**面向领域的计算机系统设计与开发**

面向企业采购的智能谈判平台设计与开发

**Design and Development of an Intelligent Negotiation Platform for Enterprise Procurement**

**谭家骐，王睿，张玉河，黄炳轮，姚睿哲**

|  |  |  |
| --- | --- | --- |
| 专业 | ： | 计算机科学与技术 |
| 学院 | ： | 计算机科学与技术学院 |

**哈尔滨工业大学**

**2024年11月**

# 目 录

[第1章 项目背景及意义 1](#_Toc280797187)

[1.1 项目背景 1](#_Toc280797188)

[1.2 项目意义 1](#_Toc280797189)

[第2章 项目相关研究及分析 3](#_Toc280797194)

[第3章 面向企业采购的智能谈判平台设计方案 4](#_Toc280797198)

[3.1 项目设计目标 4](#_Toc280797199)

[3.2 项目需求分析 4](#_Toc280797199)

[3.3 项目开发方案 6](#_Toc280797199)

[第4章](#_Toc280797198)  [面向企业采购的智能谈判平台具体实现 8](#_Toc280797198)

[4.1 项目实现方法 8](#_Toc280797199)

[4.2 项目架构及功能分析 9](#_Toc280797199)

[第5章 面向企业采购的智能谈判平台结果展示 1](#_Toc280797198)3

[第6章 面向企业采购的智能谈判平台的未来展望 2](#_Toc280797198)0

[第7章 小组分工 2](#_Toc280797198)1

[参考文献 2](#_Toc280797203)2

# 第1章 项目背景及研究的意义

## 1.1项目背景

在当今瞬息万变的商业环境中，谈判无疑是社会生活中不可或缺的一部分。特别是在商业领域，谈判不仅关乎企业的经济利益，也反映了企业的战略思维和市场洞察力。而在这些商业谈判中，企业采购谈判作为一个重要的环节，更是直接影响到企业的生产效率与成本控制。

企业在进行采购时，面对来自多家供应商的选择，需要就原材料的数量、价格、交货周期、质量标准等多种因素展开深入交流。每一轮谈判的成功与否，可能都会对企业的整体运营效率产生显著影响。然而，传统的采购谈判方式往往耗时耗力，沟通过程中容易出现信息不对称、决策不够科学等问题，导致谈判效率低下并增加成本。

因此，设计并开发一个面向企业采购的智能谈判平台显得尤为重要。通过利用先进的人工智能技术和数据分析工具，该平台能够自动化处理大量采购数据，优化谈判策略，实时评估市场行情，为采购人员提供科学的决策支持。此外，平台还可以建立公正透明的沟通渠道，确保信息的对称性，提高供应商之间的竞争性，最终实现交易条件的最优化。

在这个背景下，智能谈判平台不仅可以提高企业的采购效率，还能为企业创造更大的经济价值。通过智能化手段，企业能够在日益激烈的市场竞争中保持优势，实现可持续发展。这样的项目立足于未来，为企业采购谈判的现代化转型铺平道路。

## 1.2 项目意义

智能谈判作为人工智能的一个重要分支，已经在电子商务、资源分配和供应链管理等多个领域展示出广泛的应用前景。智能谈判平台的开发不仅可以提高谈判的效率，还能在多样化的商业环境中创造新的业务机会。然而，谈判环境的复杂性和对手行为的高度不确定性，使得智能体在制定有效策略时面临巨大挑战。尤其在动态和多变的谈判环境中，智能体不仅需要迅速应对对手的即时行为，还必须具备预见对手未来策略的能力。这种能力对于在谈判中占据主动地位至关重要，因为谈判的结果往往依赖于对对手行为和意图的准确判断。

开发一个面向企业采购的智能谈判平台，具有重要的研究目的和深远的实际意义。随着市场竞争的不断加剧，企业亟需能够快速响应市场变化的手段，以提升自身的竞争优势。智能采购谈判平台不仅能够帮助企业在激烈的竞争中立于不败之地，还能通过智能化技术的广泛应用，简化和加速采购谈判过程，减少人工干预和操作时间，显著提升整体谈判效率。同时，通过数据驱动的决策流程，平台能够增强采购过程的透明度，提升供应商之间的信任关系，促进合作的稳定性。

此外，借助智能化的功能设计，平台能够有效改善用户体验，减轻采购人员的工作负担，使其能够更专注于战略性决策而非重复性的操作。引入先进的算法和数据分析工具，能够实时监控市场趋势和变化，提供实时的市场信息和见解，让企业在谈判中做出更加明智的选择。因此，这一研究不仅将为企业带来直接的经济效益，还为整个行业的创新与进步奠定了坚实的基础，推动商业智能化的未来发展。通过这样的智能谈判平台，企业将能够更好地适应复杂的市场环境，为其长期可持续发展提供支持。

第2章 国内外研究现状及分析

## 2.1 智能谈判相关理论的发展概况

随着人工智能技术的发展，智能谈判作为自动化决策的重要领域，近年来取得了显著进展。智能谈判的目标是通过代理系统自动化谈判过程，以提高效率并减少人类谈判中的主观性影响。智能谈判技术主要研究如何模拟真实谈判场景，如何在谈判中使用自动化谈判代理，提高谈判效率，优化谈判结果。从场景上说，大致分为多Agent谈判和人机谈判。前者主要用于研究智能谈判Agent在谈判场景的谈判策略，后者涉及人机交互、情感计算、语言分析等多方面综合内容。智能谈判在多领域中都有广泛的应用，如外交、电子商务等，也涉及社会学、心理学、博弈论等多个交叉领域的相关内容。

在计算机领域的智能谈判相关技术中，近年来的研究重点是如何构建智能体，使其能够自主进行决策，并根据谈判环境中的变化实时调整策略，这主要涉及的是对手建模技术（2.1.2节中详细介绍）。其他的相关技术包括，谈判Agent需要设计谈判场景，提供多个Agent高效进行谈判的环境，进行谈判效果检验。已经有许多成熟的研究，开发智能谈判平台和框架，例如基于java开发的Genius平台，以及基于python开发的negmas库。这些平台为研究智能谈判代理的行为和策略提供了测试环境。基于此类测试环境，其他研究则致力于在不同领域（如电子商务和供应链管理）中应用智能谈判技术，通过自动化谈判系统提升谈判效率并优化资源分配。

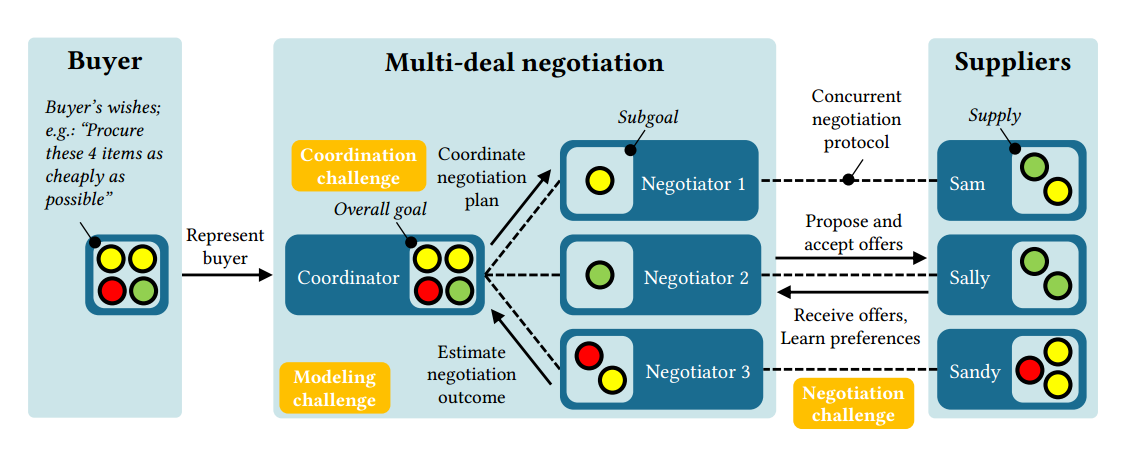
此外，为促进智能谈判研究，国际自动谈判代理大赛（ANAC）从2010年开始每年举办，最新一届的ANAC于2024年5月顺利召开。ANAC提供了丰富多样的谈判议程，各类优秀的智能代理参与竞赛，提供了良好的交流和研究环境，也展示了各智能代理在复杂的多议题谈判场景中的卓越表现。

# 第3章 面向企业采购的智能谈判平台设计

# 3.1 项目设计目标

本项目预计设计一个应用在如下场景的智能谈判系统：

* 采购场景：1个买方，同时与N个卖方就M个物品的采购数量和价格进行谈判。
* 谈判流程：每轮卖方向买方提供各商品报价和可购买数量，买方向每个卖方提供商品可接受价格和需求数量。
* 购买行为：每轮卖方报价后，买方可购买若干卖家提供的商品组合
* 时间限制：谈判规定最大时长/轮次
* 买家需求：购买的所有商品满足数量要求。
* 买卖方偏好：在交易谈判过程中争取自身效用最大化。



# 3.2 项目需求分析

#### **用户角色**

* **买方（企业采购人员）**
  + 负责发起谈判，提交需求，评估报价，并最终做出采购决策。
* **卖方（供应商）**
  + 提供报价和可供数量，参与谈判以争取更好的交易条件。

#### **系统功能需求**

* **谈判管理模块**
  + **创建谈判**: 允许创建新的采购谈判，设定相关参数（如最大时长/轮次）。
  + **轮次管理**: 管理每轮谈判的报价和反馈，记录轮次信息。
* **报价管理模块**
  + **报价提交**: 卖方能够在指定时间内提交商品的报价和可供数量。
  + **买方反馈**: 买方能够针对每个卖方的报价给予反馈，包括是否接收以及返回报价
* **需求匹配模块**
  + **需求分析**: 根据买方提供的需求，分析满足条件的卖方报价。
  + **商品组合选择**: 在每轮结束后，允许买方选择合适的商品组合以满足需求。
* **决策支持模块**
  + **效用最大化计算**: 集成效用函数算法，帮助买方和卖方评估各自的效用，支持双方在谈判过程中做出更好的决策。

#### **非功能性需求**

* **用户体验**
  + 界面设计需简洁明了，确保用户能够快速理解和操作谈判流程。
  + 实现人在回路中的智能谈判，用户可以同意智能谈判agent的决策，也可以随时介入修改谈判决策

#### **技术需求**

* **技术栈**
  + 使用 Python 和 negmas 库来实现智能谈判算法。
  + 前端框架（如 React 或 Vue）用于构建用户界面。
  + 数据库用于存储用户、报价和谈判流程的数据。
* **依赖管理**
  + 确保库版本的兼容性，进行定期更新以维护系统的安全性和稳定性。

### **流程设计**

* **谈判流程**
  1. 初始化双方谈判参数，卖方提交报。
  2. 买方查看报价，并通过买方Agent进行评估，并提供反馈，可以选择接收报价，也可以返回新的报价。
  3. 根据反馈，卖方可以调整报价，再反馈回给买方，直到买方接受最终报价。
  4. 谈判完成，更新数据记录，生成报告。

### **预期输出**

* 明确的系统文档，包括用户手册和API文档。
* 持续的用户反馈与系统迭代，确保满足实际需求并解决问题。

# 3.3 项目研究主要方案

我们采用软件工程的过程框架进行该项目的实现，分为以下五个部分：

1. 沟通：小组讨论，沟通理解该项目目标，并收集需要完成的需求。具体收集的需求见上文。
2. 策划：确定了我们需要完成的工作的技术任务，包括熟悉negmas库的使用，前端的设计，确定了核心的技术难点，根据课程目标确定了开题、中期和结题的工作进度。
3. 建模：确定了我们的项目模型和各模块的设计，以满足项目需求。
4. 构建：撰写相关代码并测试，同时优化代码以适应后续修改。
5. 部署：通过课程展示和内部测试，收集反馈并持续优化和修改。

第4章 面向企业采购的智能谈判平台实现

# 4.1 项目实现方法

### 相关技术栈

1. **编程语言**
   * **Python**: 主要编程语言，用于实现谈判逻辑及用户交互。
2. **库和框架**
   * **NegMAS**: 用于智能谈判的专用库，提供了多种谈判机制和工具。
   * **ThreadPoolExecutor**: 用于管理并发和多线程操作。
   * **Matplotlib**: 用于可视化谈判结果和分析数据。
3. **用户界面**
   * **命令行界面（CLI）**: 通过终端输入进行用户交互（可以扩展为Web界面）。
4. **并发处理**
   * **Threading**: 用于处理多线程任务，支持多个卖方同时进行谈判。
5. **数学和优化**
   * **线性与非线性聚合效用函数**: 用于确保买卖双方依照预计效用进行报价及决策。
   * **自定义效用函数**: 使用AffineFun与LinearFun类来计算效用值。

# 4.2 项目架构及功能分析

# 

* **Main Module**:
  + **Main**: 程序的入口，负责初始化和管理主线程。
* **Threading Module**:
  + **ThreadTasks**: 处理线程的任务逻辑，与 NegotiationSession 模块交互。
* **Negotiation Module**:
  + **NegotiationSession**: 负责谈判逻辑，包括创建议题和计算效用。
  + **SmartAspirationNegotiator**: 实现谈判策略的逻辑。

**主要功能分析：**

NegotiationSession 定义了一个谈判会话，用于模拟多个卖家与买家的谈判过程。类中的核心功能包括定义谈判商品、卖家的效用函数、以及执行谈判过程。下面是对该类的功能和主要函数的分析：

**1. 初始化函数 \_\_init\_\_**

* **功能**: 初始化谈判会话的各种属性。
* **初始化的成员变量**:
  + price\_ranges 和 quantity\_ranges：分别是价格和数量区间，用来约束卖家在谈判中可接受的价格和数量范围。每个卖家的每个商品都有一个区间。
  + issues：每个卖家拥有的议题列表，包括价格和数量问题，这些问题会在谈判中与买家交换。
  + favors：每个卖家的偏好系数，用于影响卖家的效用函数。

**2. create\_issues\_with\_ranges**

* **功能**: 为每个卖家生成议题（price 和 quantity），即商品的价格和数量问题。这些问题会在谈判中作为交易的一部分进行调整。
* **参数**: seller\_index，表示卖家的索引。
* **返回值**: 为指定卖家创建一个议题列表，每个议题包含商品的价格和数量区间。

**3. create\_buyer\_utility**

* **功能**: 创建买家的效用函数。效用函数用于计算买家对交易结果的满意度。
* **参数**: session（谈判会话对象）、seller\_index（卖家索引）。
* **返回值**: 返回一个非线性的效用函数，买家的效用函数由价格和数量的权重组成，其中价格的效用函数是负向的（表示价格越高，买家越不满意），数量的效用函数是线性的（表示数量越多，买家越满意）。

**4. create\_seller\_utility**

* **功能**: 创建卖家的效用函数。根据传入的 utility\_type 和卖家的偏好 favor，选择不同类型的效用函数来描述卖家对价格和数量的偏好。
* **参数**: session（谈判会话对象）、seller\_index（卖家索引）、utility\_type（效用类型）、favor（卖家的偏好）。
* **返回值**: 返回一个非线性的效用函数，效用函数的计算方式根据不同的 utility\_type 进行选择，如 "no\_favor"（没有偏好），"no\_quantity"（没有数量影响），以及默认类型。

**5. set\_seller\_favor**

* **功能**: 设置指定卖家的偏好系数 favor，影响卖家的效用函数。
* **参数**: seller\_index（卖家索引），favor（偏好系数，通常是一个三元素列表，表示对价格、数量的偏好）。
* **返回值**: 无。

**6. create\_seller**

* **功能**: 根据随机选择的策略创建一个卖家，卖家使用不同类型的谈判策略（如 LinearTBNegotiator、FirstOfferOrientedTBNegotiator 等）。
* **参数**: index（卖家索引），ufun（卖家的效用函数）。
* **返回值**: 返回一个创建的卖家对象。

**8. run\_negotiation**

* **功能**: 执行谈判过程。
* **参数**:
  + shared\_string、results、index、thr\_mutex：用于多线程共享和同步的参数。
* **流程**:
  + 为买家和卖家创建效用函数。
  + 创建一个谈判会话 session，并添加买家和卖家的谈判者。
  + 设定买家的效用函数，并根据卖家的效用函数创建相应的卖家对象。
  + 执行会话中的谈判过程，最终得到一个结果。
* **返回值**: 返回谈判的最终结果。

SmartAspirationNegotiator 类继承自 SAONegotiator 类，实现了人机交互与建议报价智能体。这个类的目的是在多轮谈判中接受或拒绝报价，并根据用户的输入动态调整谈判策略。下面是对主要方法的分析：

**1. set\_helper 方法**

* **功能**: 设置辅助谈判者 SAONegotiator。
* **实现**:
  + 将传入的 SAONegotiator 类型的对象赋值给 self.helper，helper 会在谈判过程中提供支持，特别是在用户拒绝报价时，helper 会提出新的报价。

**2. respond 方法**

* **功能**: 处理代理的响应，决定是否接受报价，或者如果拒绝报价则向辅助谈判者请求新的报价。
* **实现**:
  + 获取当前的报价 offer，如果报价为 None，则返回拒绝报价 (ResponseType.REJECT\_OFFER)。
  + 显示当前报价并等待用户输入。
  + 如果用户输入 'y'（接受报价），返回接受报价的响应（ResponseType.ACCEPT\_OFFER）。
  + 如果用户输入 'n'（拒绝报价），则让 helper（辅助谈判者）响应当前的状态，并请求新的报价。helper 会基于当前状态提出一个新的报价。
  + 最终，如果拒绝报价，则代理返回拒绝报价的响应。

**6. user\_propose 方法**

* **功能**: 由用户输入新的报价，并将其转换为一个元组形式。
* **实现**:
  + 使用 input\_i 函数获取用户输入的报价，报价是一个以逗号分隔的字符串（例如 "10, 8, 12, 10, 11, 7"）。
  + 将用户输入的字符串拆分并转换为浮动类型的元组（tuple(map(float, user\_offer.split(',')))）。

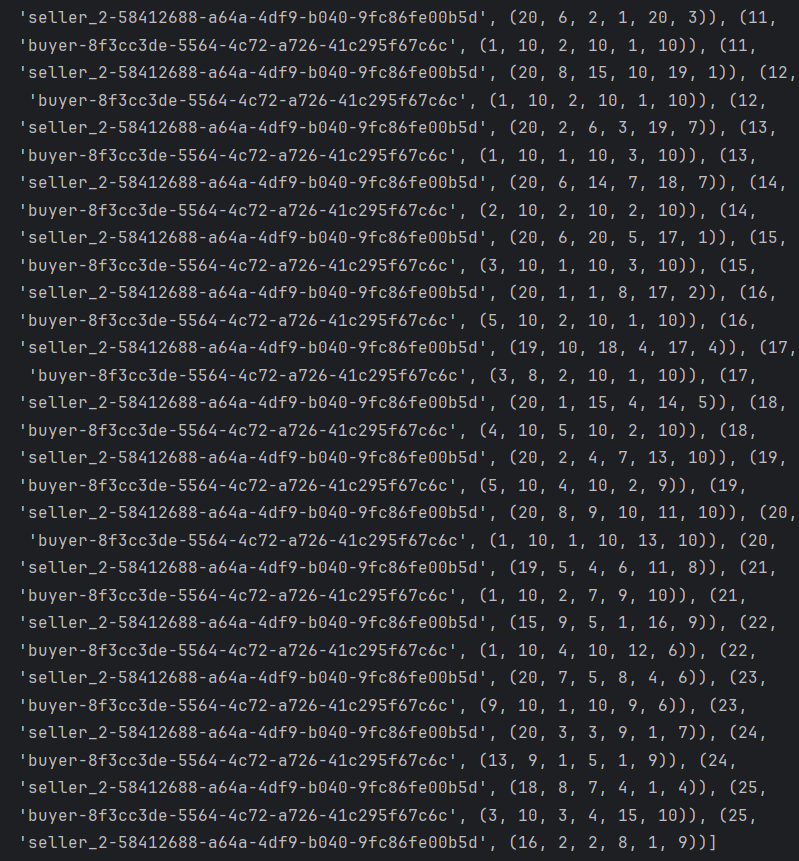
**7. propose 方法**

* **功能**: 提出一个报价。
* **实现**:
  + 调用 user\_propose 方法获取用户输入的报价，并打印该报价。
  + 返回用户提供的报价，作为谈判中的提议。

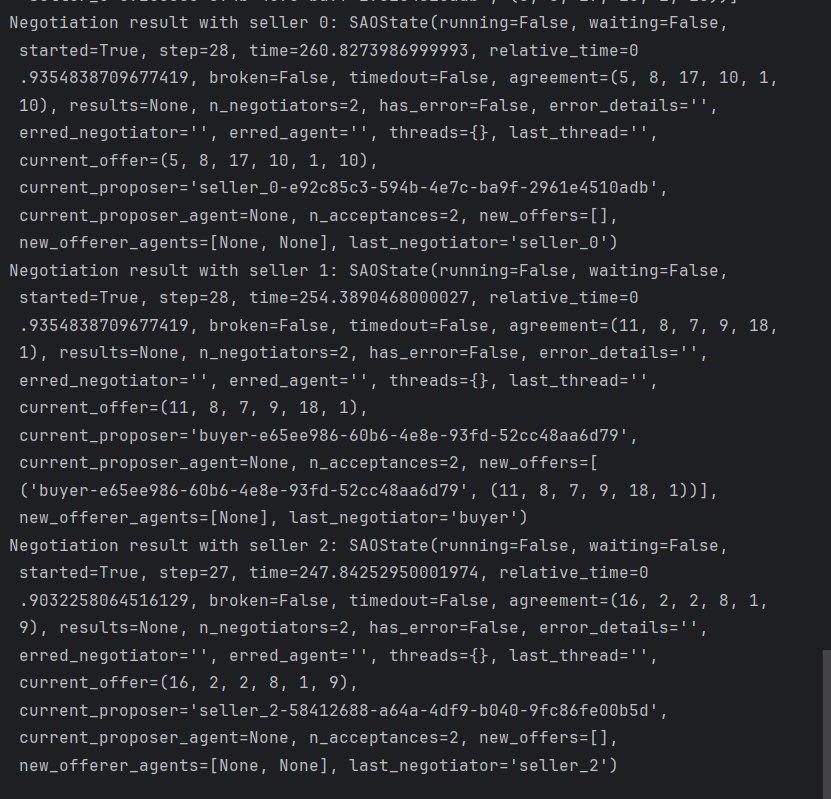
# 第5章 面向企业采购的智能谈判平台结果展示

# 5.1Agent对Agent成果展示

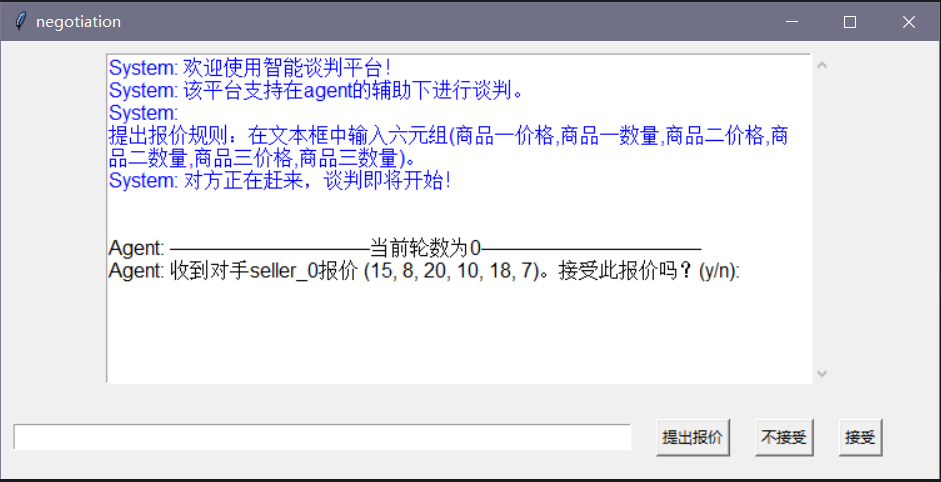
卖家agent与买家agent的交替出价记录



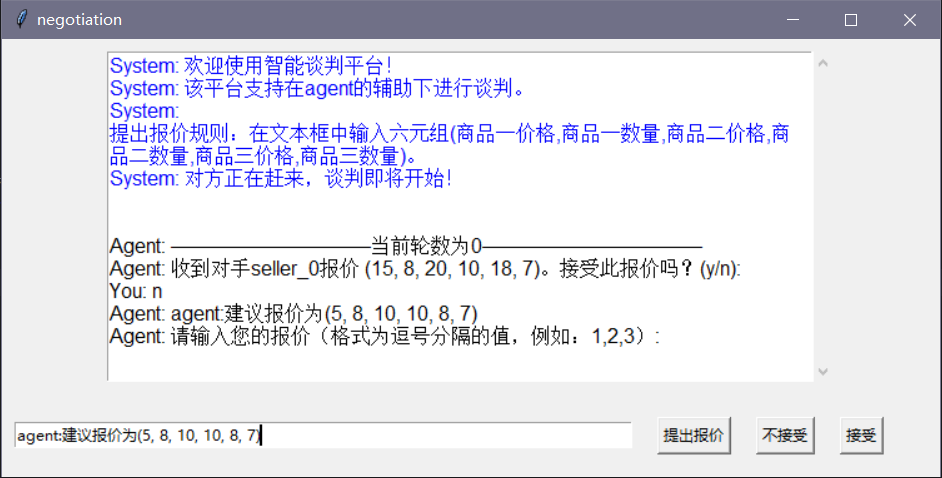
谈判结果记录



# 5.2单线程成果展示

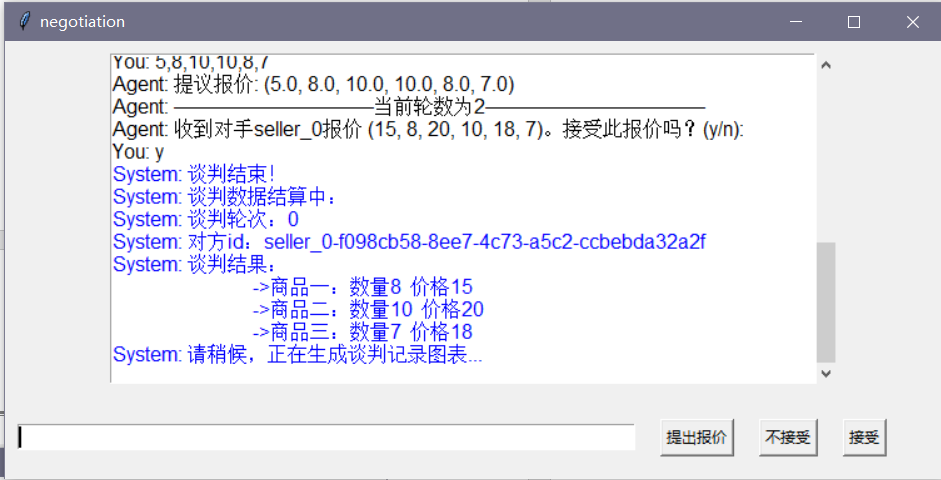


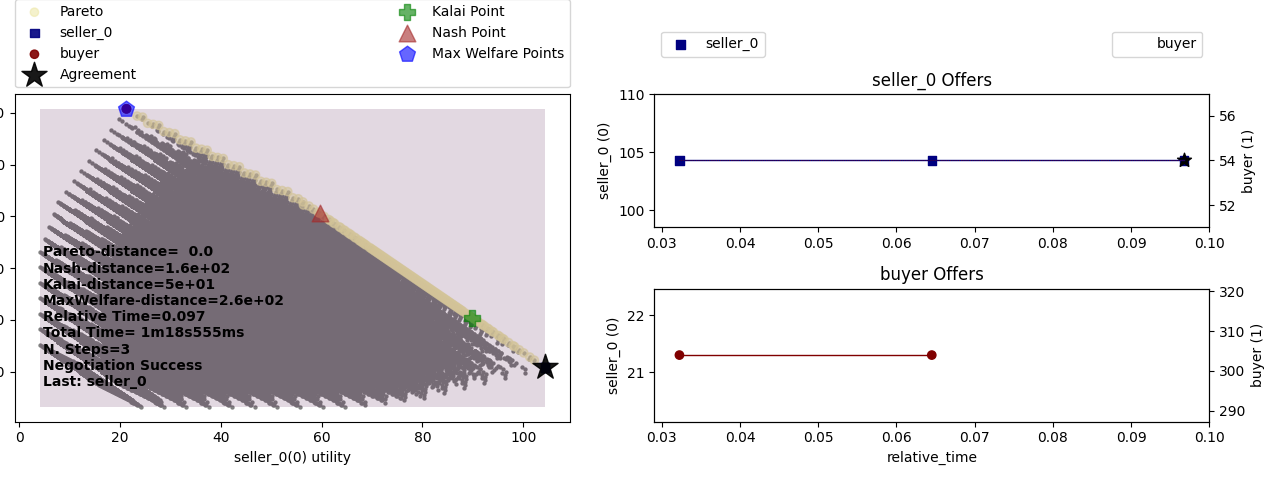
选择拒绝对方报价后，可以直接选择智能体的推荐报价也可以输入自定义报价





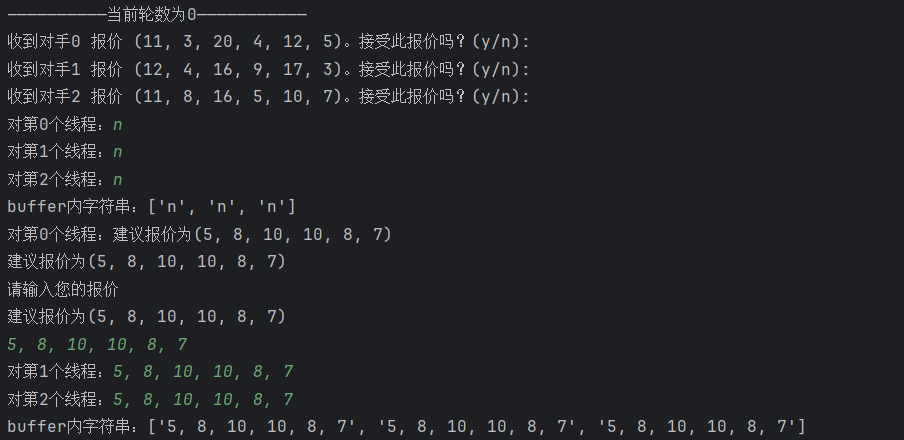
当买方选择接受报价后。谈判结束，针对谈判记录绘制图表





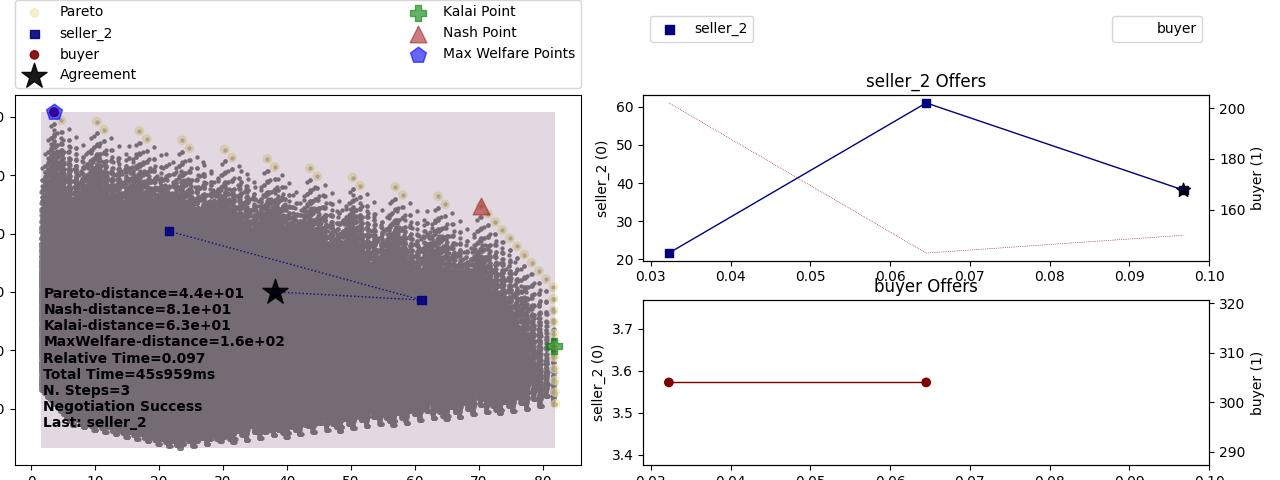
# 5.3多线程成果展示：

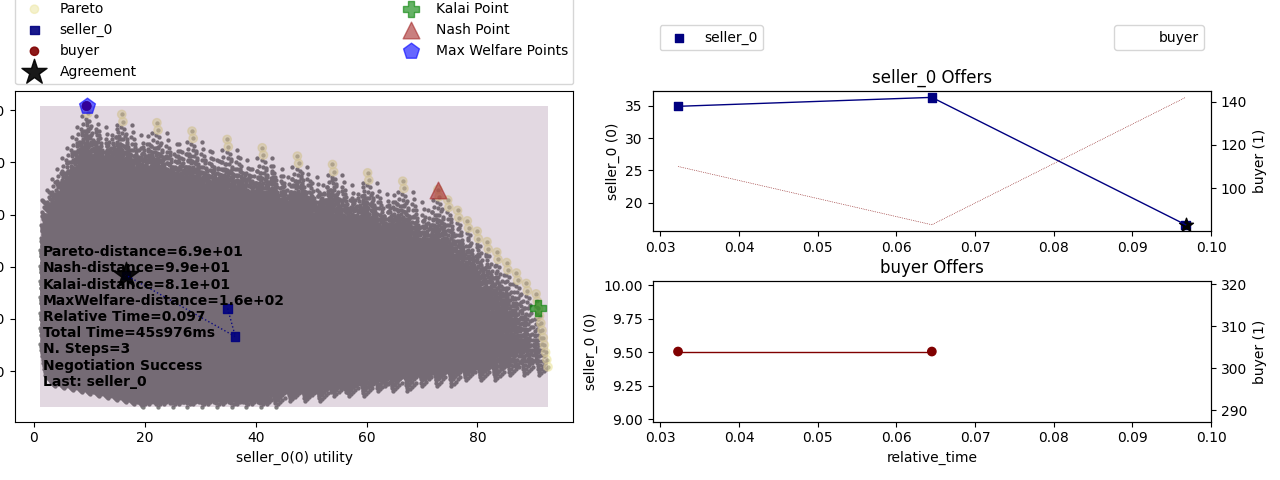
一对多谈判时，买方依次选择是否结束对方报价，并更具智能体参考报价依次回复自己的报价



# 选择结束谈判后，将总结与每个卖方的谈判记录，并计算效益函数与绘制图表

# 





# 第6章 面向企业采购的智能谈判平台的未来展望

针对目前平台的开发情况，我们对于平台目前存在的问题进行分析，主要存在以下几点不足：

1. 平台的谈判环境设计较为简单。
2. 平台提供的的谈判策略的复杂性不高，没有接入目前最先进的ANAC谈判竞赛中的模型在谈判环境中测试。
3. 受限于开发水平，平台的响应速度不高。底层实现使用SAOmechanism的堆叠交替出价协议，需要在初始化时对样本空间进行遍历，在谈判空间较大时计算效率较低。
4. 人机交互设计较为简陋，缺乏人性化、智能化的响应设计。

针对现有不足，我们提出如下展望：

1. 修改底层协议，重写一对多采购的交易机制，或考虑使用SCM(supply chain management)供应链模式重构项目。
2. 接入更多复杂策略模式，综合评判谈判模型的效果。
3. 优化模块设计，提高代码运行效率和可靠性。
4. 增加更多人机交互的可视化内容，针对现有推荐报价功能予以报价逻辑说明和预期交易可能的图表和文字展示。

# 第7章 小组分工

组长：谭家骐

项目组组长，负责工作的统筹规划，进行采购场景建模的方法研究，编写后端代码，参与报告撰写，制作汇报ppt以及汇报。

成员：王睿

项目组成员，项目需求的分析，项目代码架构重构，整合项目进度，参与报告撰写，参与制作汇报ppt。

成员：张玉河

项目组成员，进行买卖方的采购策略的研究，编写后端代码，参与报告撰写，参与制作汇报ppt。

成员：黄炳轮

项目组成员，进行项目多线程和前端人机交互内容搭建，编写代码，参与制作汇报ppt。

成员：姚睿哲

项目组成员，进行前端人机交互内容搭建，编写代码，参与制作汇报ppt。

# 参考文献

1. Mell J, Gratch J. IAGO: interactive arbitration guide online[C]//Proceedings of

the 2016 International Conference on Autonomous Agents & Multiagent Systems. 2016: 1510-1512.

1. Mohammad Y, Nakadai S, Greenwald A. Negmas: a platform for situated negotiations[C]//Recent Advances in Agent-based Negotiation: Formal Models and Human Aspects. Springer Singapore, 2021: 57-75.
2. Chatfield D C, Harrison T P, Hayya J C. SCML: An information framework to support supply chain modeling[J]. European Journal of Operational Research, 2009, 196(2): 651-660.
3. Baarslag T. Multi-deal Negotiation[C]//Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems. 2024: 2668-2673.