 的核心思想是选取一种评价方式，即用一组度量向量 $V$ 和 $U$ 来评价生产单位的效率。此处截取论文中的公式分步进行阐述，将文章的 DEA 方法简化的规划问题记为 P1：

$$\begin{aligned} \max Z_2 = & \underbrace{\sum_{k=1}^n (1 - e_k)}_{\text{所有供应商的效率之和}} \\ \left\{ \begin{array}{l} \underbrace{\sum_i^m V_{ki} I_{ik}}_{\text{供应商的投入和}} = 1, \quad \forall k \\ e_k = \sum_{i=1}^m V_{ki} I_{ik} - \sum_{j=1}^s U_{kj} O_{jk}, \quad \forall k \\ \underbrace{\sum_{i=1}^m V_{ki} I_{iz}}_{\text{投入和}} - \underbrace{\sum_{j=1}^s U_{kj} O_{jz}}_{\text{产出和}} \geq 0, \quad \forall k, \forall z (k \neq z) \\ e_k, U_{kj}, V_{kj} \geq 0, \quad \forall k, \forall j \end{array} \right. \end{aligned}$$

其中的变量 $e_k$ 代表第 $k$ 个供应商的无效率值， $1 - e_k$ 则代表第 $k$ 个供应商的效率值， $V_{ki}$ 代表第 $k$ 个供应商的第 $i$ 项投入的度量权重， $U_{kj}$ 代表第 $k$ 个供应商的第 $j$ 项产出的度量权重； $I_{ik}$ 代表第 $k$ 个供应商的第 $i$ 项投入量， $O_{jk}$ 代表第 $k$ 个供应商的第 $j$ 项产出量。这里的目标函数是使得所有选中的供应商，即整条供应链上的供应商效率和最大化；第一个约束条件是在输入向量 $V$ 的度量下，供应商的投入和为 1；第二约束条件意思为第 $k$ 个供应商的无效率值 $e_k$ 为其投入和减去产出和，可以发现 $e_k \in [0, 1]$ ；

第三个约束条件意思为，用任何一个供应商 $k$ 的效率评价方式，即一组 $U_{kj}$ 和 $V_{ki}$ ，去评价其他供应商 $z$ 的效率时，都要满足供应商 $z$ 在这种评价衡量标准的投入和大于产出和，即供应商 $z$ 的效率是小于 1 的，保证每个供应商的效率评价方式都是合理的，不会使得用供应商 $k$ 的效率评价标准去评价供应商 $z$ ，出现供应商 $z$ 的效率值大于 1 的情况。如果没有第三个约束条件保证，则通过线性规划求解， $e_k$ 必然取到 0，即这种 DEA 是无意义的。

上述即对应论文中使用的 DEA 方法思想的解释，对整条供应链上的供应商效率之和的评价。

此外，如果在上述规划问题 P1 中将第一个约束条件换成 $\sum_i^m V_{ki} I_{ik} = y_k$ ，其中 $y_k$ 为 0-1 变量，表示供应商是否被选中，即没有选中的供应商不存在投入和（ $\sum_i^m V_{ki} I_{ik} = 0$ ），选中的供应商才能进行生产投入（ $\sum_i^m V_{ki} I_{ik} = 1$ ），并加上 $\sum_{k=1}^n y_k \leq N$ 的约束，即得到论文中完整的 DEA 方法对应的规划问题 P2：

$$\max Z_2 = \sum_{k=1}^n (1 - e_k)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^m V_{ki} I_{ik} = y_k, \quad \forall k \\ e_k = \sum_{i=1}^m V_{ki} I_{ik} - \sum_{j=1}^s U_{kj} O_{jk}, \quad \forall k \\ \sum_{i=1}^m V_{ki} I_{iz} - \sum_{j=1}^s U_{kj} O_{jz} \geq 0, \quad \forall k, \forall z (k \neq z) \\ \sum_{k=1}^n y_k \leq N \\ e_k, U_{kj}, V_{kj} \geq 0, \quad \forall k, \forall j \\ y_k \text{ 为 } 0-1 \text{ 变量} \end{array} \right.$$

规划问题 P2 意思为从所有的供应商中挑选最多 $N$ 个供应商，使得整个供应链上供应商的效率和最大，则通过 DEA 方法达到文章中挑选最优供应商的目的。

### 3.2.3 动态 DEA 方法

上述对该论文单周期静态的 DEA 进行了详细的分析，但是在实际商业世界中，买方选择供应商是一个长时间的问题，并不是基于单个周期供应商的效率值进行评价的，而是基于以往多周期历史数据进行评价的。

Tone 和 Tsutsui (2010) [2]提出了动态 DEA 方法 (Dynamic DEA, DDEA)，突破了单周期的评价效率的局限性，从多周期角度评价供应商的效率，这里仅介绍算法思想。

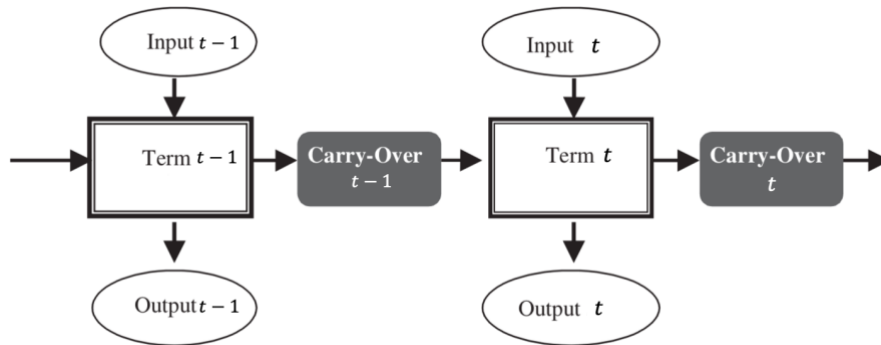


图 6 动态 DEA 算法思想

在评价供应商 $t$ 期的效率时，不仅考虑根据当前（即 $t$ 时）的投入与产出得到效率评价，也考虑前一期（即 $t-1$ 期）的效率评价。图中的 $Carry-Over(t-1)$ 为根据 $t-1$ 期评价影响，在考虑 $t$ 期时则会纳入 $Carry-Over(t-1)$ 进行综合评价，这样从最开始的评价会一直迭代到当前的评价，综合了多期的评价。

然而，[Tavana 和 Yousefi \(2017\) \[3\]](#)认为这样动态 DEA 方法仍然存在一定局限性，即过分依赖历史数据，不能够对未来效率进行预测，无法帮助企业选择下一期的供应商，因此基于动态 DEA 的思想，并结合目标规划提出了能分析供应商未来效率的方法，算法如下。

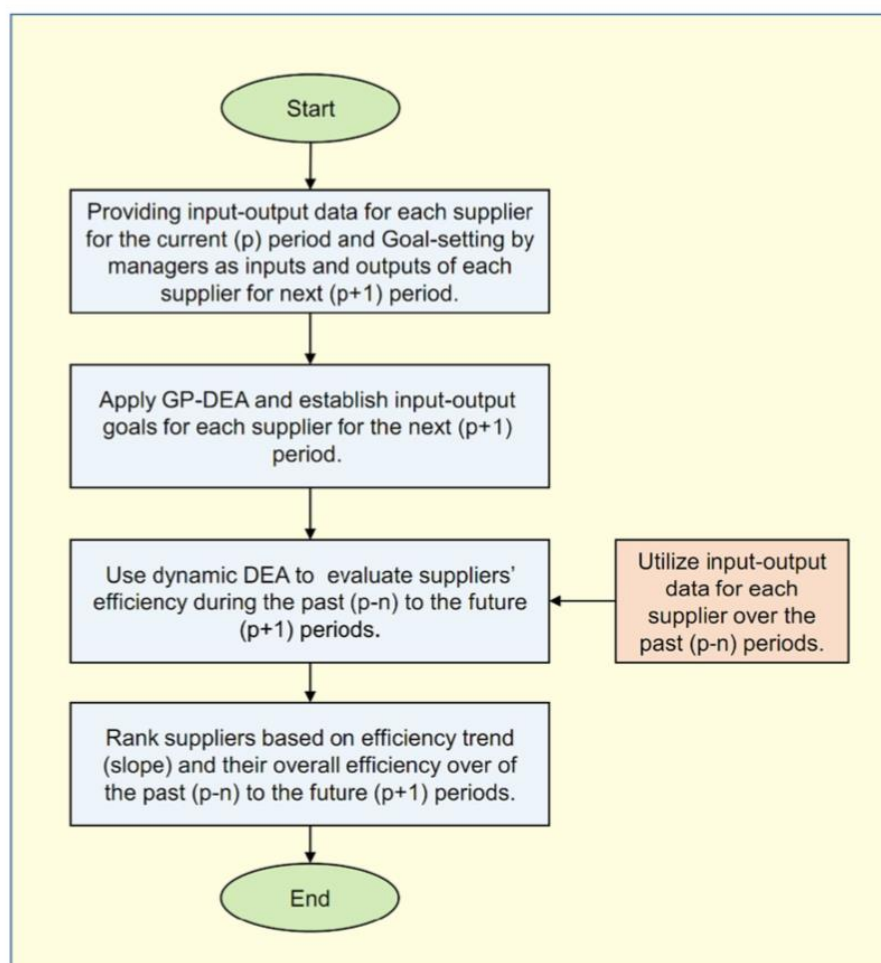


图 7 分析未来决策的 DDEA 算法

提出的算法描述为，首先确定当前 ( $p$  期) 的输入输出数据和买方对下一期 ( $p+1$  期) 的输入输出目标，接着用目标规划 (goal programming) 确定出下一期 ( $p+1$  期) 的输入输出量，然后用之前提到的动态 DEA 的算法根据  $p-n$  期到  $p+1$  期总的效率评价，最后根据总的效率评价对供应商排序，并选择在  $p+1$  期时的供应商。

综上所述，可以看出传统 DEA 方法，本文用到 DEA 方法与改进的 DEA 方法的变化与区别，体现了 DEA 的深刻之处。

### 3.3 Nash 均衡

再对论文中第二阶段的 Nash 均衡进行详细分析。

在第二阶段，对买家来说，其希望对不同的供应商分别确定合适的购买价格，且总支出不能超过其年度预算。对卖家来说，关键的是当下成本、主观效用、竞争环境、其他供应商的价格、缔约倾向以及销售额。在议价过程中，买卖双方既有不同的目标——自身效用的最大化，又有相同的目标——达成交易。而使用纳什议价模型的目的是最大化参与者的效用，以达到均衡点，并优化相关决策。

原论文使用的是在 Nash 讨价还价模型上发展而来的不对称纳什协商模型，它由海萨尼 (Harsanyi) 和塞尔腾 (Selten) 在 Nash 的工作基础上给出，关于不对称 Nash 协商解的基本形式为：

$$(u_1(x^*), \dots, u_n(x^*)) = \text{Argmax}(u_1(x) - d_1)^{a_1} \dots (u_n(x) - d_n)^{a_n} \\ \begin{cases} (u_1(x), \dots, u_n(x)) \geq (d_1, \dots, d_n) \\ (u_1(x), \dots, u_n(x)) \in S \end{cases}$$

其中 $u_1(x)$ ,  $u_2(x)$ , ...,  $u_n(x)$ 分别为决策者 1, ..., n 的效用函数,  $d_1, d_2, \dots, d_n$ 为协商失败的效用,  $S$  为协商域,  $a_i$ 则表示决策者 $i$ 的协商力量 (又称重要程度), 并且 $a_i$ 应满足如下规范性的限制条件:

$$\sum_{i=1}^n w_i = C$$

这保证了所有决策者的协商力量的和是一个常数。

本文将对原论文所建模型的核心部分进行阐述, 分析其是如何应用 Nash 不对称协商模型来描述议价过程, 并得出最终议价结果即不对称 Nash 协商解的。

目标函数(53):

$$\text{Max} Z_3 = (B - U_B) \prod_{k=1}^N (U_k - G_k) \quad (53)$$

目标函数(53)试图最大化买卖双方的效用。其中 $B$ 是买方的可用预算;  $U_B$ 是交易达成后买家的支出;  $U_k$ 是第 $k$ 个供应商的收益, 它由第一阶段所确定的供应量以及第二阶段所商定的价格共同决定,  $G_k$ 是第 $k$ 个供应商的最小期望收益。

限制条件:

$$U_B = D \sum_{k=1}^N [c_k q_k^* + A_k b_k^* + \frac{1}{2} (\frac{h_b}{D}) q_k^* q_k^*] \quad (27)$$

$$U_k = D \left( c_k q_k^* - \left[ z_k q_k^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{P_k} \right) q_k^* q_k^* \right] \right), k = 1, \dots, N \quad (28)$$

$$U_B \leq B \quad (29)$$

$$U_k \geq G_k, k = 1, \dots, N \quad (30)$$

$$U_B \geq \sum_{k=1}^N U_k \quad (31)$$

$$U_B \geq 0 \quad (32)$$

$$U_k > 0, k = 1, \dots, N \quad (33)$$

$$c_k > 0, k = 1 \dots N \quad (34)$$

上述限制条件主要依据议价过程中的实际情况, 刻画了买方预算上限、卖方预期收益下限、买卖双方支出关系的限制。

由于目标函数(53)还没有反映在议价过程中不同议价者的议价能力对效用的影响, 做出改进是自然的。在议价过程中, 起决定影响的是各个参与者的议价能力 (bargaining power), 议价能力由包括市场份额、工作经验、合作历史、经济地位、生产能力等在内的因素决定, 而一个基本的假设是, 拥有高议价能力 (bargaining power) 的参与者能够在议价过程中获得更多的收益, 事实上不难看到, 在原论文中议价能力即是不对称 Nash 协商模型中的协商力量 (重要程度)。在将议价能力 $w_i$ 纳入考量后, 可以对目标函数(53)进行调整, 并得到目标函

数(61):

$$MaxZ_3 = Max(B - U_B)^{w_b} \prod_{k=1}^N (U_k - G_k)^{w_k} \quad (61)$$

可见目标函数(61)就是前述 Nash 不对称协商模型中的目标函数，可以发现，决策者对应的议价能力越强，得出的不对称 Nash 协商解就会往相应决策者偏移，当然，对于  $w_i$  同样应当有限制条件(62):

$$w_b + \sum_{k=1}^N w_k = C \quad (62)$$

这便是对议价能力的规范性限制，它保证了所有参与者的总议价能力等于一个常数  $C$ 。

可以验证，如上建立的模型满足 Nash 讨价还价模型的基本假设，故均衡点必然存在且唯一，并处于第一象限，均衡点对应的不对称 Nash 协商解就是具有不同决策能力的各方协商后所确定的最终价格。

## 4 数值实验

原论文的第四部分引入了一个数值实验来说明模型的合理性，并使用了 lingo 求解器进行求解。

本文在处理器为 Intel(R) Core(TM) i9-10900 的 Windows10 操作系统的计算机上使用 lindoglobal 求解器（版本号：9.0.1983.157）和 MATLAB（版本号：9.10.0.1649659 (R2021a)）对原论文第四部分数值实验进行了复现，主要有如下结果：1、验证原论文 17 页的模型增加了一个关键约束后能使得模型得到正确求解（对应本文 3.1.1.2 中提到的情况）；2、在 1 的基础上发现了原论文 25 页模型求解结果存在数值错误，并发现复现结果优于原论文结果；3、在对相关数值进行修正后，得到了第一阶段模型的正确结果；4、根据第一阶段模型的结果估计了买家预算的下确界（原论文为直接提出预算），并对原论文 29 页中提到“供应商能接受的最低商品单价”的方法进行了修正，然后根据新的预算方案求解了第二阶段模型（对应本文中）；5、比较第二阶段理论解和数值解；6、使用遗传算法和分式规划求解第一阶段的模型的结果。

下述将在第一节阐述数据来源与处理情况，在第二节展示用 lindoglobal 求解器求解第一阶段模型的结果以及与原论文结果的差异（对应上文 1、2、3、4），在第三节阐述第二阶段数值实验中发现的问题以及解决方案（对应上文的 4 和 5），在第四节介绍遗传算法结合分式规划的复现情况（对应上文的 6）

### 4.1 数据来源、处理与展示

原论文数值实验假设有 10 个候选供应商，并且基于买方年度需求（200,000），买方至多选择 6 个供应商。买方和卖方的相关数据来源于 Kamali et al. 在 2011 年的论文，具体数据如下表所示：

参数	具体数值
----	------

$h_b$	2.6
$N$	6
$D$	200,000
$B$	1,750,000

表 2 买方相关数据

参数	卖方 1	卖方 2	卖方 3	卖方 4	卖方 5	卖方 6	卖方 7	卖方 8	卖方 9	卖方 10
$A_k$	40	19	25	39	27	33	30	23	20	34
$S_k$	43	39	42	30	38	42	40	39	38	32
$c_k$	9.0	9.1	8.7	10.5	9.5	8.9	8.7	10.4	9.0	10.5
$h_k$	2.29	1.96	2.74	0.54	1.50	1.25	2.00	2.09	1.90	1.71
$z_k$	4.04	6.48	7.17	5.87	6.30	4.85	5.08	7.00	6.00	5.25
$P_k$	42,000	34,000	36,500	63,000	45,500	64,000	41,500	36,000	66,500	61,000

表 3 卖方相关数据

特别地，在 DEA 模型中，首先考虑三个参数：潜在供应商数量（ $n$ ）、输入数量（ $m$ ）和输出数量（ $s$ ）。参考 Friedman and Sinuany-Stern(1998)的研究，三个参数需要满足条件 $n \geq 3(m + s)$ 。基于此，本文使用一个输入和两个输出来衡量供应商的效率。其次，本文使用运输的总成本（TC）作为输入，使用服务质量体验（EXP）和服务质量可信度（CRE）作为输出，数据来源于 Talluri and Baker (2002),具体数据如下表所示：

参数	卖方 1	卖方 2	卖方 3	卖方 4	卖方 5	卖方 6	卖方 7	卖方 8	卖方 9	卖方 10
$I_k$	253	268	259	180	257	248	330	327	330	321
$O_{1k}$	240	210	270	200	160	230	170	180	170	200
$O_{2k}$	90	80	70	70	70	80	60	70	60	80

表 4 DEA 相关数据

## 4.2 阶段一的数值实验结果

承上文所述，在阶段一中需要从两个角度考虑供应商的选择问题——供应链的总开销和供应商的效率。为了求解这个多目标模型，需要先求解两个单一目标的最值问题。

### 4.2.1 最优化供应链总开销的数值实验结果

原论文中对于供应链总开销的数值实验结果呈现在表五和表六，其中表五中关于 $Z_1$ 的最优值计算出现了数值错误，经验证（即将求解结果中的变量代入目标函数）应从 2,800,966 修正为 2,803,719.826。其余结果比较如下表所示。

说明	原论文结果	数值实验复现结果
供应链总开销（需尽量小）	2,803,719.826	2,803,484.831



供应商选择	1-6-7-9	1-6-7-9
$(q_1, q_1')$	(1727.96, 0.2100)	(1435.189, 0.2100)
$(q_6, q_6')$	(2643.08, 0.3200)	(2186.955, 0.3200)
$(q_7, q_7')$	(1707.32, 0.2075)	(1418.104, 0.2075)
$(q_9, q_9')$	(2160.02, 0.2625)	(1793.986, 0.2625)

表 5 最优化供应链总开销的数值实验结果与原论文结果的比较

从此结果中可以看出，数值实验复现中得到的目标函数值要优于原论文中的实验结果。但是复现实验中对供应商的选择和每个周期供应商的供应比例与论文中的计算结果吻合。复现结果表明，比起论文中的分配方案，买方减少每个周期的购买量而增加购买周期的个数有能使得整个供应链的总开销更小。

#### 4.2.2 最优化供应商总效率的数值实验结果

原论文中对于供应链总开销的数值实验结果呈现在表五和表六，由于本部分实验结果较为接近，因此未对原论文结果进行验算（且原论文中未呈现 $U_{kj}$ 和 $V_{ki}$ 的值）。比较结果如下表所示：

说明	原论文结果	数值实验复现结果
供应链总效率（需尽量大）	9.683425	9.645495
供应商选择	1-3-4-6	1-3-4-6
$(q_1, q_1')$	(29.93, 0.2100)	(*, 0.1825)
$(q_3, q_3')$	(26.012, 0.1825)	(*, 0.1825)
$(q_4, q_4')$	(44.90, 0.3150)	(*, 0.3150)
$(q_6, q_6')$	(41.69, 0.2925)	(*, 0.3200)

表 6 最优化供应商总效率的数值实验结果与原论文结果比较

此处未展示数值实验复现结果的 $q_i$ 是因为 $q_i$ 与 $q_i^*$ 仅差一个倍数关系 $t$ （见上文），而在复现实验中计算得到 $t = 1000$ 较为反常，但由于原论文没有给出完整的实验结果，因此难以检查其中的逻辑问题。除此之外，复现实验得到的目标函数值不如原论文结果，但其对供应商的选择和原论文一致。下表呈现的是数值实验中计算得到的其他变量的值，以供参考。

变量	$k_1$	$k_3$	$k_4$	$k_6$
$U_{kj_1}$	0.0010	0.0031	0.0010	0.0033
$U_{kj_2}$	0.0073	0.0010	0.0114	0.0010
$V_k$	0.00395	0.00386	0.00556	0.00403
$e_k$	0.10240	0.08628	0	0.16582

表 7 数值实验复现中计算得到的其他变量的值

#### 4.2.3 双目标综合优化结果

原论文中关于权重的选择是主观地选择了五组不同的权重，由于计算机硬件和时间限制上的限制，本文仅复现了权重为(0.5,0.5)的一组，因为这组最具代表性，且在数值实验第二

阶段中使用的数据也来自于这一组的求解结果。由于目标函数中的参数  $Z_1^*$  和  $Z_2^*$  发生了变化且原论文中变量求解结果缺失，因此无法就最优函数值进行比较，下表是与供应链选择相关的一些变量求解结果对比。

说明	原论文结果	数值实验复现结果
供应商选择	1-3-4-6	1-3-4-6
$(q_1, q_1')$	(1845.54, 0.2100)	(2554.578, 0.2100)
$(q_3, q_3')$	(1603.86, 0.1825)	(2109.27, 0.1683)
$(q_4, q_4')$	(2526.63, 0.2875)	(3860.53, 0.3017)
$(q_6, q_6')$	(2812.25, 0.3200)	(3933.79, 0.3200)

表 8 双目标综合优化结果与原论文结果比较

数值实验复现结果表明，原论文中对供应商的选择正确，在被选择供应商处的购买额度分配也是大致正确。值得注意的是，复现结果更倾向于在每个购买周期增加购买量来减少购买周期的数量

### 4.3 阶段二的数值实验结果

第二阶段是买家与已选定的供应商议价博弈的过程。其中的关键参数是买家的预算  $B$ ，买家的预算不同将导致最后的博弈结果有很大不同。由于阶段一的得到的结果与论文中的所述的结果有较大不同，因此本部分按照原论文逻辑对参数选择和根据阶段一复现结果计算得到的阶段二结果进行阐述。

#### 4.3.1 阶段二额外需要的参数选择说明

首先总结阶段二所需要的阶段一的结果中的数据如下：

参数	$k_1$	$k_3$	$k_4$	$k_6$
新标号	1	2	3	4
$q_k^*$	2554.578	2109.277	3860.533	3933.719
$q_k'^*$	0.2100	0.1683	0.3017	0.3200
$c_k$	9.0	8.7	10.5	8.9

表 9 阶段二所需要的阶段一的结果中的数据

此外需要指定这些供应商最低能接受的利润  $G_k$ 。其计算公式如下：

$$G_k = SAP_k - X_k = D \left( c_k q_k^* - \left( z_k q_k'^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{P_k} \right) q_k'^* q_k^* \right) \right) - X_k \quad (63)$$

其中根据原论文数据， $X_k$  由下表决定。

参数	$k_1$	$k_2$	$k_3$	$k_4$
$X_k$	41,045.7	5269.2	45,036.3	38,473.0
$SAP_k$	204,704.7	48,168.2	277,870.1	256067.2

$G_k$	163659.0	42899.1	232833.8	217594.2
-------	----------	---------	----------	----------

表 10  $X_k$  以及一些其他参数的计算结果

最后来讨论预算  $B$  的选择。实验表明，原论文中设定的预算 1,750,000 在阶段一复现结果的供应商分配方案下是不够用的，即在设定  $B = 1,750,000$  的情况下，满足第二阶段约束的模型无解（这并不令人意外，因为阶段一的计算中除了考虑全供应链的开销，还考虑了供应商的效率问题）。计算得知，倘若不进行议价博弈阶段，买方需要付出的成本为 1,887,705.3。原论文中也提到倘若不进行议价博弈阶段，按照原论文的数据买方需要支付的成本为 1,875,016，但原论文没有解释最终选取 1,750,000 作为预算的合理性，仅提到降低预算参与议价博弈可以使买方以更低的价格购得相同数量的物品。

为了探究  $B$  选取的合理性，在数值实验中我们首先计算了在满足约束要求下的最小预算的值（可以直接将  $B$  作为目标函数求解模型，也可以根据模型原理，先计算每个供应商能接受的最低商品单价，然后再计算  $B$ ，为了方便起见这里使用了前者），其值为 1,755,424.8。利用这个预算下可以得到每个供应商能接受的最低商品单价如下表：

参数	$k_1$	$k_2$	$k_3$	$k_4$
$c_{kmin}$	8.0227	8.5435	9.7536	8.2989

表 11 每个供应商能接受的最低商品单价

值得注意的是，原论文在 29 页提到的最低商品单价计算公式  $\frac{c_k \times G_k}{SAP_k}$  不严谨，这是因为  $SAP_k = D \left( c_k q_k^* - \left( z_k q_k^* + S_k b_k^* + \frac{1}{2} \left( \frac{h_k}{p_k} \right) q_k^* q_k^* \right) \right)$  对  $c_k$  没有齐次性，因此按照论文中的方法计算出的最小单价偏小。

### 4.3.2 阶段二的数值实验结果

理论上，任何位于 1,755,424.8 和 1,887,705.3 之间的值都可以作为一个合理的买家预算进行求解。这里需要强调的是 1,887,705.3 为不进行议价博弈阶段时买方需要付出的成本，其对应的单位商品价格并非在相同预算下进行议价博弈后的单位商品价格。为了模拟论文中的数值实验，首先选取  $B = 1,760,000$  作为买家预算进行了模型求解。经过计算，可以得到供应商新的单位商品价格：

变量	$k_1$	$k_2$	$k_3$	$k_4$
$c_k$	8.0445	8.5707	9.7687	8.3131
$c_{kmin}$	8.0227	8.5435	9.7536	8.2989

表 12  $B = 1,760,000$  时供应商新的单位商品价格

当  $B$  为其他数值时  $c_k$  的表现大致如下图所示（由于计算机硬件和时间限制上的限制，以 10,000 为间隔，从  $B$  等于 1,760,000 到 1,880,000 求次优解后绘图）

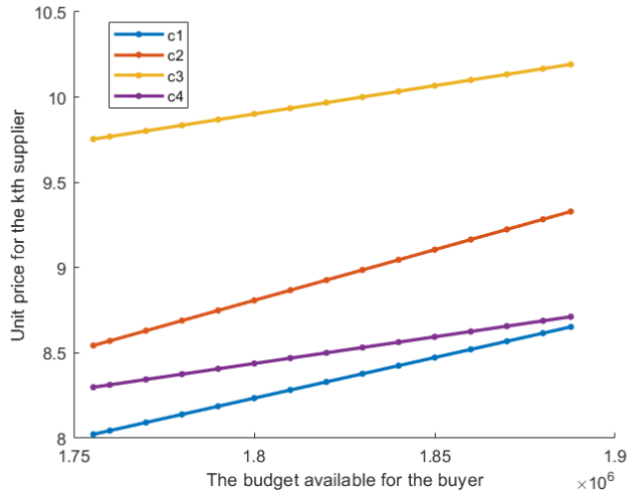


图 8  $B$  为其他数值时  $c_k$  的变动情况

有两点情况需要说明。首先观察  $c_2$ ， $c_2$  在议价博弈前的值为 8.7，而当买家预算较大时，其值超过了 8.7，这充分说明了供应商 3（原标号）的议价能力强。这主要取决于图像的“截距”和斜率，如供应商 1，尽管图像斜率与供应商 3 相似，但由于“截距”太小因此在议价博弈中无法收获比初始单位商品价格更高的单价。其次，从图像可见  $c_k$  似乎与  $B$  存在线性关系。从理论解中知道，这并非偶然（参见本文 3.1.2 中理论解显然与  $B$  有线性关系）。

#### 4.3.3 理论解和数值解的比较

进一步验证理论解的正确性的一个方法是比较理论解在特定参数下的数值表现。这里使用与数值复现实验中相同的预算  $B = 1,760,000$ ，计算结果如下表所示：

变量	$k_1$	$k_2$	$k_3$	$k_4$
$c_k$ （理论解）	8.0671	8.5988	9.7844	8.3280
$c_k$ （数值优化解）	8.0445	8.5707	9.7687	8.3131

表 11  $B = 1,760,000$  时理论解与数值优化解的比较

观察数据可以发现，理论解与数值优化解有很小的差距，考虑到数值优化使用的算法大多数为迭代算法，因此这种数值差异是可以接受的，也说明了理论解的正确性。

### 4.4 遗传算法结合分式规划的复现情况

由于阶段二已经有理论解法，因此不再讨论阶段二的数值优化方法。而本问题真正求解的难点在于阶段一，尤其在于阶段一中最优化供应链总开销的求解。求解方法的理论部分如本文 3.1 中所述，本部分主要展示数值情况：首先展示的分配问题（即已经选定了供应商 1、6、7、9，求解  $q_k$ ）的情况，随后展示结合遗传算法后对供应商的选择情况。

#### 4.4.1 两种不同的变量替换法求解结果

本节展示的结果仍使用 `lindoglobal` 求解器计算，仅作为证明两种方法的结果一样的依据。由于 `lindoglobal` 求解器不仅完成了求解工作，还完成了证明解最优的工作（原理未知）因此比较两种方法的求解器求解时间从实际上考虑是意义不大的，故不展示。

说明	新变量替换法的结果	旧变量替换法的结果
$q_1$	1,463.405	1,461.586
$q_6$	2,229.951	2,227.178
$q_7$	1,445.984	1,444.186
$q_9$	1,829.257	1,826.982

表 12 两种变量替换法求解器结果比较

可以看出两种方法的求解结果十分相近，因此认为他们是等效的是合理的。下表展示了新变量替换法的变量值，他们将作为参照与下文我们自行撰写的代码结果进行对比。

说明	变量值
$q'_1$	4.68228422
$q'_6$	7.13490929
$q'_7$	4.62654274
$q'_9$	5.85285528

表 13 新变量替换法的原求解结果

#### 4.4.2 将 $t$ 用 $q_k$ 表示的必要性

本节展示的结果是，如果不进行 3.1.1.4 中用 $q_k$ 表示 $t$ 的步骤，而直接将约束(45)作为惩罚项时，惩罚项（不带惩罚因子）的梯度情况。其中使用的参数是

说明	变量值
起始点 $(q'_1, q'_6, q'_7, q'_9, t)$	(0.21000 0.32000 0.20750 0.33250 0.03848)
惩罚系数 $\sigma_1$	1
梯度下降步长 $a$	0.000000002

表 14 梯度下降法使用的参数

使用这么小的步长和惩罚系数是无奈之举，否则很容易不收敛。下表是梯度下降前几次的情况：

说明	变量值
起点 $(q'_1, q'_6, q'_7, q'_9, t)$	(0.21000, 0.32000, 0.20750, 0.33250, 0.03848)
目标函数梯度	(-0.03848, -0.03848, -0.03848, -0.03848, -1.0700)
非线性等式约束梯度	(0.00000, 0.00000, 0.00000, 0.00000, 0.00000)

线性等式约束梯度	(0.00000,0.00000,0.00000,0.00000,0.00000)
----------	---

---

表 15 梯度下降法第一次迭代	
说明	变量值
起点( $q'_1, q'_6, q'_7, q'_9, t$ )	(0.21004,0.32004,0.20754,0.33254,0.03955)
目标函数梯度	(-0.03955, -0.03955, -0.03955, -0.03955, -1.07015)
非线性等式约束梯度	(0.04125,0.04350,0.04359,0.04745,3.00830)
线性等式约束梯度	(0.00000,0.00000,0.00000,0.00000,0.00000)

---

表 16 梯度下降法第二次迭代	
说明	变量值
起点( $q'_1, q'_6, q'_7, q'_9, t$ )	(0.21004,0.32003,0.20753,0.33253,0.03761)
目标函数梯度	(-0.03761, -0.03761, -0.03761, -0.03761, -1.07014)
非线性等式约束梯度	(-0.03122, -0.03292, -0.03299, -0.03591, -2.32353)
线性等式约束梯度	(0.00000,0.00000,0.00000,0.00000,0.00000)

---

表 17 梯度下降法第三次迭代	
说明	变量值
起点( $q'_1, q'_6, q'_7, q'_9, t$ )	(0.21011,0.32011,0.20761,0.33260,0.04100)
目标函数梯度	(-0.04100, -0.04100, -0.04100, -0.04100, -1.07042)
非线性等式约束梯度	(0.10212,0.10767,0.10791,0.11746,7.34225)
线性等式约束梯度	(0.00000,0.00000,0.00000,0.00000,0.00000)

---

表 18 梯度下降法第四次迭代	
说明	变量值
起点( $q'_1, q'_6, q'_7, q'_9, t$ )	(0.21004,0.32004,0.20754,0.33253,0.03473)
目标函数梯度	(-0.03473, -0.03473, -0.03473, -0.03473, -1.07015)
非线性等式约束梯度	(-0.12183, -0.12847, -0.12875, -0.14015, -9.37588)
线性等式约束梯度	(0.00000,0.00000,0.00000,0.00000,0.00000)

---

表 19 梯度下降法第五次迭代	
-----------------	--

从前梯度下降的前 5 步可以看出，非线性等式约束(45)带来的梯度越来越大，这将导致梯度下降法难以完成原本的目标——求出目标函数的最小值，而转变为完成找到满足约束(45)的解这一目标，因此进行 3.1.1.4 中用 $q_k$ 表示 $t$ 的步骤是必要的。

#### 4.4.3 订单分配问题结果（分式规划、罚函数和梯度下降法）

针对订单分配问题的参数选择，有两点需要注意：第一是此问题的求解速度必须较快，

因为它将作为遗传算法中的适应度函数的值，如果求解缓慢将导致遗传算法完全不可行；第二是它的参数必须有一定的鲁棒性，因为在遗传算法中需要解决的是很多个订单分配问题，而每个订单分配问题的数据都不一样。

基于这两个特点，本文在原罚函数算法的基础上对其进行了一些改进，使之适配即将应用的遗传算法，并且对其进行了测试，兹展示如下：

罚函数法算法（适配遗传算法）	
步骤一：	取 $\sigma_1 = 0.01$ ，取允许误差 $\varepsilon = 10^{-9}$ ，取初始梯度下降步长 $a_1 = 100$ ，令 $k := 1$
步骤二：	生成梯度下降起始点 $(\frac{p_k}{D})_{k \in J}$
步骤二：	以步长 $a_k$ 用梯度下降法求解无约束极值问题，迭代 $gra = 10000$ 次
$\min f = -t \sum_{k \in J} q'_k + \sigma_k \max\{C_i(x), 0\}^2$	
的最优解 $x^k = x(\sigma_k)$ 。	
步骤三：	检查约束条件是否满足：是否存在 $i$ 使得
$C_i(x^k) > \varepsilon$	
若是，则取 $\sigma_{k+1} = \rho \sigma_k, a_{k+1} = \rho^{-1} a_k$ ，其中 $\rho = 2$ ，转到步骤三；	
若否，检查迭代次数是否达到最小值 $pu_{min} = 25$ 。若是则迭代停止，得到最优解 $x^k$ ；若否，转到步骤三。	

现在对算法内容以及参数选择做出解释。使用确定方法生成初始点而非随机生成初始点的目的是令初始点满足约束条件，以降低初始时约束项梯度带来的影响；初始惩罚系数较小的目的是降低约束在算法刚开始时对整个梯度的影响；固定梯度下降迭代次数的理由是，在开始时梯度下降步长比较大无需迭代过多次数，而在罚函数算法后期，当前解已经逼近最优解，也无需迭代过多次数，10000次是多次测试后的结果；设置罚函数算法最小迭代次数的理由是，算法开始梯度下降的步长很大，且限制了迭代次数，不一定能收敛到最优解，但解很可能是满足约束的，因此需要设定一个最小迭代次数使得最终输出的结果是在梯度下降法步长较小时输出的。

改进后的算法对本问题的适配性非常好，达到了作为遗传算法适应度函数的要求，此处仅举两例。

说明	变量值
起始点 $(q'_1, q'_6, q'_7, q'_9)$	(0.21000,0.32000,0.20750,0.33250)
罚函数算法结果	(4.67660,7.12625,4.62093,5.84575)
lindoglobal 求解器结果	(4.68228,7.13491,4.62654,5.85286)
罚函数算法目标函数值	0.07133973
lindoglobal 求解器目标函数值	0.07133974
罚函数算法约束损失 $\  \cdot \ _\infty$	0.0000000008
耗时	3.299811 秒

表 20 选择供应商 1，6，7，9 时罚函数算法求解情况

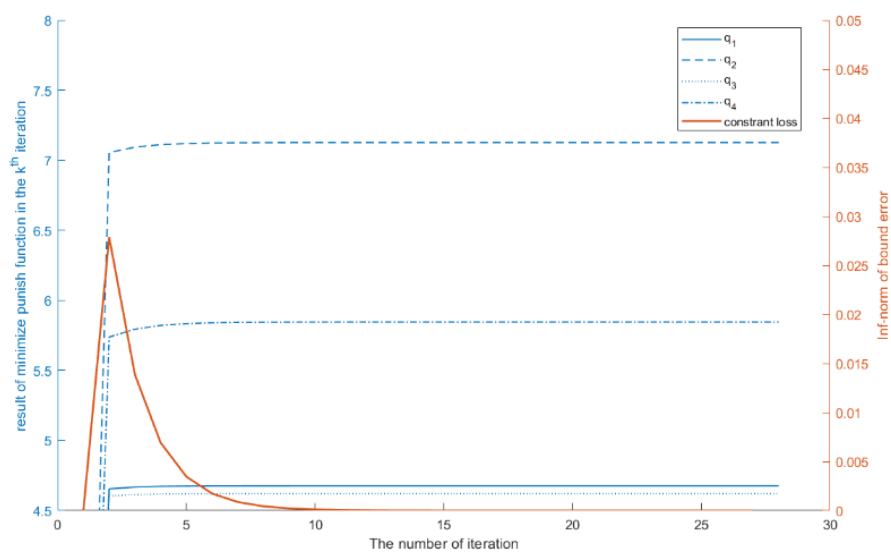


图 9 选择供应商 1, 6, 7, 9 时罚函数算法迭代过程情况

说明	变量值
起始点 $(q'_1, q'_3, q'_4, q'_9, q'_{10})$	$(0.21000, 0.32000, 0.20750, 0.33250)$
罚函数算法结果	$(4.57365, 3.32134, -0.00000, 7.24162, 6.64269)$
lindoglobal 求解器结果	$(4.57367, 3.32135, 0.00000, 7.24164, 6.64271)$
罚函数算法目标函数值	0.06646355
lindoglobal 求解器目标函数值	0.06646355
罚函数算法约束损失 $\ \cdot\ _\infty$	0.0000000005
耗时	4.044652 秒

表 21 选择供应商 1, 3, 4, 9, 10 时罚函数算法求解情况



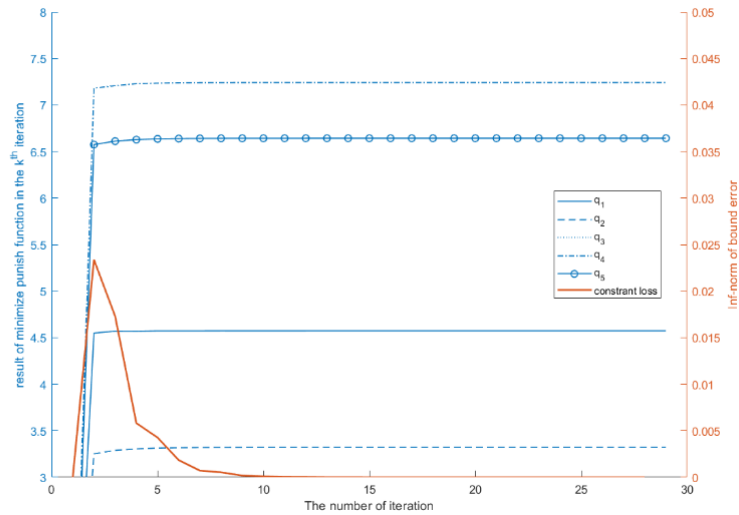


图 10 选择供应商 1, 3, 4, 9, 10 时罚函数算法迭代过程情况

从这两个例子可以看出，针对本问题的数据设计的罚函数算法是非常有效的：求解结果精准、运行速度快（与求解器求次优解的速度几乎一致）。因此，将罚函数算法的结果作为遗传算法的适应度函数，是合理的。

#### 4.4.4 供应商选择问题结果（遗传算法，以 4.4.3 中的最优化结果为适应度函数）

最后叙述使用遗传算法求解第一阶段中供应链总成本优化问题的结果，为了使结果可复现，我们在撰写 MATLAB 代码时使用了“rng default”的语句，实际应用中可以随机选择初始点多次重复运行。

以供应商选择变量构成的序列 $(y_1, y_2, \dots, y_{10})$ 为遗传算法中的染色体，例如求解器的最优结果可以表示为 $(1, 0, 0, 0, 0, 1, 1, 0, 1, 0)$ ，说明求解器认为选择第一个、第六个、第七个和第九个供应商是最优的。

对于每一组供应商（染色体）选择，可以使用 4.4.3 中的算法求解订单分配问题的最优值，将它作为染色体的适应度函数值。注意到，在每一代计算中，计算染色体池中的每一条染色体适应度是互不相干的，因此可以使用并行计算极大地加快计算速度。

遗传算法的其他参数选择如下表所示：

参数	具体数值
$n$ （种群规模、染色体池大小）	60
$pc$ （染色体发生交叉的概率）	0.6
$pm$ （染色体发生变异的概率）	0.2
$t$ （进化代数）	5

表 22 遗传算法使用的参数

最终遗传算法在第二代种群中找到了最优解 $(1, 0, 0, 0, 0, 1, 1, 0, 1, 0)$ ，而运行整个遗传算法的耗时为 160.092627 秒（运行环境已在本部分开头中叙述）。遗传算法的完整求解结果将在附录三中展示。

至此，使用求解器和自行设计算法对原论文数值实验进行复现的全部工作已经完成，并且可以认为自行设计的算法有效。

5 供应商选择与订单分配问题综述

5.1 综述总览

供应商选择通常需要通过定性与定量方法，从多维度的因素对供应商进行评价与选择；而在订单分配时，则需要通过数学规划方法对选择的供应商订单进行分配，以达到最优目标值。

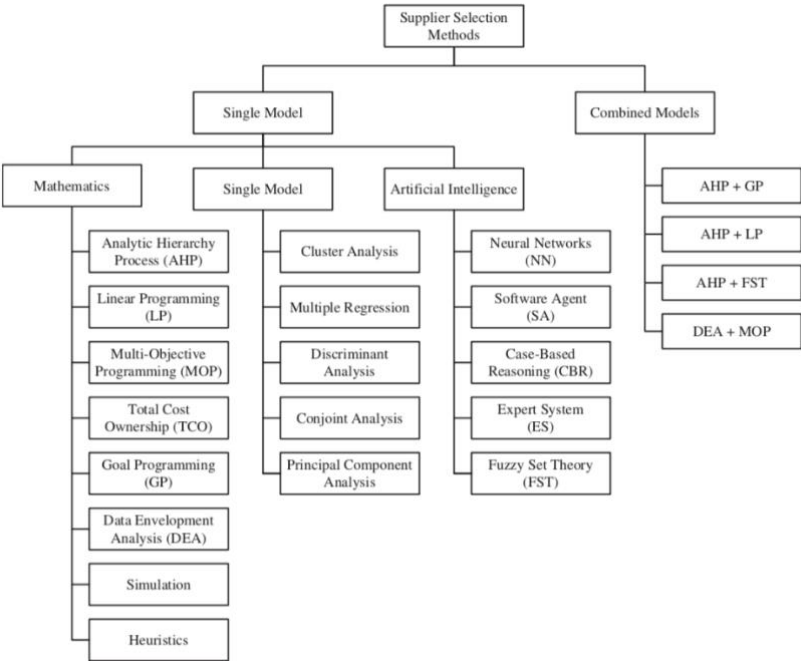


图 11 供应商选择和订单分配问题方法总结

供应商选择和订单分配问题可以总结如图所示，该类问题最常见的是通过层级分析法、线性规划、多目标规划、目标规划和数据包络分析等单个数学模型进行研究，在许多研究中为了结合各个模型的优点，将多个模型融合起来，比如 Liao (2010)[4]将层次分析法和目标规划结合起来对供应商选择问题进行分析，并且由于商业信息的不确定性，模糊集理论也常应用于研究供应商选择问题。

本文以层次分析法（AHP）和目标规划方法在该类问题的应用为例，用以阐述其他方法在供应商选择和订单分配问题中的应用，在阐述过程中我们对相应论文中的案例进行调整简化，使得读者便于理解各类方法的主要数学思想。

作者	方法	描述
Xinyang Deng, Yong Hu 等（2014）[5]	层次分析法，模糊集理论	基于已有的 AHP 方法，结合不确定信息论，提出 D-AHP 方法
Chan, Kumar（2007）[6]	层次分析法	应用层次分析法考虑组织在全球范围选取供应商
Xiaoge Zhang, Felix T. S. Chan 等（2015）[7]	网络层次分析法，模糊集理论	结合网络层次分析法（ANP）和模糊集理论，提出供应商选择方法
Bohner and Minner	线性混合整数规划	在考虑订单数量折扣的情况下，通过混合整数

(2017) [8]		线性规划找到最优解
Mendoza and Ventura	非线性混合整数规划	在考虑了购买、库存和运输成本下,用混合整数非线性规划求解
(2013) [9]		
Choudhary and Shankar	目标规划3,多目标规划	对库存运输和供应商选择问题,提出目标规划用于决策分析订单
(2014) [10]		
Rezaei 等	(2015) [11] 多目标规划,演化算法	通过多目标规划解决供应链管理问题,用启发式算法求解
Jadidi 等	(2015) [12] 目标规划,加权目标规划,模糊集理论	对传统的目标规划分步改进,得到用于解决模糊决策的目标规划
Azadi 等	(2015) [13] DEA,模糊集理论	在模糊的背景下采用 DEA 对供应商进行选择
Yousefi 等	(2021) [1] DEA,非线性混合整数规划,博弈论	将 DEA 方法与非线性混合规划结合起来评价供应商,并利用博弈论对议价问题求得纳什均衡。
Rezaei 等	(2017) [14] 多目标规划, DEA	在考虑数量折扣的情况下,利用多目标规划和 DEA 对供应商进行筛选。
Moheb-Alizadeh 和 Hand- field	(2017) 多目标非线性整数规划,随机分析	考虑了采购商需求是随机情况下,供应商选择与订单分配问题的研究
[15]		
Hosseininasab and Ahamdi	(2015) [16] 多目标的组合优化	将供应商视为一个组合,利用组合优化理论进行研究

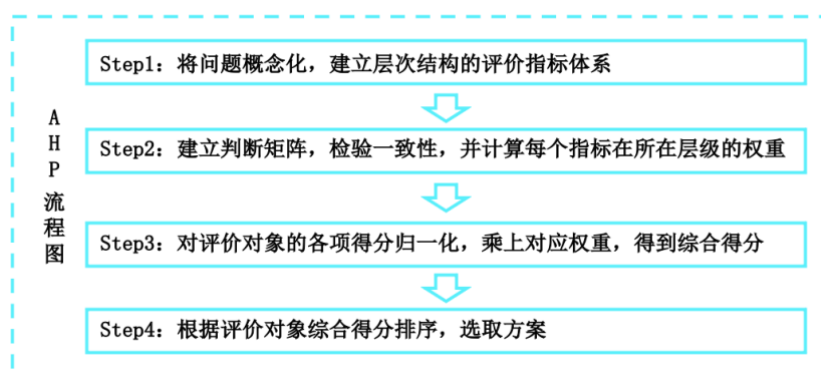
表 23 文献综述表

## 5.2 层次分析法的应用

在 5.2 本文将结合例子分析层次分析法(AHP)在供应商选择和订单分配问题中的应用。

### 5.2.1 基本的层次分析法

层次分析法 (Analytic Hierarchy Process, AHP) 由运筹学家 Saaty (1971) 提出, 主要应用于解决多个评估准则的决策问题。层次分析法可以将复杂的评价问题系统化, 进行层级分解, 通过量化运算找到综合评估的方法。目前该法已经广泛应用于资源分配、冲突分析、供应商选择和方案评价等。



3 目标规划 (goal programming) 和多目标规划 (multi-objective programming) 是两个不同的概念, 翻译成中文容易造成混淆, 值得注意。

图 12 层次分析法流程图

上图描述层次分析法的步骤，AHP 的核心任务是在多评价指标的决策当中，确定每项评价指标的权重，即步骤一和步骤二。在决策时，难以直接敲定出每项指标的权重，需要对不同指标进行两两比较重要性，建立判断矩阵，进而计算出指标的权重。

本文对 Chan（2007）[6]利用 AHP 方法解决供应商选择问题中的例子稍做简化调整，来阐述 AHP 方法解决供应商选择问题的过程。

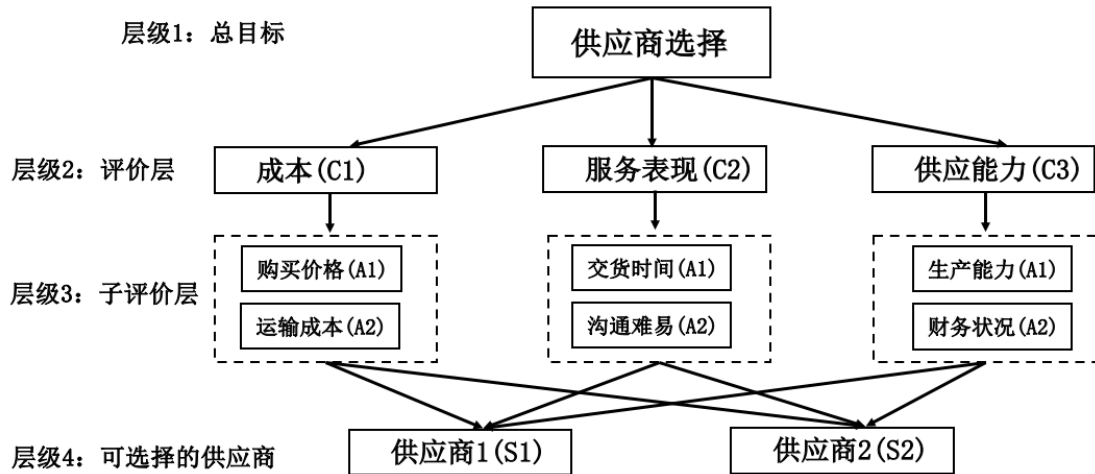


图 13 简化的供应商选择例子

第一步，可以将供应商选择问题概念化，建立层次结构的评价指标体系。供应商的选择可以从成本（C1）、服务表现（C2）和供应能力（C3）的三个维度进行衡量，即二级的评价指标（层级 2：评价层）；对于每个二级评价指标可以细分为子评价，即建立三级评价指标（层级 3：子评价层），比如对于服务表现的衡量，可以细分为从供应商交货时间的长短以及沟通的难易程度；层次 4 代表是可选择的供应商，通过比较供应商的在细分指标上综合得分进行最终选择。

	成本(C1)	服务表现(C2)	供应能力(C3)
成本(C1)	1	5	3
服务表现(C2)	1/5	1	1/3
供应能力(C3)	1/3	3	1

表 24 层次 2 评价指标的判断矩阵（即二级指标的判断矩阵）

C1	购买成本	运输成本	C2	交货时间	沟通难易	C3	生产能力	财务状况
购买成本	1	5	交货时间	1	1	生产能力	1	3
运输成本	1/5	1	沟通难易	1	1	财务状况	1/3	1

表 25 层次 3 子评价指标的判断矩阵（即三级指标的判断矩阵）

第二步，由专家确定两两指标的相对重要性，构建判断矩阵。二级指标和二级指标的判断矩阵表如上表所示，通常而言，1 代表一样重要，3 代表稍微更加重要，5 代表强烈更加重要，比如由表的第二行第五列的元素 5 可以知道，成本指标较于供应商的服务表现是强烈更加重要的。

判断矩阵的最大特征根对应的特征向量即为每项指标所对应的权重。

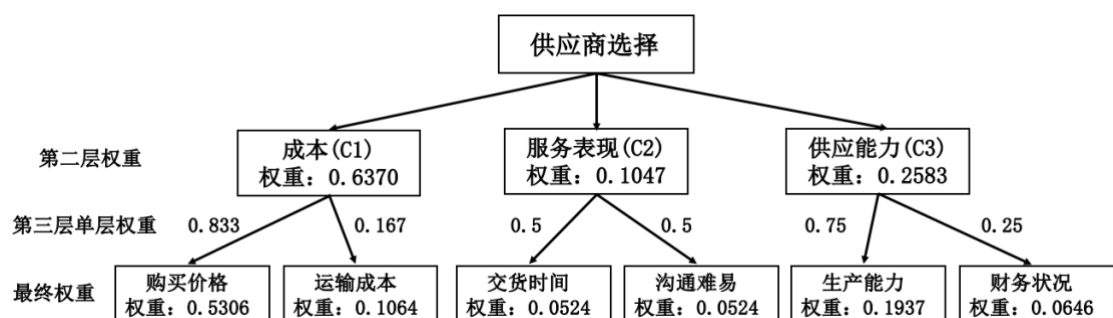


图 14 各级指标的最终权重图

图清晰地展示了各个层级指标的权重关系，可以看出购买价格、运输成本、交货时间、沟通难易、生产能力和财务状况指标的最终权重。

第三步，对可选择两个供应商 S1 和 S2 在各项得分上进行归一化，乘于权重，得到最终综合得分。在进行归一化时，每项指标的最好和最差情况是根据以往历史供应商数据得到的，并不是两个供应商最好和最坏的情况，并且归一化可以分为正向归一化和负向归一化。

		购买价格	运输成本	交货时间	沟通难易	生产能力	财务状况
最好情况		20	4	1	10	20	10
最坏情况		100	14	10	0	0	0
实际	S1	40	9	4	4	15	5
情况	S2	50	7	5	6	13	6
归一	S1	0.75	0.5	0.667	0.4	0.75	0.5
化	S2	0.625	0.7	0.556	0.6	0.65	0.6

表 26 供应商 S1 和 S2 的得分表

第四步，将供应商在每项指标上的总得分乘上对应权重，可以得到供应商 S1 和 S2 综合分数分别为 0.6846 和 0.6313，因此最终方案为选择供应商 S1。

结合上述案例，发现层次分析法将 3 个二级指标和 6 个三级指标最终化成了一个综合分数，可以全面客观地从多个角度对供应商进行合理评价。

### 5.2.2 模糊集理论改进的层次分析法

层次分析法简单易操作，但也存在一定局限性。Deng（2014）[7]和 Chan（2007）[8]指出层次分析法不能够处理带有不确定的信息，层次分析法要求人在打分或赋予指标权重时给出一个精确的值，然而人在打分和判断时具有强烈的主观性，对评价对象的感知往往是模糊而不精确的，为了使方法更加贴近实际情况，所以需要将模糊集理论引入到层次分析法当中。

这里仅简要介绍模糊集处理 AHP 问题的思想，相对于原有的 AHP 的方法，引入模糊集的方法则是通过三元组的方式来描述确切的比重。

$$fuzzy\ number = ( \underbrace{n_1}_{\text{可能出现的最小数}}, \underbrace{n_2}_{\text{最有可能出现的数}}, \underbrace{n_3}_{\text{可能出现的最大数}} )$$

比如专家们在判断成本指标(C1)是服务表现指标(C2)的多少倍重要时，可以进行较为模糊的判断，专家们认为成本指标至少是服务表现指标的 2 倍重要，至多是服务指标 4 倍重要，最应该是服务指标的 3 倍重要，这样专家的判断就不再是一个精确的数，则是转化一个三元组，即  $fuzzy\ number = (2, 3, 4)$ ，这个三元组可能更好地反应专家们的判断。

对三元组进行模糊集中的运算同样可以得到单个的数，其余步骤则是与原有的 AHP 方法相似，模糊集理论改进层次分析法的思想大致如上。

### 5.3 目标规划的应用

在 5.2 中本文分析目标规划在供应商选择和订单分配问题中的应用。

#### 5.3.1 目标的规划基本原理

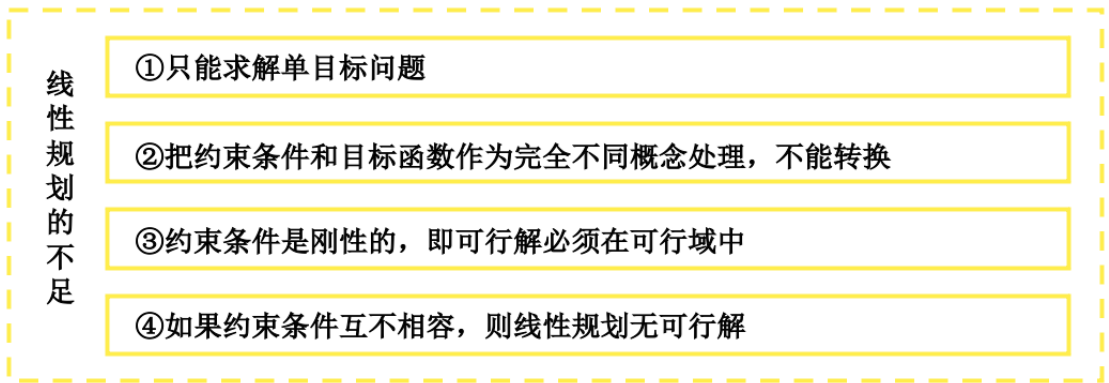


图 15 线性规划不足

相对于线性规划，目标规划（Goal Programming）可以解决多目标规划的问题，求出来的是一个满意解。线性规划求解的出来是满足所有约束条件，使目标函数最优的解，而目标规划允许约束条件有起差，得到的是尽可能达到目标的满意解。

目标规划的具体思想是先设定一个目标，要调整实际情况使得与目标的偏差尽可能小，即目标函数是偏差的和。目标规划与线性规划区别在于多了正偏差变量 $d^+$ 和负偏差变量 $d^-$ ，其中 $d^+$ 和 $d^-$ 非负。

举例而言，当企业的理想产量是 100 吨的时候，它不希望产量过多造成生产过剩，也不希望产品过少造成生产不足，如果实际生产量为 110 吨时，则正偏差变量 $d^+ = 10$ ，负偏差变量为 $d^- = 0$ ，如果实际生产量为 90 吨时，正偏差变量 $d^+ = 0$ ，负偏差变量为 $d^- = 10$ 。

目标产量	实际产量	正偏差变量 $d^+$	负偏差变量 $d^-$
100	110	10	0
	90	0	10

表 27 企业生产情况例子表

可以发现正负偏差变量其中一个必然为 0，并且可以发现如下等式

$$\underbrace{f_i(X)}_{\text{实际情况}} + \underbrace{d^-_i}_{\text{负偏差变量}} - \underbrace{d^+_i}_{\text{正偏差变量}} = \underbrace{b_i}_{\text{理想目标值}}$$

$d^+_i$ 和 $d^-_i$ 分别为正负偏差变量， $d^+_i$ 和 $d^-_i$ 非负，正负偏差变量用于描述实际情况与目标值的偏离情况，实际情况与目标值之间的偏离情况越小越好，所以目标函数为最小化偏差。

$$\min Z = d^-_i + d^+_i$$

如果目标规划中有多个目标，比如生产企业的理想生产量为 100 吨，理想污水排放量为 30 吨时，同样可以用目标规划进行分析，即使得所有目标与对应实际情况的偏差量之和最小，即

$$\min Z = \sum_{i=1}^n (d_i^- + d_i^+)$$

### 5.3.2 目标规划应用解决供应商问题

本文对 [Liao \(2010\) \[4\]](#) 利用目标规划解决供应商选择与订单分配的研究例子进行简化，来阐述用目标规划解决该类问题的方法。

假设采购商要从国外的供应商S1和S2进行选择采购，每家供应商的单位购买价格、单位运输成本和单位关税成本的如表所示。

供应商	单位购买价格	单位运输成本	单位关税成本
S1	30	5	3
S2	32	4	2

表 28 供应商情况表

采购商对自身采购情况进行评估，确定理想目标是购买成本为 3100 元，运输成本在 430 元左右，关税成本在 260 元左右，这即对应采购商的三个目标，采购商不希望实际情况偏离预期理想情况太远，因此可以建立目标规划。

$$\min Z = \underbrace{d_1^+ + d_1^-}_{\text{第一个目标偏离程度}} + \underbrace{d_2^+ + d_2^-}_{\text{第二个目标偏离程度}} + \underbrace{d_3^+ + d_3^-}_{\text{第三个目标偏离程度}}$$

$$\begin{cases} 30x_1 + 32x_2 + d_1^- - d_1^+ = 3100 \\ 5x_1 + 4x_2 + d_2^- - d_2^+ = 430 \\ 3x_1 + 2x_2 + d_3^- - d_3^+ = 260 \\ x_1, x_2, d_1^+, d_1^-, d_2^+, d_2^-, d_3^+, d_3^- \geq 0 \end{cases}$$

其中 $x_1$ 和 $x_2$ 为采购商向供应商S1和S2采购的数量， $d_i^-$ 和 $d_i^+$ 代表第 $i$ 个目标的正负偏差量，约束条件中，第一个等式的含义对应目标一购买成本尽量在 3100 左右，第二个等式的含义是目标二运输成本尽量在 430 左右，第三个等式的含义是目标三关税成本尽量在 260 左右。

通过 lingo 求解，得到应该同时选择采购商S1和S2，并给予采购商S1的订单量为 $x_1 = 34$ ，采购商S2的订单量为 $x_2 = 65$ ，此时购买成本为 3100，运输成本为 430，关税成本为 232，只有关税成本稍微偏差，负偏差量 $d_3^-$ 为 28。

当然我们也可以在目标函数中给予不同目标的偏离程度不同的权重，对目标的重要性进行调整，在规划中权重高的目标会被优先满足，其偏离程度较小。

通过这个例子，可以发现目标规划与线性规划的不同，在目标规划中会提前确定想要达到的目标，目标规划中的多个约束条件即对应这些想要达到的目标，而目标规划的目标函数则是偏差程度之和，进行规划时使得偏差程度达到最小，因此目标规划可以用于处理多目标的问题。

## 6 结论与待解决的问题

本文对 Operations Research 期刊上的论文“Supplier selection and order allocation using two-stage hybrid supply chain model and game-based order price” [1]进行深入讨论，并以该论文为主线对供应商选择和订单分配问题进一步分析。

在模型建立方面，本文复述了原论文阶段一的供应商选择和订单分配问题的模型和阶段二的供应商议价模型，清晰地陈述了模型的目标函数与约束条件的现实经济含义，即保证整条供应链成本最小化和效率最大化的要求下，对供应商进行筛选与分配订单，并通过议价模型确定合理价格。为了进一步理解论文中的数学原理，本文对数据包络分析和 Nash 议价均衡进行具体分析，发现数据包络分析是评价生产效率的一种重要方法，Nash 均衡则可以很好地反应局中人博弈后的均衡状态。

在模型求解方面，本文首先使用了求解器验证了原论文的数值实验结果。随后针对第一阶段模型提出了一种求解算法，针对第二阶段模型进行了理论分析，并求出了解析解。数值实验结果表明，本文提出的求解算法和解析解都是十分有效的。

在文献综述方面，本文围绕着供应商选择和订单分配的主线对该类研究进行整理，发现了许多将运筹学相关知识应用该类问题的论文，其中选取了层次分析法和目标规划进行讨论，发现了层次分析法和目标规划都是解决多评价标准或者多目标问题的有效手段，能够有效地选择出符合多维标准的供应商。

在进行上述工作之后，可以发现：供应商选择和订单分配问题是一个需要考虑多个目标和维度的问题，在进行规划时在满足采购双方的约束条件下，对多个目标函数进行同时优化，这些目标函数往往都是分式形式，可以使用分式规划的理论进行分析；协商价格的过程则是一个博弈的过程。总而言之，该类问题体现了实际生产规划与运筹学的关系，实际生产规划从现实意义上推动运筹学的发展，另一方面运筹学研究则是从理论上指引了日常生产工作的进行。

此外，针对原论文，本文还有如下问题未解决：尚未对模型进行参数灵敏度分析。



## 参考文献

- [1] Yousefi S, Rezaee M J, Solimanpur M. Supplier selection and order allocation using two-stage hybrid supply chain model and game-based order price[J]. *Operational Research*, 2019: 1-36.
- [2] Tone K, Tsutsui M. Dynamic DEA: A slacks-based measure approach[J]. *Omega*, 2010, 38(3-4): 145-156.
- [3] Tavana M, Shabanpour H, Yousefi S, et al. A hybrid goal programming and dynamic data envelopment analysis framework for sustainable supplier evaluation[J]. *Neural Computing and Applications*, 2017, 28(12): 3683-3696.
- [4] Liao C N, Kao H P. Supplier selection model using Taguchi loss function, analytical hierarchy process and multi-choice goal programming[J]. *Computers & Industrial Engineering*, 2010, 58(4): 571-577.
- [5] Deng X, Hu Y, Deng Y, et al. Supplier selection using AHP methodology extended by D numbers[J]. *Expert Systems with Applications*, 2014, 41(1): 156-167.
- [6] Chan F T S, Kumar N. Global supplier development considering risk factors using fuzzy extended AHP-based approach[J]. *Omega*, 2007, 35(4): 417-431.
- [7] Zhang X, Deng Y, Chan F T S, et al. A fuzzy extended analytic network process-based approach for global supplier selection[J]. *Applied Intelligence*, 2015, 43(4): 760-772.
- [8] Bohner C, Minner S. Supplier selection under failure risk, quantity and business volume discounts[J]. *Computers & Industrial Engineering*, 2017, 104: 145-155.
- [9] Mendoza A, Ventura J A. Modeling actual transportation costs in supplier selection and order quantity allocation decisions[J]. *Operational Research*, 2013, 13(1): 5-25.
- [10] Choudhary D, Shankar R. A goal programming model for joint decision making of inventory lot-size, supplier selection and carrier selection[J]. *Computers & Industrial Engineering*, 2014, 71: 1-9.
- [11] Choudhary D, Shankar R. A goal programming model for joint decision making of inventory lot-size, supplier selection and carrier selection[J]. *Computers & Industrial Engineering*, 2014, 71: 1-9.
- [12] Jadidi O, Cavalieri S, Zolfaghari S. An improved multi-choice goal programming approach for supplier selection problems[J]. *Applied mathematical modelling*, 2015, 39(14): 4213-4222.
- [13] Azadi M, Jafarian M, Saen R F, et al. A new fuzzy DEA model for evaluation of efficiency and effectiveness of suppliers in sustainable supply chain management context[J]. *Computers & Operations Research*, 2015, 54: 274-285.
- [14] Yousefi S, Mahmoudzadeh H, Jahangoshai Rezaee M. Using supply chain visibility and cost for supplier selection: a mathematical model[J]. *International Journal of Management Science and*

Engineering Management, 2017, 12(3): 196-205.

[15] Moheb-Alizadeh H, Handfield R. An integrated chance-constrained stochastic model for efficient and sustainable supplier selection and order allocation[J]. International Journal of Production Research, 2018, 56(21): 6890-6916.

[16] Hosseininasab A, Ahmadi A. Selecting a supplier portfolio with value, development, and risk consideration[J]. European Journal of Operational Research, 2015, 245(1): 146-156.

## 附录一：程序说明

本部分为程序包 coding.zip 的使用说明。

### a. 文件夹 gams

使用 gams (24.4) 系统调用 lindoglobal 求解器。

- 1、SCAS.gms: 最小化供应商总开销。
- 2、DEA.gms: 最大化供应商总效率。
- 3、stage1.gms: 第一阶段最优化程序 (选取权重为 0.5, 0.5)。
- 4、stage2.gms: 第二阶段最优化程序 (使用第一阶段最优化结果)。
- 5、SCASest.gms 新变量替换法测试, 订单分配问题, 选取供应商 1, 6, 7, 9。
- 6、SCASest2.gms 原论文变量替换法测试, 订单分配问题, 选取供应商 1, 6, 7, 9。
- 7、SCASest3.gms 新变量替换法测试, 订单分配问题, 选取供应商 1, 3, 4, 9, 10。

### b. 文件夹 matlab

使用 MATLAB (2021a) 实现基于遗传算法、分式规划、罚函数和梯度下降的第一阶段最优化算法。

- 1、punish\_main.m: 订单分配问题主函数, 可单独测试罚函数法, 可调整参数。
- 2、genetic\_main.m: 遗传算法主函数, 可调整遗传算法相关参数。
- 3、parameters.m: 自定义的参数类, 用于实现复杂参数在各函数间的传递。
- 4、fitness.m: 遗传算法适应度函数, 可调整遗传算法中关于罚函数的参数。
- 5、gen\_crossover.m: 遗传算法实现交叉操作的函数。
- 6、gen\_mutation.m: 遗传算法实现变异操作的函数。
- 7、gen\_selection.m: 遗传算法实现选择操作的函数。
- 8、min\_punish\_function.m: 实现罚函数迭代的函数, 可以输出显示每一次罚函数迭代结果 (见代码注释), 可以自行利用 draw\_x 和 draw\_con 两个变量绘图。

9、gradient\_descent.m: 实现梯度下降的函数,可以输出显示每一步梯度(见代码注释)。

10、obj.m: 计算目标函数值的函数。

11、obj\_gradient.m: 计算目标函数梯度的函数。

12、punish\_function.m: 计算罚函数值的函数。

13、punish\_function\_gradient.m: 计算罚函数梯度的函数。

14、t.m: 计算 t 值的函数。

15、t\_gradient.m: 计算 t 梯度的函数。

### c. 使用建议

1、可以直接运行 punish\_main.m 和 genetic\_main.m 获得本文大部分自行设计算法部分的结果;

2、不建议在使用遗传算法时开启输出,因为遗传算法中使用了并行计算,若您不在意命令行的杂乱或已对本程序有充分了解则无妨。

3、建议修改测试的代码有: punish\_main.m、genetic\_main.m、fitness.m, 其他代码不建议修改测试,若您对本文中涉及的公式以及本程序有充分的了解则无妨。

4、draw4\_2\_3.m 并非核心函数,如果您想验证 4.4.3 中的绘图,可以参考,但由于版本和使用环境,不保证一定能运行成功(需要在 min\_punish\_function.m 中开启绘图相关代码)。

## 附录二：遗传算法的完整求解结果

初始种群染色体									
1	1	0	1	0	0	0	1	1	0
1	0	1	0	0	1	1	0	1	0
0	1	0	1	1	0	1	0	1	1
1	1	1	0	0	0	1	0	1	0
1	1	1	1	1	1	0	0	0	0
0	1	1	0	1	0	0	0	1	1
0	1	0	0	0	1	0	0	0	0
1	0	0	1	1	1	1	1	0	0
1	0	0	1	0	1	0	0	1	1
1	0	1	0	0	0	1	1	0	0
0	1	0	1	1	1	1	1	0	1
1	0	1	1	1	1	0	1	1	0
1	1	0	0	1	1	0	1	1	1
0	0	0	0	0	1	0	1	0	1
1	1	0	0	0	1	0	0	1	0

0	0	0	0	1	0	0	1	0	0
0	0	1	1	1	0	1	0	1	0
1	0	1	1	0	1	0	1	1	0
1	1	1	1	0	0	0	1	0	0
1	0	0	1	1	0	1	0	1	0
1	0	1	1	0	0	0	0	1	1
0	1	1	0	1	0	0	0	1	0
1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	0	0
1	1	0	0	0	0	1	0	0	1
1	0	0	1	0	0	1	1	0	0
1	1	0	1	0	1	1	1	1	1
0	1	0	1	1	0	0	0	1	1
1	0	0	1	0	1	1	1	0	0
0	1	0	1	1	1	0	0	1	0
1	0	0	0	0	0	1	1	1	1
0	0	0	0	1	0	1	0	1	1
0	1	1	0	0	0	1	0	1	0
0	1	1	0	1	1	0	0	1	1
0	1	0	1	1	0	1	0	1	0
1	0	0	0	1	0	0	1	1	1
1	1	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0	0	1
0	1	0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	0	0	1
1	1	0	0	0	1	0	0	1	0
0	0	0	1	1	0	1	0	1	0
0	1	0	1	1	0	1	1	1	0
0	0	1	0	1	1	1	0	1	1
1	1	1	0	1	1	0	1	1	1
1	1	1	0	1	0	0	0	0	1
1	1	0	0	1	1	0	1	0	0
0	0	0	1	0	0	1	0	1	1
1	0	0	0	0	1	1	1	0	0
1	0	1	1	1	1	0	0	1	0
0	1	0	1	0	1	1	0	1	0
0	0	0	0	0	1	1	0	1	1
0	1	0	0	0	1	1	1	0	0
1	1	1	0	1	0	1	1	0	0
0	1	1	0	1	0	0	1	1	0
1	0	1	0	1	1	0	0	1	0

0	0	0	1	0	0	1	1	1	0
初始代染色体最优值(1,0,1,0,1,1,0,1,0) 0.071293									
第一代种群染色体									
1	1	0	1	0	0	0	1	1	0
1	1	0	1	0	0	0	1	1	0
1	0	1	0	0	1	1	0	1	1
1	0	1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0
0	1	1	0	1	0	1	1	0	0
1	0	0	1	1	1	0	0	1	1
1	0	0	1	0	1	0	0	1	1
1	1	0	1	0	1	0	0	1	1
1	0	0	1	0	1	0	0	1	1
1	1	0	0	0	0	0	0	1	0
0	0	1	1	0	1	0	1	1	0
1	0	0	1	1	0	1	0	1	0
1	1	0	1	1	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1
1	0	0	0	0	0	1	1	1	1
1	0	0	0	0	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	0	1	1
0	1	1	0	1	0	0	0	1	1
0	1	1	0	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	0	1	0
1	0	0	0	0	0	0	1	1	1
1	0	0	0	1	0	0	1	0	1
0	0	0	0	1	0	0	1	1	1
1	0	0	0	1	0	0	1	1	1
1	0	0	0	1	0	0	1	1	1
0	1	0	1	1	1	0	1	0	0
0	1	0	1	1	1	0	1	0	1
1	1	0	0	0	1	0	0	1	0
1	1	1	0	0	1	0	0	1	0
1	1	0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	0	1	0
0	0	0	1	1	0	1	0	1	0
1	1	0	1	1	0	1	1	1	0

0	1	0	1	1	0	1	1	1	0
0	0	1	0	1	1	1	0	1	1
1	1	1	0	1	0	0	0	0	1
1	1	0	1	0	0	1	0	0	1
0	0	0	0	1	1	0	1	0	0
1	0	1	1	0	1	1	0	1	0
0	1	0	1	1	1	0	0	1	0
1	1	0	1	0	1	1	0	1	0
0	1	0	1	0	1	1	0	1	0
0	1	0	1	0	1	1	0	1	0
0	1	0	1	0	1	1	0	1	0
0	0	0	0	0	1	1	0	1	1
0	0	0	0	0	1	1	0	1	1
1	1	1	0	1	0	0	1	1	0
0	1	1	0	1	0	1	1	0	0
0	0	0	1	0	0	1	1	1	0
0	1	1	0	1	0	0	1	1	0
0	0	0	1	0	0	1	1	1	0
0	0	0	1	0	0	1	0	1	0

第一代种群染色体最优值(1,0,1,0,0,1,1,0,1,0) 0.071293

第二代种群染色体

1	0	1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0	1	1
1	0	1	0	0	1	1	0	1	0
1	0	1	0	0	1	0	0	1	0
1	0	1	0	0	1	1	0	1	0
1	0	0	0	0	1	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	0	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	1	1	1	0	0	0
1	1	1	1	1	1	0	1	0	0
1	1	0	1	0	1	0	0	1	1
1	1	1	1	1	1	0	0	0	0
1	1	0	1	0	1	0	0	1	1
1	0	0	1	0	1	0	0	1	1
1	0	0	1	0	1	0	1	1	0
0	0	1	1	0	1	0	0	1	1
1	1	0	1	1	1	0	0	1	0
1	1	0	1	1	1	0	0	1	0
1	0	0	0	0	0	1	1	1	1

1	0	0	0	0	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	0	1	1
0	1	1	1	1	0	1	0	1	0
0	0	0	0	1	0	1	0	1	1
1	0	0	0	1	0	0	1	1	1
0	1	1	1	1	0	1	0	1	0
1	0	0	0	1	0	0	1	1	1
1	0	0	0	1	0	0	1	1	1
0	1	0	1	1	1	0	1	0	0
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
1	1	1	0	0	1	0	0	1	0
1	1	1	0	0	1	0	0	1	0
1	0	0	1	1	0	1	0	1	0
0	1	1	0	0	1	0	0	1	0
0	0	1	0	1	1	0	0	1	1
0	0	0	1	1	0	1	0	1	0
1	1	1	0	1	0	0	0	0	1
1	1	1	0	1	0	0	0	1	1
1	0	1	0	1	0	0	0	0	1
1	1	1	0	1	0	0	0	0	1
1	1	1	1	0	1	1	0	1	0
1	0	0	1	0	0	1	0	0	1
1	0	1	1	0	1	1	0	1	0
1	0	1	1	0	1	1	0	1	1
0	1	0	1	1	1	0	0	1	0
0	1	0	1	1	1	0	0	1	0
0	1	0	1	0	1	1	0	0	0
0	1	0	1	1	1	0	0	1	0
1	1	1	0	1	1	0	1	1	0
0	0	0	0	0	1	1	0	1	1
0	1	1	0	1	0	0	1	1	0
0	1	1	0	1	0	0	1	1	0

第二代种群染色体最优值(1,0,0,0,1,1,0,1,0) 0.07134

第三代种群染色体									
1	0	1	0	0	1	1	0	1	1
1	0	1	0	0	1	1	0	1	1
1	0	1	0	0	1	0	0	1	0
1	0	1	0	0	1	1	0	1	0
1	0	1	0	0	1	0	0	1	0

1	0	1	0	0	1	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	1	1	0	0	0	0
1	1	1	0	1	1	0	0	0	0
1	1	1	0	1	1	0	1	0	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0
1	0	0	1	1	1	0	0	1	0
1	1	0	1	0	1	0	1	1	0
1	1	0	1	1	1	0	0	1	0
1	1	0	1	1	1	0	0	1	0
1	0	0	0	0	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1
1	0	0	0	0	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	1	0	1
0	0	0	0	1	0	1	1	1	1
0	1	1	1	1	0	1	0	1	0
1	0	0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1	1	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	0
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
1	1	1	0	0	1	0	0	0	0
1	0	0	1	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	1	1	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0	1	0
0	0	0	1	1	0	1	0	0	0
1	0	1	0	1	1	0	0	1	1
1	1	1	0	1	0	0	0	0	1
1	1	1	0	1	0	0	0	0	1
1	1	1	0	1	0	0	0	1	1



1	1	1	1	1	0	0	0	1	1
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	0	1	0	0	1
1	0	1	1	0	1	1	0	1	0
0	1	0	1	0	1	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	1	0	0	1	0	0	0
0	1	0	1	1	1	0	0	1	0
0	0	0	0	0	1	1	0	1	1
0	0	0	0	0	1	1	0	1	1
0	1	1	0	1	0	0	1	1	0

第三代种群染色体最优值(1,0,1,0,0,1,1,0,1,0) 0.071293

第四代种群染色体

1	0	1	0	0	1	1	0	1	1
1	0	1	0	0	1	0	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	0	0	1	0	1	0
1	1	1	0	1	1	0	0	0	0
1	1	1	0	1	1	0	0	0	0
1	1	1	0	1	1	0	1	0	0
1	1	1	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0
1	0	0	1	1	1	0	0	1	0
1	1	1	1	1	1	0	0	0	0
1	0	0	1	1	1	0	0	1	0
0	0	0	1	1	1	0	0	1	0
1	0	0	1	1	1	0	0	1	0
1	1	0	1	0	1	0	1	1	0
1	1	0	1	1	1	0	0	1	0
1	1	0	1	1	1	0	0	1	0
1	0	0	1	0	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
0	0	0	0	1	0	1	1	1	1
1	0	0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	1	1	1
0	1	0	1	0	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1

0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
0	1	0	1	1	1	0	1	0	1
1	0	0	1	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	0	0	1	0	0	1	0	1	0
1	1	1	0	0	1	0	0	1	0
1	1	1	0	0	1	0	0	1	0
1	1	1	0	1	0	0	0	0	1
1	1	1	0	1	0	0	0	0	1
1	1	1	0	1	0	0	0	1	1
1	1	1	0	1	0	0	0	1	1
1	0	0	1	0	0	1	0	0	1
1	1	1	0	1	0	0	0	1	1
1	0	0	1	0	0	1	0	0	1
1	0	1	1	0	1	1	0	1	0
0	1	0	1	0	1	1	0	0	0
0	1	0	1	1	1	0	0	1	0
0	1	0	1	1	1	1	0	1	1
0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	1	1	0	1	1
0	0	0	0	0	1	1	0	1	1

第四代种群染色体最优值(1,0,1,0,0,1,1,0,1,1) 0.071252