Technical Report: TR-4-96, March 08, 1996

Flows on hypergraphs

Cambini, Riccardo Gallo, Giorgio Scutella', Maria Grazia

Dipartimento di Informatica, Università di Pisa
corso Italia 40, 56125 Pisa, Italy

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.

ACM–CR Subject Classification:

Keyword and phrases: Flows, hypergraphs

1. Introduction

Directed hypergraphs have been introduced quite recently in connection with different application areas, such as propositional satisfiability (Torres and Araoz, 1988; Jeroslow, Martin, Rardin and Wang, 1993; Gallo and Rago, 1990), deductive data bases and Leontief substitution systems (Ausiello, D'Atri and Saccà, 1983; Koehler, Whinston and Wright, 1975).

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 the book of Ahuja, Magnanti and Orlin (1993)).

In the paper by Jeroslow et al. (1993), the uncapacitated minimum cost flow problem on hypergraphs is considered, and an algorithm is proposed for its solution for the particular case of directed hypergraphs having the "gainfree" property. Here, the more general capacitated case is considered. We define spanning hypertrees of a hypergraph 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.

The plan of the paper is the following. 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 $\mathtt{H}=(V,E)$, where $V=\{v_1, v_2, ..., v_n\}$ is the set of nodes, and $E=\{e_1, e_2, ..., e_m\}$ 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 $(T_e,\varnothing)$ and $(\varnothing,h_e)$, respectively. While the size of a standard graph is uniquely defined by n and m, the size of a directed hypergraph depends also on the cardinality of its hyperarcs; so, the size of $\mathtt{H}$ is defined as the sum of the cardinalities of its hyperarcs: $size(\mathtt{H})= \sum_{e_i \in E} |e_i|$. In the following, for the sake of simplicity, we shall call the directed hypergraphs simply hypergraphs.

---

[1] The directed hypergraphs presented here are a particular case of the more general directed hypergraphs introduced by Gallo, Longo, Nguyen and Pallottino (1992).

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

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

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

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

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

$$A_v(e) = \begin{cases} -\mu_v(e) & \text{if } v \in T_e \\ +1 & \text{if } v = h_e \\ 0 & \text{otherwise} \end{cases},$$

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

$$\begin{aligned} &\min \sum_{e \in E} c(e) f(e) \ (*) \\ &A f = b \\ &0 \le f(e) \le u(e) \quad \forall e \in E. \end{aligned}$$

Note that *A* has one positive component at most in each column, then it is a *pre-Leontief* matrix (Veinott, 1968). 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 $\pi(v)$ denotes the *potential* which is associated with node v, 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:

$$\max \sum_{v \in V} b(v)\pi(v) - \sum_{e \in E} \lambda(e) u(e)$$

$$\pi(h_e) - \sum_{v \in T_e} \mu_v(e)\pi(v) - \lambda(e) \le c(e), \ \forall e \in E$$

$$\lambda(e) \ge 0, \ \forall e \in E.$$

In order to describe the hypergraph-theoretic interpretation of the bases of the linear programming formulation of the problem, let us introduce some particular hypergraphs. For further definitions concerning directed hypergraphs, we refer to the paper of Gallo et al. (1992).

(*) 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.

2.1. Hyperpaths

A *directed path* P_{st} from s to t in H is a sequence $P_{st}=(v_1=\{s\}, e_1, v_2, e_2, ..., e_q, v_{q+1}=\{t\})$, where $s\in T_{e_1}$, $h_{e_q}=t$ and $v_i \in \{h_{e_{i-1}}\}\cap T_{e_i}$, $i=2, .., q$. If $s=t$, then P_{st} is a *directed cycle*. When no directed cycle exists, then H is said to be *cycle-free*.

A *directed hyperpath*, Π_{St}, from the source set S to the sink t in 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 H′ is a *permutation* of a hypergraph H if its hyperarcs are permutations of the hyperarcs of H.

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.

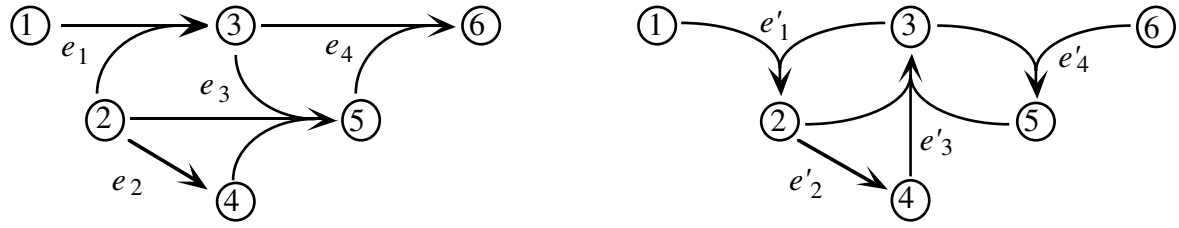

Fig. 1

2.2. Hypertrees

The hypergraph $T_R=(R\cup N, E)$ is a *directed hypertree* (or simply *hypertree*) rooted at R if: (i) it is cycle-free, (ii) $R\cap N=\varnothing$, (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 call the *root set* of 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 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 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- 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 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(T_R)
begin
    j:=0 ; V':=R; E':=E;
    while ∃ e∈E' such that T_e ⊆V' do
    begin
        E':=E' \ {e}; j = j + 1; e_j := e; v_j:= h_e; V':=V' ∪ {h_e}
    end {while}
end {Visit}.
```

VISIT($\mathtt{T}_R$) returns an ordering (R, v_1, v_2, ..., v_q) of the nodes of $\mathtt{T}_R$, and an ordering (e_1, e_2, ..., 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, v_1, e_2, v_2, . . . e_q, v_q) is a top-down traversal of $\mathtt{T}_R$. These considerations suggest an alternative definition of directed hypertrees:

$\mathtt{T}_R$=($R \cup N$, E) is a directed hypertree if a sequence (R, e_1, v_1, e_2, v_2, . . e_q, v_q) exists with N=$\{v_1, v_2, ...v_q\}$, E=$\{e_1, e_2, ... e_q\}$ and, for each j=1, 2,... q, h_{ej}=v_j and $T_{ej} \subseteq R \cup \{v_1, v_2, ..., v_{j-1}\}$.

In the following, with a little abuse of the notation, we will write $\mathtt{T}_R$=(R, e_1, v_1, e_2, v_2, . . . e_q, v_q), which amounts to represent a directed hypertree by means of one of its traversals. When specialized to hyperpaths, which are particular hypertrees, this definition coincides with the definition of hyperpath given by Torres and Araoz (1988).

Similarly, an undirected hypertree $\mathtt{T}_R$ can be defined as a sequence (R, e_1, v_1, e_2, v_2, . . . e_q, v_q), where $v_j \in (\{h_{ej}\} \cup T_{ej})$ and $(\{h_{ej}\} \cup T_{ej}) \setminus \{v_j\} \subseteq R \cup \{v_1, v_2, ..., v_{j-1}\}$, j=1, 2, ..., q. If $v \in N$, we shall denote by e_v the (unique) hyperarc incident to v in the unique hyperpath in $\mathtt{T}_R$ from R to v.

2.3. Spanning hypertrees

A *spanning hypertree of* $\mathtt{H}$=(V,E) is an undirected hypertree $\mathtt{T}_R$=(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, whereas the cardinalities of R and E_T may vary from hypertree to hypertree.

$\mathtt{T}_R$ =({5,6}, e_{10}, 4, e_4, 2, e_3, 1, e_6, 3) $\mathtt{T}_{R'}$ =({1}, e_2, 3, e_6, 4, e_3, 2, e_4, 5, e_{10}, 6)

Fig. 2

Given a hypergraph $\mathtt{H}$=(V,E) without isolated nodes, it is possible to construct a spanning hypertree for it by visiting $\mathtt{H}$ as described below. Starting with R=Ø and N=Ø, 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 $\mathtt{H}$, that is whenever there exists a hyperarc e such that $|(T_e \cup \{h_e\}) \setminus (R \cup N)| = 1$. When that is not possible anymore, the procedure considers the unexplored hyperarcs whose incidence set contains at least 2 unvisited nodes, and increments R by selecting one of these hyperarcs, and inserting into R all its unvisited nodes but one.

```
Procedure SPANNING HYPERTREE(H)
    begin
        R:= Ø; N:= Ø; E_T := Ø; j:= 0; E':= E; V':= V;
        for each e∈ E do Q_e:=T_e ∪ {h_e};
        repeat
            begin
                    select a∈ E' such that |Q_a|=min{|Q_e|: e∈ E' , Q_e≠∅};
                    select v∈ Q_a;
                    if |Q_a|=1 then
                            N := N ∪ {v}; V':= V' \{v}
                               else
                            R:= R ∪ (Q_a \ {v}); N:= N ∪ {v}; V':= V' \ Q_a;
                    j:= j +1; e_j:= a ; v_j:= v; E_T := E_T ∪ {a}; E':= E' \{a}
        until V'=Ø
    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 $O(\max\{size(H), n\log m\})$ time.

Proposition 2.1- *When applied to a hypergraph* H *having no isolated node*, SPANNING HYPERTREE *correctly finds a spanning hypertree for* H.

Given a hypergraph H with incidence matrix A, let $T_R=(V,E_T)$ be one of its spanning hypertrees and $(R, e_1, v_1, e_2, v_2, \ldots e_q, v_q)$ one of the traversals of T_R. By reordering the rows and the columns according to the orders induced by $(R, v_1, v_2, \ldots, v_q)$ and by $(e_1, e_2, \ldots, e_q, E_X)$ respectively, one can rewrite the matrix A in the following *canonical form* (relative to T_R)

$$A = \begin{array}{|c|c|} \hline B & C \\ \hline U & D \\ \hline \end{array} \quad \begin{matrix} \updownarrow R \\ \updownarrow N \end{matrix}$$

$$\underset{E_T}{\longleftrightarrow} \quad \underset{E_X}{\longleftrightarrow}$$

Remember that $E_X=E \setminus E_T$ is the set of the external hyperarcs and $N=\{v_1, v_2, \ldots, v_q\}$ is the set of the non-root nodes. Since $|N|=|E_T|$ by definition of hypertree, U is a square matrix. Note that, by construction, it is upper triangular with diagonal elements other than zero. Note also that each column of D contains one non-zero element at least. In fact, by definition, no external hyperarc can have its incidence set contained into the root set.

The spanning hypertree T_R which is associated with H may not be unique. In fact, consider one of the external hyperarcs of H, say e, and $v\in N$ be its incidence node of maximal index with respect to one of the orderings of the nodes which characterize T_R. This node exists because, by definition of spanning hypertree, $(T_e \cup\{h_e\})\not\subseteq R$. If we substitute $e_v\in E_T$ with e, we obtain a different spanning hypertree for H. It will be called *spanning hypertree induced by* e, and denoted by $T_R(e)$. Node v will be referred to as the *critical* node of $T_R(e)$, and will be denoted by v_e.

Hyperarc e thus induces on T_R a *hypercycle*, which is the union of the unique hyperpath from R to v in T_R, and of the unique hyperpath from R to v in $T_R(e)$. Hereafter, this hypercycle will be

denoted by $C(e)$. As an example, the spanning hypertree induced by e_9 on the spanning hypertree in the left of Figure 2 is $\mathtt{T}_R(e_9)=(\{5, 6\}, e_{10}, 4, e_4, 2, e_3, 1, e_9, 3)$.

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[2]. Both $f(T)$ and $d(R)$ can be determined in $O(size(\mathtt{H}))$ time by procedure *Flow* given next, which visits $\mathtt{T}_R$ in a bottom-up fashion, starting from the leaves. *Flow* has $\mathtt{H}$, $\mathtt{T}_R$, $d(N)$ and $f(X)$ as input data, while $d(R)$ and $f(T)$ are the outputs.

```
procedure Flow(H, T_R, d(N), f(X) d(R), f(T))
    begin
        for each v∈R do d(v):=0;
        for each e∈E_X do
            begin
                for each v∈(T_e ∪ {h_e}) do d(v):=d(v) - A_v(e)f(e)
            end {for};
        unvisited(v):=# of the hyperarcs of T_R incident into v, ∀v∈V;
        Queue:={leaves of T_R};
        while Queue≠Ø do
        begin
            v:=select(Queue);
            Queue:=Queue \{v};
            f(e_v ):= d(v) / A_v(e_v);
            * recall that A_v(e_v)=1 if v=h_{e_v} and A_v(e_v)= -μ_v(e_v) otherwise *
            for each w∈(T_{e_v} ∪ {h_{e_v}}) \ {v} do
                begin
                    d(w):=d(w) - A_w(e_v)f(e_v );
                    unvisited(w):=unvisited(w) - 1;
                    if unvisited(w)=1 and w∉R then Queue:=Queue ∪{w}
                end {for};
        end {while};
        for each v∈R do d(v):= -d(v)
end {Flow}.
```

Important particular cases are the followings:

i) $f(X)=0$ and $d(N)=\mathrm{u}_v$, where u_v denotes the unit vector with 1 as the component corresponding to root v, and 0 elsewhere; in this case, $\phi_v=(f(T), 0)$ is the *characteristic flow* of $\mathtt{T}_R$ relative to v, and $\rho_v=d(R)$ is the *root demand* induced by ϕ_v.

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

---

[2] If a is a n-vector and I is a subset of the set $\{1,2,..., n\}$, by $a(I)$ we denote the subvector of a containing all the components $a(i)$ with $i\in I$, so that, if $J=\{1,2,...,n\} \setminus I$, 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.

iii) $f(X)=u_e$ and $d(N)=0$, where u_e denotes 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 called 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, where, in the left hypergraph, the characteristic flow ϕ_5 and the corresponding root demand vector ρ_5 are illustrated, while, in the right hypergraph, the unit circulation ψ_{e5} together with δ_{e5} are shown. We assume $\mu_v(e)=1$ for all pairs (v, e) except for the pair $(6, e_6)$, for which $\mu_6(e_6)=2$.

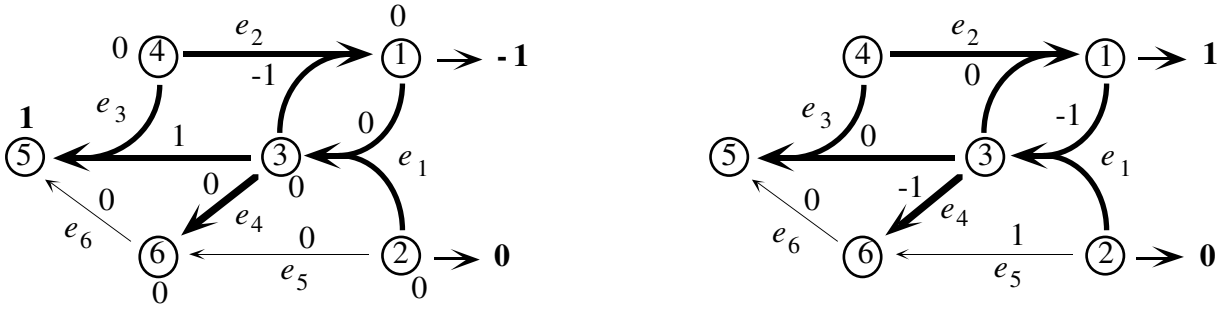

Fig. 3

The definition of characteristic flow can be extended in the following way. Let $\mathtt{T}_R=(R, e_1, v_1, e_2, v_2, \ldots, e_q, v_q)$ be one of the spanning hypertrees of $\mathtt{H}$, $e\in E_X$ be one of the corresponding external hyperarcs, and v be the critical node of $\mathtt{T}_R(e)$. Then, similarly to the definition of the characteristic flow of $\mathtt{T}_R$, we can define ϕ_v^e as the *characteristic flow of* $\mathtt{T}_R(e)$ relative to v, and ρ_v^e as the corresponding root demand vector. Clearly, these flows and the relative root demand vectors can be computed directly on the hypergraph by means of a procedure similar to *Flow*.

Let us evidentiate now a property of the characteristic flows that will be useful in the following. Let $k\in\{1,\ldots, q\}$; the characteristic flow ϕ_{vk} as computed by procedure *Flow* satisfies the following relations:

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

for the same reason, if $e\in E_X$ and $v_e=v_k$ we have that:

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

Since both $\phi_{v_e}^e$ and ϕ_{v_e} determine a flow unbalance of one unit at $v_e\in N$ and make balanced each other non-root, the flow $[\phi_{v_e}^e - \phi_{v_e}]$ is a circulation, which satisfies:

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

By recalling that ψ_e is the particular circulation, defined on the hypercycle $C(e)$, having a flow of value 1 on hyperarc e and a flow of value 0 on all the other external hyperarcs, the following properties hold true:

$$\psi_e = A_{v_e}(e)[\phi_{v_e}^e - \phi_{v_e}] \qquad \text{and} \qquad \delta_e = A_{v_e}(e)[\rho_{v_e}^e - \rho_{v_e}]. \tag{3.1}$$

Thus, the circulation ψ_e is proportional to the difference between the characteristic flow of $\mathtt{T}_R(e)$ relative to the critical node v_e and the characteristic flow of $\mathtt{T}_R$ relative to the same node. In other words, ψ_e and δ_e are respectively the unique circulation, and the unique root demand vector, which are needed in order to push a flow of value $A_{v_e}(e)$ from the roots to v_e along $\mathtt{T}_R(e)$, and a flow of value $-A_{v_e}(e)$ from the roots to v_e along $\mathtt{T}_R$.

3.2. Potentials on spanning hypertrees

Let us now provide a dual version of procedure *Flow*. Consider the cost $c(e)$ which is associated with hyperarc e, and assume that each node v is assigned a potential $\pi(v)$; according to the Linear Programming terminology, we will call *reduced cost* of e the quantity

$$c(e) + \sum_{v \in T_e} \mu_v(e)\pi(v) - \pi(h_e).$$

Observe that, 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 is equal to zero on each hyperarc of the hypergraph. The following procedure, which can be considered as a dual version of the procedure *Flow*, finds the vectors $\pi(N)$ and $c(X)$ directly on the hypergraph by means of a visit of $\mathtt{T}_R$ from the roots to the leaves. The procedure works in O(*size*(H)) time.

```
procedure Potential(H, T_R, c(T), π(R), c(X), π(N))
begin
    for each e∈E_X do c(e):=0;
    for each v∈R do
    begin
        for each e∈E such that v∈(T_e ∪{h_e}) do c(e):=c(e) -A_v(e)π(v);
    end {for};
    unvisited(e):=# of the nodes of N incident to e, ∀ e∈E;
    Queue:={e: e∈T_R and unvisited(e)=1};
    while Queue ≠Ø do
    begin
        * select any element of Queue *
        e:=select(Queue);
        Queue :=Queue \{e};
        let v be the unique unvisited node of N incident to e;
        * visit node v *
        π(v):= c(e) / A_v(e) ;
        for each e*∈E\{e} s.t. v ∈(T_e* ∪ {h_e*}) do
        begin
            c(e*):=c(e*) - A_v(e*)π(v );
            unvisited(e*):=unvisited(e*) - 1;
            if unvisited(e*)=1 and e*∉E_X then Queue:=Queue ∪{e*}
        end {for}
    end {while};
    for each e∈E_X do c(e):=-c(e);
end {Potential}.
```

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

3.3. The Root matrix

Let $\mathtt{T}_R=(V,E_T)$ be one of the spanning hypertrees of H, rooted at R, and $A = \begin{bmatrix} B & C \\ U & D \end{bmatrix}$ be the incidence matrix of H in canonical form with respect to $\mathtt{T}_R$. We call *root matrix* of H the $|R| \times |E_X|$ matrix $A_R=(C - BU^{-1}D)$. We will see later that such a matrix plays a crucial role in the solution of the minimum cost hyperflow problem.

Like in the matrix C, 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 the external hyperarc e, and by $A_R(v,*)$ the row of A_R corresponding to the root v. 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 $Flow(\mathtt{H}, \mathtt{T}_R, d(N), f(X), d(R), f(T))$ satisfy the following relations:

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

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

Proposition 3.2. - For each $v \in R$, it is $A_R(v,*) = \gamma_v$, where γ_v, 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 v 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 $Potential(\mathtt{H}, \mathtt{T}_R, c(T), \pi(R), c(X), \pi(N))$:

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

From the definition of γ_v, we have $\gamma_v=\mathrm{u}_v(C-BU^{-1}D)$. 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

Given a hypergraph H, let us consider a sub-hypergraph with n nodes and n hyperarcs, and its incidence matrix M. If M is non singular, then the sub-hypergraph cannot have isolated nodes, otherwise M should have a zero row. As a consequence, there exists a spanning hypertree, say $\mathtt{T}_R$, for it. Since $|R|=|E_X|$, M can be converted in canonical form, as described in Section 2, with C being a square matrix. As a result, the corresponding root matrix, let it be M_R, is a square matrix as well.

The following theorem provides the basis characterization in terms of spanning hypertrees. As with a wealth of earlier research in the context of network flows with side constraints (Kennington and Helgason, 1980), the key is a partition of the basis matrix columns into those corresponding to a spanning hypertree, and an external part which is suitably non singular.

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

Proof:

W.l.o.g. assume M to be in canonical form relative to $\mathtt{T}_R$. 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, since U is an upper triangular matrix with non zero diagonal elements. 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 v, and $M_R^{-1}(e,v)$ be its element in the row corresponding to the external hyperarc e. Since

$$M_R M_R^{-1}(*,v)= \sum_{e\in E_X} \delta_e M_R^{-1}(e,v)=\mathrm{u}_v \ , \tag{4.1}$$

from Proposition 3.1 it follows that:

- the column of M_R^{-1} corresponding to the root v gives the (unique) flow values to be imposed on the external hyperarcs in order to obtain a circulation with demand equal to 1 at root v and 0 at all the other roots. Similarly, by considering the row of M_R^{-1} corresponding to the external hyperarc e

$$M_R^{-1}(e, *)M_R = \sum_{v\in R} M_R^{-1}(e,v)\gamma_v =\mathrm{u}_e \ , \tag{4.2}$$

then from Proposition 3.2 it follows that:

- the row of M_R^{-1} corresponding to the external hyperarc e gives 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 the root matrix and of its inverse, let us consider the hypergraph in Fig. 4, where $\mu_6(e_6)=2$ and $\mu_v(e)=1$ otherwise.

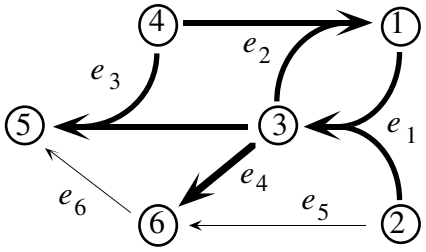

Fig. 4

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

	e_1	e_2	e_3	e_4	e_5	e_6
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

+1	-1
0	-2

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

ϕ_6 =(+1,0,0,+1,0,0), $\phi_6^{e_5}$ =(0,0,0,0,+1,0), ρ_6 =(-1,-1), $\rho_6^{e_5}$ =(0, -1).

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

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

Now, consider the spanning hypertree induced by e_6, $\mathtt{T}_R(e_6)$, and its critical node, i.e. 6. The characteristic flow $\phi_6^{e_6}$ and its corresponding root demand vector are:

$$\phi_6^{e_6} = (0,-1/2,+1/2,0,0,-1/2) \qquad \rho_6^{e_6} = (-1/2,\ 0).$$

As a consequence, since $M_6(e_6) = -2$, we get:

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

The same result holds for the circulations:

ψ_{e_5} =(-1,0,0,-1,1,0), δ_{e_5} =(+1,0), ψ_{e_6} =(2,1,-1,2,0,1), δ_{e_6} =(-1,-2).

As far as the rows of the root matrix are concerned, being R={1,2} we have:

if $\pi(R)=u_1$ $\pi(N)$ =(1,0,1,1) γ_1 =(1,-1)

if $\pi(R)=u_2$ $\pi(N)$ =(1,-1,0,1) γ_2 = [0,-2].

Let us consider now M_R^{-1}:

$$\begin{bmatrix} +1 & -1/2 \\ 0 & -1/2 \end{bmatrix}$$

If we consider its first column, relative to the root 1, from (4.1) we get:

$$(+1)\delta_{e5} = (+1)\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}.$$

If we consider the second column, relative to the root 2, we obtain:

$$(-1/2)\delta_{e5} + (-1/2)\delta_{e6} = (-1/2)\begin{bmatrix}1\\0\end{bmatrix} + (-1/2)\begin{bmatrix}-1\\-2\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}.$$

Similarly, by considering the rows of M_R^{-1}, from (4.2) it is:

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

$$(-1/2)\gamma_2 = (-1/2)\begin{bmatrix}0\\-2\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}.$$

5. Special cases

5.1. Generalized flows

If H is a standard graph, i.e. $|T_e| \leq 1 \ \forall e \in E$, then the hypergraph network flow model becomes the generalized network flow model (Ahuja, Magnanti and Orlin, 1993). In this problem, the basic solutions are characterized in terms of good augmented tree and of good augmented forests. An *augmented tree* is a tree, together with a possible additional arc (v,w), called the *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 exists in the augmented tree. An augmented tree is referred to as a *good augmented tree* if either it is a tree, or the unique cycle determined by the extra arc (*extra cycle*) is "lossy" or "gainy", i.e. it is possible to create a deficit or an excess of flow at any node along the cycle. To be more formal, let C be the extra cycle, let us fix an orientation along it, and denote by C^+ and C^-, respectively, the set of the forward arcs and the set of the backward arcs of C with respect to such an orientation. Then, define the *cycle multiplier* of C as follows:

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

If 1 unit of flow is sent along C starting from a node v, then $\mu(C)$ units of flow return to this node. As a consequence, if $\mu(C) > 1$ we create an excess at v; while if $\mu(C) < 1$ we create a deficit at v. In the first case ($\mu(C) > 1$), C is referred to as a *gainy cycle*, while in the second one ($\mu(C) < 1$), C is said to be a *lossy cycle*.

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* H *be a generalized graph with n nodes and n arcs. Then* H *is a basis graph if and only if it is a good augmented forest.*

Now, let H be a good augmented forest. First of all, note that H is a particular spanning hypertree, which is a collection of node-disjoint trees each one having the root set either empty, or singleton; the set of the external hyperarcs is the set of the extra arcs. The incidence matrix, M, of H is thus a block-diagonal matrix, with square blocks corresponding to the components of H, and H is a basis graph if and only if each block of M is non singular. Consider now one of the components of H, let it be T, and assume that the root set of T is a singleton, e.g. $\{r\}$ (if its root set is empty, then the block of M corresponding to T is clearly non singular). Let *imbalance*(r) be the imbalance at r when the flow on the extra arc (v,w) of T is set to value 1. By definition of gainy and of lossy cycle, the extra cycle of 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 (1x1) root matrix which is associated with T, and so *imbalance*(r) is not zero if and only if the root matrix of T is non singular. By combining the two properties, Theorem 5.1 can be reformulated as follows:

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

Since Theorem 5.2 is a special case of Theorem 4.1, we have that 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 by Jeroslow et al. (1992) under the hypothesis that a so called *gainfree property*, which limits the existence of directed cycles, holds.

Given a gainfree Leontief substitution flow problem, the *support hypergraph* of a basic feasible flow f is the hypergraph containing each hyperarc $e\in E$ such that $f(e)>0$, and each node which is incident in such hyperarcs.

Theorem 5.3 [Jeroslow, Martin, Rardin and Wang, 1992]**-** *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* H *be a hypergraph with n nodes and n hyperarcs, and M be its incidence matrix.* H *is a directed hypertree* T_R *with* $R=\emptyset$ *if and only if* 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 solution, $\mathtt{H}_f=(V_f, E_f)$ be its support hypergraph and M_f be the correspondent incidence matrix; denote also with $f_+>0$ the vector having the positive components of f and with b_f the vector having the components of b relative to the nodes of V_f. Note that M_f is a submatrix of the basis matrix M corresponding to f, and that $M_f f_+=b_f \geq 0$.

Observe firstly that each row of M_f has at least one positive element, since otherwise the row would have only nonpositive elements and at least one negative element, and this is absurd since $f_+>0$ and $M_f f_+\geq 0$. It follows that $|V_f|=|E_f|$; in fact $|V_f|\leq|E_f|$, because M_f is pre-Leontief with at least one positive element in each row, and if $|V_f|<|E_f|$ then $\text{rank}(M_f)\leq|V_f|<|E_f|$ so that $\text{rank}(M)<|V|$ and this is absurd since M is a basis matrix.

Since M_f is a square and pre-Leontief matrix with at least one positive element in each row, then it necessarily has exacly one positive element in each row and in each column. In addition, by Theorem 5.3 $\mathtt{H}_f$ has no directed cycles. By Lemma 5.1 it follows that $\mathtt{H}_f$ is a directed hypertree with root set empty. ♦

6. The hypergraph network simplex algorithm

In this section, we shall propose an adaptation of the Network simplex algorithm to hypergraphs, which makes use of the basis characterization and of the procedures on hypergraphs presented in Sections 3 and 4. This algorithm, which implements a plain primal simplex method, is intended only as an exemplification of how the results presented so far can be used in order to devise algorithms based on labelling techniques to solve the hypergraph flow problem, as is usually done with the network flow problem. For applications to linear production models we refer to the paper by Cambini, Gallo and Scutellà (1992).

6.1. Solving linear systems

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

Let us consider the first system, which can be interpreted as the problem of finding on the sub-hypergraph whose incidence matrix is M a flow f which satisfies a given demand vector $\bar{b}$. The following result states 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* $\mathtt{H}$ *be a basis hypergraph with incidence matrix* M, $\mathtt{T}_R$ *be a spanning hypertree for* $\mathtt{H}$, λ *be the vector* $M_R^{-1}(\bar{b}^-(R) - \sum_{v\in N} \bar{b}(v)\rho_v)$, *and* f *be the unique solution to the linear system* $Mf=\bar{b}$; *then*:

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

Proof:

By simple manipulations we get:

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

From (3.2), by recalling the definition of characteristic flow and of the corresponding root demand vector, i.e. the case $f(X)=0$, and the definition of circulation, i.e. the case $\bar{b}(N)=0$, one gets $f=(f(T),f(X))=\sum_{v\in N}\bar{b}(v)\phi_v+\sum_{e\in E_X}\lambda(e)\psi_e$, where $\lambda=M_R^{-1}(\bar{b}(R)-\sum_{v\in N}\bar{b}(v)\rho_v)$. The case relative to $R=\emptyset$ 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 $O(\max\{|R|^2, size(H)\})$ 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(H, T_R, M_R^-1, b⁻, f)
begin
    Flow(H, T_R, b⁻(N), 0, d(R), f(T));
    f(X):=M_R^-1(b̄(R) - d(R));
    Flow(H, T_R, b⁻(N), f(X), b̄(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, i.e. $d(R)$ (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 T_R done in the procedure *Flow*.

Also the linear system $\pi M=\bar{c}$ can be solved directly on the hypergraph, following an approach which can be viewed as the dual of the previous one. Let π_0 and c_0 denote the potential vector and the cost vector on the external hyperarcs returned by *Potential* if $\pi(R)=0$, and let π_1 denote 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* H *be a basis hypergraph with incidence matrix M and* T_R *be a spanning hypertree for* H. *The unique solution to the linear system* $\pi M = c^-$ *is*:

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

Proof:

With the usual notations, assuming $R \neq \emptyset$ (otherwise the proof would be trivial), one has

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

and, from (3.3):

$$\pi = (0, \bar{c}(T)U^{-1}) + (\bar{c}(X) - \bar{c}(T))U^{-1}D)M_R^{-1}(I, -BU^{-1}) = \pi_0 + \pi_1,$$

where $\pi(R)=(\bar{c}(X)-c_0)M_R^{-1}$. ♦

As for the primal case, also the linear system $\pi M=\bar{c}$ can thus be solved directly on the hypergraph. In fact, given M_R^{-1}, the following algorithm, which uses procedure *Potential* as a subroutine, allows one to solve the system in O(max($|R|^2$, *size*(H))) time. We assume $R \neq \emptyset$; otherwise π is yielded by the procedure *Potential* alone.

```
procedure Dual(H, T_R, M_R^-1, c^-, π)
    begin
        Potential(H, T_R, c^-(T), 0, c_0, π_0(N));
        π(R):=(c̄(X)-c_0)M_R^-1;
        Potential(H, T_R, c^-(T), π(R), c̄(X), π(N))
end {Dual}.
```

6.2. Initial spanning hypertree

In order to obtain an initial feasible basis, we can introduce artificial hyperarcs of sufficiently large cost C and infinite capacity in the following way. For each node $v \in V$, if v is a supply node (i.e. $b(v) < 0$), let us introduce the artificial hyperarc $(v, \emptyset)$; if v is a demand or a transshipment node (i.e. $b(v) \geq 0$), let us introduce the artificial hyperarc $(\emptyset, v)$. The set of the artificial hyperarcs so defined induces a spanning hypertree with root set empty. So, by Theorem 4.1, it is a basis spanning hypertree. It is easy to see that the corresponding flow solution is feasible. This primal solution and its corresponding dual solution can be computed in an immediate way using procedures *Flow* and *Potential* alone, since the inverse of the root matrix is the empty matrix by construction. As will be shown, the inverse of the root matrix can be updated directly on the hypergraph during each pivot operation. So, no explicit use of inversion techniques is needed. Clearly, at termination, no solution with a positive flow on any artificial hyperarc will be optimal unless the hypergraph network flow problem is infeasible.

6.3. Optimality testing and entering hyperarc

Let M be the current feasible basis, corresponding to a sub-hypergraph H* having T_R as a spanning hypertree. 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.

To determine whether the current basis is optimal, we check whether the reduced cost is ≥ 0 (≤ 0) for each hyperarc which is out of the basis at its lower bound (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, and forces it into the spanning hypertree.

6.4. Leaving hyperarc

Now we show how the leaving hyperarc can be determined. Consider the case in which $f(e')=0$. Let $\overline{f}$ 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. $\overline{f}$ is the solution to the system

$$M f = - A(*, e'),$$

which can be determined by means of procedure *Primal*. The hyperarcs whose corresponding component in $\overline{f}$ is 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 on it. In practice we compute the vector s with components

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

Then, $e^*=\text{argmin}\{s(e): e \in \{H^* \text{ with } \overline{f}(e) \neq 0\} \cup \{e'\}\}$ 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

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

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

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

In order to study how the basis changes, different cases have to be considered depending on the position of the leaving hyperarc, e^*, and of the entering hyperarc, e', in the new basis. To simplify the presentation, in the following we shall assume that the nodes are 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 $\overline{e}$ exists which can replace e^* in the hypertree, i.e. the critical node of $T_R(\overline{e})$, say v, is such that $e_v=e^*$ (according to the definitions given in section

2), then we can exchange e^* and $\overline{e}$ (i.e. $\overline{e}$ becomes a tree hyperarc and e^* becomes external), and easily reduce to the previous case (***case 2.a***).

If no such hyperarc exists (***case 2.b***), let $\overline{v}$ be the non-root of maximal index in $(T_{e^*} \cup \{h_{e^*}\})$. Two cases are possible: either $\overline{v}$ 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 $R' = R \cup \{\overline{v}\}$, and e' enters as an external hyperarc unless $(T_{e'} \cup \{h_{e'}\}) \subseteq R$; in this last 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 below for the hypergraph in Fig. 3, where $\mathtt{T}_R = (\{1,2\}, e_1, 3, e_2, 4, e_3, 5, e_4, 6)$, $R = \{1, 2\}$ and $E_X = \{e_5, e_6\}$.

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

case 1.a) $(T_{e'} \cup \{h_{e'}\}) \not\subseteq R$

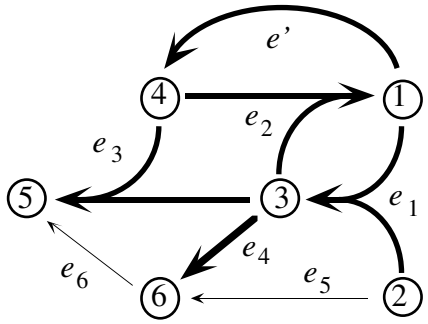

Fig. 5

$R' = R = \{1, 2\}$ $\quad E'_X = \{e_5, e_6\} \setminus \{e_5\} \cup \{e'\} = \{e_6, e'\}$

case 1.b) $(T_{e'} \cup \{h_{e'}\}) \subseteq R$

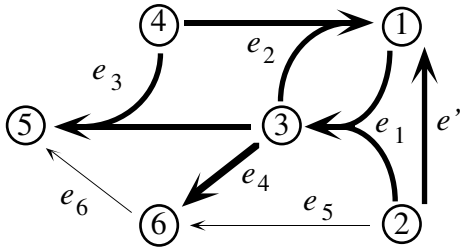

Fig. 6

$R' = R \setminus \{1\} = \{2\}$ $E'_X = E_X \setminus \{e_5\} = \{e_6\}$ $\mathtt{T}'_R = (\{2\}, e', 1, e_1, 3, e_2, 4, e_3, 5, e_4, 6)$

2) $e^* \in \mathtt{T}_R$

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 $\mathtt{T}'_R = (\{1, 2\}, e_1, 3, e_2, 4, e_4, 6, e_6, 5)$; we can then apply the transformations in *1)*.

case 2.b) $e^* = e_2$ $(\overline{v} = 4)$

case 2.b.1) $e' = (\{1,3\}, 4)$

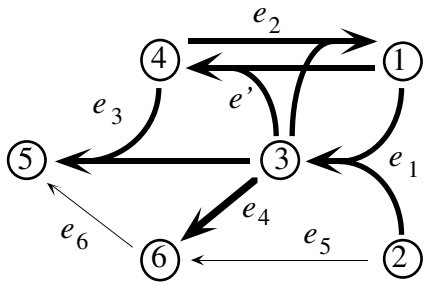

Fig. 7

$E'_X = E_X = \{e_5, e_6\}$ T $'_R = (\{1, 2\}, e_1, 3, e', 4, e_3, 5, e_4, 6)$

case 2.b.2) - $e' = (\{1,4\}, 6)$ $((T_{e'} \cup \{h_{e'}\}) \not\subseteq R)$

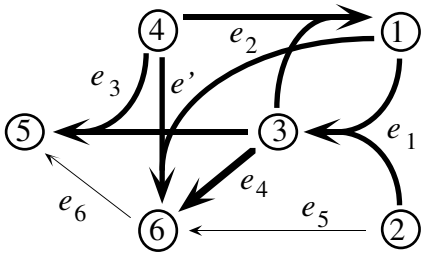

Fig. 8

$E'_X = E_X \cup \{e'\}$ $R' = R \cup \{4\}$ T $'_R = (\{1, 2, 4\}, e_1, 3, e_3, 5, e_4, 6)$

- $e' = (1, 2)$ $((T_{e'} \cup \{h_{e'}\}) \subseteq R)$

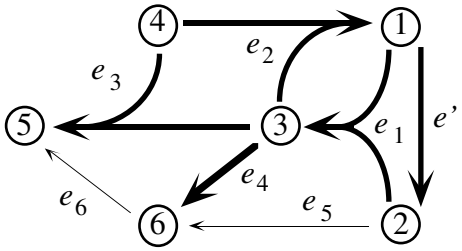

Fig. 9

$E'_X = E_X$ $R' = R \setminus \{2\} \cup \{4\} = \{1,4\}$ 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 $M_R^{-1}(v,e)$ be the element of M_R^{-1} in the row corresponding to the external hyperarc e and in the column corresponding to the root v. Let ψ_v be the unique circulation which induces a demand 1 at v and a demand 0 at all the other roots, and which has value 0 on all the hyperarcs not belonging to the basis hypergraph H*. Due to the characterization of M_R^{-1} given in section 4 it is $\psi_v(e) = M_R^{-1}(v,e)$, $\forall e \in E_X$.

Clearly, for all $v \in R$, the new vector $\overline{\psi}_v$ with

$$(6.1)\quad \bar{\psi}_v(e)=\begin{cases}\psi_v(e) - \dfrac{\psi_v(e^*)}{\bar{f}(e^*)}\,\bar{f}(e), & e\in H^*,\\ -\dfrac{\psi_v(e^*)}{\bar{f}(e^*)}, & e=e',\\ 0, & \text{otherwise,}\end{cases}$$

is a circulation which induces at the roots the same demand as ψ_v 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, $M'_R{}^{-1}(v,e)=\bar{\psi}_v(e)$. In the case in which a new root, say $\bar{v}$, enters the root set, in order to apply the updating formula (6.1), the vector $\psi_{\bar{v}}$ needs to be computed. Since, with the notations previously introduced, it is $\psi_{\bar{v}}=M^{-1}u_{\bar{v}}$, it is possible to perform such a computation by means of the procedure *Primal*.

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, will be the object of a forthcoming paper.

Here we present the results of a preliminary experimentation performed on a workstation HP 9000/700. For these experiments random hypergraphs with (0,±1) incidence matrices have been used. In order to assess the effectiveness of our approach, a comparison with one of the best state-of-the art LP codes, CPLEX (Version 2.1), has been done. The results obtained, as reported in the graphs below, seem to be quite promising, and, although still preliminary, they suggest that our approach becomes more and more efficient 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 as the number of the arcs grows (see Fig.10, where n=800 and k=8). A similar behavior characterizes our implementation when both the number n of the nodes and the number m of the hyperarcs are fixed, while the maximum cardinality of the tail sets is increased (see Fig. 11 and 12, where n=400, m=6000 and n=800, m=2000, respectively). In the figures, the running times are in seconds, and each one is the average CPU time over 5 experiments in which only the seed of the hypergraph generator has been changed.

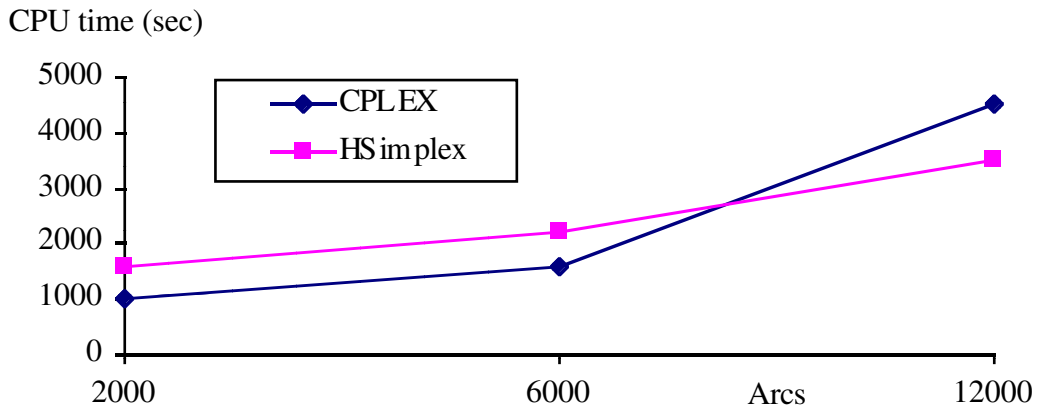

Fig. 10

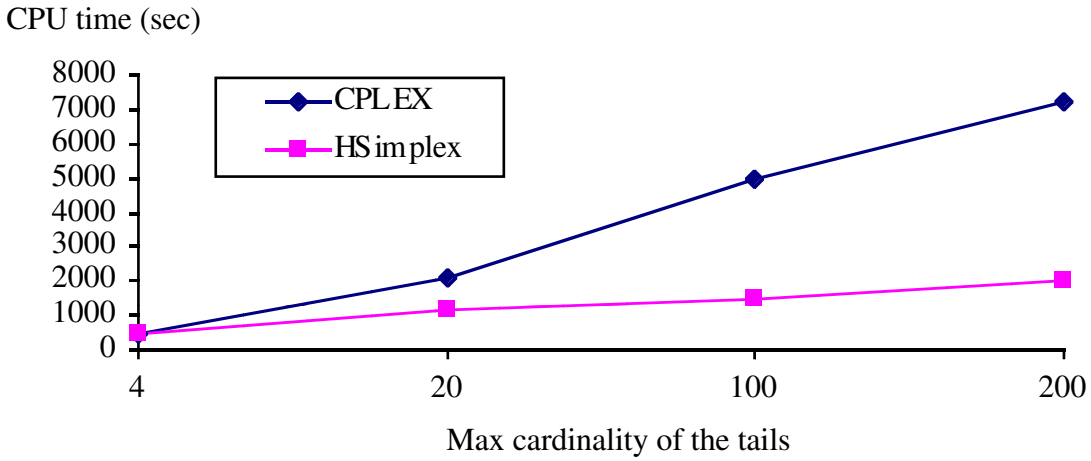

Fig. 11

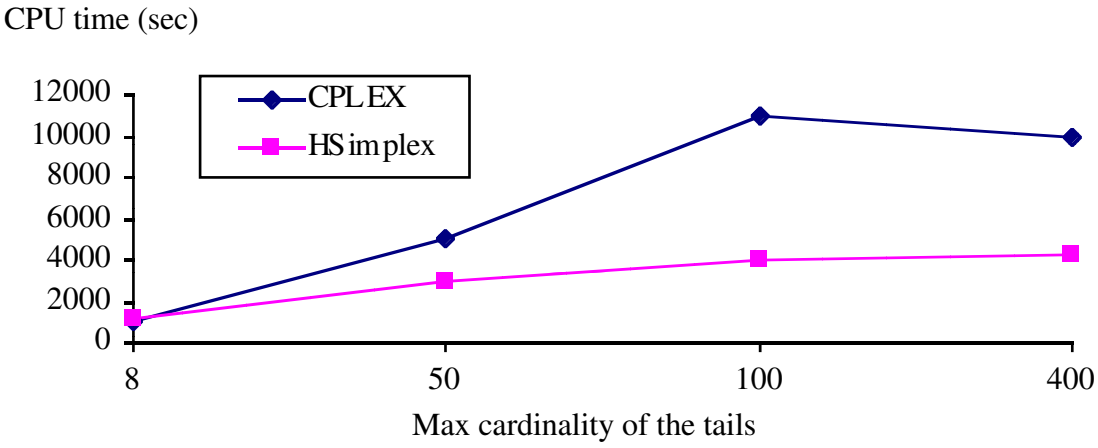

Fig. 12

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 (Cambini, Gallo and Scutellà, 1992), with reference to a particular application.

Acknowledgements

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

References

R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows*: *theory*, *algorithms*, *and applications* (Prentice Hall, N.J, 1993).

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

R. Cambini, G. Gallo and M.G. Scutellà, "Minimum Cost Flows on Hypergraphs", TR-1/92, Dipartimento di Informatica, University of Pisa (Pisa, 1992).

G. Gallo, G. Longo, S. Nguyen and S. Pallottino, "Directed hypergraphs and applications", *Discrete Applied Mathematics* 40 (1992).

G. Gallo and G. Rago, "A hypergraph approach to logical inference for datalog formulae", TR-28/90, Dipartimento di Informatica, University of Pisa (Pisa, 1990).

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

Kennington, J.L. and R.V. Helgason, *Algorithms for Network Programming* (Wiley Interscience, Chapter 7, 1980).

Koehler, Whinston and Wright, *Optimization over Leontief Substitution Systems* (North Holland, 1975).

E. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart and Winston eds, 1976).

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

A.F. Veinott Jr., "Extreme Points of Leontief Substitution Systems", *Linear Algebra and Its Application* 1 (1968) 181-194.