An extension to the problem of Minimum Cost Matching with an Algorithm for Bipartite Graphs

Sergio Giro *
CONICET - FaMAF
Ciudad Universitaria
5000 - Córdoba
Argentina

February 1, 2007

Abstract

Since the matching with minimum cost is the empty matching, the notion of matching with minimum cost is not useful. Usually, the maximum matching with minimum cost is considered. However, the maximum matching with minimum cost is not useful under some circumstances, because it forces to consider only maximal matchings. We present an alternative definition of cost in which both the edges and the nodes are labelled, and we present an algorithm to calculate this cost for bipartite graphs.

Keywords: matching, extended cost, algorithm

Since the matching with minimum cost is the empty matching, the notion of matching with minimum cost is not useful. Usually, the maximum matching with minimum cost is considered. However, the maximum matching with minimum cost is not useful under some circumstances. E. g. , we used matchings to calculate an approximation to the minimum edit distance for graphs in [1]. Since unmatched elements correspond to added and erased arcs, by considering only maximal matchings we quantify only over the editions minimizing the added or erased arcs.

We present an alternative definition of cost in which both the edges and the nodes are labelled: the labels of the edges represent the cost of matching two elements, and the labels of the nodes represent the cost of leaving an element unmatched. Then, we present an algorithm to calculate the minimum extended cost for bipartite graphs.

1 Preliminary definitions

We present some standard definitions just to establish the terminology.

*This technical report is essentially a translation of a section of the degree thesis of the author (written in spanish). Advisor: Dr. Gabriel Valiente. See the bibliographic entry [1] for details.

Definition 1 (Directed graph)**.** A pair of sets $G = (N, E)$ is a directed graph iff $E \subseteq N \times N$, where $X \times Y$ denotes the cartesian product of X and Y. The set N is called the *set of nodes of* G and the set E is called the *set of edges of* G. Two nodes n_1, n_2 are said to be *adjacent* iff $(n_1, n_2) \in E$. If $e = (n_1, n_2)$, we say that n_1 is the *tail* of e and n_2 is the *head* of e. We say that a directed graph $G = (N, E)$ is labelled by the function l iff there exist two sets Σ_N and Σ_E (corresponding to the labels of the nodes and to the labels of the edges, respectively) such that: $l : N \cup E \to \Sigma_N \cup \Sigma_E$ and $\forall n \in N : l(n) \in \Sigma_N \wedge \forall e \in E : l(e) \in \Sigma_E$.

Definition 2 (Bipartite graph)**.** We say that a graph $G = (N, E)$ is bipartite iff N can be partitioned into two sets N_1 and N_2 (the *classes* of G) such that $\forall e \in E : \text{tail}(e) \in N_1$ and $\forall e \in E : \text{head}(e) \in N_2$.

Definition 3 (Matching)**.** A matching for a graph (N, E) is a subset M of E such that $\forall (n_1, n_2) \in M : ((n_1, n_2') \in M \Rightarrow n_2 = n_2') \wedge ((n_1', n_2) \in M \Rightarrow n_1 = n_1')$. We say that a node n is in the matching M (denoted by $n \in M$) iff $\exists n_2 : (n, n_2) \in M \vee \exists n_1 : (n_1, n) \in M$.

Here, we introduce the extended cost of a matching. We call to this cost "extended" since, when all the labels for the nodes are 0, the cost reduces to the usual cost of a matching.

Definition 4 (Extended cost of a matching)**.** Let $G = (N, E)$ be a graph labelled by E. The extended cost of a matching M is

$$\sum_{\{e \in E | e \in M\}} l(e) + \sum_{\{n \in N | n \notin M\}} l(n) .$$

2 Our algorithm

We will reduce the problem of finding the matching with minimum extended cost to a problem of network flow.

Definition 5 (Flow network)**.** Let $G = (N, E)$ be a directed graph, and let $w : E \to \mathbb{N}_0$ (the weight or cost of an edge) and $c : E \to \mathbb{N}_0$ (the capacity of an edge). We say that G is a flow network iff there exists nodes s (the *source* of the network) such that $\nexists n \in N : (n, s) \in E$ and s' (the *sink* of the network) such that $\nexists n \in N : (s', n) \in E$.

Definition 6 (Flow in a network and the associated cost)**.** Let $G = (N, E)$ be a network with capacity c, weight w, source s and sink s'. A flow in G is a function $\text{flow} : E \to \mathbb{N}_0$ such that $\forall e \in E : \text{flow}(e) \leq c(e)$ and

$$\forall n \in N \setminus \{s, s'\} : \sum_{\{e | \text{head}(e) = n\}} \text{flow}(e) = \sum_{\{e | \text{tail}(e) = n\}} \text{flow}(e) .$$

The cost of the flow is $\sum_{e \in E} \text{flow}(e) \cdot w(e)$. The size of the flow is the amount of flow that exits the source, i. e. : $\sum_{(s,n) \in E} \text{flow}(s, n)$. (It can be proven that this amount equals the amount of flow that enters the sink.)

In the following definition, we associate labelled bipartite graphs to flow networks.

Definition 7 (Network associated to a bipartite graph)**.** Given a bipartite graph $G = (N,E)$ labelled by l such that the classes of G are N_1 and N_2, we define the network $T = (N',E')$ as follows:

$$\begin{array}{rcrcl} N' & = & N & \cup & \{s,s',\Lambda_1,\Lambda_2\} \\ L' & = & L & \cup & \{(s,n) : n \in N_1\} \\ & & & \cup & \{(n,s') : n \in N_2\} \\ & & & \cup & \{(\Lambda_1,\Lambda_2)\} \\ & & & \cup & \{(\Lambda_1,n) : n \in N_2\} \\ & & & \cup & \{(n,\Lambda_2) : n \in N_1\}. \end{array}$$

The weight function w is defined by:

$$\begin{array}{lcll} w(e) & = & l(e) & \text{if } e \in E \\ w(\Lambda_1,n) & = & l(n) & \text{if } n \in N_2 \\ w(n,\Lambda_2) & = & l(n) & \text{if } n \in N_1 \\ w(e) & = & 0 & \text{otherwise}. \end{array}$$

The capacity function is defined as:

$$\begin{array}{lcll} c(s,\Lambda_1) & = & \#N_2 & \\ c(\Lambda_1,\Lambda_2) & = & \min(\#N_1,\#N_2) & \\ c(\Lambda_2,s') & = & \#N_1 & \\ c(e) & = & 1 & \text{for the other edges}. \end{array}$$

Here, $\#S$ denotes by amount of elements in the set S.

Theorem 1. *For each matching in a bipartite graph $G = (N,E)$ with classes N_1 and N_2 there is a flow F in the network $T = (N',E')$ associated to G, in such a way that the size of F is maximum and whose cost equals the extended cost of the matching. Moreover, for every flow of maximum size in T, there exists a matching in G whose cost equals the cost of the flow.*

Proof. Since the sum of the capacities of all the edges reaching the sink is $\#N$, a flow cannot be greater than $\#N$. In addition, the following flow has size $\#N$:

$$\begin{array}{lcll} \text{flow}(s,\Lambda_1) & = & \#N_2 & \\ \text{flow}(s,v) & = & c(s,n) & \text{if } n \in N_1 \cup \{\Lambda_1\} \\ \text{flow}(\Lambda_2,s') & = & \#N_1 & \\ \text{flow}(n,s') & = & c(n,s') & \text{if } n \in N_2 \cup \{\Lambda_2\} \\ \text{flow}(n,\Lambda_2) & = & 1 & \text{if } n \in N_1 \\ \text{flow}(\Lambda_1,n) & = & 1 & \text{if } n \in N_2 \\ \text{flow}(e) & = & 0 & \text{for the remaining edges}. \end{array}$$

Then, the maximum size of a flow is $\#N$.

Now, we show that every matching has associated a flow in T of size $\#N$ with the same cost. Let $M = \{(n_{1,1},n_{2,1}),\ldots,(n_{1,n},n_{2,n})\}$. We associate to this matching the

flow:

$$
\begin{array}{lcll}
\text{flow}(e) & = & 1 & \text{if } e \in M \\
\text{flow}(\Lambda_1, n) & = & 1 & \text{if } n \in N_2 \text{ and } n \notin M \\
\text{flow}(n, \Lambda_2) & = & 1 & \text{if } n \in N_1 \text{ and } n \notin M \\
\text{flow}(\Lambda_1, \Lambda_2) & = & \#M & \\
\text{flow}(s, n) & = & 1 & \text{if } n \in N_1 \text{ and } n \in M \\
\text{flow}(n, s') & = & 1 & \text{if } n \in N_2 \text{ and } n \in M \\
\text{flow}(e) & = & 0 & \text{for the remaining edges}.
\end{array}
$$

Given that only the edges in the first three equations have positive weight, the assigned flow is 1 and the weights of the edges are labels corresponding to the cost of the matching, the cost of the flow equals the cost of the matching.

For every flow f of maximum size, we can define a matching M such that $(n_1, n_2) \in M \iff n_1 \in N_1 \wedge n_2 \in N_2 \wedge f(n_1, n_2) \neq 0$. (Since f is a flow, M is effectively a matching.)

Since the flow has size $\#N$, edges with $\text{flow}(e) = 1$ exist from each node not in M to either Λ_2 or Λ_1. Then, by using a similar argument to the previous one, we conclude that the cost of the flow equals the cost of the matching. □

In order to simplify the comprehension the proof, Figures 1 and 2 show a particular matching and the corresponding flow in the network.

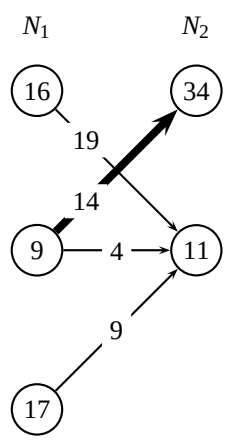

Figure 1: A particular matching

In Figure 1, the thickest line represents two assigned nodes. In Figure 2, a label of the form w, c represents that the weight of the edge is w and the capacity of the edge is c. The thinnest lines represent edges whose flow is 0, the slightly thicker lines represent edges whose flow is 1, the thicker lines represent edges whose flow is 2, and the thickest lines represent edges whose flow is 3.

Theorem 1 allows us to compute the matching having minimum extended cost by calculating the maximum flow with minimum cost in a network with $n+2$ nodes. Since there are algorithms to compute the maximum flow with minimum cost in $O(n^2 \cdot$

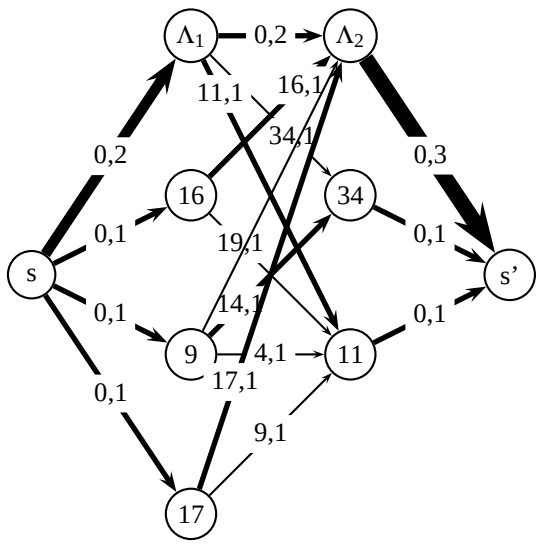

Figure 2: Flow corresponding to the matching in Figure 1

F) time (where F is the size of the maximum flow), we can compute the minimum extended cost using $O(n^3)$ time. Given that the matching is uniquely defined by the flow, the matching can be obtained easily from the flow. Then, by constructing the network and calculating the maximum flow with minimum cost, we can compute the matching with minimum extended cost using $O(n^3)$ and $O(n)$ space. For an extensive bibliography on network flow problems see [2].

3 Conclusion

We presented an alternative definition for the cost of a matching. We expect this definition to be useful when considering edit distances, since the weights of the edges linking matched elements may represent the cost of a replacement and the weights of the the nodes may represent the cost of adding or erasing an element (depending on the class of the bipartite graph in which the node is).

The procedure to calculate the matching with minimum extended cost has the same complexity as the procedure to obtain the maximum matching with minimum cost via the problem of finding the maximum flow with minimum cost. In fact, the reduction of the new problem to the problem of finding the maximum flow with minimum cost is similar to the usual one.

References

[1] Sergio Giro. Implementación y constrastación empírica de una técnica de reconocimiento de imágenes. Degree thesis (presented in order to obtain the degree of Licenciado en Ciencias de la Computación). Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario, 2004. Available at `cs.famaf.unc.edu.ar/~sgiro/tesina.ps`.

[2] M. Veldhorst. A bibliography on network flow problems. *Algorithms Review*, 1(2):97–117, 1990.