Mechanism for Robust Procurements[1]

Yingqian Zhang [a] Sicco Verwer [b]

[a] *Erasmus University Rotterdam, The Netherlands*
[b] *Radboud University Nijmegen, The Netherlands*

The increasing popularity of adopting auctions is largely due to its efficiency of allocating goods. However, in the face of uncertainties on services, the winner determination solution is often not robust enough to ensure a reliable outcome. This paper aims to design a more robust auction by introducing redundancy into the selected solution. More specifically, we construct an algorithm and a mechanism for incentivizing truth-telling in public procurement problems with uncertainties. Our contributions are the development of a framework for studying such procurement problems, proving that minimizing cost in this framework is NP-complete, developing a quick algorithm that minimizes this cost, and providing a novel multi-stage mechanism that has desirable properties such as efficient, truthful in dominant strategies, and post-execution individually rational. We show experimentally that our approach significantly outperforms the current practice in many settings.

Robust procurement problem (RPP) The procurer announces a job with a deadline D and a minimal completion probability γ. Each bidder $i \in A$ submits a bid that consists of $< c_i, d_i, \beta_i >$, where c_i is the cost of executing the job, d_i is the duration of completing the job, and β_i specifies the reservation fee. We assume the auctioneer has the information about the reliability $r_i \in [0, 1]$ of each participant bidder i. Given the set of bids and the reliability of agents, the procurer determines a set of winners $S = (A_{i_1}, ..., A_{i_m}) \subseteq A$ as the outcome ϕ. S is a total ordered set. The probability that an ordered set of contractors $S = (A_{i_1}, ..., A_{i_m})$ will finish the project within the deadline $\sum_{k=1}^{m} d_{i_k}$ is equal to $1 - \prod_{k=1}^{m}(1 - r_{i_k})$. The expected cost incurred by S then becomes: $E[Cost(S)] = \sum_{k=1}^{m}(c_{i_k} + \beta_{i_{k+1}}) \prod_{l=0}^{k-1}(1 - r_{i_l})$, where $r_{i_0} = 0$ and $\beta_{i_{m+1}} = 0$. Denote by $\mathbb{S}$ the possible ordered sets of contractors that may finish the project within deadline D with probability at least γ, that is: $\mathbb{S} = \{(A_{i_1}, ..., A_{i_m}) : \sum_{k=1}^{S} d_{i_k} \leq D \text{ } and \text{ } \prod_{k=1}^{m}(1 - r_{i_k}) \leq 1 - \gamma\}$. The robust procurement problem (RPP) can be now defined as the following constrained optimization problem: $\min_{S \in \mathbb{S}} E[Cost(S)]$.

We consider a setting where agents are self-interested and their declarations are private information. Let this so-called *type* of each agent i be denoted by θ_i. We use θ_{-i} to denote the type profile without the type of agent i. Given a type profile, a direct-revelation mechanism selects an *outcome* $\phi = f(\theta)$ using an algorithm f from the set of possible outcomes, and a payment $\bar{p}_i(\phi, \theta)$ for each agent that together define the utility of an agent $\bar{u}_i(\phi, \theta) = \bar{v}_i(\phi, \theta_i) - \bar{p}_i(\phi, \theta)$. $\bar{v}_i(\phi, \theta_i)$ specifies the expected valuation of agent i on the outcome ϕ. Let $S_\phi = (A_{a_1}, ..., A_{a_m})$ denote the ordered set given the outcome ϕ. The expected valuation of agent i (i.e. A_{a_i}) prior to execution is computed as: $\bar{v}_i(\phi, \theta_i) = -c_i \prod_{l=0}^{i-1}(1 - r_l) - \beta_i \prod_{l=0}^{i-2}(1 - r_l)$. Note the realized valuation $v_i(\phi, \theta_i)$ of a contractor A_{a_i} after execution of the procurement schedule depends on the actual execution. The expected *social welfare* is defined as: $\bar{w}(\phi) = \sum_{i=1}^{m} \bar{v}_i(\phi, \theta_i)$. The mechanism is called *efficient* when it selects outcomes where the expected social welfare is highest. In addition, we are interested in mechanisms that are truthful in dominant strategies and *post-execution* individually rational, i.e., a truthful agent will not receive a negative utility no matter what the actual execution outcome will be.

Complexity and algorithm for RPP We show that the robust procurement problem RPP is computationally hard by showing that the subproblem (i.e., the RPP without costs problem) of finding a suitable set of contractors is NP-complete. The proof is by reduction from the subset sum problem. Although this means that it is difficult to optimize in theory, in practice an optimal solution can often be found efficiently because the number of contractors with a total duration less than the job deadline is typically small. Furthermore,

[1]The full version of the paper will appear in the Proceedings of PRIMA 2012: Principles and Practice of Multi-Agent Systems.

the number of contractors required to reach the desired reliability is usually limited. The total search space is therefore typically small, and even a brute-force search should often be able to find the optimal solution within reasonable time. We propose a backtracking algorithm that iteratively appends one contractor to a partially constructed solution. The algorithm uses bounds to further reduce the search space: (1) Adding a contractor to the end of an ordering always increases the expected cost of the ordering. (2) Adding a contractor always increases the total duration of an ordering.

Mechanism for RPP One possible truthful mechanism is the VCG mechanism. However, it has been shown in [1] that VCG is individual rational in expectation only, i.e., an agent may get a negative (realized) utility after the execution of the procurement outcome. In this paper, we define a multi-stage Grove mechanism and show that it is efficient, truthful in dominant strategies, and post-execution individually rational. The proposed mechanism, called RobustProcurement, works as follows:
The auctioneer announces a job with deadline and completion probability threshold γ.

1. The contractor agents declare their types $\theta = (\theta_i, \theta_{-i})$ to the auctioneer.
2. The auctioneer then finds an optimal schedule ϕ using the proposed algorithm.
3. Every agent i in the ordered set $S_\phi = (A_{a_1}, ..., A_{a_m})$ receives its marginal contribution as payment: $p_i(\phi, \theta_i) = -\bar{w}(\phi, \theta) + \bar{w}_{-i}(\phi', \theta_{-i})$, where $\bar{w}(\phi, \theta)$ is the expected social welfare, $\phi' = f(\theta_{-i})$ is the efficient outcome without agent i's participation, and $\bar{w}_{-i}(\phi', \theta)$ is the social welfare on ϕ'.
4. The first winner A_{a_1} receives an additional payment: $p'_i(\phi, \theta_i) = v_i(\phi, \theta_i)$. A_{a_1} starts to execute the job. The second winner A_{a_2} is notified to be stand-by. The reservation cost is incurred.
5. At the deadline of agent $A_{a_{i-1}}$ for $2 \le i \le |\mathcal{S}| - 1$, do
   - Transfer the additional payment to agent i (i.e. agent A_{a_i}), where the agent's realized valuation $v_i(\phi, \theta_i)$ is computed based on the realization of the schedule ϕ, i.e.,

     $$v_i(\phi, \theta_i) = \begin{cases} -c_i - \beta_i & \text{if } A_{a_{i-1}} \text{ does not complete the job by its deadline;} \\ -\beta_i & \text{otherwise.} \end{cases}$$
   - If $A_{a_{i-1}}$ completes the job, inform the remaining agents $j \in S$, and terminate the mechanism. Otherwise, agent A_{a_i} starts to execute the job, and agent $A_{a_{i+1}}$ is notified to be stand-by.
   - Loop step 5 until the mechanism is terminated.

Experiments We investigate the performance of our mechanism and compare it to a greedy mechanism that represents the current practice in procurement: create a first-price auction, select a winning contractor, and create a new auction if the contractor fails. The mechanism iteratively selects a contractor. The performance measurements are: (1) the expected social welfare (i.e., the total cost incurred by the contractors) and (2) the expected payments of the auctioneer. We create many different sets of agents for a given combination of deadlines and reliability thresholds, and the number of agents. For every combination, we generate 50 problem instances. Every instance is solved using our robust procurement algorithm and compared with the result of running the aforementioned greedy procedure using three different costs for every iteration after the first. We compute a greedy solution first for 0 re-iteration costs, then one for 1 time the average reservation costs of the agents, and finally 2 times the average reservation costs.

The results show that in terms of social welfare, our mechanism outperforms the greedy approach in all cases except when there exist cheap and reliable agents who can finish the job in time. In terms of payments, our mechanism outperforms the current practice when there are many potential contractors and constraints in the optimization problem are tight. The results are promising especially considering that in the experiments, the potential cost increase due to misreporting of the agents is disregarded in the greedy mechanism.

References

[1] Sebastian Stein, Enrico H. Gerding, Alex Rogers, Kate Larson, and Nicholas R. Jennings. Algorithms and mechanisms for procuring services with uncertain durations using redundancy. *Artif. Intell.*, 175(14-15):2021–2060, 2011.