Privacy-Preserving Multi-Objective Optimization using Bellman-Ford Algorithm via Zero-Knowledge Proofs

Goshgar C. Ismayilov[1,*,†], Can Ozturan[1,†]

[1]Department of Computer Science, Bogazici University, Istanbul, Turkey

Abstract

Multi-objective optimization has been extensively used to efficiently solve real-world problems. In this work, we address privacy-preserving multi-objective optimization of non-fungible token bartering problem with the consideration of two conflicting objectives as: (i) the maximization of the number of bids satisfied and (ii) the maximization of the budget after the bids costs are paid. To solve this problem, we propose a novel multi-objective approach (i.e. *zkMOBF - Z*ero *K*nowledge-based *M*ulti-*Objective Bellman-Ford*) utilizing zero-knowledge proofs to preserve the privacy of these bids. Our approach takes the global bid graph as input and extracts optimized bartering solution(s) through four phases. Publicly-verifiable proof of this approach is generated off-chain and later verified on-chain on Ethereum Sepolia test network. In our empirical study, we measure proof generation/verification times, proof artifact sizes and blockchain gas consumption for three bartering scenarios. The work justifies the validity and the applicability of our approach.

Keywords

blockchain, zero-knowledge proof, privacy, multi-objective optimization, bartering

1. Introduction

Multi-objective optimization refers to a specific group of problems that involve the minimization or maximization of multiple objectives simultaneously [1]. Multi-objective modelings of real-world problems may offer certain benefits over their single-objective counterparts by allowing for the consideration of trade-offs. For instance, single-objective modelings may neglect significant objectives while trying to push the limit of only one objective (e.g. neglecting cost to minimize time) where it may lead to practically infeasible solutions (e.g. too high cost to pay). In the literature, there exist many works that adopt multi-objective techniques [2, 3, 4]. Therefore, we model a multi-objective bartering with two conflicting objectives in this work as the maximization of number of bids satisfied and maximization of the budget after bid costs are paid. The cost of a bid may result from the urgency to satisfy it where an urgent bid is associated with a lower cost.

In certain scenarios, privacy concerns may arise in real-world multi-objective optimization while processing confidential user data (e.g. recommendation systems on mobile user data [5]). In this context, we define privacy-preserving variant of multi-objective optimization as a more specific group of problems with multiple objectives, which ensures the confidentiality of certain data during computation. There also exist works that address the notion of privacy-preservation on blockchain, especially for federated learning [6]. However, to the best of our knowledge, there exists no work that considers both privacy-preservation and multi-objective optimization on blockchains in the literature. In this work, we propose a multi-objective approach for token bartering by preserving the privacy of bids via zero-knowledge proofs and deploying the corresponding contract on blockchain for proof verification.

The main contributions of our work can be enumerated as follows:

- We address privacy-preserving multi-objective optimization of our token bartering problem which involves performing a publicly-verifiable computation on the private bids to find an optimal

DLT2025: 7th Distributed Ledger Technology Workshop, June, 12-14 2025 - Pizzo, Italy

*Corresponding author.

†These authors contributed equally.

goshgar.ismayilov@boun.edu.tr (G. C. Ismayilov); ozturaca@boun.edu.tr (C. Ozturan)

0009-0007-9307-3637 (G. C. Ismayilov); 0000-0003-0465-2519 (C. Ozturan)

bartering solution by considering multiple objectives.

- We propose a novel privacy-preserving approach (i.e. *zkMOBF*) with four phases: (i) detecting cycles in bid graph with the Bellman-Ford algorithm, (ii) evaluating these cycles (i.e. solutions) for two conflicting objectives, (iii) ranking the feasible solutions with *pareto-domination* to find the non-dominated solutions and (iv) applying a utility function over the non-dominated solutions to determine the final solution. To the best of our knowledge, this is the first work that introduces novelty of solving a multi-objective problem on zero-knowledge proof in the literature.
- We perform experiments over three scenarios (i.e. small, medium and large) with respect to the number of bids to measure (i) proof generation and verification times, (ii) proof artifact size (e.g. size of proof circuit) and (iii) blockchain gas consumption for contract deployment. This study justifies the applicability of our protocol.

2. Privacy-Preserving Multi-Objective Bartering Problem

Privacy-preserving multi-objective bartering problem refers to a secure optimization problem which aims to find the best token bartering solution(s) with respect to the available bids and by considering several conflicting objectives at the same time. In the scope of the problem, we define the set of bidders $\mathcal{U}$ as:

$$\mathcal{U} = \{u_0, u_1, \dots, u_i, \dots, u_{N-1}\} \tag{1}$$

where u_i represents the ith bidder while N is the total number of bidders. We also define the set of non-fungible tokens (e.g. NFTs) as:

$$\mathcal{T} = \{\tau_0, \tau_1, ..., \tau_k, ..., \tau_{M-1}\} \tag{2}$$

where τ_k represents the kth token while M is the total number of tokens. Each bidder u_i is associated with a bid ϕ_i as:

$$\phi_i : \langle \phi_i^-, \phi_i^+, \phi_i^{cost} \rangle \tag{3}$$

where $\phi_i^- \in \mathcal{T}$ is the token to be supplied, $\phi_i^+ \in \mathcal{T}$ is the token to be demanded and ϕ_i^{cost} is the cost of the bid ϕ_i while Φ is the set of all bids. The problem is a single-token single-instance bartering problem where bidders can supply and demand at most one instance of one token ($|\phi_i^-| = |\phi_i^+| = 1$). Here, we also assume that each bidder u_i can propose one bid ϕ_i at most at a time.

The multi-objective modeling of this problem evaluates the set of feasible solutions with several objectives simultaneously by mapping each solution $x_j : \langle x_{j0}, x_{j1}, \dots, x_{ji}, \dots, x_{j,N-1} \rangle$ and $x_{ji} \in \{0, 1\}$ from the decision space to a point in the objective space:

$$max. \quad F(x_j) = \{F_1(x_j), F_2(x_j)\} \tag{4}$$

$$F_1(x_j) = \sum_{i=0}^{N-1} x_{ji} \tag{5}$$

$$F_2(x_j) = B - \sum_{i=0}^{N-1} x_{ji} \cdot \phi_i^{cost} \tag{6}$$

$$s.t. \quad \sum_{i=0}^{N-1} \phi_i^- \cdot x_{ji} \geq \sum_{i=0}^{N-1} \phi_i^+ \cdot x_{ji} \tag{7}$$

where Equation (4) refers to the multi-objective optimization of two objectives where Equation (5) defines the first objective as the maximization of the number of total bids included in the solution while Equation (6) defines the second objective as the maximization of the fixed total budget value B (which is globally shared among bidders) left after the costs of the bids included are covered. Finally, the constraint in Equation (7) implies that the total number of tokens supplied must be at least equal to or greater than the total number of tokens demanded in the solution. A solution x_j is considered as feasible if it satisfies this condition. Furthermore, each solution is assumed to have only a single simple

cycle. In the problem definition, there is no explicit supply constraint since we assume that the bidders already have sufficient balances to propose bids.

We present a simple bartering scenario in Fig. (1) where there exist six bids at total as $\{\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$ for five different tokens $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$. Out of these bids, it is possible to construct a bid graph where the tokens are the nodes and the bids are the edges. For instance, the bid ϕ_3 connects the third node (i.e. source node) τ_3 to the fourth node (i.e. target node) τ_4. In the bid graph, there are two different cycles (i.e. bartering solutions) as $x_0, x_1 \in S$ where x_0 and x_1 are shown in blue and red, respectively while S is the set of feasible solutions. The first objective F_1 of x_0 is the number of bids satisfied as 3 while the second objective F_2 is the budget left as $10 - (1 + 1 + 1) = 7$ where the total budget is taken as 10. We can compute the objectives of x_1 in similar fashion. These two solutions can be represented as points (e.g. blue and red points) in the objective space of (F_1, F_2). The resulting objective space shows that x_0 is non-dominated by outperforming x_1 at every objective (i.e. maximizing the objectives more). The set of all such non-dominated solutions forms the pareto-optimal set, *POS* [1].

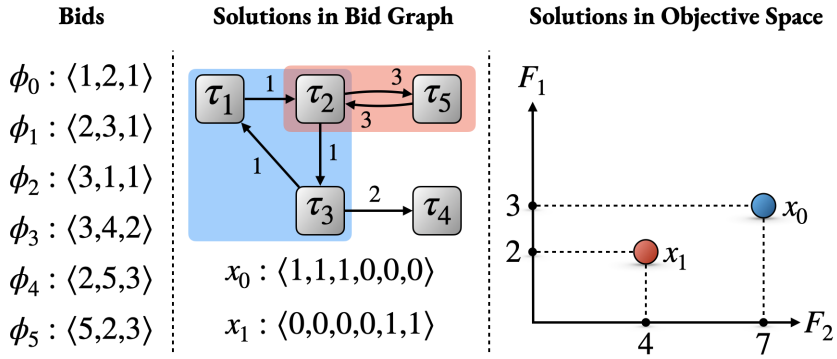

Figure 1: A Simple Illustration for Multi-Objective Bartering Scenario

3. The *zkMOBF* Approach

In this section, we propose the *zkMOBF* approach with four phases as: (i) detecting cycles on bid graph with the Bellman-Ford algorithm, (ii) evaluating feasible cycles (i.e. solutions) with the objectives, (iii) ranking solutions with pareto-domination and (iv) decision-making with utility function for the final solution. This approach is: (i) *privacy-preserving* since it protects the privacy of bids throughout the calculations, (ii) *multi-objective* since it considers the optimization of two objectives at the same time, (iii) *publicly-verifiable* since the off-chain proof generation for the calculations can be verified in blockchain, (iv) *non-interactive* since it does not require communication during proof generation or verification. We present where *zkMOBF* is positioned to solve the privacy-preserving multi-objective bartering problem in Fig. (2) where (i) we assume that barterers propose their bids through their corresponding commitments in the first stage, (ii) they all individually apply our proposed approach to arrive the same bartering solution in the second stage and (iii) they individually apply the solution to exchange tokens in the third stage. Barterers must find the same solution since they run the same deterministic proof circuit over the same inputs. In this paper, we specifically address the challenges of the second stage. Refer to the following work [7] for a potential solution to the first stage.

3.1. Detecting Cycles with Bellman-Ford Algorithm

The bids of the bidders constitute the global bid graph altogether. The constraint given in Equation (7) implicitly limits the valid bartering solutions over this graph to be complete cycles. More intuitively, the token demanded in the last bid in the cycle must be supplied from the first bid in the same cycle. This lets us to use a cycle detection algorithm (i.e. Bellman-Ford) to find the cycles available in the

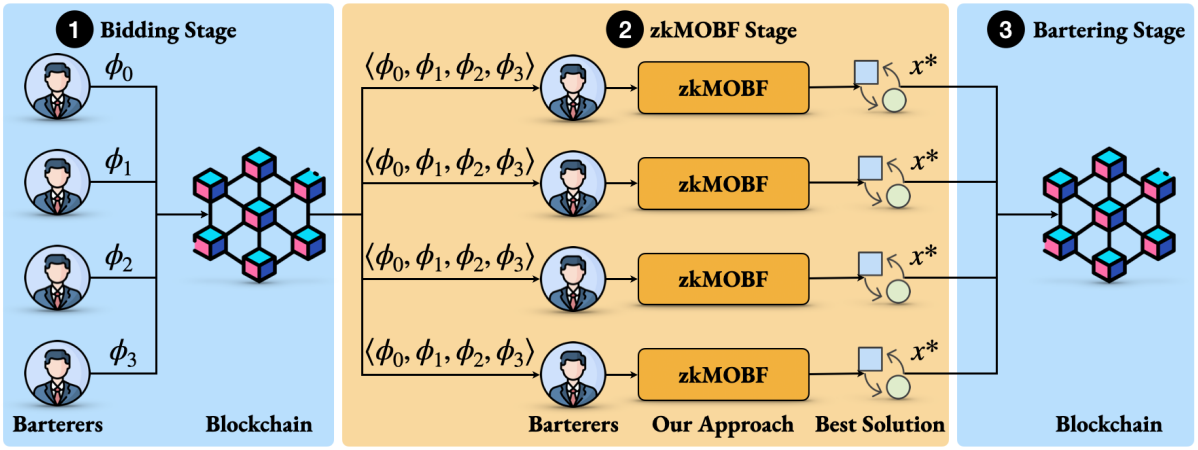

Figure 2: *zkMOBF* Approach in Privacy-Preserving Multi-Objective Bartering

```
1:  def zkMOBF(private bids Φ):
2:      Find cycles (i.e. solutions x_j) from bids Φ with Bellman-Ford algorithm
3:      for each solution x_j do
4:          Add x_j to set of feasible solutions S as: S ← S ∪ x_j
5:          Evaluate x_j with the first objective F_1 as: F_1(x_j)
6:          Evaluate x_j with the second objective F_2 as: F_2(x_j)
7:      for each solution x_j ∈ S do
8:          for each solution x'_j ∈ S that is x_j ≠ x'_j do
9:              Compare x_j and x'_j through pareto-domination
10:         Add x_j to pareto-optimal set POS if it is non-dominated
11:     for each solution x_j ∈ POS do
12:         Apply utility function over x_j to compute the final score
13:         if x_j has better score than the best score then
14:             Select x_j as the best solution as x*
15:     return the best solution found x*
```

Figure 3: Multi-Objective Optimization on Zero-Knowledge

graph. The calculations for this phase correspond to Line 2 in Fig. (3) where it traverses every node in the graph as start node by setting distances to infinity except the start node itself. Then, it iteratively relaxes all the edges to find the shortest paths by updating the distances. After its completion, in case it could not find any further improvement over the distances, it means there is no cycle. Otherwise, it simply backtracks the previous nodes to construct a complete cycle. This cycle is included into the set of feasible solutions in Line 4. This can be seen in Fig. (4) where the cycles are highlighted.

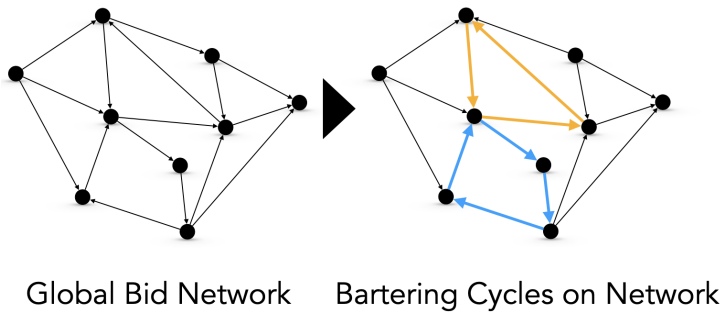

Figure 4: Cycle Detection with Bellman-Ford

3.2. Evaluating Feasible Cycles with Objectives

The cycles (i.e. feasible solutions) found in the given global bid graph must be evaluated with respect to the existing objectives of the multi-objective optimization to find their qualities. We already mathematically define our two objectives in Equation (5) and Equation (6). The calculations for this phase correspond to Lines 5-6 in Fig. (3). For the first objective, it simply counts the number of the bids satisfied for every feasible solution. For the second objective, it sums the costs of the bids and later subtracts the total cost from the available budget for every feasible solution. This phase is especially important to transform the solutions (as collection of decision variables) in the decision space to the points in the objective space. This transformation can be seen in Fig. (5).

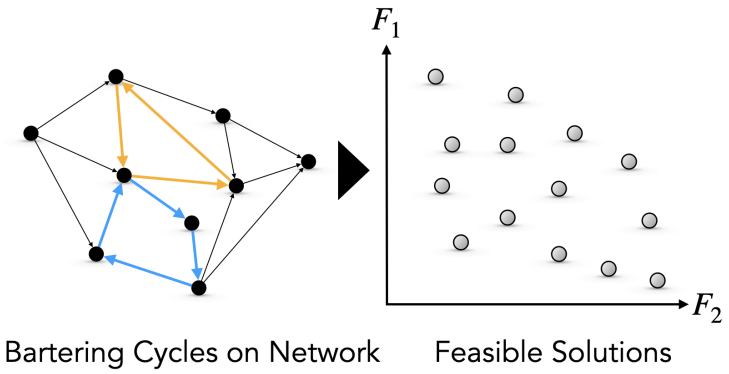

Figure 5: Transformation of Solutions to Objective Values

3.3. Ranking Solutions with Non-Dominated Sorting

In single-objective optimization, the solutions can be ranked by simply sorting their values on that objective ascendingly or descendingly with respect to the type of the problem (i.e. minimization or maximization). However, multi-objective optimization requires a more sophisticated ranking mechanism where we apply *pareto-domination* in this work. This technique is based on pairwise solution comparison where a solution x_1 dominates another solution x_2 in case x_1 is no worse than x_2 for all the objectives and x_1 is strictly better than x_2 in at least one objective, $x_2 \prec x_1$. More formally:

$$\forall i \; F_i(x_2) \leq F_i(x_1) \wedge \exists j \; F_j(x_2) < F_j(x_1) \tag{8}$$

where x_1 is a *non-dominated* solution. The calculations for this phase correspond to Lines 7-10 in Fig. (3) where it compares every solution x_j with all the other feasible solutions and marks x_j as non-dominated if there is not another solution that dominates x_j. The set of all non-dominated solutions constitutes *POS* in Line 10. This can be seen in Fig. (6) where the blue solutions are non-dominated.

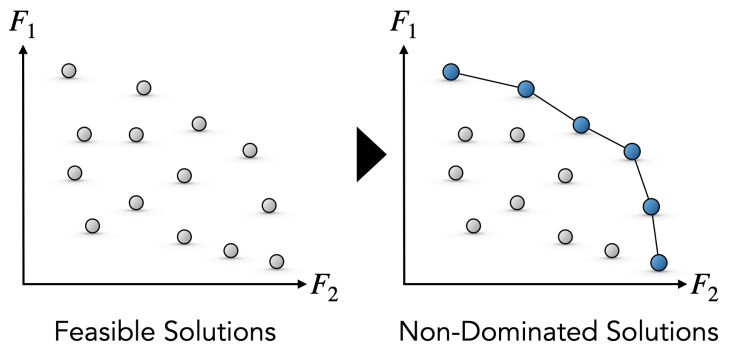

Figure 6: Non-Dominated Ranking of Solutions for *POS*

3.4. Decision-Making for Final Solution with Utility Function

In our work, the third phase generates a pareto-optimal set (i.e. *POS*) that consists of only the non-dominated solutions by cleansing the useless solutions for our problem. The selection of the final solution to be applied over the bids from this set requires an additional phase. For this phase, we incorporate a utility function A that represents a group of probabilities to measure user preferences. These preferences simply show how important an objective is with respect to the other objective(s). Since we have two objectives in our problem, we can represent this function as $A : \langle \alpha_1, \alpha_2 \rangle$ where α_1 and α_2 are the probabilities for the first and second objectives, respectively by satisfying the condition of $\alpha_1 + \alpha_2 = 1$. The calculations for this phase correspond to Lines 11-14 in Fig. (3) where it applies the preferences over the objectives for every optimal solution. It iterates over all the optimal solutions to find the single best solution with the best score in Line 14. This is seen in Fig. (7) where the green solution is shown as the best.

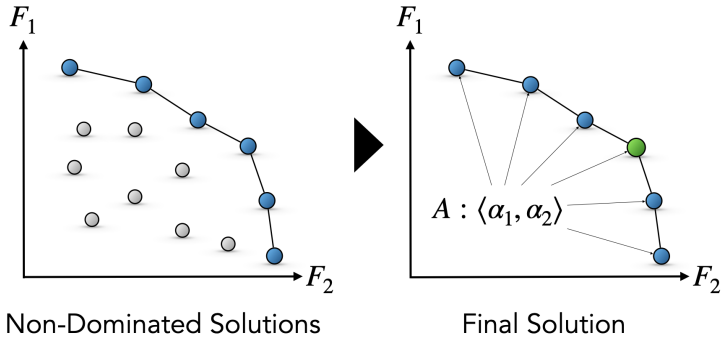

Figure 7: Selection of Best Solution from Optimal Solutions

4. Experimental Study

We use the *ZoKrates* framework [8] to implement *zkMOBF* over zero-knowledge proof. The proof generations and verifications are performed off-chain on the command line. The smart contracts corresponding to the proofs are automatically generated by the framework itself and ready to be deployed on Ethereum Sepolia only for proof verification. During our experiments, we consider three increasingly-complex graphs as small (with 10 bids), medium (with 30 bids) and large (with 50 bids). The open-source implementation of *zkMOBF* is available in our website [9] for further inspection and experimental reproducibility. The experiments are carried out on MacBook Air with M2 chip, 8 GB memory and 8 cores.

4.1. Zero-Knowledge Proof Generation/Verification Times

In this experiment, we measure the proof generation and verification times on the command line with ten independent runs. The results of the experiment are given in Table 1 for the small, medium and large bid graphs. From the table, we simply observe that the time to generate proof increases with the increasing graph complexity (e.g. from 23.67 seconds for small to 188.75 seconds for large). But, the standard deviations between runs are quite low (e.g. 0.01, 1.2 and 1.8 seconds for small, medium and large, respectively). On the other hand, we observe that the proof verification remains constant regardless of the graph complexity. This implies that the proof generation is computationally more expensive than the proof verification, especially for complex instances. Hence, *ZoKrates* strategically moves proof generation out of blockchain to external nodes while still keeping proof verification on-chain.

Table 1
Proof Generation/Verification Times (in seconds).

	Proof Generation			Proof Verification		
Run	Small	Medium	Large	Small	Medium	Large
1	23.5	90.1	189.6	0.013	0.013	0.013
2	23.7	91.1	188.4	0.016	0.012	0.012
3	23.7	92.7	188.9	0.013	0.015	0.012
4	23.6	92.3	187.8	0.013	0.012	0.012
5	23.8	90.2	189.7	0.012	0.013	0.013
6	23.7	88.9	188.5	0.012	0.013	0.012
7	23.8	90.4	189.0	0.012	0.014	0.012
8	23.6	89.1	191.4	0.013	0.013	0.013
9	23.6	89.9	184.4	0.015	0.013	0.013
10	23.7	90.5	189.8	0.013	0.012	0.016

4.2. Proof Artifact Size

In this experiment, we measure the size of several proof artifacts including the proof circuit, the proving key, the verification key and the proof itself. The results of the experiment are given in Table 2 for the small, medium and large bid graphs. From the table, we observe that the number of constraints in circuits and the proving key size increase with the increasing graph complexity (e.g. from +1.5M for small to +8.5M for large in circuits and from 0.63GB for small to 3.78GB for large in proving key). On the other hand, verification keys and proofs remain constant all the time. This is beneficial for blockchain applications since verification keys and proofs are processed on-chain while proving keys are stored off-chain. For the large graph, the proving key is x2.7M larger than the verification key while the proof is the shortest.

Table 2
Zero-Knowledge Proof Artifact Size.

Scenario	Circuit	Proving Key	Verification Key	Proof
Small	1,557,109	0.63GB	1.4KB	0.8KB
Medium	4,959,472	2.11GB	1.4KB	0.8KB
Large	8,515,531	3.78GB	1.4KB	0.8KB

4.3. Blockchain Gas Consumption

In this experiment, we simply deploy the smart contracts that the *ZoKrates* framework automatically generates to blockchain. The structures of these contracts for small, medium and large graphs are basically the same except the verification keys. These keys result from the one-time setup where they are perfectly matched with their corresponding proving keys. According to our experiments, the smart contracts need approximately 1,069,149 gas units (i.e. approximately $3.51) to be deployed on Ethereum Sepolia.

5. Conclusion

We address the privacy-preserving multi-objective bartering problem to find optimal bartering solution(s) to be applied over the bids. It considers two objectives as the maximization of the number of bids and the maximization of the budget left over. For the problem, we contribute a novel privacy-preserving approach (i.e. *zkMOBF*) on zero-knowledge proof by incorporating the Bellman-Ford algorithm, the non-

dominated ranking and the utility function-based decision-making. We evaluate the performance over three bartering scenarios with increasing complexity to measure proof generation times, proof artifact size and blockchain gas consumption. This empirical study indicates the validity and applicability of the approach. In the future, we also plan to construct solutions as union of cycles.

6. Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check, Peer review simulation. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Branke, Multiobjective optimization: Interactive and evolutionary approaches, volume 5252, Springer Science & Business Media, 2008.

[2] J. Du, M. Hu, J. Yin, W. Zhang, Multi-objective gate allocation problem based on multi-commodity network flow model, Applied Sciences 12 (2022) 9849.

[3] S. Khezri, S. Khodayifar, Joint chance-constrained multi-objective multi-commodity minimum cost network flow problem with copula theory, Computers & Operations Research 156 (2023) 106260.

[4] G. Ismayilov, H. R. Topcuoglu, Dynamic multi-objective workflow scheduling for cloud computing based on evolutionary algorithms, in: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), IEEE, 2018, pp. 103–108.

[5] C. Xu, A. S. Ding, S. S. Liao, A privacy-preserving recommendation method based on multi-objective optimisation for mobile users, International Journal of Bio-Inspired Computation 16 (2020) 23–32.

[6] J. Heiss, E. Grünewald, S. Tai, N. Haimerl, S. Schulte, Advancing blockchain-based federated learning through verifiable off-chain computations, in: 2022 IEEE international conference on blockchain (Blockchain), IEEE, 2022, pp. 194–201.

[7] G. C. Ismayilov, C. Özturan, Trustless privacy-preserving data aggregation on ethereum with hypercube network topology, Computer Communications 230 (2025) 108009.

[8] J. Eberhardt, S. Tai, Zokrates-scalable privacy-preserving off-chain computations, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2018, pp. 1084–1091.

[9] zkMOBF, zkmobf github page, 2025. URL: https://github.com/GoshgarIsmayilov/zkMOBF.