Conflict-Aware Weighted Bipartite B-Matching and Its Application to E-Commerce

Cheng Chen, *Student Member, IEEE,* Lan Zheng, Venkatesh Srinivasan, *Member, IEEE,* Alex Thomo, *Member, IEEE,* Kui Wu, *Senior Member, IEEE,* and Anthony Sukow

Abstract—The weighted bipartite b-matching problem (WBM) plays a significant role in many real-world applications, including resource allocation, scheduling, Internet advertising, and E-commerce. WBM has been widely studied and efficient matching algorithms are well known. In this work, we study a novel variant of WBM, called conflict-aware WBM (CA-WBM), where conflict constraints are present between vertices of the bipartite graph. In CA-WBM, if two vertices (on the same side) are in conflict, they may not be included in the matching result simultaneously. We present a generalized formulation of CA-WBM in the context of E-commerce, where diverse matching results are often desired (e.g., movies of different genres and merchants selling products of different categories). While WBM is efficiently solvable in polynomial-time, we show that CA-WBM is NP-hard. We propose approximate and randomized algorithms to solve CA-WBM and show that they achieve close to optimal solutions via comprehensive experiments using synthetic datasets. We derive a theoretical bound on the approximation ratio of a greedy algorithm for CA-WBM and show that it is scalable on a large-scale real-world dataset.

Index Terms—weighted bipartite matching, conflict constraint, optimization, approximation.

1 Introduction

Weighted bipartite b-matching (WBM) is one of the fundamental and widely studied problems in combinatorial optimization. Given a weighted bipartite graph $G = (U, V, E)$ with weights $W : E \rightarrow \mathbb{R}^+$, where U, V and E represent left vertices, right vertices and edges, the weighted bipartite b-matching problem (WBM) is to find a subgraph $M \subseteq G$ such that every vertex i in M is incident to at most $b(i)$ edges (degree constraints). The overall weight of M is given by $w(M) = \sum_{e \in M} w(e)$. Depending on the purpose, WBM tries to maximize or minimize $w(M)$. Without loss of generality, we focus on the maximization version of WBM in this paper.

In the current age of information, due to its expressive power, WBM and its variants find important applications in many areas, such as resource allocation [1], [2], scheduling [3], Internet advertising [4] and recommender systems [5]. Consider a problem of recommending items (e.g., books) to readers as an example. Typically, the recommender system should satisfy three requirements: (1) degree constraints on the number of recommendations books are part of within the maximum availability, (2) degree constraints on the number of recommendations readers receive before they become overwhelmed, and (3) recommendation of books should be based on reader preferences. This problem can be naturally modeled by WBM, where the left-side nodes and right-side nodes of the bipartite graph denote readers and books, respectively, and each edge weight represents the preference of a book to a reader. Note that edge weights can be learned beforehand using collaborative filtering algorithms [6], [7], [8]. The goal is to find a subgraph such that the total weight (preference) of the matched edges in the subgraph is maximized, while satisfying all degree constraints.

An implicit assumption of WBM is that any two nodes on the same side do not interfere with each other, even if they share similar features. For example, a recommender system running WBM can recommend several books of the same subject to a reader, as long as the subject is his/her favorite and the availability constraints of the books are not violated. This, however, does not generate desired results in some real-world scenarios. For book recommendation, a reader may not want all recommended books from the same subject but instead may prefer books of diverse subjects so that more interesting topics can be discovered. The recommender system should allow a reader to constrain the number of books from the same subject. In other words, books from the same subject are in "conflict" with each other when being recommended to a reader, and the number of such conflicts should be below the reader's tolerance threshold. Hence, the interference issue would inevitably lead to a new challenge in WBM when generating the matching result. This conflict challenge has not been fully studied and thus is the focus of this work.

In fact, interference issues are prevalent and can pose a significant effect on practical applications, especially in the booming market of E-commerce. Typically, E-commerce companies (like eBay or Terapeak[1]) have access to a massive buyer-seller bipartite graph, which contains all transactional data. Each edge and its weight represent the monetary aggregate between a buyer and seller. In order to analyze the

• C. Chen, L. Zheng, V. Srinivasan, A. Thomo and K. Wu are with the Computer Science Department, University of Victoria, Victoria, BC V8W 3P6, Canada (Contact E-mail: wkui@uvic.ca).

• A. Sukow is with Terapeak.

1. Terapeak is an E-commerce company, helping sellers on eBay or Amazon to measure and boost their sales performance. http://www.terapeak.ca/

purchase behavior and match buyers to sellers, companies often need to identify a highly profitable subgraph, where most profitable transactions take place. While this problem can be modeled by WBM, it is worth pointing out that some matching results may be redundant and not desired due to buyers (or sellers) with very similar profiles. For example, it is not desirable to have two buyers in the same household[2] simultaneously matched to a seller who sells large household items such as TVs and washers. As another example, a seller may want their buyers to be geographically distributed in different cities to promote their product in a broader area. Hence, with a limited number of choices, the seller needs to limit the number of buyers from the same city.

In this article, we introduce a new generalization of WBM, Conflict-Aware Weighted Bipartite b-Matching (CA-WBM), that can address the conflict challenges mentioned above. For ease of understanding, we describe CA-WBM in the context of E-commerce. In this scenario, the input bipartite graph is composed of buyer vertices, seller vertices, and edges between buyers and sellers. From the point of view of an E-commerce company, an edge typically associates a profit weight for each pair of a buyer and a seller. Then the goal is to maximize the total profit from all potential transactions between buyers and sellers.

In the simplest form of CA-WBM, we assume that each buyer and seller has a specific "capacity". That is, a buyer cannot be matched to more sellers than his capacity, and a seller cannot be matched more buyers than he has budget (capacity) for. The matching result should maximize the total profitability (sum of recommendation weights), while respecting the capacity constraints of the buyers and sellers. It is easy to see that this is an instance of WBM. Next, we enrich the model in a natural way by allowing the system to accommodate various matching strategies in terms of conflict constraints. To capture these constraints, we assume the buyers (sellers) could be "in conflict" with other buyers (sellers).

We also study an online version of CA-WBM (online CA-WBM), where the buyers arrive in an online fashion and the corresponding edges are revealed when each buyer arrives. Specifically, online CA-WBM is relevant to Internet advertising [4], including online adwords [9], [10] and display advertising [11], [12]. Nevertheless, these problems do not consider conflict between entities (e.g., adwords). Therefore, online CA-WBM will be valuable in providing a more flexible service to Internet advertising. For example, if a bidder expects his/her products to be exposed to a wide range of adwords, he/she can set conflict constraint based on similarities between adwords.

In this work, we propose a graph-theoretic model for CA-WBM and show that CA-WBM is NP-hard. We present efficient approximate algorithms to solve CA-WBM, which achieve close to optimal solutions in comprehensive experiments with synthetic and real-world datasets. More specifically, our contributions are as follows:

1) We initiate the study of natural extensions of classic weighted bipartite b-matching (WBM). The question we address is how to maximize the total weight when matching vertices are under both degree and conflict constraints (CA-WBM).
2) We present a general formulation of CA-WBM that directly models both the degree constraint on each vertex and conflict relationship between two vertices on the same side. We model it using semidefinite programming (SDP) and integer linear programming (ILP). To the best of our knowledge, this explicit modelling is completely new.
3) We prove that CA-WBM is NP-hard and we present greedy and linear programming (LP) based algorithms that are scalable and close to optimal. We also provide a randomized algorithm that solves the online version of CA-WBM.
4) We provide an extensive experimental evaluation on synthetic and real-world datasets validating our claims of scalability and optimality.

2. Such information could be identified in data, e.g., based on buyers' mailing address and home phone number.

The remainder of this paper is organized as follows. We formally define CA-WBM in Section 2 and prove its NP-hardness in Section 3. In Section 4, we design different algorithms to solve CA-WBM using SDP and ILP. We also propose a greedy algorithm and prove its performance bound. In Section 5, we study an online version of CA-WBM and propose a randomized algorithm to solve the online problem. We bound the competitive ratio of the randomized algorithm. We present experimental evaluation in Section 6. Related work is introduced in Section 7. We conclude the paper in Section 8.

2 Conflict-Aware Weighted Bipartite b-Matching (CA-WBM)

To facilitate understanding of CA-WBM, we start by first studying the simpler, weighted bipartite b-matching (WBM) problem without considering the conflict between buyers (sellers). To describe the problem formally, we model the buyer-seller network as a undirected, bipartite graph $G = \langle (B,S), E, W\rangle$, where $B = \{b_1, b_2, \ldots, b_m\}$ denotes the list of buyers, $S = \{s_1, s_2, \ldots, s_n\}$ the list of sellers, $E \subseteq B \times S$ the set of edges, and $W : E \rightarrow \mathbb{R}^+$ the weights of the edges.

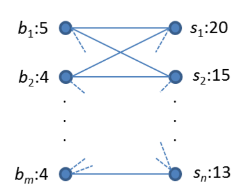


Fig. 1. WBM. The numbers are degree constraints.

For convenience, we slightly overload the notation and use an mn-dimensional vector to denote $\mathcal{E} = [e_{ij}]$, with $e_{ij} = 1$ indicating that there is an edge between buyer i and seller j and $e_{ij} = 0$ otherwise. Similarly, we use a vector to denote $W = [w_{ij}]$. If $e_{ij} = 0$, then $w_{ij} = 0$. When the subscripts are hard to read, as in the next section, we also use $W(i,j)$ to denote w_{ij}.

The weight of the edge between a buyer $i \in B$ and a seller $j \in S$, w_{ij}, reflects the profitability value if the buyer is matched to the seller. In practice, we may pre-process the bipartite graph based on various business models. For instance, we may order the buyers or sellers based on money spent and earned, recency of buys and sells, etc. In addition, we may constrain the edges from a buyer to a ranked range of sellers, so that the buyer is not matched to sellers well outside of her "tier". One possible such scenario is that we may need to match top buyers to top sellers, middle-tier buyers to middle-tier sellers, and so on, as this maximizes the chance of making both the buyers and sellers happy.

It is often desirable that only a certain number of buyers is matched to a seller, since a seller may be overwhelmed otherwise or because a seller needs to pay for the match. Similarly, it is not reasonable to match a buyer to a large number of sellers as the buyer might be annoyed later by many unsolicited requests. To avoid the problem, a good matching system should allow us to constrain the number of matches associated with individual buyers and sellers. In other words, we need to put degree constraints $D : B \cup S \rightarrow \mathbb{N}$ in the bipartite graph (Figure 1). We represent D with a $(m+n)$-dimensional column vector $D = [D(i)]^T$.

We denote by $X = [x_{ij}]^T$ the mn-dimensional column vector of 0-1 variables, with $x_{ij} = 1$ indicating buyer i is matched to seller j and $x_{ij} = 0$ otherwise. Then the degree-constrained matching problem is to *find the set of matches such that the total profit is maximized under the degree constraints*, i.e.,

$$\begin{aligned} \max_{X} \quad & WX \\ s.t. \quad & \mathbb{A}X(i) \leq D(i), \forall i, 1 \leq i \leq m+n \\ & x_{ij} \in \{0,1\}, \forall i,j, 1 \leq i \leq m, 1 \leq j \leq n, \end{aligned} \tag{1}$$

where matrix $\mathbb{A}$ is an $(m+n) \times mn$ matrix defined by

$$\underbrace{\begin{bmatrix} [e_{11},\ldots,e_{1n}] & & \\ & [e_{21},\ldots,e_{2n}] & \\ & \ddots & [e_{m1},\ldots,e_{mn}] \\ [e_{11},0,\ldots,0] & \cdots & [e_{m1},0,\ldots,0] \\ \ddots & \ddots & \ddots \\ [0,\ldots,0,e_{1n}] & \cdots & [0,\ldots,0,e_{mn}] \end{bmatrix}}_{(m+n)\times mn}. \tag{2}$$

The degree constraints are given by $\mathbb{A}X(i) \leq D(i)$, where $\mathbb{A}X(i)$ denotes the i-th element in (vector) $\mathbb{A}X$ and $D(i)$ the i-th element in D.

It has been shown that the WBM problem (as a *bipartite maximum weight b-matching* problem) could be reduced to the transportation problem in operations research [13], [14], and as such we can obtain an LP formulation that can be solved efficiently by modern solvers. Furthermore, note that matrix $\mathbb{A}$ is the incidence matrix of the buyer-seller bipartite graph, which can be proven to be totally unimodular [15], [16]. As [14], [17] show, the polyhedron $P = \{X : \mathbb{A}X \leq D\}$ is integral, and there is a polynomial time algorithm which finds an integral optimal solution.

In order to capture the conflict constraint, we now consider a natural extension of the model above. In the following, we focus on integrating the conflict between buyers. Note that the formulation and proposed algorithms (with little modification) also apply to the case where we simultaneously consider the conflict on both buyer and seller sides. We omit the latter to avoid repeated depiction.

In some scenarios, sellers will prefer a *diverse* list of buyers that avoids certain redundancies. For example, a seller might prefer that their list does not include more than one buyer from each household. Advertising a given merchandise to more than one potential buyer in a household is, in most cases, unnecessary. Formally, *we say that two buyers are in conflict with each other if matching them to the same seller is not desirable.* We will represent the presence of such conflicts between two buyers using conflict edges (Figure 2).

We call this problem conflict-aware constrained matching (CA-WBM). The goal is to *compute a maximum profit subgraph satisfying the degree constraints with the additional requirement that the number of conflict edges within a list of buyers matched to any particular seller is smaller than a threshold.*

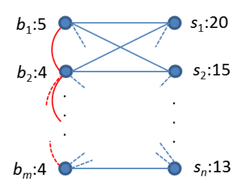


Fig. 2. CA-WBM. The numbers are degree constraints and the red edges represent conflicts.

To describe CA-WBM more precisely, the input to CA-WBM consists of the following information:

1) An undirected, weighted graph $G = \langle (B,S), E \cup C, W \rangle$ with $E \subseteq B \times S$, $C \subseteq B \times B$ and weights $W : E \rightarrow \mathbb{R}^+$;
2) Degree constraints $D : B \cup S \rightarrow \mathbb{N}$;
3) A conflict threshold t.

The goal in CA-WBM is to compute a maximum weight subgraph G' of G, $G' = ((B,S), E' \cup C', W)$, that satisfies the following two constraints:

1) For any i in $B \cup S$, $d_{G'}(i) \leq D(i)$.
2) For any k in S, $|\{(i,j)|(i,k),(j,k) \in E', (i,j) \in C'\}| \leq t$.

Here, $d_{G'}(i)$ denotes the degree of vertex i in the subgraph G'.

It is easy to see that WBM is a special case of CA-WBM. We next show that the constraints in CA-WBM pose a great challenge and significantly increase the complexity of the matching problem.

3 NP-HARDNESS RESULT FOR CA-WBM

In this section, we provide strong evidence that CA-WBM is highly unlikely to have an efficient (i.e, polynomial time) algorithm by showing that it is NP-hard.

Theorem 1. CA-WBM *is NP-hard.*

Proof. We give a polynomial-time reduction from the NP-hard problem REVENUE MAXIMIZATION IN INTERVAL SCHEDULING [18], [19], [20].

REVENUE MAXIMIZATION IN INTERVAL SCHEDULING (RMIS)

Instance: A set $M = \{m_1, m_2, \ldots, m_t\}$ of t machines and a set $J = \{j_1, j_2, \ldots, j_n\}$ of n jobs. For each job j in J, we are given three parameters: (1) $S(j)$, the set of machines on which j can be processed (2) $R(j,-)$, the set of possible revenues obtained when job j is processed on different machines (3) $I(j)$, the time interval during which job j must be processed.

Goal: Find a feasible schedule of a subset of jobs on the machines that maximizes the total revenue of the jobs scheduled.

We now describe a reduction from RMIS to CA-WBM. Given an instance I of the revenue maximization problem, construct a graph $G(I)$, which is an instance of CA-WBM. Define $G(I) = \langle (J, M), E \cup C, W\rangle$ with $E \subseteq J \times M$, $C \subseteq J \times J$ and weights $W : E \rightarrow \mathbb{R}^+$ as follows.

- $E = \{(j_k, m_l)|m_l \in S(j_k)\}$.
- $C = \{(j_k, j_l)|I(j_k) \cap I(j_l) \neq \emptyset\}$.
- $W(j_k, m_l) = R(j_k, m_l)$.
- $D(j_k) = 1$ for all k and $D(m_l) = n$ for all l.
- t = 0.

We now explain the reduction above. There is an edge between job j_k and machine m_l if m_l belongs to $S(j_k)$, the set of machines in which job j_k can be processed. There is a conflict edge between job j_k and job j_l if their time intervals for processing overlap. The weight on an edge (j_k, m_l) represents the revenue obtained if job j_k is processed on machine m_l. Since each job can be assigned to at most one machine, their degree constraints are set to 1. There is no constraint on the number of jobs assigned to any machine and hence their degree constraint is set to n. Finally, there must be no conflict between two jobs assigned to same machine. Therefore, t is set to 0.

It can be easily seen that an optimal solution for $G(I)$, an instance of CA-WBM yields an optimal solution for I, an instance of RMIS. In other words, a maximum weight subgraph of $G(I)$ satisfying the degree constraints and conflict constraints as described above exactly corresponds to a revenue maximizing schedule in I. Furthermore, we observe that the reduction above is a polynomial-time reduction. Therefore we conclude that CA-WBM is NP-hard. □

4 Algorithms for Solving CA-WBM

4.1 A SDP Algorithm for CA-WBM

In the previous section, we showed that CA-WBM is NP-hard. In this and following sections, we will design efficient algorithms for CA-WBM that provide high-quality solutions that are close to optimal.

Our first algorithm for CA-WBM is based on a semidefinite programming approach. To understand the motivation for this approach, recall that we described a IP formulation in Section 2. Using the terminology from Section 2 and 3, the conflict constraint can be described as follows:

$$\sum_{(j_k, j_l) \in C} x_{ki} x_{li} \leq t \;\forall\; i \in S \tag{3}$$

That is, the conflict constraint is quadratic. We use a single t for illustration purpose. In practice, different sellers can have different values of t. We will now show how to formulate CA-WBM as a semidefinite program. Define a $mn \times mn$ symmetric matrix $\mathbb{Y} = XX^T$ where X is as in Section 2. The CA-WBM problem can be described as

$$\begin{aligned} \max \quad & Trace(\mathbb{W}\mathbb{Y}) \\ s.t. \quad & Trace(\mathbb{D}_i^b \mathbb{Y}) \leq D(i), \forall\, i \in B \\ & Trace(\mathbb{D}_i^s \mathbb{Y}) \leq D(i), \forall\, i \in S \\ & Trace(\mathbb{C}_i \mathbb{Y}) \leq t, \forall\, i \in S \\ & \mathbb{Y} = XX^T \succeq 0 \end{aligned} \tag{4}$$

where $\mathbb{W}, \mathbb{D}_i^b, \mathbb{D}_i^s$ and $\mathbb{C}_i$ are suitably defined $mn \times mn$ symmetric matrices described below.

1) $\mathbb{W}$ is a diagonal matrix with diagonal weights w_{ij}.
2) $\mathbb{D}_i^b$ is diagonal matrix with a 1 for row indexed by (i, j) if $(i, j) \in E$ and 0 otherwise.
3) $\mathbb{D}_i^s$ is diagonal matrix with a 1 for row indexed by (k, i) if $(k, i) \in E$ and 0 otherwise.
4) Finally, $\mathbb{C}_i$ is a matrix with entries 1/2 and 0. An entry indexed by row (j, i) and column (k, i) is equal to 1/2 if $(j, i), (k, i) \in E$ and $(j, k) \in C$. It is 0 otherwise.

Our SDP based algorithm for CA-WBM is as follows:

1) Solve the semidefinite program relaxation to obtain optimal solution $\mathbb{Y}$. From now on, we refer to $Trace(\mathbb{W}\mathbb{Y})$ as the *SDP optimal*.
2) Using the Cholesky decomposition [21] of $\mathbb{Y}$, obtain the vectors x_{ij} corresponding to $\mathbb{Y}$.
3) Use a two-step rounding procedure, random projection [22] followed by threshold rounding, to obtain $0, 1$ values for x_{ij}. The result after this step is referred to as the *SDP with rounding*.

In step (1) of our SDP algorithm, the SDP described above is solved using a generic SDP solver. The output of step (1) is a semidefinite matrix $\mathbb{Y}$. In step (2) of our algorithm, we use a well-known fact that any semidefinite matrix $\mathbb{Y}$ can be written as $\mathbb{Y} = \mathbb{V}\mathbb{V}^T$ where $\mathbb{V}$ is a $mn \times mn$ lower triangular matrix. This decompostion is known as Cholesky decomposition of $\mathbb{Y}$ [21]. The columns of $\mathbb{V}$ give us a vector solution for the variables x_{ij}. Thus, the output of step (2) of our algorithm are mn vectors, one for each x_{ij}. These vectors correspond to the optimal solution of the SDP. In the last step of our algorithm, we convert the vectors x_{ij} to integral $\{0, 1\}$ values using a two-step rounding procedure. In the first step, we convert x_{ij}'s to fractional values by a random projection [22]. That is, we pick a random vector x of dimension mn by picking each of its coordinates from the normal distribution $N(0, 1)$ and define each $x_{i,j}$ as the length of its projection on to x. Finally, we sort x_{ij} and round each non-zero value to 1 provided doing so does not violate the degree constraints or the conflict constraints. Otherwise, we set it to 0.

4.2 ILP Formulation of CA-WBM

Solving the SDP formulation requires large physical memory to store the $mn \times mn$ matrix $\mathbb{Y}$. In practice (e.g., the eBay purchase graph of a certain category), large values for m (the number of buyers) and n (the number of sellers) inevitably restrict the applicability of the SDP approach. This limitation, however, can be alleviated if we could model CA-WBM as an integer linear programming (ILP) problem.

In order to achieve this goal, we introduce a new 0-1 variable $z_{i,(j,k)}$ to formulate Inequality 3 as a linear constraint. For each seller i, $z_{i,(k,l)}$ equals 1 if and only if there is a conflict edge between two buyers k and l, and both edge e_{ki} and e_{li} are recommended in the graph. Using the terminology from Sections 2 and 3, this constraint can be described as follows:

$$1 - x_{ki} - x_{li} + z_{i,(k,l)} \geq 0, \; \forall \, i \in S, \; \forall (k,l) \in C_i \quad (5)$$

$$x_{ki} + x_{li} - 2z_{i,(k,l)} \geq 0, \; \forall \, i \in S, \; \forall (k,l) \in C_i \quad (6)$$

$$\sum_{(k,l) \in C_i} z_{i,(k,l)} \leq t, \; \forall \, i \in S \quad (7)$$

In constraints (5), (6) and (7), C_i is defined as follows: $C_i = \{(k,l) \in C | (k,i) \in E \wedge (l,i) \in E\}$. That is, C_i represents the set of conflicts within the set of buyers linked to seller i.

The linear conflict constraints can be easily incorporated into Problem 1. Let c denote the total number of conflict constraints with respect to all sellers in the graph. Then we can obtain a linear programming formulation, where $X = [x_{ij}]^T$ is a $(mn + c)$-dimensional column vector of 0-1 variables. In addition, matrix $\mathbb{A}$ and vectors W and D can be changed accordingly.

By eliminating the need to store large dimensional matrices, now we can use an ILP solver to tackle CA-WBM problems of larger sizes. Since obtaining an integer solution in CA-WBM is NP-hard, in order to further improve efficiency, we use a rounding procedure after solving the linear program relaxation. Our LP based algorithm for CA-WBM is as follows:

1) Solve the linear program relaxation to obtain optimal solution X.
2) Sort the first mn elements of X from largest to smallest. We round each non-zero value to 1 provided doing so does not violate the degree constraints or the conflict constraints. Otherwise, we set it to 0. The result after this step is referred to as the *LP relaxation with rounding*.

4.3 A Greedy Algorithm for CA-WBM

In this section, we describe and study the performance of a simple greedy algorithm for this problem. This algorithm has the advantage that it is highly scalable and provides good quality solutions in practice.

The greedy algorithm, denoted as GREEDY, for CA-WBM is as follows:

1) Sort all the edges in E by weights from largest to smallest.
2) To construct the maximum weight subgraph G', consider every edge in the sorted list. Add this edge to G' if doing so does not violate any degree constraint or conflict constraint.
3) Continue until we reach the end of the sorted list.

We will now prove a theoretical guarantee on the performance of GREEDY.

Theorem 2. *Let* $d = \max_{v \in B} |\{(v,v')|(v,v') \in C\}|$. *Algorithm GREEDY is a* $(2+d)$*- approximation algorithm.*

Proof. We use the concept of a *k-extendible system* to provide performance guarantees of a greedy algorithm. Mestre introduced the notion of a k-extendible system in his study of the performance of the greedy technique as an approximation algorithm [23].

Definition 1 (k-Extendible System [23]). *Let U be a finite set and $\mathcal{F}$, $\mathcal{F} \subseteq 2^U$, be a collection of subsets of U. Set system $(U, \mathcal{F})$ is called a k-*extendible system *if it satisfies the following properties:*

1) Downward-closure*: If $A \subseteq B$ and $B \in \mathcal{F}$, then $A \in \mathcal{F}$.*
2) Exchange*: Let $A, B \in \mathcal{F}$ with $A \subseteq B$, and let $x \in U - B$ be such that $A \cup \{x\} \in \mathcal{F}$. Then there exists $Y \subseteq B - A$, $|Y| \leq k$, such that $(B - Y) \cup \{x\} \in \mathcal{F}$. In other words, let us start with any choice of two sets A and B such that B is an extension of A. Suppose that there is an element x such that the set A with x added to it also belongs to $\mathcal{F}$. Then we will be able to find a subset Y inside B of size at most k such that if we remove the elements of Y from B and add the element x to the resulting set, it will also belong to the collection $\mathcal{F}$.*

Informally, Mestre showed that if the set of all feasible solutions forms a k-extendible system, algorithm GREEDY gives a k-approximation algorithm. That is, on any instance, the solution output by GREEDY differs from the optimal solution by a multiplicative factor of at most k. We now state his result more formally.

Theorem 3 (Mestre [23]). *Let $(U, \mathcal{F})$ be a k-extendible system for some k. Let $W : U \rightarrow I\!R^+$ be a positive weight function on U. The greedy algorithm gives a k-approximation algorithm for the optimization problem that asks to determine $\max_{F \in \mathcal{F}} W(F)$ where $W(F) = \sum_{s \in F} W(s)$ for any $F \in \mathcal{F}$.*

To apply this result to our problem, we will check that the set of all feasible solutions to CA-WBM forms a $(2 + d)$-extendible system. For the CA-WBM problem, $U = E$ and $\mathcal{F}$ be the set of all subgraphs of G satisfying the degree and conflict constraints. Then it is easy to see that $(U, \mathcal{F})$ is downward closed. That is, removing an edge from a feasible solution H will always result in be a feasible solution as this will not cause any violation of constraints.

For the exchange property, consider the case when a new edge $e = (u,v)$, $u \in B$ and $v \in S$, is added to a feasible solution H. We make two observations: (1) Adding e could result in violation of degree constraint at u and v. However, this can be rectified by removing two other edges, one incident on u and other incident on v; (2) Adding e could result in a violation of the conflict constraint at v. Rectifying this could require removing at most d edges where $d = \max_{v \in B} |\{(v,v')|(v,v') \in C\}|$. Therefore, we obtain a $(2 + d)$-extendible system. ☐

We remark that this analysis is worst-case. In practice, GREEDY shows far superior performance, as demonstrated in our later test with real-world datasets.

5 Online CA-WBM and a Randomized Algorithm

In real-world scenarios, we may not know all buyers/sellers in advance. In online business, the set of sellers is relatively stable compared to the set of buyers. As such, we mainly focus on one typical problem where new buyers join the system as time goes. Other variations of the problem, e.g., both new buyers and new sellers joining the system, could be studied with the idea presented in this section, but their analysis remains challenging.

In the online version of CA-WBM, an algorithm for the problem only knows the set S when it starts, and the sequence of vertices $b_1, b_2, ..., b_m$ arrives online one by one. When a vertex $b_i \in B$ arrives, all edges incident to b_i, as well as their weights, and conflict edges associated with other buyer vertices which arrived earlier, are revealed. The algorithm should immediately make recommendation to b_i and cannot change the recommendation at a later time.

Ting and Xiang [24] proposed a near optimal randomized algorithm for online bipartite maximum weighted b-matching. According to their definition, while each fixed vertex can be matched to at most b arrival vertices, each arrival vertex can be matched to only one fixed vertex. We extend their algorithm and analysis by incorporating the conflict constraint and allowing the degree constraint of each arrival vertex to be larger than 1.

5.1 Assumptions and Settings

5.1.1 Seller Vertex

Denote the degree constraint of a seller s by $D(s)$ and we assume $D(s) \geq 1$. To ease the analysis, we make $D(s)$ copies of each seller vertex s and form them as a group. An example is shown in Figure 3. In addition, recall that in CA-WBM, the number of conflict edges within a list of buyers matched to any particular seller is smaller than a threshold t.

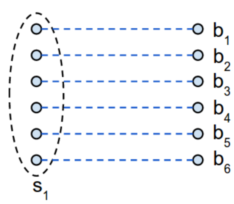


Fig. 3. An example: copies of a seller node s_1. $(b_1, ..., b_6)$ are adjacent buyer nodes of s_1. Assume that the degree constraint of s_1 is 6 and the conflict threshold t is 0. Each copy of s_1 can be matched to only one buyer node. The 6 copies as a whole cannot be matched to any conflict pair of buyers.

5.1.2 Buyer Vertex

Let $D(b)$ denote the degree constraint of a buyer vertex b. In the following, we first study the special case where each buyer can be matched to only one seller, i.e., the degree constraint of each buyer vertex is 1. Later, we will extend our result to the general case where the degree constraint of each buyer is more than one ($D(b) > 1$).

5.1.3 Edge Weight

In addition to the set S, we assume that the maximum weight of edges ($w_{\max}$) can be estimated and is known to the algorithm. In the buyer to seller matching, this is a reasonable assumption since the system can estimate the maximum amount of money based on transaction records from buyers and sellers.

5.2 The Randomized Algorithm for $D(b) = 1$

In this special case, the degree constraint of each buyer vertex is 1. Inspired by the algorithm in [24], we propose a randomized algorithm for online CA-WBM, ***Randomized-CA-WBM***, that takes the constraint on conflicts into account.

Algorithm 1: ***Randomized-CA-WBM***

Input: A seller set S, the maximum edge weight $w_{\max}$
Output: Generate buyer to seller recommendation (the matched edge) on the fly

Let $g = \lceil \ln(1 + w_{\max}) \rceil$, choose an integer k uniformly from $\{0, 1, 2, ..., g-1\}$;
Set $\tau = e^k$;
while *a new vertex* $b \in B$ *arrives* **do**
  T = $\{\hat{s} \mid \hat{s}$ is a copy vertex of s, which is a seller incident to b in S and $w((s, b)) \geq \tau\}$;
  if $T = \emptyset$ **then**
    leave b unmatched forever;
  else
    match b to an arbitrary vertex in T, if doing so does not violate its conflict constraint (When $T \neq \emptyset$, a vertex b may not be matched to any vertex due to the conflict constraint.)

The performance of an online algorithm is often analysed with competitive analysis, proposed by Sleator and Tarjan [25]. In our weight maximization problem, let $\mathcal{A}$ denote a randomized online algorithm, let σ denote the arrival sequence, and let $\mathbb{E}[w(\mathcal{A}(\sigma)]$ be the expected solution output by $\mathcal{A}$ when processing the arrival sequence σ. Let $w(\mathcal{OPT}(\sigma))$ denote the output of the optimal offline algorithm $\mathcal{OPT}$ when processing σ. We say that a randomized online algorithm $\mathcal{A}$ is R-*competitive*, if the ratio of $\mathbb{E}[w(\mathcal{A}(\sigma)]$ to $w(\mathcal{OPT}(\sigma))$ is at least $1/R$,

$$\frac{\mathbb{E}[w(\mathcal{A}(\sigma)]}{w(\mathcal{OPT}(\sigma))} \geq \frac{1}{R}$$

for all arrival sequences σ. The smallest such R is called the competitive ratio of $\mathcal{A}$. To evaluate ***Randomized-CA-WBM***, we compare its performance to that of the optimal offline algorithm. Theorem 4 gives the upper bound of the competitive ratio of ***Randomized-CA-WBM***.

Theorem 4. ***Randomized-CA-WBM*** *achieves a competitive ratio of* $(\alpha+1)e\lceil \ln(1+w_{\max}) \rceil$, *where* $\alpha = \max(d_1 - 1, d_2)$, *and* $d_1 = \max_{s \in S} D(s)$ *and* $d_2 = \max_{v \in B} |\{(v, v') | (v, v') \in C\}|$.

In order to prove Theorem 4, we extend and adapt the analysis in [24] so that it applies to Algorithm 1. Let M denote the set of edges output by ***Randomized-CA-WBM*** for the graph $G = \langle (S, B), E \cup C, W \rangle$ with any arrival sequence on B, and let $S(M) = \{s \in S \mid \exists b \text{ s.t.} (s, b) \in M\}$ be the set of seller ends of the edges in M. Let $w(M) = \sum_{(s,b) \in M} w((s, b))$ be the total weight of edges in M, where $w((s, b))$ is the weight of the edge (s, b). For any $i \geq 0$, let $M_{\geq e^i}$ be the result if the threshold τ is e^i. Then the expectation of $w(M)$ for any arrival sequence is $\mathbb{E}[w(M)] = \sum_{0 \leq i \leq g-1} w(M_{\geq e^i}) \frac{1}{g}$.

Denote O as the optimal maximum weight subgraph of G. Let $O_{[e^i,e^{i+1})} = \{x \in O | w(x) \in [e^i, e^{i+1})\}$ be the set of edges in O, with each edge's weight $w(x) \in [e^i, e^{i+1})$.

Lemma 1. *$\forall i \in \{0, 1, ..., g-1\}$, $|S(O_{[e^i,e^{i+1})}) - S(M_{\geq e^i})| \leq \alpha\,|S(M_{\geq e^i})|$, where $\alpha = \max(d_1 - 1, d_2)$, and $d_1 = \max_{s \in S} D(s)$ and $d_2 = \max_{v \in B} |\{(v, v')|(v, v') \in C\}|$.*

Proof. Consider a copy vertex $\hat{s}$ of any seller s, and $\hat{s} \in S(O_{[e^i,e^{i+1})}) - S(M_{\geq e^i})$. Note that $\hat{s} \in S(O_{[e^i,e^{i+1})})$ but $\hat{s} \notin S(M_{\geq e^i})$. The refusal of matching $\hat{s}$ to a valid buyer vertex b (since $w((\hat{s}, b)) \geq e^i$) in the result output by ***Randomized-CA-WBM*** suggests two possible cases,

1) b is matched to another $\hat{s}' \in S(M_{\geq e^i})$
2) b is unmatched due to the conflict constraint

For every vertex $\hat{s} \in S(O_{[e^i,e^{i+1})}) - S(M_{\geq e^i})$ of the first case, each of them can be mapped to a unique vertex $\hat{s}' \in S(M_{\geq e^i})$.

For the second case, we can analyze the matching result of the worst case for a certain seller s_1, as shown in Figure 4.

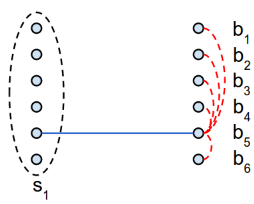


Fig. 4. The worst case: if one of the copies is matched to a buyer, none of the rest can be matched due to the conflict threshold (which is 0 in the worst case). b_5 arrives first and it is in conflict with other buyers.

Let $degree_{s_1}$ be the degree constraint of s_1 and $conflict_{(s_1,b)}$ be the largest possible number of conflict edges associated with any s_1's neighboring buyer node. For example, in Figure 4, $degree_{s_1} = 6$ and $conflict_{(s_1,b)} = 5$ (b_5 is in conflict with 5 buyer nodes).

For a seller s_1, if $(degree_{s_1} - 1) \leq conflict_{(s_1,b)}$, the largest possible number of unmatched copies is $(degree_{s_1} - 1)$; otherwise, the number is $(conflict_{(s_1,b)})$.

If we consider the case when $d_1 = \max_{s \in S} D(s)$ and $d_2 = \max_{v \in B} |\{(v, v')|(v, v') \in C\}|$ are known ahead of time, up to $\max(d_1 - 1, d_2)$ vertices in $S(O_{[e^i,e^{i+1})}) - S(M_{\geq e^i})$ can be mapped to a unique vertex $s' \in S(M_{\geq e^i})$; then we have the lemma. □

We remark that this upper bound is tight. For example, in the worst case shown in Figure 4, $d_1 = 6$ and $d_2 = 5$. Suppose that in the optimal solution, b_5 is not matched to s_1, then all other buyer vertices are able to be matched. Thus, $|S(O_{[e^i,e^{i+1})}) - S(M_{\geq e^i})| = 5$, $|S(M_{\geq e^i})| = 1$, and $\max(d_1 - 1, d_2) = 5$. In addition, if the conflict threshold t becomes larger than 0, more copies of s_1 will be allowed to be matched to buyer vertices. Then the number of unmatched copies is less than $\max(d_1 - 1, d_2)$. Therefore, the upper bound still holds.

Lemma 2. *$\forall i \in \{0, 1, ..., g-1\}$, $w(M_{\geq e^i}) \geq \frac{1}{(\alpha+1)e} w(O_{[e^i,e^{i+1})})$, where $\alpha = \max(d_1 - 1, d_2)$.*

Proof. Since each copy ($\hat{s}$) of any seller vertex s can be matched to only one buyer vertex b, the number of matched seller copies is equal to the number of matched edges. By Lemma 1, for any $i \in \{0, 1, ..., g-1\}$, we have

$$\begin{aligned}
|O_{[e^i,e^{i+1})}| &= |S(O_{[e^i,e^{i+1})})| \\
&= |S(O_{[e^i,e^{i+1})}) \cap S(M_{\geq e^i})| + \\
&\quad |S(O_{[e^i,e^{i+1})}) - S(M_{\geq e^i})| \\
&\leq (\max(d_1 - 1, d_2) + 1)|S(M_{\geq e^i})| \\
&= (\max(d_1 - 1, d_2) + 1)|M_{\geq e^i}|
\end{aligned}$$

then

$$\begin{aligned}
w(M_{\geq e^i}) \geq e^i |M_{\geq e^i}| &\geq \frac{e^i}{\max(d_1 - 1, d_2) + 1} |O_{[e^i,e^{i+1})}| \\
&\geq \frac{1}{(\max(d_1 - 1, d_2) + 1)e} w(O_{[e^i,e^{i+1})})
\end{aligned}$$

the lemma follows. □

Now, we can prove Theorem 4 using Lemma 2.

Proof.

$$\begin{aligned}
\mathbb{E}[w(M)] &= \sum\nolimits_{0 \leq i \leq g-1} w(M_{\geq e^i}) \frac{1}{g} \\
&\geq \frac{1}{(\alpha+1)eg} \sum\nolimits_{0 \leq i \leq g-1} w(O_{[e^i,e^{i+1})}) \\
&= \frac{1}{(\alpha+1)eg} w(O).
\end{aligned}$$

□

5.3 The Randomized Algorithm for $D(b) \geq 1$

In general, each buyer vertex can be matched to more than one seller. By making copies of each buyer vertex upon arrival, Algorithm 1 for $D(b) = 1$ can be shown to adapt well with little modification. Let $D(b_1)$ denote the degree constraint of a buyer vertex b_1. Upon arrival of this buyer, the algorithm immediately makes $D(b_1)$ copies, $(b_{1,1}, ..., b_{1,D(b)})$. Instead of grouping the copies, the algorithm considers one copy at a time. We require that each buyer copy can be matched to at most one seller, and two copies of the same buyer cannot be matched to the same seller. Since the input of CA-WBM remain the same (i.e., a set of sellers S and the maximum edge weight $w_{\max}$), the theoretical analysis also applies.

5.4 The Lower Bound on Competitive Ratio

Ting et al. [24] proved that for the online maximum weighted b-matching problem, no randomized algorithm can be better than $\frac{\lceil \log_2(w_{\max}+1) \rceil + 1}{2}$-*competitive*. Since their problem, online maximum weighted b-matching problem, is a special case of our problem, we get the same lower bound as in [24] showing that the performance of our algorithm, Randomized CA-WBM, is near optimal.

Theorem 5. *For online CA-WBM, no randomized algorithm can achieve a competitive ratio better than $\frac{\lceil \log_2(w_{\max}+1) \rceil + 1}{2}$.*

6 EXPERIMENTAL EVALUATION

In this section, our main focus is to illustrate the proposed algorithms' optimality and scalability for CA-WBM and online CA-WBM. Specifically,

- CA-WBM. Since CA-WBM is proven to be NP-hard, we mainly focus on the performance of the proposed approximate algorithms. For the SDP formulation, we evaluate its performance by comparing the optimal solution and the solution obtained by the rounding procedure for several combinations of degree and conflict constraints. For the ILP formulation, we perform experiments to compare the integral solution with the result of the linear programming (LP) relaxation with rounding on much larger graphs. Finally, we demonstrate the scalability of our greedy algorithm.
- Online CA-WBM. We evaluate the proposed randomized algorithm by repeating experiments for different random arrival sequences. We calculate the competitive ratio for each run and compare the average behavior to the upper bound.

We conducted comprehensive experiments with eBay's transactional data provided by Terapeak. The original data consists of three-month transactions across all categories of eBay Canada and eBay US in 2013. We used the data of a specific category (Cell phones and Accessories) from both eBay Canada and eBay US. In order to test the algorithms on datasets of different scales, we created three datasets for evaluation: a small-scale synthetic dataset (eBay Canada), a moderate-scale synthetic dataset (eBay Canada), and a large-scale real-world dataset (eBay US).

The datasets of eBay Canada are called "*synthetic*" because we changed the original graph structure with an imputation step to generate bipartite graphs of different sizes. The details about how we created synthetic datasets will be explained in Sections 6.1, 6.1.1 and 6.1.2. The real-world eBay US dataset contains purchase information of transactions between buyers and sellers in the category of cell phones and accessories. Note that even though we used data from Terapeak, similar transaction data can also be collected via the eBay API[3]. The basic information of each dataset is summarized in Table 1.

Experiments with the small-scale and moderate-scale datasets were run on a 64-bit Ubuntu 12.04 desktop of 3.40GHz * 8 Intel Core i7 CPU and 3.8 GB memory. Experiments with the large-scale dataset were run on a similar desktop with 12 GB memory.

6.1 CA-WBM

To create synthetic datasets for small-scale and moderate-scale experiments, we use eBay Canada data and firstly sort buyers, as well as sellers, by the total monetary purchases and total profits, respectively. Adding edges between sorted buyers and sellers, we can create bipartite graphs of different density settings. In the experiments, edges are generated in a manner that each seller has the same number of connected buyers. For example, if the density is set to 0.5%, each seller will be connected to roughly 90 buyers (calculated from the synthetic-moderate data in Table 1). The first seller (top ranked) is connected with the first 90 buyers (from 1 to 90), and the second seller is connected with buyers from 11 to 100, and so on. The weight of an edge is the sum of the buyer's total monetary purchases (on all sellers within the category) and the seller's total profits (from all buyers within the category). Of course, any monotone weight function can be used. For each buyer (seller) in the bipartite graph, the degree constraint ratio is the proportion of the maximum number of matched sellers (buyers) among all candidates. Every node (buyer or seller) in the experiment has the same degree constraint ratio.

We also introduce conflict edges between pairs of buyers. The requirement could be set by sellers who do not want to be recommended many buyers that are in conflict with each other, e.g., buyers who share similar features such as the same address. By limiting the number of conflicting buyers, we could promote diversity of the matching results for sellers.

In the following experiments, we randomly create conflicts between buyers according to different conflict pair ratios. This ratio is the number of buyer pairs in conflict divided by the total number of buyer pairs.

6.1.1 CA-WBM on Small-scale Synthetic Datasets

As discussed in Section 2, the SDP formulation of *CA-WBM* can be solved by a SDP solver plus a rounding procedure. We use SDPT3 [26], [27] as the SDP solver. For ILP, we use a fast linear programming (LP) solver, Gurobi[4] for Matlab, to solve *CA-WBM* at different scales.

The applicability of the SDP approach for *CA-WBM* severely suffers from the limitation of physical memory, i.e., the need to store a large-dimensional matrix [28]. In the real-world scenario, the dimension of the matrix can easily reach hundreds of thousands or more, which is beyond the capacity of a stand-alone machine. Due to this reason, we can only test the performance of the SDP approach in small-scale datasets.

We create a subgraph of small size consisting of top 5 sellers and top 26 buyers (ranked by money). Each seller is connected with 10 buyers so that each buyer can be assigned to multiple sellers. Specifically, the first seller is connected with buyers from 1 to 10, the second seller is connected with buyers from 5 to 14, and so on. We use two types of edge weight, money and node rank. Money weight is computed in the same way described earlier, i.e., the sum of the buyer's total monetary purchases (on all sellers within the category) and the seller's total profits (from all buyers within the category). Of course, any monotone weight function can be used. Rank weight equals the multiplication of a big constant (the total number of buyer and seller nodes in the entire graph, $20,626$ in our case) and the reciprocal of the sum of the buyer's rank and the seller's rank.

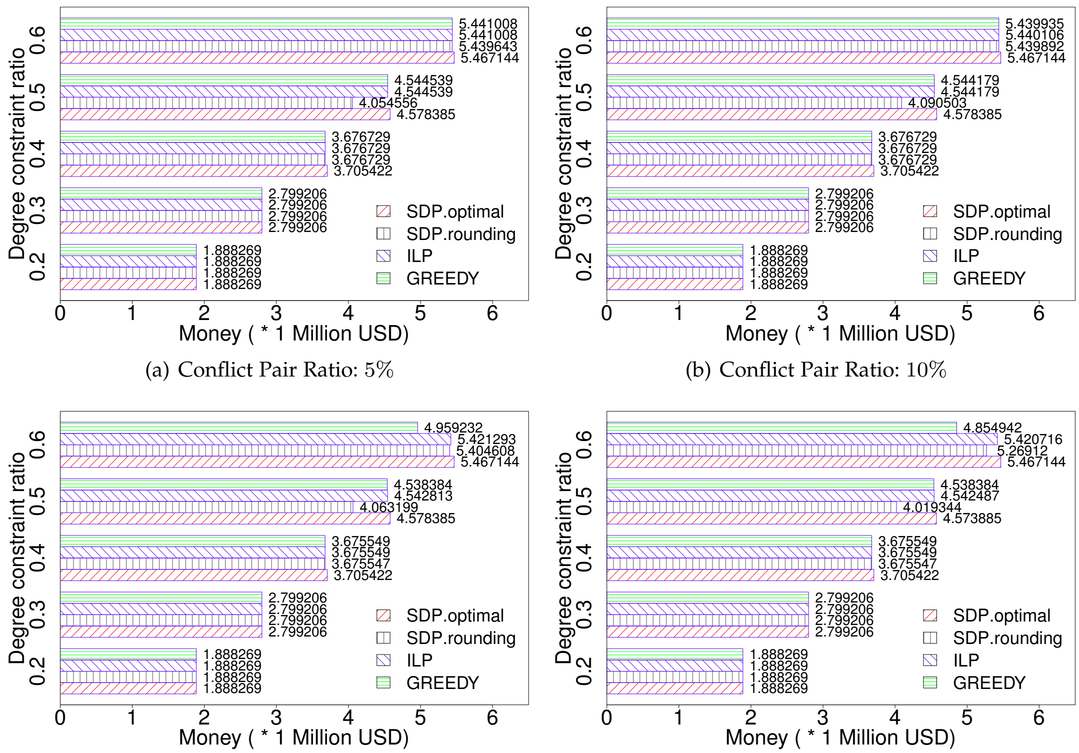
We create different number of conflict buyer pairs by randomly sampling the number of buyer pairs with the following percentages of total possible buyer pairs, $\{5\%, 10\%, 15\%, 20\%\}$. We set different degree constraint ratios for each node, $\{20\%, 30\%, 40\%, 50\%, 60\%\}$. In addition, we use a constant ("1") for each seller's conflict constraint, which means each seller can at most accept one conflict

3. https://go.developer.ebay.com/

4. http://www.gurobi.com/

TABLE 1
Basic Information of Synthetic and Real-world Datasets

	synthetic-small	synthetic-moderate	eBay US
Algorithms tested	SDP optimal, SDP with rounding, ILP, GREEDY	ILP, LP with rounding, GREEDY	GREEDY
Number of buyers	26	18,742	5,751,334
Number of sellers	5	1,884	126,101
Number of edges	50	56,520	11,387,517



(a) Conflict Pair Ratio: 5%

(b) Conflict Pair Ratio: 10%

(c) Conflict Pair Ratio: 15%

(d) Conflict Pair Ratio: 20%

Fig. 5. Money solution of different conflict pair ratios. When degree constraint ratio and conflict pair ratio are both low, GREEDY shows close to optimal solutions while SDP with rounding is weaker. Values of each bar are actual solutions. All values and running times can be found in Figure 5 of the supplementary material of this paper.

buyer pair. We use this constant to amplify the impact of conflict constraint on the small-scale subgraph. For SDP with rounding, we run the randomized rounding procedure 20 times and take the best solution.

Figure 5 and Figure 6 depict the comparison of different approaches, including SDP based approaches (Section 4.1), ILP (Section 4.2), and GREEDY (Section 4.3), for the money weight and rank weight, respectively. The result obtained by ILP is the integral optimal.

In Figure 5, we observe that regardless of conflict pair ratio, solutions of the different methods are very similar when degree constraint ratio is smaller than 60%, with SDP with rounding being slightly weaker than others. The reason is that when the degree constraint is tight, the conflict constraint of each seller is less likely to be activated, i.e., chances are rare for multiple conflict buyers to be matched to a seller. The difference arises when the degree constraint ratio and the conflict pair ratio are both weak and the higher conflict pair ratio results in larger performance drop for both SDP with rounding and GREEDY (e.g., comparing the sets of bars of 60% degree constraint ratio in (c) and (d)).

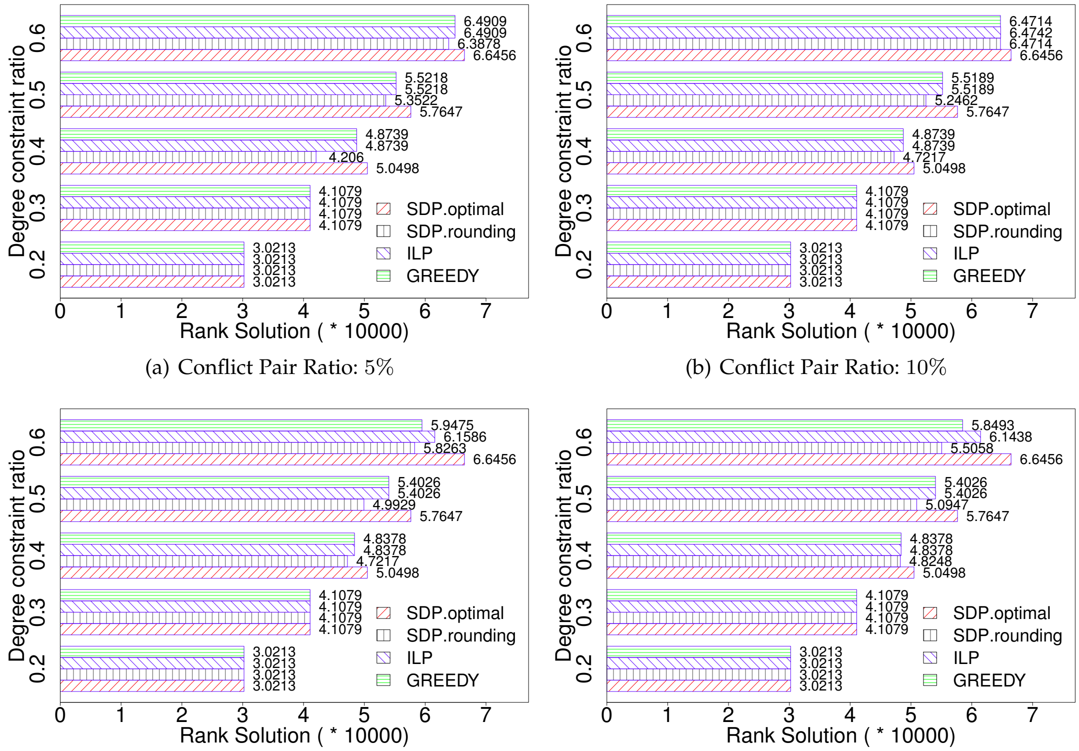
Figure 6 shows a similar performance change trend for the approaches. For example, the solution difference becomes larger when degree constraints are weaker and the performance of SDP with rounding and GREEDY gradually decreases as the conflict pair ratio increases. Meanwhile, we also observe that the performance of GREEDY is always superior to that of SDP with rounding.

We also observe that both SDP with rounding and GREEDY achieve close to optimal solutions (compared to the result of ILP). Specifically in the experiments, GREEDY exhibits a superior performance compared to the theoretical analysis.

6.1.2 CA-WBM on Moderate-scale Synthetic Datasets

ILP formulation enables us to take full advantage of the LP solver (Gurobi) to solve CA-WBM problems with larger sizes. In this section, we perform moderate-scale experiments to compare solutions of different methods, i.e., ILP (Section 4.2), LP relaxation with rounding (Section 4.2) and GREEDY (Section 4.3).

We create a 0.16%-density bipartite graph using the same method described in Section 6.1. Note that the density of the corresponding real-world buyer-seller graph is



(a) Conflict Pair Ratio: 5%

(b) Conflict Pair Ratio: 10%

(c) Conflict Pair Ratio: 15%

(d) Conflict Pair Ratio: 20%

Fig. 6. Rank solution of different conflict pair ratios. GREEDY achieves close to optimal performance in all cases. The solutions obtained by SDP with rounding are slightly worse compared to GREEDY. Values of each bar are actual solutions. All values and running times can be found in Figure 5 of the supplementary material of this paper.

0.016%. The full graph consists of 18742 buyers, 1884 sellers and 56520 edges. We also extract three subsets of different sizes from the full graph, i.e., using 25%, 50% and 75% of the total number of edges (the number of buyer and seller nodes decreases accordingly). The degree constraint ratio, conflict pair ratio are 50% and 10%, respectively. The conflict threshold t (refer to Section 2) for each seller is set to be 50% of the total number of conflicting buyer pairs associated to the seller. Figure 7 shows the solution comparison of different methods on different datasets.

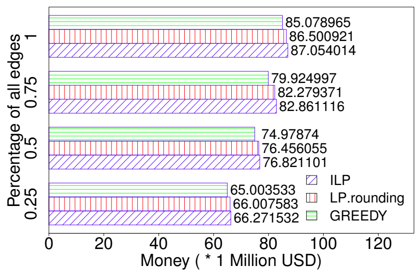
Figure 7 shows very promising results for LP relaxation with rounding and GREEDY algorithms on both money weights and rank weights; they are only slightly worse than the optimal integral solution obtained by ILP. Comparing to the SDP experiments, this experiment also verifies the effectiveness of both LP relaxation with rounding and GREEDY on moderate datasets because the graph we use in this section is considerably larger than the one used for SDP experiments. The density (0.16%) is also 10 times larger than the corresponding real-world buyer-seller graph (0.016%). Therefore, our ILP formulation improves the scalability of solving moderate-scale CA-WBM problems.

6.1.3 CA-WBM on Large-scale Real-world Dataset

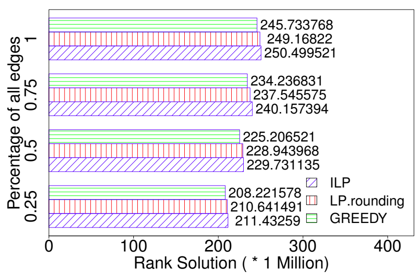
When the size of the graph grows larger, however, the ILP formulation and the corresponding LP relaxation become untameable with existing ILP/LP solvers due to the enormous number of variables (as shown in Table 2) and prohibitive computational requirement. In this case, GREEDY shows its most important advantage that it scales very well when applied to even larger datasets. We run GREEDY on the large-scale eBay US dataset, which contains $5,751,334$ buyers, $126,101$ sellers and $11,387,517$ edges. The weight of edge between a buyer and a seller represents the total amount of money spent by the buyer on this seller. To show its scalability as graph size increases, we extract three subsets of different sizes from the full graph, i.e., using 25%, 50% and 75% of the total number of edges. The degree constraint ratio, conflict pair ratio are 20% and 1%, respectively. A larger conflict pair ratio results in more z variables. The conflict threshold t (refer to Section 2) for each seller is set to be 20% of the total number of conflicting buyer pairs associated to the seller. Table 3 summarizes the statistical information of each subset. The running time of GREEDY for each of them is shown in Figure 8.

TABLE 2
Problem Size of LP Formulation for Each Subset of eBay US

	25%	50%	75%	100%
# constraints	$6,022,924$	$6,860,587$	$8,488,620$	$10,043,227$
# variables	$2,935,592$	$6,183,430$	$9,843,523$	$13,467,108$
# z variables	$88,712$	$489,671$	$1,302,885$	$2,079,591$



(a) Money Solution



(b) Rank Solution

Fig. 7. ILP experiments of CA-WBM on moderate-scale datasets. Both LP relaxation with rounding and GREEDY achieve close to optimal solutions. Values of each bar are actual solutions. All values and running times can be found in Figure 6 of the supplementary material of this paper.

TABLE 3
Basic Information of Three Subsets of eBay US

	25%	50%	75%	100%
# buyers	$1,574,114$	$2,988,717$	$4,300,322$	$5,751,334$
# sellers	$66,751$	$90,925$	$109,511$	$126,101$
# edges	$2,846,880$	$5,693,759$	$8,540,638$	$11,387,517$

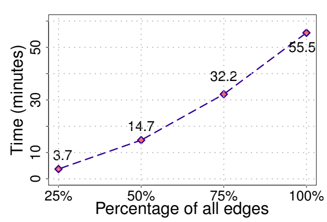


Fig. 8. Greedy algorithm on large-scale datasets showing its scalability.

In Figure 8, the running time increases nearly linearly, and it only requires less than one hour to get a solution using a single desktop computer when the number of edges is considerably as large as $11,387,517$.

6.2 Online CA-WBM

We use the same 0.16%-density bipartite graph as in Section 6.1.2 and we retain all edges (full size). The settings for degree constraint ratio, conflict pair ratio and conflict threshold t also remain the same. We repeat $10,000$ tests for money weight and rank weight, respectively. In each run, we randomly generate the arrival sequence for buyers and we compute its competitive ratio after it finishes. Figure 9 shows the box plot of competitive ratios for money and rank weight. In both cases, the majority of competitive ratios are located under 4. The average competitive ratio out of 10000 tests is 3.25 for money weight, and 3.63 for rank weight, respectively. Therefore, on average, the online algorithm indeed shows promising performance.

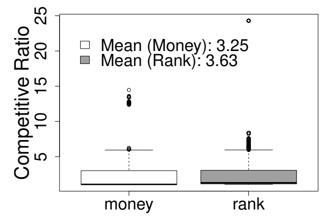


Fig. 9. Competitive ratios of 10000 runs for Alg. 1. For money weight, first quartile, median and third quartile are 1.047, 1.067 and 3.012, respectively. For rank weight, the values are 1.14, 1.296 and 3.068, respectively.

7 Related Work

CA-WBM is a non-trivial extension of the classical weighted bipartite b-matching (WBM), a fundamental problem in computer science. The broader applicability of CA-WBM and the difficulties of solving the problem are discussed in the supplementary material of this paper.

WBM finds applications in different scientific fields, including resource allocation [1], [2], scheduling [3], Internet advertising [4], [9], [10], [12], [29] and recommender systems [5], [30]. In addition, weighted b-matching in general graphs has been shown to be useful for a wide range of machine learning tasks, including classification [31], [32], structured prediction models [33], spectral clustering [34], graph embedding [35], semi-supervised learning [36], and manifold learning [37].

Since WBM can be reduced to the transportation problem (or the maximum flow problem) in operations research [13], [14], it can be solved in polynomial time by a number of classical algorithms. Denoting the number of vertices as n and the number of edges as m, WBM can solved in $O(m\sqrt{n})$ time using Dinic's algorithm [38], [39]. Similarly, Hopcroft-Karp algorithm [40] finds a maximum matching in bipartite graphs in $O(n^{2.5})$ time.

Approximation algorithms for WBM also exist to improve the efficiency on large-scale real-world datasets, which often have millions of vertices and edges. In a centralized setting, Mestre [41] showed that a greedy algorithm can achieve 2-approximation in $O(m\log n)$ time. The author also provided two linear time approximation algorithms: a 2-approximation algorithm that runs in O(bm) time and a $(\frac{2}{3} - \epsilon)$-approximation algorithm that runs in expected $O(bm\log\frac{1}{\epsilon})$ time. In a distributed setting, Hoepman [42] showed that the greedy algorithm can be easily distributed and achieves 2-approximation in $O(m)$ time. Morales et al. [43] showed how to implement the similar greedy algorithm in MapReduce paradigm. None of the above algorithms includes conflict constraints. Recently, Manshadi et al. [44] modelled the generalized matching problem as linear program and proposed a distributed algorithm to cope with large-scale datasets. The algorithm allows for a small violation of lower- and upper-bound constraints in the optimization. Since their focus was on the scalability issue of linear program, they did not consider and analyze the impact of conflict constraints either.

From the point of view of practical applications, CA-WBM is related to the task of constraint-based recommendation ([45], [46], [47], [48], [49], [50], [51]). The first three works considered constraints on item features (attributes) and user preference relaxation [47], thus they did not study the same type of constraints as we do. The rest of works are more closely related to our problem but they did not consider conflict constraints either. Karimzadehgan et al. [48], [49], [52] studied the problem of optimizing the review assignments of scientific papers. They employed constraints on the quota of papers each reviewer is assigned. However, differently from our approach, in their optimization setup, matching of reviewers with a paper was done based on matching of multiple aspects of expertise. Xie, Lakshmanan, and Wood in [50] studied the problem of composite recommendations, where each recommendation comprises a set of items. They also considered constraints including the number of items that can be recommended to a user. Their objective, however, was to minimize the cost of a recommended set of items when each item has a price to be paid. Parameswaran, Venetis, and Garcia-Molina in [51] studied the problem of course recommendations with course requirement constraints. Similarly as [50], the goal of [51] was to come up with set recommendations. However, the challenge they addressed was the modeling of complex academic requirements (e.g., take 2 out of a set of 5 math courses to meet the degree requirement). Such constraints are different from those that we consider in this paper. Diversity is also a relevant topic in recommender systems. Adomavicius and Kwon [53] proposed a number of different ranking approaches for improving recommendation diversity. Their approach was mostly based on item popularity and they focused on controlling accuracy-diversity trade-off. They did not consider the conflict constraints between similar items.

CA-WBM is also related to graph-based recommendations, such as [54], [55], [56]. Guan et al. [54] studied resource recommendation based on tagging data. The authors proposed a graph-based representation learning algorithm to investigate the relationship between users, tags and documents. Zhao et al. [55] tackled the problem of personalized tag recommendation. They modelled the complex relationships in tagging data as a heterogeneous graph. A novel ranking algorithmic framework was proposed to deal with multi-type interrelated objects. Guan et al. [56] analyzed members' web surfing data and utilized a probabilistic graphical model to investigate fine-grained knowledge acquiring and sharing in collaborative environments.

Additionally, CA-WBM can be applied to facilitate design of negotiation-based trade mechanisms in the context of bilateral markets [57]. A bilateral market consists of sellers and buyers who wish to exchange goods. The market's main objective is to compute the optimal allocation that maximizes gain from trade. Naturally, bilateral automated negotiation among rational agents (market participants) with one-shot protocol (where one participant proposes a deal and the other one may only accept or refuse it) can be conceptualized as WBM [58], [59] and can be solved efficiently. Typical research work in this field includes exploring the agent's utility spaces [60], [61] and designing negotiation strategies that achieve Pareto-efficient agreements [62], [63]. By incorporating conflicts, CA-WBM can be used to capture multiple compatibility issues among requests or offers [63], [64], and could lead to novel negotiation strategy design.

The online version of CA-WBM is related to online WBM. Karp et al. [65] first introduced the online bipartite matching problem, in which $b = 1$ and edges weights are all 1. They showed a greedy algorithm *GREEDY* with competitive ratio of $\frac{1}{2}$ and a randomized algorithm *RANKING* (under the assumption of adversarial order) with the optimal competitive ratio of $1 - \frac{1}{e} \approx 0.632$. Later, simpler proofs for the competitive ratio of *RANKING* were given in [66], [67], [68]. Kalyanasundaram and Pruhs [69] studied online unweighted b-matching and presented a deterministic algorithm *BALANCE* which achieves an optimal competitive ratio of $1 - \frac{1}{(1+\frac{1}{b})^b}$. For online WBM, Ting and Xiang [24] proposed a randomized algorithm and a deterministic algorithm. Both algorithms were proven to be near optimal. Online WBM has also been intensely studied in the emerging domain of Internet advertising [4]. Typical practical scenarios include online adwords [9], [10] and display advertising [11], [12]. Mehta [4] summarized various results in different arrival models (adversarial order, random order and known IID) for online matching problems, including bipartite matching, vertex-weighted bipartite matching (online adwords) and edge-weighted and capacitated bipartite matching (display advertising). Typically, these problems do not consider conflict between entities. Therefore, online CA-WBM is valuable to provide more flexible service to online advertising.

8 Conclusions

We initiated the study of a novel extension of classic weighted bipartite b-matching (WBM). The question we addressed is how to maximize the total weight when matching vertices under both degree and conflict constraints (CA-WBM). The CA-WBM problem is general and can find many applications in the domain of E-Commerce, such as Internet advertising and personalized recommendation.

We provided a formal definition of the central problem, CA-WBM, that directly models both the degree constraint on each vertex and conflict relationship between vertices on the same side. We showed that by considering the conflict constraints, the complexity of WBM increases significantly. We proved that CA-WBM is NP-hard. We modelled it using semidefinite programming (SDP) and integer linear programming (ILP). Then we proposed a SDP algorithm with rounding, LP relaxation with rounding, and a greedy algorithm to solve CA-WBM. We also proposed a randomized algorithm to solve online CA-WBM. We showed that they achieve close to optimal solutions via comprehensive experiments using synthetic datasets. We derived a theoretical bound on the approximation ratio of the greedy algorithm and showed that it is scalable on a large-scale real-world dataset.

Acknowledgments

We would like to thank Terapeak Inc. for providing the eBay data for this research. We also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for the funding support.

References

[1] D. P. Bertsekas, "A new algorithm for the assignment problem," *Mathematical Programming*, vol. 21, no. 1, pp. 152–171, 1981.

[2] R. Meir, Y. Chen, and M. Feldman, "Efficient parking allocation as online bipartite matching with posted prices," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 303–310.

[3] M. Fayyazi, D. Kaeli, and W. Meleis, "Parallel maximum weight bipartite matching algorithms for scheduling in input-queued switches," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE, 2004, p. 4.

[4] A. Mehta, "Online matching and ad allocation," *Foundations and Trends in Theoretical Computer Science*, vol. 8, no. 4, pp. 265–368, 2013.

[5] G. Adomavicius and Y. Kwon, "Maximizing aggregate recommendation diversity: A graph-theoretic approach," in *Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)*, 2011, p. 3.

[6] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, vol. 2009, pp. 4:2–4:2, Jan. 2009.

[7] Y. Koren and R. M. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, 2011, pp. 145–186.

[8] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 3:1–3:45, May 2014.

[9] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani, "Adwords and generalized online matching," *J. ACM*, vol. 54, no. 5, 2007.

[10] N. R. Devanur and T. P. Hayes, "The adwords problem: online keyword matching with budgeted bidders under random permutations," in *Proceedings 10th ACM Conference on Electronic Commerce (EC-2009), Stanford, California, USA, July*, 2009, pp. 71–78.

[11] J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein, "Online stochastic packing applied to display ad allocation," in *Proceedings of the 18th Annual European Conference on Algorithms: Part I*, ser. ESA'10, 2010, pp. 182–194.

[12] A. Bhalgat, N. Korula, H. Leontyev, M. Lin, and V. S. Mirrokni, "Partner tiering in display advertising," in *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, 2014, pp. 133–142.

[13] O. Amini, D. Peleg, S. Pérennes, I. Sau, and S. Saurabh, "On the approximability of some degree-constrained subgraph problems," *Discrete Applied Mathematics*, vol. 160, no. 12, pp. 1661–1679, 2012.

[14] A. Schrijver, *Theory of Linear and Integer Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1986.

[15] A. S. Asratian, T. M. J. Denley, and R. Häggkvist, *Bipartite Graphs and Their Applications*. New York, NY, USA: Cambridge University Press, 1998.

[16] M. Yannakakis, "On a class of totally unimodular matrices," in *Foundations of Computer Science, 1980., 21st Annual Symposium on*, Oct 1980, pp. 10–16.

[17] K. R. Rebman, "Total unimodularity and the transportation problem: a generalization," *Linear Algebra and its Applications*, vol. 8, no. 1, pp. 11 – 24, 1974.

[18] E. M. Arkin and E. B. Silverberg, "Scheduling jobs with fixed start and end times," *Discrete Appl. Math.*, vol. 18, no. 1, pp. 1–8, Nov. 1987.

[19] R. Bhatia, J. Chuzhoy, A. Freund, and J. S. Naor, "Algorithmic aspects of bandwidth trading," *ACM Trans. Algorithms*, vol. 3, no. 1, pp. 10:1–10:19, Feb. 2007.

[20] A. W. Kolen, J. K. Lenstra, C. H. Papadimitriou, and F. C. Spieksma, "Interval scheduling: A survey," *Naval Research Logistics (NRL)*, vol. 54, no. 5, pp. 530–543, 2007.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C: The art of scientific computing (second edition)*. Cambridge University Press, 1992.

[22] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[23] J. Mestre, "Greedy in approximation algorithms," in *ESA*, 2006, pp. 528–539.

[24] H. Ting and X. Xiang, "Near optimal algorithms for online maximum weighted b-matching," in *Frontiers in Algorithmics - 8th International Workshop, FAW 2014, Zhangjiajie, China, June 28-30, 2014. Proceedings*, 2014, pp. 240–251.

[25] D. D. Sleator and R. E. Tarjan, "Amortized efficiency of list update and paging rules," *Commun. ACM*, vol. 28, no. 2, pp. 202–208, 1985.

[26] K. C. Toh, M. Todd, and R. H. Tütüncü, "Sdpt3 – a matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 1999.

[27] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Math. Program.*, vol. 95, no. 2, pp. 189–217, 2003.

[28] M. Kocvara and M. Stingl, "On the solution of large-scale sdp problems by the modified barrier method using iterative solvers," *Math. Program.*, vol. 109, no. 2-3, pp. 413–444, 2007.

[29] B. EDELMAN, M. OSTROVSKY, and M. SCHWARZ, "Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords," *The American economic review*, vol. 97, no. 1, pp. 242–259, 2007.

[30] W. Lu, S. Chen, K. Li, and L. V. S. Lakshmanan, "Show me the money: Dynamic recommendations for revenue maximization," *PVLDB*, vol. 7, no. 14, pp. 1785–1796, 2014.

[31] B. C. Huang and T. Jebara, "Loopy belief propagation for bipartite maximum weight b-matching," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 195–202.

[32] ——, "Fast b-matching via sufficient selection belief propagation," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 361–369.

[33] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, "Learning structured prediction models: A large margin approach," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 896–903.

[34] T. Jebara and V. Shchogolev, "B-matching for spectral clustering," in *Machine Learning: ECML 2006*. Springer, 2006, pp. 679–686.

[35] B. Shaw and T. Jebara, "Minimum volume embedding," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 460–467.

[36] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 441–448.

[37] B. Shaw and T. Jebara, "Structure preserving embedding," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 937–944.

[38] E. Dinic, "An algorithm for the solution of the max-flow problem with the polynomial estimation," *Doklady Akademii Nauk SSSR*, vol. 194, no. 4, pp. 1277–1280, 1970.

[39] Y. Dinitz, "Dinitzalgorithm: The original version and evens version," in *Theoretical Computer Science*. Springer, 2006, pp. 218–240.

[40] J. E. Hopcroft and R. M. Karp, "An nˆ5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on computing*, vol. 2, no. 4, pp. 225–231, 1973.

[41] J. Mestre, "Greedy in approximation algorithms," in *Algorithms–ESA 2006*. Springer, 2006, pp. 528–539.

[42] J.-H. Hoepman, "Simple distributed weighted matchings," *arXiv preprint cs/0410047*, 2004.

[43] G. De Francisci Morales, A. Gionis, and M. Sozio, "Social content matching in mapreduce," *Proceedings of the VLDB Endowment*, vol. 4, no. 7, pp. 460–469, 2011.

[44] F. M. Manshadi, B. Awerbuch, R. Gemula, R. Khandekar, J. Mestre, and M. Sozio, "A distributed algorithm for large-scale generalized matching," *PVLDB*, vol. 6, no. 9, pp. 613–624, 2013.

[45] A. Felfernig and R. D. Burke, "Constraint-based recommender systems: technologies and research issues," in *ICEC*, 2008, p. 3.

[46] M. Zanker and M. Jessenitschnig, "Case-studies on exploiting explicit customer requirements in recommender systems," *User Model. User-Adapt. Interact.*, vol. 19, no. 1-2, pp. 133–166, 2009.

[47] M. Zanker, M. Jessenitschnig, and W. Schmid, "Preference reasoning with soft constraints in constraint-based recommender systems," *Constraints*, vol. 15, no. 4, pp. 574–595, 2010.

[48] M. Karimzadehgan and C. Zhai, "Constrained multi-aspect expertise matching for committee review assignment," in *CIKM*, 2009, pp. 1697–1700.

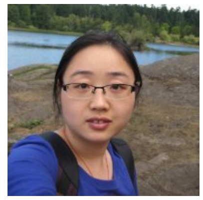
[49] ——, "Integer linear programming for constrained multi-aspect committee review assignment," *Information processing & management*, vol. 48, no. 4, pp. 725–740, 2012.

[50] M. Xie, L. V. S. Lakshmanan, and P. T. Wood, "Breaking out of the box of recommendations: from items to packages," in *RecSys*, 2010, pp. 151–158.

[51] A. G. Parameswaran, P. Venetis, and H. Garcia-Molina, "Recommendation systems with complex constraints: A course recom-

mendation perspective," *ACM Trans. Inf. Syst.*, vol. 29, no. 4, p. 20, 2011.
[52] M. Karimzadehgan, C. Zhai, and G. Belford, "Multi-aspect expertise matching for review assignment," in *CIKM*. ACM, 2008, pp. 1113–1122.
[53] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 896–911, 2012.
[54] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He, "Document recommendation in social tagging services," in *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, 2010, pp. 391–400.
[55] W. Zhao, Z. Guan, and Z. Liu, "Ranking on heterogeneous manifolds for tag recommendation in social tagging services," *Neurocomputing*, vol. 148, pp. 521–534, 2015.
[56] Z. Guan, S. Yang, H. Sun, M. Srivatsa, and X. Yan, "Fine-grained knowledge sharing in collaborative environments," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2163–2174, 2015.
[57] S. Fatima, S. Kraus, and M. Wooldridge, *Principles of Automated Negotiation*. Cambridge University Press, 2014.
[58] J. S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. Cambridge, MA, USA: MIT Press, 1994.
[59] H. Raiffa, *The art and science of negotiation*. Harvard University Press, 1982.
[60] Y. Bartal, R. Gonen, and P. La Mura, "Negotiation-range mechanisms: Exploring the limits of truthful efficient markets," in *Proceedings of the 5th ACM Conference on Electronic Commerce*, ser. EC '04. New York, NY, USA: ACM, 2004, pp. 1–8.
[61] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, T. Ito, M. Klein, and K. Fujita, "Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, 2009, pp. 214–219.
[62] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra, "Automated negotiation: prospects, methods and challenges," *Group Decision and Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.
[63] A. Ragone, T. Di Noia, E. Di Sciascio, and F. M. Donini, "Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 3, pp. 249–270, 2008.
[64] ——, "Dl-based alternating-offers protocol for automated multi-issue bilateral negotiation," in *20th International Workshop on Description Logics DL07*. Citeseer, 2007, pp. 443–450.
[65] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," in *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990*, 1990, pp. 352–358.
[66] G. Goel and A. Mehta, "Online budgeted matching in random input models with applications to adwords," in *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2008, pp. 982–991.
[67] B. Birnbaum and C. Mathieu, "On-line bipartite matching made simple," *ACM SIGACT News*, vol. 39, no. 1, pp. 80–87, 2008.
[68] N. R. Devanur, K. Jain, and R. D. Kleinberg, "Randomized primal-dual analysis of ranking for online bipartite matching," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2013, pp. 101–107.
[69] B. Kalyanasundaram and K. R. Pruhs, "An optimal deterministic algorithm for online b-matching," *Theoretical Computer Science*, vol. 233, no. 1, pp. 319–325, 2000.

Cheng Chen received the B.Sc. degree in Computer Science from Beijing University of Posts and Telecommunications, China in 2010. He received the M.Sc. degree in Computer Science from the University of Victoria, Canada in 2012, and he is currently a Ph.D. Candidate there. His research interests are online social networks, recommender systems, and distributed algorithms for graph mining.

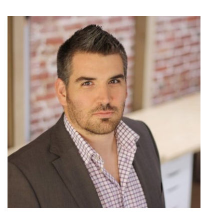


Lan Zheng received a bachelor degree in Information Management and Information System and a master degree in Management Science from Beijing University of Posts and Telecommunications. She is currently a master student in Computer Science at University of Victoria. Her research interest is recommender system and online social network.

Venkatesh Srinivasan received a B.Eng degree from Birla Institute of Technology and Science, India, in 1994, and a Ph.D. degree from Tata Institute of Fundamental Research, India, in 2000. His research focuses on theory of algorithmics and its applications to databases, data mining and security. His awards and honours include prestigious postdoctoral fellowships at the Institute for Advanced study at Princeton, the Center for Discrete Mathematics and Theoretical Computer Science at Rutgers University and the Max-Planck Institute for Informatics in Germany. He was also the Canada Research Chair on the Complexity of Computing, a program initiated by the Canadian Government to attract and retain top international researchers.

Alex Thomo received his B.Sc. degree from University of Piraeus, Greece in 1996 and his M.Sc. and Ph.D. degrees from Concordia University, Canada in 2001 and 2003 respectively. He joined the Department of Computer Science at the University of Victoria, Canada in 2004 and is currently an Associate Professor there. His research interests include algorithms for big graphs, social networks, privacy, and distributed computing.

Kui Wu received the B.Sc. and the M.Sc. degrees in Computer Science from Wuhan University, China in 1990 and 1993, respectively, and the Ph.D. degree in Computing Science from the University of Alberta, Canada, in 2002. He joined the Department of Computer Science at the University of Victoria, Canada in 2002 and is currently a Professor there. His research interests cover online social networks, Internet of Things, and parallel and distributed algorithms. He is a member of the Association for Computing Machinery (ACM) and a senior member of the Institute of Electrical and Electronics Engineers (IEEE).



Anthony Sukow is a Co-Founder of Terapeak. He has worked developing Business Intelligence tools for e-commerce merchants for the past 10 years. Under his lead, Terapeak, as the Master Licensing Partner of eBay e-commerce Data, has created systems to manage over 18 billion transactions, representing over 425 billion worth of online spending from 100 million consumers around the world. As a Co-Founder, he provides vision and passion for understanding how to build high value data warehouses based on big e-commerce data.