Model-based Approaches to Multi-Attribute Diverse Matching

Jiachen Zhang, Giovanni Lo Bianco, and J. Christopher Beck

Department of Mechanical and Industrial Engineering, University of Toronto, Toronto ON M5S 3G8, Canada
{jasonzjc,giolb,jcb}@mie.utoronto.ca

Abstract. Bipartite b-matching is a classical model that is used for utility maximization in various applications such as marketing, healthcare, education, and general resource allocation. Multi-attribute diverse weighted bipartite b-matching (MDWBM) balances the quality of the matching with its diversity. The recent paper by Ahmadi et al. (2020) introduced the MDWBM but presented an incorrect mixed integer quadratic program (MIQP) and a flawed local exchange algorithm. In this work, we develop two constraint programming (CP) models, a binary quadratic programming (BQP) model, and a quadratic unconstrained binary optimization (QUBO) model for both the unconstrained and constrained MDWBM. A thorough empirical evaluation using commercial solvers and specialized QUBO hardware shows that the hardware-based QUBO approach dominates, finding best-known solutions on all tested instances up to an order of magnitude faster than the other approaches. CP is able to achieve better solutions than BQP on unconstrained problems but under-performs on constrained problems.

Keywords: Specialized Hardware · QUBO · Constraint Programming · Digital Annealer · Diverse Matching.

1 Introduction

Bipartite matching problems assign an agent on one side of a market to an agent on the other side. Weighted bipartite b-matching generalizes such problems to the setting where matches have real-valued weights and agents on one side of the market can be matched to at most b agents on the other side. The weighted bipartite b-matching problem serves as the core of applications such as general resource allocation [6] and recommender systems [18].

The Multi-attribute Diverse Weighted Bipartite b-Matching (MDWBM) problem has been recently introduced to simultaneously maximize the quality and diversity of a bipartite b-matching [1]. The quality is measured by weighted costs of assignments and the diversity is calculated in terms of differences across multiple feature classes. Ahmadi et al. [1] proved that MDWBM is NP-hard and tackled it with a mixed integer quadratic programming (MIQP) model and an exact local exchange algorithm. However, there are flaws in both of these approaches. In this work, we address MDWBM and its constrained variant with three model-based paradigms: Constraint Programming (CP), Binary Quadratic Programming (BQP), and Quadratic Unconstrained Binary Optimization (QUBO). We make three primary contributions:

1. We propose two CP models, a BQP model, and a QUBO model of MDWBM.
2. By adding several practical constraints, we introduce constrained MDWBM.
3. We obtain state-of-the-art results for the standard and constrained MDWBM on software-based solvers with the CP and BQP models and on specialized hardware with the QUBO model, demonstrating that recent hardware architectures can be harnessed for combinatorial optimization problems.

2 Diverse Matching

A matching market often aims to maximize quality subject to some fairness constraints, such as assuring equal opportunity amongst agents. Benabbou et al. [5] study the trade-off between social welfare and diversity for the Singapore housing allocation, modeling diversity with constraints added to a model of an extension of the classic assignment problem.

Diverse bipartite b-matching [2] represents the trade-off between efficiency and diversity, where a matching provides good coverage over different varieties of agents. Diversity has been generally measured by some expression of coverage of the space of possible variation. Mathematically, researchers have used submodular functions, which encode the diminishing returns of similarity. For example, submodular diversity metrics are used in information retrieval communities, including determinantal point processes [21] and Maximum Marginal Relevance (MMR) [7]. Multi-attribute diverse weighted bipartite b-matching is a more general problem as it deals with multiple classes of features. The goal is to form diverse matchings with respect to all feature classes.

To our knowledge, there is no work on *constrained* diverse matching in the literature other than that on conflict and degree constraints in non-diverse bipartite b-matching in e-commerce [8] and vehicular networks [14].

2.1 Multi-Attribute Diverse Weighted Bipartite *b*-Matching

In the Multi-Attribute Diverse Weighted Bipartite b-Matching problem (MDWBM) [1], there are two sets of nodes U and V in a bipartite graph. Every node in V has multiple features, each of which belongs to a different feature class. Let F be the set of feature classes, for example in a worker-team assignment context, $F = \{\text{Gender}, \text{Nationality}\}$. For each $f \in F$, we use G_f to denote the set of values for class f, such as $G_{\text{Nationality}} = \{\text{France}, \text{Canada}\}$. The set of nodes in V with feature value g_f for feature f is denoted by $F_f^{g_f}$. A b-matching on a bipartite graph allows a node from U to be connected to multiple (b) nodes in V via edges. An example of the bipartite b-matching is illustrated in Fig. 1. Each edge is weighted by a cost, hence connecting nodes comes with the cost. The purpose of MDWBM is to minimize the weighted sum of cost and diversity.

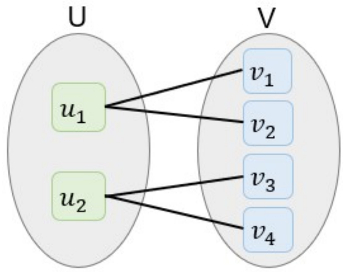

Fig. 1. Bipartite 2-matching.

For MDWBM, the objective function to be minimized is:

$$obj = S + W = \sum_{f \in F} \sum_{u \in U} \sum_{g_f \in G_f} \left(\lambda_f \cdot \left(c_{u,f,g_f} \right)^2 + \lambda_0 \cdot w_{u,f,g_f} \cdot c_{u,f,g_f} \right), \quad (1)$$

where S represents the similarity of the matching and W represents the weighted assignment cost. c_{u,f,g_f} denotes the number of nodes in V connected to node $u \in U$ having value g_f for feature class f. Accordingly, if S_f is the similarity of a matching w.r.t. feature class f, then S can also be represented by:

$$S = \sum_{f \in F} \lambda_f \cdot S_f = \sum_{f \in F} \lambda_f \cdot \sum_{u \in U} \sum_{g_f \in G_f} \left(c_{u,f,g_f} \right)^2 . \quad (2)$$

where $\lambda_f \in \mathbb{Z}^+$ is a weight expressing the importance of feature f. Minimizing S, namely the supermodular similarity function[1] w.r.t. multiple features, has been proved to be NP-hard [1].

A weight is associated with a connection between a feature value g_f to a node $u \in U$. Specifically, the weight $w_{u,f,g_f} \in \mathbb{Z}^+$ represents the cost of assigning a node in V whose feature class f value is g_f to node $u \in U$. The costs are assumed to be integers [1]. The total cost of a matching is

$$W = \lambda_0 \sum_{f \in F} \sum_{u \in U} \sum_{g_f \in G_f} w_{u,f,g_f} \cdot c_{u,f,g_f} \quad (3)$$

where $\lambda_0 \in \mathbb{Z}^+$.

Each node $u \in U$ has a degree of d_u, specifying that the number of nodes in V connected to u is exactly d_u in a matching. Each node $v \in V$ can only be connected to at most one node in U. Ahmadi et al. address problems with these degree constraints, a special case of the general MDWBM.

Constraints. Assignment/allocation problems often contain constraints that, to our knowledge, have not been included in any formulations of MDWBM. In this paper, we address six types of constraints as follows.

[1] The negative submodular diversity is equivalent to the supermodular similarity.

– *Conflict (C)*: Two nodes $v_1, v_2 \in V$ cannot both be assigned to a node $u \in U$. Consider an example in the worker-team assignment context: two workers cannot be assigned to the same team due to a personal conflict.
– *Binding (B)*: Nodes $v_1, v_2 \in V$ must be assigned to the same node $u \in U$.
– *Conflict Assignment (CA)*: A node $v \in V$ cannot be assigned to a node $u \in U$. For example in the paper review context, a reviewer cannot be assigned to a paper due to the conflict of interest.
– *Binding Assignment (BA)*: A node $v \in V$ must be assigned to a node $u \in U$. For example, a particular reviewer must be assigned to a specific paper.
– *Must-Have (MH)*: A node $u \in U$ must be assigned at least one node $v \in V$ with a specific value $g_f \in G_f$. For instance, in an engineering course project, a team must have at least one student who is good at coding.
– *Not-Alone (NA)*: A node $u \in U$ is assigned either 0 or at least E ($E \geq 2$) nodes in V, which have value $g_f \in G_f$. For example in engineering course projects, each group must have either 0 or at least two female students.

We call the multi-attribute diverse matching with degree constraints the *standard MDWBM* and the extension including any of the six practical constraints the *constrained MDWBM*.

2.2 Related Work

There is only one work in the literature that studied the MDWBM. Ahmadi et al. [1] proposed an mixed integer quadratic programming (MIQP) model for the standard MDWBM and also introduced an local exchange algorithm based on negative cycle detection. However, both the model and the algorithm are flawed.

The key problem of the MIQP model is that it uses c_{u,f,g_f} defined above as the decision variable. However, c_{u,f,g_f} does not represent an assignment but rather the number of nodes with a particular feature value assigned to a node. Thus, there is no bijection between the set of assignments and the set of solutions to the MIQP model. In fact, the decision variable choice decouples the combination of feature values from an assignment and hence can provide superoptimal solutions (i.e. the model is a relaxation of the true problem).

The local exchange algorithm uses the identification of negative cycles to improve a matching. A series of moves that leads to a decrease in the objective is called a negative cycle. In each iteration, the algorithm evaluates a neighborhood of solutions via node movement and detects the existence of negative cycles. The algorithm stops when it cannot find any negative cycle. The authors claimed and proved that the algorithm terminates at a global optimum [1]. However, the claim is false as there exists potential objective decrease that cannot be captured by the negative cycles.

The detailed information and counterexamples for both the model and the algorithm flaws are provided in the online appendix.[2]

[2] `https://tidel.mie.utoronto.ca/pubs/Appendix_Matching_CPAIOR22.pdf`.

3 Constraint Programming Models for MDWBM

We propose two constraint programming (CP) models for MDWBM. The first is based on integer assignment variables, like most CP models for bin packing problems. The second model manipulates a list of integer selection variables for each node $u \in U$, requiring more effort to link variables and parameters. For convenience, we use *assignment CP* (ACP) and *selection CP* (SCP), respectively, to represent the two models.

3.1 Assignment CP Model

The ACP is as follows:

$$\min_x \; W + S \tag{4a}$$

$$\text{s.t.} \quad \text{cardinality}(\{x_1, ..., x_{|V|}\}, \{0, 1, ..., |U|\}, \{|V| - \sum_{u \in U} d_u, d_1, ..., d_{|U|}\}), \tag{4b}$$

$$\text{knapsack}(\mathbf{x}_{f,g_f}, \{c_{0,f,g_f}, ..., c_{|U|,f,g_f}\}, \{1, ..., 1\}), \; \forall f \in F, \forall g_f \in G_f, \tag{4c}$$

$$\text{spread}(\{c_{u,f,1}, ..., c_{u,f,|G_f|}\}, \frac{d_u}{|G_f|}, \sigma_{u,f}), \; \forall u \in U, \forall f \in F, \tag{4d}$$

$$W = \lambda_0 \sum_{v \in V} \mathbf{A}_{x_v,v}, \tag{4e}$$

$$S = \sum_{f \in F} \sum_{u \in U} \lambda_f \cdot \left(\sigma_{u,f}^2 \cdot |G_f| + \frac{d_u^2}{|G_f|} \right), \tag{4f}$$

$$x_v \in \{0, 1, ..., |U|\}, \; \forall v \in V, \tag{4g}$$

$$c_{u,f,g_f} \in \{0, 1, ..., d_u\}, \; \forall u \in U \cup \{0\}, \forall f \in F, \forall g_f \in G_f, \tag{4h}$$

$$x_{v_1} \neq x_{v_2}, \; \forall (v_1, v_2) \in C_C, \tag{4i}$$

$$x_{v_1} = x_{v_2}, \; \forall (v_1, v_2) \in C_B, \tag{4j}$$

$$x_v \neq u, \; \forall (u, v) \in C_{CA}, \tag{4k}$$

$$x_v = u, \; \forall (u, v) \in C_{CB}, \tag{4l}$$

$$c_{u,f,g_f} \geq 1, \; \forall (u, f, g_f) \in C_{MH}, \tag{4m}$$

$$c_{u,f,g_f} \in \{0, E, E+1, ..., d_u\}, \; \forall (u, f, g_f, E) \in C_{NA}. \tag{4n}$$

The integer decision variable $x_v = u$ if the node v is assigned to node u, and 0 if the node u is assigned to a dummy node $0 \notin U$. The dummy node does not have a fixed upper bound on degree and is used when some node in V is not matched to any node in U. The integer variable c_{u,f,g_f} represents the number of nodes in V connected to node $u \in U \cup \{0\}$ having value g_f for feature class f. In our implementation, the dummy node is assigned the last index instead of the first one to better fit the typical default search algorithm of CP solvers.

In the model, constraint (4b) is the *global cardinality constraint (gcc)* ensuring the number of nodes in V matched to node $u \in U$ is exactly d_u. The rest of the nodes are matched to the dummy node, which has degree $|V| - \sum_{u \in U} d_u$.

Constraint (4c) is the *multi-knapsack* constraint that links variables $\mathbf{x}$ and $\mathbf{c}$ together. The variable set $\mathbf{x}_{f,g_f}$ contains all the x_v, where node v has g_f as the feature value of feature class f. Constraint (4d) is the *spread* constraint to obtain the standard deviation $\sigma_{u,f}$ of c_{u,f,g_f} (excluding the dummy node) over $g_f \in G_f$. Note that the mean value of c_{u,f,g_f} over $g_f \in G_f$ is the constant $d_u/|G_f|$ as there are exactly d_u nodes in V that are matched to u.

Constraint (4e) represents the assignment cost of the matching, which is also the first component of the objective function to minimize.[3] The (u, v) entry of the matrix $\mathbf{A}$ is the cost of assigning v to u, which is pre-calculated by

$$\mathbf{A}_{u,v} = \sum_{(f,g_f) \text{ if } v \in F_f^{g_f}} w_{u,f,g_f}. \tag{5}$$

Constraint (4f) expresses the similarity of the matching, which is equivalent to term (2). Constraints (4g) and (4h) address the variable ranges.

The standard MDWBM is modeled by objective (4a) and constraints (4b) - (4h), while constraints (4i) - (4n) are for constrained MDWBM. Constraint (4i) is the conflict constraint where C_C contains pairs of conflict nodes in V. Constraint (4j) is the binding constraint where C_B contains pairs of binding nodes in V. Constraint (4k) is the conflict assignment constraint where C_{CA} contains node pairs that cannot be connected. Constraint (4l) is the binding assignment constraint where C_{BA} contains node pairs that must be connected by the assignment. Constraint (4m) is the must-have constraint where C_{MH} contains the 3-tuples $\langle$node in U, feature class, feature value$\rangle$. Constraint (4n) is the not-alone constraint where C_{NA} contains the 4-tuples $\langle$node in U, feature class, feature value, the number of eligible nodes in $V\rangle$. Note that constraints (4k) - (4n) are implemented via direct domain pruning in the model.

3.2 Selection CP Model

The SCP is as follows:

$$\min_x \; W + S \tag{6a}$$

$$\text{s.t.} \quad \text{alldifferent}(\{x_{u,k}, \forall u \in U, \forall k = 1, ..., d_u\}), \tag{6b}$$

$$\text{table}(\mathbf{T}, x_{u,k}, \{z_{u,k}^1, ..., z_{u,k}^{|F|}\}),\ \forall u \in U, \forall k = 1, ..., d_u, \tag{6c}$$

$$\text{cardinality}(\{\{z_{u,1}^f, ..., z_{u,d_u}^f\}, \{1, ..., |G_f|\}, \{c_{u,f,1}, ..., c_{u,f,|G_f|}\}\}), \quad \forall u \in U, \forall f \in F, \tag{6d}$$

$$\text{spread}(\{c_{u,f,1}, ..., c_{u,f,|G_f|}\}, \frac{d_u}{|G_f|}, \sigma_{u,f}),\ \forall u \in U, \forall f \in F, \tag{6e}$$

$$W = \lambda_0 \sum_{u \in U} \sum_{1 \le k \le d_u} \mathbf{A}_{u,x_{u,k}}, \tag{6f}$$

[3] We use (4e) instead of (3) according to the superior results in our experiments.

$$S = \sum_{f \in F} \sum_{u \in U} \lambda_f \cdot \left(\sigma_{u,f}^2 \cdot |G_f| + \frac{d_u^2}{|G_f|} \right), \tag{6g}$$

$$x_{u,k} \in \{1, ..., |V|\},\ \forall u \in U, \forall k = 1, ..., d_u, \tag{6h}$$

$$c_{u,f,g_f} \in \{0, 1, ..., d_u\},\ \forall f \in F, \forall g_f \in G_f, \tag{6i}$$

$$\text{count}(\{x_{u,1}, ..., x_{u,d_u}\}, v_1) + \text{count}(\{x_{u,1}, ..., x_{u,d_u}\}, v_2) \leq 1, \quad \forall u \in U, \forall (v_1, v_2) \in C_C, \tag{6j}$$

$$\text{count}(\{x_{u,1}, ..., x_{u,d_u}\}, v_1) = \text{count}(\{x_{u,1}, ..., x_{u,d_u}\}, v_2), \quad \forall u \in U, \forall (v_1, v_2) \in C_B, \tag{6k}$$

$$x_{u,k} \neq v,\ \forall k = 1, ..., d_u, \forall (u, v) \in C_{CA}, \tag{6l}$$

$$\text{count}(\{x_{u,1}, ..., x_{u,d_u}\}, v) = 1,\ \forall (u, v) \in C_{CB}, \tag{6m}$$

$$c_{u,f,g_f} \geq 1,\ \forall (u, f, g_f) \in C_{MH}, \tag{6n}$$

$$c_{u,f,g_f} \in \{0, E, E+1, ..., d_u\},\ \forall (u, f, g_f, E) \in C_{NA}. \tag{6o}$$

The integer decision variable $x_{u,k} = v$ if v is the k-th node selected by node u. The integer variable c_{u,f,g_f} is the same as in ACP. In the model, constraint (6b) is the *all-different* constraint guaranteeing that nodes in U select distinct nodes in V. Constraint (6c) is the *table* constraint that links $\mathbf{x}$ and $\mathbf{z}$. $\mathbf{T}$ is the feature matrix, where the (v, f) entry is the value of feature f of node v. Then, $z_{u,k}^f$ represents the feature value of the k-th node matched to u. Constraint (6d) is the *gcc* constraint that links $\mathbf{c}$ and $\mathbf{z}$. Constraint (6e) is the same as (4d) and constraint (6f) represents the assignment cost of the matching, with the same cost matrix $\mathbf{A}$ as in ACP. Constraint (6g) is the same as (4f). Constraints (6h) and (6i) express the variable ranges.

The standard MDWBM is modeled by (6a) - (6i), while constraints (6j) - (6o) are for the constrained MDWBM. Similar to ACP, constraints (6j), (6k), (6l), (6m), (6n), and (6o) represent the conflict, binding, conflict assignment, binding assignment, must-have, and not-alone constraints, respectively.

4 Quadratic Models for MDWBM

In this section, we propose a Binary Quadratic Programming (BQP) model and a Quadratic Unconstrained Binary Optimization (QUBO) model for MDWBM.

4.1 BQP Model

We introduce a BQP model with binary assignment variables. The decision variable $x_{u,v} = 1$ if node $v \in V$ is assigned to node $u \in U$, and 0 otherwise. Based on the feature values of each node $v \in V$, we can generate a feature matrix $\mathbf{B} = \{b_{v,f,g_f}\}$ where $b_{v,f,g_f} = 1$ if node v has value g_f for feature f, and 0 otherwise. Similarly, we can generate a weighted cost matrix $\mathbf{C} = \{c_{u,v,f}\}$ based on weighted cost parameters w_{u,f,g_f}. We use $c_{u,v,f}$ to represent the cost for feature class f if node $v \in V$ is assigned to node $u \in U$. We set $c_{u,v,f} = w_{u,f,g_f}$ if node

v has value g_f for feature class f and 0 otherwise. In addition, we also add a dummy node indexed by 0 to deal with situations where a node in V might not be matched to any node in U. Our BQP model is shown below.

$$\min_x \quad \lambda_0 \cdot \sum_{f \in F} \sum_{u \in U} \sum_{v \in V} c_{u,v,f} \cdot x_{u,v} + \tag{7a}$$

$$\sum_{f \in F} \lambda_f \cdot \sum_{u \in U} \sum_{g_f \in G_f} \left(\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \right)^2 \tag{7b}$$

$$\text{s.t.} \quad \sum_{v \in V} x_{u,v} = d_u, \ \forall u \in U, \tag{7c}$$

$$\sum_{u \in U \cup \{0\}} x_{u,v} = 1, \ \forall v \in V, \tag{7d}$$

$$x_{u,v_1} + x_{u,v_2} \leq 1, \ \forall u \in U, \forall (v_1, v_2) \in C_C, \tag{7e}$$

$$x_{u,v_1} = x_{u,v_2}, \ \forall u \in U, \forall (v_1, v_2) \in C_B, \tag{7f}$$

$$x_{u,v} = 0, \ \forall (u, v) \in C_{CA}, \tag{7g}$$

$$x_{u,v} = 1, \ \forall (u, v) \in C_{BA}, \tag{7h}$$

$$\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \geq 1, \ \forall (u, f, g_f) \in C_{MH}, \tag{7i}$$

$$\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \geq E \cdot y_{u,f,g_f}, \ \forall (u, f, g_f, E) \in C_{NA}, \tag{7j}$$

$$\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \leq d_u \cdot y_{u,f,g_f}, \ \forall (u, f, g_f, E) \in C_{NA}, \tag{7k}$$

$$x_{u,v} \in \{0, 1\}, \ \forall u \in U \cup \{0\}, \forall v \in V, \tag{7l}$$

$$y_{u,f,g_f} \in \{0, 1\}, \ \forall u \in U \cup \{0\}, \forall f \in F, \forall g_f \in G_f. \tag{7m}$$

Term (7a) represents the weighted cost of the assignment. Term (7b) represents the supermodular similarity w.r.t. all feature classes. Constraint (7c) guarantees that node $u \in U$ (excluding the dummy node) has a degree of d_u. Constraint (7d) ensures that each node in V is only assigned to one node in U. These components form the BQP model for the standard MDWBM. Constraints (7e) to (7i) model the conflict constraints to must-have constraints, respectively.

Constraints (7j) and (7k) represent the not-alone constraints. The variable $y_{u,f,g_f} = 0$ if node u is not matched to any node in V with feature value g_f of feature f and $y_{u,f,g_f} = 1$ if the number of such nodes matched to u is greater than or equal to E.

4.2 Quadratic Unconstrained Binary Optimization

The recent emergence of specialized hardware has opened up new ways to solve specific computational tasks, such as combinatorial optimization problems [26]. Including adiabatic and gate-based quantum computers [23] and CMOS annealers [3], these novel technologies represent a variety of designs and underlying models of computation. Many of the designs for combinatorial optimization target problems formulated as an Ising model or equivalently as a Quadratic Unconstrained Binary Optimization (QUBO) model [9], which is the following problem:

$$\min y = \frac{1}{2}\sum_i \sum_{j \neq i} W_{i,j} x_i x_j + \sum_i b_i x_i + c, \tag{8}$$

where $x \in \{0,1\}^n$ are binary decision variables, $W \in \mathbf{M}_{n,n}(\mathbb{R})$ is a symmetric weight matrix, $b \in \mathbb{R}^n$ is a bias vector, and $c \in \mathbb{R}$ is a constant [20]. QUBO has been used to represent problems in combinatorial scientific computing [26], machine learning [10], and finance [25]. With multiple feature classes, the supermodular similarity function of MDWBM is naturally quadratic, suggesting that it might be a good candidate for such novel hardware.

In our QUBO model, we use the same binary assignment variables $x_{u,v}$ as in the BQP model. The QUBO model of MDWBM is shown below.

$$\min_x \quad \lambda_0 \cdot \sum_{f \in F} \sum_{u \in U} \sum_{v \in V} c_{u,v,f} \cdot x_{u,v} + \tag{9a}$$

$$\sum_{f \in F} \lambda_f \cdot \sum_{u \in U} \sum_{g_f \in G_f} \left(\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \right)^2 + \tag{9b}$$

$$p_1 \cdot \sum_{u \in U} \left(\sum_{v \in V} x_{u,v} - d_u \right)^2 + \tag{9c}$$

$$p_2 \cdot \sum_{v \in V} \left(\sum_{u \in U \cup \{0\}} x_{u,v} - 1 \right)^2 + \tag{9d}$$

$$p_3 \cdot \sum_{(v_1,v_2) \in C_C} \sum_{u \in U} x_{u,v_1} \cdot x_{u,v_2} + \tag{9e}$$

$$p_4 \cdot \sum_{(v_1,v_2) \in C_B} \sum_{u=1}^{|U|} (x_{u,v_1} - x_{u,v_2})^2 + \tag{9f}$$

$$p_5 \cdot \sum_{(u,v) \in C_{CA}} x_{u,v} + \tag{9g}$$

$$p_6 \cdot \sum_{(u,v) \in C_{BA}} (x_{u,v} - 1)^2 + \tag{9h}$$

$$p_7 \cdot \sum_{(u,f,g_f) \in C_{MH}} \left(\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} - 1 - s_7 \right)^2 + \tag{9i}$$

$$p_8 \cdot \sum_{(u,f,g_f,E) \in C_{NA}} \left(\sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} + s_8 - d_u \cdot y_{u,f,g_f} \right)^2 . \tag{9j}$$

Term (9a) and (9b) are the same as (7a) and (7b). p_1 to p_8 are penalty coefficients that are set to $10 * |F|$. Terms (9c) to (9i) are the penalized terms of constraints (7c) to (7i). Take (7c) and (9c) as an example; as we are minimizing the overall objective function, we want (9c) to evaluate to 0 when constraint (7c) is satisfied and to a non-zero value proportional to its violation when it is not. In term (9i), s_7 is the non-negative slack variable. The lower bound of s_7 is 0 and the upper bound is $|V|-1$. Thus, we use binary variables $z_1, ..., z_{|V|-1}$ to represent s_7 in QUBO, i.e., $s_7 = z_1 + \cdots + z_{|V|-1}$. Similarly, $s_8 = z_1 + ... + z_{d_u - E}$ is the slack variable in term (9j) for not-alone constraints. Note that y_{u,f,g_f} is the same indicator variable as in ACP.

5 Empirical Evaluation

In this section, we present our experimental results on standard and constrained MDWBM with the commercial constraint programming solver CP Optimizer (CPO) v20.1.0, the commercial mathematical programming solver Gurobi v9.5.0, a multistart tabu search algorithm for QUBO [24], and a computer architecture designed for QUBO: the Fujitsu Digital Annealer. CPO/Gurobi are the state-of-the-art for general purpose constraint/mathematical programming. Though the multistart tabu search was developed more than 10 years ago, according to recent work, it is still one of the best metaheuristic approaches to QUBO [13].We use the software-based implementation of the multi-start tabu search version 2 by D-Wave [11].

5.1 Fujitsu Digital Annealer

The Fujitsu Digital Annealer (DA) is a recent computer architecture designed for solving QUBO problems [22]. The third generation DA (DA3), a hybrid system of hardware and software, can represent QUBOs with up to 100000 variables. For our DA environment,[4] the integer coefficients for the quadratic terms range from -2^{62} to 2^{62} and those for the linear terms range from -2^{73} to 2^{73} [16]. The DA algorithm is based on Simulated Annealing (SA), however it takes advantage of the massive parallelization provided by the custom CMOS hardware. The difference between the SA and DA algorithm are as follows:

- DA utilizes parallel tempering that runs a number of problem solving processes (*replicas*) in parallel with different temperatures [12]. Replicas swap temperatures to diversify the search. In each replica, each Monte Carlo step considers all possible one-bit flips in parallel [4].
- DA employs a dynamic offset to raise the energy of a state to escape local minima.
- DA supports a dedicated bit flip mechanism, over a subset of variables belonging to one-hot equivalent constraints when using DA3.

[4] All experiments were conducted on the Digital Annealer environment prepared exclusively for the research at the University of Toronto.

– DA can deal with inequality constraints that are not modeled in QUBO. As a consequence, the terms (9i) and (9j) are not included in the QUBO model when using DA3. Instead, they are represented as the following constraints:

$$1 - \sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \leq 0, \forall (u, f, g_f) \in C_{MH}. \tag{9i'}$$

$$E \cdot y_{u,f,g_f} \leq \sum_{v \in V} b_{v,f,g_f} \cdot x_{u,v} \leq d_u \cdot y_{u,f,g_f}, \forall (u, f, g_f, E) \in C_{NA}. \tag{9j'}$$

In our experiments, we run the DA3 on a remote computer and do not include the communication time in our runtime limits and results. The programs (for running DA, CPO, Gurobi, and tabu search) are written in Python 3.7 and conducted on a Window PC with Intel(R) Core(TM) i7-8700K CPU @3.20GHz with 16 GB RAM.

5.2 Experimental Setting

The proposed QUBO model is tested with three solvers (Gurobi, DA, and tabu search), while the proposed CP and BQP models are tested with CPO and Gurobi, respectively. The six model-solver combinations are each run for 600 seconds for each instance.

Since the runtime limits are the same for the six approaches, we use the best objective value, the time of finding the best objective, and the mean relative error as performance measures. Denote by $B_{i,t,a}$ the best solution attained by runtime t of approach a for instance i. The relative error at time t for approach a on instance i is given by

$$RE(i, t, a) = \frac{B_{i,t,a} - B_i}{B_i} \tag{11}$$

where B_i represents the best solution over all approaches at the end of runtime. For a minimization problem, this expression is always non-negative. The mean relative error of approach a at time t, $MRE(t, a)$, can be computed as

$$MRE(t, a) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} RE(i, t, a). \tag{12}$$

5.3 Experiments on Standard MDWBM

For standard MDWBM, we first run the paper review [17] benchmark dataset from UIUC [19] that were used by Ahmadi et al. It contains 73 papers accepted by SIGIR 2007, 189 prospective reviewers, and 25 major topics. For each paper, a 25-dimensional label is provided based on its relevance to those topics. Similarly for the 189 reviewers, a 25-dimensional expertise representation is provided.

Following Ahmadi et al., we first use spectral clustering to divide reviewers into five clusters based on their topic vectors. We treat the cluster label as the first feature. The assignment cost of a reviewer to a paper is calculated as the relevance of each cluster for each paper. We take the average cosine similarity of label vectors of reviewers in that cluster and the paper. The reviewer demand of each paper is set to 4 (b=4) and no reviewer is assigned to more than 1 paper. Again, following the methodology of Ahmadi et al., to increase the dataset size and the feature number, we create a copy of each reviewer and invert the gender in the copy. The gender is considered as the second feature. We set $\lambda_0 = \lambda_1 = \lambda_2 = 1000$ and round the assignment costs after multiplying by λ_0 as DA only supports integral coefficients. The results are summarized in Table 1. The number of bits is the number of $x_{u,v}$ variables in the QUBO model.

Parameters				Exact methods				Non-exact methods	
$\lvert U\rvert$	$\lvert V\rvert$	$\lvert F\rvert$	#bits	ACP	SCP	GBQP	GBQB	DA3	TABU
3	378	2	1512	**45911**	**45911**	**45911**	**45911**	**45911**	**45911**
13	378	2	5292	**201139**	**201139**	**201139**	**201139**	**201139**	**201139**
23	378	2	9072	**356652**	**356652**	**356652**	357927	**356652**	**356652**
33	378	2	12852	**512177**	**512177**	**512177**	513621	**512177**	**512177**
43	378	2	16632	**669264**	**669264**	**669264**	676947	**669264**	MemOut
53	378	2	20412	**824742**	**824742**	**824742**	MemOut	**824742**	MemOut
63	378	2	24192	**979525**	**979525**	**979525**	MemOut	**979525**	MemOut
73	378	2	27972	**1136424**	**1136424**	**1136424**	MemOut	**1136424**	MemOut

Table 1. Objective results of UIUC paper review instances.

GBQP and GBQB represent Gurobi with the BQP and QUBO models, respectively. The results show that ACP, SCP, GBQP, and DA3 achieve the same solutions, but none proves optimality for any instance. TABU finds the same solutions for the first four instances, but runs out of memory on larger problems. GBQB is outperformed by all other approaches.

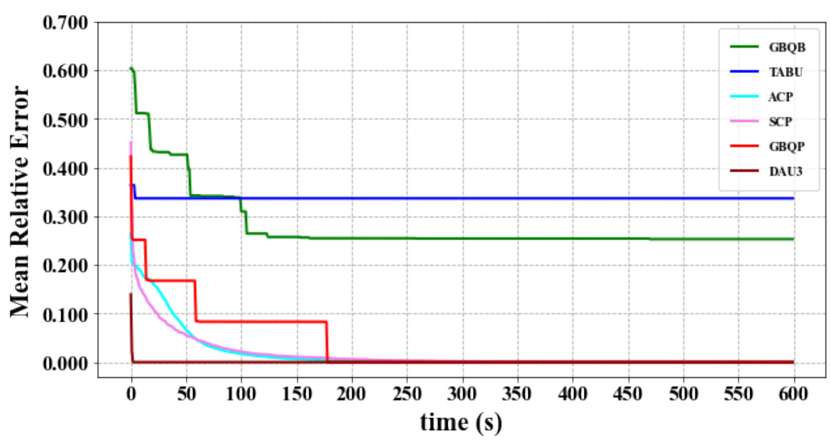

Fig. 2. MRE plot of UIUC paper review instances.

MRE comparisons are shown in Figure 2. To avoid infinite MREs, for instances that induce a memory error, we use a simple heuristic to find an initial solution and use it for each time point during the generation of MRE plots. The heuristic assigns reviewer $\{1, 2, 3, 4\}$ to the first paper, reviewer $\{5, 6, 7, 8\}$ to the second paper, and so forth. From the MRE plot we see that the non-exact methods, TABU and DA3, reach solutions immediately and rarely improve the quality after 10 seconds, though DA3 is much better than TABU. The exact methods gradually increase their solution quality, with ACP, SCP, and GBQP eventually finding the same solutions as DA3. The performance of GBQB, however, never surpasses DA3 at 10 seconds. While we have shown that the two solution approaches of Ahmadi et al. are incorrect in the online appendix, we note that that their reported runtimes are up to 4 orders of magnitude longer than the runtime of DA3. These numeric results, therefore lead to very different conclusions w.r.t. solving MDWBM problems in practice.

	Parameters				Exact				Non-exact	
ID	$\|U\|$	$\|V\|$	$\|F\|$	#bits	ACP	SCP	GBQP	GBQB	DA3	TABU
1-5	25	100	10	2600	0.027	0.032	0.011	0.016	**0.000**	0.028
6-10	25	100	100	2600	0.030	0.035	0.006	0.008	**0.000**	0.014
11-15	50	200	10	10200	0.049	0.052	0.115	0.019	**0.000**	0.030
16-20	50	200	100	10200	0.073	0.065	MemOut	0.009	**0.000**	0.018

Table 2. MRE of randomly generated instances.

In the UIUC dataset, there are only two feature classes. In other application contexts such as machine learning [15], the number of feature classes can be very large. We hence uniformly randomly generated instances with more feature classes [27]. The newly generated instances are of the sizes in terms of $|V| \times |U|$: $\{100 \times 25, 200 \times 50\}$, and the number of feature classes $|F|$: $\{10, 100\}$. The assignment costs are uniformly distributed from 1 to 5. Each feature class randomly has 2 to 10 different values. Each node in U needs to have a degree of 4 (b=4). We generate 5 instances for each size and set $\lambda_0 = \lambda_1 = ... = \lambda_{|F|} = 1$. The MRE results are shown in Table 2 and Figure 3.

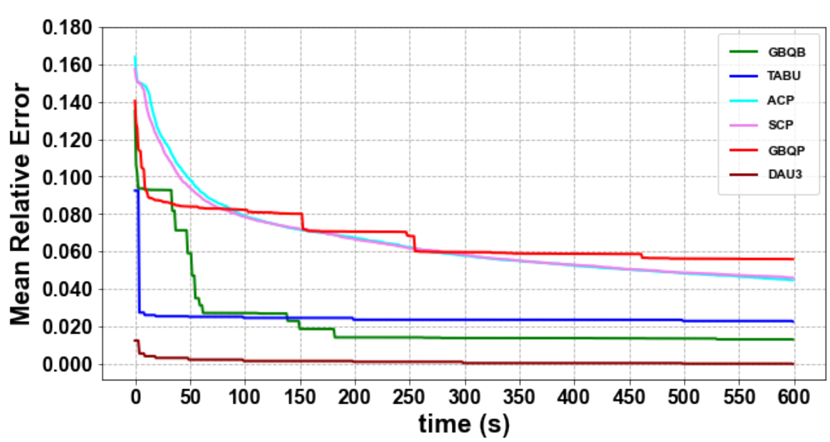

Fig. 3. MRE plot of randomly generated instances.

For the 20 randomly generated instances, DA3 remains the best solution approach. Unlike for UIUC instances, DA3 gradually improves its solution quality, though the improvement is small. For these instances, QUBO-based methods are much better than CP/BQP-based methods, as Gurobi and CPO at 600s do not produce a better solution than DA3/TABU/GBQB at 60s. CPO with both CP models is worse than GBQP initially, but achieves better performance after 280s. Also note that the performance of CP approaches degrades when the problem size or the number of feature classes increases.

5.4 Experiments on Constrained MDWBM

In this section, we test constrained MDWBM. Focusing more on the constraints, we consider the multi-class instances with IDs 1, 2, 6, and 7. Due to limited paper space, we select four different constraint patterns according to the number of each type of constraints, as shown in Table 3.

	Constraint					
Pattern	C	B	CA	BA	MH	NA
PT1	5	5	5	5	5	5
PT2	10	10	10	10	10	10
PT3	0	20	0	0	20	20
PT4	50	0	50	0	0	0

Table 3. Number of constraints in different patterns.

MDWBM with PT2 is more constrained than that with PT1. PT3 and PT4 are practically interesting constraint patterns. PT3 illustrates the situation that nodes in V have binding preferences while nodes in U need to meet specific requirements, while PT4 reflects the circumstances when there are only conflict constraints. The constraints are randomly generated. The results of the constrained MDWBM are shown in Table 4 and Figure 4.

	Constraint	Exact				Non-exact	
ID	Pattern	ACP	SCP	GBQP	GBQB	DA3	TABU
1,2,6,7	PT1	0.032	0.026	0.004	0.040	**0.000**	0.035
1,2,6,7	PT2	0.027	0.026	0.006	0.071	**0.000**	0.065
1,2,6,7	PT3	0.028	0.019	0.004	0.114	**0.000**	0.127
1,2,6,7	PT4	0.043	0.034	0.008	0.014	**0.000**	0.016

Table 4. MRE of instances with constraints.

For constrained MDWBM, though DA3 with QUBO models is still the state-of-the-art, CP/BQP-based methods perform better than other QUBO-based methods. One of the reasons is that CP/BQP models can naturally deal with constraints while QUBO has to convert constraints to penalty terms. Surprisingly, though designed for constrained optimization, CP approaches are worse than GBQP and DA3. We speculate that CPO might not deal with the six types of constraints efficiently as they are not expressed in terms of global constraints with effective filtering algorithms. TABU and GBQB are worse than DA3 by around 6%. Though TABU and GBQB improve the solution quality during their runs, the solutions at 600s are still far from competitive.

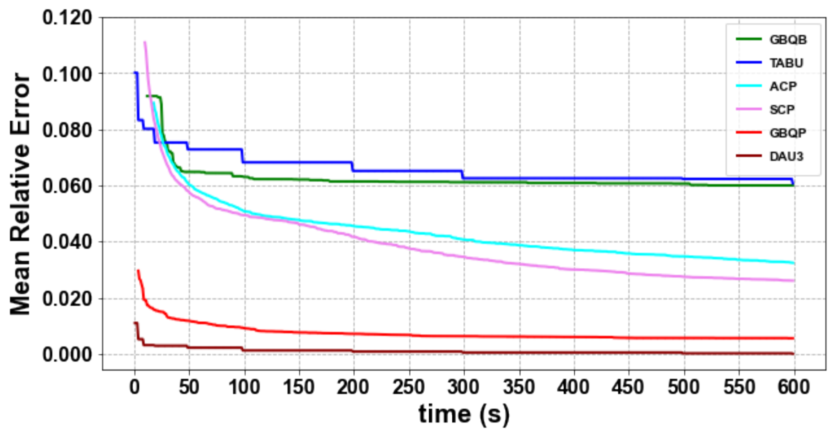

Fig. 4. MRE plot of instances with constraints.

6 Conclusions

We have developed two constraint programming, a binary quadratic programming, and a quadratic unconstrained binary optimization models for the multi-attribute diverse weighted bipartite b-matching problem and introduced practical constraints into the models. Experiments on the standard and constrained MDWBM show that novel hardware DA3 with the QUBO model has an advantage over CP Optimizer with the CP models, Gurobi with the BQP and QUBO models, and tabu search on the QUBO model. We have also identified flaws in existing approaches for standard MDWBM [1].

As traditional computers suffer from the end of Moore's law, it is increasingly important to understand how AI and OR problems can benefit from novel hardware and computation architectures. Our work has demonstrated that for multi-attribute diverse weighted bipartite b-matching, state-of-the-art performance can be delivered by such hardware using a natural model.

Acknowledgement

The authors would like to thank Fujitsu Ltd. and Fujitsu Consulting (Canada) Inc. for providing financial support and access to the Digital Annealer at the University of Toronto. Partial funding for this work was provided by Fujitsu Ltd. and the Natural Sciences and Engineering Research Council of Canada.

References

1. Ahmadi, S., Ahmed, F., Dickerson, J.P., Fuge, M., Khuller, S.: An algorithm for multi-attribute diverse matching. In: Proc. of the 29th International Joint Conference on Artificial Intelligence (IJCAI). pp. 3–9. AAAI Press (2020)
2. Ahmed, F., Dickerson, J.P., Fuge, M.: Diverse weighted bipartite b-matching. In: Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI). pp. 35–41. AAAI Press (2017)
3. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. Frontiers in Physics **7**, 48 (2019)
4. Bagherbeik, M., Ashtari, P., Mousavi, S.F., Kanda, K., Tamura, H., Sheikholeslami, A.: A permutational boltzmann machine with parallel tempering for solving combinatorial optimization problems. In: International Conference on Parallel Problem Solving from Nature. pp. 317–331. Springer (2020)
5. Benabbou, N., Chakraborty, M., Ho, X.V., Sliwinski, J., Zick, Y.: Diversity constraints in public housing allocation. In: 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018) (2018)
6. Bertsimas, D., Papalexopoulos, T., Trichakis, N., Wang, Y., Hirose, R., Vagefi, P.A.: Balancing efficiency and fairness in liver transplant access: tradeoff curves for the assessment of organ distribution policies. Transplantation **104**(5), 981–987 (2020)
7. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 335–336 (1998)
8. Chen, C., Zheng, L., Srinivasan, V., Thomo, A., Wu, K., Sukow, A.: Conflict-aware weighted bipartite b-matching and its application to e-commerce. IEEE Transactions on Knowledge and Data Engineering **28**(6), 1475–1488 (2016)
9. Coffrin, C., Nagarajan, H., Bent, R.: Evaluating ising processing units with integer programming. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 163–181. Springer (2019)
10. Cohen, E., Senderovich, A., Beck, J.C.: An ising framework for constrained clustering on special purpose hardware. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 130–147. Springer (2020)
11. D-Wave System Inc.: D-wave tabu (2021), `https://docs.ocean.dwavesys.com/projects/tabu/en/latest/`, last accessed on 2021-07-21
12. Dabiri, K., Malekmohammadi, M., Sheikholeslami, A., Tamura, H.: Replica exchange mcmc hardware with automatic temperature selection and parallel trial. IEEE Transactions on Parallel and Distributed Systems **31**(7), 1681–1692 (2020)
13. Dunning, I., Gupta, S., Silberholz, J.: What works best when? a systematic evaluation of heuristics for max-cut and qubo. INFORMS Journal on Computing **30**(3), 608–624 (2018)
14. Fazliu, Z.L., Chiasserini, C.F., Malandrino, F., Nordio, A.: Graph-based model for beam management in mmwave vehicular networks. In: Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing. pp. 363–367 (2020)

15. Fern, X.Z., Brodley, C.E., et al.: Cluster ensembles for high dimensional clustering: An empirical study. Technical Report CS06-30-02, Oregon State University (2006)
16. Fujitsu Limited: The third generation of the digital annealer (2021), https://www.fujitsu.com/jp/group/labs/en/documents/about/resources/tech/techintro/3rd-g-da_en.pdf, last accessed on 2021-08-20
17. de Givry, S., Schiex, T., Schutt, A., Simonis, H.: Modelling the conference paper assignment problem. In: 19th Workshop on Constraint Modeling and Reformulation (ModRef-20) (2020)
18. Kadıoğlu, S., Kleynhans, B., Wang, X.: Optimized item selection to boost exploration for recommender systems. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 427–445. Springer (2021)
19. Karimzadehgan, M., Zhai, C.: Constrained multi-aspect expertise matching for committee review assignment. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 1697–1700 (2009)
20. Kochenberger, G., Hao, J.K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. Journal of combinatorial optimization **28**(1), 58–81 (2014)
21. Kulesza, A., Taskar, B.: Determinantal point processes for machine learning. arXiv preprint arXiv:1207.6083 (2012)
22. Matsubara, S., Takatsu, M., Miyazawa, T., Shibasaki, T., Watanabe, Y., Takemoto, K., Tamura, H.: Digital annealer for high-speed solving of combinatorial optimization problems and its applications. In: 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC). pp. 667–672. IEEE (2020)
23. Mohseni, M., Read, P., Neven, H., Boixo, S., Denchev, V., Babbush, R., Fowler, A., Smelyanskiy, V., Martinis, J.: Commercialize quantum technologies in five years. Nature News **543**(7644), 171 (2017)
24. Palubeckis, G.: Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. Annals of Operations Research **131**(1), 259–282 (2004)
25. Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., De Prado, M.L.: Solving the optimal trading trajectory problem using a quantum annealer. IEEE Journal of Selected Topics in Signal Processing **10**(6), 1053–1060 (2016)
26. Tran, T.T., Wang, Z., Do, M., Rieffel, E.G., Frank, J., O'Gorman, B., Venturelli, D., Beck, J.C.: Explorations of quantum-classical approaches to scheduling a mars lander activity problem. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
27. Zhang, J., Bianco, G.L., Beck, J.C.: MDWBM Instances (2021), https://github.com/JasonZhangjc/mdwbm-instances, last accessed on 2022-02-11