Algorithms for multiplayer multicommodity flow problems

**Attila Bernáth · Tamás Király · Erika
Renáta Kovács · Gergely Mádi-Nagy ·
Gyula Pap · Júlia Pap · Jácint Szabó ·
László Végh**

Abstract We investigate the MULTIPLAYER MULTICOMMODITY FLOW PROB-
LEM (MMFP): several players have different networks and commodities over
a common node set. Pairs of players have contracts where one of them agrees
to route the flow of the other player (up to a given capacity) between two
specified nodes. In return, the second player pays an amount proportional to
the flow value.

We show that the social optimum can be computed by linear program-
ming, and we propose algorithms based on column generation and Lagrangian
relaxation. In contrast, we prove that it is hard to decide if an equilibrium
solution exists, although some natural conditions guarantee its existence.

Keywords optimization · multicommodity flows · equilibria

1 Introduction

The DIRECTED MULTICOMMODITY FLOW PROBLEM is a well-studied opti-
mization problem with extensive literature. The fractional version, which is
relevant to the topic of this paper, has a polynomial-size LP formulation and
therefore it is solvable in polynomial time. However, the simplex algorithm in
itself is inefficient due to the large size of the LP, so several improvements and
other algorithmic approaches have been proposed, see e.g. [3,6,2]. Two general
approaches that are applicable to this problem are column generation, which
was already proposed by Ford and Fulkerson [5], and Lagrangian relaxation,
which is used for example in [2].

In this paper we investigate a generalization of the multicommodity flow
problem that we call the MULTIPLAYER MULTICOMMODITY FLOW PROBLEM
(MMFP). It involves several players, each having different networks and com-
modities over a common node set. The input includes contracts between pairs

MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös Loránd
University, Budapest. E-mail: tkiraly@cs.elte.hu

of players; each contract obliges one player to route the flow of the other player between two specified nodes (up to a given capacity). In return, the other player pays an amount proportional to the flow value. We also allow a multiplier applied to the amount of flow to be routed by the second player, which may model e.g. the increase of data size due to conversion between networks.

Multiplayer flows have already been studied in the literature in the context of cooperative games, but the usual setting is to have a single network where each player owns a subset of the arcs, and other players are allowed to buy a fraction of the arc capacity [4,1]. Our network model is more general in the sense that players can choose how to route demands generated by contracts with other players.

The flow multipliers associated with the contracts add another layer of generalization that is subtle in the sense that the interpretation of feasible solutions in terms of actual transportation of commodities becomes tricky. We show in Section 2 that such an interpretation is always possible if the flow multipliers are at least 1, using the notion of *realization of a path* which corresponds naturally to the transportation of a commodity. In contrast, multipliers smaller than 1 may result in feasible solutions that do not have such an interpretation.

In the rest of the current section we give the necessary definitions, including the notion of social optimum, safe instance and equilibrium solution. It is shown in Section 2 that the problem of finding a social optimum can be solved in polynomial time using linear programming, and we give a "common sense" interpretation of feasible solutions in terms of realizations of paths and acyclic solutions. The aim of Section 3 is to extend the column generation and Lagrangian relaxation methods for multicommodity flows to MMFP. In Section 4, we prove using the Kakutani Fixed Point Theorem that an equilibrium solution always exists in a safe instance, while in general it is NP-complete to decide its existence.

1.1 Definitions and notation

Given a set of players I, each player has a digraph $D^i = (V, A^i)$ (with disjoint arc sets) with costs $c : \cup_{i \in I} A^i \to \mathbb{R}_+$, capacities $u : \cup_{i \in I} A^i \to \mathbb{R}_+$, and *normal* demands: a flow of size d_j^i has to be sent from s_j^i to t_j^i ($j \in J^i$). There is a subset of arcs $B^i \subseteq A^i$ called *contractual arcs*, where the player has a contract with another player requiring the other player to route the flow between the two endnodes of the arc in her own network for a specified price (proportional to the amount of flow). The arcs in $A^i \setminus B^i$ are called *normal arcs*. Let $A = \cup_{i \in I} A^i$ and $B = \cup_{i \in I} B^i$.

The following additional values are given for each contractual arc $a \in B_i$. i_a is the player that has to route the flow going through a. $f_a : [0, u_a] \to \mathbb{R}_+$ is a continuous monotone increasing function that gives the amount of flow to be routed, i.e. if the flow value of $a = uv$ is x, then player i_a has to route a flow of

size $f_a(x)$ from u to v. This function may model the increase of data size due to the conversion between networks. Usually we will consider the linear case $f_a(x) = d_a x$. We always assume that $d_a \geq 1$ (or, in general $f_a(x) \geq x$ for any $x \in \mathbb{R}_+$) holds for every $a \in B$. The resulting demand in the network of i_a is called a *contractual demand* for player i_a. The identifier of this demand is k_a. Let K^i be the set of identifiers of the contractual demands for player i; then $k_a \in K^{i_a}$ for $a \in B^i$. We assume that all the identifier sets $J^i, K^i (i \in I)$ are disjoint and let $J = \cup_{i \in I} J^i$ and $K = \cup_{i \in I} K^i$. Note that $a \mapsto k_a$ is a bijection between B and K.

A contractual arc a has a *contract price*, denoted by p_a. If the flow value of $a = uv$ is x, then player i has to pay $p_a x$ to player i_a in exchange for sending the flow.

It may seem confusing at first that a contractual arc has both a contract price p_a and a cost c_a. A possible interpretation is that the contract price determines the payment to another player, while the cost of a contractual arc represents the cost of transferring the flow to another network. We do not make any restrictions on the costs, but it is probably a natural assumption that costs of contractual arcs are much lower than costs of normal arcs.

1.2 Feasible and equilibrium solutions

In a *feasible solution* each player has a multicommodity flow (one commodity for every $j \in J^i \cup K^i$) that satisfies all normal and contractual demands. Note that here the contractual demands depend on the multicommodity flows of the other players.

A *social optimum solution* is a feasible solution which is optimal for the cost function c. The contract prices play no role in this definition.

Let us motivate the assumption $d_a \geq 1$ ($a \in B$). Consider the following two-player example where $V = \{s, t\}$. Let $A^1 = B^1 = \{a = st\}$ and $A^2 = B^2 = \{b = st\}$ (i.e. both players have a contract with the other) and the only normal demand is for player 1 who has to ship 1 unit from s to t (i.e. let $J^1 = \{j_1\}$ with these parameters and let $J^2 = \emptyset$). Let the capacities be given as $u_a = 2$ and $u_b = 1$ and the multipliers as $d_a = 1/2$ and $d_b = 1$. The following is a feasible solution: the first player ships 2 units from s to t on arc a which is the sum of two flows x^{j_1} (satisfying the normal demand $j_1 \in J^1$) and x^{k_b} (satisfying the contractual demand k_b to be described later). The flow on arc a induces a contractual demand of value 1 for player 2, since $d_a = 1/2$, which is shipped on arc b, i.e. $x^{k_a}(b) = 1$. This in turn generates a contractual demand of value 1 for player 1 (since $d_b = 1$). Clearly, this is a feasible solution of the problem, although from a practical point of view it does not make sense (for example this solution does not have a physical routing). This example shows that the assumption $d_a \geq 1$ (or, in general $f_a(x) \geq x$ for any $a \in B$ and $x \in \mathbb{R}_+$) is indeed natural.

We say that an instance of MMFP is a *safe instance* if each player i has a feasible solution for the *standard* multicommodity flow problem in D^i (in-

cluding the arcs in B^i!) for all her normal demands *plus* all her contractual demands set at their maximum (for a contractual arc $a = uv$, the maximal contractual demand for player i_a is $f(u_a)$ from u to v).

An *equilibrium solution* is a feasible solution which is a minimum cost multicommodity flow restricted to each player's network for the player's normal and contractual demands induced by the solution. Here the cost of arc $a \in A^i \setminus B^i$ is c_a, while the cost of arc $a \in B^i$ is $c_a + p_a$. It may seem strange that we do not consider the contract prices that a player *receives*; this is because these received payments depend only on the multicommodity flows of the *other* players, so they are irrelevant to whether a player's multicommodity flow is minimum cost or not in her own network (they add a constant to the cost if the flows of the other players are fixed).

It is important to note that this notion of equilibrium is stronger than the usual Nash-equilibrium, because here players cannot improve their objective even with strategy changes that render the whole solution infeasible.

2 LP formulation and interpretation

2.1 LP formulation

In case of $f_a(x) = d_a x$, there are two main LP models for finding the social optimum: the arc-flow and the path-flow formulations. The first one can be written as follows.

$$\min \sum_{i \in I} \sum_{a \in A^i} c_a \left(\sum_{j \in J^i} x_a^j + \sum_{k \in K^i} x_a^k \right) \tag{1}$$

$$\sum_{j \in J^i} x_a^j + \sum_{k \in K^i} x_a^k \le u_a \qquad \forall a \in A^i, \ \forall i \in I, \tag{2}$$

$$N^i x^j = d_j^i \delta^j \qquad \forall j \in J^i, \ \forall i \in I, \tag{3}$$

$$N^{i_a} x^{k_a} = d_a \left(\sum_{j \in J^i} x_a^j + \sum_{k \in K^i} x_a^k \right) \delta^a \qquad \forall a \in B^i, \ \forall i \in I, \tag{4}$$

$$x_a^j \ge 0 \qquad \forall a \in A^i, \ j \in J^i, \ \forall i \in I, \tag{5}$$

$$x_a^k \ge 0 \qquad \forall a \in A^i, \ k \in K^i, \ \forall i \in I. \tag{6}$$

Here, N^i is the network matrix of the ith player, δ^j is a vector with only two non-zero components: -1 at the supply node and 1 at the demand node. Similarly, δ^a is a vector with only two non-zero components: -1 at the tail and 1 at the head of the arc a. The variable x^j is the flow of commodity $j \in J^i$ on the arcs of the network of the ith player. The variable x^k is the flow of contractual commodity $k \in K^i$ on the arcs of the network of the ith player.

The path-flow formulation of the problem requires some additional notation. Let $\mathcal{P}_j^i$ denote the set of paths in D^i from the source to the sink of the jth

commodity of player i. Similarly, $\mathcal{P}_k^i$ denotes the set of paths in D^i from the source to the sink of the kth contractual commodity of player i. We also assume that if there is more than one (normal or contractual) demand in D^i between s and t for a certain pair $s, t \in V$, then for each demand there is a separate $\mathcal{P}_j^i$ ($j \in J^i \cup K^i$) that contains all s-t paths in D^i. Let $\mathcal{P}^i = \cup_{j \in J^i} \mathcal{P}_j^i \bigcup \cup_{k \in K^i} \mathcal{P}_k^i$ and $\mathcal{P} = \cup_{i \in I} \mathcal{P}^i$. The corresponding LP problem is

$$\min \sum_{P \in \mathcal{P}} c(P) x_P \tag{7}$$

$$x \in \mathbb{R}^{\mathcal{P}} \tag{8}$$

$$x \geq 0 \tag{9}$$

$$\sum_{P \in \mathcal{P}^i : a \in P} x_P \leq u_a \qquad \forall a \in A^i,\ i \in I, \tag{10}$$

$$\sum_{P \in \mathcal{P}_j^i} x_P \geq d_j^i \qquad \forall j \in J^i,\ i \in I, \tag{11}$$

$$\sum_{P \in \mathcal{P}_{k_a}^{i_a}} x_P \geq d_a \left(\sum_{P \in \mathcal{P}^i : a \in P} x_P \right) \qquad \forall a \in B^i,\ i \in I. \tag{12}$$

The variable x_P, $P \in \mathcal{P}_j^i$ ($P \in \mathcal{P}_k^i$) belongs to the flow of (contractual) commodity j (k) on the path P, and $c(P) = \sum_{a \in P} c_a$.

The correspondence between the solutions of (2)-(6) and the solutions of (8)-(12) is the following. If x^l ($l \in J \cup K$) is a solution of (2)-(6), then we can take a path-decomposition of every flow x^l (leaving out the possible cycles) and get a solution of (8)-(12). On the other hand, if x_P ($P \in \mathcal{P}$) is a solution of (8)-(12), then $x^l = \sum_{P \in \mathcal{P}_l^i} x_P$ for every $i \in I$ and $l \in J^i \cup K^i$ gives a solution of (2)-(6).

2.2 Interpretation of feasible solutions

We assume $d_a \geq 1$ for all contractual arcs $a \in B_i$, hence $f_a(x) \geq x$ for all $a \in B_i, x \in [0, u_a]$. In this section we consider the path representation of the MMFP, using the notation $\mathcal{P}, \mathcal{P}^i, \mathcal{P}_j^i$ of the previous subsection. Our aim is to formalize the intuitive notion that the commodities corresponding to normal demands are actually transferred to their destination with a finite number of transfers between networks of different players.

A feasible solution in the path representation is given by a vector $x \in \mathbb{R}_+^{\mathcal{P}}$ that satisfies the constraints (10)-(12). Intuitively, a solution of this system amounts to a physical routing along paths, in which we satisfy a portion of the jth demand of the ith player by a certain path $P_1 \in \mathcal{P}_j^i$. However, if P_1 contains a contractual arc a, then the corresponding contractual demand is satisfied by another path $P_2 \in \mathcal{P}_{k_a}^{i_a}$, and so on. P_1 may contain several contractual arcs, thus we may need paths other than P_2 to satisfy contractual
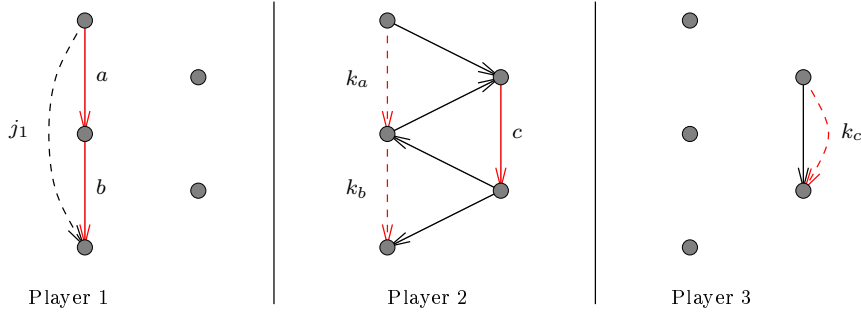
Fig. 1 A path whose realization cannot be represented by an arborescence. Black denotes normal and red denotes contractual; a solid line is an arc while a dotted line is a demand. In the realization of the path ab, the sink (which corresponds to the single-arc path in the network of player 3) has in-degree 2.

demands generated by P_1. Then again, P_2, and all the other paths will generate further contractual demands that have to be satisfied. Thus we may iteratively generate further contractual demands, with no end in sight. Even worse, the number of contractual demands may increase with every iteration, and also we could run into an infinite loop. This we have to avoid by all means if we want to demonstrate that a solution to the MMFP gives a realizable routing of the normal demands. Further, this motivates the following definition, which essentially formulates the finite realization of a path. We say that a path $P \in \mathcal{P}^i$ is a *normal path* if it contains only normal arcs.

Definition 1 Let $P \in \mathcal{P}$ be arbitrary. We give a recursive definition of a digraph (R, F) with $R \subseteq \mathcal{P}$: start with $R = \{P\}$ and $F = \emptyset$, and in every step choose a node $Q \in R$ that is not a normal path and has out-degree zero in (R, F). For every contractual arc $a \in Q$ choose an (arbitrary) path $Q_a \in \mathcal{P}^{i_a}_{k_a}$ and add it to R (if it was not already in it) and include the arc QQ_a in F, too. Clearly, the construction is finite. Thus the out-degree of any node Q in (R, F) equals the number of contractual arcs in Q. If (R, F) is acyclic, then we say that it is a *realization of P* (a path P might have many different realizations).

We are only interested in realization of paths $P \in \mathcal{P}^i_j$ for $j \in J^i$. The set of realized paths is denoted by $\mathcal{R}$. Let $\mathcal{R}^i_j$ denote the set of realized paths that realize a path in $\mathcal{P}^i_j$ for an $i \in I, j \in J^i$, and also let $\mathcal{R}^i := \cup_{j \in J^i} \mathcal{R}^i_j$ for any $i \in I$.

If (R, F) is a realization of P, then let $d_{(R,F)} : R \to \mathbb{R}_+$ denote the flow values needed to realize the path P at the root, which is defined as follows: $d_{(R,F)}(P) := 1$, and if $Q \in R \cap \mathcal{P}^{i_a}_{k_a}$ for some $a \in B$, then $d_{(R,F)}(Q) := d_a \sum_{j=1}^t d_{(R,F)}(Q_j)$ where $Q_1, Q_2, \ldots, Q_t$ are the parents of Q in (R, F).

One might think that the above definition is more laborious than necessary: a realized path could simply be an arborescence (R, F) with $R \subseteq \mathcal{P}$. However, the example in Figure 1 shows that we cannot make this assumption.

To interpret this definition in terms of shipping of commodities, consider a realized path (R, F) such that $P \in \mathcal{P}$ is the root in (R, F). Assume we want to ship one unit of the commodity from the first node of P to the last node of P. We may do this, of course, via P if it only uses normal arcs, and in this case (R, F) is a single node. Otherwise P may contain a number of contractual arcs, which may be realized by the children of P in (R, F). This, by constraint (12), requires the shipping of d_a units of the contractual commodity k_a by contractor i_a. Thus when we encounter a contractual arc a in $P \cap \mathcal{P}_j^i$, we satisfy the contractual demand generated by arc a using the unique child Q of P with $Q \in \mathcal{P}_{k_a}^{i_a}$. Note that we need to make $|F|$ such calls in total, since (R, F) is acyclic.

Given a solution $x \in \mathbb{R}_+^{\mathcal{P}}$ of (10)-(12), a realization of x, i.e. a *realization of a feasible solution* is given by a non-negative vector $y \in \mathbb{R}_+^{\mathcal{R}}$ that satisfies

$$\sum_{(R,F) \in \mathcal{R}: P \in R} y_{(R,F)} d_{(R,F)}(P) \leq x_P \qquad \forall P \in \mathcal{P} \qquad (13)$$

$$\sum_{(R,F) \in \mathcal{R}_j^i} y_{(R,F)} \geq d_j^i \qquad \forall j \in J^i, \ i \in I. \qquad (14)$$

A solution y of this system means that a realized path (R, F) is used at throughput $y_{(R,F)}$ in our realized routing. By the first constraint we make sure that y comes from a realization of x, and by the second constraint we satisfy all the demands.

In order to demonstrate that the model of MMFP is meaningful, we should show that feasible solutions can be realized with a routing of normal commodities using realization of paths, as defined above. The following Lemma is a formal statement of this claim.

Lemma 1 *Given a solution $x \in \mathbb{R}_+^{\mathcal{P}}$ of (10)-(12), there is a solution y of (13)-(14). Also, given x one can determine a solution y of (13)-(14) in polynomial time.*

In order to prove this lemma we define *acyclic solutions*.

2.2.1 Acyclic solutions

Definition 2 A feasible solution x of (2)-(6) is an *acyclic solution* if there is a linear order $\prec$ on $J \cup K$ such that if $l \in J \cup K$, $x^l(a) > 0$, and a is a contractual arc, then $l \prec k_a$. Similarly, a feasible solution of (8)-(12) is an *acyclic solution* if there is a linear order $\prec$ of $supp(x)$ such that for any $P \in supp(x)$ and any contractual arc $a \in P \cap B$ we have $P \prec P'$ for every $P' \in \mathcal{P}_{k_a}^{i_a} \cap supp(x)$.

Definition 3 For a feasible solution x of (8)-(12) we define the following edge-weighted directed graph D_x. The node set of D_x is $J \cup K \cup supp(x)$. The nodes in $supp(x)$ have in-degree 1: the only arc entering a path $P \in supp(x) \cap \mathcal{P}_l^i$ comes from node l (where $i \in I$ and $l \in J^i \cup K^i$) and this arc has weight x_P.

The arcs leaving this path P go to the nodes $k_a \in K$ for every $a \in P \cap B$ (i.e. the out-degree of P is zero if and only if P is a normal path) and these arcs have weight $d_a x_P$. Note that if $l \in J$ then l is not the head of any arc, and the digraph D_x is bipartite with color classes $J \cup K$ and $supp(x)$. Observe that x is acyclic if and only if this digraph is acyclic.

Proposition 1 *If x is a feasible solution of (8)-(12), then there is an acyclic solution x^* such that $x^* \leq x$.*

Proof Suppose indirectly that the nodes $P_1, k_{a_1}, P_2, k_{a_2}, \ldots, P_m, k_{a_m}$ form a cycle in D_x (in this order). Let $x' := x - \epsilon \sum_{i=1}^{m} \chi_{\{P_i\}}$ with ϵ maximum to maintain the non-negativity of x'. We claim that x' is a feasible solution of (8)-(12). Clearly, (8)-(10) holds for x' too, since $0 \leq x' \leq x$. (11) follows from the fact that $x'|_{\mathcal{P}_j^i} = x|_{\mathcal{P}_j^i}$ for every $i \in I$ and $j \in J^i$, since a cycle does not contain nodes in J. Note that (12) states that at any node $k_a \in K$, the sum of the weights of the arcs leaving the node k_a is at least the sum of the weights entering this node. Observe that $D_{x'}$ can be obtained from D_x by appropriately decreasing the weights of every arc entering or leaving the nodes $P_1, P_2, \ldots, P_m$ (and deleting the nodes in $supp(x) - supp(x')$). If k_a is not in our cycle for some $a \in B$ then the weight of the arcs leaving k_a did not decrease (while the weight of those entering k_a might have decreased) therefore (12) also holds for x' and a. On the other hand, if k_a is in the cycle for some $a \in B$, then the weight of exactly one arc leaving k_a decreased by ϵ and at least one arc entering k_a has had its weight decreased by $d_a \epsilon$. Since $d_a \geq 1$, this shows that (12) also holds for x' and a. Thus indeed x' is a solution to (8)-(12).

Since $supp(x') \subsetneq supp(x)$, by induction on $supp(x)$ there is an acyclic solution $x^* \leq x' \leq x$, proving the claim. It is also clear that this solution x^* can be found in polynomial time. $\square$

Corollary 1 *The social optimum is always achieved by an acyclic solution.*
                                                                                            $\square$

Proposition 2 *An acyclic solution can be decomposed into realized paths.*

Proof Suppose that x is a counterexample with $|supp(x)| + \sum_i |J^i|$ minimal. Consider the digraph D_x and choose an arbitrary node $j \in J^i$ (where $i \in I$) and an arbitrary path $P \in \mathcal{P}_j^i$. A realization (R, F) of P can be defined the following way: in every step the successors of a node Q should be chosen from $supp(x)$ (equivalently, this could be obtained from a subgraph of D_x by a suitable contraction). Since x is feasible, this can be done, and since x is acyclic, this will indeed give a realization of P. Then we define $x'_P := x_P - \epsilon d_{(R,F)}(P)$ if $P \in R$, and $x'_P := x_P$ if $P \notin R$, and $d_j'^i := d_j^i - \epsilon$, and $d_f'^e := d_f^e$ for $(e, f) \neq (i, j)$. Choose ϵ to be maximal such that d' and x' remain non-negative. We remove j from J^i if $d_j'^i$ becomes zero. Then x', d' is not a counterexample because of the minimality of x, hence there is a solution y' of (13)-(14) with respect to x', d'. Consequently, $y := y' + \epsilon \chi_{(R,F)}$ is a solution of (13)-(14) with respect to x, d. $\square$

Remark. The proof works with nonlinear functions with the assumption $f_a(x) \geq x$, too. In this case (12) can be nonlinear, i.e.,

$$\sum_{P \in \mathcal{P}^{ia}_{k_a}} x_P \geq f_a \left(\sum_{P \in \mathcal{P}^i : a \in P} x_P \right) \ \forall a \in B^i, \ i \in I.$$

Of course, we need restrictions on the functions f_a in order to be able to compute a realization in polynomial time as in Lemma 1.

Remark. The analogue of Corollary 1 for equilibria is not true: it is possible that there are equilibrium solutions but none of them is acyclic.

Remark. In case of $d_a < 1$, simple (and safe) counterexamples to Lemma 1 (in fact, to Claim 1) exist. Such an example was given in Subsection 1.2.

3 Calculating the social optimum

Our aim is to find the *social optimum*: a feasible solution which is optimal for the cost function c. We have seen that this can be written as an LP of polynomial size, so it can be solved in polynomial time. In practice, the running time can be dramatically reduced by the use of suitable solution methods: column generation and Lagrangian relaxation. This section introduces the solution algorithms suited to MMFP.

3.1 Column generation

In the following, we describe a column generation method based on the method of Ford and Fulkerson [5] to solve the problem. Let problem (7)-(12) be called the master LP. The column generation method takes a subset of paths, and solves the LP restricted to the variables corresponding to those paths. Then it checks whether the dual solution obtained is feasible for the master LP. If it is, then the primal solution is optimal for the master LP. If it is not, we can find a variable for which the corresponding dual inequality is violated, and we can add this variable to the restricted LP.

The crucial question is how to find a dual inequality that is violated, since there are exponentially many inequalities. We show that in our problem it is

easy. First we describe the dual problem.

$$\max \sum_{i \in I} \sum_{j \in J^i} d_j^i z_j^i - \sum_{i \in I} \sum_{a \in A^i} u_a y_a \tag{15}$$

$$y \in \mathbb{R}^{\sum_{i \in I} |A^i|} \tag{16}$$

$$z \in \mathbb{R}^{\sum_{i \in I} |J^i| + \sum_{i \in I} |B^i|} \tag{17}$$

$$y, z \geq 0 \tag{18}$$

$$z_j^i - \sum_{a \in (A^i \setminus B^i) \cap P} y_a - \sum_{a \in B^i \cap P} (y_a + d_a z_a^i) \leq c(P) \qquad \forall P \in \mathcal{P}_j^i, j \in J^i,$$
$$\forall i \in I, \tag{19}$$

$$z_a^i - \sum_{a' \in (A^{i_a} \setminus B^{i_a}) \cap P} y_{a'} - \sum_{a' \in B^{i_a} \cap P} (y_{a'} + d_{a'} z_{a'}^{i_a}) \leq c(P) \quad \forall P \in \mathcal{P}_{k_a}^{i_a}, a \in B^i,$$
$$\forall i \in I. \tag{20}$$

Suppose that we have a dual vector (y^*, z^*), and we want to decide if it is feasible, so we have to check inequalities (19) and (20). In order to find a path $P \in \mathcal{P}_j^i$ that violates (19), we have to find the shortest path from the source to the sink of the commodity $j \in J^i$ relative to the non-negative length function $l_a = y_a^* + c_a$ for $a \in A^i \setminus B^i$ and $l_a = y_a^* + d_a z_a^{i*} + c_a$ for $a \in B^i$. If the length of the shortest path is less than z_j^{i*}, then we have found a path for which (19) is violated, otherwise there is no such path in $\mathcal{P}_j^i$.

For $P \in \mathcal{P}_{k_a}^{i_a}$, we have to solve a similar problem with length function $l_{a'} = y_{a'}^* + c_{a'}$ for $a' \in A^{i_a} \setminus B^{i_a}$ and $l_{a'} = y_{a'}^* + d_{a'} z_{a'}^{i_a*} + c_{a'}$ for $a' \in B^{i_a}$. To sum up, we can test the feasibility of (y^*, z^*) by running the Dijkstra algorithm $|J| + |K|$ times.

3.2 Lagrangian relaxation

Consider the arc-flow formulation. Let us relax the capacity constraints (2) and assign to them the dual variable vector $y \geq 0$. The Lagrangian dual problem is

$$\max_{y \geq 0} \mathcal{L}(y),$$

where $\mathcal{L}(y)$ is defined as the optimum of

$$\min\ L(x, y) \tag{21}$$

$$N^i x^j = d^i_j \delta^j \qquad\qquad \forall j \in J^i,\ \forall i \in I, \tag{22}$$

$$N^{i_a} x^{k_a} = d_a \left(\sum_{j \in J^i} x^j_a + \sum_{k \in K^i} x^k_a \right) \delta^a \qquad \forall a \in B^i,\ \forall i \in I, \tag{23}$$

$$x^j \geq 0 \qquad\qquad \forall j \in J^i,\ \forall i \in I, \tag{24}$$

$$x^k_a \geq 0 \qquad\qquad \forall k \in K^i,\ \forall i \in I, \tag{25}$$

and $L(x, y)$ is the Lagrangian function

$$L(x, y) = \sum_{i \in I} \sum_{a \in A^i} c_a \left(\sum_{j \in J^i} x^j_a + \sum_{k \in K^i} x^k_a \right) + \sum_{i \in I} \sum_{a \in A^i} y_a \left(\sum_{j \in J^i} x^j_a + \sum_{k \in K^i} x^k_a - u_a \right).$$

Because the Lagrangian dual is the minimum of linear forms in y, it is concave. Moreover, it is possible to exhibit an element of the anti-subgradient at $\bar{y}$ if we know an optimal solution $\bar{x}$ of (21)-(25) at $\bar{y}$. Indeed, for any y, we have from the definition of $\mathcal{L}$,

$$\mathcal{L}(y) \leq \sum_{i \in I} \sum_{a \in A^i} c_a \left(\sum_{j \in J^i} \bar{x}^j_a + \sum_{k \in K^i} \bar{x}^k_a \right) + \sum_{i \in I} \sum_{a \in A^i} y_a \left(\sum_{j \in J^i} \bar{x}^j_a + \sum_{k \in K^i} \bar{x}^k_a - u_a \right). \tag{26}$$

Inequality (26) clearly shows that $-u + \bar{x}^j + \bar{x}^k$ is an anti-subgradient. Inequality (26) is sometimes referred to as an optimality cut for $\mathcal{L}$.

The Lagrangian function can be converted into the following form:

$$L(x, y) = -\sum_{i \in I} \sum_{a \in A^i} y_a u_a + \sum_{i \in I} \sum_{a \in A^i} (c_a + y_a) \left(\sum_{j \in J^i} x^j_a + \sum_{k \in K^i} x^k_a \right).$$

Hence, for any $y \geq 0$, problem (21)-(25) can be considered as an MMFP on the same $D^i = (V, A^i)$, $i \in I$, with costs $c_a + y_a$, $a \in \cup_{i \in I} A^i$ and without capacity constraints (i.e. $u_a = +\infty$, $a \in \cup_{i \in I} A^i$).

The solution of the MMFP without capacity constraints can be carried out in the following way. It is easy to see that without contractual arcs, problem (21)-(25) is separable into shortest-path problems from the source to the sink of the commodity j in J^i relative to the non-negative length function $l_a = c_a + y_a$ for $a \in A^i$, $i \in I$. In case of the existence of contractual arcs two kinds of demands have to be considered: the normal and the contractual. However, Lemma 1 says that there are normal realizations behind the contractual arcs. Hence, if we know the minimal cost of a realization of one unit of flow on a contractual arc $a \in B^i$ (let us denote it by n_a), then problem (21)-(25) can be considered as shortest-path problems from the source to the sink of the commodity j in J^i relative to the non-negative length function $l_a = c_a + y_a$

for $a \in A^i \backslash B^i$ and $l_a = c_a + y_a + n_a$ for $a \in B^i$, $i \in I$. In order to find n_a for all $a \in \cup_{i \in I} B^i$ we apply the following

Dynamic programming algorithm to find the minimal cost of a realization of one unit of flow on a contractual arc

Step 0. Let $n_a(0) = +\infty$ for all $a \in \cup_{i \in I} B^i$. Let $k = 0$.

Step 1. Let $n_a(k + 1) = \min(n_a(k), d_a l(P_a(k)))$ for all $a \in \cup_{i \in I} B^i$, where $l(P_a(k))$ is the length of the shortest path between the tail and head of a in $D^{i_a} = (V, A^{i_a})$ with arc-lengths $l_a = c_a + y_a$ for $a \in A^{i_a} \backslash B^{i_a}$ and $l_a = c_a + y_a + n_a(k)$ for $a \in B^{i_a}$.

Step 2. If $n_a(k + 1) = n_a(k)$ for all $a \in \cup_{i \in I} B^i$ then STOP. Else $k \leftarrow k + 1$ GOTO Step 1.

Theorem 1 *The algorithm stops within at most $\sum_{i \in I} |B_i| + 1$ iterations.*

Proof Corollary of Lemma 1.   □

By the use of the above method the values as well as the anti-subgardients of the concave function $\mathcal{L}(y)$ can be calculated. Hence, several algorithms exist to find the maximum of it. For example, the the analytic center cutting-plane method (ACCPM) of [7] can be applied.

As a consequence, we can solve MMFP using slight modifications of multicommodity flow algorithms based on Lagrangian relaxation. In particular, the algorithm of [2], which uses partial Lagrangian relaxation with proximal-ACCPM (see [9]) can be easily converted into an algorithm for MMFP, the only difference being the computation of $\mathcal{L}(y)$, as detailed above.

4 Results on equilibria

Equilibrium solutions, defined in Subsection 1.2, are solutions where no player has the intention to change the routing in his network. In this section we show that in general it is NP-complete to decide if an equilibrium solution exists, while in safe instances there always exists one. However, even in safe instances the social cost of equilibria can be arbitrarily high, and it is NP-complete to decide if there is one with social cost smaller than C.

4.1 Existence of equilibrium solutions

In this subsection we show that an equilibrium solution always exists in a safe instances, even under the very weak assumption that the functions f_a are continuous and monotone increasing.

Theorem 2 *In a safe instance of the problem there is always an equilibrium solution.*

Proof As before, let A be the set of all arcs, B the set of all contractual arcs, and let

$$Q = \times_{a \in B}[0, u_a],$$

i.e. Q is the space of all possible flow values on contractual arcs. Q is a compact convex set in $\mathbb{R}^B$. Since contractual demands are determined by the flow values on contractual arcs, we can associate to an element $x \in Q$ a set of all (contractual plus normal) demands $\mathrm{Dem}(x)$.

We define a set-valued function $\phi : Q \to \mathcal{P}(Q)$ the following way. For $x \in Q$, we consider the demands $\mathrm{Dem}(x)$, and take the direct sum of the polyhedra of all minimum cost multicommodity flows (w.r.t. cost function c for normal arcs and $c + p$ for contractual arcs) for each player. Let $\phi(x)$ be the projection of this polyhedron to the coordinates corresponding to contractual arcs. In other words, $\phi(x)$ contains all vectors on B that arise as arc values of minimum cost multicommodity flows for demands $\mathrm{Dem}(x)$.

It is clear that $\phi(x)$ is closed and convex for every x, since it is a polyhedron. We show that ϕ is upper semi-continuous, i.e. its graph is closed. Let $x_i \in Q$ ($i \in \mathbb{N}$) be a convergent sequence, $x_i \to x \in Q$. Let furthermore $y_i \in \phi(x_i)$ ($i \in \mathbb{N}$) such that $y_i \to y \in Q$. For y_i, let $\overline{y}_i \in \mathbb{R}^A$ be a minimum cost multicommodity flow for demands $\mathrm{Dem}(x_i)$ whose restriction to B is y_i. By the standard argument, we can select a subsequence $i_1, i_2, \ldots$ so that $\overline{y}_{i_j}(a)$ is convergent for every $a \in A$. Let $\overline{y}$ denote the limit. Since the functions f_a ($a \in B$) are continuous and the multicommodity flow problem is linear, $\overline{y}$ is a minimum cost multicommodity flow for demands $\mathrm{Dem}(x)$. This proves that $y \in \phi(x)$.

By Kakutani's fixed point theorem [8], there is an element $x^* \in Q$ such that $x^* \in \phi(x^*)$. This means that for demands $\mathrm{Dem}(x^*)$, there are minimum cost multicommodity flows for each player with contractual arc values equal to x^*. This corresponds to an equilibrium solution. $\qquad\square$

4.2 NP-completeness

Let us define an auxiliary directed graph $D^* = (I, A^*)$ on the set of players. There is an arc from i to i_a for each contractual arc $a \in B$. We allow parallel arcs, so $|A^*| = |B|$. We say that the *contracts are acyclic* if D^* is acyclic.

We know that in a safe instance there is always an equilibrium solution. In contrast, the following result shows that for non-safe instances it is hard to decide if there is an equilibrium.

Theorem 3 *In general it is NP-compete to decide whether there is an equilibrium solution. This holds even if the contracts are acyclic and the instance is almost safe in the sense that there is only one contractual demand that cannot be routed.*

Proof We reduce SAT to this problem. Suppose there are m clauses and n variables $x_1, x_2, \ldots, x_n$. We construct an instance with $n + 2$ players: the first

n players correspond to variables, the network of the $(n+1)$-th player models the clauses, and the $(n+2)$-th player is a dummy player with no arcs in his network. We show that the instance has an equilibrium solution if and only if the SAT instance is satisfiable.

The j-th player $(j = 1, \ldots, n)$ has a normal demand of one unit from u_j to v_j and his network contains two disjoint paths from u_j to v_j, one corresponding to x_j, the other to $\overline{x_j}$. These paths contain several contractual arcs: one for each clause in which x_j (resp. $\overline{x_j}$) appears. The contractor for these arcs is the $(n+1)$-th player. All costs are 0 and all capacities are 1.

The $(n+1)$-th player has m normal demands of one unit, one for each clause. The i-th demand is from s_i to t_i, and there is a normal arc from s_i to t_i with high cost and unit capacity. In addition, for each $i \in \{1, \ldots, m\}$ there is a path from s_i to t_i with cost 0 and unit capacity, and these paths share a single arc st (but apart from the nodes s and t they are node-disjoint). The arc st is a contractual arc with the $(n+2)$-th player, who has no $s-t$ path in his network, so this contract cannot be realized.

All the contractual demands of the $(n+1)$-th player lie on these paths: for all literals in clause i, the contractual demands corresponding to the literals cover disjoint sections of the path (not containing st).

In an equilibrium solution, the $(n+1)$-th player must route his normal demands on the high-cost $s_i - t_i$ arcs, because the low-cost paths contain an unrealizable contract. By the definition of equilibrium, he must have a minimum cost multicommodity flow in his network, which means that all the low cost paths must contain an arc saturated by the flow corresponding to a contractual demand, i.e. there must be a contractual demand of value 1 on each low cost path. This means that the flows of the first n players must be paths that correspond to an evaluation that satisfies every clause.

To see the other direction, consider an evaluation that satisfies every clause. This defines an equilibrium solution the following way. The j-th player $(j = 1, \ldots, n)$ routes his normal demand on the path corresponding to x_j if x_j is true in the evaluation, and on the path corresponding to $\overline{x_j}$ if it is false. The $(n+1)$-th player routes his normal demands on the high-cost $s_i - t_i$ arcs. It can be seen by the same argument as above that this is an equilibrium solution.

$\square$

Remark. A similar proof shows that in a safe instance it is NP-complete to decide if there is an equilibrium solution with social cost smaller than C, even if the contracts are acyclic. The only modification is that the "high cost" in the construction above should be small compared to C, and a normal arc st of cost C and capacity m should be added to the network of the $(n+2)$-th player.

References

1. R. Agarwal and Ö. Ergun, *Mechanism design for a multicommodity flow game in service network alliances*, Operations Research Letters 36 (2008), 520–524.
2. F. Babonneau, O. du Merle and J-P. Vial, *Solving Large-Scale Linear Multicommodity Flow Problems with an Active Set Strategy and Proximal-ACCPM*, Oper. Res. 54(1) (2006), 184–197.
3. A. Bompadre, J.B. Orlin, *A simple method for improving the primal simplex method for the multicommodity flow problem*, Networks 51 (2008), 63–77.
4. J.J.M. Derks and S.H. Tijs, *Stable outcomes for multicommodity flow games*, Methods of Operations Research, 50 (1985), 493–504.
5. L.R. Ford, jr. and D.R. Fulkerson, *A suggested computation for maximal multi-commodity network flows*, Manage. Sci. 5 (1958), 97–101.
6. N. Garg, J. Könemann, *Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems*, SIAM J. Comput. 37 (2007), 630–652.
7. J.-L. Goffin, A. Haurie and J.-P. Vial, *Decomposition and nondifferentiable optimization with the projective algorithm*, Management Sci. 38(2) (1992), 284–302.
8. S. Kakutani, *A generalization of Brouwer's fixed point theorem*, Duke Math. J. 8 (1941), 457–459.
9. O. du Merle and J.-P. Vial, *Proximal ACCPM, a cutting plane method for column generation and Lagrangian relaxation: Application to the p-median problem*, Technical report, Logilab, University of Geneva, Geneva, Switzerland, 2002.