frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Original Research
PUBLISHED 29 March 2023
DOI 10.3389/fams.2023.1150921

OPEN ACCESS

EDITED BY
Jianke Zhang,
Xi'an University of Posts and Telecommunications, China

REVIEWED BY
Rudy Raymond,
IBM, Japan
P. Senthil Kumar,
Mohan Babu University, India

*CORRESPONDENCE
Keisuke Otaki
✉ otaki@mosk.tytlabs.co.jp

SPECIALTY SECTION
This article was submitted to Optimization,
a section of the journal
Frontiers in Applied Mathematics and Statistics

RECEIVED 25 January 2023
ACCEPTED 22 February 2023
PUBLISHED 29 March 2023

CITATION
Otaki K, Okada A and Yoshida H (2023) Experimental study on the information disclosure problem: Branch-and-bound and QUBO solver.
Front. Appl. Math. Stat. 9:1150921.
doi: 10.3389/fams.2023.1150921

Experimental study on the information disclosure problem: Branch-and-bound and QUBO solver

Keisuke Otaki*, Akihisa Okada and Hiroaki Yoshida

Toyota Central R&D Labs., Inc., Bunkyo-ku, Tokyo, Japan

The aim of this study was to explore the information disclosure (ID) problem, which involves selecting pairs of two sides before matching toward user-oriented optimization. This problem is known to be useful for mobility-on-demand (MoD) platforms because drivers' choice behaviors are appropriately modeled, but solving the problem is still under development, although heuristic solvers have been proposed. We develop new branch-and-bound-based (BnB) solvers and a new heuristic solver based on a quadratic unconstrained binary optimization (QUBO) formulation. Our numerical experiments show that the QUBO-based solver indeed works within the limit of available bits, and the BnB solver performs slightly better than existing heuristic ones.

KEYWORDS

information disclosure, heuristics, branch-and-bound, quantum annealing, quadratic unconstrained binary optimization

1. Introduction

Matching between two sides (e.g., items and users, items and markets) is an essential task in many real-world applications. Bipartite graph matching has been investigated as a fundamental problem to model such a matching between two sides [1]. A weighted variant of bipartite matching is often applied to find the best matching in terms of associated weights and some global objective functions defined on bipartite graphs. Individual weights could represent various indicators, such as prices, distances, times, and probabilities of taking certain pairs. Real-world applications of matching include matching between children and schools [2, 3], resource allocation [4, 5], and transportation [6, 7]. In another category of settings, the weights of edges can be defined following some probabilistic semantics to represent intuitionistic phenomena [8]. From the optimization viewpoint related to matching, particularly on transportation, other related studies using Fuzzy logic for the intuitionistic phenomena are found in Kumar [9, 10]. Previously, several global properties desired by participants (e.g., platformer/service provider, and individual users) have been studied for bipartite graph matching; an example is to consider stable matching with stated preferences (e.g., given preferences concerning items on another side) [11]. Other examples representing preferences include the use of ranked lists of elements to represent preferences and the use of utility values to quantify preferences (e.g., [12–14]).

Figure 1 illustrates a global optimization task on bipartite matching between drivers and customer orders in the transportation domain. Therefore, each driver has four possible assignments. Typically, driver–order pairs have *given* weights, representing benefits earned by the service, for example. Then, weighted bipartite matching is used to decide a bipartite matching to select the most profitable pairs as a traditional global optimization task. The optimized result on matching is meaningful for some platformers. However, it is

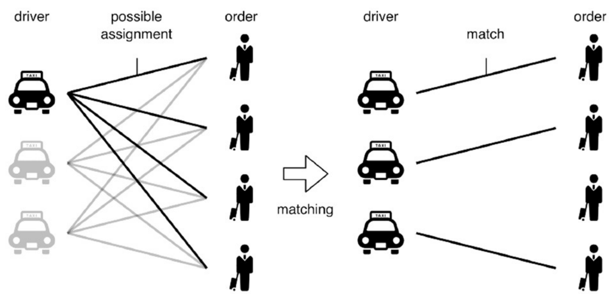

FIGURE 1
Global optimization task on bipartite graphs for the transportation domain. Out of possible driver–order assignments, pairs among drivers and orders are decided in terms of some global metric (e.g., total profit).

not always valuable and acceptable for individuals; for example, some drivers in the traditional matching should pick up some less profitable orders due to the global optimal matching criteria.

Focusing on individual participants on each side of bipartite matching is now an emerging topic for human-oriented decision-making. In contrast to the traditional scenario illustrated in Figure 1, possible assignments are not equivalently acceptable, and weights are not fixed by users from the user-oriented perspective. Yang et al. [15] studied the information disclosure (ID) problem, where weights are *not directly and explicitly given*, but weights themselves *are determined* following *relative values of items displayed to drivers*. In particular, such weights can be optimized by selecting displayed items for each driver. Figure 2 illustrates the idea discussed in the ID problem. The authors adopted discrete choice models to represent the probability of each driver wanting to choose orders and try to optimize the probabilities of drivers' choices. In the ID problem, the authors proposed to prune redundant choices. Pruning redundant choices can be beneficial to drivers as they match with items with higher probabilities than those without pruning, where we can design candidates of choice for users by focusing on the aspect from the users' side. The idea of the ID problem is to cover as many orders as possible with higher probabilities using selection probability (see Section 2 for details), and finally, to solve ordinal weighted matching to realize the driver–order pairs. The authors showed that the aforementioned indirect optimization task is valuable for real-world mobility-on-demand (MoD) services. Note that such a user-oriented aspect of optimization problems has attracted much attention (e.g., user-oriented routing [16, 17] and on-demand transportation [18]). In particular, the adopted discrete choice model is connected to stated preferences, which are studied in the stable matching literature. However, the essential difference is whether or not such preferences are given. That is, in the ID problem, we *optimize* our decisions to fix such choice probabilities, which indirectly represent preferences among orders.

Along with the formulation of the ID problem, Yang et al. developed two heuristic solvers, which are reviewed in Section 2.2 and [15]. However, comparisons with other solvers have yet to be made and the ability of these solvers in terms of algorithms is unknown. Since Yang et al. [15] discussed and evaluated the ID problem in real-world MoD applications, we expect that studying solvers for the ID problem improves such service qualities directly. Furthermore, experimentally comparing the solvers give us insights into the ID problem itself, as it is a relatively new optimization problem. In article paper, we propose new approaches and make comparisons with their previous results. The systematic comparisons clarify the performance of existing methods and the newly developed three approaches: a method based on a quadratic unconstrained binary optimization (QUBO) formulation compatible with the quantum annealing, an enumeration-based exact method, and a branch-and-bound-based solver. We experimentally evaluate these approaches together with re-implemented existing solvers using two sets of randomly generated ID problems with (a) uniform distribution and (b) truncated Gaussian distribution in accordance with Yang et al. [15]. Throughout the experiments, the existing algorithms perform well empirically, particularly in instances of (b). The QUBO-based formulation is shown to give good solutions, but the problem size is rather limited mainly because of the many dummy variables required to transform the original problem. The enumeration-based and the branch-and-bound-based solvers are found to provide slightly better solutions than the existing solvers in certain ranges of the number of orders and drivers.

2. Materials and methods

2.1. Preliminary

2.1.1. Discrete choice model

Following an MoD scenario, let $D = \{d_1, \ldots, d_m\}$ and $O = \{o_1, \ldots, o_n\}$ be sets of drivers and customer orders, respectively. We assume that each driver $d \in D$ selects an order $o \in O$ with probability $p_{d,o}$[1], which is formally defined following a discrete choice model [14, 19]. From the perspective of MoD platforms, a utility[2] $\hat{U}_{d,o}$ is the sum of a deterministic term $U_{d,o}$ and a random term $\epsilon_{d,o}$, that is, $\hat{U}_{d,o} := U_{d,o} + \epsilon_{d,o}$. Therefore $U_{d,o}$ is evaluated for each transportation service with the fee f_o of an order $o \in O$, and the distance $\tau_{d,o}$ of a pair $(d, o) \in D \times O$, $U_{d,o}$ is defined as $U_{d,o} := \beta_0 + \beta_1 f_o + \beta_2 \tau_{d,o}$ with three preference parameters $\boldsymbol{\beta} := \{\beta_0, \beta_1, \beta_2\}$. In particular, these parameters $\boldsymbol{\beta}$ can be estimated from collected data by using maximum-likelihood estimation[3] [20]. The use of this model is a standard approach in MoD research (e.g., Hikima et al. [18]).

Let $O_d \subset O$ be a set of orders displayed to the driver $d \in D$. Let U_d^0 be the initial utility[4] of driver d. Furthermore, we denote by $\{s\}$ that d does not select any order from O_d. Yang et al. adopted the following *nested multinomial logit* (NMNL) model in Equation (1)

1 The original notation in Yang et al. [15] is $p_{o,d}$, but we write it as $p_{d,o}$ for the convenience.

2 The term *utility* is from economics that refers to the total (objective) satisfaction received from consuming a good or service.

3 Yang et al. [15]. reported that $\beta_0 = 0$, $\beta_1 = 1$[USD], $\beta_2 = -0.7$[USD/km], $\alpha = 1.0$, $U_d^0 = 15$[USD] in New York city.

4 Intuitively, the value U_d^0 means the objective satisfaction without any choice. If $U_{d,o} > U_d^0$ of an order o, the driver possibly chooses o.

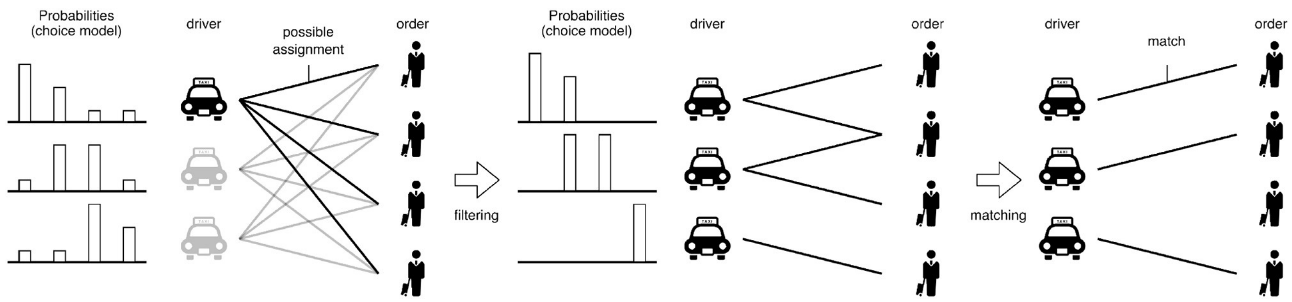

FIGURE 2
Concepts of the information disclosure, where probabilities, drivers, and orders are illustrated. In the first step, choice probabilities are optimized by selecting disclosures in the ID problem to clarify the structure among drivers and orders. In the second step, we try to solve the matching problem.

TABLE 1 Examples of evaluating probabilities $p_{d,o}$ with $O_d = \{o_1, o_2\}, \alpha = 1$ under different four (x_{d,o_1}, x_{d,o_2}) values [i.e., (0, 0), (0, 1), (1, 0), and (1, 1)].

$\mathcal{X} = (x_{d,o_1}, x_{d,o_2})$	(0, 0)	(0, 1)	(1, 0)	(1, 1)
p_{d,o_1}	0.0	0.0	0.982	0.867
p_{d,o_2}	0.0	0.881	0.0	0.117
$p_{d,\{s\}}$	1.0	0.119	0.018	0.016

To reproduce these examples, we can use $U_d^0 = 8, U_{d,o_1} = 12$, and $U_{d,o_2} = 11$.

to define $p_{d,o}$ with the parameter α [15][5]:

$$p_{d,o} = \begin{cases} p(o \mid O_d)p(O_d) & \text{if } o \in O_d \\ p(\{s\}) & \text{if } o = s \end{cases}, \tag{1}$$

Where $p(O_d) := \frac{\exp(\alpha V_{O_d})}{\exp(U_d^0)+\exp(\alpha V_{O_d})}$, $p(\{s\}) := \frac{\exp(U_d^0)}{\exp(U_d^0)+\exp(\alpha V_{O_d})}$, $p(o \mid O_d) := \frac{\exp(U_{d,o})}{\sum_{o'} \exp(U_{d,o'})}$, and $V_{O_d} := \ln\left(\sum_o^{O_d} \exp(\frac{U_{d,o}}{\alpha})\right)$.

2.1.2. Evaluation with decision variables

We prepare the binary decision variables $\mathcal{X} = \{x_{d,o} \in \{0,1\} \mid d \in D, o \in O\}$, where $x_{d,o} = 1$ means the system displays o to d. Table 1 shows an example of evaluating $p_{d,o}$ when $U_{d,o}$ and U_d^0 are given. Figure 3 illustrates four possible displays for the orders $O_d = \{o_1, o_2\}$; that is, we have $(x_{d,o_1}, x_{d,o_2}) \in \{(0,0),(0,1),(1,0),(1,1)\}$. The probabilities computed with different $\mathcal{X}$ are summarized in Table 1. In the following, $p_{d,o}(;\mathcal{X})$ means the probabilities evaluated with the decision variables $\mathcal{X}$.

The ID problem studied in Yang et al. [15] is an unconstrained maximization problem of the objective function:

$$f_{\text{ID}}(\mathcal{X}) := \sum_{o \in O} \mathbb{P}_o(\mathcal{X}), \tag{2}$$

Where the term $\mathbb{P}_o(\mathcal{X}) := 1 - \prod_{d \in D}\left[1 - p_{d,o}(;\mathcal{X})\right]$ is the probability of at least one driver selecting o for each order $o \in O$.

5 The choice becomes at random ($\alpha \to 0$) and seems to be deterministic ($\alpha \to \infty$).

2.2. Existing solvers

The solvers studied in Yang et al. [15] are classified into two categories: (1) closed-form solvers and (2) optimization-based solvers. Therefore, we focus on (2), which is actually the main contribution of Yang et al. [15] showing their solvers to be promising. Optimization-based solvers in (2) are designed to maximize Equation (2). The authors proposed a greedy local search method called *intuitive edge cutting* (IEC) and a method called *minimal-loss edge cutting* (MLEC), which accelerates IEC by pruning redundant candidates.

The basic idea of the two methods is to design a greedy method by evaluating the increments of objective values. Let $\mathcal{X}$ be a current disclosure solution, and $\mathcal{X}^{\text{update}}$ be a modified solution, where one edge $(d,o) \in D \times O$ is removed from $\mathcal{X}$, that is, $\mathcal{X}^{\text{update}} := \mathcal{X} \setminus \{(d,o)\}$. The difference $\delta_{d,o'} := p_{d,o'}^{\text{update}} - p_{d,o'}$ for $o' \in O_d \setminus \{o\}$ of the probability for (d,o') affects the probability $\mathbb{P}_{o'}$ as $\mathbb{P}_{o'}^{\text{update}} - \mathbb{P}_{o'} := \delta_{d,o'} \prod_{d' \in D \setminus \{d\}}(1 - p_{d',o'}(;\mathcal{X}^{\text{update}}) \cdot x_{d',o'})$. Furthermore, the difference $\mathbb{P}_o^{\text{update}} - \mathbb{P}_o := -p_{d,o} \prod_{d' \in D \setminus \{d\}}(1 - p_{d',o}(;\mathcal{X}) \cdot x_{d',o})$ arises for o. With these differences combined, the gain of removing (d,o) is defined as follows:

$$\Delta := \sum_{o' \in O_d \setminus \{o\}} (\mathbb{P}_{o'}^{\text{update}} - \mathbb{P}_{o'}) - (\mathbb{P}_o^{\text{update}} - \mathbb{P}_o). \tag{3}$$

With Equation (3), IEC and MLEC are executed as follows:

- IEC: Begin with $\mathcal{X} = \{x_{d,o} = 1 \mid d \in D, o \in O\}$. Find (d',o') that maximizes Δ, and updates $\mathcal{X} \leftarrow \mathcal{X} \setminus \{(d',o')\}$ while such (d',o') exists.
- MLEC: When computing Δ in IEC, filter (d,o) with the probabilities $p_{d,o}(;\mathcal{X})$ and reduces the number of candidates of size $O(nm)$ to $O(m)$.

Recall that Yang et al. stated that (1) MLEC can be faster than IEC, (2) MLEC can find better solutions within limited computational resources, and (3) the use of ID problem enhances matching results (e.g., more beneficial for MoD platforms) [15]. However, we have no insights on how well IEC and MLEC perform, as they are greedy solvers.

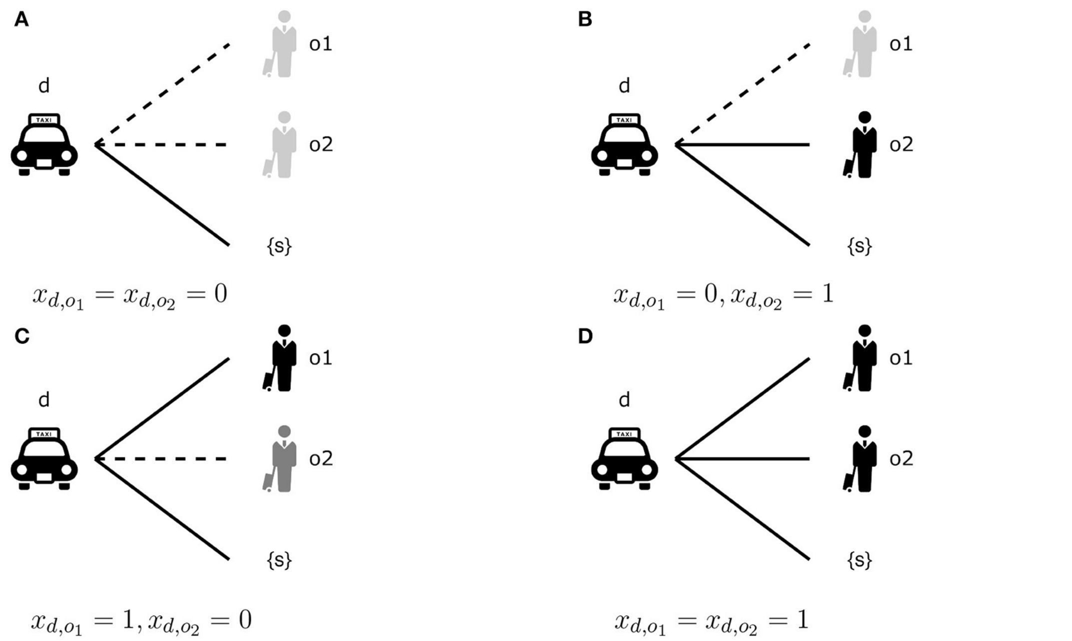

FIGURE 3
Four possible disclosure instances. **(A)** means no choice is shown. **(B, C)** Mean only one order is displayed. **(D)** The case all orders are given to the driver d. Note that $p_{d,o} = 0$ if $x_{d,o} = 0$, i.e., a driver cannot select an order that is not displayed.

2.3. Proposed methods

To deeply understand and conduct experimental comparisons to analyze the ID problem, we develop new solvers. For complex combinatorial optimization problems such as the ID problem in this article, the standard approach to design a solver is to follow the branch-and-bound idea [21]. Therefore, this article develops a branch-and-bound-based solver for the experimental study of the ID problem. Furthermore, recent advantages of quantum-based methods have attracted much attention to the use of quantum-inspired methods to solve combinatorial problems. From this perspective, we also design how to apply such a quantum-inspired method to the ID problem.

2.3.1. QUBO formulation

Quantum annealing (QA) is a heuristic optimization method that is to solve combinatorial optimization problems [22] and has been attracting attention from the theory and industry communities [23]. Its theoretical and physics-based backgrounds are stated in Lucas [22]. The QA is supposed to efficiently and approximately solve QUBO problems, which minimizes as follows:

$$\mathcal{H}(s_1, \ldots, s_N) = -\sum_{i<j} Q_{i,j} s_i s_j - \sum_{i=1}^{N} Q_{i,i} s_i,$$
$$s_i \in \{0, +1\}.$$

Although the current objective function in Equation (2) is not a quadratic function, any higher order terms are essentially transformed into linear combinations of quadratic terms once the problem is expressed in terms of binary variables. We, therefore, try to develop a QUBO-based solver, considering that the upcoming performance of QUBO solvers, including the QA should be promising for optimization [23, 24].

2.3.1.1. Formulation revisited for QUBO formulation

From $\mathbb{P}_o(\mathcal{X}) = 1 - \prod_d (1 - p_{d,o}(; \mathcal{X}))$ and $\max \sum_{o \in O} \mathbb{P}_o(\mathcal{X})$, we solve $\min \sum_{o \in O} \prod_{d \in D} (1 - p_{d,o}(; \mathcal{X}))$. From Equation (1),

$$\begin{aligned}
p_{d,o}(; \mathcal{X}) &= p(o \mid O_d) p(O_d), \\
&= \frac{\exp(U_{d,o}) \cdot x_{d,o}}{\sum_{o' \in O_d} \exp(U_{d,o'}) \cdot x_{d,o'}} \\
&\times \frac{\sum_{o' \in O_d} \exp(U_{d,o'}) \cdot x_{d,o'}}{\exp(U_d^0) + \sum_{o' \in O_d} \exp(U_{d,o'}) \cdot x_{d,o'}} \\
&= \frac{\exp(U_{d,o}) \cdot x_{d,o}}{\exp(U_d^0) + \sum_{o' \in O_d} \exp(U_{d,o'}) \cdot x_{d,o'}} \\
&= \frac{F_1(o,d)}{F_2(d)},
\end{aligned}$$

and $1 - p_{d,o}(; \mathcal{X})$ leads to a fraction of the following form:

$$\begin{aligned}
&1 - p_{o,d}(; \mathcal{X}) \\
&= \frac{F_2(d) - F_1(o,d)}{F_2(d)} \\
&= \frac{F_3(o,d)}{F_2(d)}.
\end{aligned}$$

We then need to minimize $\sum_o \prod_d \frac{F_3(o,d)}{F_2(d)}$. To solve this fractional optimization problem (i.e., $\frac{F_3(o,d)}{F_2(d)}$) with higher order multiplications (i.e., $\prod_d$), we apply a technique [25, 26] to minimize

$$f_{\text{ID}}^{\text{QUBO}} := \mathcal{H}_{\text{num}}(; \mathcal{X}) - K \times \mathcal{H}_{\text{den}}(; \mathcal{X}), \tag{4}$$

Where $\mathcal{H}_{\text{num}}(;\mathcal{X}) := \sum_o \prod_d F_3(o,d)$, $\mathcal{H}_{\text{den}}(;\mathcal{X}) := \prod_d F_2(d)$, and the positive number $K \in \mathbb{R}_+$. This technique is based on the fact that minimizing $\frac{A}{B}$ is equal to simultaneously minimizing A and maximizing B. A direct method of minimizing Equation (4) is labeled `QUBO-Exact` in this article.

2.3.1.2. Implementation

As mentioned earlier, any higher order unconstrained binary optimization problem (HUBO) is essentially transformable to a QUBO problem; see Zaman et al. [27] for the transformation algorithm. Note that, however, the transformation often requires many binary variables on top of the original ones.

Then, we implement a method labeled `QUBO-Approx`, which removes the redundant higher-order terms of $\mathcal{H}_{\text{num}}$ and/or $\mathcal{H}_{\text{den}}$ of Equation (4). Note that $p_{d,o}(;\mathcal{X}) = \frac{F_1(o,d)}{F_2(d)} = \frac{\exp(L)\cdot F_1(o,d)}{\exp(L)\cdot F_2(d)}$, which means that values in F_1 and F_2 can be exponentially shifted by a constant, such as $\exp(L)$. With this technique, parts of terms in $\mathcal{H}_{\text{den}}$ and/or $\mathcal{H}_{\text{num}}$ can reach zero in the HUBO optimization problem. Hence, we expect that `QUBO-approx` could approximately solve ID problem instances with a smaller number of bits than `QUBO-Exact`.

2.3.2. Branch-and-bound-based solver

We develop a solver labeled `BnB`, based on the branch-and-bound idea [21].

2.3.2.1. Branching phase

We have the set $\mathcal{X} = \{x_{d,o} \in \{0,1\} \mid d \in D, o \in O\}$ and $|\mathcal{X}| = |D| \times |O| = mn$. The number of possible assignments of $\mathcal{X}$ is 2^{mn} if we branch the problem into two sub-problems, namely, with $x_{d,o} = 0$ or $x_{d,o} = 1$. There are different possibilities for splitting the problem, e.g., driver-wise branching for $d \in D$ enables us to generate 2^n with n orders in O. An appropriate branching implementation is selected considering the scalability and/or possibilities of evaluating objective values in the bounding phase.

2.3.2.2. Bounding phase

We need to assess the upper/lower bounds of a sub-problem to prune any redundant sub-problems generated. Since the upper bound of $p_{d,o}$, denoted by $\bar{p}_{d,o}$, is obtained from $x_{d,o} = 1$ and $x_{d,o'} = 0$ for $o' \in O_d \setminus \{o\}$, collecting $\bar{p}_{d,o}$ for $d \in D$ leads to the upper bound of $\mathbb{P}_o$, denoted by $\overline{\mathbb{P}}$ for $o \in O$. Therefore, we can prune redundant sub-problems if $\sum_{o \in O} \overline{\mathbb{P}} < f_{\text{ID}}(\mathcal{X}')$ for the current-best feasible solution $\mathcal{X}'$.

Together with the branching phase, we can discuss the computational aspect of branch-and-bound solvers for the ID problem. That is, the computational complexity of BnB solvers is straightforward $\mathcal{O}(2^{mn})$ in the worst case. Numerical evaluations of the average running times of the branch-and-bound-based solvers still need to be investigated in the current status.

2.3.2.3. Incremental deepening with implicit constraints

Since the set of 2^{nm} decision variables is quite large, the aforementioned branch-and-bound-based solver is still inefficient for some ID problem instances. We then incorporate a new perspective into our branch-and-bound-based solver for the ID problem using implicit constraints. Typical solutions obtained with IEC/MLEC often satisfy $\sum_{o \in O_d} x_{d,o} = 1$ for $d \in D$. This is because increasing the number of displayed orders reduces $p_{d,o}$, and so the value of $\mathbb{P}_o$.

Based on the aforementioned implicit constraints, we propose a heuristic, range-constrained branch-and-bound-based solver. Here, a range $[L_d, H_d]$ should be given for each $d \in D$, which means that items whose numbers are in $[L_d, H_d]$ are assumed to be displayed. If $[L_d, H_d] = [1, 1]$, then d accepts only one order. These range constraints drastically reduce the number of sub-problems. The solutions obtained with the aforementioned range constraints are approximations when $H_d < m = |O|$. In this article, we always assume that $L_d = 1$ for $d \in D$ and that $H_d = H$ for $d \in D$. This range-constrained branch-and-bound-based solver is labeled `BnB`H with value H. Then, we implement an iterative deepening procedure [28] for the hyperparameter H, i.e., we run `BnB`H iteratively as $H = 1, 2, \ldots$ and if the obtained solutions with H and $H + 1$ are the same, then we halt the increment of H.

TABLE 2 Summary of two datasets, `UnifID` and `NormID`.

	U	U^0	$N = \lvert O\rvert = \lvert D\rvert$	trial
`UnifID`	Unif(8, 14)	Unif(8, 14)	$2, 3, \ldots, 8$	50
`NormID` [15]	TrunNorm(20, 10, 5, 40)	15	$2, 3, \ldots, 8$	50

Value *trial* means the number of generated instances for each size N.

3. Results and discussions

We experimentally evaluate methods for the ID problem.

3.1. Datasets

We prepare two sets of randomly generated instances and name them `UnifID` and `NormID`. `UnifID` is generated using uniform distributions, denoted by **Unif** in Table 2, for U and U^0, while `NormID` is based on truncated normal distributions, denoted by **TrunNorm** in Table 2, in accordance with Yang et al. [15]. For simplicity, we focus on the case, where $N = m = n$ (i.e., $N = |O| = |D|$) following the experiments using the synthetic datasets in Yang et al. [15]. Table 2 summarizes two datasets.

3.2. Methods

Here, we compare the following methods:

- `Random`: a random assignment of $\mathcal{X}$,
- `ExactEnum`: generate-and-test method for 2^{nm} variables,
- `BnB`: branch-and-bound-based method,
- `BnB`H: heuristic solver with `BnB` and range constraints H,
- `QUBO-Exact`: a naive QUBO-based method,
- `QUBO-Approx`: a method approximating higher-order terms up to dim 3, and
- `IEC/MLEC`: our re-implementation of the IEC/MLEC [15].

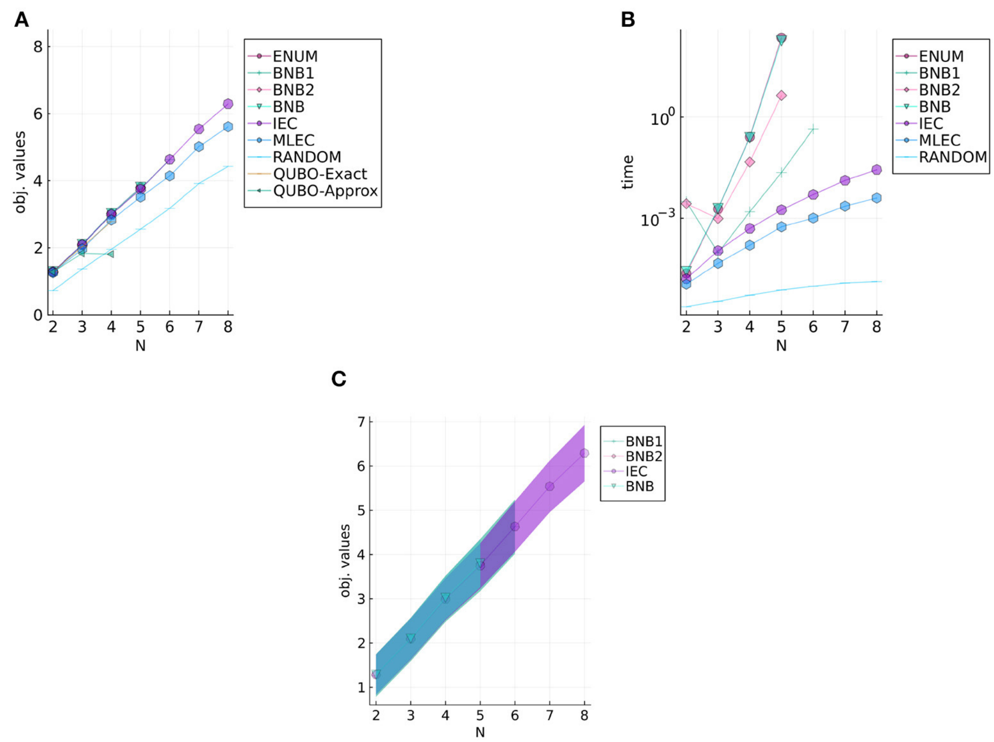

FIGURE 4
Results for `UnifID`, objective values, computational times, and objective values of selected methods. **(A)** Objective values of `UnifID`. **(B)** Computational times of `UnifID`. **(C)** Selected results of `UnifID` with standard variations in filled regions.

3.3. Experimental results

Figures 4, 5 show experimental results of `UnifID` and `NormID`, respectively. Both figures illustrate the mean objective values (Figures 4A, 5A), computational times (Figures 4B, 5B), and objective values (Figures 4C, 5C) of selected methods with standard variations with filled ribbons. Note that computational times of QUBO-based methods are not illustrated because we use a GPU-based annealing solver, and we repeatedly use them (e.g., 30 times) fixed times. According to the results on the two datasets, the two existing heuristic solvers (`IEC` and `MLEC`) efficiently find approximate solutions. As seen in Figures 4C, 5C, compared with results of `BnB` (up to $N = 5$, meaning that 2^{25} binary variables in $\mathcal{X}$), `IEC` and `MLEC` are efficient heuristic solvers. Our results seem to indicate that `IEC` is better than `MLEC` for small instances, although Yang et al. [15] reported that `MLEC` could find better solutions with shorter computational times. That is, we can conjecture the two methods are almost comparable for small instances used in this article. We can interpret our experimental results as follows. First, obtained solutions by `IEC` and `MLEC` were almost competitive for `NormID`, as shown in Figure 5A. Second, Yang et al. did not investigate uniform distributions to evaluate their heuristic solvers in their experiments using synthetic data [15]. One possible reason is that `MLEC` is not good at solving instances generated in `UnifID` since `MLEC` halts at some local optima. Another possible explanation is that the two solvers, `IEC` and `MLEC`, have different properties according to the sizes of instances. We only adopt small instances in this experimental study due to computational issues of the current status of developed solvers.

Furthermore, the QUBO-based methods did find optimal solutions for $N \leq 3$. The applicability is still limited for practical use; however, since the binary variables consumed for transformation to the QUBO problem are significant for the current QUBO solver. We expect that the significance of these methods will be pronounced as the development of hardware for the QUBO solvers.

Compared with `BnB` and `ExactEnum`, the current `BnB` is not much efficient; yet, since the computational times of the two methods are similar. We conjecture that this is because (1) current branching evaluations are not tight enough and (2) the targeting ID objective function f_{ID} does not suit the branch-and-bound-based solver. Since $\mathbb{P}_o$ depends on the whole $\mathcal{X}$, there is no clear structure (e.g., independence) for dividing f_{ID}, and `BnB` could be less effective in estimating the upper and lower bounds of f_{ID}. Linear relaxations of $x_{d,o}$ (i.e., relaxation of $x_{d,o} \in \{0,1\}$ to $0 \leq x_{d,o} \leq 1$) do not accelerate the bounding phase as f_{ID} is the sum of high-dimensional products. Therefore, developing an efficient bounding method would be an important next step in elucidating the branch-and-bound-based method for the ID problem. Such a development could be valuable also for evaluating the performance of `ICE` and `MLEC` since the current experimental study in this article focuses on small instances.

To estimate the performance of `IEC` in greater detail, we compare `BnB`, `BnB1`, and `IEC` on 50 random instances sized $N \in$

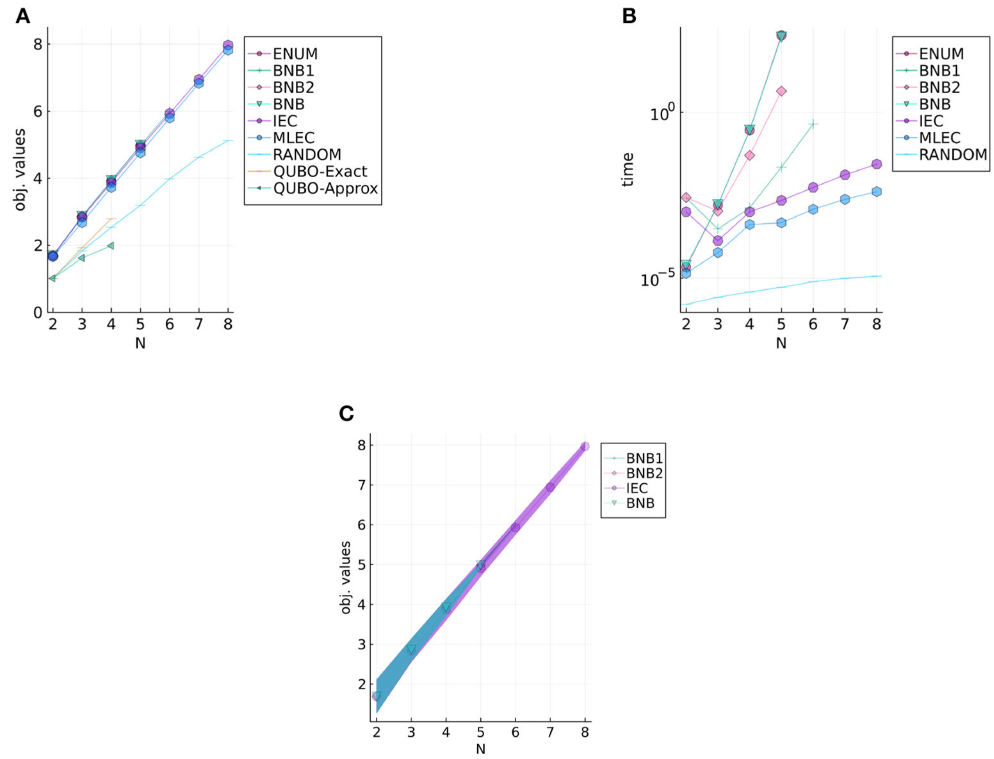

FIGURE 5
Results for NormID, objective values, computational times, and objective values of selected methods. **(A)** Objective values of NormID. **(B)** Computational times of NormID. **(C)** Selected results of NormID with standard variations in filled regions.

TABLE 3 Objective values comparison with BnB1 and IEC for UnifID and NormID.

	NormID			UnifID		
	BnB1 and IEC			BnB1 and IEC		
N	>	=	<	>	=	<
2	0	47	3	1	32	17
3	9	40	1	6	26	18
4	17	33	0	8	21	21
5	24	26	0	12	14	24

Columns (>, =, <) indicate the number of instances whose objective values are better than counterparts.

TABLE 4 Objective values comparison with BnB and IEC for UnifID and NormID.

	NormID		UnifID	
	BnB and IEC		BnB and IEC	
N	>	=	>	=
2	0	50	1	49
3	9	41	7	43
4	17	33	9	41
5	24	26	18	32

Columns (>, =, <) indicate the number of instances whose objective values are better than counterparts.

$\{2, 3, 4, 5\}$ with both NormID and UnifID settings. The results of BnB1 and IEC are summarized in Table 3 and those between BnB and IEC are given in Table 4. The columns show the number of instances after a comparison of objective values with the two methods. For example, with $N = 2$ on NormID, BnB1, and IEC obtain the same objective values for 47 out of 50 instances. According to Table 3, BnB1 obtains better solutions than IEC on average, particularly on NormID. Similar results are observed in Table 4. Note that BnB requires much longer computational times, as shown in Figures 4B, 5B, indicating that BnB1 is to find better solutions than those of IEC. These results may be due to the distributions of $U_{d,o}$ in the computation $p_{d,o}$, i.e., the possible ranges in UnifID are less than those in NormID, and so $p_{d,o}$ values are similar (i.e., modifications in $x_{d,o}$ change objective values drastically).

4. Conclusion and future study

In the present article, we have studied the ID problem by revisiting the methods proposed by Yang et al. [15]. Our experimental evaluations comparing their solvers revealed that the heuristic solver IEC and MLEC are comparable for small instances. These results can also support the findings in Yang et al. [15]. In addition, experimental results on uniformly random data (i.e., `UnifID`) indicate that `MLEC` cannot find better solutions than those by `IEC` on specific instances. We also expect that we could improve the current heuristic `MLEC` according to the property of input data in future study to find better solutions to the ID problem. We confirm that the QUBO-based methods give optimal solutions for $N \leq 3$, but significant hardware development is demanded for practical applications with larger N. This is because the transformation to the QUBO problem requires many dummy binary variables. Better approximation incorporating the effect of higher order terms than the one we employed in this study as a first step would accelerate the application of QUBO solvers. For branch-and-bound-based solvers, `BnB1` slightly improves `IEC` solutions with shorter computational times than `BnB`. We expect that developing better branching estimation for `BnB1` and `BnB` could improve the quality of solutions with shorter times.

These results for a class of optimization problems based on decision models (i.e., probabilities of selecting items) are valuable for discussions of human-centric optimization tasks. In our future study, we will investigate the theoretical background of problems and other objective functions for real applications involving MoD services.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

KO conceived of the presented idea and developed the method and performed the computations. AO and HY verified the analytical methods and results, encouraged KO to investigate quantum annealing solvers and quadratic unconstrained binary optimization (QUBO) problems to discuss the targeting optimization problem, and supervised the findings of this work. All authors discussed the results and contributed to the final manuscript. All authors contributed to the article and approved the submitted version.

Conflict of interest

KO, AO, and HY are employed by Toyota Central R&D Labs., Inc.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fams.2023.1150921/full#supplementary-material

References

1. Korte B, Vygen J. *Combinatorial Optimization: Theory and Algorithms*. Springer (2018). Available online at: https://link.springer.com/book/10.1007/978-3-662-56039-6

2. Kamada Y, Kojima F. Efficient matching under distributional constraints: theory and applications. *Am Econ Rev*. (2015) 105:67–99. doi: 10.1257/aer.20101552

3. Kurata R, Hamada N, Iwasaki A, Yokoo M. Controlled school choice with soft bounds and overlapping types. *J Artif Intell Res*. (2017) 58:153–84. doi: 10.1613/jair.5297

4. De Francisci Morales G, Gionis A, Sozio M. Social content matching in MapReduce. *Proc VLDB2011*. (2011) 4:460–9. doi: 10.14778/1988776.1988782

5. Mehta A. Online matching and ad allocation. *Foundat Trends® Theor Comput Sci*. (2013) 8:265–368. doi: 10.1561/0400000057

6. Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc Natl Acad Sci USA*. (2017) 114:462–7. doi: 10.1073/pnas.1611675114

7. Zhang H, Zhao J. Mobility sharing as a preference matching problem. *IEEE Trans Intell Transport Syst*. (2018) 20:2584–92. doi: 10.1109/TITS.2018.2868366

8. Hussain RJ, Kumar PS. Algorithmic approach for solving intuitionistic fuzzy transportation problem. *Appl Math Sci*. (2012) 6:3981–9. Available online at: http://www.m-hikari.com/ams/ams-2012/ams-77-80-2012/senthilkumarAMS77-80-2012.pdf

9. Kumar PS. Intuitionistic fuzzy solid assignment problems: a software-based approach. *Int J Syst Assurance Eng Manag*. (2019) 10:661–75. doi: 10.1007/s13198-019-00794-w

10. Kumar PS. Algorithms for solving the optimization problems using fuzzy and intuitionistic fuzzy set. *Int J Syst Assurance Eng Manag*. (2020) 11:189–222. doi: 10.1007/s13198-019-00941-3

11. Gale D, Shapley LS. College admissions and the stability of marriage. *Am Math Monthly*. (1962) 69:9–15. doi: 10.1080/00029890.1962.11989827

12. Domshlak C, Hüllermeier E, Kaci S, Prade H. Preferences in AI: an overview. *Artif Intell*. (2011) 175:1037–52. doi: 10.1016/j.artint.2011.03.004

13. Debreu G. Representation of a preference ordering by a numerical function. *Decis Proc*. (1954) 3:159–65.

14. McFadden D. Economic choices. *Am Econ Rev*. (2001) 91:351–78. doi: 10.1257/aer.91.3.351

15. Yang Y, Shi Y, Wang D, Chen Q, Xu L, Li H, et al. Improving the information disclosure in mobility-on-demand systems. *In: Proceedings of the KDD2021*. (2021). p. 3854–64.

16. Mandi J, Canoy R, Lopez VB, Guns T. Data driven VRP: a neural network model to learn hidden preferences for VRP. In: *Proceedings of the CP2021*. (2021). p. 42.

17. Canoy R, Bucarey V, Mandi J, Guns T. Learn-n-Route: learning implicit preferences for vehicle routing. *arXiv:210103936*. (2021). doi: 10.48550/arXiv.2101.03936

18. Hikima Y, Kohjima M, Akagi Y, Kurashima T, Toda H. Price and time optimization for utility-aware taxi dispatching. In: *Proceedings of the PRICAI2021*. (2021). p. 370–81.

19. Train KE. *Discrete Choice Methods With Simulation*. Cambridge: Cambridge University Press (2009).

20. Henningsen A, Toomet O. maxLik: a package for maximum likelihood estimation in R. *Comput Stat*. (2011) 26:443–58. doi: 10.1007/s00180-010-0217-1

21. Lawler EL, Wood DE. Branch-and-bound methods: a survey. *Operat Res*. (1966) 14:699–719. doi: 10.1287/opre.14.4.699

22. Lucas A. Ising formulations of many NP problems. *Front Phys*. (2014) 2:5. doi: 10.3389/fphy.2014.00005

23. Yarkoni S, Raponi E, Schmitt S, Bäck T. Quantum annealing for industry applications: introduction and review. *arXiv:211207491*. (2021) doi: 10.1088/1361-6633/ac8c54

24. Vert D, Sirdey R, Louise S. Benchmarking quantum annealing against "hard" instances of the bipartite matching problem. *SN Comput Sci*. (2021) 2:1–12. doi: 10.1007/s42979-021-00483-1

25. Ajagekar A, Humble T, You F. Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Comput Chem Eng*. (2020) 132:106630. doi: 10.1016/j.compchemeng.2019.106630

26. Wils K. Quantum computing for structural optimization (2020). Available online at: https://repository.tudelft.nl/islandora/object/uuid%3A5167107a-7c26-4779-825a-e99702949870

27. Zaman M, Tanahashi K, Tanaka S. PyQUBO: Python library for mapping combinatorial optimization problems to QUBO form. *arXiv preprint* arXiv:210301708. (2021) doi: 10.1109/TC.2021.3063618

28. Russell SJ. *Artificial Intelligence a Modern Approach*. Hoboken, NJ: Pearson Education, Inc. (2010).