UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

ON MINIMIZING SYMMETRIC SET FUNCTIONS

Romeo Rizzi

July 1999

Technical Report # DIT-02-0055

Also: in *Combinatorica* 20 (3) (2000) 445--450.

On minimizing symmetric set functions

Romeo Rizzi*

July 27, 1999

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
romeo@cwi.nl

Abstract

Mader proved that every loopless undirected graph contains a pair (u,v) of nodes such that the star of v is a minimum cut separating u and v. Nagamochi and Ibaraki showed that the last two nodes of a "max-back order" form such a pair and used this fact to develop an elegant min-cut algorithm. M. Queyranne extended this approach to minimize symmetric submodular functions. With the help of a short and simple proof, here we show that the same algorithm works for an even more general class of set functions.

Key words: max-back order, minimum cut, symmetric submodular functions.

Main Section

Let V be a finite set. A value $d(\{S,T\})$ is given for every unordered pair of disjoint subsets S,T of V. For convenience, function d is called a *map on* V, even if it is actually defined on a subset of $2^V \times 2^V$. We also rely on the shorthand $d(S,T) = d(\{S,T\})$ and leave the fact that $d(S,T) = d(T,S)$ as understood. Function d is called *monotone* if $d(S,T') \leq d(S,T)$ for any S,T disjoint and $T' \subseteq T$. Finally, d is *consistent* if $d(A, W \cup B) \geq d(B, W \cup A)$ whenever A,B,W are disjoint sets such that $d(A,W) \geq d(B,W)$. As an example, when $G = (V,E)$ is an undirected graph, then $d(S,T) = |\{st \in E\ : s \in S, t \in T\}|$ for any disjoint sets $S,T \subseteq V$, is a monotone and consistent map on V. A subset S of V is said *nontrivial* when $\emptyset \neq S \neq V$. We give an efficient algorithm to solve the following problem (*minimum bipartition problem*):

> *Given a finite set V and a monotone and consistent map d on V, find a nontrivial subset S of V for which $d(S, V \setminus S)$ is minimum.*

A *max-back order* for (V,d) is an ordering $v_1, v_2, \ldots, v_n$ of the elements in V such that

$$d(v_i, \{v_1, \ldots, v_{i-1}\}) \geq d(v_j, \{v_1, \ldots, v_{i-1}\}) \qquad \text{for } 2 \leq i < j \leq n$$

Let s and t be two elements of V. An *st-set* is a subset S of V with $|S \cap \{s,t\}| = 1$.

An ordered pair (s,t) of elements of V is *good* if $d(\{t\}, V \setminus \{t\}) \leq d(S, V \setminus S)$ holds for every st-set S. Before the end of this section we will prove the following lemma:

*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.

Lemma 1 *Let $v_1, \ldots, v_n$ be a max-back order for (V, d). Then (v_{n-1}, v_n) is good for (V, d).*

Lemma 1 gives an efficient procedure, called *a_Good_Pair*, to find a good pair. When (s, t) is a good pair two cases are possible: either $\{t\}$ is an optimal solution to our problem, or no optimal solution S to the problem is an st-set. This motivates the following definitions: Let s and t be any two elements of V. Consider *identifying* s and t into a single new element v_{st} thus obtaining a new set $V_{st} = V \setminus \{s,t\} \cup \{v_{st}\}$. Now, to reconsider a subset X of V_{st} as a subset of V, we define $\langle X \rangle = X$ if $v_{st} \notin X$ and $\langle X \rangle = X \setminus \{v_{st}\} \cup \{s,t\}$ if $v_{st} \in X$. When S and T are disjoint subsets of V_{st} then $\langle S \rangle$ and $\langle T \rangle$ are disjoint subsets of V and we define:

$$d_{st}(S, T) = d(\langle S \rangle, \langle T \rangle)$$

Note that, when d is a monotone and consistent map on V, then d_{st} is a monotone and consistent map on V_{st}. To conclude, the following algorithm solves the minimum bipartition problem.

Algorithm 1 MIN_BIPARTITION (V, d)

1. if $|V| = 2$ then return either of the two nontrivial subsets of V;
2. $(s,t) \leftarrow a\_Good\_Pair(V, d)$;
3. return the best set among $\{t\}$ and $\langle Min\_Bipartition(V_{st}, d_{st}) \rangle$;

We have now enough motivation to prove Lemma 1.

Proof of Lemma 1: The lemma is true for $n = 3$ since $d(v_2, v_1) \geq d(v_3, v_1)$ implies $d(\{v_1, v_3\}, v_2) \geq d(\{v_1, v_2\}, v_3)$ for d is consistent. Let $\mathcal{S}$ be any $v_n v_{n-1}$-set. We must show that:

$$d(\mathcal{S}, V \setminus \mathcal{S}) \geq d(\{v_n\}, V \setminus \{v_n\}) \tag{1}$$

Clearly, $v_{v_1 v_2}, v_3, v_4, \ldots, v_n$ is a max-back order for $(V_{v_1 v_2}, d_{v_1 v_2})$. Thus, either (1) follows by induction or $\mathcal{S}$ is a $v_1 v_2$-set. Since d is monotone, $v_1, v_{v_2 v_3}, v_4, \ldots, v_n$ is max-back for $(V_{v_2 v_3}, d_{v_2 v_3})$ and either (1) follows or $\mathcal{S}$ is a $v_2 v_3$-set. Assume therefore that $\mathcal{S}$ is both a $v_1 v_2$-set and a $v_2 v_3$-set. But then $\mathcal{S}$ is not a $v_1 v_3$-set and to derive (1) it suffices to show that $v_2, v_{v_1 v_3}, v_4, \ldots, v_n$ is max-back for $(V_{v_1 v_3}, d_{v_1 v_3})$. Assume on the contrary $d_{v_1 v_3}(v_k, v_2) > d_{v_1 v_3}(v_{v_1 v_3}, v_2)$. However $d(v_2, v_1) \geq d(v_3, v_1)$ and $d(v_3, \{v_1, v_2\}) \geq d(v_k, \{v_1, v_2\})$ since $v_1, \ldots, v_n$ is max-back for (V, d). Since d is monotone and consistent, we get $d(v_3, \{v_1, v_2\}) \geq d(v_k, \{v_1, v_2\}) \geq d(v_k, v_2) = d_{v_1 v_3}(v_k, v_2) > d_{v_1 v_3}(v_{v_1 v_3}, v_2) = d(\{v_1, v_3\}, v_2) \geq d(v_3, \{v_1, v_2\})$, a contradiction. $\square$

Some Applications

A couple of observations and a list of applications will follow. In Application 1, Queyranne's important result on minimizing symmetric submodular functions is derived as a special case of our framework. The generalization is strict as shown in Applications 2 and 3.

Note that Algorithm 1 can also be used to solve maximization problems when $-d$ is a monotone and consistent map. In practice it follows that we can maximize $d'(S, V \setminus S)$ over the nontrivial subsets S of V whenever d' is a map on V with the following properties:

(i) $d'(S, T') \geq d'(S, T)$ for any S, T disjoint and $T' \subseteq T$ – (reverse monotonicity);

(ii) $d'(A, W \cup B) \geq d'(B, W \cup A)$ whenever A, B, W are disjoint sets such that $d'(A, W) \geq d'(B, W)$ – (consistency);

In contrast, maximizing $d(S, V \setminus S)$ for a generic monotone and consistent map d is an *NP*-complete problem since it contains as a special case the max-cut problem, which is known to be *NP*-complete [4].

Application 1 (symmetric submodular functions [9])

Consider a finite set V and a real function f on 2^V. We are interested in finding a nontrivial subset of V which minimizes f. For this reason we consider an ordered pair (s, t) of elements of V to be *good* if $\{t\}$ is an st-set minimizing f. For any two disjoint subsets S, T of V let us define

$$d_f(S, T) = f(S) + f(T) - f(S \cup T)$$

If f is *symmetric* (that is, $f(S) = f(V \setminus S)$ for every subset S of V), then a pair is good with respect to f if and only if it is good with respect to d_f.

Note that d_f is consistent. Assume indeed A, B, W to be disjoint and such that $d_f(A, W) \geq d_f(B, W)$. This means $f(A) + f(W) - f(A \cup W) \geq f(B) + f(W) - f(B \cup W)$. But then $d_f(A, B \cup W) = f(A) + f(B \cup W) - f(A \cup B \cup W) \geq f(B) + f(A \cup W) - f(A \cup B \cup W) = d_f(B, A \cup W)$.

So we are interested in characterizing those f for which d_f is monotone, that is, $d_f(S, T_1) \leq d_f(S, T_1 \cup T_2)$ for any S, T_1, T_2, all disjoint and non-empty. In terms of f this means, $f(S) + f(T_1) - f(S \cup T_1) \leq f(S) + f(T_1 \cup T_2) - f(S \cup T_1 \cup T_2)$, or equivalently, $f(S \cup T_1 \cup T_2) + f(T_1) \leq f(T_1 \cup T_2) + f(S \cup T_1)$. Hence d_f is monotone if and only if f satisfies the *submodular inequality* $f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$ for any sets A and B such that $A \setminus B, B \setminus A, A \cap B$ and $V \setminus A \setminus B$ are all non-empty.

In [8], Nagamochi and Ibaraki called such a function f *crossing submodular* and observed that the approach proposed by Queyranne in [9] to minimize symmetric submodular functions (where the submodular inequality has to hold for any sets A and B), was also valid for symmetric crossing submodular functions.

Algorithm 1 was first employed by Nagamochi and Ibaraki [7] to find minimum cuts in undirected graphs. A simple proof of the validity of Nagamochi and Ibaraki's min-cut algorithm had been obtained by Frank [2] and Stoer and Wagner [10], while Queyranne was deriving his important, but less simple, extension. Recently, in [3], Fujishige gave another short proof of the validity of Nagamochi and Ibaraki's min-cut algorithm and indicated how to employ his arguments to obtain a compact proof of Queyranne's result.

In the next application we show that our simple approach actually embraces an even broader class of problems.

Application 2 (short distance partitions)

Let G be a graph. A symmetric distance $\lambda(u, v)$ is given for every two nodes u, v. Assume we want to bipartition the node set V of G as to keep the maximum distance among two nodes on different sides of the partition as small as possible. Even if this problem can easily be solved directly, define $d(S, T) = max\{\lambda(s, t) : s \in S, t \in T\}$. Note that d is a monotone and consistent map in general. Consider the graph $(V, E) = (\{a, b, c, d\}, \{ab, bc, cd, da)\})$ and for every $u, v \in V$ define the distance $\lambda(u, v)$ as the length of a shortest path between u and v. (Hence $\lambda(a, c) = \lambda(b, d) = 2$, and $\lambda(a, b) = \lambda(b, c) = \lambda(c, d) = \lambda(a, d) = 1$). The sets $S = \{a, c\}$ and $T = \{a, d\}$ show that the function f on 2^V defined by $f(S) = d(S, V \setminus S)$ for every $S \subseteq V$, is not crossing submodular in this special case.

Application 3 (critical cuts)

Let (G, w) be a weighted graph. Assume to be interested in those spanning trees T of G such that $max\{w(e) : e \in T\}$ is as small as possible. Then it is natural to define the cost of a cut $\delta(S)$ as the minimum of $w(e)$ for $e \in \delta(S)$ and to search for a cut of maximum cost. This is clearly a bottleneck problem and admits a direct and simple solution.

Define $d(S, T) = min\{w(e) : e$ has an endpoint in S and the other in $T\}$. Note that $-d$ is a monotone and consistent map. This is indeed a reformulation of the above problem on short distance partitions (see Application 2). Hence we also have that the function f on 2^V defined by $f(S) = d(S, V \setminus S)$ for every $S \subseteq V$, is not crossing supermodular in general. (A function f is called *crossing supermodular* if $f(A \cap B) + f(A \cup B) \geq f(A) + f(B)$ for any sets A and B such that $A \setminus B, B \setminus A, A \cap B$ and $V \setminus A \setminus B$ are all non-empty).

Application 4 (minimum cuts in hypergraphs [5])

Hypergraphs generalize graphs. When $G = (V, H)$ is an *hypergraph*, then the *hyperedges* in H are arbitrary subsets of the node set V. Thus a graph is an hypergraph in which every hyperedge has cardinality 2. Klimmek and Wagner [5] proposed a Nagamochi-Ibaraki type algorithm to find a minimum cut in an hypergraph. Indeed, the cut function of an hypergraph is symmetric and submodular [5, 9]. Consider the bottleneck version of this problem, that is, finding a cut which minimizes the maximum weight of an hyperedge belonging to it. Submodularity is lost but still we would have to deal with a monotone and consistent map.

Application 5 (partitions minimizing ambivalence)

Let G be a graph. Partition the node set V as $S \cup (V \setminus S)$ in such a way as to minimize the number of nodes with neighbors in both sides of the partition. This problem can be formulated as an hypergraph min cut problem (for every node v, we have an hyperedge h_v made of the neighbors of v in G). The problem hence falls in the framework of Stoer and Wagner [10], but also in that of Queyranne [9], or finally in our framework.

Acknowledgments

I am indebted to Tamás Fleiner, András Frank and Bert Gerards. Tamás Fleiner patiently helped in cleaning the presentation and was of great support also giving motivation. András Frank contributed with suggestions, observations and questions. I thank all three of them.

References

[1] Jr. L.R. Ford and D.R.Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N.J. (1962)

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

[3] S. Fujishige, Another simple proof of the validity of Nagamochi and Ibaraki's min-cut algorithm and Queyranne's extension to symmetric submodular function minimization. *Journal of the Operations Research Society of Japan* 41 (1998) 626–628.

[4] R.M. Karp, Reducibility among combinatorial problems. *Complexity of Computer Computations* (R.E. Miller, J.W. Thatcher eds), Plenum Press, New York (1972) 85–103.

[5] 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

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

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

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

[9] M. Queyranne, Minimizing symmetric submodular functions. *Mathematical Programming* 82 (1998) 3-12. Also in Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, January 1995, 98-101.

[10] M. Stoer and F. Wagner, A simple min-cut algorithm. *Journal of the ACM* 44 (1997) 585–591. Also in Lecture Notes in Computer Science 855 (1994) 141–147.