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

(Extended Abstract*)

Cheng Chen†, Lan Zheng†, Venkatesh Srinivasan†, Alex Thomo†, Kui Wu†
Department of Computer Science, University of Victoria
Victoria, BC, Canada
Email: †{cchenv, lanzheng, srinivas, thomo, wkui}@uvic.ca

Anthony Sukow‡
Terapeak Inc.
Victoria, BC, Canada
Email: ‡asukow@terapeak.ca

I. 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 $W : E \to \mathbb{R}^+$, where U, V and E represent left vertices, right vertices and edges, respectively, the weighted bipartite b-matching problem (WBM) is to find a subgraph $H \subseteq G$ with maximum total weight $w(H) = \sum_{e\in H} w(e)$, such that every vertex i in H is incident to at most $b(i)$ edges.

An implicit assumption of WBM is that any two nodes on the same side do not conflict with each other, even if they share similar features in practice. 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. However, considering conflicts would inevitably lead to new challenges for WBM when generating the matching result. This conflict challenge has not been fully studied and thus is the focus of this work.

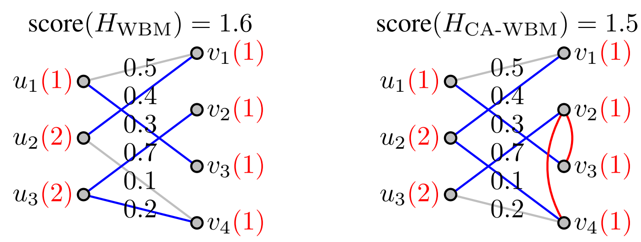
In this article, we introduce a new generalization of WBM, Conflict-Aware Weighted Bipartite b-Matching (CA-WBM), that can address the conflict challenges mentioned above.

II. Problem Definition And Its Hardness

In order to capture the conflict constraint, we consider a natural extension of WBM. Formally, *we say that two vertices are in conflict with each other if matching them to the same vertex is not desirable.*

CA-WBM imposes a set of conflict pairs C, requiring that each $u \in U$ is adjacent to at most $\tau(u)$ of those pairs:

Definition 1. Conflict-Aware Weighted Bipartite B-Matching (CA-WBM) Given G, vertex-labelling functions $B : U \bigcup V \to \mathbb{N}$, $\tau : U \to \mathbb{N}$, and a set of unordered pairs $C \subseteq V \times V$, find the subgraph $H = ((U, V), E', W)$ maximising $\sum_{e\in E'} W(e)$ with every vertex $u \in U \bigcup V$ adjacent to at most $B(u)$ edges and every vertex $u \in U$ adjacent to at most $\tau(u)$ pairs of vertices $v, v' \in V$ that appear as an unordered pair $(v, v') \in C$.



(a) H_{WBM} (b) $H_{\text{CA-WBM}}$

Fig. 1: The CA-WBM problem contrasted with WBM. Two conflicts, $(v_2, v_3), (v_2, v_4)$, are introduced, e.g., because the products are too similar. If u_3 has a conflict threshold $\tau(u_3) = 0$, then it cannot match both v_2 and v_4, leading to a lower score, but potentially more diverse, solution.

In terms of hardness result, we prove that:

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

III. Algorithms for Solving CA-WBM

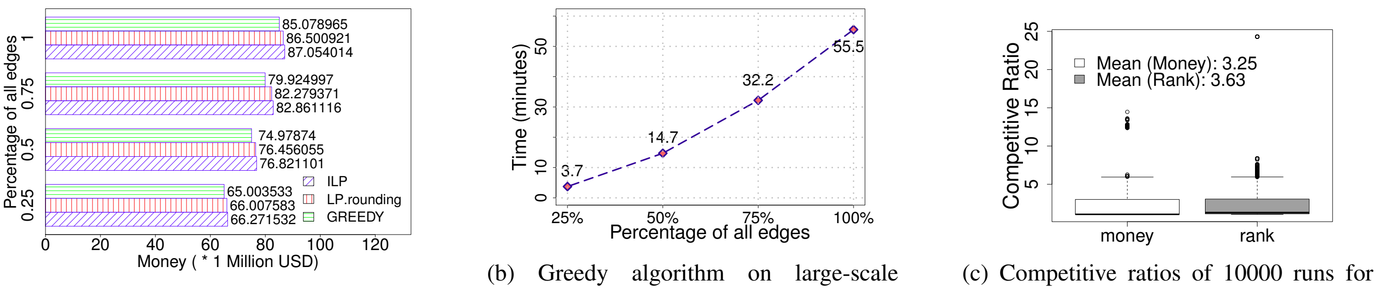
We present two formulations of CA-WBM, capturing the conflict constraint and propose corresponding algorithms for solving CA-WBM. We also study an online version of CA-WBM. Let $m = |U|$ and $n = |V|$. We represent the degree constraint for all vertices, as specified by the vertex-labelling function B, 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 i is matched to j and $x_{ij} = 0$ otherwise.

A. Integer Linear Program (ILP)-based Algorithm

CA-WBM without conflict constraints can be described as

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

* A longer version of this paper was published in *IEEE TKDE* [1].



(a) LP-based algorithms.

(b) Greedy algorithm on large-scale datasets showing its scalability.

(c) Competitive ratios of 10000 runs for *Randomized CA-WBM*.

Fig. 2: Experiment plots for the ILP, LPR, and GREEDY algorithms on CA-WBM and online CA-WBM

where matrix $\mathbb{A}$ is an $(m+n)\times mn$ matrix indicating adjacent edges for vertices in U and V.

To formulate conflict constraints linearly, we introduce a new 0-1 variable $z_{i,(j,k)}$. For each vertex $i \in U$, $z_{i,(k,l)}$ equals 1 if and only if there is a conflict edge between two vertices $k, l \in V$, and both edges e_{ki} and e_{li} are matched in the graph. Therefore, conflict constraints can be described as follows:

$$1 - x_{ki} - x_{li} + z_{i,(k,l)} \geq 0,\ \forall\ i \in U,\ \forall (k,l) \in C_i \tag{2}$$

$$x_{ki} + x_{li} - 2z_{i,(k,l)} \geq 0,\ \forall\ i \in U,\ \forall (k,l) \in C_i \tag{3}$$

$$\sum_{(k,l)\in C_i} z_{i,(k,l)} \leq \tau,\ \forall\ i \in U \tag{4}$$

In constraints (2), (3) and (4), C_i is defined as: $C_i = \{(k,l) \in C | (k,i) \in E \wedge (l,i) \in E\}$. That is, C_i represents the set of conflicts within the set of vertices in V linked to vertex $i \in U$.

This problem can be solved by ILP solvers, such as Gurobi[1]. Since obtaining an integer solution in CA-WBM is NP-hard, in order to improve scalability, we use a rounding procedure after solving the linear program (LP) 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* (LPR).

B. A Greedy Algorithm

To further improve scalability, an intuitive greedy algorithm can be used to obtain approximate solution. This algorithm (GREEDY) tries to match candidate edges with the maximum weight if doing so does not violate any degree constraint or conflict constraint. We showed performance guarantees of GREEDY by connecting CA-WBM to *k-extendible system* [2], specifically for $k = 2$.

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

C. An Online Algorithm

We also study an online version of CA-WBM (online CA-WBM), where vertices of a particular side (e.g., buyers) arrive in an online fashion and the corresponding edges, as well as their weights, and conflict edges are revealed when each vertex arrives [3], [4]. Upon arrival of a vertex v, the system is assumed to immediately generate matchings to v and cannot change the matchings at a later time.

For approximately solving online CA-WBM, we propose an online randomized algorithm *Randomized CA-WBM*, inspired by the algorithm in [5]. Denoting the maximum weight of any edge in G by $w_{\max}$, the performance is given as:

Theorem 3. ***Randomized CA-WBM** achieves a competitive ratio of $(\alpha+1)e\lceil\ln(1+w_{\max})\rceil$, where $\alpha = \max(d_1-1, d_2)$, $d_1 = \max_{s\in U} B(s)$ and $d_2 = \max_{b\in V} |\{(b,b')|(b,b') \in C\}|$.*

We showed that *Randomized CA-WBM* is near optimal by giving the following lower bound of performance on randomized algorithms.

Theorem 4. *For online CA-WBM, no randomized algorithm can achieve a competitive ratio better than $\frac{\lceil\log_2(w_{\max}+1)\rceil+1}{2}$.*

IV. Results and Evaluation

We implemented corresponding exact and approximate algorithms, and evaluated their performance of optimality and scalability on both synthetic and real-world datasets. The results (plotted in Fig.2) show that the proposed approximate algorithms can consistently achieve solutions at least 85% of the optimal. The greedy algorithm shows linearly increasing running time and is especially scalable to large-scale datasets. In addition, *Randomized CA-WBM* achieves approximately 3.5 competitive ratio [6] on average, which indicates promising performance in practice.

References

[1] C. Chen, L. Zheng, V. Srinivasan, A. Thomo, K. Wu, and A. Sukow, "Conflict-aware weighted bipartite b-matching and its application to e-commerce," *IEEE Trans. on Knowl. and Data Eng.*, vol. 28, no. 6, pp. 1475–1488, Jun. 2016.

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

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

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

[5] H. Ting and X. Xiang, "Near optimal algorithms for online maximum weighted b-matching," in *FAW*, 2014, pp. 240–251.

[6] 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.

[1] http://www.gurobi.com/, accessed December 2016.