Univ. Beograd. Publ. Elektrotehn. Fak.
Ser. Mat. 3 (1992), 27–33

SOME REMARKS ON THE PROBLEM OF MULTILATERAL COMPENSATION

Slobodan Simić, Vladan Milanović

In this note we first give an exact (polynomial time) algorithm for solving the problem of multilateral compensation. For the large scale instances we propose some heuristics for finding suboptimal solutions.

0. INTRODUCTION

Let $D(w) = (V, A; w)$ be a weighted digraph with the vertex set V and the weight function $w : A \mapsto \mathbb{N}_0$ defined on its arc set A, i.e. $w(a)$ is a nonnegative integer for each $a \in A$. If $w(a) = 0$ for some a, we will sometimes ignore a as an element of A. Given $D(w)$, D denotes the underlying digraph. A reduction $D(w')$ of $D(w)$ is a weighted digraph obtained from $D(w)$ by reducing the weights of some arcs, i.e. $D(w') = (V, A; w')$ with $0 \leq w'(a) \leq w(a)$ for each $a \in A$. Two reductions $D(w')$ and $D(w'')$ are complementary (with respect to $D(w)$) if $w(a) = w'(a) + w''(a)$ for each $a \in A$. The weight of some digraph is the sum of its arc weights. All other terminology, not mentioned below, follows [**3**].

Let C be an (oriented) cycle of $D(w)$, or D. If the weights of all arcs of C are reduced by the same (integral) amount so that the resulting arc weights are still nonnegative, we say that C is partially compensated or, for short, only compensated. In particular, if the weight of at least one arc of C became zero, we rather say that C is totally compensated. Clearly, any compensation of some cycle, or some cycles in turn, gives a reduction of $D(w)$. A reduction obtained as a sequence of cycle compensations is called a compensation of $D(w)$. A total compensation is a compensation which does not admit any further cycle compensation. We are now in the position to state our problem:

For a weighted digraph $D(w)$, find a compensation having the minimum weight.

1991 AMS Subject Classification: Primary 05 C 85, Secondary 05 C 90

This problem will be in further addressed as the problem of multilateral compensation (MLC) problem. Before attempting to solve it we give some comments. At first moment it seems that a simple greedy algorithm consisting of cycle compensations leads to a solution. To see that this is not true, consider the following example:

Both digraphs (a) and (c) are total compensations of the digraph (b), each of them obtained by only one cycle compensation.

A compensation which reduces the weight of all arc to zero is called a perfect compensation. In general, not every weighted digraph admits a perfect compensation. For instance, if the underlying digraph is not strongly connected, then the arcs between different components do not belong to any cycle and hence their weights cannot be reduced.

The balance of some vertex $v \in V$ is the difference between the weight sums of outgoing and ingoing arcs at v, i.e.

$$\delta(v) = \sum_{vx \in A} w(vx) - \sum_{yv \in A} w(yv),$$

where ab denotes the arc between vertices a and b. If $\delta(v) = 0$ for some vertex v, we say that v itself is balanced, or a 0-*vertex*; otherwise, if $\delta(v) > 0$ (or $\delta(v) < 0$) holds, then v is non-balanced and moreover, it is a p-vertex (resp. an n-vertex). The meaning of vertex sets V_0, V_p, V_n (which partition V) is now evident from the above context.

1. SOME PRELIMINARY OBSERVATIONS

Our MLC problem (as usual with problems of combinatorial optimization) can be formulated as an LP (linear programming) problem. For this purpose we first give some more notation.

Recall, an elementary cycle is a cycle passing through each vertex at most once. Denote by $\mathcal{C}$ the set of all elementary cycles of $D(w)$ (or D). Let $\mathcal{C}_a \subseteq \mathcal{C}$ be the set of those cycles containing an arc a. For any cycle $C_i \in \mathcal{C}$, let l_i be its length (number of arcs), and let x_i be the amount by which the weights of its arcs are reduced. Then our MLC problem (in the maximization form) becomes the

following LP problem:

$$\max\left\{\sum_{C_i\in\mathcal{C}} l_i x_i\right\}, \quad \text{subjected to} \quad \sum_{C_i\in\mathcal{C}_a} x_i \le w(a) \qquad (a\in A),\ x_i \ge 0.$$

Although this formulation gives an immediate algorithm, it is far from being practical. One reason is that it concerns the whole cycle space whose cardinality, in the worst case, is exponential in the size of digraph, i.e. number of vertices. So we are in start restricted only to digraphs of small sizes. The only significance of this formulation is the theoretical one.

We now turn our attention to more promising technique based on combinatorial tools. The key observation is the next theorem (a straightforward reformulation of the EULER theorem, see [**3**]).

Theorem 1. *A weighted digraph $D(w)$ admits a perfect compensation if and only if all its vertices are balanced.*

Note first that the condition is necessary (any compensation does not change the vertex balance). To prove the sufficiency, we will transform our weighted digraph $D(w)$ to multidigraph D_m as follows: each arc a of $D(w)$ is replaced in D_m by $w(a)$ parallel arcs joining the same pair of vertices. Since each vertex of $D(w)$ is balanced, it follows that each vertex of D_m has equal ingoing and outgoing degrees. By the EULER theorem it follows that the arc set of D_m can be covered by disjoint cycles, and thus $D(w)$ admits a perfect compensation. □

An immediate consequence of this theorem is that our original problem is equivalent to the problem of finding a reduction of $D(w)$ which is both, balanced and has the largest weight. In other words, if $D(w)$ is represented as the sum of two complementary reductions, say $D(w')$ and $D(w'')$, where the former one is balanced, then we have to find either $D(w')$ of maximum weight, or $D(w'')$ of minimum weight. In the next section we will provide an efficient solution based on the latter possibility.

2. EXACT ALGORITHM FOR MLC PROBLEM

Suppose now we want to decompose $D(w)$ as the (formal) sum of two complementary reductions $D(w')$ and $D(w'')$, where $D(w')$ is balanced. Of course, we may assume that $D(w)$ is not balanced itself, since otherwise our problem is trivial. To get the desired decomposition, we first extend $D(w)$ to $D'(W)$ by adding two vertices, say s (source) and t (sink), such that:

— s is joined to all vertices of positive balance (belonging to V_p), and if $x\in V_p$ then the weight of the arc sx is $\delta(x)$;

— each vertex of negative balance (belonging to V_n) is joined to t, and if $y\in V_n$ then the weight of the arc yt is $-\delta(y)$.

Note now that all vertices in $D'(W)$, except s and t, are balanced.

Theorem 2. *If s and t are the source and the sink of $D'(W)$, then $\delta(s) = -\delta(t)$.*

If we interpret the arc weights of $D'(W)$ as the arc flows, then the assertion is an immediate consequence of the conservation law, i.e. the first Kirchhoff's law. □

If further on we interpret $D'(W)$ as a capacited network with arc capacities equal to the arc weights, then the maximum amount of flow that we can push from s to t is bounded from above by $\delta(s)$. Moreover, this amount of flow can be pushed indeed. To see this, add to $D'(W)$ an arc ts with a weight $\delta(s)$. The digraph obtained has all vertices balanced, and that (by Theorem 1) proves our claim.

Suppose now we have pushed in $D'(W)$ the maximum flow from s to t. In other words, we have some flow $\varphi(a)$ $(0 \leq \varphi(a) \leq w(a))$ for each $a \in A$ (an arc of $D(w)$); flows in arcs incident to s or t are maximal and equal to their capacities. We next show how this flow pattern induces a desired decomposition of $D(w)$. For each $a \in A$ define $w'(a) = w(a) - \varphi(a)$ and $w''(a) = \varphi(a)$. Then all vertices of $D(w')$ are balanced (by the conservation of flow at each vertex). On the other hand, if we have some decomposition of $D(w)$ as required, it induces (by reversing the above) a maximal flow in the extended digraph $D'(W)$. So to get a solution to our optimization problem we only need to find a maximum flow for which the sum of flows through all arcs of $D(w)$ is the smallest possible. In other words, if we take a cost function which assigns one unit for sending a unit amount of flow through each arc, then our problem becomes a minimum cost flow (MCF) problem in the extended digraph $D'(W)$, where $\delta(s)$ is a value of flow (i.e. target flow) to be pushed from s to t. These conclusions can be summarized in the following theorem.

Theorem 3. *The MLC problem is (polynomially) reducible to the MCF problem.*

We are now in a position to comment the efficiency of algorithms by which we can solve the MLC problem. Since the MCF problem can be solved by the algorithm of complexity $O(n^2m)$, where n is the number of vertices and m the number of arcs (see [**6**, **7**], the same applies to MLC problem since the reduction efforts can be neglected. Furthermore, to speed up any MCF algorithm applied, we can first decompose D (i.e. $D(w)$) into strongly connected components and then treat each component in turn. (Note that the algorithm for determining strongly connected components of any digraph is linear in m, see [**8**]). At this place it is worthwhile mentioning that during the experiments with large digraphs the authors have always observed a situation with one giant component of strong connectedness and a few small ones. The explanation of this phenomenon was recently given in [**5**].

3. LARGE SCALE INSTANCES OF MLC PROBLEM

In this section we are concerned with some ideas of treating the instances of MLC problem where the size of the digraph $D(w)$ is very large (say, number of vertices is more than 10^4). If so the running time of the algorithmic solution is too big (usually more than 24 hours). In further we will discuss some possibilities of finding not optimal, but rather near optimal solutions.

The first very general strategy was based on making use of some approximations to our original problem. For example, we can eliminate all arcs with small weights, or all vertices which are very unbalanced etc. The most powerful approximation from the experimental evidence, was obtained by eliminating all vertices with small capacities.

The capacity of some vertex $v \in V$ is the minimum of weight sums of outgoing and ingoing arcs at v, i.e.

$$c(v) = \min\left\{ \sum_{vx \in A} w(vx), \sum_{yv \in A} w(yv) \right\}.$$

In the rest of this section we are offering a procedure for finding an approximate solution to the original MLC problem (not with modified input instance) which is (by the experimental evidence) very close to the optimal one. It consists of two phases. In the first one (Phase I) we find a total compensation of $D(w)$ (not necessarily of minimum weight) in time which is comparatively shorter than the time required by the exact algorithm. In the second one (Phase II) we make use of an iterative procedure to reduce the weight of the total compensation so far obtained, keeping it total all the time. The main benefit of this approach is that we can stop our iterative procedure whenever we want — we always have a feasible solution to our problem.

Phase I: Let $D'(W)$ be the extended digraph of $D(w)$ as described in the previous section. Consider then a sequence of digraphs $D_1(w_1), \ldots, D_k(w_k)$ $(k < n)$ obtained as follows. Let $D_1(w_1) = D'(W)$. $D_{i+1}(w_{i+1})$ is obtained from $D_i(w_i)$ by making use of the layered digraph L_i which consists of: the source s (0-th layer), the sink t ($(i+2)$-th layer) and $i+1$ layers between; the j-th layer of L_i $(1 \leq j \leq i+1)$ consists of those vertices of $D_i(w_i)$ which are at distance j from the source s, where by the distance we assume the length of the shortest path in the underlying digraph. Each vertex from some layer is joined by an arc (of the same weight) to any vertex from the next layer, if they were adjacent in $D_i(w_i)$. To get $D_{i+1}(w_{i+1})$, we first push from s to t in L_i the maximum (possible) flow. This flow pattern is then used to reduce the arc weights of $D_i(w_i)$ and hence to obtain $D_{i+1}(w_{i+1})$. This procedure is repeated until the total flow through all layered digraphs became equal to maximum flow (target flow) in $D'(W)$. By making use of the max-flow min-cut theorem of Ford and Fulkerson, we can easily see that this flow can be indeed pushed (in this way) from s to t in digraph $D'(W)$.

Phase II: We first notice that in the final digraph, i.e. $D_k(w_k)$, the vertices s and t are isolated. By deleting s and t we get a digraph with all vertices being balanced. This digraph was earlier (within the exact algorithm) denoted by $D(w')$. Its complementary reduction (with respect to $D(w)$) is $D(w'')$. It corresponds to a total compensation of $D(w)$. Generally, its weight need not be minimal. The purpose of this phase is to exchange some weighted paths between the complementary reductions in order to improve the feasible solution obtained so far. The basic idea is as follows. Let a and b be two vertices of $D(w')$, and $D(w'')$ as well. Let

P' of length l' (P'' of length l'') be a path from a to b in $D(w')$ (resp. $D(w'')$). Suppose also $\varepsilon > 0$ is some sufficiently small number. If $l' < l''$, we can increase the arc weights of P' in $D(w')$ by decreasing the arc weights of corresponding arcs in $D(w'')$, and also decrease the arc weights of P'' in $D(w'')$ by increasing the arc weights of corresponding arcs in $D(w'')$. With this modification our complementary reductions remain as required: the only difference is that the weight of $(l'' - l')\varepsilon$ is exchanged between the reductions. In other words, our feasible solution is improved by the above amount. To realize these modifications we make use of some heuristics. Any other approach is time consuming.

4. CONCLUSION

In this paper we have offered an exact algorithm for solving the MLC problem reducing it to the MCF problem. By applying the algorithm of Busacker and Gowen for solving MCF we have achieved the satisfactory results only for graphs up to 500 vertices (the running time was within few hours on the target machine). With input graphs having more than 10 000 vertices the running time was prohibitively large. Applying the proposed two phase heuristic, we were able to achieve the suboptimal solutions in a fraction of minute; the relative deviation (w.r.t. the optimal solutions) was always less than 5% for the graphs up to 500 vertices.

Some further experimental results are summarized in the next table.

No. vertices	No. arcs	% MLC	time (min)
156	725	6.02	0.5
1641	21597	11.84	3.4
6336	127631	11.52	51.5
9861	231090	17.69	95.0
12417	363629	22.80	103.5

Besides the number of vertices and arcs of input graphs, we have included the relative amount being compensated (w.r.t. the total weight of the input graph, i.e. the sum of all arc weights at beginning) and the running time of our heuristic.

More details on experiments (and implementations as well) are reported at the meeting of FINASOFT (see [9]). Finally we need to add that this work has been motivated by an actual problem encountered by "The Social Accountancy Service of Yugoslavia". The authors are very thankful to this Institution for enabling us to use their computer facilities in conducting the large amount of experiments on real data.

REFERENCES

1. : *A procedure for determining a family of minimal cost network flow patterns.* ORO Technical Paper, **15** (1961).

2. : *Flows in Networks.* Princeton University Press, 1962.

3. : *Graph Theory.* Addison-Wesley, 1969.

4. : *A new method of solving transportation network problems.* J. Op. Res. Soc. Japan, **3** (1960), 27–87.

5. : *The transitive closure of a random digraph.* Random Structures Algorithms **1** (1) (1990), 73–93.

6. : *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

7. : *Discrete Optimization Algorithms.* Prentice-Hall, Inc., 1983.

8. : *Depth-first search and graph algorithms.* SIAM J. Comput., **1** (1972), 146–160.

9. *Collected papers from the seminar: Multilateral Compensation in the system of paying* (held in Belgrade, March 25–26, 1992).

Faculty of Electrical Engineering,
University of Belgrade,
P.O. Box 816, 11001 Belgrade,
Yugoslavia

(Received December 28, 1991)