A Game Theoretic Approach to Resource Allocation in Minimum Intrusion Grid (MiG)

René Brask Sørensen (brask@diku.dk)

September 15, 2006

Contents

1 Introduction

Game theory is a field of mathematics and microeconomics that is used to study strategic interaction between two or more rational, self-interested agents that seek to maximize their utility. In this project I will describe the resource allocation mechanism of MiG by interpreting the source-code, and based on that, use game theory to analyze the resource allocation mechanism. By modelling the MiG mechanism as a system of self-interested agents competing in an auction, I will prove that the bidding strategies of resources are not optimal and thereby conclude that the current mechanism is not optimal. I.e., by imposing a better strategy, resources will gain more profit.

Based on the analysis of the current MiG mechanism I will design a new and improved mechanism using mechanism design which is the sub-field of microeconomics and game theory that considers how to implement system-wide solutions to problems involving multiple self-interested agent, each with private information about their preferences. In recent years mechanism design has shown to be suitable within many computational applications, e.g. online auctions, Internet routing, and distributed scheduling problems.

When designing a new mechanism for MiG I will consider the aspects of computational mechanism design which is a new and emerging field of mechanism design. In computational mechanism design both tractability and good game theoretic properties are considered. In classic mechanism design computational considerations are largely ignored, e.g. it is common to assume that agents can reveal their complete preferences over all possible outcomes and that the mechanism can solve an optimization problem to select the best outcome. In computational mechanism design these assumptions can incur both expensive communication and computation costs which make the mechanisms poor.

To verify the correctness of my proposed mechanism, I will implement a prototype of the mechanism. The prototype will be tested by running it in a simulation of the MiG system, provided by the MiG team.

2 Introduction to Game Theory and Mechanism Design

This section provides a brief introduction to the fundamentals of game theory and mechanism design. Only concepts relevant to the project will be treated, including various definitions used in the following chapters. This section can be skipped if familiar with game theory and mechanism design. For a more detailed introduction, Fudenberg and Tirole [7] and Gibbons [1] provide a good introduction to game theory with a economic perspective. Myerson [4] provides a more precise and advanced treatment of the subject, including proofs of various definitions. In [5] Parkes give a overview of mechanism design from a computational point of view, and in [3] Krishna gives a thorough description of mechanism design and auction theory.

In the following I will start the introduction by explaining agent behaviour and define the concept of utility and profit maximization in section 2.1. Proceeding in section 2.2 with a description of the fundamentals of game theory, including definition of relevant solution concepts. In section 2.3 I will explain the concept of mechanism design and define relevant concepts and properties.

2.1 Decision and Utility Theory

Game theory is the study of system-wide solutions to problems involving two or more individuals making decisions that will influence one another's welfare. The individuals involved in a game may be called agents or players. And in order to predict anything about these agents we have to make some assumptions about their behaviour. In game theory it is a common assumption that agents are intelligent rational decision-makers. An agent is rational if she makes decisions consistently in pursuit of her own objectives. In game theory, we assume that each agent's objective is to maximize the expected value of her own payoff, which is measured in some utility scale. An agent is intelligent if he knows everything that we know about the game and she can make any inferences about the game that we can, see [4, chap. 1] for a detailed description of rationality and intelligence. I will sometimes refer to a rational and intelligent agent as simply a self-interested agent.

Utility When reasoning about an agent's utility, I will use the words utility, payoff, and profit interchangeably, which is also the case in most of the literature. An agent's utility can be expressed by the type of an agent, which determines the preferences of an agent over different outcomes of a game.

Definition 1 (Utility)**.** *Let $\theta \in \Theta$ denote the type of an agent i, from a set of possible types Θ. An agent's preferences over different outcomes $o \in O$ for a* *set O of possible outcomes, can then be expressed in terms of a utility function that is parameterized on the type θ. The utility to agent i is denoted $u_i(o, \theta_i)$ and it is the assumption that a rational agent seeks to maximize the utility.*

$$\max_{\theta \in \Theta} \quad u_i(o, \theta_i)$$

Uncertainty In situations of uncertainty I will define agent utility using the von Neumann-Morgenstern expected utility function, which is defined for agents in environments with uncertain payoff of different outcomes, see [8, sec. 12.3] for a detailed description. To model agent behaviour in situations of uncertainty about different outcomes we have to make assumptions about agent's risk-willingness. In mechanism design and auction theory it is a common assumption that agents are risk neutral and have quasi-linear preferences, which will be defined below. I will pursue this assumption in this project unless stated otherwise.

When an agent is risk neutral, she is indifferent between participating in a lottery with a 50 % change of winning \$100 and receive \$50 with certainty. It is because her expected payoff in both cases is equal to 50, in the lottery there is a 50 % change of winning \$100 which gives utility $u = 0.5 * 100 = 50$ which is just the same as receiving \$50 with certainty.

Quasi-linear Preferences As stated above I will assume that agents have quasi-linear preferences in this project unless stated otherwise.

Definition 2 (Quasi Linear Preferences)**.** *A quasi-linear utility function for agent i with type θ_i is of the form :*

$$u_i(o, \theta_i) = v_i(x, \theta_i) - p_i$$

where outcome o defines a choice $x \in \mathcal{K}$ from a discrete choice set and a payment p_i by the agent. $v_i(\cdot)$ is a valuation function specifying agent i's valuation for a choice x, given the type θ_i.

The quasi-linear utility function is composed of a linear fraction, the valuation function $v(\cdot)$ and a constant fraction, the payment p_i. In a allocation problem the alternatives $\mathcal{K}$ represent different allocations, and the payment p_i represent the payments to the auctioneer. See [8, chap. 4] for a thorough description of quasi-linear preferences.

With the definition of agent utility and types, it is now possible to make a preference ordering over different outcomes. I.e. it is possible for an agent to state her preferences over different outcomes. The type of an agent determines, in the case of mechanism design and auction theory, her valuation of the item auctioned over. E.g. if the item is a painting by Da Vinci and the agent thinks it is worth \$1 million., her valuation or type is equal to 1 million.

To talk about agent strategies we need the utility definition for determining the preferences over different outcomes. I.e. by the usage of a utility function it is possible to determine which actions produces the highest amount of utility.

Strategy A strategy for an agent is a complete plan of action, it specifies a feasible action for the agent in every contingency in which the agent might be called upon to act. Let $s_i(\theta_i) \in \Sigma_i$ denote the strategy of agent i given type θ_i and Σ_i is the set of all possible strategies available to agent i. Without loss of generality I will sometimes let the conditioning on an agent's type be left implicit, and write s_i for the strategy selected by agent i given its type.

2.2 Game Theory

Recall the concept of game theory is to find a solution to a system-wide problem involving two or more agents. To give a more thoroughly elaboration of game theory, I will start by defining the structure of a game in section 2.2.1, and then move on with a description of different types of games and their solutions in section 2.2.2. Lastly I will compare these solution concepts in section 2.2.3.

2.2.1 Structure of a Game

In game theory the two most common definitions of a game is the strategic- and extended form. All games can be defined by both definitions, but normally one prefer to describe static games in strategic or normal form and dynamic games in extended form. A game is static if all agents in the game move simultaneously. A game is dynamic if there is more than one period in a game and agents interact. Unlike static games, agents have at least some information about the strategies chosen by others, and thus may contingent their play on past moves. An example of a dynamic game is poker where player's strategies may depend on past moves.

A game defined in strategic form includes a specification of the set of agents in the game N, the set of options available to each agent, and the way that agent's payoff depend on the options that they chose. Formally a strategic form game is defined as

$$G = \{N; \Sigma_1, ..., \Sigma_n; u_1, ..., u_n\}$$

Specifying the number of agents participating in the game, the strategy space available to each agent and their payoff function.

An extension of the strategic form game is used when describing Bayesian games with incomplete information. When information is incomplete agents have private information, i.e. their preferences are private. A game defined as a Bayesian form game specifies the set of agents in the game, N, the set of actions available to each agent A_i, the set of possible types available to each agent Θ_i, a probability function representing their beliefs p_i, and a payoff function u_i. The formal definition is

$$G = \{N; A_1, ..., A_n; \Theta_1, ..., \Theta_n; p_1, ..., p_n; u_1, ..., u_n\}$$

Agent i's type θ_i is private knowledge, only known by agent i. The expected payoff received by agent i is determined by the actions of all agents and the type θ_i of agent i

$$u_i = (a_1, ..., a_n; \theta_i)$$

Agent i's beliefs $p_i(\theta_{-i}|\theta_i)$ describes i's uncertainty about the $n-1$ other agent's possible types, θ_{-1}, given i's own type θ_i. In other words $p_i(\theta_{-i}|\theta_i)$ denote the subjective probability that i would assign to the event that θ_{-i} is the actual profile of types for the $n-1$ agents, if i's own type is θ_i. Having defined the structure of a game, we move on to defining the solution or equilibrium of the game.

2.2.2 Solution concepts

In game theory the concept of computing a equilibrium of a game with rational agents, given assumptions about agent preferences, and information available to an agent about other agents is called a solution concept and differs from game to game. The most well-known and relevant for the project are

- Dominant Strategy equilibrium
- Nash equilibrium
- Bayesian Nash equilibrium

I the following I will define and describe each of them.

Definition 3 (Dominant Strategy Equilibrium). *In a strategic-form game* $G = \{N, \Sigma_1, ..., \Sigma_n; u_1, ..., u_n\}$*, let* s_i *and* $s_i^{'}$ *be feasible strategies of agent* i*, i.e.* s_i *and* $s_i^{'}$ *belongs to the set* Σ_i*. Strategy* s_i *is a dominant strategy if it (weakly) maximizes agent* i*'s expected utility for all possible strategies of the other agents,*

$$u_i(s_i, s_{-i}) \geq u_i(s_i^{'}, s_{-i}), \text{ for all } s_i^{'} \neq s_i, s_{-i} \in N$$

Thus in other words a dominant strategy s_i is a best response strategy to agent i if it maximizes expected utility, *whatever* the strategies played by the other agents. As we will see later, this is a very nice property if concerned about tractability.

Definition 4 (Nash Equilibrium). *In the strategic-form game* $G = \{N, \Sigma_1, ..., \Sigma_n; u_1, ..., u_n\}$*, the strategy profile* $s^* = (s_1^*, ..., s_n^*)$ *are a Nash equilibrium if, for each agent* i*,* s_i^* *is agent* i*'s best response strategy to the strategies specified for the* $n-1$ *other agents,* $(s_1^*, ..., s_{i-1}^*, s_{i+1}^*, ..., s_n^*)$ *:*

$$u_i(s_1^*, ..., s_{i-1}^*, s_i^*, s_{i+1}^*, ..., s_n^*) \geq u_i(s_1^*, ..., s_{i-1}^*, s_i, s_{i+1}^*, ..., s_n^*), \text{for all } s_i \neq s_i^*$$

Thus for a strategy s_i^* to constitute a Nash equilibrium it has to be a best response strategy to agent i, given the best response strategies of the $n-1$ other agents. This implies strong assumptions on the level of information available to each agent, as every agent need information about all other agent's preferences.

Unlike both dominant strategy and Nash solution concepts, in a Bayesian game the strategy space is not defined in the strategic-form representation of the game. Instead, in a Bayesian game the strategy space is constructed from the type and action spaces, i.e. agent i's set of possible (pure) strategies, Σ_i, is the set of all possible functions with domain Θ_i and range A_i.

Definition 5 (Bayesian Strategy)**.** *In the Bayesian game* $G = \{N; A_1, ..., A_n; \Theta_1, ..., \Theta_n; p_1, ..., p_n; u_1, ..., u_n\}$*, a strategy for agent i is a function* $s_i(\theta_i)$*, where for each type* $\theta_i \in \Theta_i$*,* $s_i(\theta_i)$ *specifies the action from the feasible set* A_i *that type* θ_i *would chose if drawn by nature*

Given the definition of a strategy in a Bayesian game, we can define the Bayesian Nash equilibrium.

Definition 6 (Bayesian Nash Equilibrium)**.** *In the static Bayesian game* $G = \{N; A_1, ..., A_n; \Theta_1, ..., \Theta_n; p_1, ..., p_n; u_1, ..., u_n\}$*, the strategy profile* $s^* = (s_1^*, ..., s_n^*)$ *are a (pure-strategy) Bayesian Nash equilibrium if for each agent i and for each of i's types* $\theta_i \in \Theta_i$*,* $s_i^*(\theta_i)$ *solves*

$$\max_{a_i \in A_i} \sum_{\theta-i \in \Theta_{-i}} u_i(s_1^*(\theta_1), ..., s_{i-1}^*(\theta_{i-1}), a_i, s_{i+1}^*(\theta_{i+1}), ..., s_n^*(\theta_n); \theta) p_i(\theta_{-i}|\theta_i)$$

That is, no agent wants to change his or her strategy, even if the change involves only one action by one type

In other words each agent's strategy must be a best response to the other agens's strategies given distributional information. That is, a Bayesian Nash equilibrium is just a Nash equilibrium in a Bayesian game with incomplete information.

2.2.3 Comparison of Solution Concepts

The valuation or power of the solution concepts described above depends highly on the perspective in which you are concerned. From an economic point of view one would argue, that a Nash equilibrium is a stronger solution concept than a dominant strategy concept in the following sense. If the strategy profile $s^* = (s_1^*, ..., s_n^*)$ constitute a dominant strategy equilibrium, then s^* is the unique Nash equilibrium of the game. I other words, if you have a game with a dominant strategy equilibrium, then this equilibrium also is the unique Nash equilibrium of the game. But the reverse is not the case. If you have a unique Nash equilibrium of a game, then it is possible that there exists no dominant strategy equilibrium. This makes Nash a stronger solution concept than dominant strategy, seen from a perspective where the existence of a unique equilibrium is of greater concern than the assumptions about agent information about other agents.

Seen from another point of view, the dominant strategy solution concept is very robust, because it makes no assumptions about the information available to the agents about each other. This makes it a good candidate in computational mechanism design where tractability is a concern. Further a dominant strategy does not require an agent to believe that other agents will behave rationally when selecting its own optimal strategy. Recall the definition of a dominant strategy, a strategy is dominant if it is a best response regardless of the strategies of the other agents. As we will see later in the context of mechanism design, dominant strategy implementations of social choice functions are much more desirable than Nash implementations.

The Bayesian Nash solution concept is like a Nash solution concept except that it makes more reasonable assumptions about agent information. That is an agent i's strategy $s_i(\theta_i)$ must be at best response to the distribution over strategies of the other agents, given distributional information about preferences of the other agents. This is a big difference from the Nash solution concept where agent i's strategy is a best response to the strategies of the other agents. Still the Bayesian Nash solution concept is weaker than the dominant strategy concept, where agents optimal strategy does not depend on the strategies of the other agents. This brings us to the following preference ordering of solution concepts, when tractability is of concern.

$$\text{dominant strategy} \succeq \text{Bayesian Nash} \succeq \text{Nash}$$

Which I will pursue in the design of a mechanism in this project.

2.3 Mechanism Design

The field of mechanism design aims to implement a system-wide solution to a decentralized optimization problem. The goal is to implement a social choice solution within a system of self-interested agents with private information about their preferences seeking to maximize their own profit. It is the mechanism designer's job to design a mechanism that gives the agents a incentive to announce their true preferences as to find the system-wide solution or social choice.

In the following I will explain and define the key-properties of social choice functions and mechanisms. Note that the definitions are presented under the assumption that agents are risk-neutral and have quasi-linear preferences. For a more general definition of mechanisms see [5] and [3].

Definition 7 (Mechanism)**.** *A mechanism* $\mathcal{M} = (\Sigma_1, ..., \Sigma_n, k(\cdot), t_1(\cdot), ..., t_n(\cdot))$ *defines the set of strategies available to each agent, and an choice rule* $k : \Sigma_1 \times ... \times \Sigma_n \rightarrow \mathcal{K}$*, such that k(s) is the choice selected by the mechanism for the strategy profile* $s = (s_1, ..., s_n)$ *and a transfer rule* $t_i : \Sigma_1 \times ... \times \Sigma_n \rightarrow \mathbb{R}$*, one for each agent i*

In other words a mechanism defines the rules of the game, and implements the function to select the final outcome of the game. E.g. in a auction the mechanism defines the strategies available to the agents, bid at least the asked price, and selects the final outcome, the agent with the highest bid wins the auction.

Definition 8 (Social Choice Function)**.** *A social choice function* $f : \Theta_1 \times ... \times \Theta_n \rightarrow \mathcal{O}$ *chooses an outcome* $f(\theta) \in \mathcal{O}$*, given agent types* $\theta = (\theta_1, ..., \theta_n)$

Recall from section 2.1 that the utility to an agent i is denoted by $u_i = (o, \theta_i)$, where agent i's type θ_i determines its preferences over different outcomes. The goal of a mechanism design problem is defined by a social choice function, and given agent types $\theta = (\theta_1, ..., \theta_n)$ it selects the outcome $f(\theta)$. With quasi-linear agent preferences we can seperate the outcome of a social choice function into a choice $x(\theta) \in \mathcal{K}$ and a payment $p_i(\theta)$ made by each agent i :

$$f(\theta) = (x(\theta), p_1(\theta), ..., p_n(\theta))$$

for preferences $\theta = (\theta_1, ..., \theta_n)$.

Given a mechanism $\mathcal{M}$ with choice function $k(\cdot)$ we say that $\mathcal{M}$ implements a social choice function $f(\theta)$, if the choice selected by $\mathcal{M}$ with the equilibrium strategies of the agents is a solution to the social choice function $f(\theta)$ for all possible agent types.

Definition 9 (Mechanism implementation)**.** *Mechanism* $\mathcal{M} = (\Sigma_1, ..., \Sigma_n, k(\cdot), t_1(\cdot), ..., t_n(\cdot))$ *implements social choice function* $f(\theta)$ *if* $k(s^*) = f(\theta)$*, where the strategy profile* $s^* = (s_1^*(\theta_1), ..., s_n^*(\theta_n))$ *is a equilibrium solution to the game induced by* $\mathcal{M}$

At this point the solution concept is left out deliberately, but it could be any of those described in section 2.2.2. Recall that the dominant strategy solution concept is the most preferable because of its weak assumptions on an agent's information about other agents.

2.3.1 Properties of Social Choice Function

Many of the properties of a mechanism is derived from the social choice function, as the goal of a mechanism is defined by the social choice function.

Definition 10 (Pareto Optimal)**.** *A social choice function* $f(\theta)$ *is pareto optimal if for every outcome* $o' \neq f(\theta)$*, and all types* $\theta = (\theta_1, ..., \theta_n)$*,*

$$u_i(o', \theta_i) > u_i(o, \theta_i) \Rightarrow \exists j \in u_j(o', \theta_j) < u_j(o, \theta_j)$$

For a social choice function to be pareto optimal it must not be possible to make one agent happier without making another agent less happy.

Given that we are using quasi-linear preferences the properties of a social choice function can be separated for the choice selected and the payments.

Definition 11 (Allocative Efficiency)**.** *Social choice function $f(\theta) = (x(\theta), p(\theta))$ is allocative efficient if for all preferences $\theta = (\theta_1, ..., \theta_n)$,*

$$\sum_{i=1} v_i(x(\theta), \theta_i) \geq \sum_i v_i(x^{'}, \theta_i), \quad \textit{for all } x^{'} \in \mathcal{K}$$

For a social choice function to be allocative efficient it must maximize the total value over all agents.

Definition 12 (Budget Balancing)**.** *Social choice function $f(\theta) = (x(\theta), p(\theta))$ is budget balanced if for all preferences $\theta = (\theta_1, ..., \theta_n)$,*

$$\sum_{i=1} p_i(\theta) = 0$$

For a social choice function to be budget balanced the net transfer to and from the mechanism must be zero. And a weaker definitions is weakly budget balancing.

Definition 13 (Weakly Budget Balancing)**.** *Social choice function $f(\theta) = (x(\theta), p(\theta))$ is weakly budget balanced if for all preferences $\theta = (\theta_1, ..., \theta_n)$,*

$$\sum_{i=1} p_i(\theta) \geq 0$$

which states that for a social choice function to be weakly budget balanced the total payment to the mechanism must be positive. I.e. no net transfer from the mechanism to the agents.

It should be noted that if combining allocative efficiency and budget balance, we get pareto optimality which is a very strong property.

2.3.2 Properties of Mechanism

Not all properties of mechanisms are directly related to the social choice function implemented by the mechanism. The following properties are constraints on mechanism used to ensure the behaviour of agents.

Individual rationality also known as “voluntary participation” is a constraint on a mechanism allowing for the idea that an agent is not forced to participate in a mechanism, but can decide weather or not to participate.

Definition 14 (Individual Rationality)**.** *A mechanism $\mathcal{M}$ is (interim) individual-rational, if for all agent preferences θ_i it implements a social choice function $f(\theta)$ with*

$$u_i(f(\theta_i, \theta_{-i})) \geq \bar{u_i}(\theta_i)$$

where $u_i(f(\theta_i, \theta_{-i}))$ is the expected utility for agent i at the outcome, given distributional information about the preferences θ_{-i} of the other agents, and $\bar{u_i}(\theta_i)$ is the expected utility for a non-participation agent i

I other words, for a mechanism $\mathcal{M}$ to be individual-rational the expected utility received by agent i given prior beliefs about the preferences of the other agents, have to be greater than or equal to the expected utility agent i can get by not participating.

In a direct revelation mechanism the only strategies available to agents are to make direct claims about their preferences. Recall the broader definition of a strategy as a complete plan of action. These actions are in the case of direct revelation reduced to only reporting types.

Definition 15 (Direct Revealing)**.** *A direct-revelation mechanism $\mathcal{M} = (\Theta_1, ..., \Theta_n, k(\cdot), t_1(\cdot), ..., t_n(\cdot))$ restricts the strategy set $\Sigma_i = \Theta_i$ for all i, and has choice rule $k : \Theta_1 \times ... \times \Theta_n \rightarrow \mathcal{K}$ which selects an choice $k(\hat{\theta})$ based on reported preferences $\hat{\theta} = (\hat{\theta_1}, ..., \hat{\theta_n})$*

Note that an agent i is not restricted to report its true preferences, i.e. agent i could in a first-price sealed bid auction report a valuation that is lower that her true valuation if it gives higher expected utility. This can easily result in a non pareto optimal allocation. Imagine another agent j with a lower valuation than agent i reporting its true valuation, which is higher than the reported valuation of i. This results in agent j winning the auction, but agent i valuates the item higher that j, which is non pareto optimal, see 10 for a definition of pareto optimality.

Definition 16 (Truth Revealing)**.** *A strategy $s_i \in \Sigma_i$ is truth-revealing if $s_i(\theta_i) = \theta_i$ for all $\theta_i \in \Theta_i$*

A truth-revealing strategy is to report true information about preferences, for all possible preferences.

If we combine direct-revelation and truth-revelation we get what is called incentive compatibility, a constraint on mechanisms that require agents to prefer to act in accordance with the system-wide solution, i.e. to report their true preferences.

Definition 17 (Incentive Compatible)**.** *An direct-revelation mechanism $\mathcal{M}$ is a incentive-compatible mechanism if truth-revelation is an equilibrium solution to the game induced by the mechanism $\mathcal{M}$*

For a mechanism $\mathcal{M}$ to be incentive compatible agents are required to reveal their true preferences to the mechanism which have to constitute a equilibrium solution.

One particular important type of incentive-compatibility is strategy-proof or dominant-strategy incentive-compatibility.

Definition 18 (Strategy Proof)**.** *A direct-revelation mechanism $\mathcal{M}$ is strategy-proof if truth-revelation is a dominant-strategy equilibrium*

In other words, in a strategy-proof mechanism it is agent i's best strategy to report her true preferences regardless of the strategies of other agents. This makes strategy-proof mechanisms very interesting both from a game theoretic and computational perspective. The dominant strategy equilibrium is a very robust concept because of the few assumptions that are made on agent information and rationality. Computationally dominant strategy solutions have both good computation and communication properties. Agents can compute their strategies without modelling the strategies of the other agents, and they don't have to communicate with each other.

I will in this project pursue the design of a strategy-proof mechanism

2.3.3 Revelation Principle

An important theoretic tool in mechanism design is the *revelation principle* which states that under quite weak conditions any mechanism can be transformed into a equivalent incentive-compatible direct-revelation mechanism, see [5, sec. 2.3.2] for a proof. With the revelation principal in hand it is possible to prove what is possible and what is impossible in mechanism design. I will only state an important result of impossibility, conducted by Hurwicz [2], known as Hurwicz impossibility result

Theorem 1 (Hurwicz Impossibility Theorem)**.** *It is impossible to implement an efficient, budget-balanced, and strategy-proof mechanism in a simple exchange economy with quasi-linear preferences*

This is a important theorem to bear in mind when designing mechanisms. It actually says that it is not possible to design a mechanism that is both pareto optimal and strategy-proof. See [5] for a detailed description of possibility and impossibility of properties of mechanisms.

3 MiG Resource Allocation

In the following the resource allocation mechanism used in MiG will be described and analysed with respect to the economic model it implements. The description will be conducted by interpreting the source code of the MiG system. Based on the description I will use game theory to analyse the resource allocation used in MiG.

First I will give a short overview of the resource allocation mechanism of MiG in section 3.1. Followed by a detailed description in section 3.2. The analysis of the MiG mechanism in section 3.3 will consist of a proof that the current resource bidding strategy is not optimal. Followed by a deduction of the optimal bidding strategy.

3.1 Overview

The MiG resource allocation is using an online algorithm where resources request jobs from a grid server. The resources are volunteer and heterogeneous which imply that the grid server does not control their activities. To make resources obtain jobs, the mechanism gives resources an incentive to take jobs by paying them. In this stage it is not relevant if it is real or virtual currency. When a resource requests a job it gives the grid server information about how long it can work and for which minimum price it is willing to take a job. When users submit a job to the grid, they are giving information about how much they are willing to pay to get the job done. In this way the grid server receives a maximum job price from the user and a minimum job price from the resources. With these prices the grid server seeks to find the best suitable price for the user by analyzing all resources that it knows about. The best suitable resource is the one with the lowest price.

The MiG mechanism seeks to implement a market economy where users submit a price representing their maximum willingness to pay and resources submit their minimum supply price. The pricing strategy of resources is to let the price of their work follow demand. When demand is rising their price is rising and when demand is falling their price is falling. The pricing strategy of the users is not part of the model and is set exogenous. With these prices the grid server seeks to find the best suitable price for the user by analyzing all resources that it knows about. The best suitable resource is the one with the lowest price.

The goal of the MiG resource allocation mechanism is to implement fairness among both job submitters and resources as in a competitive market economy.

3.2 Description of MiG Mechanism

When a resource requests a job from a grid server, the server goes through the following steps :

Job Filtering Filtering jobs suitable for execution at resource.

Job Migration Migration of jobs offered a better price remotely

Job Scheduling Jobs updated by the job filtering procedure. are scheduled using one of more scheduling algorithms.

Price Updating The price of the requesting resource is updated according to the new demand.

I will in the following treat each of these steps in details.

3.2.1 Job Filtering

For each job in the job queue the best fitting resource is found by analyzing all resources known to the grid server. After the best resource is selected for a job, the job it is marked either to GO, STAY or MIGRATE. If a given job is suitable for the requesting resource it is marked to GO otherwise it is marked to STAY and another resource receives it. Finally it is marked to MIGRATE if a remote resource offers a better price. The following is a generalized version of the algorithm used for resource allocation, which is called upon a resource request

```
for each job in job-queue
    for each resource in resource-list
        best = min(best.price,resource.price)
    if best.dist > 0
        mark job to MIGRATE
    else if best != request
        mark job to STAY
    else
        mark job to GO
```

To find the best suitable resource the server divides the resources into local- and remote resources, where local resources are those connected directly to the given grid server and remote resources are those connected to other grid servers. Both local and remote resources are searched for the resource with the lowest price and distance, which is marked as the best resource. Distance is used to differentiate between local and remote resources, and in job migration to calculate migration penalty as described in section 3.2.2. The distance is 0 for local resources and varies for remote resources with a minimum of 1.

If the best suitable resources for a job, is not the requesting resource. There is collected a list of resources using the previous selected best resource to select all resources with a 5 % higher price. If the requesting resource is in the list it is selected as the best resource, otherwise the previous selected best resource prevails. This implies that the requesting resource receives a mark up on 5 % which is paid for by the user. This is not fairness because the user is suppose to have the job executed at the resource with the lowest price.

After the best resource is found there is calculated an expected delay which is a resource specific estimate stating when a resource can request a new job. If the requesting resource is the best resource then the expected delay is equal to the *cpu_ time*, otherwise expected delay is calculated as

$$expected\_delay = scheduling\_chance * cpu\_time$$

where *scheduling_ chance* is a estimate for how well the resource fits the job, i.e. if the requesting resource is the only resource included in the best resource list, the *scheduling_ chance* is equal to 1. Expected delay is used when calculating the price of a resource by imposing a delay penalty to the price the resource offers. Delay penalty is determined by

$$delay\_penalty = scale * (e^{expected\_delay * expire\_factor}) - 1$$

where *scale* is a global constant equal 0.00005 and *expire_factor* equals $16/conf.expire\_after$. Because of the low value of *scale* the delay penalty is only significant when expected delay is near Time To Live (TTL), which is defined as :

$$ttl = expire\_after - (now - queued\_timestamp)$$

The inclusion of *delay_ penalty* can from an economic viewpoint be seen as an ability for resources to make beforehand bids. I.e. bid on a job even before the resource is ready to receive to job. I will comment further on this subject later in chapter 4.

3.2.2 Job Migration

The MiG resource allocation mechanism uses a job migration strategy to implement price balancing. When resources are selected for jobs, remote resources are taken into account. And if a remote resource offers the cheapest price plus a migration penalty the job is migrated to the grid server where the remote resource is connected. In this way grid servers act as brokers buying and selling jobs.

Remote resources get a migration penalty when competing with local resources for jobs. The migration penalty is defined as

$$migration\_penalty = size + migration\_cost + thrashing\_cost$$

Where *size* is a constant equal 0.1 and *migration_ cost* is a server specific constant. Thrashing cost is a expression used to penalize already migrated jobs to avoid thrashing which is defined as

$$thrashing\_cost = 2 * (migration\_count + res\_dist)$$

where *migration_ count* is the number of times a job is migrated and *res_ dist* is the resources distance. the *thrashing_ cost* includes *migration_ count* which is a counter for the number of times the job has been migrated. Seen from an economic point of view the *trashing_ cost* is a historic cost which implies that the given resource is paying for something that has happened in the past. For the resource the payment of the *trashing_ cost* is sunk cost which is, without loss of generality an unrecoverable cost, see [8, section 20.6] for a detailed description. This suggest that the *trashing_ cost* could be discarded from the *migration_penalty*. Reversely it could be argued, from a network designer point of view that the *trashing_ cost* acts as congestion control, ensuring that migrating jobs are not congesting the grid server communication.

When remote and local resources are compared against prices, the *migration_penalty* of the remote resources is added to the price of the remote resource. In this way local resources are getting favored.

3.2.3 Job scheduling

After job filtering the jobs are scheduled for execution using one of the following scheduling algorithms.

First In First Out (FIFO) The server will examine the oldest job in the queue. If it is marked to GO the job will be popped and is executed. Otherwise the job is left on the queue and the requesting resource does not receive a job.

FirstFit The server will select the first job in the queue that is suitable for the given resource. I.e. the first job marked to GO

BestFit The server selects the jobs that fits the resource best, i.e. fills out most of the available resources on the resource. The selection is performed using a list of attributes which is prioritized from 0 to 100 as follows :

- **cpucount** Number of Central Processing Unit (CPU)'s at resource with priority 100.
- **nodecount** Number of Nodes at resource with priority 100.
- **cputime** The CPU time available at the resource with priority 60.

- **memory** The amount of memory available at the resource with priority 30.
- **disk** The amount of disk space available at the resource with priority 10.

FairFit This algorithm is an expansion of the BestFit algorithm that is modified to including job expire time in the fitness selection. In this way a potential problem of job starvation is compensated.

Random The server constructs a list including the jobs that is suitable for execution at the resource. A random job is selected from the list and is executed at the resource.

3.2.4 Price updating

In MiG the pricing strategy of resources is based on demand and when demand for the resource is rising the price is rising. Just like in a market economy. The price of a resource, *cur_price* is composed of a minimum price and a load multiplier. The minimum price (*min_price*) is configurable by each resource. The load multiplier (*load_multiply*) is a factor indicating the demand of the resource. Every time the resource requests a job its price is updated using the following equation:

$$cur\_price = min\_price * load\_multiply$$

By updating *load_multiply* each time the resource requests a job, *cur_price* can adapt to the current demand. If the resource gets a job, *load_multiply* is incremented by a factor called *multiplier_delta* which is a global constant. If the new price i above the given jobs maximum price, *load_multiply* is scaled down as to produce a *cur_price* that equals *max_price*. If the resource does not get a job its *load_multiply* is decremented by *multiplier_delta*. The price cannot get below the resource's minimum price.

While examining the source-code I found what seems to be an error. It is at the point where the *load_multiply* factor is decremented. The *load_multiply* is never decremented, either it stays unchanged or it is set to 1.0, see appendix E on page 66.

3.3 Analysis of MiG Mechanism

In the analysis of the MiG resource allocation mechanism I will prove that the current resource bidding strategy is not optimal for the resources. A strategy is optimal if it is a best response strategy to the solution concept in use. E.g. in a Bayesian Game an optimal strategy is a best response strategy given the believes about the strategies of the other agent's. Generally an optimal strategy is one that maximizes agent profit, i.e. it is an agent's best strategy to maximize her profit. As we will see later in chapter 4 profit maximization is a necessary condition for a allocation to economical fair.

I will use a reverse first-price sealed bid auction also known as an procurement auction to analyse the economic model used in MiG. Where bidders simultaneously submits their bids. The one with the lowest bid wins the auction to the lowest price. That is bidder i has valuation v_i for the good and bids b_i. If bidder i wins the auction, she receives payoff equal to $b_i - v_i$, while the other bidders receives 0 payoff. In case of a tie, the winner is determined by flipping a coin. Further the bidders are risk neutral and all of this i common knowledge.

To analyse a procurement auction with strategic agents as a game in a game theoretic perspective, I'll use a Bayesian Nash solution concept with incomplete information using only two agents, it will be straight forward to show the same for n agents.

3.3.1 Restrictions

When analyzing the current MiG resource allocation mechanism I will not include the following aspects.

- Will not include job migration. I will instead focus on a single grid-server and view it as a centralized system. The inclusion of job migration expands the model to include a market where the individual grid servers acts as brokers selling and buying jobs from each other. The modelling of the whole grid as market of jobs is a interesting topic and a potential future research topic. Actually this topic may extend to distributed mechanism design which is a open problem, see [6].

- Will not include collusion between resources. This is an interesting topic and a potential future research topic. I will in this project assume that resources are self interested and not capable of merging into collusions.

These restrictions is included to keep a general view and focus on the core mechanism instead of including functionalities not central to the mechanism.

3.3.2 A Bayesian game

In a Bayesian game with incomplete information agents knows the payoff function and strategy space of every other agent in the game, but the valuation of the good at auction is private information to each agent.

In order to formulate this problem as a Bayesian game, we must first identify the action space A, the type space Θ, the beliefs, and the payoff functions. Agent i's action is to submit a bid b_i, and her type is her valuation v_i. The formally definition of a two-player Bayesian game is $G =$

$\{A_1, A_2, \Theta_1, \Theta_2, p_1, p_2, u_1, u_2\}$. The action space of agent i is $A_i = [0, 1]$ and the type space is $\Theta_i = [0, 1]$. Because the valuations are independent (private value), agent i believes that v_j is uniformly distributed on $[0, 1]$, no matter the value of v_i. Finally equation (1) states the payoff u_i to an agent i given it's bid and valuation.

$$u_i(b_i, v_i) = \begin{cases} b_i - v_i & \text{if } b_i < b_j, \\ \frac{b_i - v_i}{2} & \text{if } b_i = b_j, \\ 0 & \text{if } b_i > b_j \end{cases} \quad (1)$$

If agent i's bid is the lowest it receives payoff equal to the surplus it can generate. If two agents both submit the lowest bid, a coin toss is used to decide the outcome. Finally agent i receives 0 payoff if another agent submit the lowest bid.

3.3.3 Current MiG bidding strategy

In the following I will deduce a mathematical expression for the bidding strategy of the current MiG resources and proof that the current MiG bidding strategy is not optimal. First I will proof that the strategy is not optimal for period 1, second I will prove it for period n.

From section 3.2.4 we have a description of the price-setting strategy of MiG resources. This strategy can be expressed mathematically as in equation (2).

$$b_i^{MiG}(t) = \begin{cases} b_{last} + \delta & \text{if } x_{last} = 1 \vee b_{last} + \delta > v_i, \\ b_{last} & \text{if } x_{last} = 1 \vee b_{last} + \delta > p_{max}, \\ b_{last} - \delta & \text{if } x_{last} = 0 \vee b_{last} - \delta > v_i, \\ v_i & \text{if } x_{last} = 0 \vee b_{last} - \delta < v_i, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Where b_{last} equals the last bid by agent i, x_{last} equals 1 if agent i won the last auction, and 0 otherwise. p_{max} is a reverse reservation price set by the job submitter, indicating the highest acceptable bid.

3.3.3.1 Proof on period 1

To prove that the MiG bidding strategy is not optimal in period 1 I introduce two agents, i and j and show that i's best strategy is not the MiG strategy.

In period 1 agents have no information of previous rounds and chooses therefore to bid their true valuation, which is common knowledge. Because agent i knows that agent j chooses to bid $b_j = v_j$, agent i can maximize her payoff by the following expression :

Table 1: agent utility

	$v_i < v_j$	$v_i > v_j$
$b_i < v_i$	$-$	$0 \wedge -$
$b_i = v_i$	0	0
$b_i > v_i$	$+ \wedge 0$	0

$$\max_{b_i} \quad (b_i - v_i)\text{Prob}\{b_i < v_j\} + \tfrac{1}{2}(b_i - v_i)\text{Prob}\{b_i = v_j\} \tag{3}$$

The maximization expression is not differential so it's not possible to deduce the optimal strategy by finding the first order condition. Instead I plot the payoff given by the different cases of the bid of agent i against v_i as shown in table 1 which is explained by figure 1 and 2.

Figure 1 shows the development in agent i's payoff when $v_i < v_j$. If agent i's bid is below her own valuation v_i it's clear that her payoff will be negative, and this would certainly be a unwise bid. When agent i's bid is above v_i but below v_j the payoff is positive. Finally when i's bid is above v_j the payoff is 0 because i won't win the auction.

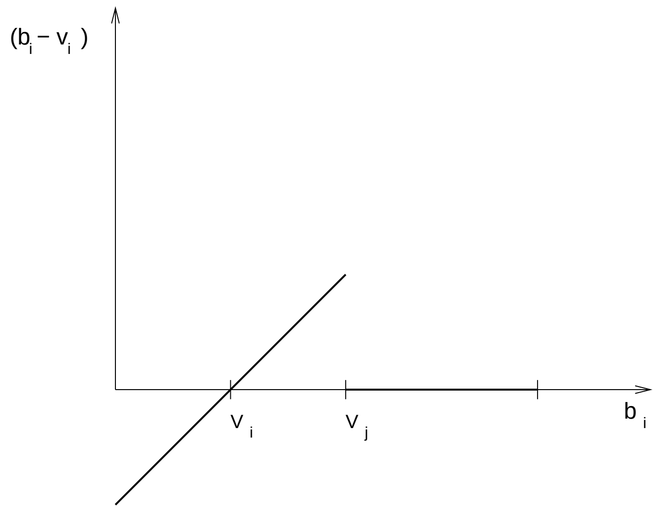

Figure 1: Payoff to agent i when $v_i < v_j$

Figure 2 shows the development in agent i's payoff when $v_i > v_j$. If agent i's bid is below her own valuation the payoff depends on the valuation of agent j. If b_i is below v_j and thereby b_j then agent i's payoff will be negative because she win the auction to a price that's below her valuation. If agent i's bid i above v_j, agent j wins the auction and i's payoff is 0.

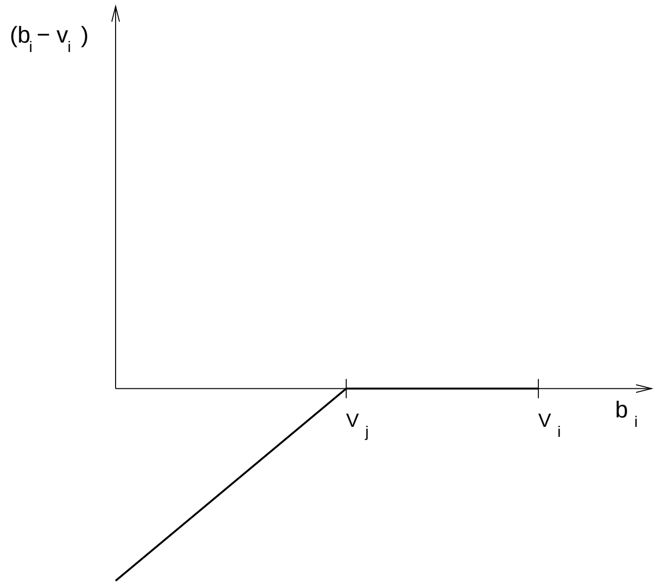

Figure 2: Payoff to agent i when $v_i > v_j$

There is only one strategy that can give positive payoff and that is in the case where agent i is bidding above her own valuation v_i and $v_i < v_j$, but agent i don't know v_j and if i bids above v_j she will gain 0 payoff and agent j will win the auction. Despite the chance of bidding above v_j, it's a better strategy for agent i to bid above her own valuation v_i because there is positive probability of winning. Consequently it's a better strategy for agent i to bid above her own valuation v_i than follow the MiG strategy in period 1. Following the MiG strategy and bidding v_i produces 0 payoff regardless of winning. While bidding above v_i implies positive probability of producing positive payoff.

3.3.3.2 Proof on period n

To prove that the MiG bidding strategy is not optimal in period n, I show the development in agents bidding strategies over time and deduce aggregated payoff. By introducing an agent with an improved strategy that gives higher payoff, I prove that the MiG strategy is not optimal.

The proof is divided into two parts. One where the bid increment, which is denoted δ is lower than the difference between the valuation of the agents, $\delta < (v_i - v_j)$, and one where the bid increment is higher than the valuation difference, $\delta > (v_i - v_j)$.

if $\delta < (v_j - v_i)$ By figure 3 the development in agent i's and j's bids are shown. It's important to notice that agent i wins in the first 2 periods because δ is lower than $v_i - v_j$. Further we see that agent i wins half the periods with positive payoff, while agent j wins the other half but with payoff equal to 0.

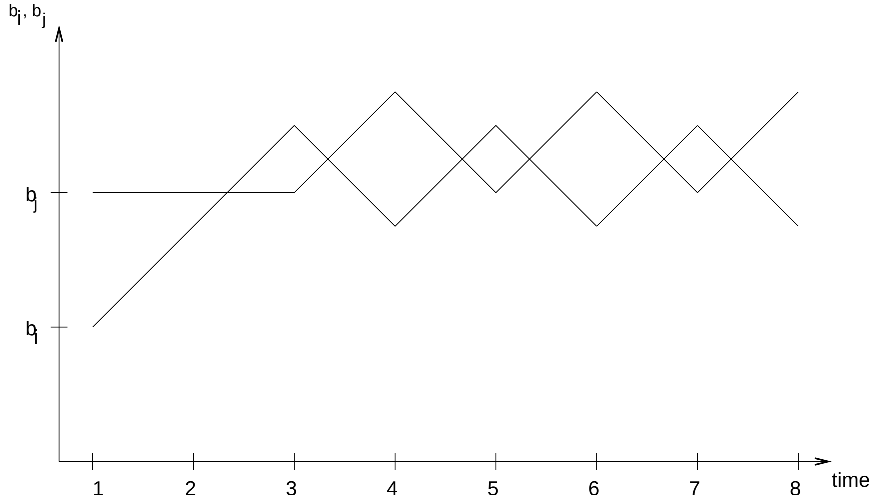

Figure 3: Payoff development over time when $\delta < (v_j - v_i)$

From the figure we can now deduce the aggregated payoff to an agent. From (1) we know that the payoff to agent i is equal to $b_i - v_i$ if agent i wins and otherwise 0. This gives the following aggregated payoff Π_i to agent i as a sum over the periods.

$$\begin{aligned}\Pi_i &= 0 + b_i - v_i + 0 + b_i - v_i + 0 + b_i - v_i ... \\ &= \begin{cases} \frac{t}{2} b_i - v_i & \text{if } t \text{ is even} \\ \frac{t-1}{2} b_i - v_i & \text{if } t \text{ is odd} \end{cases}\end{aligned}$$

if $\delta > (v_j - v_i)$ By figure 4 the development in agent i's and j's bids are shown. It's important to notice that δ in this case is greater than $v_i - v_j$.

The tendency is the same as of $\delta < (v_i - v_j)$, half the periods agent i wins and the other half agent j wins. Contrary to $\delta < (v_i - v_j)$ both agent i and j receives 0 payoff when winning because they win when they bid their valuation. This gives aggregated payoff Π equal to 0.

$$\Pi_i = 0$$

By gathering the two equations above we get the following expression for aggregated payoff to a MiG agent.

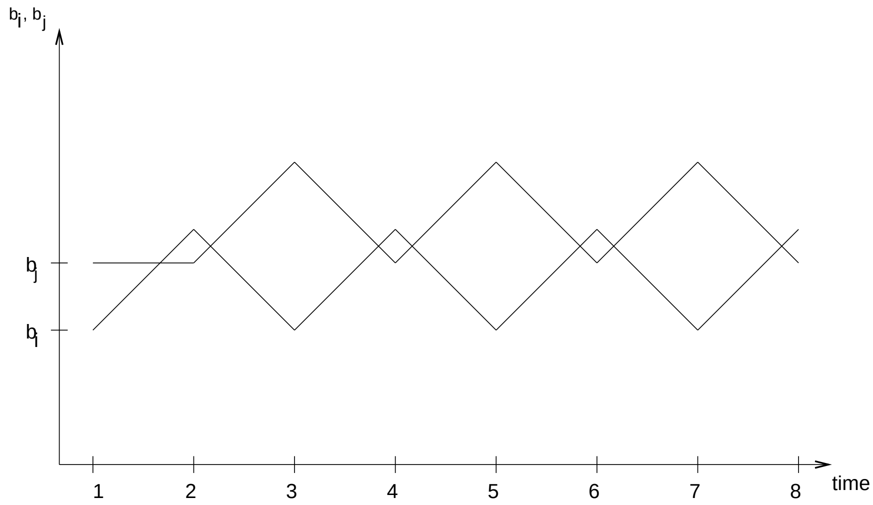

Figure 4: Payoff development over time when $\delta > (v_j - v_i)$

$$\Pi_i = \begin{cases} \frac{t}{2}\delta & \text{if } t \text{ is even } \vee v_i < v_j \\ \frac{t-1}{2}\delta & \text{if } t \text{ is odd } \vee v_i > v_j \\ 0 & \text{if } v_i \geq v_j \end{cases} \tag{4}$$

To show that this is not optimal, we need to introduce an agent with a improved strategy that produces a higher value of payoff. By looking at figure 3 and 4 we see that a agent only wins half the time, which implies for a strategy that improve on that.

One strategy that improves on the MiG strategy is for an agent i to increment it's bids until loosing and then lower the bids again with ϵ for $\epsilon < \delta$, until winning again as shown on figure 5.

Now agent i wins in $t-1$ periods with the following aggregated payoff

$$\Pi_i^* = (b_i - v_i - \epsilon)(t-1)$$

Which is greater than the payoff received by an agent playing the MiG strategy. This proofs that the MiG strategy is not optimal in period n. Still the improved strategy is not optimal because it does not constitute a Bayesian Nash equilibrium. A strategy have to be a best response strategy to the other agent's strategies to be optimal in a Bayesian Nash Game, see 6 for a definition of Bayesian Nash equilibrium. And the next step is to show which strategy is optimal under the conditions set by the MiG environment.

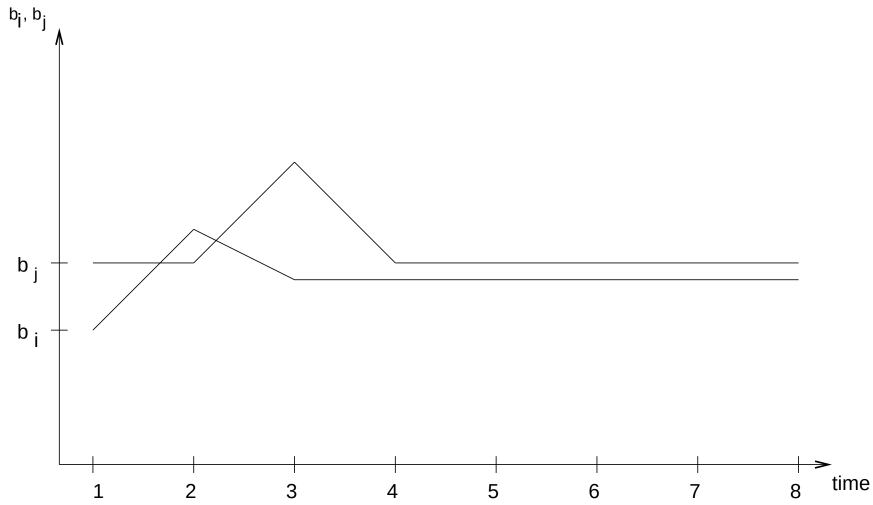

Figure 5: Payoff development for an improved strategy

3.3.4 Bayesian Nash Equilibrium

In the following section I will deduce the optimal bidding strategy and show that it is different from the current MiG strategy. First I will deduce it for a two-player game and afterwards show the N-player solution. Recall from section 3.3.2 the formal description of the game.

In a Bayesian game a strategy is a function from type to action, thus a strategy for agent i is a function $b_i(v_i)$ specifying the bid that each of i's types (i.e. valuations) would choose. In a Bayesian game, agent i's strategy $b_i(v_i)$ is a best response to agent j's strategy $b_j(v_j)$ and vice versa. Thus the Bayesian Nash equilibrium, if one exists, can be expressed by solving equation (5).

$$\max_{b_i} \quad (b_i - v_i)\text{Prob}\{b_i < b_j(v_j)\} + \tfrac{1}{2}(b_i - v_i)\text{Prob}\{b_i = b_j(v_j)\} \qquad (5)$$

To compute the equilibrium and thereby the agent's optimal strategies we must solve equation (5) by finding the first order condition. The calculus is left out and can be fund in [9] for a procurement auction, and in [1] for a first-price sealed bid auction. By solving the first order condition we get the following equilibrium bid strategy :

$$b_i^*(v_i) = \frac{1}{2}(v_i + 1) \qquad (6)$$

Which states that an agent will bid half her valuation plus a half which indicates her beliefs about the other agents bids. Recall that in a procurement auction an agent have to bid below her own valuation to gain positive profit, but how much depends on the number of participating agents, and the agent's beliefs about the strategies of the other agents. In the above equation it is assumed that the valuation of the other agents are uniformly distribution on $[0, 1]$. I.e. the valuation of the other agents takes a value between 0 and 1 with equal probability.

For a N-player game we get the following equilibrium bid strategy, again the calculus is left out and can be found in [9] :

$$b_i^*(v_i) = \frac{1}{n}(v_i + 1) \tag{7}$$

As the number of bidders approaches infinity each agent i's bid will approach her valuation v_i. That is $\lim_{n\to\infty} b_i(v_i) = v_i$. Thus when the number of agents participating in an auction i high, agents are bidding their true valuation which as we will see later is a nice property in mechanism design.

3.3.5 First-price Sealed Bid Auction

The resource allocation mechanism of MiG implements a procurement auction where resources are bidding for jobs and the one with the lowest bid receives the job to the lowest bidding price. A procurement auction is also known as a reverse first-price sealed bid auction and is similar to a normal first-price sealed bid auction where the agent with the highest bid receives the god to the highest price.

A first-price sealed bid auction have some drawbacks seen from both the agent and auctioneers perspective. A first-price sealed bid auction is not guaranteed to produce a pareto optimal output. The optimal bid for each of the agents depends on their beliefs about the valuation of the other agents. Imagine two agents i and j bidding for a god x where i values the god $v_i = 10$ and j values it $v_j = 7$. It is clear that for the allocation to be pareto optimal, agent i should receive the god. But i does not know the valuation of j and may think that j values the god 5 and then bid 6, resulting in j winning if bidding it's valuation v_j. Which is not pareto optimal because agent i values the god more than agent j. I should be noticed that if the agents is using the optimal biding strategy showed i section 3.3.4 the above allocation would produce a pareto optimal output.

Seen from a mechanism design point of view the first-price sealed bid auction have two further drawbacks. It is not truth revealing which means that agents does not reveal their true valuation to the mechanism. Second agents need distributed information about all other participating agents to calculate a optimal bidding strategy in a Bayesian Game. Thats why it is important in a computational setting to include tractability as in computational mechanism design, which will be discussed further in chapter 4.

3.3.6 Conclusion

From the above analysis we can conclude that the current MiG mechanism have the following drawbacks :

- Agents are not using the optimal bidding strategy which produces non-optimal output.
- The usage of a procurement auction encourage agents not to submit their true valuation, this can result in non pareto optimal output.
- In a procurement auction agents have no dominant strategy. To calculate their optimal strategy they need distributed information about payoff functions and strategy spaces of every other agent.

4 Designing a New Mechanism

When designing a new mechanism for the MiG resource allocation I will follow the directions determined in section 2.3 on page 9 and pursue the following properties :

- The new mechanism should be strategy-proof, because strategy-proof mechanisms produce both good game-theoretic properties and is tractable. From the dominant strategy property of a strategy-proof mechanism, the agent's strategy computation is simple, because agents can computer their own strategy without modeling the strategies of the other agents. Further strategy-proof mechanisms guarantees truth-revelation of agent preferences, which is a important property
- The mechanism should be allocative efficient which is needed to ensure economic fairness
- The mechanism should be at least weakly budget balanced to ensure that the mechanism is not running a deficit
- The mechanism should be individual-rational to ensure agent participation

In the following I will present a mechanism which conducts these properties. First I will state the delimits of my design in section 4.1. Followed by a motivation in section 4.2 for a new mechanism with a discussion of both economic fairness and choice of auction. In section 4.3 I will present a class of mechanisms with the desired properties and elaborate on the Vickrey auction. In section 4.4 I will present the new MiG mechanism. Followed by a description of the prototype implementation in section 4.5. Lastly there will be a test of the prototype in section 4.6.

4.1 Delimitation

I will in the design of a new resource allocation mechanism for MiG make the following assumptions :

Rationality I will assume that agents are rational, i.e. they are profit maximizing

Collusion I will assume that agents are self-interested, which implies that agents do not collude. In games with collusion, agents are not guaranteed to maximize their own profit, but instead maximize profit for the colluding group. I will restrict the design to only include self-interested agents

Private Value I will assume that agents have private valuation of the good in question, i.e the good may be worth different values to different agents. The opposite of private-value auctions is common-value auctions, where the value of the good is common knowledge.

Payment I will not consider payment schemes in the implementation of a prototype. I will just assume that there is some virtual currency in the system. The field of payment and virtual currency is large and incur a somewhat complex cryptographic setup ensuring nobody can cheat with the payments.

Security Issues I will assume that agents are not malicious and generally not consider any security issues in the design.

4.2 Motivation For a New Mechanism

In Chapter 3 I analysed the MiG mechanism and proved that it does not provide resources with a optimal strategy which as we will see, results in the mechanism not being economic fair.

4.2.1 Fairness

The current MiG resource allocation mechanism seeks to implement fairness for both buyers (job submitters) and sellers (resources) as in a competitive market economy.

Lets define economic fairness and compare it to the current MiG mechanism. For an economic allocation to be fair it has to be both equitable and pareto optimal. An allocation is equitable if no agent prefers any other agent's bundle of goods to his own, and an allocation is pareto optimal if it contains a competitive equilibrium. I.e. in a pareto optimal allocation you can not make any agent happier without making at least one agent worse off. See [8, chap. 31] for an elaboration on welfare.

In chapter 3 I proved that the current bidding strategy used by resources is not optimal. When their strategy is not optimal, they can not be profit maximizing which implies that they would gain more profit from another solution. First of all, the lose of profit maximization contradicts the assumption of agent rationality. Further the existence of a competitive equilibrium requires agents to use a profit maximizing strategy. Consequently the MiG mechanism can not be economic fair. Further the loss of profit by resources is gained as surplus by job submitters, i.e. the job submitters are favoured.

It is hard to conduct a global definition of fairness, because depending on the perspective there are some contradictions. I.e. in economic fairness, as described above there is no queue culture. Meaning that jobs bidding to low are starved. From a computational scheduling perspective, fairness can be defined as ensuring that starvation can not happen or at least minimizing the change of starvation. This can be done in many ways and on many levels, see section 3.2.3 for a description of the available scheduling algorithms in the current MiG mechanism. The fairfit scheduling algorithm of the current MiG mechanism implement fairness by compensating old jobs, i.e. the longer a job has been queued the larger a compensation it receives. It is evident that this kind of fairness contradicts economic fairness and will therefore not be included in my design.

4.2.2 Choice of Auction

By the way the current mechanism is implemented it is possible for resources to place beforehand bids on jobs. I.e. resources working on already scheduled jobs are bidding on new jobs, while receiving a delay penalty when competing with the requesting resource, see section 3.2.1. It is my assumption that the ability for resources to make beforehand bids might impose a unfair auction for the job submitters. Only the requesting resource and eventually resources without a job are placing a bid equal to their respective valuation. Resources working on an already scheduled job are imposed a delay penalty on their bids. This may have the same effect as just reducing the number of participating resources, resulting in the possibility for the remaining resources too raise their bids. This motivates for the design of a mechanism that do not allow beforehand bidding.

Based on statistics from the MiG team, it is known that most of the time there are more jobs queued at grid servers than requests from resources. This combined with the problem of beforehand bidding imposes for a design of a new auction. One solution is to shift from the current resource auction to a job submitter auction. In which job submitters are bidding on resources, causing a greater number of participants in an auction.

An alternative solution which I will not pursue in this project would be to implement a double auction in which both sellers (resources) and buyers (job submitters) submit bids which are then ranked highest to lowest. In this way it is possible to implement a supply and demand economy, where selling offers (sorted ascending) are matched with buying offers (sorted descending). This is a potential future research topic.

4.3 VCG Mechanism

In the following I will give a brief description of the Vickrey Clarke Groves (VCG) mechanism class, and argue for the choice of a Vickrey auction for the MiG resource allocation problem.

The VCG class of mechanisms are characterized by being the only mechanisms that are allocative efficient and strategy-proof for agents with quasi-linear preferences, amongst direct-revelation mechanisms. In special cases VCG mechanisms are also weakly budget balanced and satisfies individual-rationality which is the case for a Vickrey auction. In [5] Parkes proves that VCG mechanisms are strategy-proof.

A special case of the VCG mechanism, is the Vickrey auction also known as the second-price sealed bid auction. Formally defined by William S. Vickrey, who received the Nobel prize in 1996 partly for the contribution of the Vickrey auction. In a Vickrey auction an agent i with valuation v_i and bid b_i wins the auction if b_i is the highest bid. If agent j is the agent with the second highest bid, then agent i is paying b_j for the item won, and agent j is paying nothing. The payoff u_i received by agent i equals $u_i = v_i - b_j$, and note that the payment incurred by agent i do not depend on b_i which is a general property of the VCG mechanisms.

Generally the expected payoff received by agent i can be defined as

$$u_i(v_i, b_i, b_j) = \begin{cases} v_i - b_j & \text{if } b_i > b_j, \\ 0 & \text{if } b_i \leq b_j \end{cases}$$

If agent i wins, she receives payoff equal to the difference between her valuation and the bid of agent j, assumed that j is the agent with the second highest bid. In case of a tie, a coin toss is used to select the outcome, but note that the payoff to agent i still is zero if she wins the coin toss. If $b_i = b_j$ then the second highest bid is equal to agent i's valuation v_i and then she receives zero payoff. Also if agent i's bid is not the highest, i receives zero payoff.

In a Vickrey auction the optimal strategy is for agent i is always to bid her true valuation, as

$$b_i(v_i) = v_i$$

Compared with the optimal bidding strategy of a first-price sealed bid auction, derived in section 3.3.4 on page 25, the optimal bidding strategy in a Vickrey auction is fairly simple. Note that the computation of the optimal strategy for agent i in a Vickrey auction does not depend on the strategies of the other agents, nor the beliefs about their strategies as in a first-price sealed bid auction. I will now conduct the proof of both the above strategy being the optimal strategy.

Optimal Vickrey strategy. Lets look at the special case where we have two bidders agent i and j with valuations v_i and v_j and bids b_i and b_j. if $v_i > b_j$ then it is in agent i's best interest to win the auction, which she can do by setting $b_i = v_i$. Else if $v_i < b_j$ then it is in agent i's best interest not to win the auction, which she can do by setting $b_i = v_i$. In either case it is agent i's best strategy to bid her true valuation. □

This also proofs that agents in a Vickrey auction plays dominant strategies because the optimal strategy only depends on agent i's own true valuation, and this strategy is a best response no matter the strategies played by the other agents, see definition 3 of a dominant strategy.

To see that a Vickrey auction is allocative efficient, notice that the maximized valuation over all agents equals the valuation of the winning agent's valuation. As in the above allocation where agent i wins, the maximized valuation over all agents is v_i.

Further the allocation is weakly budget balanced. The payment of agent i equals the second highest bid b_j which is non-negative and all other agent pays nothing.

Recall that for a mechanism to be individual-rational, the mechanism must always impose at least as much payoff to an agent as the agent can get by not participating. To see that the above allocation satisfy individual-rationality, observe that the second highest bid b_j is no greater than the highest bid b_i, which is equal to the winning agent's equilibrium valuation.

As can be seen from the above description of the Vickrey auction, it satisfy all the property conditions outlined in the introduction to this chapter. For this reason I will use the Vickrey auction to implement a new resource allocation mechanism to MiG.

4.4 The MiG Mechanism

As stated above I will implement the new MiG mechanism as a Vickrey auction. The auction will be reverted in proportion to the current MiG mechanism as described in section 4.2.2. The participants in the auction will be the job submitters, which will be denoted agents. The agents will be competing for available resources, in the sense that every time a resource requests a new job, there will be held a auction between the jobs queued at a given grid server. The winning job will be scheduled to execute at the requesting resource.

In the following I will give a formal definition of the mechanism including definition of agent utility functions and social choice function.

4.4.1 Defining agent utility

It is a requirement that agent's preferences are quasi-linear as described i section 2.1 where utility are defined as a combination of a valuation function and a payment as :

$$u_i(o, \theta_i) = v_i(x, \theta_i) - p_i$$

The type $\theta_i \in \Theta_i$ is in a Vickrey auction equal to agent i's valuation of the auctioned resource. The valuation function $v_i(\cdot)$ can be any function, but I will assume it is continuous and differential to ensure the possibility of profit maximization. In the prototype implementation, I will define the valuation function as :

$$v_i(x, \theta_i) = \theta_i$$

Though the valuation function is just a constant equal to agent i's valuation. This seems quit natural when compared with the current MiG mechanism where a job submitter's valuation is best defined as the job submitter's maximum willingness to pay value. It would be straight forward to change the valuation function in the future, and the mechanism will be unaffected.

It should be noted that in a strategy-proof mechanism like the Vickrey auction truth-revelation is a dominant strategy. I.e. an agents true valuation is the value θ_i that maximizes agent i's profit as in

$$\max_{\theta} \quad u_i(o, \theta_i)$$

Given agent i's type $\theta_i \in \Theta_i$ and outcome $o \in \mathcal{O}$ where an outcome is an allocation. In other words agent i will always chose to announce the valuation that maximizes her profit. This ensures that the job submitters are profit maximizing which is important for the mechanism to produce a efficient allocation.

4.4.2 Defining the mechanism

Given that the new mechanism will be implemented as a Vickrey auction we simplify the definition of the mechanism. From section 2.3 we formally defined a mechanism $\mathcal{M}$ as :

$$\mathcal{M} = (\Sigma_1, ..., \Sigma_n, k(\cdot), t_1(\cdot), ..., t_n(\cdot))$$

Selecting a choice defined by the choice rule $k(s^*)$ given the strategy profile $s^* = (s_1^*(\theta_1), ..., s_n^*(\theta_1))$. In a strategy-proof mechanism like the Vickrey auction, the only strategy available to an agent i is to submit her true valuation θ_i.This reduces the strategy set Σ_i available to agent i to Θ_i, and a mechanism $\mathcal{M}$ can be defined as :

$$\mathcal{M} = (\Theta_1, ..., \Theta_n, k(\theta), t_1(\theta), ..., t_n(\theta))$$

Which reduces the strategy profile s^* to $\theta = (\theta_1, ..., \theta_1)$ such that the choice rule can be redefined as $k : \Theta_1 \times ... \times \Theta_n \rightarrow \mathcal{O}$.

For this mechanism to implement the proposed properties of allocative efficiency and weakly budget balance, the belonging social choice function must implement these properties, and the choice implemented by the choice rule $k(\theta)$ must equal the choice chosen by the social choice function $f(\theta)$.

I will by implementing a prototype verify that the mechanism I have designed implements the properties defined for the social choice function. That is, I will by testing the prototype implementation verify that the selected outcome is both allocative efficient and weakly budget balancing.

4.5 Prototype Implementation

The prototype of the mechanism will be implemented in python which is a requirement from the MiG team.

The process will be as follows :

1. A resource request a job
2. Suitable jobs are selected by the grid server to participate in the auction
3. Each job place a bid
4. Bids are sorted and the highest and second highest are extracted
5. The job with the highest bid are scheduled to the resource and charged a price equal the second highest bid

A request should be a tuple of *(resource_id, execution_time, reservation_price)*, where *resource_id* is a unique identifier of a resource, *execution_time* specifies the amount of time for which the resource is available for work, in seconds. The *reservation_price* specifies the minimum price for which the resource is willing to work.

The job request from the resource is what triggers the auction, and before the auction can start, the grid server must filter off jobs which have a *execution_time* larger than the *execution_time* of the requesting resource. The *execution_time* of a job is estimated by the job submitter and specified when the job is submitted.

A bid should be a tuple of *(bid, job_id)*, where the *bid* specifies the job's bid on the requesting resource, and *job_id* is a unique identifier of a job. It should be noted that, because it is a sealed bid auction, the order of the bids can be arbitrary. The bids are collected in a list of *(bid, job_id)* tuples. The list is sorted after all participating jobs have places a bid. In python a list of tuples is sorted ascending, starting with the first element of the tuples, which are the bids.

After the list of bids is sorted, we can extract the highest and second highest bids just by popping the list twice. The winner is determined by extracting the *job_id* of the highest bid tuple and the payment determined by extracting the *bid* of the second highest bid tuple. In case of a tie, the equally highest bids are collected in a list and a random one is picked as winner. This is similar to the case of only two agents where we use a coin toss to determine the winner.

The source-code for the mechanism is available in appendix B on page 42.

4.6 Testing

I will test the prototype implementation by usage of an simulation of the MiG system. The simulation is implemented by Jonas Bardino, a member of the

MiG team. The simulation is simulating the scheduling of jobs to resources by running a discrete number of steps. Where users are submitting jobs and resources requesting jobs in each step. The simulation configuration used in the test is enclosed in appendix C on page 44.

I will verify the correctness of the implemented mechanism by including the following properties in a test strategy:

- Filtering off jobs with too high execution time
- Obey reservation prices
- Handling ties
- Allocative efficiency
- Weakly budget balanced

Each of these properties will be verified by logging different stages of the mechanism.

Filtering off jobs with too high execution time To verify that jobs with a too high execution time are filtered off, I log the execution time of jobs with an execution time larger than the execution time of the requesting resource.

```
13:58:56,315 : Job user-1-4 is filtered off 9 > 8
13:58:56,315 : Job user-0-5 is filtered off 10 > 8
13:58:56,316 : Job user-0-10 is filtered off 10 > 8
```

In this extract of an auction we see that three jobs are filtered off because their execution time are above the execution time of the resource. The extract is line 24-26 from the simulation output enclosed in appendix D.1 on page 45

Obey reservation prices To verify that reservation prices are obeyed, I log bids that are lower than the reservation price of the requesting resource.

```
15:28:32,245 : Bid from user-1-1 lower than reservation price,
56 < 57
15:28:32,245 : Agent user-0-2 bids 57 on resource resource-0
with reservation price 57
15:28:32,246 : Agent user-0-4 bids 57 on resource resource-0
with reservation price 57
15:28:32,246 : Highest bid 57 by user-0-4, second highest 57
by user-0-4
15:28:32,246 : Winner : agent user-0-4 bidding 57 and
pays 57
```

In this extract of an auction we see a job with a bid that is lower that the reservation price of the requesting resource. Consequently the bid is not included in the auction. The extract is line 5-9 from the simulation output enclosed in appendix D.2 on page 57.

Handling ties To verify that the ties are handled correct, I both log the number of agents with an equal highest bid, all bids, and the winner of the auction. In this way I can verify that the winning agent is one of the agents bidding equally highest.

```
13:58:56,334 : Agent user-1-11 bids 56 on resource resource-0
with reservation price 0
13:58:56,334 : Agent user-1-12 bids 56 on resource resource-0
with reservation price 0
13:58:56,335 : Agent user-1-20 bids 56 on resource resource-0
with reservation price 0
13:58:56,335 : Highest bid 56 by user-1-20, second highest 56
by user-1-12
13:58:56,335 : Got a tie with 3 agents
13:58:56,336 : Winner : agent user-1-11 bidding 56 and
pays 56
```

In this extract of a auction we see a tie where three jobs have places an equal highest bid of 56. The tie is resolved by choosing one of the jobs randomly as winner. The extract is line 66-71 from the simulation output enclosed in appendix D.1 on page 45.

Allocative efficiency To verify that the outcome of the auction is allocative efficient, I have verify that the winning agent is the one with the highest bid.

```
13:58:56,413 : Agent user-1-26 bids 56 on resource resource-0
with reservation price 0
13:58:56,413 : Agent user-1-29 bids 56 on resource resource-0
with reservation price 0
13:58:56,414 : Agent user-0-42 bids 57 on resource resource-0
with reservation price 0
13:58:56,414 : Highest bid 57 by user-0-42, second highest 56
by user-1-29
13:58:56,414 : Winner : agent user-0-42 bidding 57 and
pays 56
```

In this extract of a auction we see three jobs placing bids on the requesting resource. Job *user-0-42* has the highest bid and both of the two others has the second highest bid. Recall that an agent's valuation is only positive if winning and otherwise it is zero, given the outcome of the auction. This implies that the accumulated valuation for the three agents equals the valuation of the winning agent, which is equal to 57. This is exactly the allocative efficient outcome of the auction. The extract is line 316-320 from the simulation output enclosed in appendix D.1 on page 45.

Weakly budget balanced To verify that the outcome is weakly budget balanced, I just verify that the winning agent pays a positive amount.

```
13:58:56,378 : Highest bid 57 by user-0-28, second highest 56
by user-1-33
13:58:56,379 : Winner : agent user-0-28 bidding 57 and
pays 56
```

In this extract of a auction we see that the winning agent pays an amount equal to the second highest bid. Recall that an auction is weakly budget balanced if the net transfer to the mechanism is zero or positive. The only job paying in the above auction is the winning job which is paying 56. I.e. the auction is weakly budget balanced. This extract is line 206-207 from simulation output enclosed in appendix D.1 on page 45.

Overall the test verifies that the implemented mechanism is correct with respect to the social choice function it implements and otherwise the properties listed above.

5 Future Work

The following is a list of proposal for future work:

- analyse current MiG mechanism as an iterative Baysian Nash game
- mechanism utilizing makespan minimization
- investigate implications of collusion between agents
- analyse mechanisms implementing a double auction
- analyse iterative mechanisms allowing agent strategies to include history
- analyse implication of private- vs. common-value assumption
- inclusion of job migration between grid servers which imposes a market between grid servers, acting as brokers buying and selling jobs.

6 Conclusion

From the analysis in section 3.3 I have proven that the current MiG resource allocation mechanism is not optimal for the participating resources. By modelling the MiG mechanism as a procurement auction, using a static Bayesian game. I have first proved that the current pricing strategy of resources is not the best one by imposing an agent with a better strategy. Afterwards I have deduced the optimal strategy for an agent in a first price sealed bid auction.

From the proof of the pricing strategy of resources being non optimal I have proclaimed that the MiG mechanism is not implementing economic fairness. If a resource is not using an optimal strategy it can not be profit maximizing, which implies that the allocation can not be pareto optimal. For an allocation to be economical fair, it must be both pareto optimal and equitable which is not the case for the current MiG mechanism.

I have based on the description and analysis of the current MiG mechanism designed a new mechanism. Constrained myself to consider tractability in the design of a new mechanism, because the mechanism is implemented in a computational application. The goal of the mechanism is to produce a higher degree of economic fairness while being tractable.

By implementing the mechanism as a Vickrey auction which is from the VCG class of mechanisms. I preserve good tractability while providing a strategy-proof mechanism that is both allocative efficient, weakly budget balanced, and individual rational.

I have implemented a prototype of the mechanism and verified that it is both allocative efficient and weakly budget balanced using a simulation of the MiG system. Suggesting that it produces economic fairness.

Bibliography

[1] *Game Theory for Applied Economists.* N/A, Robert Gibbons.

[2] Leonid Hurwicz. On the existence of allocation systems whose manipulative nash equilibria are pareto optimal. 1975.

[3] Vijay Krishna. *Auction Theory.* Academic Press, 2002.

[4] Roger B. Myerson. *Game Theory Analysis of Conflict.* Harvard University Press, 1997.

[5] David C. Parkes. *Classic Mechanism Design.* Ph.d.-afhandling, Univesity of Pennsylvania, may 2001.

[6] Joan Feigenbaum & Scott Shenken. Distributed algorithmic mechanism design: Recent results and future directions. Yale University and ICIs Berkeley, 2002.

[7] Drew Fudenberg & Jean Tirole. *Game Theory.* The MIT Press, 1993.

[8] Hal R. Varian. *Intermediate Microeconomics, A Modern Approach.* Norton, fifth edition udgave, 1999.

[9] Mihkel Tombak & Wei Wang. Multistage procurement auction. Queen's School of Business, Kingston, Canada, April 2003.

A Acronyms

MiG Minimum Intrusion Grid

FIFO First In First Out

TTL Time To Live

VCG Vickrey Clarke Groves

CPU Central Processing Unit

B Source-Code

```
import random
import logging

import JobQueue
import Scheduler

class VickreyAuction(Scheduler.Scheduler):

    filtered_jobs = None
    sealed_bids = None

    def __init__(self, logger, config):
        Scheduler.Scheduler.__init__(self, logger, config)
        self.name = 'VickreyAuction'
        self.filtered_jobs = []
        self.sealed_bids = []

        logging.basicConfig(level=logging.DEBUG,
                            format='%(asctime)s,%(msecs)d : %(message)s',
                            datefmt='%H:%M:%S',
                            filename='vickrey.log',
                            filemode='w')
    #end __init__

    def Schedule(self, resource_conf):
        return self.HoldAuction(resource_conf)

    def FilterJobs(self, resource_conf):
        """Filter jobs with to long execution time"""

        res_time = resource_conf["CPUTIME"]
        qlen = self.job_queue.QueueLength()

        self.sealed_bids = []
            self.filtered_jobs = []

        for i in range(qlen) :
            job = self.job_queue.GetJob(i)

            if job["CPUTIME"] <= res_time :
                self.filtered_jobs.append(job)
            else :
                    logging.info("Job %s is filtered off %d > %d", job["
                        JOB_ID"], job["CPUTIME"], res_time)

    def HoldAuction(self, resource_conf):
        """Executes a Vickrey auction"""

        #filter jobs
        self.FilterJobs(resource_conf)

        #Receives sealed bids from jobs
        for job in self.filtered_jobs:
            #only inclues bids equal to or above the reservation price
            #MAXPRICE =  bid, MINPRICE = reservation_price
            max = self.GetMaxPrice(job)
                min = self.GetMinPrice(resource_conf, job["RUNTIMEENVIRONMENT
                    "])

            if(max >= min) :
                self.sealed_bids.append((max, job["JOB_ID"]))
                logging.info("Agent %s bids %d on resource %s with 
                    reservation price %d",
                            job["JOB_ID"], max, resource_conf["RESOURCE_ID"
                                ], min)
            else :
                logging.info("Bid from %s lower than reservation price, %d < 
                    %d",
                            job["JOB_ID"], max, min)
        #sorting bids
```

```
        self.sealed_bids.sort()

        #Extract the highest and second highest bids
            if len(self.sealed_bids) <= 0 :
                return None

            highest = self.sealed_bids.pop()

        if len(self.sealed_bids) > 1 :
            second = self.sealed_bids.pop()
        else :
           second = highest

            logging.info("Highest bid %d by %s, second highest %d by %s",
                highest[0], highest[1], second[0], second[1])

        winner = dict()

        #determin outcome
        if (highest[0] == second[0] and highest[1] != second[1]) :
            #toss a coin to select a winner
            equal_bids = []
                equal_bids.append(highest)
                equal_bids.append(second)

                self.sealed_bids.reverse()

                #collections all bids equal to the highest
                for bid in self.sealed_bids :
                    if bid[0] == highest[0] :
                            equal_bids.append(bid)
                    else :
                            break

        logging.info("Got a tie with %d agents", len(equal_bids))

            #both inclusive
            winning_bid = equal_bids.pop(random.randint(0,len(equal_bids)-1))
            winner = self.FindWinner(winning_bid[1])

        else :
            winner = self.FindWinner(highest[1])

        #payment and scheduling
            winner["EXEC_PRICE"] = float(second[0])
        #burde måske være max - min
            winner["EXEC_RAWDIFF"] = float(highest[0]) - float(second[0])
            winner["EXEC_RESOURCE"] = resource_conf["RESOURCE_ID"]

            logging.info("Winner : agent %s bidding %d and payes %d", winner[
                "JOB_ID"], highest[0], second[0])

            #schedule winning job til requesting resource
            self.job_queue.DequeueJobByJobId(winner["JOB_ID"])
        self.UpdateHistory(winner, resource_conf)

        return winner

    def FindWinner(self, job_id):
        """Finds the winning job, given a job_id"""
        qlen = self.job_queue.QueueLength()

        for i in range (qlen) :
            job = self.job_queue.GetJob(i)

            if job["JOB_ID"] == job_id :
                return job

            raise Exception("winner %s not found in queue!" % job_id)
```

C Simulation Configuration

C.1 Basic 1-2

One resource and two users

```
[DEFAULT]
timesteps = 40
topology = linear

[server-0]
type = server

[resource-0]
type = resource
server = server-0
minprice = 0.0
request_probability = 0.8

[user-0]
type = user
server = server-0
maxprice = 57.0
submit_probability = 0.3

[user-1]
type = user
server = server-0
maxprice = 56.0
submit_probability = 0.3
```

C.2 Reservation Price 1-2

One resource and two users and reservation price of resource is higher than maximum price of one of the users.

```
[DEFAULT]
timesteps = 40
topology = linear

[server-0]
type = server

[resource-0]
type = resource
server = server-0
minprice = 57.0
request_probability = 0.8

[user-0]
type = user
server = server-0
maxprice = 57.0
submit_probability = 0.3

[user-1]
type = user
server = server-0
maxprice = 56.0
submit_probability = 0.3
```

D Test-Data

D.1 Correctness Test

```
13:58:56,299 : Agent user-0-0 bids 57 on resource resource-0 with reservation price 0
13:58:56,300 : Highest bid 57 by user-0-0, second highest 57 by user-0-0
13:58:56,300 : Winner : agent user-0-0 bidding 57 and payes 57
13:58:56,303 : Agent user-1-1 bids 56 on resource resource-0 with reservation price 0
13:58:56,303 : Highest bid 56 by user-1-1, second highest 56 by user-1-1
13:58:56,303 : Winner : agent user-1-1 bidding 56 and payes 56
13:58:56,309 : Job user-0-3 is filtered off 8 > 1
13:58:56,309 : Job user-1-4 is filtered off 9 > 1
13:58:56,309 : Job user-0-5 is filtered off 10 > 1
13:58:56,309 : Job user-0-6 is filtered off 2 > 1
13:58:56,310 : Job user-0-7 is filtered off 6 > 1
13:58:56,310 : Job user-0-8 is filtered off 6 > 1
13:58:56,310 : Agent user-1-2 bids 56 on resource resource-0 with reservation price 0
13:58:56,310 : Highest bid 56 by user-1-2, second highest 56 by user-1-2
13:58:56,310 : Winner : agent user-1-2 bidding 56 and payes 56
13:58:56,312 : Job user-0-3 is filtered off 8 > 4
13:58:56,312 : Job user-1-4 is filtered off 9 > 4
13:58:56,313 : Job user-0-5 is filtered off 10 > 4
13:58:56,313 : Job user-0-7 is filtered off 6 > 4
13:58:56,313 : Job user-0-8 is filtered off 6 > 4
13:58:56,313 : Agent user-0-6 bids 57 on resource resource-0 with reservation price 0
13:58:56,313 : Highest bid 57 by user-0-6, second highest 57 by user-0-6
13:58:56,314 : Winner : agent user-0-6 bidding 57 and payes 57
13:58:56,315 : Job user-1-4 is filtered off 9 > 8
13:58:56,315 : Job user-0-5 is filtered off 10 > 8
13:58:56,316 : Job user-0-10 is filtered off 10 > 8
13:58:56,316 : Agent user-0-3 bids 57 on resource resource-0 with reservation price 0
13:58:56,316 : Agent user-0-7 bids 57 on resource resource-0 with reservation price 0
13:58:56,317 : Agent user-0-8 bids 57 on resource resource-0 with reservation price 0
13:58:56,317 : Agent user-0-9 bids 57 on resource resource-0 with reservation price 0
13:58:56,317 : Highest bid 57 by user-0-9, second highest 57 by user-0-8
13:58:56,317 : Got a tie with 4 agents
13:58:56,318 : Winner : agent user-0-9 bidding 57 and payes 57
13:58:56,322 : Agent user-0-3 bids 57 on resource resource-0 with reservation price 0
13:58:56,323 : Agent user-1-4 bids 56 on resource resource-0 with reservation price 0
13:58:56,323 : Agent user-0-5 bids 57 on resource resource-0 with reservation price 0
13:58:56,323 : Agent user-0-7 bids 57 on resource resource-0 with reservation price 0
13:58:56,324 : Agent user-0-8 bids 57 on resource resource-0 with reservation price 0
13:58:56,324 : Agent user-0-10 bids 57 on resource resource-0 with reservation price 0
13:58:56,324 : Agent user-1-11 bids 56 on resource resource-0 with reservation price 0
13:58:56,324 : Agent user-1-12 bids 56 on resource resource-0 with reservation price 0
13:58:56,325 : Agent user-0-13 bids 57 on resource resource-0 with reservation price 0
13:58:56,325 : Agent user-0-14 bids 57 on resource resource-0 with reservation price 0
13:58:56,325 : Agent user-1-15 bids 56 on resource resource-0 with reservation price 0
13:58:56,326 : Agent user-0-16 bids 57 on resource resource-0 with reservation price 0
```

```
13:58:56,326 : Agent user-0-17 bids 57 on resource resource-0 with reservation price 0
13:58:56,326 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,326 : Got a tie with 9 agents
13:58:56,327 : Winner : agent user-0-10 bidding 57 and payes 57
13:58:56,331 : Job user-0-3 is filtered off 8 > 4
13:58:56,331 : Job user-1-4 is filtered off 9 > 4
13:58:56,331 : Job user-0-5 is filtered off 10 > 4
13:58:56,332 : Job user-0-7 is filtered off 6 > 4
13:58:56,332 : Job user-0-8 is filtered off 6 > 4
13:58:56,332 : Job user-0-13 is filtered off 8 > 4
13:58:56,332 : Job user-0-14 is filtered off 10 > 4
13:58:56,332 : Job user-1-15 is filtered off 7 > 4
13:58:56,332 : Job user-0-16 is filtered off 5 > 4
13:58:56,333 : Job user-0-17 is filtered off 10 > 4
13:58:56,333 : Job user-1-18 is filtered off 10 > 4
13:58:56,333 : Job user-0-19 is filtered off 7 > 4
13:58:56,333 : Job user-0-21 is filtered off 9 > 4
13:58:56,333 : Job user-1-22 is filtered off 6 > 4
13:58:56,334 : Job user-1-23 is filtered off 5 > 4
13:58:56,334 : Job user-1-24 is filtered off 7 > 4
13:58:56,334 : Agent user-1-11 bids 56 on resource resource-0 with reservation price 0
13:58:56,334 : Agent user-1-12 bids 56 on resource resource-0 with reservation price 0
13:58:56,335 : Agent user-1-20 bids 56 on resource resource-0 with reservation price 0
13:58:56,335 : Highest bid 56 by user-1-20, second highest 56 by user-1-12
13:58:56,335 : Got a tie with 3 agents
13:58:56,336 : Winner : agent user-1-11 bidding 56 and payes 56
13:58:56,338 : Job user-0-3 is filtered off 8 > 1
13:58:56,338 : Job user-1-4 is filtered off 9 > 1
13:58:56,338 : Job user-0-5 is filtered off 10 > 1
13:58:56,339 : Job user-0-7 is filtered off 6 > 1
13:58:56,339 : Job user-0-8 is filtered off 6 > 1
13:58:56,339 : Job user-0-13 is filtered off 8 > 1
13:58:56,339 : Job user-0-14 is filtered off 10 > 1
13:58:56,340 : Job user-1-15 is filtered off 7 > 1
13:58:56,340 : Job user-0-16 is filtered off 5 > 1
13:58:56,340 : Job user-0-17 is filtered off 10 > 1
13:58:56,340 : Job user-1-18 is filtered off 10 > 1
13:58:56,341 : Job user-0-19 is filtered off 7 > 1
13:58:56,341 : Job user-1-20 is filtered off 3 > 1
13:58:56,341 : Job user-0-21 is filtered off 9 > 1
13:58:56,341 : Job user-1-22 is filtered off 6 > 1
13:58:56,342 : Job user-1-23 is filtered off 5 > 1
13:58:56,342 : Job user-1-24 is filtered off 7 > 1
13:58:56,342 : Job user-0-25 is filtered off 2 > 1
13:58:56,342 : Job user-1-26 is filtered off 2 > 1
13:58:56,343 : Job user-0-27 is filtered off 6 > 1
13:58:56,343 : Job user-0-28 is filtered off 2 > 1
13:58:56,343 : Agent user-1-12 bids 56 on resource resource-0 with reservation price 0
13:58:56,343 : Highest bid 56 by user-1-12, second highest 56 by user-1-12
13:58:56,344 : Winner : agent user-1-12 bidding 56 and payes 56
13:58:56,346 : Job user-0-3 is filtered off 8 > 7
13:58:56,346 : Job user-1-4 is filtered off 9 > 7
13:58:56,346 : Job user-0-5 is filtered off 10 > 7
13:58:56,346 : Job user-0-13 is filtered off 8 > 7
13:58:56,347 : Job user-0-14 is filtered off 10 > 7
13:58:56,347 : Job user-0-17 is filtered off 10 > 7
13:58:56,347 : Job user-1-18 is filtered off 10 > 7
13:58:56,347 : Job user-0-21 is filtered off 9 > 7
13:58:56,348 : Agent user-0-7 bids 57 on resource resource-0 with reservation price 0
13:58:56,348 : Agent user-0-8 bids 57 on resource resource-0 with reservation price 0
13:58:56,348 : Agent user-1-15 bids 56 on resource resource-0 with reservation price 0
13:58:56,349 : Agent user-0-16 bids 57 on resource resource-0 with reservation price 0
13:58:56,349 : Agent user-0-19 bids 57 on resource resource-0 with reservation price 0
```

```
13:58:56,349 : Agent user-1-20 bids 56 on resource resource-0 with reservation price 0
13:58:56,350 : Agent user-1-22 bids 56 on resource resource-0 with reservation price 0
13:58:56,350 : Agent user-1-23 bids 56 on resource resource-0 with reservation price 0
13:58:56,350 : Agent user-1-24 bids 56 on resource resource-0 with reservation price 0
13:58:56,351 : Agent user-0-25 bids 57 on resource resource-0 with reservation price 0
13:58:56,351 : Agent user-1-26 bids 56 on resource resource-0 with reservation price 0
13:58:56,351 : Agent user-0-27 bids 57 on resource resource-0 with reservation price 0
13:58:56,351 : Agent user-0-28 bids 57 on resource resource-0 with reservation price 0
13:58:56,352 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,352 : Got a tie with 7 agents
13:58:56,352 : Winner : agent user-0-25 bidding 57 and payes 57
13:58:56,354 : Job user-1-4 is filtered off 9 > 8
13:58:56,355 : Job user-0-5 is filtered off 10 > 8
13:58:56,355 : Job user-0-14 is filtered off 10 > 8
13:58:56,355 : Job user-0-17 is filtered off 10 > 8
13:58:56,355 : Job user-1-18 is filtered off 10 > 8
13:58:56,355 : Job user-0-21 is filtered off 9 > 8
13:58:56,356 : Job user-0-31 is filtered off 10 > 8
13:58:56,356 : Agent user-0-3 bids 57 on resource resource-0 with reservation price 0
13:58:56,356 : Agent user-0-7 bids 57 on resource resource-0 with reservation price 0
13:58:56,356 : Agent user-0-8 bids 57 on resource resource-0 with reservation price 0
13:58:56,357 : Agent user-0-13 bids 57 on resource resource-0 with reservation price 0
13:58:56,357 : Agent user-1-15 bids 56 on resource resource-0 with reservation price 0
13:58:56,357 : Agent user-0-16 bids 57 on resource resource-0 with reservation price 0
13:58:56,358 : Agent user-0-19 bids 57 on resource resource-0 with reservation price 0
13:58:56,358 : Agent user-1-20 bids 56 on resource resource-0 with reservation price 0
13:58:56,358 : Agent user-1-22 bids 56 on resource resource-0 with reservation price 0
13:58:56,359 : Agent user-1-23 bids 56 on resource resource-0 with reservation price 0
13:58:56,359 : Agent user-1-24 bids 56 on resource resource-0 with reservation price 0
13:58:56,360 : Agent user-1-26 bids 56 on resource resource-0 with reservation price 0
13:58:56,360 : Agent user-0-27 bids 57 on resource resource-0 with reservation price 0
13:58:56,360 : Agent user-0-28 bids 57 on resource resource-0 with reservation price 0
13:58:56,360 : Agent user-1-29 bids 56 on resource resource-0 with reservation price 0
13:58:56,361 : Agent user-1-30 bids 56 on resource resource-0 with reservation price 0
13:58:56,361 : Agent user-0-32 bids 57 on resource resource-0 with reservation price 0
13:58:56,362 : Agent user-1-33 bids 56 on resource resource-0 with reservation price 0
13:58:56,362 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,362 : Got a tie with 9 agents
13:58:56,362 : Winner : agent user-0-27 bidding 57 and payes 57
13:58:56,365 : Job user-0-3 is filtered off 8 > 1
13:58:56,366 : Job user-1-4 is filtered off 9 > 1
13:58:56,366 : Job user-0-5 is filtered off 10 > 1
13:58:56,366 : Job user-0-7 is filtered off 6 > 1
13:58:56,366 : Job user-0-8 is filtered off 6 > 1
13:58:56,366 : Job user-0-13 is filtered off 8 > 1
13:58:56,367 : Job user-0-14 is filtered off 10 > 1
13:58:56,367 : Job user-1-15 is filtered off 7 > 1
13:58:56,367 : Job user-0-16 is filtered off 5 > 1
```

```
13:58:56,367 : Job user-0-17 is filtered off 10 > 1
13:58:56,367 : Job user-1-18 is filtered off 10 > 1
13:58:56,368 : Job user-0-19 is filtered off 7 > 1
13:58:56,368 : Job user-1-20 is filtered off 3 > 1
13:58:56,368 : Job user-0-21 is filtered off 9 > 1
13:58:56,368 : Job user-1-22 is filtered off 6 > 1
13:58:56,368 : Job user-1-23 is filtered off 5 > 1
13:58:56,368 : Job user-1-24 is filtered off 7 > 1
13:58:56,369 : Job user-1-26 is filtered off 2 > 1
13:58:56,369 : Job user-0-28 is filtered off 2 > 1
13:58:56,369 : Job user-1-29 is filtered off 2 > 1
13:58:56,369 : Job user-1-30 is filtered off 8 > 1
13:58:56,370 : Job user-0-31 is filtered off 10 > 1
13:58:56,370 : Job user-0-32 is filtered off 7 > 1
13:58:56,370 : Job user-1-33 is filtered off 3 > 1
13:58:56,370 : Job user-1-34 is filtered off 6 > 1
13:58:56,370 : Job user-1-35 is filtered off 9 > 1
13:58:56,371 : Job user-0-36 is filtered off 5 > 1
13:58:56,371 : Job user-1-37 is filtered off 9 > 1
13:58:56,371 : Job user-0-38 is filtered off 7 > 1
13:58:56,372 : Job user-0-3 is filtered off 8 > 3
13:58:56,372 : Job user-1-4 is filtered off 9 > 3
13:58:56,372 : Job user-0-5 is filtered off 10 > 3
13:58:56,373 : Job user-0-7 is filtered off 6 > 3
13:58:56,373 : Job user-0-8 is filtered off 6 > 3
13:58:56,373 : Job user-0-13 is filtered off 8 > 3
13:58:56,373 : Job user-0-14 is filtered off 10 > 3
13:58:56,373 : Job user-1-15 is filtered off 7 > 3
13:58:56,374 : Job user-0-16 is filtered off 5 > 3
13:58:56,374 : Job user-0-17 is filtered off 10 > 3
13:58:56,374 : Job user-1-18 is filtered off 10 > 3
13:58:56,374 : Job user-0-19 is filtered off 7 > 3
13:58:56,374 : Job user-0-21 is filtered off 9 > 3
13:58:56,375 : Job user-1-22 is filtered off 6 > 3
13:58:56,375 : Job user-1-23 is filtered off 5 > 3
13:58:56,375 : Job user-1-24 is filtered off 7 > 3
13:58:56,375 : Job user-1-30 is filtered off 8 > 3
13:58:56,376 : Job user-0-31 is filtered off 10 > 3
13:58:56,376 : Job user-0-32 is filtered off 7 > 3
13:58:56,376 : Job user-1-34 is filtered off 6 > 3
13:58:56,376 : Job user-1-35 is filtered off 9 > 3
13:58:56,376 : Job user-0-36 is filtered off 5 > 3
13:58:56,377 : Job user-1-37 is filtered off 9 > 3
13:58:56,377 : Job user-0-38 is filtered off 7 > 3
13:58:56,377 : Agent user-1-20 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,377 : Agent user-1-26 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,378 : Agent user-0-28 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,378 : Agent user-1-29 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,378 : Agent user-1-33 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,378 : Highest bid 57 by user-0-28, second highest 56 by user-1-33
13:58:56,379 : Winner : agent user-0-28 bidding 57 and payes 56
13:58:56,380 : Agent user-0-3 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,381 : Agent user-1-4 bids 56 on resource resource-0 with reservation
     price 0
13:58:56,381 : Agent user-0-5 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,382 : Agent user-0-7 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,382 : Agent user-0-8 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,382 : Agent user-0-13 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,382 : Agent user-0-14 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,383 : Agent user-1-15 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,383 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 0
```

```
13:58:56,383 : Agent user-0-17 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,384 : Agent user-1-18 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,384 : Agent user-0-19 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,384 : Agent user-1-20 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,385 : Agent user-0-21 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,385 : Agent user-1-22 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,385 : Agent user-1-23 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,385 : Agent user-1-24 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,386 : Agent user-1-26 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,386 : Agent user-1-29 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,386 : Agent user-1-30 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,387 : Agent user-0-31 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,387 : Agent user-0-32 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,387 : Agent user-1-33 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,388 : Agent user-1-34 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,388 : Agent user-1-35 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,388 : Agent user-0-36 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,388 : Agent user-1-37 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,389 : Agent user-0-38 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,389 : Agent user-0-39 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,389 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,390 : Got a tie with 15 agents
13:58:56,390 : Winner : agent user-0-32 bidding 57 and payes 57
13:58:56,395 : Job user-0-3 is filtered off 8 > 3
13:58:56,395 : Job user-1-4 is filtered off 9 > 3
13:58:56,395 : Job user-0-5 is filtered off 10 > 3
13:58:56,396 : Job user-0-7 is filtered off 6 > 3
13:58:56,396 : Job user-0-8 is filtered off 6 > 3
13:58:56,396 : Job user-0-13 is filtered off 8 > 3
13:58:56,396 : Job user-0-14 is filtered off 10 > 3
13:58:56,396 : Job user-1-15 is filtered off 7 > 3
13:58:56,397 : Job user-0-16 is filtered off 5 > 3
13:58:56,397 : Job user-0-17 is filtered off 10 > 3
13:58:56,397 : Job user-1-18 is filtered off 10 > 3
13:58:56,397 : Job user-0-19 is filtered off 7 > 3
13:58:56,397 : Job user-0-21 is filtered off 9 > 3
13:58:56,398 : Job user-1-22 is filtered off 6 > 3
13:58:56,398 : Job user-1-23 is filtered off 5 > 3
13:58:56,398 : Job user-1-24 is filtered off 7 > 3
13:58:56,398 : Job user-1-30 is filtered off 8 > 3
13:58:56,398 : Job user-0-31 is filtered off 10 > 3
13:58:56,399 : Job user-1-34 is filtered off 6 > 3
13:58:56,399 : Job user-1-35 is filtered off 9 > 3
13:58:56,399 : Job user-0-36 is filtered off 5 > 3
13:58:56,399 : Job user-1-37 is filtered off 9 > 3
13:58:56,399 : Job user-0-38 is filtered off 7 > 3
13:58:56,400 : Job user-0-39 is filtered off 9 > 3
13:58:56,400 : Job user-1-40 is filtered off 9 > 3
13:58:56,400 : Job user-1-41 is filtered off 6 > 3
13:58:56,400 : Job user-0-43 is filtered off 9 > 3
13:58:56,400 : Job user-0-44 is filtered off 10 > 3
13:58:56,401 : Job user-1-45 is filtered off 8 > 3
13:58:56,401 : Job user-0-46 is filtered off 5 > 3
13:58:56,401 : Job user-1-47 is filtered off 4 > 3
```

```
13:58:56,401 : Agent user-1-20 bids 56 on resource resource-0 with reservation price 0
13:58:56,402 : Agent user-1-26 bids 56 on resource resource-0 with reservation price 0
13:58:56,402 : Agent user-1-29 bids 56 on resource resource-0 with reservation price 0
13:58:56,402 : Agent user-1-33 bids 56 on resource resource-0 with reservation price 0
13:58:56,403 : Agent user-0-42 bids 57 on resource resource-0 with reservation price 0
13:58:56,403 : Agent user-0-48 bids 57 on resource resource-0 with reservation price 0
13:58:56,403 : Highest bid 57 by user-0-48, second highest 57 by user-0-42
13:58:56,403 : Got a tie with 2 agents
13:58:56,404 : Winner : agent user-0-48 bidding 57 and payes 57
13:58:56,406 : Job user-0-3 is filtered off 8 > 2
13:58:56,406 : Job user-1-4 is filtered off 9 > 2
13:58:56,406 : Job user-0-5 is filtered off 10 > 2
13:58:56,406 : Job user-0-7 is filtered off 6 > 2
13:58:56,407 : Job user-0-8 is filtered off 6 > 2
13:58:56,407 : Job user-0-13 is filtered off 8 > 2
13:58:56,407 : Job user-0-14 is filtered off 10 > 2
13:58:56,407 : Job user-1-15 is filtered off 7 > 2
13:58:56,407 : Job user-0-16 is filtered off 5 > 2
13:58:56,408 : Job user-0-17 is filtered off 10 > 2
13:58:56,408 : Job user-1-18 is filtered off 10 > 2
13:58:56,408 : Job user-0-19 is filtered off 7 > 2
13:58:56,408 : Job user-1-20 is filtered off 3 > 2
13:58:56,408 : Job user-0-21 is filtered off 9 > 2
13:58:56,408 : Job user-1-22 is filtered off 6 > 2
13:58:56,409 : Job user-1-23 is filtered off 5 > 2
13:58:56,409 : Job user-1-24 is filtered off 7 > 2
13:58:56,409 : Job user-1-30 is filtered off 8 > 2
13:58:56,409 : Job user-0-31 is filtered off 10 > 2
13:58:56,409 : Job user-1-33 is filtered off 3 > 2
13:58:56,410 : Job user-1-34 is filtered off 6 > 2
13:58:56,410 : Job user-1-35 is filtered off 9 > 2
13:58:56,410 : Job user-0-36 is filtered off 5 > 2
13:58:56,410 : Job user-1-37 is filtered off 9 > 2
13:58:56,411 : Job user-0-38 is filtered off 7 > 2
13:58:56,411 : Job user-0-39 is filtered off 9 > 2
13:58:56,411 : Job user-1-40 is filtered off 9 > 2
13:58:56,411 : Job user-1-41 is filtered off 6 > 2
13:58:56,411 : Job user-0-43 is filtered off 9 > 2
13:58:56,412 : Job user-0-44 is filtered off 10 > 2
13:58:56,412 : Job user-1-45 is filtered off 8 > 2
13:58:56,412 : Job user-0-46 is filtered off 5 > 2
13:58:56,412 : Job user-1-47 is filtered off 4 > 2
13:58:56,412 : Job user-0-49 is filtered off 3 > 2
13:58:56,413 : Job user-0-50 is filtered off 4 > 2
13:58:56,413 : Job user-0-51 is filtered off 6 > 2
13:58:56,413 : Agent user-1-26 bids 56 on resource resource-0 with reservation price 0
13:58:56,413 : Agent user-1-29 bids 56 on resource resource-0 with reservation price 0
13:58:56,414 : Agent user-0-42 bids 57 on resource resource-0 with reservation price 0
13:58:56,414 : Highest bid 57 by user-0-42, second highest 56 by user-1-29
13:58:56,414 : Winner : agent user-0-42 bidding 57 and payes 56
13:58:56,416 : Job user-0-3 is filtered off 8 > 6
13:58:56,416 : Job user-1-4 is filtered off 9 > 6
13:58:56,416 : Job user-0-5 is filtered off 10 > 6
13:58:56,417 : Job user-0-13 is filtered off 8 > 6
13:58:56,417 : Job user-0-14 is filtered off 10 > 6
13:58:56,417 : Job user-1-15 is filtered off 7 > 6
13:58:56,417 : Job user-0-17 is filtered off 10 > 6
13:58:56,417 : Job user-1-18 is filtered off 10 > 6
13:58:56,418 : Job user-0-19 is filtered off 7 > 6
13:58:56,418 : Job user-0-21 is filtered off 9 > 6
13:58:56,418 : Job user-1-24 is filtered off 7 > 6
13:58:56,418 : Job user-1-30 is filtered off 8 > 6
13:58:56,419 : Job user-0-31 is filtered off 10 > 6
13:58:56,419 : Job user-1-35 is filtered off 9 > 6
13:58:56,419 : Job user-1-37 is filtered off 9 > 6
```

```
13:58:56,419 : Job user-0-38 is filtered off 7 > 6
13:58:56,419 : Job user-0-39 is filtered off 9 > 6
13:58:56,420 : Job user-1-40 is filtered off 9 > 6
13:58:56,420 : Job user-0-43 is filtered off 9 > 6
13:58:56,420 : Job user-0-44 is filtered off 10 > 6
13:58:56,420 : Job user-1-45 is filtered off 8 > 6
13:58:56,421 : Job user-1-52 is filtered off 7 > 6
13:58:56,421 : Agent user-0-7 bids 57 on resource resource-0 with reservation price 0
13:58:56,421 : Agent user-0-8 bids 57 on resource resource-0 with reservation price 0
13:58:56,422 : Agent user-0-16 bids 57 on resource resource-0 with reservation price 0
13:58:56,422 : Agent user-1-20 bids 56 on resource resource-0 with reservation price 0
13:58:56,422 : Agent user-1-22 bids 56 on resource resource-0 with reservation price 0
13:58:56,423 : Agent user-1-23 bids 56 on resource resource-0 with reservation price 0
13:58:56,423 : Agent user-1-26 bids 56 on resource resource-0 with reservation price 0
13:58:56,423 : Agent user-1-29 bids 56 on resource resource-0 with reservation price 0
13:58:56,424 : Agent user-1-33 bids 56 on resource resource-0 with reservation price 0
13:58:56,424 : Agent user-1-34 bids 56 on resource resource-0 with reservation price 0
13:58:56,424 : Agent user-0-36 bids 57 on resource resource-0 with reservation price 0
13:58:56,424 : Agent user-1-41 bids 56 on resource resource-0 with reservation price 0
13:58:56,425 : Agent user-0-46 bids 57 on resource resource-0 with reservation price 0
13:58:56,425 : Agent user-1-47 bids 56 on resource resource-0 with reservation price 0
13:58:56,425 : Agent user-0-49 bids 57 on resource resource-0 with reservation price 0
13:58:56,426 : Agent user-0-50 bids 57 on resource resource-0 with reservation price 0
13:58:56,426 : Agent user-0-51 bids 57 on resource resource-0 with reservation price 0
13:58:56,426 : Agent user-0-53 bids 57 on resource resource-0 with reservation price 0
13:58:56,426 : Agent user-1-54 bids 56 on resource resource-0 with reservation price 0
13:58:56,427 : Agent user-1-55 bids 56 on resource resource-0 with reservation price 0
13:58:56,427 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,427 : Got a tie with 9 agents
13:58:56,428 : Winner : agent user-0-49 bidding 57 and payes 57
13:58:56,429 : Job user-0-3 is filtered off 8 > 4
13:58:56,430 : Job user-1-4 is filtered off 9 > 4
13:58:56,430 : Job user-0-5 is filtered off 10 > 4
13:58:56,430 : Job user-0-7 is filtered off 6 > 4
13:58:56,430 : Job user-0-8 is filtered off 6 > 4
13:58:56,431 : Job user-0-13 is filtered off 8 > 4
13:58:56,431 : Job user-0-14 is filtered off 10 > 4
13:58:56,431 : Job user-1-15 is filtered off 7 > 4
13:58:56,432 : Job user-0-16 is filtered off 5 > 4
13:58:56,432 : Job user-0-17 is filtered off 10 > 4
13:58:56,432 : Job user-1-18 is filtered off 10 > 4
13:58:56,432 : Job user-0-19 is filtered off 7 > 4
13:58:56,433 : Job user-0-21 is filtered off 9 > 4
13:58:56,433 : Job user-1-22 is filtered off 6 > 4
13:58:56,433 : Job user-1-23 is filtered off 5 > 4
13:58:56,434 : Job user-1-24 is filtered off 7 > 4
13:58:56,434 : Job user-1-30 is filtered off 8 > 4
13:58:56,434 : Job user-0-31 is filtered off 10 > 4
13:58:56,435 : Job user-1-34 is filtered off 6 > 4
13:58:56,435 : Job user-1-35 is filtered off 9 > 4
13:58:56,435 : Job user-0-36 is filtered off 5 > 4
13:58:56,436 : Job user-1-37 is filtered off 9 > 4
13:58:56,436 : Job user-0-38 is filtered off 7 > 4
13:58:56,436 : Job user-0-39 is filtered off 9 > 4
```

```
13:58:56,437 : Job user-1-40 is filtered off 9 > 4
13:58:56,437 : Job user-1-41 is filtered off 6 > 4
13:58:56,437 : Job user-0-43 is filtered off 9 > 4
13:58:56,438 : Job user-0-44 is filtered off 10 > 4
13:58:56,438 : Job user-1-45 is filtered off 8 > 4
13:58:56,438 : Job user-0-46 is filtered off 5 > 4
13:58:56,438 : Job user-0-51 is filtered off 6 > 4
13:58:56,438 : Job user-1-52 is filtered off 7 > 4
13:58:56,439 : Job user-0-53 is filtered off 5 > 4
13:58:56,439 : Job user-1-57 is filtered off 6 > 4
13:58:56,439 : Agent user-1-20 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,440 : Agent user-1-26 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,440 : Agent user-1-29 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,441 : Agent user-1-33 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,441 : Agent user-1-47 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,441 : Agent user-0-50 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,442 : Agent user-1-54 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,442 : Agent user-1-55 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,442 : Agent user-1-56 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,443 : Highest bid 57 by user-0-50, second highest 56 by user-1-56
13:58:56,443 : Winner : agent user-0-50 bidding 57 and payes 56
13:58:56,447 : Job user-0-3 is filtered off 8 > 1
13:58:56,448 : Job user-1-4 is filtered off 9 > 1
13:58:56,448 : Job user-0-5 is filtered off 10 > 1
13:58:56,448 : Job user-0-7 is filtered off 6 > 1
13:58:56,448 : Job user-0-8 is filtered off 6 > 1
13:58:56,448 : Job user-0-13 is filtered off 8 > 1
13:58:56,448 : Job user-0-14 is filtered off 10 > 1
13:58:56,449 : Job user-1-15 is filtered off 7 > 1
13:58:56,449 : Job user-0-16 is filtered off 5 > 1
13:58:56,449 : Job user-0-17 is filtered off 10 > 1
13:58:56,449 : Job user-1-18 is filtered off 10 > 1
13:58:56,449 : Job user-0-19 is filtered off 7 > 1
13:58:56,450 : Job user-1-20 is filtered off 3 > 1
13:58:56,450 : Job user-0-21 is filtered off 9 > 1
13:58:56,450 : Job user-1-22 is filtered off 6 > 1
13:58:56,450 : Job user-1-23 is filtered off 5 > 1
13:58:56,450 : Job user-1-24 is filtered off 7 > 1
13:58:56,450 : Job user-1-26 is filtered off 2 > 1
13:58:56,451 : Job user-1-29 is filtered off 2 > 1
13:58:56,451 : Job user-1-30 is filtered off 8 > 1
13:58:56,451 : Job user-0-31 is filtered off 10 > 1
13:58:56,451 : Job user-1-33 is filtered off 3 > 1
13:58:56,452 : Job user-1-34 is filtered off 6 > 1
13:58:56,452 : Job user-1-35 is filtered off 9 > 1
13:58:56,452 : Job user-0-36 is filtered off 5 > 1
13:58:56,452 : Job user-1-37 is filtered off 9 > 1
13:58:56,453 : Job user-0-38 is filtered off 7 > 1
13:58:56,453 : Job user-0-39 is filtered off 9 > 1
13:58:56,453 : Job user-1-40 is filtered off 9 > 1
13:58:56,453 : Job user-1-41 is filtered off 6 > 1
13:58:56,454 : Job user-0-43 is filtered off 9 > 1
13:58:56,454 : Job user-0-44 is filtered off 10 > 1
13:58:56,454 : Job user-1-45 is filtered off 8 > 1
13:58:56,455 : Job user-0-46 is filtered off 5 > 1
13:58:56,455 : Job user-1-47 is filtered off 4 > 1
13:58:56,455 : Job user-0-51 is filtered off 6 > 1
13:58:56,455 : Job user-1-52 is filtered off 7 > 1
13:58:56,455 : Job user-0-53 is filtered off 5 > 1
13:58:56,456 : Job user-1-54 is filtered off 2 > 1
13:58:56,456 : Job user-1-55 is filtered off 2 > 1
13:58:56,456 : Job user-1-57 is filtered off 6 > 1
13:58:56,457 : Job user-0-58 is filtered off 2 > 1
13:58:56,457 : Job user-1-59 is filtered off 9 > 1
13:58:56,457 : Job user-1-61 is filtered off 9 > 1
```

```
13:58:56,457 : Agent user-1-56 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,458 : Agent user-0-60 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,458 : Highest bid 57 by user-0-60, second highest 57 by user-0-60
13:58:56,459 : Winner : agent user-0-60 bidding 57 and payes 57
13:58:56,460 : Job user-0-3 is filtered off 8 > 1
13:58:56,460 : Job user-1-4 is filtered off 9 > 1
13:58:56,461 : Job user-0-5 is filtered off 10 > 1
13:58:56,461 : Job user-0-7 is filtered off 6 > 1
13:58:56,461 : Job user-0-8 is filtered off 6 > 1
13:58:56,461 : Job user-0-13 is filtered off 8 > 1
13:58:56,462 : Job user-0-14 is filtered off 10 > 1
13:58:56,462 : Job user-1-15 is filtered off 7 > 1
13:58:56,462 : Job user-0-16 is filtered off 5 > 1
13:58:56,462 : Job user-0-17 is filtered off 10 > 1
13:58:56,463 : Job user-1-18 is filtered off 10 > 1
13:58:56,463 : Job user-0-19 is filtered off 7 > 1
13:58:56,463 : Job user-1-20 is filtered off 3 > 1
13:58:56,463 : Job user-0-21 is filtered off 9 > 1
13:58:56,464 : Job user-1-22 is filtered off 6 > 1
13:58:56,464 : Job user-1-23 is filtered off 5 > 1
13:58:56,464 : Job user-1-24 is filtered off 7 > 1
13:58:56,465 : Job user-1-26 is filtered off 2 > 1
13:58:56,465 : Job user-1-29 is filtered off 2 > 1
13:58:56,465 : Job user-1-30 is filtered off 8 > 1
13:58:56,465 : Job user-0-31 is filtered off 10 > 1
13:58:56,465 : Job user-1-33 is filtered off 3 > 1
13:58:56,466 : Job user-1-34 is filtered off 6 > 1
13:58:56,466 : Job user-1-35 is filtered off 9 > 1
13:58:56,466 : Job user-0-36 is filtered off 5 > 1
13:58:56,467 : Job user-1-37 is filtered off 9 > 1
13:58:56,468 : Job user-0-38 is filtered off 7 > 1
13:58:56,468 : Job user-0-39 is filtered off 9 > 1
13:58:56,468 : Job user-1-40 is filtered off 9 > 1
13:58:56,469 : Job user-1-41 is filtered off 6 > 1
13:58:56,469 : Job user-0-43 is filtered off 9 > 1
13:58:56,469 : Job user-0-44 is filtered off 10 > 1
13:58:56,469 : Job user-1-45 is filtered off 8 > 1
13:58:56,470 : Job user-0-46 is filtered off 5 > 1
13:58:56,470 : Job user-1-47 is filtered off 4 > 1
13:58:56,470 : Job user-0-51 is filtered off 6 > 1
13:58:56,470 : Job user-1-52 is filtered off 7 > 1
13:58:56,471 : Job user-0-53 is filtered off 5 > 1
13:58:56,471 : Job user-1-54 is filtered off 2 > 1
13:58:56,472 : Job user-1-55 is filtered off 2 > 1
13:58:56,472 : Job user-1-57 is filtered off 6 > 1
13:58:56,472 : Job user-0-58 is filtered off 2 > 1
13:58:56,472 : Job user-1-59 is filtered off 9 > 1
13:58:56,473 : Job user-1-61 is filtered off 9 > 1
13:58:56,473 : Agent user-1-56 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,473 : Highest bid 56 by user-1-56, second highest 56 by user-1-56
13:58:56,474 : Winner : agent user-1-56 bidding 56 and payes 56
13:58:56,476 : Job user-0-3 is filtered off 8 > 7
13:58:56,477 : Job user-1-4 is filtered off 9 > 7
13:58:56,477 : Job user-0-5 is filtered off 10 > 7
13:58:56,478 : Job user-0-13 is filtered off 8 > 7
13:58:56,479 : Job user-0-14 is filtered off 10 > 7
13:58:56,479 : Job user-0-17 is filtered off 10 > 7
13:58:56,479 : Job user-1-18 is filtered off 10 > 7
13:58:56,480 : Job user-0-21 is filtered off 9 > 7
13:58:56,480 : Job user-1-30 is filtered off 8 > 7
13:58:56,480 : Job user-0-31 is filtered off 10 > 7
13:58:56,480 : Job user-1-35 is filtered off 9 > 7
13:58:56,481 : Job user-1-37 is filtered off 9 > 7
13:58:56,482 : Job user-0-39 is filtered off 9 > 7
13:58:56,482 : Job user-1-40 is filtered off 9 > 7
13:58:56,482 : Job user-0-43 is filtered off 9 > 7
13:58:56,483 : Job user-0-44 is filtered off 10 > 7
13:58:56,483 : Job user-1-45 is filtered off 8 > 7
13:58:56,483 : Job user-1-59 is filtered off 9 > 7
13:58:56,483 : Job user-1-61 is filtered off 9 > 7
```

```
13:58:56,484 : Agent user-0-7 bids 57 on resource resource-0 with reservation price 0
13:58:56,484 : Agent user-0-8 bids 57 on resource resource-0 with reservation price 0
13:58:56,485 : Agent user-1-15 bids 56 on resource resource-0 with reservation price 0
13:58:56,485 : Agent user-0-16 bids 57 on resource resource-0 with reservation price 0
13:58:56,485 : Agent user-0-19 bids 57 on resource resource-0 with reservation price 0
13:58:56,486 : Agent user-1-20 bids 56 on resource resource-0 with reservation price 0
13:58:56,486 : Agent user-1-22 bids 56 on resource resource-0 with reservation price 0
13:58:56,486 : Agent user-1-23 bids 56 on resource resource-0 with reservation price 0
13:58:56,487 : Agent user-1-24 bids 56 on resource resource-0 with reservation price 0
13:58:56,487 : Agent user-1-26 bids 56 on resource resource-0 with reservation price 0
13:58:56,488 : Agent user-1-29 bids 56 on resource resource-0 with reservation price 0
13:58:56,488 : Agent user-1-33 bids 56 on resource resource-0 with reservation price 0
13:58:56,488 : Agent user-1-34 bids 56 on resource resource-0 with reservation price 0
13:58:56,489 : Agent user-0-36 bids 57 on resource resource-0 with reservation price 0
13:58:56,489 : Agent user-0-38 bids 57 on resource resource-0 with reservation price 0
13:58:56,490 : Agent user-1-41 bids 56 on resource resource-0 with reservation price 0
13:58:56,490 : Agent user-0-46 bids 57 on resource resource-0 with reservation price 0
13:58:56,490 : Agent user-1-47 bids 56 on resource resource-0 with reservation price 0
13:58:56,491 : Agent user-0-51 bids 57 on resource resource-0 with reservation price 0
13:58:56,493 : Agent user-1-52 bids 56 on resource resource-0 with reservation price 0
13:58:56,494 : Agent user-0-53 bids 57 on resource resource-0 with reservation price 0
13:58:56,494 : Agent user-1-54 bids 56 on resource resource-0 with reservation price 0
13:58:56,495 : Agent user-1-55 bids 56 on resource resource-0 with reservation price 0
13:58:56,496 : Agent user-1-57 bids 56 on resource resource-0 with reservation price 0
13:58:56,497 : Agent user-0-58 bids 57 on resource resource-0 with reservation price 0
13:58:56,498 : Agent user-1-62 bids 56 on resource resource-0 with reservation price 0
13:58:56,498 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,499 : Got a tie with 10 agents
13:58:56,500 : Winner : agent user-0-51 bidding 57 and payes 57
13:58:56,503 : Job user-0-3 is filtered off 8 > 1
13:58:56,504 : Job user-1-4 is filtered off 9 > 1
13:58:56,504 : Job user-0-5 is filtered off 10 > 1
13:58:56,504 : Job user-0-7 is filtered off 6 > 1
13:58:56,504 : Job user-0-8 is filtered off 6 > 1
13:58:56,504 : Job user-0-13 is filtered off 8 > 1
13:58:56,505 : Job user-0-14 is filtered off 10 > 1
13:58:56,505 : Job user-1-15 is filtered off 7 > 1
13:58:56,505 : Job user-0-16 is filtered off 5 > 1
13:58:56,505 : Job user-0-17 is filtered off 10 > 1
13:58:56,505 : Job user-1-18 is filtered off 10 > 1
13:58:56,506 : Job user-0-19 is filtered off 7 > 1
13:58:56,506 : Job user-1-20 is filtered off 3 > 1
13:58:56,506 : Job user-0-21 is filtered off 9 > 1
13:58:56,506 : Job user-1-22 is filtered off 6 > 1
13:58:56,506 : Job user-1-23 is filtered off 5 > 1
13:58:56,506 : Job user-1-24 is filtered off 7 > 1
13:58:56,507 : Job user-1-26 is filtered off 2 > 1
13:58:56,507 : Job user-1-29 is filtered off 2 > 1
```

```
13:58:56,507 : Job user-1-30 is filtered off 8 > 1
13:58:56,507 : Job user-0-31 is filtered off 10 > 1
13:58:56,507 : Job user-1-33 is filtered off 3 > 1
13:58:56,508 : Job user-1-34 is filtered off 6 > 1
13:58:56,508 : Job user-1-35 is filtered off 9 > 1
13:58:56,508 : Job user-0-36 is filtered off 5 > 1
13:58:56,508 : Job user-1-37 is filtered off 9 > 1
13:58:56,508 : Job user-0-38 is filtered off 7 > 1
13:58:56,509 : Job user-0-39 is filtered off 9 > 1
13:58:56,509 : Job user-1-40 is filtered off 9 > 1
13:58:56,509 : Job user-1-41 is filtered off 6 > 1
13:58:56,509 : Job user-0-43 is filtered off 9 > 1
13:58:56,509 : Job user-0-44 is filtered off 10 > 1
13:58:56,509 : Job user-1-45 is filtered off 8 > 1
13:58:56,510 : Job user-0-46 is filtered off 5 > 1
13:58:56,510 : Job user-1-47 is filtered off 4 > 1
13:58:56,510 : Job user-1-52 is filtered off 7 > 1
13:58:56,510 : Job user-0-53 is filtered off 5 > 1
13:58:56,510 : Job user-1-54 is filtered off 2 > 1
13:58:56,511 : Job user-1-55 is filtered off 2 > 1
13:58:56,511 : Job user-1-57 is filtered off 6 > 1
13:58:56,511 : Job user-0-58 is filtered off 2 > 1
13:58:56,511 : Job user-1-59 is filtered off 9 > 1
13:58:56,511 : Job user-1-61 is filtered off 9 > 1
13:58:56,512 : Job user-0-63 is filtered off 7 > 1
13:58:56,512 : Job user-1-64 is filtered off 5 > 1
13:58:56,512 : Job user-0-65 is filtered off 3 > 1
13:58:56,512 : Job user-0-66 is filtered off 4 > 1
13:58:56,513 : Job user-1-67 is filtered off 4 > 1
13:58:56,513 : Agent user-1-62 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,513 : Highest bid 56 by user-1-62, second highest 56 by user-1-62
13:58:56,514 : Winner : agent user-1-62 bidding 56 and payes 56
13:58:56,515 : Job user-0-3 is filtered off 8 > 6
13:58:56,515 : Job user-1-4 is filtered off 9 > 6
13:58:56,515 : Job user-0-5 is filtered off 10 > 6
13:58:56,515 : Job user-0-13 is filtered off 8 > 6
13:58:56,515 : Job user-0-14 is filtered off 10 > 6
13:58:56,516 : Job user-1-15 is filtered off 7 > 6
13:58:56,516 : Job user-0-17 is filtered off 10 > 6
13:58:56,516 : Job user-1-18 is filtered off 10 > 6
13:58:56,516 : Job user-0-19 is filtered off 7 > 6
13:58:56,516 : Job user-0-21 is filtered off 9 > 6
13:58:56,517 : Job user-1-24 is filtered off 7 > 6
13:58:56,517 : Job user-1-30 is filtered off 8 > 6
13:58:56,517 : Job user-0-31 is filtered off 10 > 6
13:58:56,517 : Job user-1-35 is filtered off 9 > 6
13:58:56,517 : Job user-1-37 is filtered off 9 > 6
13:58:56,518 : Job user-0-38 is filtered off 7 > 6
13:58:56,518 : Job user-0-39 is filtered off 9 > 6
13:58:56,518 : Job user-1-40 is filtered off 9 > 6
13:58:56,518 : Job user-0-43 is filtered off 9 > 6
13:58:56,518 : Job user-0-44 is filtered off 10 > 6
13:58:56,518 : Job user-1-45 is filtered off 8 > 6
13:58:56,519 : Job user-1-52 is filtered off 7 > 6
13:58:56,519 : Job user-1-59 is filtered off 9 > 6
13:58:56,519 : Job user-1-61 is filtered off 9 > 6
13:58:56,519 : Job user-0-63 is filtered off 7 > 6
13:58:56,519 : Job user-1-68 is filtered off 8 > 6
13:58:56,520 : Job user-1-69 is filtered off 9 > 6
13:58:56,520 : Agent user-0-7 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,520 : Agent user-0-8 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,521 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,521 : Agent user-1-20 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,521 : Agent user-1-22 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,521 : Agent user-1-23 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,522 : Agent user-1-26 bids 56 on resource resource-0 with
    reservation price 0
```

```
13:58:56,522 : Agent user-1-29 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,522 : Agent user-1-33 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,523 : Agent user-1-34 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,523 : Agent user-0-36 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,524 : Agent user-1-41 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,524 : Agent user-0-46 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,524 : Agent user-1-47 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,524 : Agent user-0-53 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,525 : Agent user-1-54 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,525 : Agent user-1-55 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,525 : Agent user-1-57 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,525 : Agent user-0-58 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,526 : Agent user-1-64 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,526 : Agent user-0-65 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,526 : Agent user-0-66 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,527 : Agent user-1-67 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,527 : Highest bid 57 by user-0-8, second highest 57 by user-0-7
13:58:56,527 : Got a tie with 9 agents
13:58:56,528 : Winner : agent user-0-58 bidding 57 and payes 57
13:58:56,529 : Job user-0-5 is filtered off 10 > 9
13:58:56,529 : Job user-0-14 is filtered off 10 > 9
13:58:56,530 : Job user-0-17 is filtered off 10 > 9
13:58:56,530 : Job user-1-18 is filtered off 10 > 9
13:58:56,530 : Job user-0-31 is filtered off 10 > 9
13:58:56,530 : Job user-0-44 is filtered off 10 > 9
13:58:56,531 : Agent user-0-3 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,531 : Agent user-1-4 bids 56 on resource resource-0 with reservation
     price 0
13:58:56,531 : Agent user-0-7 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,532 : Agent user-0-8 bids 57 on resource resource-0 with reservation
     price 0
13:58:56,532 : Agent user-0-13 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,533 : Agent user-1-15 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,533 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,533 : Agent user-0-19 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,534 : Agent user-1-20 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,534 : Agent user-0-21 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,534 : Agent user-1-22 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,534 : Agent user-1-23 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,535 : Agent user-1-24 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,535 : Agent user-1-26 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,536 : Agent user-1-29 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,536 : Agent user-1-30 bids 56 on resource resource-0 with
    reservation price 0
```

```
13:58:56,536 : Agent user-1-33 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,537 : Agent user-1-34 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,537 : Agent user-1-35 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,537 : Agent user-0-36 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,538 : Agent user-1-37 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,538 : Agent user-0-38 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,539 : Agent user-0-39 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,539 : Agent user-1-40 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,539 : Agent user-1-41 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,540 : Agent user-0-43 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,540 : Agent user-1-45 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,540 : Agent user-0-46 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,541 : Agent user-1-47 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,541 : Agent user-1-52 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,541 : Agent user-0-53 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,542 : Agent user-1-54 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,542 : Agent user-1-55 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,543 : Agent user-1-57 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,543 : Agent user-1-59 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,543 : Agent user-1-61 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,544 : Agent user-0-63 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,544 : Agent user-1-64 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,544 : Agent user-0-65 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,544 : Agent user-0-66 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,545 : Agent user-1-67 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,545 : Agent user-1-68 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,545 : Agent user-1-69 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,546 : Agent user-0-70 bids 57 on resource resource-0 with
    reservation price 0
13:58:56,546 : Agent user-1-71 bids 56 on resource resource-0 with
    reservation price 0
13:58:56,546 : Highest bid 57 by user-0-8, second highest 57 by user-0-70
13:58:56,546 : Got a tie with 17 agents
13:58:56,547 : Winner : agent user-0-53 bidding 57 and payes 57
```

D.2 Reservation-Price Test

```
15:28:32,237 : Agent user-0-0 bids 57 on resource resource-0 with reservation
     price 57
15:28:32,237 : Highest bid 57 by user-0-0, second highest 57 by user-0-0
15:28:32,238 : Winner : agent user-0-0 bidding 57 and payes 57
15:28:32,244 : Job user-1-3 is filtered off 8 > 7
15:28:32,245 : Bid from user-1-1 lower than reservation price , 56 < 57
```

```
15:28:32,245 : Agent user-0-2 bids 57 on resource resource-0 with reservation price 57
15:28:32,246 : Agent user-0-4 bids 57 on resource resource-0 with reservation price 57
15:28:32,246 : Highest bid 57 by user-0-4, second highest 57 by user-0-4
15:28:32,246 : Winner : agent user-0-4 bidding 57 and payes 57
15:28:32,248 : Job user-1-3 is filtered off 8 > 2
15:28:32,248 : Job user-0-5 is filtered off 6 > 2
15:28:32,248 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,249 : Agent user-0-2 bids 57 on resource resource-0 with reservation price 57
15:28:32,249 : Highest bid 57 by user-0-2, second highest 57 by user-0-2
15:28:32,249 : Winner : agent user-0-2 bidding 57 and payes 57
15:28:32,252 : Job user-1-3 is filtered off 8 > 7
15:28:32,253 : Job user-1-8 is filtered off 10 > 7
15:28:32,253 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,253 : Agent user-0-5 bids 57 on resource resource-0 with reservation price 57
15:28:32,254 : Agent user-0-6 bids 57 on resource resource-0 with reservation price 57
15:28:32,254 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,254 : Highest bid 57 by user-0-6, second highest 57 by user-0-6
15:28:32,254 : Winner : agent user-0-6 bidding 57 and payes 57
15:28:32,260 : Job user-1-3 is filtered off 8 > 5
15:28:32,260 : Job user-0-5 is filtered off 6 > 5
15:28:32,260 : Job user-1-8 is filtered off 10 > 5
15:28:32,261 : Job user-0-10 is filtered off 6 > 5
15:28:32,261 : Job user-0-11 is filtered off 10 > 5
15:28:32,261 : Job user-1-12 is filtered off 8 > 5
15:28:32,261 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,262 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,262 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,263 : Agent user-0-13 bids 57 on resource resource-0 with reservation price 57
15:28:32,263 : Highest bid 57 by user-0-13, second highest 57 by user-0-13
15:28:32,264 : Winner : agent user-0-13 bidding 57 and payes 57
15:28:32,266 : Job user-1-3 is filtered off 8 > 6
15:28:32,266 : Job user-1-8 is filtered off 10 > 6
15:28:32,266 : Job user-0-11 is filtered off 10 > 6
15:28:32,267 : Job user-1-12 is filtered off 8 > 6
15:28:32,267 : Job user-0-15 is filtered off 7 > 6
15:28:32,268 : Job user-0-16 is filtered off 7 > 6
15:28:32,268 : Job user-1-17 is filtered off 9 > 6
15:28:32,268 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,269 : Agent user-0-5 bids 57 on resource resource-0 with reservation price 57
15:28:32,269 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,269 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,270 : Agent user-0-10 bids 57 on resource resource-0 with reservation price 57
15:28:32,270 : Agent user-0-14 bids 57 on resource resource-0 with reservation price 57
15:28:32,270 : Highest bid 57 by user-0-5, second highest 57 by user-0-14
15:28:32,271 : Got a tie with 3 agents
15:28:32,271 : Winner : agent user-0-5 bidding 57 and payes 57
15:28:32,275 : Job user-1-1 is filtered off 2 > 1
15:28:32,275 : Job user-1-3 is filtered off 8 > 1
15:28:32,276 : Job user-1-7 is filtered off 2 > 1
15:28:32,276 : Job user-1-8 is filtered off 10 > 1
15:28:32,276 : Job user-1-9 is filtered off 2 > 1
15:28:32,276 : Job user-0-10 is filtered off 6 > 1
15:28:32,277 : Job user-0-11 is filtered off 10 > 1
15:28:32,277 : Job user-1-12 is filtered off 8 > 1
15:28:32,277 : Job user-0-15 is filtered off 7 > 1
15:28:32,277 : Job user-0-16 is filtered off 7 > 1
15:28:32,277 : Job user-1-17 is filtered off 9 > 1
15:28:32,278 : Job user-1-18 is filtered off 7 > 1
15:28:32,278 : Job user-1-19 is filtered off 10 > 1
15:28:32,278 : Job user-0-20 is filtered off 3 > 1
15:28:32,278 : Job user-0-21 is filtered off 8 > 1
15:28:32,279 : Job user-0-23 is filtered off 9 > 1
15:28:32,279 : Agent user-0-14 bids 57 on resource resource-0 with reservation price 57
15:28:32,280 : Bid from user-1-22 lower than reservation price, 56 < 57
```

```
15:28:32,280 : Highest bid 57 by user-0-14, second highest 57 by user-0-14
15:28:32,280 : Winner : agent user-0-14 bidding 57 and payes 57
15:28:32,282 : Job user-1-8 is filtered off 10 > 9
15:28:32,282 : Job user-0-11 is filtered off 10 > 9
15:28:32,283 : Job user-1-19 is filtered off 10 > 9
15:28:32,283 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,283 : Bid from user-1-3 lower than reservation price, 56 < 57
15:28:32,284 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,284 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,284 : Agent user-0-10 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,285 : Bid from user-1-12 lower than reservation price, 56 < 57
15:28:32,285 : Agent user-0-15 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,285 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,286 : Bid from user-1-17 lower than reservation price, 56 < 57
15:28:32,286 : Bid from user-1-18 lower than reservation price, 56 < 57
15:28:32,287 : Agent user-0-20 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,287 : Agent user-0-21 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,289 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,290 : Agent user-0-23 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,290 : Agent user-0-24 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,290 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,290 : Bid from user-1-26 lower than reservation price, 56 < 57
15:28:32,291 : Highest bid 57 by user-0-25, second highest 57 by user-0-24
15:28:32,291 : Got a tie with 8 agents
15:28:32,291 : Winner : agent user-0-15 bidding 57 and payes 57
15:28:32,296 : Job user-1-3 is filtered off 8 > 7
15:28:32,296 : Job user-1-8 is filtered off 10 > 7
15:28:32,296 : Job user-0-11 is filtered off 10 > 7
15:28:32,296 : Job user-1-12 is filtered off 8 > 7
15:28:32,297 : Job user-1-17 is filtered off 9 > 7
15:28:32,297 : Job user-1-19 is filtered off 10 > 7
15:28:32,297 : Job user-0-21 is filtered off 8 > 7
15:28:32,297 : Job user-0-23 is filtered off 9 > 7
15:28:32,298 : Job user-0-28 is filtered off 8 > 7
15:28:32,298 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,298 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,299 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,299 : Agent user-0-10 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,300 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,300 : Bid from user-1-18 lower than reservation price, 56 < 57
15:28:32,300 : Agent user-0-20 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,301 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,301 : Agent user-0-24 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,301 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,302 : Bid from user-1-26 lower than reservation price, 56 < 57
15:28:32,302 : Agent user-0-27 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,302 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,303 : Highest bid 57 by user-0-27, second highest 57 by user-0-25
15:28:32,303 : Got a tie with 6 agents
15:28:32,303 : Winner : agent user-0-10 bidding 57 and payes 57
15:28:32,306 : Job user-1-1 is filtered off 2 > 1
15:28:32,306 : Job user-1-3 is filtered off 8 > 1
15:28:32,306 : Job user-1-7 is filtered off 2 > 1
15:28:32,306 : Job user-1-8 is filtered off 10 > 1
15:28:32,307 : Job user-1-9 is filtered off 2 > 1
15:28:32,307 : Job user-0-11 is filtered off 10 > 1
15:28:32,307 : Job user-1-12 is filtered off 8 > 1
15:28:32,307 : Job user-0-16 is filtered off 7 > 1
15:28:32,308 : Job user-1-17 is filtered off 9 > 1
15:28:32,308 : Job user-1-18 is filtered off 7 > 1
```

```
15:28:32,308 : Job user-1-19 is filtered off 10 > 1
15:28:32,308 : Job user-0-20 is filtered off 3 > 1
15:28:32,308 : Job user-0-21 is filtered off 8 > 1
15:28:32,308 : Job user-0-23 is filtered off 9 > 1
15:28:32,309 : Job user-0-24 is filtered off 2 > 1
15:28:32,309 : Job user-0-25 is filtered off 3 > 1
15:28:32,309 : Job user-1-26 is filtered off 7 > 1
15:28:32,309 : Job user-0-27 is filtered off 5 > 1
15:28:32,309 : Job user-0-28 is filtered off 8 > 1
15:28:32,309 : Job user-1-29 is filtered off 4 > 1
15:28:32,310 : Job user-0-30 is filtered off 5 > 1
15:28:32,310 : Job user-0-31 is filtered off 9 > 1
15:28:32,310 : Job user-1-32 is filtered off 7 > 1
15:28:32,310 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,312 : Job user-1-3 is filtered off 8 > 7
15:28:32,312 : Job user-1-8 is filtered off 10 > 7
15:28:32,312 : Job user-0-11 is filtered off 10 > 7
15:28:32,312 : Job user-1-12 is filtered off 8 > 7
15:28:32,312 : Job user-1-17 is filtered off 9 > 7
15:28:32,313 : Job user-1-19 is filtered off 10 > 7
15:28:32,313 : Job user-0-21 is filtered off 8 > 7
15:28:32,313 : Job user-0-23 is filtered off 9 > 7
15:28:32,313 : Job user-0-28 is filtered off 8 > 7
15:28:32,313 : Job user-0-31 is filtered off 9 > 7
15:28:32,314 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,314 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,315 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,315 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,316 : Bid from user-1-18 lower than reservation price, 56 < 57
15:28:32,316 : Agent user-0-20 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,316 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,317 : Agent user-0-24 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,317 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,317 : Bid from user-1-26 lower than reservation price, 56 < 57
15:28:32,318 : Agent user-0-27 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,318 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,319 : Agent user-0-30 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,319 : Bid from user-1-32 lower than reservation price, 56 < 57
15:28:32,319 : Highest bid 57 by user-0-30, second highest 57 by user-0-27
15:28:32,320 : Got a tie with 6 agents
15:28:32,320 : Winner : agent user-0-20 bidding 57 and payes 57
15:28:32,322 : Job user-1-8 is filtered off 10 > 9
15:28:32,322 : Job user-0-11 is filtered off 10 > 9
15:28:32,323 : Job user-1-19 is filtered off 10 > 9
15:28:32,323 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,324 : Bid from user-1-3 lower than reservation price, 56 < 57
15:28:32,324 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,324 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,324 : Bid from user-1-12 lower than reservation price, 56 < 57
15:28:32,325 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,325 : Bid from user-1-17 lower than reservation price, 56 < 57
15:28:32,325 : Bid from user-1-18 lower than reservation price, 56 < 57
15:28:32,326 : Agent user-0-21 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,326 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,327 : Agent user-0-23 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,327 : Agent user-0-24 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,328 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,328 : Bid from user-1-26 lower than reservation price, 56 < 57
15:28:32,328 : Agent user-0-27 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,328 : Agent user-0-28 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,329 : Bid from user-1-29 lower than reservation price, 56 < 57
```

```
15:28:32,329 : Agent user-0-30 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,329 : Agent user-0-31 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,329 : Bid from user-1-32 lower than reservation price, 56 < 57
15:28:32,330 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,330 : Highest bid 57 by user-0-31, second highest 57 by user-0-30
15:28:32,330 : Got a tie with 9 agents
15:28:32,330 : Winner : agent user-0-30 bidding 57 and payes 57
15:28:32,333 : Job user-1-8 is filtered off 10 > 9
15:28:32,333 : Job user-0-11 is filtered off 10 > 9
15:28:32,333 : Job user-1-19 is filtered off 10 > 9
15:28:32,334 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,334 : Bid from user-1-3 lower than reservation price, 56 < 57
15:28:32,334 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,335 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,335 : Bid from user-1-12 lower than reservation price, 56 < 57
15:28:32,335 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,336 : Bid from user-1-17 lower than reservation price, 56 < 57
15:28:32,336 : Bid from user-1-18 lower than reservation price, 56 < 57
15:28:32,336 : Agent user-0-21 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,337 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,337 : Agent user-0-23 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,337 : Agent user-0-24 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,338 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,338 : Bid from user-1-26 lower than reservation price, 56 < 57
15:28:32,338 : Agent user-0-27 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,339 : Agent user-0-28 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,340 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,340 : Agent user-0-31 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,340 : Bid from user-1-32 lower than reservation price, 56 < 57
15:28:32,341 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,341 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,342 : Bid from user-1-35 lower than reservation price, 56 < 57
15:28:32,342 : Highest bid 57 by user-0-31, second highest 57 by user-0-28
15:28:32,342 : Got a tie with 8 agents
15:28:32,343 : Winner : agent user-0-31 bidding 57 and payes 57
15:28:32,352 : Job user-1-8 is filtered off 10 > 9
15:28:32,352 : Job user-0-11 is filtered off 10 > 9
15:28:32,353 : Job user-1-19 is filtered off 10 > 9
15:28:32,353 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,353 : Bid from user-1-3 lower than reservation price, 56 < 57
15:28:32,354 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,354 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,354 : Bid from user-1-12 lower than reservation price, 56 < 57
15:28:32,355 : Agent user-0-16 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,356 : Bid from user-1-17 lower than reservation price, 56 < 57
15:28:32,356 : Bid from user-1-18 lower than reservation price, 56 < 57
15:28:32,357 : Agent user-0-21 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,358 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,358 : Agent user-0-23 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,359 : Agent user-0-24 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,359 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,359 : Bid from user-1-26 lower than reservation price, 56 < 57
15:28:32,360 : Agent user-0-27 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,360 : Agent user-0-28 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,360 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,360 : Bid from user-1-32 lower than reservation price, 56 < 57
15:28:32,361 : Bid from user-1-33 lower than reservation price, 56 < 57
```

```
15:28:32,361 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,362 : Bid from user-1-35 lower than reservation price, 56 < 57
15:28:32,362 : Bid from user-1-36 lower than reservation price, 56 < 57
15:28:32,362 : Bid from user-1-37 lower than reservation price, 56 < 57
15:28:32,363 : Agent user-0-38 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,363 : Highest bid 57 by user-0-38, second highest 57 by user-0-28
15:28:32,364 : Got a tie with 8 agents
15:28:32,364 : Winner : agent user-0-24 bidding 57 and payes 57
15:28:32,366 : Job user-1-3 is filtered off 8 > 5
15:28:32,367 : Job user-1-8 is filtered off 10 > 5
15:28:32,367 : Job user-0-11 is filtered off 10 > 5
15:28:32,367 : Job user-1-12 is filtered off 8 > 5
15:28:32,367 : Job user-0-16 is filtered off 7 > 5
15:28:32,368 : Job user-1-17 is filtered off 9 > 5
15:28:32,368 : Job user-1-18 is filtered off 7 > 5
15:28:32,368 : Job user-1-19 is filtered off 10 > 5
15:28:32,368 : Job user-0-21 is filtered off 8 > 5
15:28:32,368 : Job user-0-23 is filtered off 9 > 5
15:28:32,369 : Job user-1-26 is filtered off 7 > 5
15:28:32,369 : Job user-0-28 is filtered off 8 > 5
15:28:32,369 : Job user-1-32 is filtered off 7 > 5
15:28:32,369 : Job user-1-35 is filtered off 8 > 5
15:28:32,369 : Job user-1-37 is filtered off 6 > 5
15:28:32,370 : Job user-0-38 is filtered off 8 > 5
15:28:32,370 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,370 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,370 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,371 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,371 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,371 : Agent user-0-27 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,372 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,372 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,372 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,372 : Bid from user-1-36 lower than reservation price, 56 < 57
15:28:32,373 : Highest bid 57 by user-0-27, second highest 57 by user-0-27
15:28:32,373 : Winner : agent user-0-27 bidding 57 and payes 57
15:28:32,376 : Job user-1-3 is filtered off 8 > 4
15:28:32,376 : Job user-1-8 is filtered off 10 > 4
15:28:32,376 : Job user-0-11 is filtered off 10 > 4
15:28:32,377 : Job user-1-12 is filtered off 8 > 4
15:28:32,377 : Job user-0-16 is filtered off 7 > 4
15:28:32,377 : Job user-1-17 is filtered off 9 > 4
15:28:32,378 : Job user-1-18 is filtered off 7 > 4
15:28:32,378 : Job user-1-19 is filtered off 10 > 4
15:28:32,378 : Job user-0-21 is filtered off 8 > 4
15:28:32,379 : Job user-0-23 is filtered off 9 > 4
15:28:32,379 : Job user-1-26 is filtered off 7 > 4
15:28:32,380 : Job user-0-28 is filtered off 8 > 4
15:28:32,380 : Job user-1-32 is filtered off 7 > 4
15:28:32,380 : Job user-1-34 is filtered off 5 > 4
15:28:32,380 : Job user-1-35 is filtered off 8 > 4
15:28:32,381 : Job user-1-36 is filtered off 5 > 4
15:28:32,381 : Job user-1-37 is filtered off 6 > 4
15:28:32,381 : Job user-0-38 is filtered off 8 > 4
15:28:32,381 : Job user-0-40 is filtered off 8 > 4
15:28:32,381 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,382 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,382 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,385 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,386 : Agent user-0-25 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,386 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,386 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,387 : Bid from user-1-39 lower than reservation price, 56 < 57
15:28:32,387 : Highest bid 57 by user-0-25, second highest 57 by user-0-25
15:28:32,388 : Winner : agent user-0-25 bidding 57 and payes 57
15:28:32,392 : Job user-1-1 is filtered off 2 > 1
15:28:32,392 : Job user-1-3 is filtered off 8 > 1
15:28:32,393 : Job user-1-7 is filtered off 2 > 1
15:28:32,393 : Job user-1-8 is filtered off 10 > 1
15:28:32,393 : Job user-1-9 is filtered off 2 > 1
```

```
15:28:32,393 : Job user-0-11 is filtered off 10 > 1
15:28:32,393 : Job user-1-12 is filtered off 8 > 1
15:28:32,393 : Job user-0-16 is filtered off 7 > 1
15:28:32,394 : Job user-1-17 is filtered off 9 > 1
15:28:32,394 : Job user-1-18 is filtered off 7 > 1
15:28:32,394 : Job user-1-19 is filtered off 10 > 1
15:28:32,394 : Job user-0-21 is filtered off 8 > 1
15:28:32,394 : Job user-0-23 is filtered off 9 > 1
15:28:32,395 : Job user-1-26 is filtered off 7 > 1
15:28:32,395 : Job user-0-28 is filtered off 8 > 1
15:28:32,395 : Job user-1-29 is filtered off 4 > 1
15:28:32,395 : Job user-1-32 is filtered off 7 > 1
15:28:32,396 : Job user-1-34 is filtered off 5 > 1
15:28:32,396 : Job user-1-35 is filtered off 8 > 1
15:28:32,396 : Job user-1-36 is filtered off 5 > 1
15:28:32,396 : Job user-1-37 is filtered off 6 > 1
15:28:32,396 : Job user-0-38 is filtered off 8 > 1
15:28:32,396 : Job user-1-39 is filtered off 4 > 1
15:28:32,397 : Job user-0-40 is filtered off 8 > 1
15:28:32,397 : Job user-0-41 is filtered off 9 > 1
15:28:32,397 : Job user-1-42 is filtered off 7 > 1
15:28:32,397 : Job user-0-43 is filtered off 5 > 1
15:28:32,397 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,398 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,398 : Job user-1-3 is filtered off 8 > 5
15:28:32,399 : Job user-1-8 is filtered off 10 > 5
15:28:32,399 : Job user-0-11 is filtered off 10 > 5
15:28:32,400 : Job user-1-12 is filtered off 8 > 5
15:28:32,400 : Job user-0-16 is filtered off 7 > 5
15:28:32,400 : Job user-1-17 is filtered off 9 > 5
15:28:32,400 : Job user-1-18 is filtered off 7 > 5
15:28:32,400 : Job user-1-19 is filtered off 10 > 5
15:28:32,401 : Job user-0-21 is filtered off 8 > 5
15:28:32,401 : Job user-0-23 is filtered off 9 > 5
15:28:32,401 : Job user-1-26 is filtered off 7 > 5
15:28:32,402 : Job user-0-28 is filtered off 8 > 5
15:28:32,402 : Job user-1-32 is filtered off 7 > 5
15:28:32,402 : Job user-1-35 is filtered off 8 > 5
15:28:32,402 : Job user-1-37 is filtered off 6 > 5
15:28:32,403 : Job user-0-38 is filtered off 8 > 5
15:28:32,403 : Job user-0-40 is filtered off 8 > 5
15:28:32,403 : Job user-0-41 is filtered off 9 > 5
15:28:32,403 : Job user-1-42 is filtered off 7 > 5
15:28:32,403 : Job user-0-44 is filtered off 7 > 5
15:28:32,404 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,404 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,405 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,405 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,405 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,406 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,406 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,406 : Bid from user-1-36 lower than reservation price, 56 < 57
15:28:32,407 : Bid from user-1-39 lower than reservation price, 56 < 57
15:28:32,407 : Agent user-0-43 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,407 : Highest bid 57 by user-0-43, second highest 57 by user-0-43
15:28:32,408 : Winner : agent user-0-43 bidding 57 and payes 57
15:28:32,410 : Job user-1-3 is filtered off 8 > 5
15:28:32,411 : Job user-1-8 is filtered off 10 > 5
15:28:32,411 : Job user-0-11 is filtered off 10 > 5
15:28:32,411 : Job user-1-12 is filtered off 8 > 5
15:28:32,412 : Job user-0-16 is filtered off 7 > 5
15:28:32,412 : Job user-1-17 is filtered off 9 > 5
15:28:32,412 : Job user-1-18 is filtered off 7 > 5
15:28:32,412 : Job user-1-19 is filtered off 10 > 5
15:28:32,412 : Job user-0-21 is filtered off 8 > 5
15:28:32,413 : Job user-0-23 is filtered off 9 > 5
15:28:32,413 : Job user-1-26 is filtered off 7 > 5
15:28:32,413 : Job user-0-28 is filtered off 8 > 5
15:28:32,413 : Job user-1-32 is filtered off 7 > 5
15:28:32,414 : Job user-1-35 is filtered off 8 > 5
15:28:32,414 : Job user-1-37 is filtered off 6 > 5
15:28:32,414 : Job user-0-38 is filtered off 8 > 5
15:28:32,414 : Job user-0-40 is filtered off 8 > 5
```

```
15:28:32,414 : Job user-0-41 is filtered off 9 > 5
15:28:32,415 : Job user-1-42 is filtered off 7 > 5
15:28:32,415 : Job user-0-44 is filtered off 7 > 5
15:28:32,416 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,416 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,416 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,417 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,417 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,417 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,418 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,418 : Bid from user-1-36 lower than reservation price, 56 < 57
15:28:32,418 : Bid from user-1-39 lower than reservation price, 56 < 57
15:28:32,419 : Job user-1-3 is filtered off 8 > 5
15:28:32,420 : Job user-1-8 is filtered off 10 > 5
15:28:32,420 : Job user-0-11 is filtered off 10 > 5
15:28:32,420 : Job user-1-12 is filtered off 8 > 5
15:28:32,420 : Job user-0-16 is filtered off 7 > 5
15:28:32,421 : Job user-1-17 is filtered off 9 > 5
15:28:32,421 : Job user-1-18 is filtered off 7 > 5
15:28:32,421 : Job user-1-19 is filtered off 10 > 5
15:28:32,421 : Job user-0-21 is filtered off 8 > 5
15:28:32,421 : Job user-0-23 is filtered off 9 > 5
15:28:32,422 : Job user-1-26 is filtered off 7 > 5
15:28:32,422 : Job user-0-28 is filtered off 8 > 5
15:28:32,422 : Job user-1-32 is filtered off 7 > 5
15:28:32,422 : Job user-1-35 is filtered off 8 > 5
15:28:32,423 : Job user-1-37 is filtered off 6 > 5
15:28:32,423 : Job user-0-38 is filtered off 8 > 5
15:28:32,423 : Job user-0-40 is filtered off 8 > 5
15:28:32,424 : Job user-0-41 is filtered off 9 > 5
15:28:32,424 : Job user-1-42 is filtered off 7 > 5
15:28:32,424 : Job user-0-44 is filtered off 7 > 5
15:28:32,424 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,425 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,425 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,426 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,426 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,426 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,427 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,427 : Bid from user-1-36 lower than reservation price, 56 < 57
15:28:32,428 : Bid from user-1-39 lower than reservation price, 56 < 57
15:28:32,428 : Agent user-0-45 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,429 : Highest bid 57 by user-0-45, second highest 57 by user-0-45
15:28:32,429 : Winner : agent user-0-45 bidding 57 and payes 57
15:28:32,433 : Job user-1-3 is filtered off 8 > 6
15:28:32,433 : Job user-1-8 is filtered off 10 > 6
15:28:32,434 : Job user-0-11 is filtered off 10 > 6
15:28:32,434 : Job user-1-12 is filtered off 8 > 6
15:28:32,434 : Job user-0-16 is filtered off 7 > 6
15:28:32,434 : Job user-1-17 is filtered off 9 > 6
15:28:32,434 : Job user-1-18 is filtered off 7 > 6
15:28:32,435 : Job user-1-19 is filtered off 10 > 6
15:28:32,435 : Job user-0-21 is filtered off 8 > 6
15:28:32,435 : Job user-0-23 is filtered off 9 > 6
15:28:32,436 : Job user-1-26 is filtered off 7 > 6
15:28:32,436 : Job user-0-28 is filtered off 8 > 6
15:28:32,436 : Job user-1-32 is filtered off 7 > 6
15:28:32,436 : Job user-1-35 is filtered off 8 > 6
15:28:32,436 : Job user-0-38 is filtered off 8 > 6
15:28:32,437 : Job user-0-40 is filtered off 8 > 6
15:28:32,437 : Job user-0-41 is filtered off 9 > 6
15:28:32,437 : Job user-1-42 is filtered off 7 > 6
15:28:32,437 : Job user-0-44 is filtered off 7 > 6
15:28:32,437 : Job user-0-49 is filtered off 8 > 6
15:28:32,438 : Bid from user-1-1 lower than reservation price, 56 < 57
15:28:32,438 : Bid from user-1-7 lower than reservation price, 56 < 57
15:28:32,439 : Bid from user-1-9 lower than reservation price, 56 < 57
15:28:32,439 : Bid from user-1-22 lower than reservation price, 56 < 57
15:28:32,439 : Bid from user-1-29 lower than reservation price, 56 < 57
15:28:32,440 : Bid from user-1-33 lower than reservation price, 56 < 57
15:28:32,440 : Bid from user-1-34 lower than reservation price, 56 < 57
15:28:32,441 : Bid from user-1-36 lower than reservation price, 56 < 57
15:28:32,441 : Bid from user-1-37 lower than reservation price, 56 < 57
```

```
15:28:32,441 : Bid from user-1-39 lower than reservation price, 56 < 57
15:28:32,442 : Bid from user-1-46 lower than reservation price, 56 < 57
15:28:32,442 : Bid from user-1-47 lower than reservation price, 56 < 57
15:28:32,442 : Agent user-0-48 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,442 : Agent user-0-50 bids 57 on resource resource-0 with
    reservation price 57
15:28:32,443 : Highest bid 57 by user-0-50, second highest 57 by user-0-50
15:28:32,443 : Winner : agent user-0-50 bidding 57 and payes 57
```

E Price Update error

There seems to be a error in the source-code at this point. The following is from *server/Scheduler.py* at line 657 which is executed when the resources do not obtain a job.

```
# Didn't get a job so price should fall
res_dict["LOAD_MULTIPLY"] -= self.multiply_delta
res_dict["LOAD_MULTIPLY"] = max(mult, 1.0)
```

As can bee seen LOAD_MULTIPLY is first decremented and then overwritten by the maximum of either 1 or the value of a variable *mult*. I think the source-code should have looked something like this

```
# Didn't get a job so price should fall
mult -= self.multiply_delta
res_dict["LOAD_MULTIPLY"] = max(mult, 1.0)
```

In this way you will get the effect of the decrement if *load_ multiply* is above 1 as seems to be the intention.