Blind Multi-Stage Scoring Auctions with Two-Sided Incomplete Information for Government Procurement

Jad Soucar[1]

[1]Department of Industrial and Systems Engineering, University of Southern California, CA 90089, USA

1 Introduction

Competitive bidding is the dominant mechanism through which the government allocates billions of dollars worth of assets every year. The approach is rooted in basic auction theory which would imply that a competitive bidding process can help reduce costs and guarantee the government favorable terms. In the case of government procurement the types of auctions a government agency can employ are strictly codified and regulated, however there are certain auction styles employed by governments where subjective scoring still creep in.

For example consider the multi-round *best-value* auction style frequently used in public works bidding. Under this format a panel of "reviewers" will receive n bids from n bidders. This panel then assigns scores across multiple criteria and combines them to form a total score which is returned to the bidder at the end of a round [1]. Such criteria can often times be difficult to quantify. For example under FAR Part 15 [2], an agency may conduct a *best-value* auction where bidders submit proposals evaluated on technical merit and past performance. In other words the panel has a multi-attribute value function $V(x) : \mathbb{R}^d \to [0, 100]$ which scores a d attribute bid x and returns a normalized score between 0 and 100. This process continues iteratively with bidders submitting bids and receiving scores during each round until a terminal round where the bidder with the highest score is awarded the government contract.

We argue that this reliance on qualitative evaluation introduces significant uncertainty into the bidding process. Additionally that uncertainty is shaped less by economic fundamentals and more by the personal preferences of the auctioneer. Biases within this context have been well documented and effect both the bidder and the auctioneer.For example, biases like Lower-Bid Bias where a government agency will tend towards awarding a contract to the lowest bid despite the risk of reduced quality, are commonplace according to an experimental study conducted by Dekel and Schurr. [3].

Due to panel biases and parameters in the panel's value function that are unknown to the bidder, a bidder is forced to roughly estimate the value function $V(x)$ within a few rounds of bidding in order to efficiently apply inter-round adjustments to their bids. We refer to that problem as "few-shot value function estimation". We recognize that this problem is more general then government procurement. And so we first focus our attention on developing a general algorithm to estimate a value function with only a few (input,output) pairs. Next we elicit a value function from an expert in the field of public works construction bidding. Finally we simulate a multi-round bidding process, where a bidder uses our proposed algorithm to quickly estimate the elicited value function within a few rounds of simulated bidding.

2 Few Shot Value Function Estimation

To accomplish few-shot value function estimation, we first establish a few definitions in order to formalize the problem. First we assume that the true value function $V : \mathbb{R}^d \to [0, 100]$ is additive over d single attribute value functions $v_i : \mathbb{R} \to [0, 100]$, each of which is weighted based on importance. As such the value function is designed to attribute a value to a d dimensional input x.

$$V(x) = \sum_{i=1}^{d} w_i \cdot v_i(x_i), \quad x \in \mathbb{R}^d, w \in \Delta_d$$

$$\Delta_d = \left\{ \sum_{i=1}^{d} w_i = 1, w \geq 0 \right\}$$

At this stage, its reasonable to assume two features about our value function. First that x is in a bounded and closed subspace of $\mathbb{R}^d$. This is an intuitive simplifying assumption since in most practical cases an input x with infinite value represents an agent with infinite resources which is practically unreasonable. We formalize the feasible region as follows, where β_i^- and β_i^+ are respectively the lower and upper bound of each attribute dimension i.

$$x \in \Pi_{i=1}^{d} [\beta_i^-, \beta_i^+] \tag{1}$$

We can simplify our feasible region further by adopting the following normalization function and shrinking our box shaped feasible region to $[0, 1]^d$.

$$f : [\boldsymbol{\beta}^-, \boldsymbol{\beta}^+] \to [0, 1]^d, \quad f(x)_i = \frac{x_i - \beta_i^-}{\beta_i^+ - \beta_i^-}, \quad \forall i = 1, \ldots, d \tag{2}$$

On the other hand if the value function is categorical with n ordered categories like $x_i = \{\text{Good,Medium,Bad}\}$ then we assume that the normalized numerical encoding of the single attribute value function domain is $\{\frac{i}{n-1} : i = 1, ..., n-1\}$. For example $\text{Good} = 1, \text{Medium} = 0.5, \text{Bad} = 0$. From here on out we assume that $x \in [0, 1]^d$ and that there exists an invertible function $f(x)$ whose inverse returns the nominal value of the input x. Next we make the simplifying assumption that $v_i(x_i)$ falls within a finite set of parametric functions. For example $v_i(y)$ may be exponential, linear, triangular, etc.

$$v_i(x) = \alpha_i^0 x + \alpha_i^1 \tag{3}$$

$$v_i(x) = \alpha_i^0 e^{\alpha_i^1 x} \tag{4}$$

$$v_i(x) = \begin{cases} \alpha_i^0 x + \alpha_i^1 & x \leq \alpha_i^5 \\ \alpha_i^3 x + \alpha_i^4 & \text{o.w} \end{cases} \tag{5}$$

We denote these prototype single attribute value function forms as $\phi_k(y;\alpha_k) : [0,1] \times \mathbb{R}^{|\alpha_i|} \rightarrow [0,100]$ where α_i is a vector of parameters necessary to define the function $\phi_k(x_i)$. We also define a "library" of preselected prototype functions of size M.

$$\mathcal{F} = \{\phi_k(y;\alpha_k)\}_{k=1}^{M} \tag{6}$$

Now that we've defined the value function V and "prototype" library $\mathcal{F}$ we can define an approximation of V.

$$\tilde{V}(x) = \sum_{i=1}^{d} w_i \sum_{k=1}^{M} z_k^i \phi_k(x_i;\alpha_k^i) \tag{7}$$

$$\text{s.t} \quad z_k^i \in \{0,1\}, \quad \sum_{k=1}^{M} z_k^i = 1 \quad \forall i \in \{1,\dots,d\}, \quad w \in \Delta_d \tag{8}$$

In other words we've defined an approximation which must pick one basis function out of a library of size M to serve as the single attribute value function for each of the d dimensions of x, while still ensuring that the weights w are unit.

We now transition to the question of finding the best approximation given only a few data points. In other words given a history of inputs and returned values $\mathcal{H}^t = \{(x^s, V(x^s)) : s = 1,...,t\}$ where t is small, we would like to find an parameters $\{z_k^i, \alpha_k^i, w_i : k = 1,\dots M, i = 1,...d\}$ such that our value approximation $\tilde{V}(x)$ is close to the true value function $V(x)$. To this end we define the following mixed integer program, which may be non-linear depending on the library of prototypes $\mathcal{F}$ that we choose.

$$\min_{\{z_k^i,\alpha_k^i\}_{i,k=1}^{d,M},\{w_i\}_{i=1}^{d}} \left[\frac{1}{t}\sum_{s=1}^{t}(V(x^s) - \tilde{V(x^s)})^2\right] \tag{9}$$

$$\text{s.t} \quad z_k^i \in \{0,1\}, \quad \sum_{k=1}^{M} z_k^i = 1 \quad \forall i \in \{1,\dots,d\}, \quad w \in \Delta_d \tag{10}$$

With the problem of few-shot value function estimation clearly formalized, we can begin exploring a solution method.

2.1 Online Greedy Solver

The minimization problem detailed in equation 10 is a combinatorial optimization program that, depending on the basis functions chosen, may be non-linear and non-convex. As a result the problem is computationally expensive to solve. So we investigate a greedy solver that splits the problem at each iteration into $d \times M$ subproblems and a master problem.

Assume that we are given a history of $\mathcal{H}^t = \{(x^s, V(x^s)) : s = 1,...,t\}$. We begin by fixing $w_i = \frac{1}{d}$ for all $i = 1,\dots,d$, then solve the following optimization problem for each dimension d. This problem represents a greedy solution since it aims to choose basis functions that can best explain the the entirety of the observed values for each input dimension.

$$\phi_{k_i^*}(y;\alpha_k^{i*}) = \arg\min_{\phi_k\in\mathcal{F}} \min_{\alpha_k^i} \sum_{(x^s,V(x^s))\in\mathcal{H}^t} (V(x^s) - w_i\phi_k(x_i^s;\alpha_k^i))^2 \tag{11}$$

Once we've obtained a set of optimal basis functions and their respective optimal parameters for each input dimension, $\{\phi_{k_i^*}(y;\alpha_k^{i*}) : i = 1,\ldots,d\}$, we can then solve the master problem which retrieves the optimal weights. Namely we solve

$$w^* = \arg\min_{w\in\Delta_d} \sum_{(x^s,V(x^s))\in\mathcal{H}^t} \left(V(x^s) - \sum_{i=1}^{d} w_i\phi_{k_i^*}(x_i^s;\alpha_k^{i*})\right)^2 \tag{12}$$

Once new data is introduced and we obtain an expanded/richer $\mathcal{H}^{t+1}$ we recompute this process, except instead of initializing $w_i = \frac{1}{d}$ while we solve the next d sub-problems, we allow w to equal w^* from the last iteration. Given that at each iteration we are given new data $(x^{t+1}, V(x^{t+1}))$ the full greedy solution algorithm can be described compactly as follows.

Algorithm 1 Online Greedy Solver

1: **input:** T−Number of Iterations
2: **input:** $\mathcal{F}$−Library of Prototypes
3: **input:** $V(x) : [0,1]^d \to [0,100]$
4: **set:** $w^1 = [\frac{1}{d},\ldots,\frac{1}{d}] \in \mathbb{R}^d$
5: **set:** $\mathcal{H}^1 = \{(x^1, V(x^1))\}$
6: **for** $t = 1$ to T **do**
7:     **for** $i = 1$ to d **do**
8:         **for** $\phi_k \in \mathcal{F}$ **do**
9:             $\ell_i[\phi_k(y;\alpha_k^{i*})] \leftarrow \min_{\alpha_k^i} \sum_{(x^s,V(x^s))\in\mathcal{H}^t} \left(V(x^s) - w_i^t\phi_k(x_i^s;\alpha_k^i)\right)^2$
10:         **end for**
11:         $\phi_{k_i^*}(y;\alpha_k^{i*}) \leftarrow \arg\min_{\phi_k\in\mathcal{F}} \ell_i[\phi_k(y;\alpha_k^{i*})]$
12:     **end for**
13:     $w^* = \arg\min_{w\in\Delta_d} \sum_{(x^s,V(x^s))\in\mathcal{H}^t} \left(V(x^s) - \sum_{i=1}^d w_i\phi_{k_i^*}(x_i^s;\alpha_k^{i*})\right)^2$
14:     $w^{t+1} \leftarrow w^*$
15:     $\mathcal{H}^{t+1} \leftarrow \mathcal{H}^t \cup \{(x^{t+1}, V(x^{t+1}))\}$
16: **end for**

$$\min_{\alpha_k^i} \sum_{(x^s,V(x^s))\in\mathcal{H}^t} \left(V(x^s) - w_i^t\phi_k(x_i^s;\alpha_k^i)\right)^2 \tag{13}$$

We note that for convex ϕ_k the sub-sub-problem denoted above and on line 9 of algorithm 1 reduces to a convex problem which is especially simply to solve. Take for example the prototype $\phi_k(y;\alpha) = \alpha^0 x + \alpha^1$, where the problem reduces to a least squares problem.

$$\min_{\alpha \in \mathbb{R}^2} \sum_{(x^s, V(x^s)) \in \mathcal{H}^t} \left(V(x^s) - w_i^t \phi_k(x_i^s; \alpha) \right)^2 \tag{14}$$

$$= \min_{\alpha_k^i} \|v - A\theta\|_2^2, \quad A = w_i^t \cdot \begin{bmatrix} x_i^1 & 1 \\ \vdots & \vdots \\ x_i^t & 1 \end{bmatrix} \in \mathbb{R}^{t \times 2}, v = \begin{bmatrix} V(x^1) \\ \vdots \\ V(x^t) \end{bmatrix} \in \mathbb{R}^t, \quad \theta = \begin{bmatrix} \alpha^0 & \alpha^1 \end{bmatrix} \tag{15}$$

$$\implies \theta^* = \frac{1}{w_i^t} (A^T A)^{-1} A^T v \tag{16}$$

However For a non-convex ϕ_k the sub-sub-problem denoted on line 9 of algorithm 1 may be difficult to solve. This problem arises even with simple prototypes such as $\phi_k = \min\{\alpha^0 x + \alpha^1, \alpha^2 x + \alpha^3\}$. A case like this isn't inconceivable. Take for example an agent who wishes to guess a target x_0 and who's value of a guess x diminishes as they move away from the target in either direction. In such a case one may consider an approach that leverages a multi-layer perceptron. Specifically given training data $\mathcal{H}^t$ we minimize the following loss function using stochastic gradient descent or ADAM [4].

$$\mathcal{L}(\alpha) = \frac{1}{t} \sum_{s=1}^{t} (V(x^s) - \phi_k(x_i^s))^2 \tag{17}$$

$$\phi_k(y; \alpha) = \begin{cases} \alpha^0 x + \alpha^1 & y \leq \alpha^4 \\ \alpha^2 x + \alpha^3 & y > \alpha^4 \end{cases} \tag{18}$$

Finally the minimization on line 11 of algorithm 1 is a simple minimization over a finite set, whereas the master problem on line 13 of algorithm 1 reduces to a constrained quadratic program which can be solved simply using a number of techniques including projected gradient descent. To formulate the master problem we let $Z \in \mathbb{R}^{t \times d}$ be the matrix with entries $[\phi_{k_i^*}(x_i^s; \alpha_k^{i*})]_{s,i=1}^{t,d}$ and $v \in \mathbb{R}^t$ be vector with entries $V(x^s)$. Then the problem becomes

$$\min_{w \in \Delta_d} \frac{1}{2} w^T Z^T Z w - w^T Z v \tag{19}$$

We conclude by noting that the algorithm is tractable but requires solving $T \cdot [M \times d + 1]$ optimization problems, which may become costly at scale. In the following section we describe an application of the greedy online solver to a case study in government procurement.

3 Case Study: Public Works Procurement

Our goal will be to apply the online greedy solver to simulate several rounds of bidding within a public works procurement procedure. Namely a government agency will set up a multi-round best value auction, where bidders can bid to win a construction contract. Within this auction format the bidders submit bids and receive a score corresponding with their bid during every round of bidding. Throughout the rounds each bidder can adjust their bid based on the score received in order to improve their chance of winning the contract. In order to simulate such an auction we must first elicit a value function from an expert in the field of public works procurement. This value function will represent the government's scoring panel. Then we can apply our greedy online solver to simulate a bidders bidding behavior.

3.1 Expert Elicitation

To apply our model to the specific case of public works bidding, we first elicit a value function from an expert in the field through the SMART method [5]. We chose to interview a construction manager whose been working in the industry for approximately 40 years [6]. We first interviewed the expert and elicited an objective hierarchy. The expert outlined the objectives detailed in table 1.

Attribute	**Metric**
Safety Score	EMR, OSHA incident rate
Personnel Experience	Avg. years of experience, certifications
Number of Similar Projects	Projects completed of comparable size and complexity
On-Time Completion History	Average completion time (% over or under planned time)
Project Manager Procedure	Quality of PM plan, tools used
Cost of Project Bid	Total bid estimate
Projected Duration	% over or under requested completion time

Table 1: Construction Bid Evaluation: Metrics for Each Attribute

Next we applied the swing weight procedure to determine the importance ranking and the weights of each objective within the value function. These results are detailed in table 2. Swing weights are usually considered to be the most accurate method for deriving attribute weights [7].

Attribute	Swing Rank	Swing Rating	Swing Weight
Cost	1	100	0.1812
Number of Similar Projects	2	90	0.1630
Personnel Experience	3	87	0.1576
Safety Score	4	85	0.1540
Projected Duration	5	70	0.1268
Project Manager Procedure	6	65	0.1178
On-Time Completion History	7	55	0.0996

Table 2: Swing Weighting for Construction Bid Attributes

Next we elicit the value function itself by applying the bisection method for all natural attributes and ranking for constructed attributes. We outline those finding in table 3. We note that cost has a triangular value function which is linear between $[0, 0.9]$ and $[1, 1.25]$. This structure is intuitive since, if a bidder has an expected cost of 0 a reviewer will naturally be skeptical of such a bid and assign a value of 0. On the flip side, a bid the 25% greater then the government's explicit budget will also be assigned a value of 0 since such a cost would be a disqualifying feature of the bid. We plot the value functions of each attribute in figure 1, where each value function is a linear interpolation of the elicited points in table 3.

Attribute	Point 1 (Value)	Point 2 (Value)	Point 3 (Value)	Point 4 (Value)
Safety Score	$\leq$ 1.0 (100)	2.0 (50)	3.0 (25)	$\geq$ 5.0 (0)
Personnel Experience	Excellent (100)	Acceptable (50)	Below Standard (10)	Unacceptable (0)
# of Similar Projects	0 (0)	3 (30)	7 (70)	$\geq$ 10 (100)
On-Time completion history	0% over (100)	25% over (75)	50% over (50)	100% over (0)
PM Procedure	Excellent (100)	Acceptable (60)	Below Standard (20)	Unacceptable (0)
Projected Duration	20% under (100)	10% under (75)	On target (50)	20% over (0)
Cost	$0\times$ (0)	$.9\times$ (100)	$1\times$ (90)	$\geq 1.25\times$ (0)

Table 3: Representative Value Function Points for Each Attribute

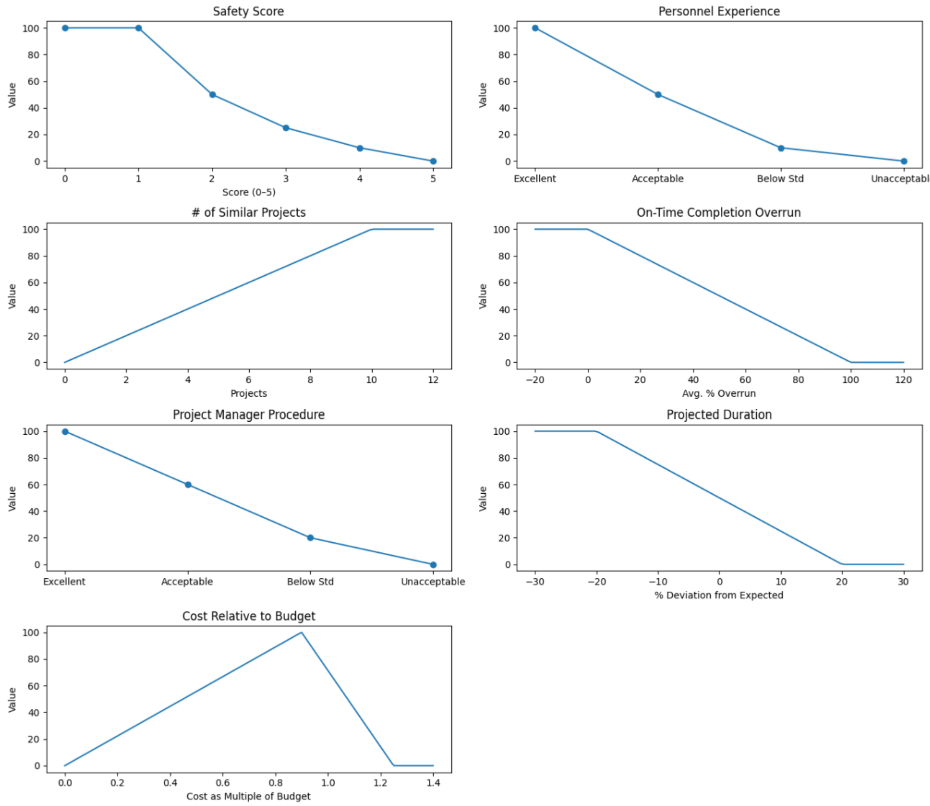

Figure 1: Value Functions for Each Attribute

We end our description of the value function elicitation interview by noting that a multi-stage scoring auctions have proven to be effective within government settings. For example Gao et.al found that scored bidding procedures, when adopted at the federal level, saved tax payers over 100 billion dollars in 2016 across several agencies [8].

3.2 Bidding Simulation

In order to apply the greedy online solver described in 1 to simulate an auction we first define a library of single-attribute value functions $\phi_k(x) : [0, 1] \to [0, 100]$. We choose the following library $\mathcal{F}$, which captures positive and negative linear/exponential and "tent"-shaped functions.

$$\mathcal{F} = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\} \tag{20}$$

$$\phi_1(x;\alpha_1) = 100x \tag{21}$$

$$\phi_1(x;\alpha_2) = -100x + 100 \tag{22}$$

$$\phi_3(x;\alpha_3) = \frac{e^{\alpha_3 x} - 1}{e^{\alpha_3} - 1} \tag{23}$$

$$\phi_4(x;\alpha_4) = 100 \cdot \frac{2 - (e^{\alpha_4 x})}{e^{\alpha_4} - 1} \tag{24}$$

$$\phi_5(x;\alpha_5) = \begin{cases} \frac{100x}{\alpha_5}, & \text{if } x \leq \alpha_5 \\ \\ 100 - \frac{100(x-\alpha_5)}{1-\alpha_5}, & x > \alpha_5 \end{cases} \tag{25}$$

We must also formally define the value function elicited in section 3.1. Note that there are 7 objectives outlined in table 1 and therefore we have 7 corresponding single attribute value functions which we denote to be v_i. v_i is defined to be the linear interpolation of the points in table 3, however we normalize all inputs to be in $[0,1]$ as described in equation 2.

$$V(x) = \sum_{i=1}^{7} w_i v_i(x_i) \tag{26}$$

To begin our simulation we allow the fictitious bidder to begin with a random bid $x^0 \in \text{Uniform}([0,1]^d)$. Then at the beginning of every round the bidder will maximize their current approximation of the auctioneer's value function $\tilde{V}(x)$ and choose their next bid as $x^{t+1} \leftarrow \arg\max_{x\in[0,1]^d} \tilde{V}(x)$. This is inline with a competitive bidding environment where T is unknown to the bidders and so each bidder must submit their best bid at every round in order to hedge against the current round being their last. Bellow are the results of how well each single attribute value function was estimated 2, along with the estimated weight of each single attribute objective 3. We also provide the correlation between the approximated value of an arbitrary bid and the true value of that bid 4. The correlation between the the outputs of the estimated value function and the true value function are observed to be $\rho = .989$, which indicates that $\tilde{V}(x) \approx V(x)$.

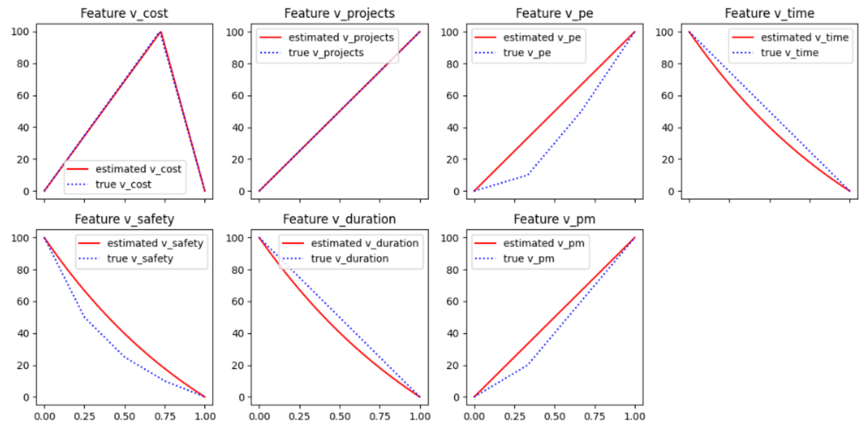

Figure 2: Approximated single attribute value functions after $T = 20$ rounds

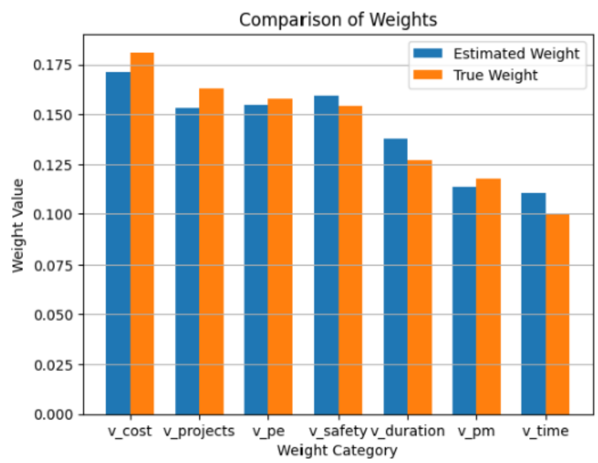

Figure 3: Approximated weights for each objective after $T = 20$

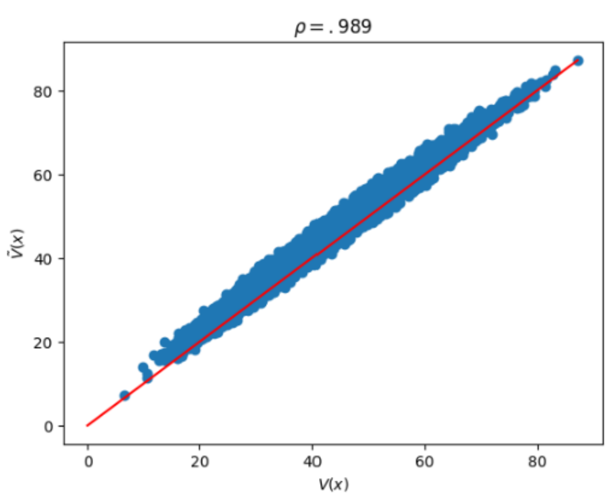

Figure 4: Correlations between $\tilde{V}(x)$ and $V(x)$

Bellow we provide the score achieved at every round of the bidding simulation 5. The online greedy value function estimator is able to find a high value bid along with a rough estimate of the auctioneer's value function structure, within a few rounds. This would imply that the online greedy solver is able to find high value bids faster then the structure of the value function. However because of the problem's non-convexity, the algorithm has a tendency to plateau for several rounds. Meaning that a bidder is struggling to pin down the exact value function structure and as a result is incapable of producing a bid of higher value then its last submission.

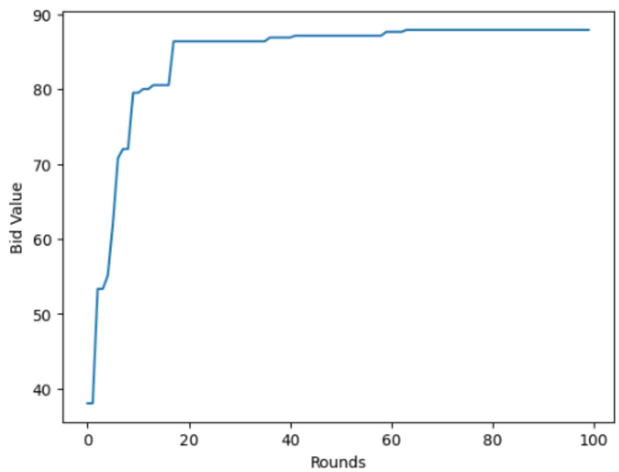

Figure 5: Value of Best Bid at round $t = 1, \ldots, 100$

We note that if the bidder is familiar with the structure of the auctioneer's value function, then the problem reduces to finding the weights associated with each attribute. This setting would correspond with an unbiased auctioneer who releases the exact scoring criteria for each objective to bidders. With the only uncertainty being the auctioneers preference in weighting the objectives. In this case finding the weights requires solving the same optimization problem outlined in equation 19. Bellow we show that even after 1 round of bidding $\|w^* - w\|_2^2 = .07$ and after 6 round $w^* = w$, where w^* are the true weights and w are the estimated weights 6.

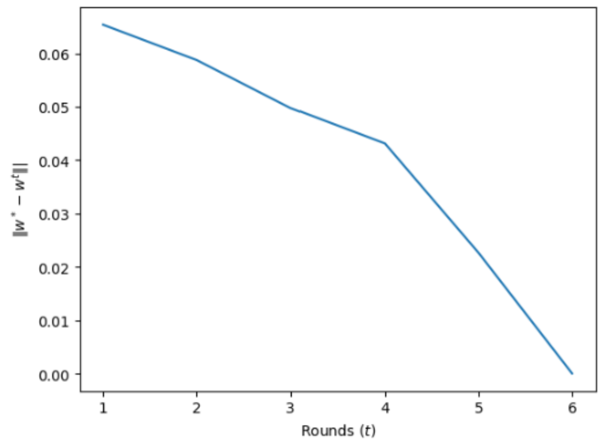

Figure 6: Error of estimated weights after $t = 1, \ldots, 6$ when the ϕ_k's are known for each objective.

4 Conclusion

Throughout this short study, we explored the problem of blind multi-stage scoring auctions with two-sided incomplete information. We identified the problem of few-shot value function estimation as key step in the bidding process of such an auction. To that end we introduced a greedy online solver that approximates the structure of an unknown multi-attribute value function using a limited number of observed bids and their associated scores. Our algorithm breaks the approximation problem into tractable sub-problems then returns the shape of each objective's value function along with their corresponding weights.

We applied the proposed method to a case study in public works construction bidding. We first elicited a realistic value function through an expert interview, then simulated a bidder's bidding behavior using online value function estimation. Our results indicate that the greedy online solver can quickly identify high-value bids and accurately approximate both the shape and weights of the government's value function within a small number of rounds. However, we also identify certain limitations. Namely that the algorithm tends to plateau due to local non-convexity. Further work may include exploring the efficacy of hybrid methods that combine greedy search with neural network-based meta-learning strategies.

Overall, our results suggest that few-shot learning algorithms can substantially mitigate the strategic uncertainty inherent in multi-stage scoring auctions.

References

[1] John Asker and Estelle Cantillon. Properties of scoring auctions. *RAND Journal of Economics*, 39(1):69–85, 2008.

[2] Federal acquisition regulation part 14: Sealed bidding. `https://www.acquisition.gov/far/part-14`, 2024. Accessed March 2025.

[3] Omer Dekel and Amos Schurr. Cognitive biases in government procurement – an experimental study. *Review of Law Economics*, 10(2):169–200, 2014.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Ward Edwards and F. H. Barron. Smarts and smarter: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*, 60(3):306–325, 1994.

[6] Jamil Soucar. *Real life construction management guide from A - Z*. Maple Leaf Publishing, March 2022.

[7] David M. Messick. D. von winterfeldt and w. edwards. (1986). decision analysis and behavioral research. cambridge: Cambridge university press. pp. 604. *Psychometrika*, 54(2):363–364, 1989.

[8] U.S. Government Accountability Office. Federal buying power: Reverse auctions offer benefits, but guidance is limited. Technical Report GAO-17-59, Government Accountability Office, 2016.