Excluding a simple good pair approach to directed cuts

Romeo Rizzi*

May 15, 2000

Dipartimento di Matematica Pura ed Applicata, Università di Padova
Via Belzoni 7, 35131 Padova, Italy
`romeo@euler.math.unipd.it`

Abstract

In 1972, Mader proved that every undirected graph has a good pair, that is, an ordered pair (u, v) of nodes such that the star of v is a minimum cut separating u and v. In 1992, Nagamochi and Ibaraki gave a simple procedure to find a good pair as the basis of an elegant and very efficient algorithm to find minimum cuts in graphs. This paper rules out the simple good pair approach for the problem of finding a minimum directed cut in a digraph and for the more general problem of minimizing submodular functions. In fact, we construct a digraph with no good pair. Note that if a graph has no good pair, then it may not possess a so-called cut-equivalent tree. Benczúr constructed a digraph with no cut-equivalent tree; our counterexample thus extends Benczúr's one.

Key words: good pairs, cut-trees, directed connectivity, counterexample.

1 Introduction

Let $G(V, E)$ be an undirected graph. Given a node set $S \subset V$ with $\emptyset \neq S \neq V$, the *cut* $\delta(S)$ is the set of those edges in E with precisely one endnode in S. Every edge $e \in E$ is also given a non-negative *cost* c_e. The *cost of a cut* $\delta(S)$ is the value $\sum_{e \in \delta(S)} c_e$. The *minimum cut problem* asks for a cut of minimum cost in (G, c).

A cut $\delta(S)$ is called an *s,t-cut* when precisely one of s and t belongs to S. An ordered pair of nodes (s, t) is called *good* if $\delta(t)$ is an s,t-cut of minimum cost. When (s, t) is a good pair two cases are possible: either no cut of minimum cost is an s,t-cut or $\delta(t)$ is a cut of minimum cost. So, if there exists a good pair (s, t) in G, then we can reduce the minimum cut problem by identifying nodes s and t. In 1972, Mader [7] proved that, when G is undirected, then a good pair always exists. In 1992, Nagamochi and Ibaraki [8] gave a simple procedure to find a good pair as the basis of an elegant and very efficient algorithm to find minimum cuts in graphs. This approach was subsequently generalized to hypergraphs [6], symmetric submodular functions [10] and more general symmetric set functions [9, 11].

We address the following question:

> Is there any possibility that the simple good pair approach can be extended to the problem of finding a minimum dicut in a directed graph, or even to the more general problem of minimizing submodular functions?

*Research carried out with financial support of the project TMR-DONET nr. ERB FMRX-CT98-0202 of the European Community and partially supported by a post-doc fellowship by the "Dipartimento di Matematica Pura ed Applicata" of the University of Padova.

Let $D(V, A)$ be a directed graph. Given a node set $S \subset V$ with $\emptyset \neq S \neq V$, the *dicut* $\delta^+(S)$ is the set of those arcs $\vec{uv} \in A$ with $u \in S$ and $v \notin S$. Every arc $a \in A$ is also given a non-negative *cost* c_a. Define the *cost of a dicut* $\delta^+(S)$ as $c(\delta^+(S)) = \sum_{a \in \delta^+(S)} c_a$. The *minimum dicut problem* asks for a dicut of minimum cost in (D, c). A dicut $\delta^+(S)$ is called a u, v-*dicut* when precisely one of u and v belongs to S. We denote by $\delta^-(S)$ the dicut $\delta^+(V \setminus S)$. A pair of nodes (u, v) is called *good* if at least one of the following dicuts is a u, v-dicut of minimum cost: $\delta^+(u), \delta^+(v), \delta^-(u), \delta^-(v)$. When (u, v) is a good pair two cases are possible: either no dicut of minimum cost is a u, v-dicut or at least one of the four dicuts listed above is a dicut of minimum cost. So, if there exists a good pair (u, v) in D, then we can reduce the minimum dicut problem by identifying nodes u and v. Hao and Orlin [5] gave an algorithm with low time complexity bound for the minimum dicut problem. Their approach bases however on flows and no good-pair-type algorithm is known at present.

In the next section, a digraph with no good pair is constructed. In [1], Benczúr had given a digraph with no cut-equivalent tree (definitions in Section 3). The counterexample of Benczúr was meant to point out an invalidating error in constructions proposed in the literature to obtain such trees. Our counterexample is stronger than the one of Benczúr in the sense that, having no good pair, our digraph does not admit any cut-equivalent tree also.

2 The construction

The digraph $\mathcal{D}$ given in Figure 1 has no good pair.

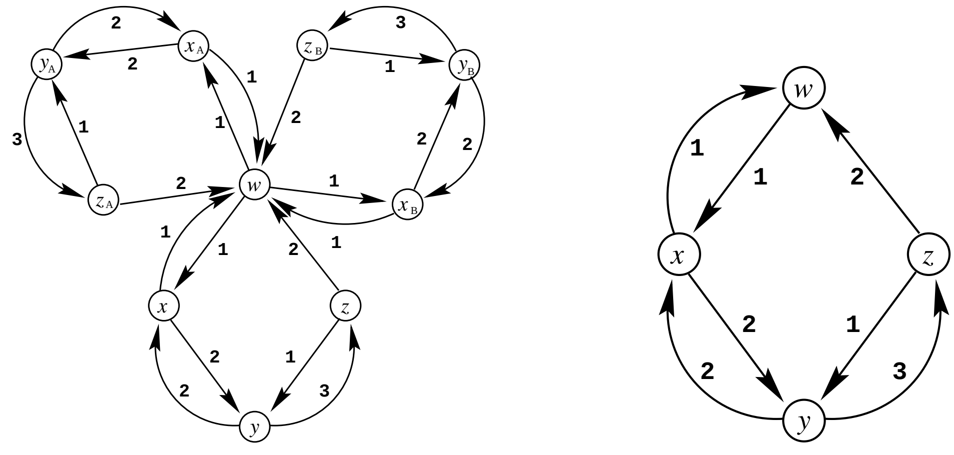

Figure 1: A digraph $\mathcal{D}$ having no good pair. A block of $\mathcal{D}$.

In $\mathcal{D}$, $\min\{c(\delta^-(v)) : v \in V(\mathcal{D})\} = 3$ and $\min\{c(\delta^+(v)) : v \in V(\mathcal{D})\} = 3$. However, for any two nodes $u, v \in V(\mathcal{D})$, a minimum u, v-dicut has cost at most 2. To convince ourselves that this is indeed the case consider a single block of $\mathcal{D}$. (Figure 1 on the right). Here $c(\delta^-(\{x, y\})) = 2$ and $c(\delta^-(\{z, y\})) = 2$. So, if u and v belong to a same block, then either $\delta^-(\{x, y\})$ or $\delta^-(\{z, y\})$ is a u, v-dicut of cost 2. If otherwise u and v belong to different blocks then consider u and w. Since u and w belong to a same block, then in $\mathcal{D}$ there exists a u, w-dicut of cost at most 2. Moreover in $\mathcal{D}$ every minimum u, w-dicut is also a u, v-dicut.

3 Cut-equivalent tree

In [3], Gomory and Hu introduced the fundamental notion of cut-equivalent tree of an undirected graph. In this section, we define cut-equivalent trees for digraphs and observe that the digraph $\mathcal{D}$ given in Figure 1 does not admit any cut-equivalent tree also. An incorrect result of Schnorr [12], which also lead to the derivation of further incorrect results in [4], stated that every digraph has a cut-equivalent tree. The error was pointed out by Benczúr [1], who first gave a digraph with no cut-equivalent tree.

Let $D(V, A)$ be a digraph and c_a be a non-negative cost assigned to every $a \in A$. Let T be an undirected tree with $V(T) = V$ and w_e be a non-negative weight assigned to every edge $e \in E(T)$. The pair (T, w) is a *cut equivalent tree* of (D, c) if for every two nodes $u, v \in V$ the following property holds: if e is any edge of minimum weight in the unique path between u and v in T and S_1 and S_2 are the two connected components in the graph obtained from T by deleting e, then $\delta^+(S_1)$ or $\delta^+(S_2)$ is a minimum u,v-dicut for (D, c).

It is well known that every tree has a leaf. Let v be a leaf of T and let u be the neighbour of v in T. Since uv is an edge of minimum weight in the unique path between u and v in T, then $\delta^+(v)$ or $\delta^-(v)$ is a minimum u,v-dicut for (D, c), once (T, w) is a cut equivalent tree. This means that (u, v) is a good pair. Hence, our counterexample is stronger than the one of Benczúr in the sense that, having no good pair, our digraph does not admit any cut-equivalent tree either.

Acknowledgements

Thanks are due to the anonymous referee. His advice helped improving the presentation.

References

[1] Benczúr, A.A.: Counterexamples for directed and node capacitated cut-trees. SIAM Journal on Computing **24 (3)**, 505–510 (1995)

[2] Frank, A.: On the edge-connectivity algorithm of Nagamochi and Ibaraki. (1994)

[3] Gomory, R.E., Hu, T.C.: Multi-terminal network flows. J. Soc. Indust. Appl. Math. **9**, 551–570 (1961)

[4] Gusfield, D., Naor, D.: Efficient algorithms for generalized cut-trees trees. Networks **21**, 505–520 (1991)

[5] Hao, J., Orlin, J.B.: A faster algorithm for finding the minimum cut in a graph. Proc. of 3rd ACM-SIAM Symp. on Discrete Algorithms, 165–174 (1992)

[6] R. Klimmek and F. Wagner, A Simple Hypergraph Min Cut Algorithm. *Internal Report B 96-02* Bericht FU Berlin Fachbereich Mathematik und Informatik (1995). http://www.inf.fu-berlin.de/inst/pubs/index96.html

[7] Mader, W.: Über minimal n-fach zusammenhängende, unendliche Graphen und ein Extremalproblem. Arch. Math. **23**, 553–560 (1972)

[8] Nagamochi, H., Ibaraki, T.: Computing edge connectivity in multigraphs and capacitated graphs. SIAM Journal on Discrete Mathematics **5**, 54–66 (1992)

[9] H. Nagamochi and T. Ibaraki, A note on minimizing submodular functions. *Information Processing Letters* 67 (1998) 239–244.

[10] Queyranne, M.: Minimizing symmetric submodular functions. Mathematical Programming **82**, 3-12 (1998)

[11] Rizzi, R.: On minimizing symmetric set functions. *to appear on Combinatorica*

[12] Schnorr, C.P.: Bottlenecks and edge connectivity in unsymmetrical networks. SIAM Journal on Computing **8**, 265–274 (1979)