Balanced Network Flows. VII. Primal-Dual Algorithms

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

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

We discuss an adaptation of the famous primal-dual 1-matching algorithm to balanced network flows which can be viewed as a network flow description of capacitated matching problems. This method is endowed with a sophisticated start-up procedure which eventually makes the algorithm strongly polynomial. We apply the primal-dual algorithm to the shortest valid path problem with arbitrary arc lengths, and so end up with a new complexity bound for this problem. © 2002 John Wiley & Sons, Inc.

Keywords: capacitated matching problems; *b*-matching problems; network flows; primal-dual algorithm; shortest path problems

1. PRELIMINARIES

In this paper, we discuss an adaptation of the famous primal-dual 1-matching algorithm of Edmonds [3] to balanced network flows. The reader who is familiar with matching theory will easily recognize the specialization to Edmonds' algorithm.

Before we start the description of the algorithm, we summarize the polyhedral and duality results given in Part (VI) of this series [6]. We do not repeat all notation and theory which would be helpful for a complete understanding. The reader is asked to consult Part (I) [4] for an introduction to the general framework. Parts (II)–(VI) or other sources about matching problems are not needed.

Received November 11, 2000; accepted August 1, 2001

Correspondence to: C. Fremuth-Paeger; e mail: christian.fremuth@math.uni-augsburg.de

DOI 10.1002/net.10002

In Part (VI), we defined an **odd skew cut** (A_1, A_2) of a balanced flow network as follows: There is a bipartition of the node set $V(N) = U \uplus \bar{U}$ such that U (the **interior**) and $\bar{U}$ (the **exterior**) are symmetric, $\gamma_N(U, \bar{U}) = A_1 \uplus A_2$, and

$$scap(A_1, A_2) := ucap(A_1) - lcap(A_2)$$

is odd. The set of odd skew cuts is denoted by $\mathcal{O}(N)$. We proved that

minimize
$$\sum_{a \in A(N)} c(a) f(a)$$
subject to

$$\begin{array}{lll}
(p1a) & f(a) \geq lcap(a) & \forall a \in A(N) \\
(p1b) & f(a) \leq ucap(a) & \forall a \in A(N) \\
(p2) & f(a) = f(a') & \forall a \in A(N) \\
(p3) & e(v) = 0 & \forall v \in V(N) \\
(p4) & f(A_2) - f(A_1) \geq scap(A_1, A_2) + 1 & \forall (A_1, A_2) \in \mathcal{O}(N)
\end{array}$$

is a complete (but redundant) LP description for the problem of finding a min-cost balanced circulation. As an example, consider the network in Figure 1 with 0–1 capacities and the cost labels shown in this figure. This network admits a fractional balanced circulation $f_1 = \frac{1}{2}\chi^p, p = (1, 2, 4', 3, 4, 2', 1)$. Here, $\chi^p := f^p + f^{p'}$ and f^p is the **elementary flow** or **incidence vector** of the path p, defined by

$$f_p(a) := \left\{ \begin{array}{ll} +1 & \text{if } a \text{ is a forward arc,} \\ -1 & \text{if } a \text{ is a backward arc,} \\ 0 & \text{otherwise.} \end{array} \right\}$$

A half-integral balanced circulation is said to be **pseudobasic** if the fractional arcs form disjoint, self-symmetric cycles. These cycles form the **odd-cycles system.** The circulation f_1 in our example is pseudobasic with odd cycles $(1, 2, 1', 2', 1)$ and $(3, 4, 3', 4', 3)$. We mention that f_1 is a vertex of the **fractional balanced circulation polytope** which is constituted by the constraints

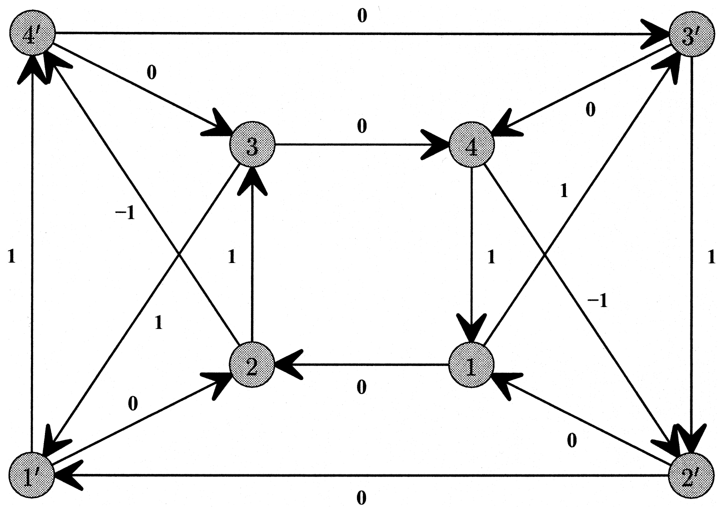

FIG. 1. A balanced flow network.

(p1a), (p1b), (p2), and (p3). On the other hand, the solution f_1 violates the skew-cut constraint

$$f(2,4') - f(1',4') - f(2,3) - f(1,3') \leq 0$$

and, hence, is not a convex combination of balanced circulations.

We do not need an explicit LP-dual problem here. Instead of this, we define a **symmetric dual solution** as a pair (π, ϕ), where $\pi \in \mathbf{R}^{V(N)}, \phi \in \mathbf{R}^{\mathcal{O}(N)}, \phi \geq 0, \pi(v) = -\pi(v)$ for every node $v \in V(N)$, and $\phi(A_1, A_2) = \phi(A_1', A_2')$ for every skew cut $(A_1, A_2) \in \mathcal{O}(N)$.

Note that the LP-dual would not include any symmetry constraints. But adding such constraints preserves the optimal dual objective value which has an important benefit for our later algorithms: The reduced-length labels which we will define next are symmetric.

34. COMPLEMENTARITY

The **incidence vector** χ^{A_1,A_2} of a skew cut (A_1, A_2) is defined by

$$\chi^{A_1,A_2}(a) := \begin{Bmatrix} +1 & \text{if } a \in A_1, \\ -1 & \text{if } a \in A_2, \\ 0 & \text{otherwise.} \end{Bmatrix}$$

In accordance with [7], and as an extension of the reduced-length labels for ordinary min-cost flow problems, we call

$$c_\pi^\phi(a) := c(a) + \pi(a^-) - \pi(a^+) + \sum_{(A_1,A_2)\in\mathcal{O}(N)} \chi^{A_1,A_2}(a)\phi(A_1, A_2)$$

the **modified length** of the arc a. Both incidence vectors and modified-length labels are extended to the residual network by taking $\chi^{A_1,A_2}(\bar{a}) = -\chi^{A_1,A_2}(a)$. One obtains the important **complementary slackness** optimality criterion [6]:

Theorem 34.1. *Let f be a balanced circulation on a balanced flow network N. Then, f is optimal iff there is a dual (π, ϕ) so that*

$(cs1)$	$c_\pi^\phi(a) \geq 0,$	*if* $rescap_f(a) > 0,$
$(cs2)$	$\phi(A_1, A_2) = 0,$	*if* $(A_1, A_2) \in \mathcal{O}(N)$ *is not tight.*

A balanced pseudoflow f and a symmetric dual (π, ϕ) which satisfy the complementary slackness conditions (cs1) and (cs2) are called a **complementary pair.** Let

$$ucap_0(a) := \begin{Bmatrix} ucap(a), & \text{if } c_\phi^\pi(a) \leq 0 \\ lcap(a), & \text{if } c_\phi^\pi(a) > 0 \end{Bmatrix},$$

$$lcap_0(a) := \begin{Bmatrix} lcap(a), & \text{if } c_\phi^\pi(a) \geq 0 \\ ucap(a), & \text{if } c_\phi^\pi(a) < 0 \end{Bmatrix}.$$

By N_ϕ^π, we denote the balanced flow network which is formed by $V(N), A(N)$ and the capacity labels $cap_0, lower_0$. This network is called the **admissible graph** with respect to (π, ϕ). If there is no confusion about the dual solution, we write N_0 instead of N_ϕ^π. Note that $N_0(f)$ contains only arcs which have zero modified length.

Corollary 34.2. *Let f and (π, ϕ) be optimal. Then, f is feasible for N_0.*

This statement is "one way" and does not indicate how admissible graphs can help in finding primal optima. In fact, our introductory example admits a circulation $f_2 = \chi^q, q = (1, 2, 3, 4, 1)$, which is not optimal, but is feasible for $N_\phi^\pi, \pi \equiv 0, \phi(\{24'\}, \{1'4', 23, 13'\}) = 1$.

But suppose that one has a complementary pair f and (π, ϕ), where f is a balanced pseudoflow rather than a circulation. Then, under certain circumstances, one can augment on a pair of valid paths in $N_0(f)$, decreasing the node imbalances while maintaining the compatibility: An augmenting path is called a **traversal path** if it is valid with respect to $N_0(f)$ and if it satisfies $\chi^p \chi^{A_1,A_2} = 0$ for every tight skew cut $(A_1, A_2) \in \mathcal{O}(N)$.

Lemma 34.3. *Let f and (π, ϕ) be a complementary pair, and p, a traversal path. Then, the pair $g :\equiv f + \chi^p$ and (π, ϕ) is also complementary.*

Proof. Since g is feasible for N_0, it satisfies the slackness constraints (cs1). Let $(A_1, A_2) \in \mathcal{O}(N)$ be a skew cut. Then, we have

$$\begin{aligned} g(A_1) - g(A_2) &= f(A_1) + \chi^p(A_1) - f(A_2) - \chi^p(A_2) \\ &= f(A_1) - f(A_2) + \chi^p \chi^{A_1,A_2}. \end{aligned}$$

If (A_1, A_2) is tight with respect to f, then, by hypothesis, $\chi^p \chi^{A_1,A_2} = 0$, and, thus, it is also tight with respect to g. But if (A_1, A_2) is not tight with respect to g, then $\phi(A_1, A_2) = 0$ holds by the complementarity of f and (π, ϕ). ∎

A balanced flow which is optimal among all balanced flows with equal value is called **extreme.** The next statement shows the strong relationship between PD algorithms and shortest path algorithms.

Observation 34.4. *Let f be a balanced st-flow, and (π, ϕ), a symmetric dual which form a complementary pair. Then, f is extreme. If p is a traversal st-path, then p is a shortest valid st-path and $c(p) = -2\pi(s)$.*

Proof. Put $lcap(ts) = f(ts) = ucap(ts) := val(f)$. Then, f and (π, ϕ) are still complementary. Hence, f is optimal for the modified problem by Theorem 34.1, that is, f is extreme. If one applies Lemma 34.3 and repeats the argument for $g := f + \chi^p$, one can observe that this flow is also extreme. But, then, p must be a shortest valid st-path.

For every st-path in $N_0(f)$, we have $c^\pi_\phi(p) = 0$. Note that in $c^\pi_\phi(p)$ all node potentials π but $\pi(s)$ and $\pi(t)$ cancel out. Since p is a traversal path, the terms $\chi^{A_1,A_2}(a)\phi(A_1, A_2)$ in the definition of c^π_ϕ also vanish, that is,

$$c^\pi_\phi(p) = c(p) + \pi(s) - \pi(t) = c(p) + 2\pi(s) = 0. \qquad ∎$$

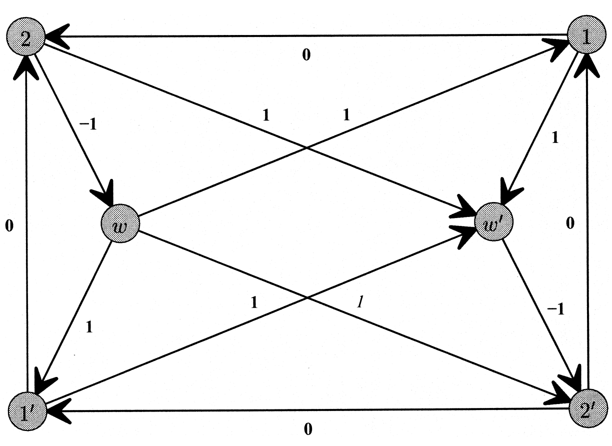

FIG. 2. Shrinking a fragment.

We mention the possibility of algorithms which maintain a feasible balanced circulation and a symmetric dual solution and which reach compatibility eventually. Such algorithms are called **primal.** However, the performance of primal algorithms is inferior even in the 1-matching case and, in the general situation, the existence of a polynomial primal algorithm is an open problem.

In the next two sections, we develop the algorithmic concepts which allow construction of traversal st-paths. Since primal, shortest path, and primal-dual (PD) algorithms all augment along traversal paths, these concepts are somewhat more general than is the PD approach.

35. SHRINKING FAMILIES

As in [7], a **fragment** of a balanced network N is a pair (U, a), where U is a self-complementary node set, called the **interior,** and a is an arc in $N[\bar{U}, U]$, called the **prop.** Blossoms and nuclei together with their props are reasonable examples of fragments.

Shrinking a fragment (U, a) of a balanced network N does the following: All interior nodes and arcs are deleted from N. Instead of these, a new pair w, w' of nodes is introduced. The arcs incident with U are redirected as follows:

- The new end node of a is w.
- The new start node of a' is w'.
- The start node of other arcs in $N[U, \bar{U}]$ is w.
- The end node of other arcs in $N[\bar{U}, U]$ is w'.

This modification preserves the skew-symmetry of the network. In Figure 2, the fragment $(\{3, 4, 3', 4'\}, 24')$ of our introductory example is shrunk.

A **shrinking family in** N is a set $\mathcal{S}$ of fragments so that their interiors form a nested family. By this definition, the cardinality of a shrinking family is bounded by $O(n)$. This is important since, in practical matching algorithms, all nonzero dual variables can be identified with some member of the shrinking family.

The network which results from shrinking all maximal fragments of a family $\mathcal{S}$ is called the **surface graph.** Note that these shrinking operations commute. Hence, we can write $N_{\mathcal{S}}$ irrespective of the special order. If no confusion about $\mathcal{S}$ is possible, we write $\bar{N}$ instead of $N_{\mathcal{S}}$. Our main interest is in the network $\overline{N_0(f)}$.

Lemma 35.1. *Let U denote some blossom or nucleus of the network $N_0(f)$. Then, there is a tight skew cut with interior U.*

Proof. For the arc $\hat{a} = prop(U)'$, we have $rescap_f(\hat{a}) = rescap_f(\hat{a}') = 1$. Let $a \in N[U, \bar{U}] \backslash \{\hat{a}\}$. Note that $rescap_f(a) = 0$ or $rescap_f(\bar{a}) = 0$, since, otherwise, $c^\phi_\pi(a) = 0$ would hold by the compatibility of f and (π, ϕ). But, then, a would be bicursal in $N_0(f)$.

ID Line: November 16, 2001, 3:41pm W-Networks (39:1) 8u3b1002

Hence, we can partition $N[U,\bar{U}] = A_1 \uplus A_2 \uplus \{\hat{a}\}$ so that

$$A_1 := \{a \in N[U,\bar{U}] : rescap_f(a) = 0, a \neq \hat{a}\},$$

$$A_2 := \{a \in N[U,\bar{U}] : rescap_f(a) > 0, a \neq \hat{a}\},$$

and $f(A_1) = ucap(A_1), f(A_2) = lcap(A_2)$. If $\hat{a}$ is a forward arc, it is added to A_1. Otherwise, $\bar{\hat{a}}$ is added to A_2. This yields the desired tight skew cut.

We finally need to show that $ucap(A_1) - lcap(A_2)$ is odd. But this follows directly from Corollary 30.1 in [6] which says that $f(A_1) - f(A_2)$ is even. ■

Let us extend this construction principle to arbitrary fragments (U,a) in the obvious way and call the resulting cut the **skew cut associated with the fragment** (U,a). A triple $(\pi,\phi,\mathscr{S})$ is called **strongly dual feasible** if

(d1) (π,ϕ) is a symmetric dual solution,
(d2) $\mathscr{S}$ is a shrinking family in N,
(d3) every skew cut (A_1,A_2) with $\phi(A_1,A_2) > 0$ is associated with a fragment in $\mathscr{S}$.

The PD algorithm maintains a strongly feasible dual solution. When it detects a blossom in $\overline{N_0(f)}$, then $N_0(f)$ also admits a blossom, and the corresponding fragment is added to the shrinking family $\mathscr{S}$. This effectively introduces a pair of pseudonodes w,w' and allocates the dual variable ϕ for the associated skew cut (A_1,A_2). It is convenient to extend the potential π to pseudonodes and to associate with $-\pi(w)$ the dual variable $\phi(A_1,A_2)$ and with $\pi(w')$ the variable $\phi(A_1',A_2')$.

As matching algorithms work mainly on the surface graph, it is helpful to distinguish the (pseudo) nodes which are in the surface graph from the (pseudo) nodes which are shrunk into a fragment. We call the former nodes **exterior** and the latter nodes **interior.**

Lemma 35.2. *Let v be an exterior (pseudo) node, and a, an arc with $lcap(a) < ucap(a)$. If one increases $\pi(v)$ by an amount ϵ, and decreases $\pi(v')$ by ϵ, this*

(*a*) *increases $c^\pi_\phi(a)$ by ϵ, if the start node of a in $\overline{N_0(f)}$ is v or v',*
(*b*) *decreases $c^\pi_\phi(a)$ by ϵ, if the end node of a in $\overline{N_0(f)}$ is v or v',*
(*c*) *does not change $c^\pi_\phi(a)$ otherwise.*

Proof. The statement is obvious if v is an original node. If v' is associated with a fragment rather than v, we can replace v by v' and ϵ by $-\epsilon$ to apply the subsequent arguments. Hence, we may suppose that v is associated with the fragment $(U,\hat{a})$ and the skew cut (A_1,A_2).

If a is internal or nonincident with v and v', it is clear that $c^\pi_\phi(a)$ does not change. If the start node of a is v or v', we may consider the reverse arc $\bar{a}$ instead of a.

First, let a end at v in $\overline{N_0(f)}$ so that $a = \hat{a}$. If a is a forward arc, we have $a' \in A_1$, and $\bar{a} \in A_2$ otherwise. Using the symmetries $c^\pi_\phi(a) = c^\pi_\phi(a')$ and $c^\pi_\phi(a) = -c^\pi_\phi(\bar{a})$, the assertion follows.

Finally, let a end at v' in $\overline{N_0(f)}$. We obtain $a \in A_2$ if a is forward arc and $\bar{a} \in A_1$ otherwise. Apply the above symmetries. ■

Let v^* denote the maximal fragment containing an interior node v. Computing v^* essentially is the `find` operation of a disjoint set union (DSU) data structure. In contrast to the nonweighted matching case, one needs an operation which is inverse to the shrinking of fragments and called the **expansion** of a fragment.

An adequate implementation needs $O(\log n)$ steps for `find` operations and $O(k \log n)$ for `shrink` and `expand` operations, where k denotes the number of nested fragments. Under those circumstances, the DSU data structure is not critical for the complexity of the PD algorithm.

36. STRONGLY COMPLEMENTARY PAIRS

A pseudoflow f and a triple $(\pi,\phi,\mathscr{S})$ are called **strongly complementary** if

(c1) $(\pi,\phi,\mathscr{S})$ is strongly dual feasible,
(c2) f is feasible for N_0,
(c3) the skew cuts associated with the fragments in $\mathscr{S}$ are tight,
(c4) every interior node v has flow excess equal to zero,
(c5) for every fragment (U,a) and every node $v \in U$, there is a valid path in $N_0[U]$ starting with a and ending at v.

If not stated otherwise, strong complementarity is assumed. It follows that (cs1) and (cs2) hold likewise.

For sake of simplicity, call an exterior node **reachable** if it is strongly s-reachable, that is, reachable from s on a valid path. The next two statements show why we simply can search for valid paths in the surface graph.

Observation 36.1. *A valid path in $N_0(f)$ is a traversal path if and only if every fragment $(U,a) \in \mathscr{S}$ is traversed at most once and reached by a or left by a'.*

Proof. Let (A_1,A_2) denote the skew cut associated with the fragment $(U,\hat{a}) \in \mathscr{S}$. By Property (c3), this skew cut is tight, and except for $\hat{a}$ and $\hat{a}'$, we have $f(a) = lcap(a)$ for every $a \in A_2$ and $f(a) = ucap(a)$ for every $a \in A_1$. But, then,

$$\chi^p(a)\chi^{A_1,A_2}(a) := \begin{cases} +1 & \text{if } a \in \{\hat{a},\hat{a}'\}, \\ -1 & \text{if } a \in (\overline{A_1} \cup A_2 \cup \overline{A_1'} \cup A_2')\backslash\{\hat{a},\hat{a}'\}, \\ 0 & \text{otherwise} \end{cases}$$

holds for the arcs $a \in p$. Because $rescap(\hat{a}) = 1$, only one of the arcs $\hat{a}$ and $\hat{a}'$ can occur on p. ■

Observation 36.2. *A node v is reachable in $N_0(f)$ if and only if v^* is reachable in $\overline{N_0(f)}$.*

A traversal st-path can be computed from a valid st-path in the surface graph as follows:

Add to the data structure for the shrinking family two operations `block` and `unblock`. The `block` operation effectively expands a maximal fragment but keeps the information which is necessary to shrink this fragment again. This is done later by the `unblock` operation.

Both operations may take $O(l \log n)$ time for a fragment of size l. The path-expansion procedure determines a traversal path and, hence, needs to traverse a fragment at most once. If follows that `block` and `unblock` operations require $O(n \log n)$ time altogether.

The path-expansion method consists of two procedures `traverse` and `expand`. Both methods are called with two arcs a_{in}, a_{out} of the later traversal path.

The `expand` operation fills the gap between a_{in} and a_{out} in the current surface graph and calls `traverse` for every pair a_1, a_2 of adjacent intermediary arcs.

The `traverse` operation checks if the common end node $v = a_1^+ = a_2^-$ is an original node. In that case, it fixes the predecessor arc of v on p. If v is a pseudonode, `traverse` blocks the corresponding fragment (U, a), calls `expand` with a_1 and a_2, and unblocks the fragment again. To repair Property (c5) for this fragment, `traverse` also updates the prop to $\overline{a_2}$ if $a = a_1$ and to $\overline{a_1'}$ if $a = a_2'$.

Corollary 36.3. *A traversal path can be expanded in $O(n \log n)$ time. If an st-path is expanded as described, the PD pair which results from augmentation is strongly complementary again.*

Proof. The complexity statement is obvious. The operations do not affect the dual variables and the interiors of the fragment so that (c1) still holds. The properties (c2) and (c3) are maintained by Lemma 34.3. Since s and t are exterior nodes, (c4) is also maintained. Finally, one can check that (c5) holds by the described update of the fragment props. ■

We note an application to primal algorithms. A corresponding statement can be found in [7] and for 1-matchings in [2]. We do not require Property (c1) here, but only that f is a feasible circulation on N.

Observation 36.4. *Let p be a valid cycle in $\overline{N(f)}$ which expands to the cycle q. Then, $c(q) = c_\phi^\pi(q) = c_\phi^\pi(p)$ holds.*

37. OUTLINE OF THE PD ALGORITHM

So far, we have given only a vague idea of what the PD algorithm does. We now give a high-level description of the PD method and discuss the possible implementations.

For sake of simplicity, assume that all arc-length labels are nonnegative and that we are interested in a maximum balanced st-flow where $s' = t$.

PD Method:

(1) Put $f :\equiv 0, \pi :\equiv 0, \phi :\equiv 0$.
(2) If $\overline{N_0(f)}$ admits a valid st-path, expand this path, augment f, and repeat step (2).
(3) Shrink the blossoms of $\overline{N_0(f)}$ if any exist.
(4) If possible, adjust (π, ϕ) to grow the set of s-reachable nodes in $\overline{N_0(f)}$, expand some fragments, and go to Step (2).
(5) Otherwise, conclude that f is maximum and stop.

Step (2) updates the primal solution, that is, it augments the flow. Step (4) is called the **dual update** and turns out to be the critical operation for the time complexity of the PD algorithm. Define

$$\begin{aligned}
\epsilon_1 &:= \min\{c_\pi^\phi(uv) : u \text{ is reachable}, v \text{ and } v' \text{ are not}\}\\
\epsilon_2 &:= \tfrac{1}{2}\min\{c_\pi^\phi(uv) : u \text{ and } v' \text{ are reachable}\}\\
\epsilon_3 &:= \min\{\phi(U, a) : (U, a)' \text{ is reachable}\}\\
\epsilon &:= \min\{\epsilon_1, \epsilon_2, \epsilon_3\}.
\end{aligned}$$

Due to the slackness condition (cs1), we have $\epsilon \geq 0$. If $\epsilon < \infty$, put

$$\pi(v) := \pi(v) - \epsilon, \quad \pi(v') := \pi(v') + \epsilon$$

for every s-reachable node. If $\epsilon = \epsilon_3$, some fragments are assigned zero potentials by that operation. These fragments are expanded by the algorithm either simultaneously or one by one. In the first case, we have $\epsilon > 0$.

If u and v are reachable nodes, then $c_\pi^\phi(uv)$ does not change by the dual update, and, hence:

Observation 37.1. *By a dual update, no exterior node becomes nonreachable. Even more, at least one node becomes reachable or a fragment is expanded.*

Lemma 37.2. *If $\epsilon_1 = \epsilon_2 = \infty$, then f is a maximum balanced st-flow.*

Proof. Suppose that a node v exists which is strongly s-reachable in $N(f)$, but not in $\overline{N_0(f)}$. Let u denote the predecessor of v on a valid sv-path in $N(f)$ and assume that u is reachable. If v' is also reachable, then $\epsilon_1 \leq c_\pi^\phi(uv)$. Otherwise, $\epsilon_2 \leq c_\pi^\phi(uv)$ holds.

If such a node v does not exist, then f can be augmented. But this should have been done in Step (2) before. ■

Let us call the period between two augmentations of the flow an **iteration** of the PD algorithm. Denote the time which is needed for an iteration by $\rho(n, m)$.

Lemma 37.3. *There are $O(n)$ dual updates during an iteration. More precisely, every (pseudo) node can be shrunk or expanded at most once.*

Proof. There are $O(n)$ dual updates by which nodes become reachable. During the other dual updates, at least one fragment is expanded. This fragment (U, a) is maximal, and $(U, a)'$ is reachable in the current surface graph.

ID Line: November 16, 2001, 3:41pm W-Networks (39:1) 8u3b1002

This shows that (U, a) has been shrunk before the last augmentation. But there were only $O(n)$ (pseudo) nodes of the shrinking family at the moment of the last augmentation. ∎

The various implementations of the PD algorithm differ mainly by their data structures for modified-length labels and the dual-update mechanism.

If there is no special data structure for the modified-length labels, their computation takes $O(n)$ time, but no update is necessary. This implementation is reasonable only for geometrical matching problems to save computer storage.

If the modified-length labels are stored explicitly, a dual update requires $O(m)$ steps, but a retrieval is an elementary operation. The computer code prepared by us works this way.

Theorem 37.4. *Let ν denote the value of a maximum balanced st-flow. Then, the PD algorithm runs in $O(\nu\rho(n, m))$ time and $\rho(n, m)$ is $O(nm)$.*

Proof. Suppose that the PD algorithm does a balanced network search (BNS) before the Steps (2) and (3) are performed. Then, by Lemma 37.3, only $O(n)$ BNS operations can occur, each of which needs $O(m)$ time. A computation of ϵ and the update of the modified-length labels also take $O(m)$ time.

If the DSU operations are implemented as assumed in Section 35, the expansion of the augmenting path, the fragment shrinking, and expansion operations need $O(n \log n)$ time altogether. ∎

This proof illustrates that the dual update can be improved more or less independently from the other parts of the PD algorithm.

It should be possible to improve the complexity for dual updates to $O(n^2)$ and even $O(m \log n)$. Corresponding techniques are well known for 1-matchings (Ball and Derigs [1] gave a good survey of the different implementations).

Goldberg and Karzanov [7] reworked the different dual-update techniques for the shortest valid path problem and claimed the same bounds, but missed saying something about the impact of expansion operations on the data structures (our implementation has turned out some additional technical difficulties compared with the 1-matching case).

We mention that in many situations the last PD iteration requires one-half of the dual updates. Hence, if no feasible solution is known *a priori*, it is strongly recommended to test for feasibility before starting the PD algorithm.

38. A STRONGLY POLYNOMIAL ALGORITHM

We now describe an improved method which heavily depends on the existence of good min-cost flow algorithms. Compared with the method presented in Section 37, the new method determines a circulation rather than an *st*-flow and the network may contain negative arc-length labels.

The general idea is to start the PD algorithm with a near-optimal solution, which is obtained by a solver for ordinary network flow problems. More explicitly, the method is as follows:

Enhanced PD Algorithm

(1) Determine a min-cost circulation f_0 on N and a complementary dual solution π_0.

(2) Put $\phi := 0$ and define symmetric solutions f and π by

$$f(a) := \frac{1}{2}[f_0(a) + f_0(a')],$$

$$\pi(v) := \frac{1}{2}[\pi_0(v) - \pi_0(v')].$$

(3) Transform f into a pseudobasic solution without manipulating the integral arc flow values.

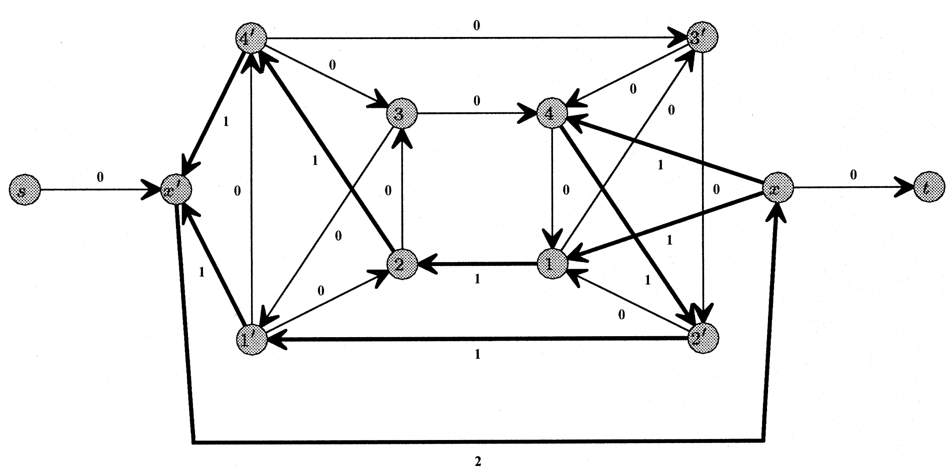

FIG. 3. Symmetrizing a flow.

ID Line: November 16, 2001, 3:41pm W-Networks (39:1) 8u3b1002

Determine the odd cycles $Q_1, Q_2, \ldots, Q_{2k}$ of f.

(4) On every odd cycle Q_i, choose a node v_i with $\pi(v_i) \geq 0$.
Write $Q_i = q_i \circ \overline{q_i'}$, where q_i is a $v_i v_i'$-path.
Push a half-unit of flow through q_i and q_i'.

(5) Add two complementary node pairs x, x', y, y'.
Add arcs yx and $x'y'$ with $f(yx) = lcap(yx) = 0$ and $ucap(yx) = 2k$.
Add two parallel arcs xx' with $lcap(yx) = 0$ and $f(xx') = ucap(xx') = k$.
Add arcs $x'v_i$ and $v_i'x$ with $lcap(x'v_i) = 0$ and $f(x'v_i) = ucap(x'v_i) = 1$.
All new nodes are assigned zero node potentials; all new arcs are fitted with zero-length labels.

(6) Use a straight PD method to find an extreme maximum balanced yy' flow on the modified network.

(7) If the flow value is $2k$, dismiss the artificial arcs. Otherwise, report that no balanced circulation exists.

Before we prove correctness for this algorithm, we illustrate the method by an application to our introductory example.

Consider the circulation $f_0 := f^{p_3}, p_3 = (2, 4', 3, 4, 2', 1, 2)$. This circulation is optimal, which can be seen by computing the distance labels in $N(f)$ from a specified node, say 1. This distance labels constitute optimal node potentials

$$\pi(1) = \pi(1') = \pi(2) = \pi(2') = 0,$$

$$\pi(3) = \pi(3') = \pi(4) = \pi(4') = 1,$$

and one may check the reduced-cost optimality criterion.

If we symmetrize both solutions, we obtain $f = \frac{1}{2}\chi^{p_3}$ and $\pi \equiv 0, \phi \equiv 0$. Note that f is already pseudobasic with odd cycles $Q_1 = (1, 2, 1', 2', 1)$ and $Q_2 = (3, 4, 3', 4', 3)$. If we choose $v_1 = 1$ and $v_2 = 4$ in Step (4), then Step (5) returns the network shown in Figure 3. The arc labels show the flow values at this stage.

The PD algorithm initially shrinks the fragments $v = (\{1, 2, 1', 2'\}, x1')$ and $w = (\{3, 4, 3', 4'\}, x4')$, which corresponds to the former odd cycles and to the skew cuts $(\{1'4', 23, 13'\}, \{24', 1x'\})$ and $(\{41, 3'2', 31'\}, \{42', 4x'\})$.

For the following dual update, one obtains $\epsilon_1 = \epsilon_3 = \infty$ and $\epsilon_2 = \frac{1}{2}$, and the minimum for ϵ_2 is achieved by all arcs joining v to w' or w to v'. Hence, each of these arcs may be used to shrink a blossom.

Then, the flow is augmented depending on which petal (arc closing a blossom) is chosen, and the procedure halts with an optimum balanced circulation. If, for example, the PD algorithm chooses the petal $2'4$, the zero circulation is returned.

Theorem 38.1. *Let $\mu(n, m)$ denote the time necessary to find a min-cost circulation on a network with n nodes and m arcs. Then, the enhanced PD algorithm finds a min-cost balanced circulation in $O(\mu(n, m) + n\rho(n, m))$ time or shows that no balanced circulation exists.*

Proof. We first prove correctness and claim that Step (6) starts with a complementary pair f and (π, ϕ).

First, observe that

$$\begin{aligned} c(a)f(a) + c(a')f(a') &= \frac{1}{2}c(a)[f_0(a) + f_0(a')] \\ &\quad + \frac{1}{2}c(a')[f_0(a) + f_0(a')] \\ &= \frac{1}{2}c(a)[2f_0(a) + 2f_0(a')] \\ &= c(a)f_0(a) + c(a')f_0(a'), \end{aligned}$$

which implies that f and f_0 have equal costs and that

$$\begin{aligned} c^{\pi}(a) &= c(a) + \frac{1}{2}[\pi_0(u) - \pi_0(u')] - \frac{1}{2}[\pi_0(v) - \pi_0(v')] \\ &= \frac{1}{2}[c(a) + \pi_0(u) - \pi_0(u')] \\ &\quad + \frac{1}{2}[c(a') + \pi_0(v) - \pi_0(v')] \\ &= \frac{1}{2}c^{\pi_0}(a) + \frac{1}{2}c^{\pi_0}(a') = c^{\pi}(a'). \end{aligned}$$

Since f and π_0 are complementary, and since f is fractional balanced, $c^{\pi}(a), c^{\pi_0}(a), c^{\pi_0}(a')$ cannot have different signs unless $lcap(a) = ucap(a)$. It follows that f and π are complementary at the end of Step (2).

Steps (3) and (4) manipulate only flow values of f which are nonintegral. Since the reduced-length labels on these arcs are zero, the compatibility of f and π is maintained. At the end of Step (4), f is integral.

Step (5) is needed to remove the imbalances of the nodes $v_1, v_1', v_2, \ldots, v_{2k}'$. The reader may verify without any effort that the extended solutions are still complementary.

This step ends with a circulation which is optimal not only for the ordinary circulation problem, but also for the balanced circulation problem (on the modified network)! This circulation may be viewed as a (0)-optimal balanced yy' flow.

To see the complexity statement, observe that Steps (2), (4), and (5) run in linear time. Step (3) can also be implemented in $O(m)$ time as shown in [5]. Since the flow value is restricted by $2k$, only k phases of the PD algorithm can occur. The claimed complexity follows. ∎

39. APPLICATION TO THE SHORTEST PATH PROBLEM

As mentioned, the presented PD approach is closely related to the shortest valid path algorithms presented by Goldberg and Karzanov [7]. We stress this relationship and give an explicit procedure for finding a shortest valid st-path in a balanced network N, where $t = s'$ and N does not admit a negative-length valid cycle.

This shortest path problem turns into a balanced circulation problem N_1 if we add a return arc ts, put $lcap(ts) = ucap(ts) = 2, c(ts) = 0$, and $ucap \equiv rescap, lcap \equiv 0$ for the original arcs.

If all length labels are nonnegative, this circulation problem is solved by a single iteration of the PD algorithm, that is, the complexity of this problem is $O(\rho(n, m))$. Otherwise, an optimal solution to the circulation problem can be decomposed in $O(nm)$ time into a shortest valid *st*-path pair and some pairs of valid cycles with zero length. It turns out that this operation is dominated by the solution of the circulation problem.

Note that the flow decomposition can be used as a check for negative-length valid cycles in N and that the circulation problem has a feasible solution if and only if there is a valid *st*-path in N.

Let k denote the number of negative arc length labels in N. Put $\pi, \phi :\equiv 0, f(a) := 1$ for all arcs with $c(a) < 0$, and $f(a) := 0$ for the arcs with $c(a) \geq 0$, in order to obtain an initial complementary pair. One can satisfy the mass-balance equations by a problem transformation which works the same way as does the transformation presented in the last section. Let N_2 denote the resulting network.

The maximum value of a balanced flow on N_2 is k if and only if there is a valid *st*-path in N and smaller otherwise. In the positive case, the restriction of an extreme maximum balanced *st*-flow on N_2 yields a balanced circulation on N_1.

We obtain a worst-case time bound of $O(k\rho(n, m) + nm)$ for the shortest path problem. Note that k is $O(m)$, but k can be restricted to $O(n)$ if one uses the simple preprocessing step introduced in [7], namely, replace each node v by a pair v^I and v^O and an arc $v^I v^O$, and put $c(v^I v^O) := 0$. Redirect the original arcs so that the arc uv starts at u^O and ends at v^I. For each original node v, compute $\pi(v^I) := \min\{c(a) : a \in N, a^+ = v \text{ or } a^- = v\}$ and put $\pi(v^O) := -\pi(v^I)$. Then, we have $c^\pi(v^I v^O) = 2\pi(v^I)$ and $c^\pi(a) \geq 0$ for the original arcs.

This transformation can be performed in $O(m)$ time so that we have

Corollary 39.1. *A shortest valid st-path can be computed in $O(n\rho(n, m) + nm)$ time.*

The whole algorithm can be speeded up empirically if the ordinary circulation problem on N_1 is solved first and then symmetrized by the enhanced PD method presented in the last section.

Note that every phase of the PD algorithm essentially solves a shortest path problem on the residual network for the modified-length labels. Hence, $\rho(n, m)$ actually is the problem complexity of solving a shortest valid path problem with nonnegative length labels. If one can establish a complexity of $O(m \log n)$ or $O(n^2)$ as Goldberg and Karzanov [7] claim, our solution of the shortest path problem with negative lengths improves the bound of $O(n^3 + nm \log n)$ presented in [7].

REFERENCES

[1] M.O. Ball and U. Derigs, An analysis of alternative strategies for implementing matching algorithms. Networks 13 (1983), 517–549.

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

[3] J. Edmonds, Maximum matching and a polyhedron with 0,1 vertices. J Res Nat Bur Stand Sect B 69 (1965), 125–130.

[4] 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.

[5] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows (V): Cycle-canceling algorithms. Networks 37 (2001), 202–209.

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

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