Balanced Network Flows. V. Cycle-Canceling Algorithms

Christian Fremuth-Paeger, Dieter Jungnickel
*Lehrstuhl für Diskrete Mathematik, Optimierung und Operations Research, University of Augsburg,
D-86135 Augsburg, Germany*

We discuss Anstee's approach for solving generalized matching problems by solving an ordinary flow problem on a balanced network first. We give a description of the algorithm which applies not only to matching problems, but also to the general setting. © 2001 John Wiley & Sons, Inc.

Keywords: capacitated matching problems; network flows; balanced-flow networks; skew-symmetric graphs; antisymmetrical digraphs

27. FRACTIONAL BALANCED CIRCULATIONS

The present paper continues our study of balanced network flows and general matching problems [6–9]. Beside the techniques presented in these papers, there is a further strategy to obtain maximum balanced flows. This approach is due to Anstee [2] and results in state-of-the-art algorithms for capacitated matching problems.

In contrast to the balanced augmentation algorithm and the scaling methods, it does not maintain balanced flows for each augmentation. The idea is to compute an optimum fractional solution first and to turn it into an integral flow afterward.

We neither repeat any definition nor the pseudocode formalism. The reader is asked to consult parts (I) and (II) in case of confusion. At least, Sections 2–4 are necessary for understanding the terminology used.

Before we outline the algorithm, we consider the polytope $\mathscr{F}(N)$ of fractional balanced circulations on the balanced-flow network N. We will consider the polytopes associated with general matching problems and balanced network-flow problems in a forthcoming paper [10]. The polytope $\mathscr{F}(N)$ is described by the constraints

$$\begin{array}{llll} (1a) & lower(a) \leq f(a) & \text{for all arcs } a \in A(N), \\ (1b) & f(a) \leq cap(a) & \text{for all arcs } a \in A(N), \\ (2) & f(a) = f(a') & \text{for all arcs } a \in A(N), \\ (3) & e(v) = 0 & \text{for all nodes } v \in V(N). \end{array}$$

Let f be a fractional balanced flow on a balanced-flow network N. In what follows, we will search modified residual networks of N, namely, the **fractional part** residual network $N_{frac}(f)$ and the **integral part** residual network $N_{int}(f)$. These networks are defined by

$$fraccap(a) := \left\{ \begin{array}{ll} rescap(a) \bmod 2, & \text{if } a \text{ is a loop} \\ rescap(a) \bmod 1, & \text{if } a \text{ is not a loop} \end{array} \right\}$$

and $intcap(a) := rescap(a) - fraccap(a)$. Note that both networks are balanced again. An arc a with $fraccap(a) > 0$ is called **fractional.** As it turns out, in all interesting cases, we have that $2f$ is integral. We then speak of a **half-integral** flow. In extension of our previous definition, we call a path **valid** in $N(f)$ iff it is valid in $N_{int}(f)$.

An arc $a \in A(N)$ is called **free** iff $rescap(a)$ and also $rescap(\bar{a})$ are strictly positive. In particular, any fractional arc is free. A **free path** is a path that entirely consists of free arcs. To decide whether a fractional balanced circulation corresponds to a vertex of $\mathscr{F}(N)$, the investigation of free cycles turns out to be crucial.

We call a cycle in N **odd** iff it is simple and contains arcs a and $\bar{a}'$ always pairwise [cycles $Q = a \circ \bar{a}'$, where $a \in A(N)$ is a loop are included]. Note that an odd-cycle Q can be written as $Q = q \circ \bar{q}'$, where q is a strictly simple vv'-path and $v \in Q$ is arbitrary. Note that q has an odd length since N is bipartite. Hence, if we have $N = N_{\mathscr{M}}$ for some subgraph network $\mathscr{M}$, then q corresponds to an odd-length cycle of $G(\mathscr{M})$.

In [6], we introduced the notion of an elementary flow f_p associated with some path p. For reasons of brevity, we will write $\chi(p) := f_p + f_{p'}$. Note that $\chi(p) = \chi(p')$ and that $\chi(p) \equiv 0$ if p is an odd cycle.

Theorem 27.1. *Let f be a fractional balanced flow on a balanced-flow network N. Then, f is a vertex of the polytope $\mathscr{F}(N)$ iff every free cycle in $N(f)$ is odd.*

Received May 1998; accepted December 2000
Correspondence to: C. Fremuth-Paeger; e-mail: christian.fremuth@math.uni-augsburg.de
The results of this paper form part of the first author's doctoral thesis which was written under the supervision of the second author.

Proof. (→) Let f be a fractional balanced circulation on N. Assume the existence of a free cycle r in $N(f)$ which contains at least one arc a such that $\bar{a}'$ is not on r. Without loss of generality, let r be simple.

Put $\epsilon := \frac{1}{2}\min\{rescap(r), rescap(\bar{r})\}$. Then, $f^+ :\equiv f + \epsilon\chi(r)$ and $f^- :\equiv f - \epsilon\chi(r))$ are different fractional balanced circulations and f is a proper convex combination of f^+ and f^-, that is, f is not a vertex of $\mathscr{F}(N)$.

(←) Suppose that f is not a vertex of $\mathscr{F}(N)$. Then, we have $f = \lambda f_1 + (1-\lambda) f_2$, where $0 < \lambda < 1$, for different fractional balanced circulations f_1 and f_2 on N. Denote by Δ the set of arcs $a \in A(N)$ with $f_1(a) \neq f_2(a)$ which are free in $N(f)$. By the flow-conservation condition, a node cannot be incident with exactly one arc of Δ. Hence, each $a \in \Delta$ is on a free cycle Q.

Since each free cycle is assumed to be odd, Δ splits into strictly disjoint odd cycles. Since a occurs on Q in a forward, but a' occurs in a backward direction, we would get $f(a) = f_2(a) + \lambda(f_1(a) - f_2(a))$ and $f(a') = f_2(a) - \lambda(f_1(a) - f_2(a))$. But f is fractional-balanced, a contradiction. ∎

Corollary 27.2. *Let f be a vertex of the polytope $\mathscr{F}(N)$. Then f is half-integral.*

Proof. In $N_{frac}(f)$, no node can have degree one. Hence, every fractional arc is on a cycle Q that entirely consists of fractional arcs. Since Q is free, it is odd by the last theorem. It follows that the fractional arcs form strictly disjoint odd cycles.

Let v be a node on $Q; u'$, the predecessor; and w', the successor of v on Q. If x' is another node adjacent with v, we have $fraccap(v, x') = 0$. By the flow-conservation condition, $fraccap(v, w') + fraccap(v, u') = 1$ holds. Thus, all arcs on Q have equal fractional capacity. Since $v'w$ is also on Q, we obtain $fraccap(w, v') = fraccap(v, w') = fraccap(v', w) = \frac{1}{2}$. Since cap is integral, f is half-integral. ∎

Now, let $N = N_{\mathscr{M}}$ for some subgraph network $\mathscr{M}$ with degree sequences $a = b, f$ be a vertex of $\mathscr{F}(N)$, and x, the corresponding factor of $\mathscr{M}$. Note that the return arc ts is free unless we have $val(f) = 0$. Up to the cycle (s, t, s) which may occur, the free cycles correspond to cycles in $G(\mathscr{M})$.

Let $\mathscr{F}(\mathscr{M})$ denote the set of all fractional factors of $\mathscr{M}$. Since the mapping between the fractional factors of $\mathscr{M}$ and the fractional balanced circulations on N is linear and bijective, $\mathscr{F}(\mathscr{M})$ is a polytope. Moreover, vertices of $\mathscr{F}(\mathscr{M})$ correspond to vertices of $\mathscr{F}(N)$ and vice versa. Hence, if x is a vertex of $\mathscr{F}(\mathscr{M})$, it satisfies the conditions

(v1) The factor x is half-integral.

(v2) The fractional edges of x form pairwise disjoint, simple odd-length cycles.

If we restrict attention to the cardinality matching problem, the conditions (v1) and (v2) are also sufficient for x to be a vertex. Here, in N_G, free and fractional arcs are the same. This statement is due to Balinski [4] and can be extended to the f-factor problem.

Corollary 27.3. *Let $\mathscr{M}$ be a subgraph network with $a \equiv b, c \equiv 1$, and x, a fractional factor. Then x is a vertex of the polytope $\mathscr{F}(\mathscr{M})$ iff (v1) and (v2) are satisfied.* ∎

We call a fractional balanced flow f **basic** iff f corresponds to a vertex of the polytope $\mathscr{F}(\mathscr{M})$ and call f **pseudobasic** iff f is half-integral, and the fractional arcs form pairwise disjoint odd cycles. These cycles $Q_1, Q_2, \ldots, Q_r$ are referred to as the **odd-cycle system** associated with f. Note that the notions of basic and pseudobasic flows coincide only if we are concerned with the problems mentioned in Corollary 27.3.

Lemma 27.4. *Let N be a balanced-flow network, and f, a pseudobasic maximum st-flow on N. Then, at most one node of a complementary pair is s-reachable in $N(f)$ and a node can be reached from s by strictly simple paths only.*

Proof. If a pair of complementary nodes v and v' would be s-reachable in $N(f)$ by paths p and q, then $p \circ q'$ would be augmenting in $N(f)$, a contradiction. ∎

Corollary 27.5. *Under the assumption of Lemma* 27.4, *no odd cycle is s-reachable.* ∎

There is another intuitive observation:

Lemma 27.6. *Let N be a balanced-flow network, and f, a pseudobasic balanced circulation on N with an odd-cycle system $Q_1, Q_2, \ldots, Q_r$. Let $U \subseteq V(N), U = U'$ so that no odd cycle meets the cut $[U, \bar{U}]$. Write $odd(U) := |\{i : Q_i \subseteq N[U]\}|$. Then,*

$$odd(U) \equiv f(U, \bar{U}) \ mod \ 2.$$

Proof. Let $U = W \uplus W'$. Then,

$$0 = \sum_{v \in W} e(v) = f(\bar{U}, W) - f(W, \bar{U}) + f(W', W) - f(W, W').$$

Each of the cuts $[W, W']$ and $[W', W]$ splits into pairs of complementary arcs. Let $a \in [W, W'] \cup [W', W]$. Note that $f(a) + f(a')$ contributes an even amount to $f(W', W) - f(W, W')$ if a is integral and contributes an odd amount if a is on some odd cycle. Thus, one has

$$odd(U) \equiv f(W', W) - f(W, W') \bmod 2$$

and, hence,

$$\begin{aligned} odd(U) &\equiv f(\bar{U}, W) - f(W, \bar{U}) \equiv f(\bar{U}, W) + f(W, \bar{U}) \\ &\equiv f(\bar{U}, W) + f(\bar{U}, W') \equiv f(\bar{U}, U) \\ &\equiv f(U, \bar{U}) \bmod 2. \end{aligned}$$

∎

Corollary 27.7. *The number of odd cycles in a pseudobasic circulation is even.*

Proof. Observe that $odd(U) \equiv f(U,\bar{U}) \equiv f(\bar{U},U) \equiv odd(\bar{U}) \bmod 2$. ■

28. CANCELING EVEN CYCLES

In this section, we present a technique to turn an integral nonbalanced flow into a pseudobasic or even basic fractional balanced flow. The idea has been described Anstee [2] before and can be applied to weighted matching problems also. Hence, we speak of *optimum st*-flows rather than *maximum st*-flows. Depending on the type of the matching problem considered, we use the term optimum for maximum flows, admissible flows, or even minimum-weighted flows.

It is well known (and easy to see) that every flow network with integral capacities admits an integral maximum flow. But even more, there is an integral optimum flow in the general setting! This results from the total unimodularity of adjacency matrices of digraphs, which we shall not discuss (see, e.g., [15]).

Note that all the known network-flow algorithms actually yield integral solutions (at least, the preflow-push method [13, 14] and all network simplex-based algorithms [1, 5]).

Thus, we can determine an integral optimum flow f^*, say of value v^*, instead of a balanced optimum flow. This flow does not satisfy conditions (b1) and (b3) in general, but can be turned into the flow f_0 defined by

$$f_0(a) := \frac{1}{2}(f^*(a) + f^*(a')), \tag{1}$$

which satisfies condition (b1), still has flow value $val(f_0) = v^*$, but is no longer integral. Obviously, f_0 is half-integral.

Note that f_0 is not basic in general, even if f^* was a basic solution to the ordinary network-flow problem. For instance, the graph of our running example in Figure 2 of [6] has the integral maximum flow shown in Figure 1. In this representation, the nodes s and t are omitted and only the boldface arcs carry flow (namely, 1 unit iff there is one arrowhead and 2 units iff there are two arrowheads). Of course, the arcs incident with s and t are likewise omitted, but the reader may easily reconstruct the flow values on these arcs from the figure.

This flow turns into the half-integral flow f_0 of Figure 3 which is discussed later. The reader who is familiar with the network simplex algorithm will recognize that the tree shown there becomes a strongly feasible spanning tree if 7 is chosen as the root and the arcs (t,s) and $(1',t)$ are added. We will not refer to the network simplex algorithm any more.

We can turn the optimum flow f_0 iteratively into optimum flows $f_1, f_2, \ldots, f_k$ where in every iteration the number of nonintegral arc flow values is reduced. All the intermediate flows f_i are half-integral and have value v^* again.

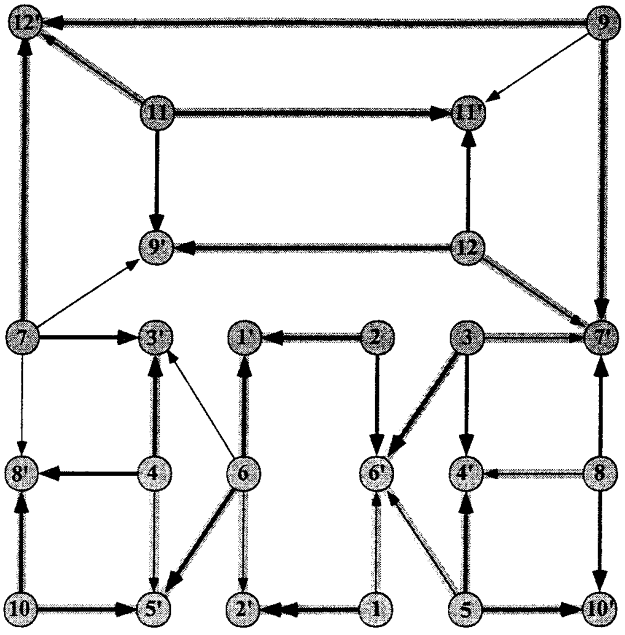

FIG. 1. A maximum integral flow and a feasible spanning tree.

In each iteration, a strictly simple cycle r is determined which entirely consists of nonintegral arcs and which corresponds to an even-length cycle of the underlying graph. Then, the present flow f_i is updated according to

$$f_{i+1} :\equiv f_i + \frac{1}{2}\chi(r), \tag{2}$$

which makes all flow values of arcs on r and r' integral. Note that $\tilde{f} :\equiv f_i - \frac{1}{2}(f_r + f_{r'})$ is also admissible. Since $f_i = \frac{1}{2}(f_{i+1} + \tilde{f})$ is optimum, $\tilde{f}$ and f_{i+1} are optimum again.

Procedure 1. Cycle-canceling algorithm.

```
class BAL_FLOW_NW;
private
 array q;

public
 procedure BAL_PUSH(a, δ):
 begin
 PUSH(a, δ);
 PUSH(a', δ);
 end

 procedure MAKE_INTEGRAL(p, u, v);
 var a, w;
 begin
 w := v;
```

```
    repeat
      $a := p[w]$;
      $BAL\_PUSH(\bar{a}, \frac{1}{2})$;
      $p[w], p[w'] := *$;
      $w := a^-$
    until $w = u$
    end

    procedure $CANCEL\_EVEN$;
    var $a, root, u, v, w, x$;

    begin
    for $v := 0$ to $n$ do $p[v], q[v] := *$ od;
    $INVESTIGATE$;
    for $root := 0$ to $n$ do
      $v := root$;
      while not ($v = root$ and $EOS(v)$) do
        $READ(v, a)$;
        $u := a^+$;
        if $fraccap(a) > 0$ and $p[v] \neq \bar{a}$ and
          $q[v] \neq \bar{a}$ and $q[u] \neq a$ then
          if $p[u] \neq *$ or $u = root$ then
(1)         $BAL\_PUSH(\bar{a}, \frac{1}{2})$;
            $MAKE\_INTEGRAL(p, u, v)$;
            $v := u$
          else
            if $p[u'] = *$ and $u' \neq root$ then
(2)           $p[u] := a$;
              $v := u$
            else
            $p[u] := a$;
            $w := u$;
            while $w \neq u'$ and $q[w] = *$ do $w := p[w]^-$
              od
            if $q[w] = *$ then
(3)             $v := u$;
                while $v \neq u'$ do
                  $a := p[v]$;
                  $q[v] := a$;
                  $q[v'] := \bar{a}'$;
                  $p[v] := *$;
                  $v := a^-$
                od
              else
(4)             $MAKE\_INTEGRAL(q, w', w)$;
                $x := u$
                while $x \neq w$ do
                  $a := p[x]$;
                  $BAL\_PUSH(\bar{a}, \frac{1}{2})$;
                  $p[x] := *$;
                  $x := a^-$
                od
                while $x \neq u'$ do
                  $a := p[x]$;
                  $BAL\_PUSH(a, \frac{1}{2})$;
                  $p[x] := *$;
                  if $x \neq u$ then $p[x'] := *$;
                  $x := a^-$
                od
                $v := u'$
              fi
            fi
          fi
        fi
      od
    od;
    $CLOSE$
    end
    end.
```

Let us describe how Procedure 1 manages this process: At every node $root \in V(N)$, a DFS is started which maintains a strictly simple path p as the **active**-path, connecting $root$ to the current active node v. In the procedure, this path is represented by the arc set

$$P := \{p[v] : v \in V(N), p[v] \neq *\},$$

where $p[v]$ denotes the arc on p with end node v. If the DFS investigates some nonintegral arc vu, four cases arise which cause the relative lengthiness of the procedure. If neither u nor u' appears on p, the active path is extended by vu [case (2)]. If u is already on p, the strictly simple cycle $r := p[u, v] \circ vu$ is found, all flow values are made integral, and the DFS backtracks up to node u [case (1)].

If u' is on the active path, the DFS backtracks up to u' and the odd cycle $r := p[u', v] \circ vu \circ \overline{p[u', v]}' \circ v'u'$ is considered. As an invariant of the algorithm,

$$Q := \{q[v] : v \in V(N), q[v] \neq *\}$$

splits into odd, mutually strictly disjoint cycles. If r is disjoint to Q, its arcs are added to Q [case (3)]. Otherwise, r and some adjacent cycle q of Q are reassembled to a complementary pair of cycles. In that case, all flow values of q and r are made integral and the arcs of q are deleted from Q [case (4)].

Note that there are no ordinary backtracking operations: If the current node v and $root$ differ, there is always a nonintegral arc vu which has not been investigated before since the number of half-integral successors in $N(f_0)$ is even. Hence, all nonintegral arcs are in P at some stage and are integral or part of Q in the end.

Finally, an edge can be investigated at most three times during the course of $CANCEL\_EVEN$, namely, while entering P, while moving from P to Q, and while leaving Q. We obtain that $CANCEL\_EVEN$ runs in $O(m)$ steps. By the preceding discussion, we obtain the following result:

Theorem 28.1. *Let f be a half-integral optimum flow on the balanced-flow network N. Then, $CANCEL\_EVEN$ updates f to a pseudobasic optimum flow on N. The associated odd-cycle system is represented by the q-labels.*

Consider the half-integral matching x of our running example $\mathscr{M}$ of Figure 2 in [6] which is shown in Figure 3. In this figure, a broken line e denotes $x(e) = \frac{1}{2}$. If f is the half-integral balanced flow corresponding to x, the execution of $N_{\mathscr{M}}.CANCEL\_EVEN$ behaves as follows:

The first DFS is rooted at 1 and yields $p[2'] := (1, 2')$ and $p[6] := (2', 6)$. In the next investigation step, $(6, 1')$ is found, which leads to an odd-length cycle of $G(\mathscr{M})$. Hence, we put $p[6], p[2'] := *$ and $q[1] := (6', 1), q[6'] := (2, 6'), q[2] := (1', 2), q[1'] := (6, 1'), q[6] := (2', 6)$, and $q[2'] := (1, 2')$. The DFS backtracks up to node 1 and then halts.

The next DFS affecting at least one arc is rooted at 3 and yields $p[6'] := (3, 6'), p[5] := (6', 5), p[4'] := (5, 4'), p[8] := (4', 8)$, and $p[7'] := (8, 7')$. Then, $(7', 3)$ is investigated, resulting in an even-length cycle of $G(\mathscr{M})$. Then all flow values on $(3, 6', 5, 4', 8, 7', 3)$ and its complementary cycle are made integral. The DFS backtracks up to node 3 and then halts.

There is one further nontrivial DFS, rooted at 7. We obtain $p[9'] := (7, 9')$ and $p[11] := (9', 11)$. Next, the loop $(11, 11')$ is investigated and $q[11] := (11', 11), q[11'] := (11, 11')$ are set. The DFS continues at 11 obtaining $p[12'] := (11, 12')$ and the cycle $(7, 9', 11, 12', 7)$ which gets integral flow values. Since there are no unconsidered fractional arcs remaining, the algorithm terminates. The final flow is not integral yet, but corresponds to the half-integral matching of Figure 4. The flow shown in Figure 4 is indeed basic. Note that a half-integral balanced flow obtained by Procedure 1 is not basic in general but merely pseudobasic. This is sufficient for our algorithmic purposes.

By a similar technique, one can determine a pseudobasic solution from an arbitrary fractional balanced flow. To this purpose, one must search any free arc, add an explicit backtracking rule, and determine the residual capacities of the cycles computed in case (1) of the algorithm. When such a cycle p is found, one can augment along p and the complementary cycle p', so that at least one arc on p is not free any longer.

However, the other arcs on p may still be free. Hence, one cannot continue the search by the strategy of Procedure 1. If one starts a new search from scratch, the resulting time bound is $O(m^2)$. Derigs [5] reported the same complexity bound. We do not know if an improvement is possible.

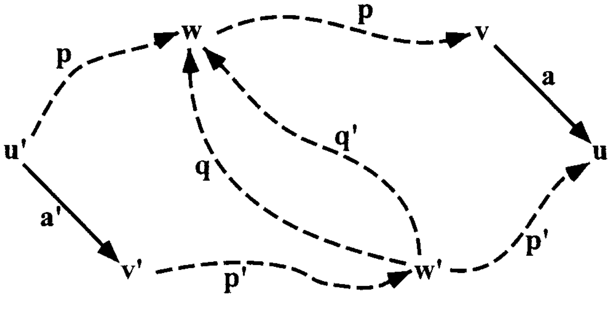

FIG. 2. Case (4) of Procedure 1.

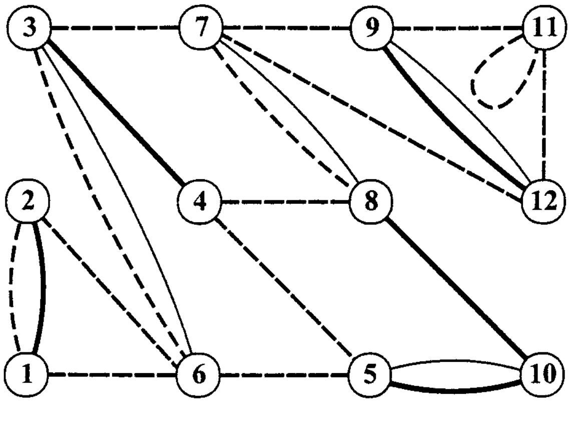

FIG. 3. A half-integral factor of the graph of Figure 2 in [6].

For the moment, let f_0 be a pseudobasic solution and $Q_1, Q_2, \ldots, Q_r$ be the odd-cycle system associated with f_0 and assume that $Q_i = q_i \circ \bar{q}_i'$. Then, f_0 can be turned into an integral flow $f^* := f_r$ by successive application of the rule

$$f_i :\equiv f_{i-1} + \frac{1}{2}(f_{q_i} - f_{q_i'}), \qquad (3)$$

which is not balanced any longer. This is the inversion of Rule (1). Note that f^* can be considered as a solution of a bipartite matching problem where all nodes and arcs appear twice.

29. CANCELING ODD CYCLES

We now return to the description of Anstee's original algorithm which determines a maximum balanced flow from any maximum flow and consists of two stages. The first stage is just a call of Procedure 1 which has already been discussed. After that call, the current odd-cycle system $Q_1, Q_2, \ldots, Q_r$ is encoded into the q-labels. We assume that $Q_i = q_i \circ \bar{q}_i'$ for $i = 1, 2, \ldots, r$.

During the second stage ($k \leq i < k + l$), we determine valid paths p in the integral part residual network $N_{int}(f_i)$

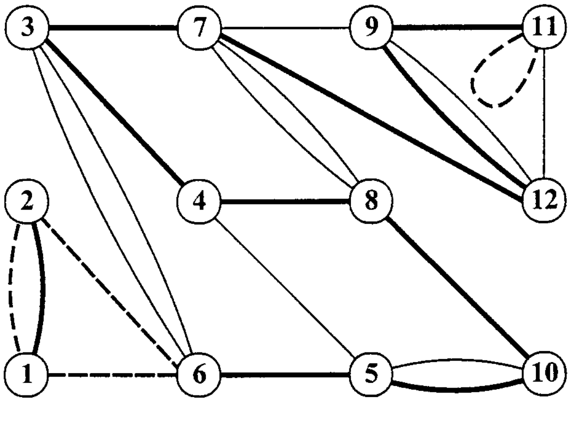

FIG. 4. A half-integral factor obtained by Anstee's algorithm.

which connect two different odd sets of $N(f_i)$, say Q_1 and Q_2. Let $u \in Q_1$ and $v \in Q_2$ be the start node and the end node of p, respectively. Without loss of generality, we can assume that q_1 starts at node u' and ends at node u and that q_2 starts at v and ends at v'. The flow f_i is changed into the flow

$$f_{i+1} :\equiv f_i + \frac{1}{2}(\chi(q_1) + \chi(q_2)) + \chi(p), \tag{4}$$

which turns all flow values of arcs on q_1 and q_2 integral. Since no further fractional arcs are affected, the odd-cycle system is updated by deleting Q_1 and Q_2. The iterated flow f_{i+1} is maximum again.

In our running example, one such iteration step is available, according to the path $p := (6, 3', 7, 9', 11)$ which is valid in $N^{int}_{\mathcal{M}}(f)$, where f is the half-integral balanced flow resulting from the first stage. The corresponding odd-cycle system is given by $q_1 = (1, 2', 6, 1')$ and $q_2 = (11, 11')$. If f is updated as described, one obtains the balanced flow corresponding to the factor of Figure 4 in [6].

Before we study the general situation, let us consider a network $N_{\mathcal{M}}$ which corresponds to some subgraph network. Then, at the end of the cycle-canceling process, the remaining odd cycles are the nuclei of $N_{\mathcal{M}}$. Furthermore, from f_{k+l}, a maximum balanced flow f^*_{bal} can be obtained by a simple operation:

Let f be pseudobasic, $Q_1, Q_2, \dots, Q_r$ be the odd cycles of $N_{\mathcal{M}}(f), Q_i := q_i \circ \bar{q}'_i$, and q_i start at some outer node v_i. Furthermore, let $\tilde{q}_i := sv_i \circ q_i \circ v'_i t$. Then, an **integral projection** of f is defined by

$$int(f) :\equiv f - \frac{1}{2}\sum_{i=1}^{r} \chi(\tilde{q}_i). \tag{5}$$

Note that this definition depends on the choice of $v_1, v_2, \dots, v_r$ unless we choose a certain node ordering. Actually, the operation can be applied to every pseudobasic flow on $N_{\mathcal{M}}$, but does not yield a maximum balanced flow in the general situation.

Theorem 29.1. *Let f be a pseudobasic maximum st-flow on $N_{\mathcal{M}}$. Then, the integral projection $int(f)$ is maximum balanced iff none of the odd cycles of $N_{\mathcal{M}}(f)$ are joined by a valid path.*

Proof. Both directions follow by contraposition: If two odd cycles are joined by a valid path, f can be transformed into another pseudobasic flow $\tilde{f}$ using Eq. (4). Obviously, $val(int(\tilde{f})) = val(int(f)) + 2$ holds.

In the converse direction, assume that $int(f)$ is not maximum balanced and let p be a valid augmenting path in $N_{\mathcal{M}}(int(f))$. Let $[S, T]$ be the st-cut corresponding to the maximum flow f. Since fractional arcs are free, none of the Q_i's cross the cut $[S, T]$. So, the arcs (s, v_i) are the only unsaturated arcs from S to T in $N_{\mathcal{M}}(int(f))$. Hence, p crosses the cut $[S, T]$ exactly once and meets the Q_i's at least once.

We now consider the complementary path p' whose last arc is $v'_i t$. To cross the cut, p' must meet another odd cycle Q_j since we have $rescap(s, v_i) = 1$. But then there is a part of p which starts at some odd cycle Q_i and ends at another odd cycle $Q_j, i \neq j$. Even if p meets some further odd cycle intermediately, we can find a portion which contradicts the choice of f. ▪

Corollary 29.2. *The balanced flow f^*_{bal} obtained by Anstee's algorithm is maximum balanced.* ▪

Theorem 29.3. *Let f be a pseudobasic maximum flow on $N_{\mathcal{M}}$ with $int(f)$ maximum balanced. Then, the core of $N_{\mathcal{M}}$ splits into the odd cycles $Q_1, Q_2, \dots, Q_r$.*

Proof. By the corresponding iteration of Eq. (5), all nodes on Q_i become strictly reachable in $N_{\mathcal{M}}(int(f))$, that is, $Q_1, Q_2, \dots, Q_r \subseteq \mathscr{C}$.

Suppose the existence of a node $v \in \mathscr{C}$ such that $v \notin Q_1, Q_2, \dots, Q_r$. We can assume that v is not s-reachable in $N(f)$. If q is a directed sv'-path in $N(f)$, it must be strictly simple by the above argument. Even more, q cannot traverse one of the Q_i's and, hence, is valid in $N_{\mathcal{M}}(int(f))$.

Every valid sv-path in $N_{\mathcal{M}}(int(f))$ must start with an arc sv_i and cannot traverse $Q_j, j \neq i$. It follows that $p[v_i, v]$ is a directed path in $N(f)$ so that p and q must be disjoint. But at least the node $base(v) = base(v')$ would be on each of p and q. Hence, $base(v) = s$ would result, contradicting the maximality of $int(f)$. ▪

We have mentioned Anstee's algorithm at this point for the following reason: The search for valid paths in $N^{int}_{\mathcal{M}}(f_i)$ is a process very similar to the process of augmenting a balanced flow on $N_{\mathcal{M}}$. Without loss of efficiency, one can exchange the second stage of Anstee's algorithm by a balanced augmentation procedure which is applied to $N_{\mathcal{M}}$ and the balanced flow $int(f_k)$. More explicitly, we compute the integral projection of f_k by Procedure 2. This algorithm uses the path labels q which have been computed by Procedure 1 before.

Procedure 2: Odd cycle canceling.

```
class $BFN\_BY\_MATCHING(BAL\_FLOW\_NW)$;
public
 procedure CANCEL_ODD;
 var $a, u, v$;
 begin
 for $v := 0$ to n do
  if $q[v] \neq *$ and $OUTER(v)$ then
   $MAKE\_INTEGRAL(q, v, v')$;
   $BAL\_PUSH(vs, 1)$;
  fi;
 od
 end;
end.
```

Observation 29.4. *If the flow f encoded into the object $N_{\mathcal{M}}$ is pseudobasic and the odd cycles are encoded into the labels q, then $N_{\mathcal{M}}$.CANCEL_ODD determines an integral projection of f.* ▪

It is evident that *CANCEL_ODD* runs in time $O(n)$. Since the number of odd cycles is bounded by n, and the value of a maximum balanced flow is bounded by the value of a maximum fractional balanced flow, f_{bal} can be augmented at most n times. We already know how to perform an augmentation step in approximately $O(m)$ steps. So, we can derive a maximum balanced flow from a maximum integral flow with complexity $O(nm)$.

So far, we have studied matching problems, as proposed by Anstee [2]. If we consider arbitrary balanced-flow networks, the concept of an integral projection does not apply. We add the return arc to the flow network which may form an additional odd cycle. We can apply the Decomposition Theorem 4.1 of [6] to pseudobasic circulations as follows:

Theorem 29.5. *Let N be a balanced-flow network which admits a balanced circulation. Let f be a pseudobasic circulation on the balanced-flow network N, and Q, an odd cycle in $N(f)$. Then, Q is reached from another odd cycle by a valid path. In other words, f can be updated according to Eq.* (4).

Proof. Because of the special structure of the odd cycles $Q_i = q_i \circ \bar{q}'_i$, we can find nodes $v_i \in Q$ which are reached by a backward arc and left by a forward arc on Q_i. Put $f :\equiv f - \frac{1}{2}\sum_{i=1}^{k}\chi(q_i)$ so that $e(v_i) = 1$ and $e(v'_i) = -1$ hold for $i = 1, 2, \ldots, k$. The remaining nodes satisfy the flow-conservation condition.

Since the number k of odd cycles is even, we may increase $f(v_{2i-1}, v'_{2i})$ and $f(v_{2i}, v'_{2i-1})$ for each $i = 1, 2, \ldots, k/2$ by a single unit. We obtain a balanced circulation $\dot{f}$ on a (possibly) modified network $\dot{N}$. If g is a balanced circulation on the original network N, it may be extended to a circulation on $\dot{N}$ in the obvious way. Now, we can decompose

$$\dot{g} - \dot{f} \equiv \sum_{i=1}^{l} \chi(p_i),$$

where the p_i's are valid cycles with respect to $\dot{N}(\dot{f})$. Without loss of generality, we may assume that the arc (v'_{2i-1}, v_{2i}) is traversed by the cycle p_i.

Since p_i must eventually return from Q_{2i} to Q_{2i-1}, we can find a portion of p_i which connects Q_{2i} to some other odd set, so that there are no intermediate odd sets. It turns out that this portion is valid in the original setting of $N(f)$ also. In the same way, one can find a portion of p'_i which connects Q_{2i-1} to some other odd cycle in $N(f)$. ▪

The general odd-cycle canceling procedure is more expensive than is computing the integral projection and may require $O(nm)$ steps all together. An explicit algorithm may be easily derived from our previous pseudocode. An update of the flow f according to Eq. (4) may decrease the flow value by at most 2. As before, we need $O(n)$ balanced augmentation steps to obtain a maximum balanced flow.

It turns out that the final balanced augmentation steps can be avoided if we circumvent the return arc in the odd-cycle canceling process as long as possible. If one chooses g maximum balanced in the proof of Theorem 29.5, the following results:

Corollary 29.6. *Let ν be the maximum value of a balanced st-flow, and f, a nonintegral pseudobasic balanced st-flow with val$(f) = \nu$. Then, f can be updated according to Eq.* (4) *without affecting the flow value.*

If we apply odd-cycle canceling to a pseudobasic minimum-cost circulation, we would like to obtain an optimum balanced flow finally or at least a (ν)-optimum balanced *st*-flow which applies to the SAP algorithm proposed by Goldberg and Karzanov [12].

Anstee [3] showed that the cycle-canceling algorithm can be used as a starting heuristic for the (dual feasible) primal-dual algorithm which was devised by Pulleyblank [16] for the *b*-matching problem. Part VII [11] of this series will establish a primal-dual algorithm for min-cost balanced *st*-flows and a polynomial refinement which finds min-cost balanced circulations.

In fact, this polynomial algorithm is started with a min-cost pseudobasic circulation. We will show that this pseudobasic solution can be considered a (ν)-optimum balanced *st*-flow in a modified network to which the PD algorithm but also the SAP algorithm applies. In this procedure, the canceling of odd cycles is just as simple as in the case of matching problems and does not require a flow decomposition.

REFERENCES

[1] R. Ahuja, T. Magnanti, and J. Orlin, Network flows, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] R. Anstee, An algorithmic proof of Tutte's f-factor theorem, J Alg 6 (1985), 112–131.

[3] R. Anstee, A polynomial algorithm for *b*-matchings: An alternative approach, Info Process Lett 24 (1987), 153–157.

[4] M. Balinski, Establishing the matching polytope, J Combin Theory (B) 13 (1972), 1–13.

[5] U. Derigs, Programming in networks and graphs, Springer, Heidelberg, 1988.

[6] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (I): A unifying framework for design and analysis of matching algorithms, Networks 33 (1999), 1–28.

[7] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (II): Simple augmentation algorithms, Networks 33 (1999), 29–41.

[8] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (III): Strongly polynomial augmentation algorithms, Networks 33 (1999), 43–56.

[9] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (IV): Duality and structure theory, Networks 37 (2001), 194–201.

[10] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (VI): Polyhedral descriptions, Networks 37 (2001), 210-218.

[11] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (VII): Primal-dual algorithms, in preparation.

[12] A. Goldberg and A. Karzanov, Path problems in skew-symmetric graphs, Combinatorica 16 (1996), 353–382.

[13] A. Goldberg and R. Tarjan, A new approach to the maximum flow problem, J ACM 35 (1988), 921–940.

[14] D. Jungnickel, Graphs, networks and algorithms, Springer, Heidelberg, Germany, 1999.

[15] C. Papadimitriou and K. Steiglitz, Combinatorial optimization, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[16] W. Pulleyblank, Faces of matching polyhedra, PhD Thesis, University of Waterloo, 1973.