Mathematical Programming 78 (1997) 195–217

Flows on hypergraphs

Riccardo Cambini [a], Giorgio Gallo [b,*], Maria Grazia Scutellà [b]

[a] *Dipartimento di Statistica e Matematica applicata all'Economia, Università di Pisa, Pisa, Italy*
[b] *Dipartimento di Informatica, Università di Pisa, Pisa, Italy*

Received 21 January 1994; revised manuscript received 27 June 1996

Abstract

We consider the capacitated minimum cost flow problem on directed hypergraphs. We define spanning hypertrees so generalizing the spanning tree of a standard graph, and show that, like in the standard and in the generalized minimum cost flow problems, a correspondence exists between bases and spanning hypertrees. Then, we show that, like for the network simplex algorithms for the standard and for the generalized minimum cost flow problems, most of the computations performed at each pivot operation have direct hypergraph interpretations.

Keywords: Flows; Leontief flows; Hypergraphs; Simplex algorithm

1. Introduction

Directed hypergraphs have been introduced quite recently in connection with different application areas, such as propositional satisfiability [6,7,11], deductive data bases and Leontief substitution systems [2,9].

Here, we deal with flows on hypergraphs, or *hyperflows*. Flows on hypergraphs can be considered as generalizations of the so-called generalized flows (for the main concepts on the flows we refer to [1]). In [7], the uncapacitated minimum cost flow problem on hypergraphs is considered, and an algorithm is proposed for the particular case of directed hypergraphs having the "gainfree" property. Here, the more general capacitated minimum cost hyperflow problem is considered, and it is shown that, in

* E-mail: gallo@di.unipi.it URL: http://www.di.unipi.it/ ~ gallo/gallo.html

Published by Elsevier Science B.V.
PII S0025-5610(97)00019-1

analogy to what happens for standard minimum cost flow problems, a correspondence exists between bases and spanning hypertrees, a generalization of spanning trees on graphs, and that most of the computations performed by simplex like algorithms have direct and elegant hypergraph interpretations.

In Section 2, the minimum cost hyperflow problem is presented, and the main hypergraph theoretic concepts such as hyperpaths, hypertrees and spanning hypertrees are introduced. In the third section, flows and potentials on spanning hypertrees are defined, and algorithms for their computation are provided. In Section 4, a characterization of the basis matrices in terms of spanning hypertrees is presented. Special cases of this basis characterization are discussed in Section 5. In Section 6, as an exemplification of the use of the concepts introduced, we present the outline of a specialized simplex type algorithm for minimum cost hyperflow computations. Finally, in Section 7, the results of a preliminary experimentation are reported.

2. The hypergraph minimum cost flow problem

A *directed hypergraph* is a pair $\mathscr{H} = (V, E)$, where $V = \{\nu_1, \nu_2, \dots, \nu_n)$ is the set of nodes, and $E = \{e_1, e_2, \dots, e_m\}$ is the set of hyperarcs. A *hyperarc* e is a pair (T_e, h_e), where $T_e \subseteq V$ is the *tail* of e and $h_e \in V \setminus T_e$ is its *head* [1]. Particular hyperarcs are the headless and the tailless hyperarcs, which will be denoted as $(T_e, \emptyset)$ and $(\emptyset, h_e)$, respectively. The size of $\mathscr{H}$ is defined as the sum of the cardinalities of its hyperarcs:

$$size(\mathscr{H}) = \sum_{e_i \in E} |e_i|.$$

In the following, for the sake of simplicity, we shall call the directed hypergraphs simply hypergraphs.

Given a hypergraph $\mathscr{H} = (V, E)$, a positive real *multiplier* $\mu_\nu(e)$ associated with each $\nu \in T_e$, $\forall e \in E$, and a real *demand* vector b associated with V, a *flow on* $\mathscr{H}$ is a function $f: E \to R$ which satisfies the following conservation constraints:

$$\sum_{\nu = h_e} f(e) - \sum_{\nu \in T_e} \mu_\nu(e) f(e) = b(\nu), \ \forall \nu \in V \quad \text{(conservation)}.$$

The flow is *feasible* if it satisfies the following capacity constraints, where u is an *upper capacity* vector associated with E:

$$0 \leqslant f(e) \leqslant u(e), \ \forall e \in E \quad \text{(capacity)}.$$

Let $c(e)$ be the cost associated with the hyperarc e, $\forall e \in E$. Then the *hypergraph minimum cost flow problem* is to find a feasible flow on $\mathscr{H}$ which minimizes the

[1] The directed hypergraphs presented here are a particular case of the more general directed hypergraphs introduced in [5].

function $\sum_{e\in E} c(e)f(e)$. If $A=[A(e_1), A(e_2), \ldots, A(e_m)]$ is the $n\times m$ incidence matrix of $\mathscr{H}$, where by $A(e)$ we denote the column with elements

$$A_\nu(e)=\begin{cases}-\mu_\nu(e) & \text{if } \nu\in T_e,\\ +1 & \text{if } \nu=h_e,\\ 0 & \text{otherwise,}\end{cases}$$

then the problem can equivalently be stated in the following matrix formulation:

$$\begin{aligned}\min \quad & \sum_{e\in E} c(e)f(e) \,^2\\ & Af=b\\ & 0\leqslant f(e)\leqslant u(e), \quad \forall e\in E.\end{aligned}$$

Note that A has one positive component at most in each column, then it is a *pre-Leontief* matrix [12]. The fact that in A the positive component is 1 is not a loss of generality, since it can always be obtained by scaling the variables.

If $\pi(\nu)$ denotes the *potential* which is associated with node ν, and $\lambda(e)$ is the dual variable associated with hyperarc e, then the dual problem, named the *potential problem on hypergraphs*, can be formulated as follows:

$$\begin{aligned}&\max \sum_{\nu\in V} b(\nu)\pi(\nu)-\sum_{e\in E}\lambda(e)u(e)\\ &\pi(h_e)-\sum_{\nu\in T_e}\mu_\nu(e)\pi(\nu)-\lambda(e)\leqslant c(e), \quad \forall e\in E\\ &\lambda(e)\geqslant 0, \quad \forall e\in E.\end{aligned}$$

We introduce next some basic hypergraph concepts. Further concepts on directed hypergraphs can be found in [5].

2.1. Hyperpaths

A *directed path* P_{st} from s to t in $\mathscr{H}$ is a sequence $P_{st}=(\nu_1=s, e_1, \nu_2, e_2, \ldots, e_q, \nu_{q+1}=t)$, where $s\in T_{e_1}$, $h_{e_q}=t$ and $\nu_i\in\{h_{e_{i-1}}\}\cap T_{e_i}$, $i=2,\ldots,q$. If $s=t$, then P_{st} is a *directed cycle*. When no directed cycle exists, then $\mathscr{H}$ is said to be *cycle-free*.

A *directed hyperpath*, Π_{St}, from the source set S to the sink t in $\mathscr{H}$ is a minimal cycle-free sub-hypergraph containing both the nodes in S and node t, and such that each node, with the exception of the nodes in S, has exactly one entering hyperarc.

A hyperarc e' is said to be a *permutation* of a hyperarc e if $T_e\cup\{h_e\}=T_{e'}\cup\{h_{e'}\}$. A hypergraph $\mathscr{H}'$ is a *permutation* of a hypergraph $\mathscr{H}$ if its hyperarcs are permutations of the hyperarcs of $\mathscr{H}$.

[2] Whenever possible, we will not define explicitly a vector as a row or a column vector, and use it in both senses in the matrix multiplications.

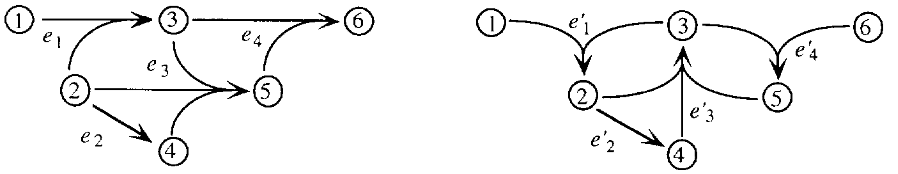

Fig. 1.

A hypergraph Π'_{St} is said to be an *undirected hyperpath* from the source set S to the sink t if among its permutations one exists which is a directed hyperpath from S to t.

Examples of directed and of undirected hyperpaths are shown in Fig. 1. The undirected hyperpath, on the right, is a permutation of the directed one, on the left. Both hyperpaths have $S = \{1, 2\}$ as the source set, and $t = 6$ as the sink. Note that an undirected hyperpath may have directed cycles.

2.2. Hypertrees

The hypergraph $\mathscr{T}_R = (R \cup N, E)$ is a *directed hypertree* (or simply *hypertree*) rooted at R if: (i) it is cycle-free, (ii) $R \cap N = \emptyset$, (iii) each node $v \in N$ has exactly one entering hyperarc and no hyperarc has a node of R as its head.

The set R is called the *root set* of $\mathscr{T}_R$, while N is the set of *non-root* nodes. Any non-root node v not contained in the tail of any hyperarc of $\mathscr{T}_R$ is a *leaf* of the hypertree. From the definition it follows that, for any $v \in N$, there is a unique directed hyperpath Π_{Rv} in $\mathscr{T}_R$.

An *undirected hypertree* rooted at R is any permutation of a directed hypertree rooted at R. In the case of undirected hypertrees, a leaf is a non-root node which belongs to exactly one hyperarc.

Property 2.1. *$\mathscr{H}$ is a (un)directed hyperpath, Π'_{St}, from S to t if and only if it is a (un)directed hypertree having S as the root set and t as the unique leaf.*

A directed hypertree $\mathscr{T}_R$ can be visited in an iterative way starting from its root set, and exploring at each step a new hyperarc and a new node by the following procedure.

procedure VISIT($\mathscr{T}_R$)
  begin
    $j := 0$; $V' := R$; $E' := E$;
    while $\exists\ e \in E'$ **such that** $T_e \subseteq V'$ **do**
      begin
        $E' := E' \setminus \{e\}$; $j = j + 1$; $e_j := e$; $v_j := h_e$; $V' := V' \cup \{h_e\}$
      end {*while*}
  end {*Visit*}.

VISIT($\mathcal{T}_R$) returns an ordering $(R, \nu_1, \nu_2, \ldots, \nu_q)$ of the nodes of $\mathcal{T}_R$, and an ordering $(e_1, e_2, \ldots, e_q)$ of the hyperarcs. VISIT is the equivalent for hypertrees of the top–down visit for standard trees, and the sequence $(R, e_1, \nu_1, e_2, \nu_2, \ldots e_q, \nu_q)$ is a top–down traversal of $\mathcal{T}_R$. Clearly, the sequence $(R, e_1, \nu_1, e_2, \nu_2, .. e_q, \nu_q)$ has the property that, for each $j = 1, 2, \ldots q$, $h_{e_j} = \nu_j$ and $T_{e_j} \subseteq R \cup \{\nu_1, \nu_2, \ldots, \nu_{j-1}\}$.

In the following, we will represent a directed hypertree by means of one of its traversals, by writing, with a little abuse of the notation, $\mathcal{T}_R = (R, e_1, \nu_1, e_2, \nu_2, \ldots e_q, \nu_q)$. When specialized to hyperpaths, which are particular hypertrees, this representation coincides with the one given in [11].

Similarly, an undirected hypertree $\mathcal{T}_R$ can be represented by a sequence $(R, e_1, \nu_1, e_2, \nu_2, \ldots e_q, \nu_q)$, where $\nu_j \in (\{h_{e_j}\} \cup T_{e_j})$ and $(\{h_{e_j}\} \cup T_{e_j}) \setminus \{\nu_j\} \subseteq R \cup \{\nu_1, \nu_2, \ldots, \nu_{j-1}\}$, $j = 1, 2, \ldots, q$. If $\nu \in N$, we shall denote by e_ν the (unique) hyperarc incident to ν in the unique hyperpath in $\mathcal{T}_R$ from R to ν.

2.3. Spanning hypertrees

A *spanning hypertree of* $\mathcal{H} = (V, E)$ is an undirected hypertree $\mathcal{T}_R = (V, E_T)$ such that $E_T \subseteq E$ and $(T_e \cup \{h_e\}) \not\subseteq R$, $\forall e \in E \setminus E_T$.

In the above definition, the property $(T_e \cup \{h_e\}) \not\subseteq R$ $\forall e \in E \setminus E_T$ is to impose a kind of "minimality" of the root set R; in fact, if a hyperarc e exists such that $(T_e \cup \{h_e\}) \subseteq R$, we can use it to span one of the nodes of R, and decrease by one the cardinality of the root set. In the following, the hyperarcs in E_T will be called *tree* hyperarcs, whereas the ones in $E_X = E \setminus E_T$ will be said *external* hyperarcs.

Examples of spanning hypertrees are shown in Fig. 2. Note that, for a given hypergraph with n nodes, it is $|R| + |E_T| = n$, for all spanning hypertrees.

Given a hypergraph $\mathcal{H} = (V, E)$ without isolated nodes, it is possible to construct a spanning hypertree for it by visiting $\mathcal{H}$ as described below. Starting with $R = \emptyset$ and $N = \emptyset$, the procedure SPANNING HYPERTREE adds a hyperarc e to the set of the tree hyperarcs E_T whenever this addition allows the visit of a new node of $\mathcal{H}$, that is whenever there exists a hyperarc e such that $|(T_e \cup \{h_e\}) \setminus (R \cup N)| = 1$. When such a hyperarc does not exist, the procedure selects a hyperarc whose incidence set contains at least 2 unvisited nodes, and inserts into R all its unvisited nodes but one.

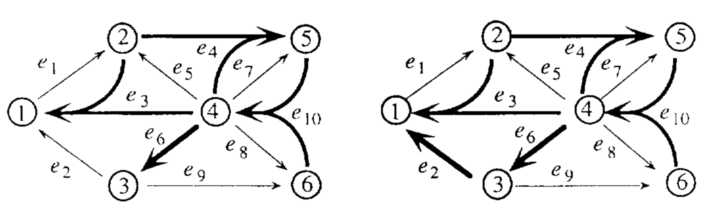

Fig. 2. Left: $\mathcal{T}_R = (\{5, 6\}, e_{10}, 4, e_4, 2, e_3, 1, e_6, 3)$, right: $\mathcal{T}_{R'} = (\{1\}, e_2, 3, e_6, 4, e_3, 2, e_4, 5, e_{10}, 6)$.

Procedure SPANNING HYPERTREE($\mathscr{H}$)
  begin
    $R := \emptyset$; $N := \emptyset$; $E_T := \emptyset$; $j := 0$; $E' := E$; $V' := V$;
    for each $e \in E$ **do** $Q_e := T_e \cup \{h_e\}$;
      begin
        select $a \in E'$ such that $|Q_a| = \min\{|Q_e|: e \in E', Q_e \neq \emptyset\}$;
        select $\nu \in Q_a$;
        if $|Q_a| = 1$ **then** $N := N \cup \{\nu\}$; $V' := V' \setminus \{\nu\}$
          else $R := R \cup (Q_a \setminus \{\nu\})$; $N := N \cup \{\nu\}$; $V' := V' \setminus Q_a$;
        $j := j+1$; $e_j := a$; $\nu_j := \nu$; $E_T := E_T \cup \{a\}$; $E' := E' \setminus \{a\}$; $Q_e := Q_e \setminus Q_a$, $\forall e \in E'$
    until $V' = \emptyset$
  end {*Spanning hypertree*}.

If the elements of E' are maintained in a heap, the algorithm can be implemented in such a way to run in $O(\max\{size(\mathscr{H}), n \log m\})$ time.

Proposition 2.1 *When applied to a hypergraph $\mathscr{H}$ having no isolated node,* SPANNING HYPERTREE *correctly finds a spanning hypertree for $\mathscr{H}$.*

Given a hypergraph $\mathscr{H}$ having no isolated nodes, with incidence matrix A, let $\mathscr{T}_R = (V, E_T)$ be one of its spanning hypertrees and $(R, e_1, \nu_1, e_2, \nu_2, \ldots, e_q, \nu_q)$ one of the traversals of $\mathscr{T}_R$. By reordering the rows and the columns according to the order induced by the traversal, one can rewrite the matrix A in the following *canonical form* (relative to $\mathscr{T}_R$):

$$A = \begin{bmatrix} BC \\ UD \end{bmatrix},$$

where B is a $|R| \times |E_T|$ matrix, C is a $|R| \times |E_X|$ matrix, U is a $|N| \times |N|$ matrix, and D is a $|N| \times |E_X|$ matrix (remember that $|N| = |E_T|$ by definition of hypertree, $E_X = E \setminus E_T$ and $N = \{\nu_1, \nu_2, \ldots, \nu_q\}$). By construction, U is non-singular upper triangular, while each column of D contains one non-zero element at least.

The spanning hypertree $\mathscr{T}_R$ which is associated with $\mathscr{H}$ may not be unique. In fact, consider any external hyperarc e, and $\nu \in N$ be its incidence node of maximal index in the order induced by the traversal. If we substitute $e_\nu \in E_T$ with e, we obtain a new spanning hypertree, $\mathscr{T}_R(e)$, which will be called the *spanning hypertree induced by* e. Node ν will be referred to as the *critical* node of $\mathscr{T}_R(e)$, and will be denoted by ν_e.

Any external hyperarc e induces on $\mathscr{T}_R$ a *hypercycle*, $C(e)$, which is the union of the unique hyperpath from R to ν in $\mathscr{T}_R$, and the unique hyperpath from R to ν in $\mathscr{T}_R(e)$. As an example, the spanning hypertree induced by e_9 on the spanning hypertree in the left of Fig. 2 is $\mathscr{T}_R(e_9) = (\{5, 6\}, e_{10}, 4, e_4, 2, e_3, 1, e_9, 3)$.

3. Flows and potentials

3.1. Flows on spanning hypertrees

For any $|N|$-vector $d(N)$ and any $|E_X|$-vector $f(X)$, there exist unique vectors $d(R)$ and $f(T)$ such that $f=(f(T), f(X))$ is a (possibly unfeasible) flow which satisfies the conservation constraints at the nodes, with $d=(d(R), d(N))$ as the demand vector. [3] Both $f(T)$ and $d(R)$ can be determined in $O(size(\mathscr{H}))$ time by procedure *Flow* given next, which visits $\mathscr{T}_R$ in a bottom-up fashion, starting from the leaves. *Flow* has $\mathscr{H}$, $\mathscr{T}_R$, $d(N)$ and $f(X)$ as input data, while $d(R)$ and $f(T)$ are the outputs.

procedure *Flow*($\mathscr{H}$, $\mathscr{T}_R$, $d(N)$, $f(X)$, $d(R)$, $f(T)$)
  begin
    for each $\nu \in R$ **do** $d(\nu) := 0$;
    for each $e \in E_X$ **do**
      begin
        for each $\nu \in (T_e \cup \{h_e\})$ **do** $d(\nu) := d(\nu) - A_\nu(e) f(e)$
      end {*for*};
    unvisited(ν) := # of the hyperarcs of $\mathscr{T}_R$ incident into ν, $\forall \nu \in V$;
    Queue := {leaves of $\mathscr{T}_R$};
    while *Queue* $\neq \emptyset$ **do**
      begin
        $\nu := select(Queue)$;
        $Queue := Queue \setminus \{\nu\}$;
        $f(e_\nu) := d(\nu)/A_\nu(e_\nu)$;
        * recall that $A_\nu(e_\nu) = 1$ if $\nu = h_{e_\nu}$ and $A_\nu(e_\nu) = -\mu_\nu(e_\nu)$ otherwise *
        for each $w \in (T_{e_\nu} \cup \{h_{e_\nu}\}) \setminus \{\nu\}$ **do**
          begin
            $d(w) := d(w) - A_w(e_\nu) f(e_\nu)$;
            unvisited(w) := *unvisited*(w) − 1;
            if *unvisited*(w) = 1 **and** $w \notin R$ **then** $Queue := Queue \cup \{w\}$
          end {*for*};
      end {*while*};
    for each $\nu \in R$ **do** $d(\nu) := -d(\nu)$
  end {*Flow*}.

[3] If a is a n-vector and I is a subset of the set $\{1, 2, \dots, n\}$, by $a(I)$ we denote the subvector of a containing all the components $a(i)$ with $i \in I$, so that, if $J = \{1, 2, \dots, n\} \setminus I$, it is $a = (a(I), a(J))$, modulo a reordering of the components. In the following, to avoid unnecessary clumsy notation, we will write $f(T)$ and $f(X)$ instead of $f(E_T)$ and $f(E_X)$, respectively.

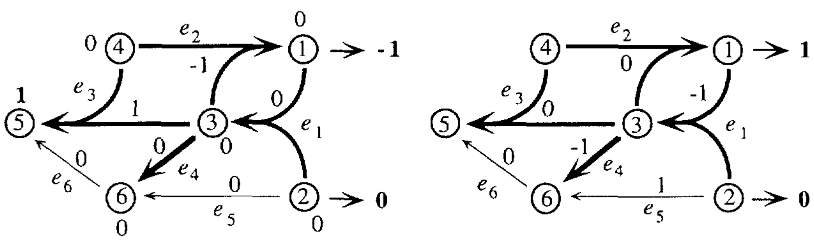

Fig. 3.

Important particular cases are the following:

(i) $f(X)=0$ and $d(N)=u_{\nu}$, where u_{ν} is the unit vector with 1 as the component corresponding to ν, and 0 elsewhere; in this case, $\phi_{\nu}=(f(T), 0)$ is the *characteristic flow* of $\mathcal{T}_R$ relative to ν, and $\rho_{\nu}=d(R)$ is the *root demand* induced by ϕ_{ν}.

(ii) $f(X)=0$ and $d(N)\neq 0$; in this case it is $(f(T), 0)=\sum_{\nu\in N} d(\nu)\phi_{\nu}$, and $d(R)=\sum_{\nu\in N} d(\nu)\rho_{\nu}$.

(iii) $f(X)=u_e$ and $d(N)=0$, where u_e is the unit vector with 1 as the component corresponding to the external hyperarc e, and 0 elsewhere; in this case, $\psi_e=(f(T), u_e)$ is the *unit circulation* on hypercycle $C(e)$, and $\delta_e=d(R)$ is the *root demand* induced by ψ_e.

(iv) $f(X)\neq 0$ and $d(N)=0$; in this case the flow returned by the procedure is called a *circulation*, and it is $(f(T), f(X))=\sum_{e\in E_X} f(e)\psi_e$, and $d(R)=\sum_{e\in E_X} f(e)\delta_e$.

Examples are provided in Fig. 3: on the left, the characteristic flow ϕ_5 and the corresponding root demand vector ρ_5 are given; on the right, the unit circulation ψ_{e5} and δ_{e5} are shown. Here, $\mu_{\nu}(e)=1$ for all pairs (ν, e) except for $(6, e_6)$, for which $\mu_6(e_6)=2$.

Let $\mathcal{T}_R=(R, e_1, \nu_1, e_2, \nu_2, \dots, e_q, \nu_q)$ be a spanning hypertree of $\mathcal{H}$, $e\in E_X$ be one of its external hyperarcs, and ν be the critical node of $\mathcal{T}_R(e)$. Then, we denote by ϕ_{ν}^{e} the *characteristic flow of* $\mathcal{T}_R(e)$ relative to ν, and by ρ_{ν}^{e} the corresponding root demand vector. Clearly, these vectors can be computed directly on the hypergraph by means of a procedure similar to *Flow*.

Observe that, for $k\in\{1,\dots,q\}$, the characteristic flow ϕ_{ν_k} as computed by procedure *Flow* satisfies the following relations:

$$\phi_{\nu_k}(e_k)=\frac{1}{A_{\nu_k}(e_k)},\quad \phi_{\nu_k}(e_j)=0,\ \forall j>k,\quad \phi_{\nu_k}(e)=0,\ \forall e\in E_X.$$

Similarly, if $e\in E_X$ and $\nu_e=\nu_k$ we have that:

$$\phi_{\nu_e}^{e}(e)=\frac{1}{A_{\nu_e}(e)},\quad \phi_{\nu_e}^{e}(e_j)=0,\ \forall j\geqslant k,\quad \phi_{\nu_e}^{e}(e')=0\ \forall e'\in E_X\setminus\{e\}.$$

Since both $\phi_{\nu_e}^{e}$ and ϕ_{ν_e} determine a flow unbalance of one unit at $\nu_e\in N$ and make balanced all the other non-root nodes, the flow $[\phi_{\nu_e}^{e}-\phi_{\nu_e}]$ is a circulation satisfying:

$$\phi_{\nu_e}^{e}(e)-\phi_{\nu_e}(e)=\frac{1}{A_{\nu_e}(e)},\quad \phi_{\nu_e}^{e}(e')-\phi_{\nu_e}(e')=0,\ \forall e'\in E_X\setminus\{e\}.$$

Being ψ_e the circulation on hypercycle $C(e)$ with a flow of value 1 on hyperarc e and a flow of value 0 on all the other external hyperarcs, the following relations hold:

$$\psi_e = A_{\nu_e}(e)\left[\phi^e_{\nu_e} - \phi_{\nu_e}\right] \quad \text{and} \quad \delta_e = A_{\nu_e}(e)\left[\rho^e_{\nu_e} - \rho_{\nu_e}\right]. \tag{3.1}$$

Thus, the circulation ψ_e is proportional to the difference between the characteristic flows of $\mathcal{T}_R(e)$ and of $\mathcal{T}_R$ relative to ν_e. That is, ψ_e and δ_e are the unique circulation and the unique root demand vector needed to push a flow of value $A_{\nu_e}(e)$ from the roots to ν_e, on $\mathcal{T}_R(e)$, and a flow of value $-A_{\nu_e}(e)$ from the roots to ν_e, on $\mathcal{T}_R$.

3.2. Potentials on spanning hypertrees

According to the linear programming terminology, we will call *reduced cost* of hyperarc e the quantity $c(e) + \sum_{\nu \in T_e} \mu_\nu(e)\pi(\nu) - \pi(h_e)$, where $c(e)$ is the cost of e, and $\pi(\nu)$ is the potential of node ν.

For any $|E_T|$-cost vector $c(T)$ and any $|R|$-vector $\pi(R)$, there exist unique potential vector, $\pi(N)$, and cost vector, $c(X)$, such that the reduced cost of each hyperarc of the hypergraph is equal to zero. The following procedure, which can be considered as a dual version of the procedure *Flow*, finds the vectors $\pi(N)$ and $c(X)$ by means of a visit of $\mathcal{T}_R$ from the roots to the leaves. The procedure works in $O(size(\mathcal{H}))$ time.

procedure *Potential*($\mathcal{H}$, $\mathcal{T}_R$, $c(T)$, $\pi(R)$, $c(X)$, $\pi(N)$)
  begin
    for each $e \in E_X$ **do** $c(e) := 0$;
    for each $\nu \in R$ **do**
      begin
        for each $e \in E$ **such that** $\nu \in (T_e \cup \{h_e\})$ **do** $c(e) := c(e) - A_\nu(e)\pi(\nu)$;
      end {for};
    unvisited(e) := # of the nodes of N incident to e, $\forall e \in E$;
    Queue := {e: $e \in \mathcal{T}_R$ and *unvisited*(e) = 1};
    while *Queue* $\neq \emptyset$ **do**
      begin
        * select any element of *Queue* *
        e := *select*(*Queue*);
        Queue := *Queue*\{e};
        let ν be the unique unvisited node of N incident to e;
        * visit node ν *
        $\pi(\nu) := c(e)/A_\nu(e)$;
        for each $e^* \in E\backslash\{e\}$ s.t. $\nu \in (T_{e^*} \cup \{h_{e^*}\})$ **do**
          begin
            $c(e^*) := c(e^*) - A_\nu(e^*)\pi(\nu)$;
            unvisited(e^*) := *unvisited*(e^*) − 1;
            if *unvisited*(e^*) = 1 **and** $e^* \notin E_X$ **then** *Queue* := *Queue* $\cup$ {e^*}

end {*for*}
end {*while*};
for each $e \in E_X$ **do** $c(e) := -c(e)$;
end {*Potential*}.

A particular important case is the one with $c(T) = 0$ and $\pi(R) = u_\nu$; in this case we denote by $\gamma_\nu = c(X)$ the unique vector of the costs to be assigned to the external hyperarcs in order to make their reduced cost zero when $\pi = (u_\nu, \pi(N))$ is the potential vector.

3.3. The root matrix

Let $\mathcal{T}_R = (V, E_T)$ be one of the spanning hypertrees of $\mathcal{H}$, rooted at R, and $A = [{}^{BC}_{UD}]$ be the incidence matrix of $\mathcal{H}$ in canonical form with respect to $\mathcal{T}_R$. We call *root matrix* of $\mathcal{H}$ the $|R| \times |E_X|$ matrix $A_R = (C - BU^{-1}D)$. Each column of A_R corresponds to one of the external hyperarcs, while each of its rows corresponds to one of the roots. In the following, we shall denote by $A_R(*, e)$ the column of A_R corresponding to hyperarc e, and by $A_R(\nu, *)$ the row of A_R corresponding to the root ν. The root matrix is strictly related to the flows and potentials introduced in the previous sections, as shown next.

Proposition 3.1. *For each $e \in E_X$, it is $A_R(*, e) = \delta_e$, where δ_e, as defined in Section* 3.1, *is the unique demand at the roots needed to have a circulation with flow equal to* 1 *on hyperarc e and* 0 *on all the other external hyperarcs.*

Proof. The flow $f = (f(T), f(X))$ and the vector $d = (d(R), d(N))$ returned by *Flow*($\mathcal{H}$, $\mathcal{T}_R$, $d(N)$, $f(X)$, $d(R)$, $f(T)$) satisfy the following relations:

$$f(T) = U^{-1}d(N) - U^{-1}Df(X) \quad \text{and}$$
$$d(R) = (C - BU^{-1}D)f(X) + BU^{-1}d(N). \tag{3.2}$$

In particular, the vector δ_e corresponds to the case $d(N) = 0$ and $f(X) = u_e$; then we have $\delta_e = (C - BU^{-1}D)u_e$, and the proof is so completed. □

Proposition 3.2. *For each $\nu \in R$, it is $A_R(\nu, *) = \gamma_\nu$, where γ_ν, as defined in Section* 3.2, *is the vector of the costs on the external hyperarcs which make the reduced costs equal to* 0 *when the potential relative to ν is fixed at value* 1, *and the one relative to all the remaining roots is fixed at value* 0 (*the costs on the tree hyperarcs are assumed to be zero*).

Proof. The following relations are satisfied by the vectors $c = (c(T), c(X))$ and $\pi = (\pi(R), \pi(N))$ returned by *Potential*($\mathcal{H}$, $\mathcal{T}_R$, $c(T)$, $\pi(R)$, $c(X)$, $\pi(N)$):

$$\pi(N) = c(T)U^{-1} - \pi(R)BU^{-1} \quad \text{and}$$
$$c(X) = c(T)U^{-1}D + \pi(R)(C - BU^{-1}D). \tag{3.3}$$

From the definition of γ_ν, we have $\gamma_\nu = u_\nu(C - BU^{-1}D)$. The proof is so completed. □

From Propositions 3.1 and 3.2 it follows that the root matrix can be computed by labelling techniques, either via procedure *Flow* or via procedure *Potential.*

4. Basis characterization

Let M be the incidence matrix of a sub-hypergraph with n nodes and n hyperarcs of a hypergraph $\mathscr{H}$. If M is non singular, then the sub-hypergraph cannot have isolated nodes, otherwise M should have a zero row, and, as a consequence, it has a spanning hypertree, say $\mathscr{T}_R$. Since $|R| = |E_X|$, M can be converted in canonical form with C and the root matrix M_R being square matrices.

We give next a basis characterization in terms of spanning hypertrees, which, in analogy to what is done in the context of network flows with side constraints [8], is based on a partition of the basis matrix columns into those corresponding to the spanning hypertree's hyperarcs and those corresponding to the external hyperarcs.

Theorem 4.1. *M is a basis if and only if either R is empty or M_R is non singular.*

Proof. Without loss of generality assume M to be in canonical form relative to $\mathscr{T}_R$. If $R = \emptyset$ then $E_X = \emptyset$ too, i.e. the sub-hypergraph associated with M is a hypertree. In this case M reduces to U, and so it is non singular. Viceversa, if M is a non singular matrix in upper triangular form, then $R = \emptyset$. Now, assume $R \neq \emptyset$, and consider the system

$$Mx = \begin{bmatrix} Bx(T) + Cx(X) \\ Ux(T) + Dx(X) \end{bmatrix} = 0,$$

from which we get $x(T) = -U^{-1}Dx(X)$ and $(C - BU^{-1}D)x(X) = M_R x(X) = 0$. Then M is non singular if and only if $x(X) = 0$ is the unique solution of $M_R x(X) = 0$, that is if and only if M_R is non singular. □

Note that the rooted spanning trees, which characterize the basis matrices in the case of standard graphs, are particular spanning hypertrees, where the root set is a singleton.

Like M_R, also its inverse, M_R^{-1}, can be interpreted in terms of flows and potentials. Consider the column of M_R^{-1} corresponding to the root ν, and $M_R^{-1}(e, \nu)$ be its element in the row corresponding to the external hyperarc e. Since

$$M_R M_R^{-1}(*, \nu) = \sum_{e \in E_X} \delta_e M_R^{-1}(e, \nu) = u_\nu, \tag{4.1}$$

from Proposition 3.1 it follows that the column $M_R^{-1}(*, \nu)$ contains the (unique) flow values to be imposed on the external hyperarcs in order to obtain a circulation with demand equal to 1 at root ν and 0 elsewhere.

Similarly, being

$$M_R^{-1}(e, *) M_R = \sum_{\nu \in R} M_R^{-1}(e, \nu) \gamma_\nu = u_e, \tag{4.2}$$

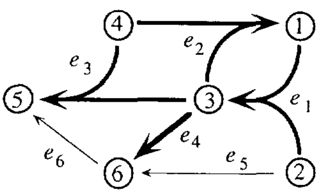

Fig. 4.

from Proposition 3.2 it follows that the row $M_R^{-1}(e, *)$ contains the unique potential values to be imposed on the roots in order to make zero all the reduced costs when the cost of e is equal to 1, and it is 0 for the all the other external hyperarcs (the costs on the tree hyperarcs are assumed to be zero).

In order to illustrate the properties of M_R and of M_R^{-1}, let us consider the hypergraph in Fig. 4, where $\mu_6(e_6) = 2$ and $\mu_\nu(e) = 1$ otherwise.

The matrix in canonical form relative to the spanning hypertree $\mathscr{T}_R = (\{1, 2\}, e_1, 3, e_2, 4, e_3, 5, e_4, 6)$ is

	e1	e2	e3	e4	e5	e6
1	-1	+1	0	0	0	0
2	-1	0	0	0	-1	0
3	+1	-1	-1	-1	0	0
4	0	-1	-1	0	0	0
5	0	0	+1	0	0	+1
6	0	0	0	+1	+1	-2

and its root matrix, $(C - BU^{-1}D)$, is $\begin{bmatrix} +1 & -1 \\ 0 & -2 \end{bmatrix}$.

Consider the spanning hypertree induced by e_5, $\mathscr{T}_R(e_5)$, and its critical node, i.e. 6. The characteristic flows ϕ_6 and $\phi_6^{e_5}$ and their corresponding root demand vectors are:

$$\phi_6 = (+1, 0, 0, +1, 0, 0), \quad \phi_6^{e_5} = (0, 0, 0, 0, +1, 0), \quad \rho_6 = (-1, -1),$$
$$\rho_6^{e_5} = (0, -1).$$

As a consequence, since $M_6(e_5) = +1$, we get

$$(c_1 - BU^{-1}d_1) = M_6(e_5)\left(\begin{bmatrix} 0 \\ -1 \end{bmatrix} - \begin{bmatrix} -1 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Similarly, by considering the spanning hypertree induced by e_6, $\mathscr{T}_R(e_6)$, and its critical node, i.e. 6, we get

$$(c_2 - BU^{-1}d_2) = M_6(e_6)\left(\begin{bmatrix} -1/2 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ -2 \end{bmatrix}.$$

The same result holds for the circulations; for instance, it is $\psi_{e_5} = (-1, 0, 0, -1, 1, 0)$ and $\delta_{e_5} = (+1, 0)$.

As far as the rows of the root matrix are concerned, being $R = \{1, 2\}$ we obtain $\gamma_1 = (1, -1)$ and $\gamma_2 = (0, -2)$. Let us consider now $M_R^{-1} = \left[\begin{smallmatrix} +1 & -1/2 \\ 0 & -1/2 \end{smallmatrix}\right]$. If we consider its first column, relative to root 1, from (4.1) we get $(+1)\delta_{e_5} = (+1)\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$. Similarly, by considering the second column, relative to root 2, it is $(-1/2)\delta_{e_5} + (-1/2)\delta_{e_6} = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$.

Finally, if we consider the rows of M_R^{-1}, from (4.2) it is:

$$(+1)\gamma_1 + (-1/2)\gamma_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad (-1/2)\gamma_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

5. Special cases

5.1. Generalized flows

If $\mathscr{H}$ is a standard graph, i.e. $|T_e| \leqslant 1 \ \forall e \in E$, then the hypergraph network flow model becomes the generalized network flow model [1]. In this problem, the basic solutions are characterized in terms of good augmented trees or forests. An *augmented tree* is either a tree or a tree plus an *extra arc*. If the extra arc exists, then the tree has a specially designed node r, called its root; otherwise, either a headless or a tailless arc must exist. The extra arc induces on the tree a cycle, the *extra cycle*. Let C be the extra cycle, we denote by C^+ and C^-, respectively, the set of the forward arcs and the set of the backward arcs of C with respect to a given orientation. The *cycle multiplier* of C, $\mu(C)$, is defined as follows:

$$\mu(C) = \prod_{(i,j) \in C^+} \mu_i((i,j)) / \prod_{(i,j) \in C^-} \mu_i((i,j)).$$

If 1 unit of flow is sent along C starting from a node ν, then $\mu(C)$ units of flow return to this node. If $\mu(C) > 1$, C is a *gainy cycle*, while if $\mu(C) < 1$, C is a *lossy cycle*.

An augmented tree is a *good augmented tree* if either it is a tree, or the unique cycle determined by the extra arc (*extra cycle*) is either "lossy" or "gainy".

An *augmented forest* is a collection of node-disjoint augmented trees which span all the nodes of the graph. The forest is a *good augmented forest* if each of its components is a good augmented tree. The following characterization holds:

Theorem 5.1. *Let $\mathscr{H}$ be a generalized graph with n nodes and n arcs. Then $\mathscr{H}$ is a basis graph if and only if it is a good augmented forest.*

A good augmented forest, $\mathscr{H}$, can be seen as a particular spanning hypertree, where the set of the external hyperarcs is the set of the extra arcs. Its incidence matrix, M, is block-diagonal, with one block for each components, and $\mathscr{H}$ is a basis graph iff each block is non singular. Let $\mathscr{T}$ be one of the components of $\mathscr{H}$ which contains an extra arc, and let r be its root node. Let *imbalance*(r) be the imbalance at r when the flow on the extra arc of $\mathscr{T}$ is set to value 1. By definition of gainy and of lossy cycle, the extra cycle of $\mathscr{T}$ is either lossy or gainy if and only if *imbalance*(r) is not zero. By the characterization of the root matrix provided in Section 3.3, *imbalance*(r) is the unique element of the (1×1) root matrix of $\mathscr{T}$, and so *imbalance*(r) is not zero if and only if the root matrix of $\mathscr{T}$ is non singular. Then, Theorem 5.1 can be reformulated as follows:

Theorem 5.2. *Let $\mathscr{H}$ be a generalized graph with n nodes and n arcs. Then $\mathscr{H}$ is a basis graph if and only if it is an augmented forest in which each component $\mathscr{T}$ has either root set empty or the associated root matrix is non singular.*

Being Theorem 5.2 a special case of Theorem 4.1, the basis characterization for generalized flows is a special case of the basis characterization for hypergraph flows provided in Section 4.

5.2. The gainfree Leontief substitution problem

When $u(e) = +\infty \; \forall e \in E$, the minimum cost hyperflow problem is known as the *Leontief flow problem*; if, in addition, the nonnegativity of the demand vector b is imposed, then it is called a *Leontief substitution flow problem*. These problems have been studied in [7] under the hypothesis that for each directed cycle $(\nu_1, e_1, \nu_2, e_2, \ldots, e_q, \nu_{q+1} = \nu_1)$ it is

$$\prod_{i=1}^{q} \mu_{\nu_i}(e_i) \leqslant 1 \quad (\textit{gainfree property}).$$

The following theorem characterizes the gainfree Leontief substitution flows in terms of their *support hypergraphs*, i.e. the hypergraphs induced by the hyperarcs $e \in E$ for which $f(e) > 0$.

Theorem 5.3 (Jeroslow et al. [7]). *If a Leontief substitution flow problem is gainfree, then the support hypergraph of every basic feasible solution has no directed cycle.*

A new characterization of gainfree Leontief substitution flow problems, which is based on hypertrees, is provided below.

Lemma 5.1. *Let $\mathscr{H}$ be a hypergraph with n nodes and n hyperarcs, and M be its incidence matrix. $\mathscr{H}$ is a directed hypertree $\mathscr{T}_R$ with $R = \emptyset$ if and only if $\mathscr{H}$ has no directed cycles and M has exactly one positive element in each row and in each column.* □

Proof. It follows easily from the definition of directed hypertree. □

Theorem 5.4. *If a Leontief substitution flow problem is gainfree, then the support hypergraph of every basic feasible solution is a directed hypertree with root set empty.*

Proof. Let f be a basic feasible flow, $\mathscr{H}_f = (V_f, E_f)$ its support hypergraph, M_f the correspondent incidence matrix and b_f the subvector of b relative to the nodes of V_f; denote by f_+ the subvector of the positive components of f. Note that M_f is a submatrix of the basis matrix M corresponding to f, and that $M_f f_+ = b_f \geqslant 0$.

Being $f_+ > 0$ and $M_f f_+ \geqslant 0$, each row of M_f has at least one positive element, which implies $|V_f| \leqslant |E_f|$ since each column has one positive element at most. The columns of M_f are linearly independent because the columns of M correspondent to the hyperarcs in E_f are l.i., belonging to a basis matrix, and their elements in the rows corresponding to nodes not in V_f are zero. Then $\text{rank}(M_f) = |E_f|$, which implies $|V_f| \geqslant |E_f|$, and hence $|V_f| = |E_f|$. Thus, M_f is a square pre-Leontief matrix with exacly one positive element in each row and in each column. By Theorem 5.3 $\mathscr{H}_f$ has no directed cycles; then, by Lemma 5.1, $\mathscr{H}_f$ is a directed hypertree with root set empty. □

6. A hypergraph network simplex algorithm

Here we describe a specialization of the primal simplex algorithm to hypergraphs, which makes use of the basis characterization and of the procedures presented in Sections 3 and 4. This algorithm is intended only as an exemplification of how the results presented so far can be used to devise algorithms based on labelling techniques for the hypergraph flow problem, similarly to what is done for network flow problems.

6.1. Solving linear systems

In the simplex method applied to the flow problem on hypergraphs, linear systems of type $Mf = \bar{b}$ and $\pi M = \bar{c}$, where M is a $n \times n$ basis and $\bar{b} = (\bar{b}(R), \bar{b}(N))$ and $\bar{c} = (\bar{c}(T), \bar{c}(X))$ are n-vectors, need to be solved.

The first system can be interpreted as the problem of finding on the sub-hypergraph whose incidence matrix is M a flow f satisfying a given demand vector $\bar{b}$. We show that the solution to the system $Mf = \bar{b}$ can be obtained as the sum of a flow and of a circulation.

Theorem 6.1. *Let $\mathscr{H}$ be a basis hypergraph with incidence matrix M, $\mathscr{T}_R$ be a spanning hypertree for $\mathscr{H}$, λ be the vector $M_R^{-1}(\bar{b}(R) - \sum_{\nu \in N} \bar{b}(\nu)\rho_\nu)$, and f be the unique solution to the linear system $Mf = \bar{b}$; then:*

$$f = \begin{cases} \sum_{\nu \in N} \bar{b}(\nu)\phi_\nu + \sum_{e \in E_X} \lambda(e)\psi_e & \text{if } R \neq \emptyset, \\ \sum_{\nu \in N} \bar{b}(\nu)\phi_\nu & \text{otherwise.} \end{cases}$$

Proof. When $R \neq \emptyset$, by simple manipulations one gets:

$$\begin{cases} f(T) = -U^{-1}Df(X) + U^{-1}\bar{b}(N), \\ f(X) = (C - BU^{-1}D)^{-1}(\bar{b}(R) - BU^{-1}\bar{b}(N)) = M_R^{-1}(\bar{b}(R) - BU^{-1}\bar{b}(N)). \end{cases}$$

From (3.2), by recalling the definitions of characteristic flow and of circulation, one gets $f = (f(T), f(X)) = \sum_{\nu \in N}\bar{b}(\nu)\phi_\nu + \sum_{e \in E_X}\lambda(e)\psi_e$, where $\lambda = M_R^{-1}(\bar{b}(R) - \sum_{\nu \in N}\bar{b}(\nu)\rho_\nu)$. The case in which $R = \emptyset$ is trivial. □

From Theorem 6.1 it follows that, given M_R^{-1}, the linear system $Mf = \bar{b}$ can be solved directly on the hypergraph; in fact the following algorithm, which uses procedure *Flow* as a subroutine, allows one to solve the system in $O(\max\{|R|^2, size(\mathscr{H})\})$ time. Only the case $R \neq \emptyset$ is considered; in fact, when $R = \emptyset$, f can be easily computed by procedure *Flow* alone, since no external hyperarc exists.

procedure *Primal*($\mathscr{H}, \mathscr{T}_R, M_R^{-1}, \bar{b}, f$)
  begin
    Flow($\mathscr{H}, \mathscr{T}_R, \bar{b}(N), 0, d(R), f(T)$);
    $f(X) := M_R^{-1}(\bar{b}(R) - d(R))$;
    Flow($\mathscr{H}, \mathscr{T}_R, \bar{b}(N), f(X), \bar{b}(R), f(T)$)
  end {*Primal*}.

Primal first finds the flow which satisfies the flow requirements at the non-roots and the relative root demand vector $d(R)$ (first call to *Flow*). Then it computes the circulation which yields a flow vector $f(X) = M_R^{-1}(\bar{b}(R) - d(R))$ on the external hyperarcs, and adds this circulation to the previously computed flow (second call to *Flow*). Therefore, to calculate the flow $f = M^{-1}\bar{b}$, only the knowledge of M_R^{-1} is needed; U^{-1} is in fact implicitly computed by means of the visit of $\mathscr{T}_R$ done in the procedure *Flow*.

Consider now the system $\pi M = \bar{c}$. Let π_0 and c_0 be the potential vector and the cost vector on the external hyperarcs returned by *Potential* when $\pi(R) = 0$, and let π_1 be the potential vector returned by *Potential* when $\bar{c}(T) = 0$ and $\pi(R) = (\bar{c}(X) - c_0)M_R^{-1}$.

Theorem 6.2. *Let $\mathscr{H}$ be a basis hypergraph with incidence matrix M and $\mathscr{T}_R$ be a spanning hypertree for $\mathscr{H}$. The unique solution to the linear system $\pi M = \bar{c}$ is:*

$$\pi = \begin{cases} \pi_0 + \pi_1 & \text{if } R \neq \emptyset, \\ \pi_0 & \text{otherwise.} \end{cases}$$

Proof. By simple algebra, assuming $R \neq \emptyset$ (otherwise the proof would be trivial), and remembering formula (3.3), one has

$$\pi(N) = \bar{c}(T)U^{-1} - \pi(R)BU^{-1},$$

$$\pi(R) = (\bar{c}(X) - \bar{c}(T)U^{-1}D)M_R^{-1} = (\bar{c}(X) - c_0)M_R^{-1};$$

hence

$$\pi = \left(0, \bar{c}(T)U^{-1}\right) + \left(\bar{c}(X) - c_0\right)M_R^{-1}\left(I, -BU^{-1}\right) = \pi_0 + \pi_1. \quad \square$$

Thus, as for the primal case, also the system $\pi M = \bar{c}$ can be solved directly on the hypergraph. In fact, given M_R^{-1}, the following algorithm solves it in $O(\max\{|R|^2, size(\mathscr{H})\})$ time. When $R \neq \emptyset$, π is yielded directly by the procedure *Potential*.

procedure *Dual*($\mathscr{H}$, $\mathscr{T}_R$, M_R^{-1}, $\bar{c}$, π)
  begin
    Potential($\mathscr{H}$, $\mathscr{T}_R$, $\bar{c}(T)$, 0, c_0, $\pi_0(N)$);
    $\pi(R) := (\bar{c}(X) - c_0)M_R^{-1}$;
    Potential($\mathscr{H}$, $\mathscr{T}_R$, $\bar{c}(T)$, $\pi(R)$, $\bar{c}(X)$, $\pi(N)$)
  end {*Dual*}.

6.2. *Initial spanning hypertree*

An initial feasible basis can be obtained by introducing, for each supply node $\nu(b(\nu) < 0)$, the artificial hyperarc $(\nu, \emptyset)$, and for each demand or transshipment node ν $(b(\nu) \geqslant 0)$, the artificial hyperarc $(\emptyset, \nu)$. Artificial hyperarcs are assigned a sufficiently large cost C and infinite capacity. The set of the artificial hyperarcs induces a primal-feasible basis spanning hypertree with root set empty. The corresponding primal and dual solutions can be computed directly by procedures *Flow* and *Potential*, since the root matrix is empty. As it will be shown, the inverse of the root matrix can be updated directly on the hypergraph at each pivot operation, so that no explicit inversion operations are needed. At termination, unless the problem is unfeasible, no artificial hyperarc carries flow.

6.3. *Optimality testing and basis change*

Let M be the current feasible basis, $\mathscr{H}^*$ the corresponding hypergraph and $\mathscr{T}_R$ one of its spanning hypertrees. Given the inverse of the root matrix, M_R^{-1}, it is easy to compute the primal basic solution $f = M^{-1}b^*$ and the corresponding dual vector $\pi = c^*M^{-1}$, where b^* is the demand vector induced on the nodes by the flows on the non-basic hyperarcs, while c^* is the cost vector relative to the basic hyperarcs. In fact, *Primal* and *Dual* can be used.

The optimality conditions are, as usual, based on the reduced costs: the out of basis hyperarcs must have reduced costs $\geqslant 0$ if their flow is zero, and reduced costs $\leqslant 0$ if their flow is at the upper bound. If these conditions are satisfied, M is optimal and the algorithm terminates. Otherwise, the algorithm selects a hyperarc e' out of basis which violates the optimality conditions (the *entering hyperarc*), and forces it into the basis.

Now, we show how the *leaving hyperarc* can be determined. Consider the case in which $f(e') = 0$. Let $\bar{f}$ be the unique flow vector on the basic hyperarcs which, together with a unit flow on e', satisfies the flow conservation constraints at all the nodes, i.e. $\bar{f}$ is the solution to the system $M f = -A(*, e')$, which can be determined by means of procedure *Primal*. The hyperarcs whose corresponding components in $\bar{f}$ are different from zero, together with e', constitute a hypercycle. The basis change is obtained by pushing the maximum allowed flow on such hypercycle; the leaving hyperarc is one of the bottleneck hyperarcs. In practice we compute the vector s with components

$$s(e) = \begin{cases} u(e) & \text{if } e = e', \\ (u(e) - f(e))/\bar{f}(e) & \text{if } e \in \mathscr{H}^*,\ e \neq e',\ \bar{f}(e) > 0, \\ f(e)/-\bar{f}(e) & \text{if } e \in \mathscr{H}^*,\ e \neq e',\ \bar{f}(e) < 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then, $e^* = \operatorname{argmin}\{s(e)\colon\ e \in \{\mathscr{H}^* \text{ with } \bar{f}(e) \neq 0\} \cup \{e'\}\}$ is the hyperarc which leaves the basis. Clearly, if $e^* = e'$, then the basis remains unchanged and the effect of the pivot operation is that the flow on e' goes from value zero to the value $u(e')$. The flow f' with

$$f'(e) = \begin{cases} f(e) + s(e^*) & \text{if } e = e', \\ f(e) + \bar{f}(e) s(e^*) & \text{if } e \in \mathscr{H}^*, \\ f(e) & \text{otherwise,} \end{cases}$$

is the new basic flow. Similar formulae can be found when $f(e') = u(e')$.

6.4. How to update the spanning hypertree and M_R^{-1}

To analyze the basis change, different cases must be considered. To simplify the presentation, in the following we shall assume the nodes ordered according to the row order of matrix M, as described in Section 2.

(1) If e^* is external, then two cases are possible: either $(T_{e'} \cup \{h_{e'}\}) \not\subseteq R$ (*case 1.a*), in which case e' enters as an external hyperarc, or $(T_{e'} \cup \{h_{e'}\}) \subseteq R$ (*case 1.b*), in which case e' enters as a tree hyperarc. In the former case the root set does not change ($R' = R$), while in the latter case one of the nodes in $(T_{e'} \cup \{h_{e'}\})$ is dropped from the root set R, so decreasing by one the cardinality of the root set and the size of the root matrix ($R' \subset R$).

(2) If e^* is a tree hyperarc and an external hyperarc $\bar{e}$ exists which can replace e^* in the hypertree (*case 2.a*), i.e. the critical node of $\mathscr{T}_R(\bar{e})$, say ν, is such that $e_\nu = e^*$ (according to the definitions given in Section 2), then we can exchange e^* and $\bar{e}$ (i.e. $\bar{e}$ becomes a tree hyperarc and e^* becomes external), and easily reduce to the previous case.

If no such hyperarc exists (*case 2.b*), let $\bar{\nu}$ be the non-root node of maximal index in $(T_{e^*} \cup \{h_{e^*}\})$. Two cases are possible: either $\bar{\nu}$ is the node of maximal index in $(T_{e'} \cup \{h_{e'}\})$ as well, in which case e' replaces e^* as a tree hyperarc (*case 2.b.1*), or the above property is not true (*case 2.b.2*); in the latter case the new root set is given by $R' = R \cup \{\bar{\nu}\}$, and e' enters as an external hyperarc unless $(T_{e'} \cup \{h_{e'}\}) \subseteq R$; in this last

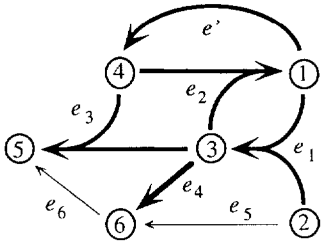

Fig. 5.

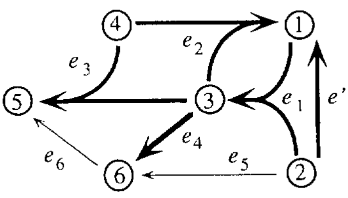

Fig. 6.

case one of the nodes of $(T_{e'} \cup \{h_{e'}\})$ can be dropped, and the cardinality of R does not change.

The above cases are illustrated for the hypergraph in Fig. 3, where $\mathcal{T}_R = (\{1, 2\}, e_1, 3, e_2, 4, e_3, 5, e_4, 6)$, $R = \{1, 2\}$ and $E_X = \{e_5, e_6\}$.

Case 1: $e^* = e_5$ $(e^* \in E_X)$.

Case 1.a: $(T_{e'} \cup \{h_{e'}\}) \not\subseteq R$. See Fig. 5: $R' = R = \{1, 2\}$, $E'_X = \{e_5, e_6\} \setminus \{e_5\} \cup \{e'\} = \{e_6, e'\}$.

Case 1.b: $(T_{e'} \cup \{h_{e'}\}) \subseteq R$. See Fig. 6: $R' = R \setminus \{1\} = \{2\}$, $E'_X = E_X \setminus \{e_5\} = \{e_6\}$, $\mathcal{T}'_R = (\{2\}, e', 1, e_1, 3, e_2, 4, e_3, 5, e_4, 6)$.

Case 2: $e^* \in \mathcal{T}_R$.

Case 2.a: $e^* = e_3$: in this case, independently of e', $e_6 \in E_X$ can replace e_3 as a tree hyperarc, so obtaining $E'_X = \{e_5, e_3\}$ and $\mathcal{T}'_R = (\{1, 2\}, e_1, 3, e_2, 4, e_4, 6, e_6, 5)$; we can then apply the transformations in Case 1.

Case 2.b: $e^* = e_2$ $(\bar{v} = 4)$.

Case 2.b.1: $e' = (\{1, 3\}, 4)$. See Fig. 7: $E'_X = E_X = \{e_5, e_6\}$, $\mathcal{T}'_R = (\{1, 2\}, e_1, 3, e', 4, e_3, 5, e_4, 6)$.

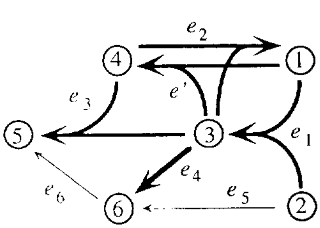

Fig. 7.

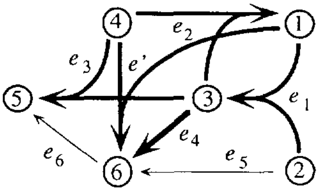

Fig. 8.

Case 2.b.2: $e' = (\{1, 4\}, 6)$ $((T_{e'} \cup \{h_{e'}\}) \not\subseteq R)$. See Fig. 8: $E'_X = E_X \cup \{e'\}$, $R' = R \cup \{4\}$, $\mathcal{T}'_R = (\{1, 2, 4\}, e_1, 3, e_3, 5, e_4, 6)$
$e' = (1, 2)$ $(T_{e'} \cup \{h_{e'}\}) \subseteq R)$. See Fig. 9: $E'_X = E_X$, $R' = R \setminus \{2\} \cup \{4\} = \{1, 4\}$, $\mathcal{T}'_R = (\{1, 4\}, e', 2, e_1, 3, e_3, 5, e_4, 6)$.

Now, let us consider the updating of M_R^{-1}. Let $M_R^{-1}(e, \nu)$ be the element of M_R^{-1} in the row corresponding to the external hyperarc e and in the column corresponding to the root ν. Let ψ_ν be the unique circulation which induces a demand 1 at ν and a demand 0 at all the other root nodes, and which has value 0 on all the hyperarcs not belonging to the basis hypergraph $\mathscr{H}^*$. Due to the characterization of M_R^{-1} given in Section 4 it is $\psi_\nu(e) = M_R^{-1}(e, \nu)$, $\forall e \in E_X$. Clearly, for all $\nu \in R$, the new vector $\overline{\psi}_\nu$ with

$$\overline{\psi}_\nu(e) = \begin{cases} \psi_\nu(e) - \dfrac{\psi_\nu(e^*)}{\bar{f}(e^*)}\bar{f}(e), & e \in \mathscr{H}^*, \\ -\dfrac{\psi_\nu(e^*)}{\bar{f}(e^*)}, & e = e', \\ 0, & \text{otherwise}, \end{cases} \tag{6.1}$$

is a circulation which induces at the root nodes the same demand as ψ_ν and has a value different from zero only on the hyperarcs belonging to the new basis. That allows us to update the columns of the inverse of the new root matrix, $M_R'^{-1}$, in the case in which it is $R' \subseteq R$: in fact, for each external hyperarc e, $M_R'^{-1}(e, \nu) = \overline{\psi}_\nu(e)$. In the case in which a new root, say $\bar{\nu}$, enters the root set, in order to apply the updating formula (6.1), the vector $\psi_{\bar{\nu}}$ needs to be computed. Since, with the notations previously introduced, it is $\psi_{\bar{\nu}} = M^{-1} u_{\bar{\nu}}$, it is possible to perform such a computation by means of the procedure *Primal*.

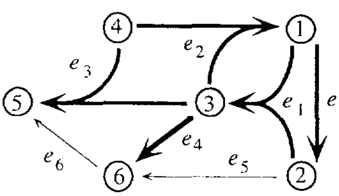

Fig. 9.

7. Experimental results

A prototype hypergraph simplex algorithm based on the ideas described in this paper has been implemented in C language. A detailed description of the implementation choices (starting feasible basis, pricing rule, reoptimization techniques, etc.), together with the results of a wide experimentation, can be found in [4].

Here we present some results of the experimentation, which has been performed on a workstation HP 9000/710, using random hypergraphs with (0, ± 1) incidence matrices. In order to assess the effectiveness of our approach, a comparison with a state-of the art LP code, the primal version of CPLEX (Version 3.0), has been done. The results obtained, as reported in Figs. 10, 11 and 12, are promising, and, although still preliminary, suggest that our approach becomes increasingly effective as the size and the density of the instances increase, and that the solution time is a rather slow increasing function of the size. In particular, when the number n of the nodes and the maximum cardinality k of the tail sets are fixed, our implementation becomes faster than CPLEX

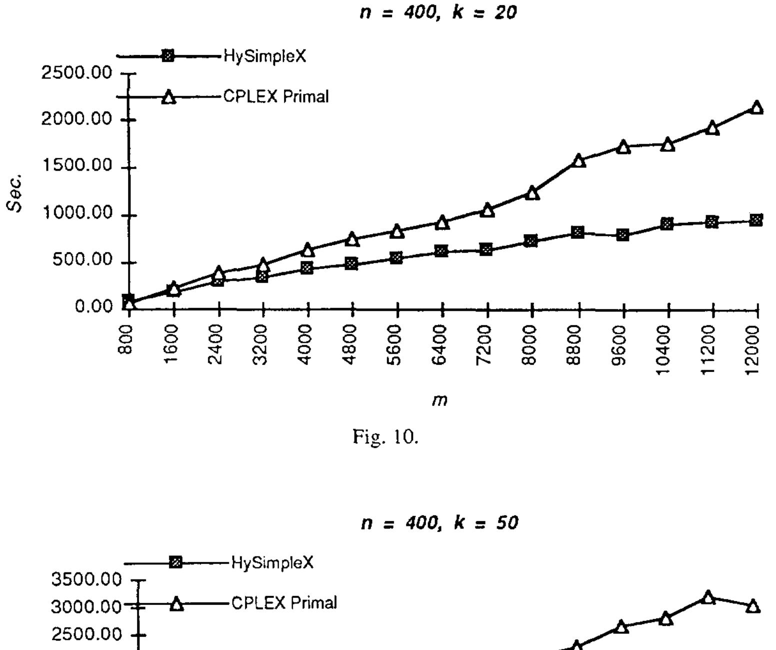

Fig. 10.

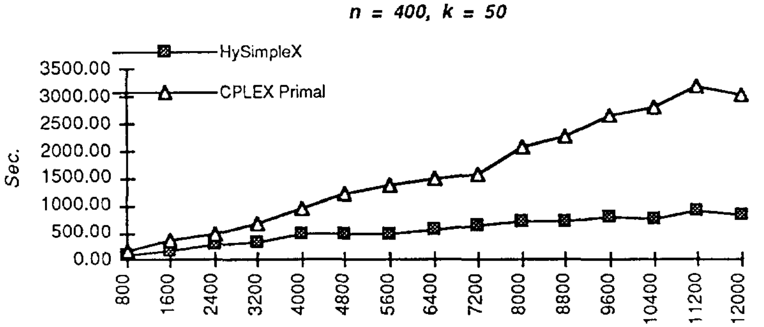

Fig. 11.

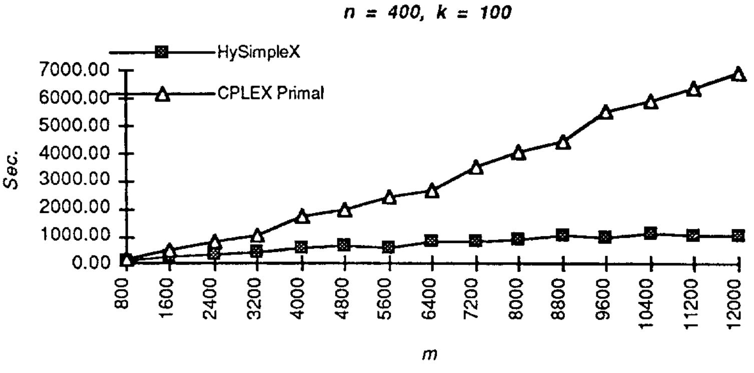

Fig. 12.

as the number of the arcs grows. In the figures, the running times are in seconds, and each one is the average CPU time over 5 runs in which only the seed of the hypergraph generator has been changed.

Remarks. As suggested by an anonymous referee, it is important to observe that the minimum cost hyperflow problem includes all the bounded linear programming problems. In fact, if A denotes the constraint matrix, it is sufficient to replace each column $A(*, j)$ with a column $A^+(*, j)$, having the positive coefficients of $A(*, j)$, with a column $A^-(*, j)$, having its negative coefficients, and with a constraint equating the two corresponding variables. Then, substitution for the variable corresponding to $A^+(*, j)$ by the slack in its upper bound constraint produces a direct hypergraph format. This observation was already present in [3], with reference to a particular application.

Acknowledgements

The authors are indebted with the referees for their suggestions and helpful comments.

References

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications* (Prentice Hall, Englewood Cliffs, NJ, 1993).

[2] G. Ausiello, A. D'Atri and D. Saccà, Graph algorithms for functional dependency manipulation, *Journal of ACM* 30 (1983) 752–766.

[3] R. Cambini, G. Gallo and M.G. Scutellà, Minimum cost flows on hypergraphs, Technical Report TR-1/92, Dipartimento di Informatica, University of Pisa, Pisa, 1992.

[4] G. Gallo, F. Licheri and M.G. Scutellà, The hypergraph simplex approach: Some experimental results, *Ricerca Operativa ANNO* XXVI, 78 (1996) 21–54.

[5] G. Gallo, G. Longo, S. Nguyen and S. Pallottino, Directed hypergraphs and applications, *Discrete Applied Mathematics* 42 (1993) 177–201.

[6] G. Gallo and G. Rago, A hypergraph approach to logical inference for datalog formulae, Technical Report TR-28/90, Dipartimento di Informatica, University of Pisa, Pisa, 1990.

[7] R.G. Jeroslow, R.K. Martin, R.R. Rardin and J. Wang, Gainfree Leontief substitution flow problems, *Mathematical Programming* 57 (1992) 375–414.

[8] J.L. Kennington and R.V. Helgason, *Algorithms for Network Programming* (Wiley/Interscience, New York, 1980) Chapter 7.

[9] G.J. Koehler, A.B. Whinston and G.P. Wright, *Optimization over Leontief Substitution Systems* (North-Holland, Amsterdam, 1975).

[10] E. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart and Winston, New York, 1976).

[11] A.F. Torres and J.D. Araoz, Combinatorial models for searching in knowledge bases, *Mathematicas, Acta Cientifica Venezolana* 39 (1988) 387–394.

[12] A.F. Veinott Jr, Extreme points of Leontief substitution systems, *Linear Algebra and Its Application* 1 (1968) 181–194.