Appendix to "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

1 Introduction

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.

2 Flaws in Ahmadi et al.

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.

2.1 The MIQP model

Ahmadi et al.'s MIQP model is as follows:

$$\min_{c} \quad \lambda_0 \cdot \sum_{f\in F}\sum_{u\in U}\sum_{g_f\in G_f} w_{u,f,g_f}\cdot c_{u,f,g_f} + \tag{1a}$$

$$\sum_{f\in F}\lambda_f \cdot \sum_{u\in U}\sum_{g_f\in G_f}\left(c_{u,f,g_f}\right)^2 \tag{1b}$$

$$\text{s.t.} \quad \sum_{f\in F}\sum_{g_f\in G_f} c_{u,f,g_f} = d_u, \forall u\in U, \tag{1c}$$

$$\sum_{u\in U} c_{u,f,g_f} = |F_f^g|, \forall f\in F, \forall g_f\in G_f, \tag{1d}$$

$$c_{u,f,g_f} \geq 0, \forall u\in U, \forall f\in F, \forall g_f\in G_f.. \tag{1e}$$

Here c_{u,f,g_f} are integer decision variables representing the number of workers assigned to team u having value $g_f\in G_f$ for feature class $f\in F$. The constraint

(1d) is to ensure that the number of workers with feature value g_f assigned to all the teams equals $|F_f^g|$.

There is a likely typographical error in (1c). The summation equals $|F| \times d_u$ instead of d_u if the double sum is being used. The correct form of the constraint is:

$$\sum_{g_f \in G_f} c_{u,f,g_f} = d_u, \forall u \in U, \forall f \in F. \tag{1c'}$$

However, even with this correction, the model is still incorrect w.r.t. the problem definition. The key insight is that the decision variables do not represent assignments. Thus, there is no bijection between the set of assignments and the set of solutions to the MIQP model. We denote the MIQP model after correction (1a, 1b, 1c', 1d, 1e) by Q_M. We use a worker-team assignment example in Table 1 to show the problem.

	Feature		
Worker	f_1	f_2	f_3
r_1	1	0	1
r_2	1	1	0
r_3	0	0	0
r_4	0	1	1

Table 1. Example for MIQP Model.

The instance has two teams, four workers, and three feature classes. Each feature class has two possible values: 0 and 1. Let the assignment cost of each worker to each team be the same. We want to assign two workers to each of the two teams and maximize diversity, i.e., make two workers have different values for as many feature classes as possible. There exists no solution that is diverse in terms of the three classes, because each pair of workers matches on one feature class.

However, if we use the MIQP model Q_M, we are able to achieve diversity on all the three feature classes. For example, $\{c_{u,f,g_f} = 1, \ \forall u \in \{1,2\}, \forall f \in \{1,2,3\}, \forall g_f \in \{0,1\}\}$ is an optimal solution to Q_M. We can see that constraints (1c') are satisfied:

$$c_{1,1,0} + c_{1,1,1} = 2, \quad c_{1,2,0} + c_{1,2,1} = 2, c_{1,3,0} + c_{1,3,1} = 2,$$
$$c_{2,1,0} + c_{2,1,1} = 2, \quad c_{2,2,0} + c_{2,2,1} = 2, c_{2,3,0} + c_{2,3,1} = 2.$$

Also, constraints (1d) are satisfied:

$$c_{1,1,0} + c_{2,1,0} = |F_1^0| = 2 = c_{1,1,1} + c_{2,1,1} = |F_1^1|,$$
$$c_{1,2,0} + c_{2,2,0} = |F_2^0| = 2 = c_{1,2,1} + c_{2,2,1} = |F_2^1|,$$
$$c_{1,3,0} + c_{2,3,0} = |F_3^0| = 2 = c_{1,3,1} + c_{2,3,1} = |F_3^1|.$$

Thus, Q_M is incorrect since it decouples the combination of feature values from an assignment.

2.2 The local exchange algorithm

Ahmadi et al. also proposed a local exchange (LE) algorithm for MDWBM. The algorithm starts from a feasible solution and changes assignment to decrease the objective while satisfying all the constraints. A series of moves that leads to a decrease in the objective is called a negative cycle. The algorithm stops when it cannot find negative cycle. Obviously, the algorithm stops at a local optimum. The authors claimed and proved that the local optimum is also the global optimum [1]. However, we show a counter-example in Table 2 that disproves the claim.

	Feature value			Assignment cost	
Worker	g_1	g_2	g_3	t_1	t_2
r_1	0	6	2	8	9
r_2	0	0	4	10	9
r_3	0	0	1	11	9
r_4	1	6	3	7	12
r_5	1	5	3	7	15
r_6	0	7	0	9	14

Table 2. Example for local exchange algorithm.

The example is also in the worker-team assignment context. There are two teams and six workers with three feature classes. Each team needs exactly three workers. The three feature values of each worker and the costs of assigning workers to the two teams are shown in the table. With $\lambda_0 = 1, \lambda_1 = 2, \lambda_2 = 5, \lambda_3 = 4$, a locally optimal solution s^L assigns workers r_1, r_2, r_5 to t_1 and workers r_3, r_4, r_6 to t_2. The objective value of this solution is 134. By contrast, the globally optimal s^* solution assigns workers r_2, r_4, r_6 to t_1 and workers r_1, r_3, r_5 to t_2, with the objective being 133. The local exchange algorithm cannot escape the locally optimal solution. To demonstrate this fact, we present the marginal costs of local moves from s^L in Table 3.

move	cost	move	cost	move	cost
1i3o	11	2i3o	-1	5i3o	20
1i4o	5	2i4o	13	5i4o	8
1i6o	11	2i6o	9	5i6o	20
3i1o	12	4i1o	7	6i1o	-5
3i2o	2	4i2o	17	6i2o	-5
3i5o	16	4i5o	5	6i5o	-1

Table 3. Marginal costs of local moves from s^L.

The '1i3o' in the table is a local step moves r_1 into t_2 and r_3 out of t_2. The cost of '1i3o' is composed of two parts. The first is the objective (including the weighted costs and similarity measures) difference between $[t_1 : (r_2, r_5), t_2 : (r_1, r_3, r_4, r_6)]$ and s^L. The second part adds $-2\lambda_f$ to the cost if r_1 and r_3 have the same feature value for feature class f [1]. The LE algorithm considers all the cycles in the table. For example, 1i3o $\rightarrow$ 3i2o $\rightarrow$ 2i6o $\rightarrow$ 6i1o forms a cycle. The total gain of the cycle is $11 + 2 + 9 - 5 = 17 > 0$.

Any cycle which moves a node from a team (e.g., 3o in 2i3o) must also contain a move that inserts the same node (e.g., 3i). There are moves that can convert s^L to s^*, such as the cycle 1i6o $\rightarrow$ 6i5o $\rightarrow$ 5i4o $\rightarrow$ 4i1o. Observe that any negative cycle must contain at least one move with a negative cost. For each negative 'o' in the table, all corresponding 'i' moves result in a non-negative cost. Thus, any cycle extracted from Table 3 has a non-negative gain. The LE algorithm hence gets stuck in s^L and does not guarantee to find the globally optimal solution.

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)