

A Neural Network Auction For Group Decision Making Over a Continuous Space

Yoram Bachrach , Ian Gemp , Marta Garnelo , Janos Kramar , Tom Eccles ,
Dan Rosenbaum , Thore Graepel

DeepMind

{yorambac, imgemp, garnelo, janosk, eccles, danro, thore}@google.com

Abstract

We propose a system for conducting an auction over locations in a continuous space. It enables participants to express their preferences over possible choices of location in the space, selecting the location that maximizes the total utility of all agents. We prevent agents from tricking the system into selecting a location that improves their individual utility at the expense of others by using a pricing rule that gives agents no incentive to misreport their true preferences. The system queries participants for their utility in many random locations, then trains a neural network to approximate the preference function of each participant. The parameters of these neural network models are transmitted and processed by the auction mechanism, which composes these into differentiable models that are optimized through gradient ascent to compute the final chosen location and charged prices.

1 Introduction

We consider multiple agents, which must agree on a location in a continuous space, such as a point in \mathbb{R}^n , and who each have private preferences over the possible locations.

Motivating examples: As an example, consider multiple people who must agree on a single place to locate a certain joint facility, such as a school, a factory or a stadium in a city. Denote the set of possible locations as \mathcal{D} . Each person knows how much value she associates with each possible location $x \in \mathcal{D}$. For instance, one person might want the factory to be located close to them as they work there, whereas another person would want the factory away from them as the noise disturbs them. Similarly a person might want the school close to them, as this saves them money on fuel for their car when driving their kids to school. In this case, each possible location x for the facility has monetary consequences for a participant, and the preferences of a participant i can be captured as a function $v_i : \mathcal{D} \rightarrow \mathbb{R}$ mapping each possible location $x \in \mathcal{D}$ to the utility participant i derives from having the facility being placed in location x . We refer to the function v_i as a “heatmap”, since when locations are points in the plane, it can be visualized as a plot where the color indicates the degree of preference to that location (see example in Figure 1).

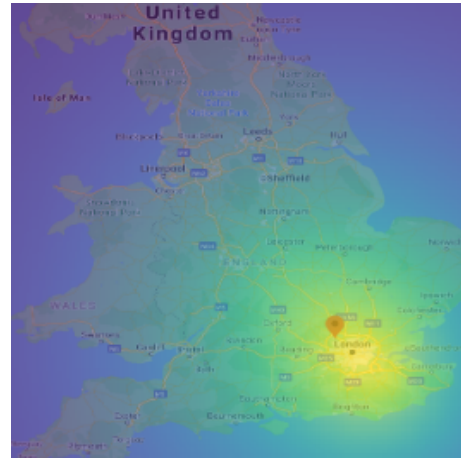


Figure 1: Example heatmap depicting preferences over choices of a location on a map.

Another example is multiple managers who need to decide on the composition of a fund holding a portfolio of k underlying assets. A composition can be captured as $x = (p_1, \dots, p_k)$ where each $0 \leq p_i \leq 1$ is the proportion of the funds allocated to the i 'th underlying asset (so $\sum_{i=1}^k p_i = 1$, and x is a point on the unit simplex). We denote the set of possible compositions as \mathcal{D} . Each manager i may have a different prediction regarding how each asset composition x would do, in terms of expected profit and the risk, and hence has a monetary preference for each alternative. This may again be captured as a function $v_i : \mathcal{D} \rightarrow \mathbb{R}$. In both examples, a participant i knows its own function v_i , but does not the preferences of other participants. How can we select an outcome, such as the facility's location or asset composition, so as to maximize the utility of all participants?

Mechanism design explores how institutions can achieve desirable economic outcomes when self-interested agents interact. Such institutions are essential to ensuring collaboration in human societies, and have the potential of enable cooperation between artificial intelligence agents as well [Dash *et al.*, 2003; Boella *et al.*, 2006; Lopes *et al.*, 2008; Fatima *et al.*, 2014; Rahwan *et al.*, 2019; Dafoe *et al.*, 2020]. Consider agents who must jointly agree on an alternative a from a set A of possible alternatives, with each agent i hav-

ing a valuation function $v_i : A \rightarrow \mathbb{R}$ mapping any alternative $a \in A$ to the value of the alternative a to agent i . These valuations are *private information*, with agent i knowing only their own valuation v_i (but not those of the other agents). A central optimizer can select the alternative maximizing social welfare $a^* = \arg \max_{a \in A} \sum_i v_i(a)$, but this requires knowing all the valuation functions v_i . The central optimizer can request agents to report their valuations v_i , but then agents may have an incentive to misreport their valuations so as to manipulate the optimizer to choose an alternative they prefer.

The Vickrey-Clarke-Groves (VCG) mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973] is a key mechanism design success, prescribing an auction framework for charging agents prices based on reported valuations, that disincentivizes such manipulations and enables choosing the alternative maximizing the social welfare.¹

Our contribution: Existing VCG applications typically deal with a discrete (finite) set A of alternatives, so agents can report the utility function u_i as a table, listing $u_i(a)$ for each $a \in A$. When A is finite but large, earlier research also investigated representation languages for concisely communicating utilities [Boutilier and Hoos, 2001; Lahaie *et al.*, 2008; Nisan, 2000; 2006; Nisan and Ronen, 2001].

We propose a framework to enable applying the VCG mechanism in a **continuous domain** \mathcal{D} , such as participants having valuation over locations in a subset of the 2D plane ($\mathcal{D} \subset \mathbb{R}^2$) or higher dimensional space ($\mathcal{D} \subset \mathbb{R}^m$), where participants cannot communicate their preferences to the central optimizer as a simple table (with a finite number of rows).

Our framework operates by querying each participant i for their valuations in randomly sampled locations, approximating their valuation function $v_i : \mathcal{D} \rightarrow \mathbb{R}$ by training a deep neural network. The parameters of these neural networks are communicated to the central optimizer, which executes the VCG mechanism to compute the chosen location and calculate prices to be paid. We show how the VCG formula can be computed (approximately) using gradient ascent, as neural networks are differentiable models.²

2 Expressing and Aggregating Preferences on Continuous Spaces

A *continuous preferences domain* consists of a continuous domain \mathcal{D} (e.g. $\mathcal{D} \subset \mathbb{R}^2$), and n agents who must agree on a *single* chosen location $r \in \mathcal{D}$; Each agent i has a valuation

¹A key application of VCG or similar auctions is for sponsored search and display advertising [Edelman *et al.*, 2007; Lahaie *et al.*, 2008; Bachrach *et al.*, 2014; Jin *et al.*, 2018], where the alternative set is discrete. In contrast, we focus on continuous alternative sets.

²Our approach is related to other neural auctions [Feng *et al.*, 2018; Tacchetti *et al.*, 2019; Shen *et al.*, 2019], but rather than using neural networks to learn the rules of the auction, we use them as a *language* enabling agents to *express* preferences. Our method is akin to neural methods for aggregating responses [Gaunt *et al.*, 2016; Li’ang Yin *et al.*, 2017; Atarashi *et al.*, 2018], but we focus on strategic agents that may misreport preferences, and examine the VCG auction. Other methods learn user preferences [Gal *et al.*, 2004; Coehoorn and Jennings, 2004; Albrecht and Stone, 2018], but neural networks are differentiable and offer universal function approximation so are well-suited for our use-case.

function $v_i : \mathcal{D} \rightarrow \mathbb{R}$ mapping each element $p \in \mathcal{D}$ to a real value $v_i(p)$ indicating the utility agent i derives from having the agreed location be x . When considering a bounded part of the plane, e.g. $\mathcal{D} = [0, 1] \times [0, 1]$ (i.e. coordinates (x, y) in the plane where $0 \leq x \leq 1$ and $0 \leq y \leq 1$), one can visualize the function v_i as a heatmap, but we emphasize our framework works for the general case $\mathcal{D} = \mathbb{R}^m$.

The central optimizer requires each agent to report their valuations $\{v_i\}_{i=1}^n$ (where each is a function $v_i : \mathcal{D} \rightarrow \mathbb{R}$). We refer to this report representing an agent’s valuation as the agent’s “type” $\theta \in \Theta$ (where Θ denotes the set of all possible such types). A representation method is an encoder e , a reversible function converting an agent’s valuation v_i to an encoding $e(v_i)$, which can be transmitted to the central optimizer.³ The decoder is e^{-1} , the inverse function of an encoder, mapping a representation θ back into a function $v_\theta : \mathcal{D} \rightarrow \mathbb{R}$. We say an agent i with valuation v_i is *truthful* if it transmits $\theta = e(v_i)$, though agents may transmit any valid representation $\theta' \in \Theta$, representing some function $v_{\theta'}$, where $v_{\theta'}$ may be very different from v_i .⁴

We denote the reported representations of agent i as θ'_i . The center takes the valuation reports $\theta'_1, \dots, \theta'_n$ and decodes them to obtain $v_{\theta'_i} = e^{-1}(\theta'_i)$: given i ’s report θ'_i , it determines i ’s value of location k as $v_i(k, \theta'_i) = v_{\theta'_i}(k)$. The mechanism used by the center is specified by two functions f, t , both taking the reports $\theta'_1, \dots, \theta'_n$, where $f : \Theta^n \rightarrow \mathcal{D}$ specifies how to choose the final location $k \in \mathcal{D}$ and $t : \Theta^n \rightarrow \mathbb{R}^n$ specifies the payments t_1, \dots, t_n to extract (t_i is the amount paid by i under the reports $\theta'_1, \dots, \theta'_n$).

A Neural Network Variant of the VCG Mechanism: We use the **VCG mechanism** [Vickrey, 1961; Clarke, 1971; Groves, 1973], which guarantees that participants are incentivized to be truthful, and report a valuation function $v_{\theta'_i}$ that matches their true valuation function v_i .

For brevity we denote $\theta' = (\theta'_1, \dots, \theta'_n)$, and denote $\theta'_{-i} = (\theta'_1, \dots, \theta'_{i-1}, \theta'_{i+1}, \dots, \theta'_n)$. The VCG framework uses an outcome choice function f aimed at maximizing the sum of the agent utilities according to the submitted reports:

$$f(\theta') = \arg \max_{d \in \mathcal{D}} \sum_{i=1}^n v_{\theta'_i}(d)$$

We denote this optimal location according to the agent reports θ' as $d^*(\theta') = f(\theta')$. The Groves family of mechanisms sets the payments to be:

$$t_i(\theta') = h_i(\theta'_{-i}) - \sum_{j \neq i} v_{\theta'_j}(d^*(\theta'))$$

Here, $h_i(\theta'_{-i})$ relates to any function $h_i : \Theta^{n-1} \rightarrow \mathcal{R}$ which only takes into account the reports of agents other than i (i.e. this function does *not* depend on the report θ'_i of agent i).

The VCG mechanism is a special case of the Groves family, which sets h_i as follows: $h_i(\theta'_{-i}) =$

³Standard VCG implementations for discrete domains [Groves, 1973; Nisan and Ronen, 2001] deal with a finite set of alternatives A , so the encoding may be a list of $|A|$ numbers, consisting of $v_i(a)$ for each $a \in A$. Clearly, this is intractable for continuous domains.

⁴The encoding and decoding functions are public information.

$\sum_{j \neq i} v_{\theta'_j}(d_{-i}^*(\theta'_{-i}))$. Here, $d_{-i}^*(\theta'_{-i})$ relates the the (welfare-maximizing) outcome that would have been chosen based solely on the reports of participants other than i , i.e: $d_{-i}^* = \arg \max_{d \in \mathcal{D}} \sum_{j \neq i} v_{\theta'_j}(d)$. In discrete domains \mathcal{D} is a finite set of alternatives; each agent i can transmit its valuation function v_i as a table of $|\mathcal{D}|$ values, and the center computes $\arg \max_{d \in \mathcal{D}} \sum_{i=1}^n v_i(d)$ by iterating over all elements $d \in \mathcal{D}$. Neither is directly feasible where \mathcal{D} is a continuous space, which is where our approach comes in.

2.1 Using Neural Networks as a Representation Language for Preferences

Our approach is based on executing the VCG mechanism by using neural networks as the representation language for the preferences of agents. More precisely, we propose using the parameters of a neural network to express the function $v_i : \mathcal{D} \rightarrow \mathbb{R}$ of each participant. Neural networks are universal function approximators [Csáji and others, 2001; Hornik *et al.*, 1989], i.e. given a large enough number of parameters, neural networks can approximate arbitrarily complex functions. Hence, we can use neural networks as a language for representing participant preferences v_i without needing to impose any restriction on these valuations.

Our framework operates as follows.

1. For every participant i , we randomly sample B elements in the domain \mathcal{D} , denoted as d_1, \dots, d_B .
2. We train a neural network to approximate v_i given these samples, obtaining the neural network parameters θ'_i . There are many possible choices for the neural network architecture, the loss used for the training, or the optimization hyperparameters such as the learning rate. A simple implementation can use a feedforward neural network, and an L_2 loss $\mathcal{L} = (f_{\theta'}(d) - v_i(d))^2$. However, many other choices are also valid, and may yield better results, depending on the domain.
3. Each participant i transmits the trained parameters θ'_i to the center. Note that participants may attempt to manipulate by making a false report i.e. $\theta' \neq \theta$, but the VCG pricing rule computed in the next step disincentivizes such dishonest behavior.
4. The center computes the chosen outcome $d^*(\theta') = \arg \max_{d \in \mathcal{D}} \sum_{i=1}^n v_{\theta'_i}(d)$. As $s = \sum_{i=1}^n v_{\theta'_i}(d)$ is a sum of the functions $v_{\theta'_i}$ (and similarly to each $v_{\theta'_i}$, the sum s is also a function $s : \mathcal{D} \rightarrow \mathbb{R}$); as each $v_{\theta'_i}$ is captured by a neural network, it is a differentiable model, so as a result s is also differentiable. Given a differentiable model s , one can optimize for $\arg \max_{d \in \mathcal{D}} s(d)$ using any method for finding a local maximum of a differentiable function, and we suggest applying the commonly used Gradient Ascent [Sra *et al.*, 2012; Ruder, 2016].
5. The center computes the VCG prices t_i . Similarly to the previous step, to perform the optimization $d_{-i}^* = \arg \max_{d \in \mathcal{D}} \sum_{j \neq i} v_{\theta'_j}(d)$, we note that the partial sum $s_{-i} = \sum_{j \neq i} v_{\theta'_j}$ is a sum of differentiable models $v_{\theta'_j}$, and is a differentiable function, so one can optimize for $\arg \max_{d \in \mathcal{D}_{s_{-i}}(d)}$ using Gradient Ascent.

A limitation of this method is that the VCG formula requires computing *global* optima $\arg \max_{d \in \mathcal{D}} \sum_{i=1}^n v_{\theta'_i}(d)$, but Gradient Ascent may converge on a local rather than global optimum. One may ameliorate this by starting off from many random locations when applying Gradient Ascent (and selecting the optimal location across these runs), or consider placing a restriction on the form that the function v_i can take and using a representation form that allows for an optimization algorithm with better guarantees.⁵

3 Conclusion

Our approach enables users to agree on a location or parameter configuration in a continuous space, taking a form of a VCG auction, executed using neural networks for expressing the preferences of participants. Users may express valuations through a graphical interface, providing data to train neural networks to capture and represent their preferences, and finally compute the VCG outcome and prices. One could also use a programmatic interface, where users provide valuation functions as code, with the system then querying this code to train the neural network. The VCG pricing rule we use disincentivizes tricking the system into selecting a location that improves their individual utility at the expense of others.

Several questions remain open. First, how can we improve the optimization procedure so as to obtain a global rather than local optima? Second, can the neural network architecture be better tailored to specific domains? Finally, could neural network based representations of user preferences be used in other settings, such as voting and social choice?

References

- [Albrecht and Stone, 2018] Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.
- [Atarashi *et al.*, 2018] Kyohei Atarashi, Satoshi Oyama, and Masahito Kurihara. Semi-supervised learning from crowds using deep generative models. In *AAAI*, 2018.
- [Ausubel and Milgrom, 2006] Lawrence M Ausubel and Paul Milgrom. The lovely but lonely Vickrey auction. *Combinatorial Auctions*, 17:22–26, 2006.
- [Bachrach *et al.*, 2010] Yoram Bachrach, Peter Key, and Morteza Zadimoghaddam. Collusion in vcg path procurement auctions. In *WINE*, 2010.
- [Bachrach *et al.*, 2011] Yoram Bachrach, Morteza Zadimoghaddam, and Peter Key. A cooperative approach to collusion in auctions. *ACM SIGecom Exchanges*, 10(1):17–22, 2011.

⁵Another known limitation of the VCG mechanism is that it deters *individual* manipulations. However, VCG is still susceptible to collusion, where *multiple* agents bid strategically [Sandholm, 1996; Ausubel and Milgrom, 2006; Bachrach, 2010; Bachrach *et al.*, 2011; Bachrach, 2010; Bachrach *et al.*, 2010; 2011; Decarolis *et al.*, 2020]. As an implementation of a VCG auction, our approach is also still susceptible to collusion.

- [Bachrach *et al.*, 2014] Yoram Bachrach, Sofia Ceppi, Ian A Kash, Peter Key, and David Kurokawa. Optimising trade-offs among stakeholders in ad auctions. In *ACM EC*, 2014.
- [Bachrach, 2010] Yoram Bachrach. Honor among thieves: collusion in multi-unit auctions. In *AAMAS*, 2010.
- [Boella *et al.*, 2006] Guido Boella, Leendert Van Der Torre, and Harko Verhagen. Introduction to normative multiagent systems. *Comp. & Math. Organization Theory*, 2006.
- [Boutilier and Hoos, 2001] Craig Boutilier and Holger H Hoos. Bidding languages for combinatorial auctions. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 1211–1217. Citeseer, 2001.
- [Clarke, 1971] Edward H Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.
- [Coehoorn and Jennings, 2004] Robert M Coehoorn and Nicholas R Jennings. Learning on opponent’s preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th international conference on Electronic commerce*, pages 59–68, 2004.
- [Csáji and others, 2001] Balázs Csanád Csáji et al. Approximation with artificial neural networks. *Faculty of Sciences, Eötvös Loránd University, Hungary*, 24(48):7, 2001.
- [Dafoe *et al.*, 2020] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tatum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.
- [Dash *et al.*, 2003] Rajdeep K Dash, Nicholas R Jennings, and David C Parkes. Computational-mechanism design: A call to arms. *IEEE intelligent systems*, 18(6):40–47, 2003.
- [Decarolis *et al.*, 2020] Francesco Decarolis, Maris Goldmannis, and Antonio Penta. Marketing agencies and collusive bidding in online ad auctions. *Management Science*, 66(10):4433–4454, 2020.
- [Edelman *et al.*, 2007] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.
- [Fatima *et al.*, 2014] Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. *Principles of automated negotiation*. Cambridge University Press, 2014.
- [Feng *et al.*, 2018] Zhe Feng, Harikrishna Narasimhan, and David C Parkes. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 354–362. International Foundation for Autonomous Agents and Multiagent Systems, 2018.
- [Gal *et al.*, 2004] Ya’akov Gal, Avi Pfeffer, Francesca Marzo, and Barbara J Grosz. Learning social preferences in games. In *AAAI*, 2004.
- [Gaunt *et al.*, 2016] Alex Gaunt, Diana Borsa, and Yoram Bachrach. Training deep neural nets to aggregate crowd-sourced responses. In *UAI*, 2016.
- [Groves, 1973] Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973.
- [Hornik *et al.*, 1989] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [Jin *et al.*, 2018] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *CIKM*, pages 2193–2201, 2018.
- [Lahaie *et al.*, 2008] Sébastien Lahaie, David C Parkes, and David M Pennock. An expressive auction design for online display advertising. In *AAAI*, pages 108–113, 2008.
- [Li’ang Yin *et al.*, 2017] Jianhua Han Li’ang Yin, Weinan Zhang, and Yong Yu. Aggregating crowd wisdoms with label-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. *AAAI Press*, pages 1325–1331, 2017.
- [Lopes *et al.*, 2008] Fernando Lopes, Michael Wooldridge, and Augusto Q Novais. Negotiation among autonomous computational agents: principles, analysis and challenges. *Artificial Intelligence Review*, 29(1):1–44, 2008.
- [Nisan and Ronen, 2001] Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35(1-2):166–196, 2001.
- [Nisan, 2000] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the Second ACM Conference on Electronic Commerce*, pages 1–12, 2000.
- [Nisan, 2006] Noam Nisan. Bidding languages. *Combinatorial auctions*, pages 400–420, 2006.
- [Rahwan *et al.*, 2019] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019.
- [Ruder, 2016] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [Sandholm, 1996] Tuomas W Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In *ICMAS*, 1996.
- [Shen *et al.*, 2019] Weiran Shen, Pingzhong Tang, and Song Zuo. Automated mechanism design via neural networks. In *AAMAS*, 2019.
- [Sra *et al.*, 2012] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. Mit Press, 2012.
- [Tacchetti *et al.*, 2019] Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. A neural architecture for designing truthful and efficient auctions. *arXiv preprint arXiv:1907.05181*, 2019.
- [Vickrey, 1961] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.