Imperfect b-matching

Kui Tang

May 2, 2013

This code implements imperfect maximum-weight b-matching where b is allowed to vary by node. We describe the reduction to perfect maximum-weight 1-matching, and the API and implementation of this code.

Let $G = (V, E)$ be a graph and let $W \in \mathbb{N}^{|V|\times|V|}$ and $P \in \mathbb{B}^{|V|\times|V|}$ denote symmetric weight and adjacency matrices. Let δ_u denote the degree of node u. Let $\mathbf{b} \in \mathbb{N}^{|V|}$ be a vector such that entry b_i upper-bounds the number of neighbors node i may match. The code then solves

$$\begin{aligned} \max_P \sum_{ij} P_{ij} W_{ij} \\ \text{s.t.} \sum_j P_{ij} \le b_i \end{aligned} \tag{1}$$

by combining reductions due to Bondy and Murty [1] (b-matching to 1-matching) and Schäfer [3] (imperfect to perfect matching) and then calling out to the BlossomV solver [2].

The same code can also solve a minimum-weight matching with a lower bound constraint. To see, let $\overline{P}$ denote the relative complement of P in G. That is, $\overline{P}_{ij} = 1$ if and only if $P_{ij} = 0$ and $(i, j) \in E$. Then problem (1) is equivalent to

$$\begin{aligned} \min_{\overline{P}} \sum_{ij} \overline{P}_{ij} W_{ij} \\ \text{s.t.} \sum_j \overline{P}_{ij} \ge \delta_i - b_i \end{aligned} \tag{2}$$

This is because since vertex i has δ_i neighbors, if we picked at most b_i of its neighbors that maximized the objective, then there main at least $\delta_i - b_i$ neighbors, which, if matched, would minimize the objective.

In practice, the weights W are nonnegative real, but BlossomV requires integral weights. We thus scale by normalizing weights to $[0, 1]$ and then scaling by a large number (10^7 works) and rounding, keeping 7 significant figures. As a final detail, BlossomV solves a *minimum*-weight matching, but handles negative weights, so we simply negate W before calling out.

To simplify our proofs, we assume elements of W are unique, which can always be arranged in practice by solving a perturbed problem.

1 Reducing to perfect 1-matching

Bondy and Murty [1, pp. 431] describe a reduction from perfect b-matching to perfect 1-matching (adding polynomially more nodes) and [3, Sec. 1.5.2] describes a reduction from imperfect 1-matching to perfect 1-matching (doubling the number of nodes). With some care, we can combine the two ideas and obtain a polynomial-sized reduction from imperfect b-matching to perfect 1-matching.

1.1 Perfect b-matching to perfect 1-matching

To reduce a perfect b-matching a perfect 1-matching, convert each of original vertices to a bipartite graph, consisting of a *core* vertices and a *peripheral* vertices. The edges in the original graph map to edges between peripheral vertices, and extra edges between the core and periphery enforce the b constraint.

For each vertex $u \in V$, form $\delta_u - b_u$ vertices in the core set X_u and $\delta(u)$ vertices in the peripheral set Y_u. Form a zero-weight edge from each vertex in X_u to each vertex in Y_u, called *interior edges.* Now X_u and Y_u form a bipartite subgraph. Form $\tilde{V}$ as the union of the new vertices.

For each edge $(u, v) \in E$, we can find a unique pair $y_u \in Y_u$ and $y_v \in Y_v$. This is simply because $|Y_u| = \delta(u)$, so for every edge incident to u, one vertex in Y_u can be devoted to representing that edge. Similarly for v. Form $\tilde{E}$ containing each *peripheral edge* (y_u, y_v), and copy the weights $\tilde{W}_{y_u y_v} = W_{uv}$.

We can see how the interior edges enforce the b constraint: vertex u has δ_u neighbors, of which b_u must be matched. This means that $\delta_u - b_u$ neighbors are unmatched, which is the exact number of core vertices. Each core vertex will match a peripheral vertex, leaving b_u peripheral vertices left to match with peripheral vertices of other subgraphs.

Now given a perfect 1-matching $\tilde{P}$ of $\tilde{G}$, we can *collapse* it to get a perfect b-matching P of G by remembering the origins of each peripheral edge. Begin with P empty. For each $(y_u, y_v) \in \tilde{P}$ where $y_u \in Y_u$ and $y_v \in Y_v$, add (u, v) to P. Since $\tilde{P}$ is a 1-matching, each extracted y_u and y_v is unique. Because vertex u has b_u peripheral vertices matched to other peripheral vertices in $\tilde{P}$, u will be matched exactly b_u times in P.

1.2 Imperfect 1-matching to perfect 1-matching

To reduce an imperfect 1-matching of G to a perfect 1-matching, we first create an isomorphic copy called $G + N$ and add zero-weight edges between each vertex in G to its isomorphic image in $G + N$.

More precisely, assume $V = \{1, \ldots, N\}$. Let $V{+}N := \{v{+}N | v \in V\}$, $E{+}N := \{(u + N, v + N)|(u, v) \in E\}$, $F := \{(u, f(u))\,|u \in V\}$. Define an isomorphism $f : V \to V + N$ as $f(u) := u + N$ and define $G' := (V', E')$ where

$$\begin{aligned} V' &:= V \cup V + N \\ E' &:= E \cup E + N \cup F \end{aligned}$$

The isomorphic copy gives an escape route: If in G we can get a higher total weight by not matching some vertex $v \in G$ at all, then in G' we match it instead to $f(v)$—still a perfect matching.

Proposition 1 ([3, Lemma 1.5.1])**.** *Let P' be a maximum-weight perfect 1-matching of G' and let P be the upper-left $N \times N$ block of $\tilde{P}$. Then P is a maximum-weight imperfect 1-matching of G. Conversely, if P is a maximum-weight imperfect 1-matching of G, then there exists a maximum-weight perfect 1-matching of G' such that its upper-left $N \times N$ block is P.*

1.3 Imperfect b-matching to perfect 1-matching

To reduce an imperfect b-matching to a perfect 1-matching, we first take the Bondy-Murty reduction $\tilde{G} = (\tilde{V}, \tilde{E})$ and again create an isomorphic copy $\tilde{G} + M$, where $M = |\tilde{V}|$. The difference here is that we add edges across the isomorphic copies only for peripheral vertices. That is, with $H = \{(y_u, f(y_u))|y_u \in Y_u \text{ for some } u \in V\}$, define $\tilde{G}^{+} = (\tilde{V}^{+}, \tilde{E}^{+})$ where

$$\begin{aligned} \tilde{V}^{+} &:= \tilde{V} \cup \tilde{V} + M \\ \tilde{E}^{+} &:= \tilde{E} \cup \tilde{E} + M \cup H \end{aligned}$$

We can interpret the reduction as follows: we ultimately care about peripheral vertices because the edges from the original graph are the edges between peripheral vertices. Each peripheral vertex is incident to exactly one other peripheral vertex in $\tilde{G}$. If there were no core vertices, then each peripheral vertex would be matched, and the matching corresponds to the entire original graph G. It is in fact a maximum-weight perfect b-matching where $b_u = \delta_u$ for each $u \in V$.

In general, $b_u \leq \delta_u$, so we form $\delta_u - b_u$ core vertices. On the subgraph for vertex u, each core vertex is connected to each peripheral vertex. Inner edge weights are zero, so the total weight is unchanged. For a matching on $\tilde{G}$ to be perfect, each core vertex matches a peripheral vertex on the same subgraph. This leaves b_u peripheral vertices to match with peripheral vertices on other subgraphs, corresponding to matching u with b_u of its neighbors in the original G.

Now suppose we could attain a higher total weight by matching vertex u with $c_u < b_u$ neighbors. Then we would have $\delta_u - (\delta_u - b_u) - c_u = b_u - c_u > 0$ peripheral vertex which match neither a vertex nor a peripheral vertex in $\tilde{G}$. These leftover vertices then match their isomorphic copies, contributing zero weight. Note if the original weights W are unique, then both copies have isomorphic matchings. We formalize this story in the following proof:

Proposition 2. *Let $\tilde{P}^+$ be a maximum-weight perfect 1-matching of $\tilde{G}^+$ whose upper-left $M \times M$ block is $\tilde{P}$. Then $\tilde{P}$ can be collapsed to a maximum-weight imperfect b-matching P of G.*

Conversely, if there exists a maximum-weight imperfect b-matching P of G, then there exists a maximum-weight perfect 1-matching of $\tilde{G}^+$ whose upper-left $M \times M$ block can be collapsed to P.

Proof. Because $\tilde{P}^+$ is an adjacency matrix and vertex i is isomorphic to vertex $i + M$, we can decompose in block form

$$\tilde{P}^+ = \begin{bmatrix} \tilde{P} & Q \\ Q^\top & \tilde{P} + M \end{bmatrix}$$

Suppose $\tilde{P}^+$ is a maximum-weight perfect 1-matching of $\tilde{G}^+$, but $\tilde{P}$ were not a maximum-weight imperfect 1-matching of $\tilde{G}$. Let $\tilde{P}_M$ be an imperfect 1-matching of $\tilde{G}$ and form $\tilde{P}_M^+$ whose upper-left block is $\tilde{P}_M$, lower-right block is $\tilde{P}_M + M$, and off-diagonal blocks set to satisfy the equality constraint $\sum_j \tilde{P}_{ij}^+ = 1$. Then $\tilde{P}_M^+$ is a perfect 1-matching. Since the off-diagonal blocks have zero weight, the total weight is $\sum_{ij} \tilde{P}_{M,ij}^+ \tilde{W}_{ij}^+ = 2 \sum_{i'j'} \tilde{P}_{M,i'j'} \tilde{W}_{i'j'} > 2 \sum_{i'j'} \tilde{P}_{i'j'} \tilde{W}_{i'j'} = \sum_{ij} \tilde{P}_{ij}^+ \tilde{W}_{ij}^+$, contradicting that $\tilde{P}^+$ is maximum weight.

For any $u \in G$, consider $x_u \in X_u$. Since x_u is a core vertex, the edge $(x_u, x_u + M)$ does not exist in $\tilde{G}^+$, so $\tilde{P}$ must match x_u to some vertex in Y_u in $\tilde{V}$. Since $\tilde{P}$ is an imperfect 1-matching, at most $\delta_u - (\delta_u - b_u) = b_u$ vertices remain in Y_u that are not matched to vertices in X_u. Let y_u be any one of those. By construction, y_u is incident to at most 2 edges in $\tilde{E}^+$: (y_u, y_w) for one particular $w \in V$ such that $(u, w) \in E$, and $(y_u, y_u + M)$. Since $\tilde{P}^+$ is a perfect matching, exactly one of the two edges is chosen. Collapsing as discussed in Section 1.1 reuslts in u's matching at most b_u neighbors in the original graph G.

Now suppose P is a maximum-weight imperfect b-matching of G. Then P corresponds to peripheral edges in $\tilde{G}$; put these edges in $\tilde{P}$. Then $\tilde{P}$ is a maximum-weight imperfect 1-matching of $\tilde{G}$, because the non zero-weight edges of $\tilde{G}$ are exactly those in G. Consider $u \in V$. Since u matches at most b_u neighbors in G, at least $\delta_u - b_u$ elements of Y_u will not yet be matched in $\tilde{P}$, so augment to match them to the core vertices X_u. If there are not enough core vertices, then augment $\tilde{P}$ to match the remaining y_u's to their isomorphic images $y_u + N$. Copy $\tilde{P}$ onto the isomorphic copy $\tilde{G} + N$ and call it $\tilde{P}^+$. note that the edges between isomorphic copies are symmetric. These extra edges have zero weight, and $\tilde{P} + N$ is a maximum-weight matching of $\tilde{G} + N$ by the isomorphism (since all edges across the isomorphic copies have zero weight). Now each vertex in $\tilde{G}^+$ is matched to exactly one neighbor in $\tilde{P}^+$, so we have a maximum-weight 1-matching. □

2 Code interface

References

[1] Adrian Bondy and U.S.R. Bondy. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 3 edition, 2008. 1

[2] Vladimir Kolmogorov. Blossom v: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1(1):43–67, 2009. 1

[3] Guido Schäfer. Weighted matchings in general graphs. Master's thesis, Universität des Saarlandes, 2000. 1, 2