On Stable Matchings and Flows

Tamás Fleiner*

Budapest University of Technology and Economics,
Department of Computer Science and Information Theory,
Magyar tudósok körútja 2. H-1117, Budapest, Hungary
fleiner@cs.bme.hu

Abstract. We describe a flow model that generalizes ordinary network flows the same way as stable matchings generalize the bipartite matching problem. We prove that there always exists a stable flow and generalize the lattice structure of stable marriages to stable flows. Our main tool is a straightforward reduction of the stable flow problem to stable allocations.

Keywords: Stable marriages; stable allocations; network flows.

1 Introduction

In the stable marriage problem of Gale and Shapley [6], there are n men and n women and each person ranks the members of the opposite gender by an arbitrary strict, individual preference order. A *marriage scheme* in this model is a set of marriages between different men and women. Such a scheme is *unstable* if there exists a *blocking pair*, that is, a man m and a woman w in such a way that m is either unmarried or m prefers w to his wife, and at the same time, w is either unmarried or prefers m to her partner. A marriage scheme is *stable* if it is not unstable, that is, not blocked by any pair. It is a natural problem to find a stable marriage scheme if it exists at all. Nowadays, it is already folklore that for any preference rankings of the n men and n women, a stable marriage scheme does exist. This theorem was proved first by Gale and Shapley in [6]. They constructed a special stable marriage scheme with the help of a finite procedure, the so-called deferred acceptance algorithm. It also turned out that for the existence of a stable scheme, it is not necessary that the number of men is the same as the number of women or that for each person, all members of the opposite gender are acceptable: the deferred acceptance algorithm is so robust that it works properly in these more general settings.

Several interesting properties about the structure of stable marriage schemes are known. Donald Knuth [7] attributes to John Conway the observation that

* Research is supported by OTKA grant K69027 and the MTA-ELTE Egerváry Research Group (EGRES).

D.M. Thilikos (Ed.): WG 2010, LNCS 6410, pp. 51–62, 2010.

stable marriages have a lattice structure: if each man picks the better assignment out of two stable marriage schemes then another stable marriage scheme is created in which each women receives the worse out of the two husbands.

There are further known extensions of the stable marriage problem. Baïou and Balinski proved in [1] that if each edge of the underlying bipartite graph has a nonnegative capacity and each vertex has a nonnegative quota then the accordingly modified deferred acceptance algorithm shows that there always exists a so called stable allocation. An allocation is an assignment of nonnegative values to the edges that do not exceed the corresponding capacities such that the total allocation of no vertex exceeds its quota. (That is, a "marriage" can be formed with an "intensity" different from 0 and 1 and each participant has an individual upper bound on his/her total "marriage intensity".) An allocation is stable if any unsaturated edge e has a saturated end vertex v such that no edge e' incident to v and preferred by v less than e has a positive value. Beyond proving the existence of stable assignments, Baïou and Balinski used flow-type arguments to speed up the deferred acceptance algorithm in [1]. Later, Dean and Munshi came up with an even faster algorithm for the same problem [3] that also has to do with network flows.

It is fairly well-known that the bipartite matching problem can be formulated in the more general network flow model, and the alternating path algorithm for maximum bipartite matchings is a special case of the augmenting path algorithm of Ford and Fulkerson for maximum flows. However, it seems that the question whether there exists a flow generalization of the stable marriage theorem has not been addressed so far. This very problem is in the focus of our present work. In section 2, we formulate the stable flow problem and state a result from [1] by Baïou and Balinski on stable allocations. Section 3 contains the stable flow theorem, a generalization of the Gale-Shapley theorem to flows. Our reduction of the stable flow problem to the stable allocation problem resembles to the reduction of the maximum flow problem to the maximum b-matching problem. Actually, our construction has to do also with the one that Cechlárová and Fleiner used in [2] to extend the stable roommates model to a multiple partner model. Section 4 is devoted to certain structural results on stable flows, in particular we generalize the lattice structure of stable marriages. To achieve this, we lean on the construction we used for the reduction. The interested reader can find the extended version of our work with the proofs and with an application showing a certain "linking property of flows" in [5].

It turned out that our model is closely related to so-called "supply chains" well-known in the Economics literature. Prior to our work, Ostrovsky had a related result in [8]. There, he considers only acyclic networks, but instead of the Kirchhoff law, he requires a less restrictive property that he calls "same side substitutability" and "cross side complementarity". In [8] the author proves the existence of a "chain stable network" and justifies that these "chain stable networks" form a lattice under a natural partial order. Ostrovsky's results are very close to ours and these cry for a common generalization. This will be subject of a future work.

2 Preliminaries

Recall that by a *network* we mean a quadruple (D, s, t, c), where $D = (V, A)$ is a digraph, s and t are different nodes of D and $c : A \to \mathbb{R}_+$ is a function that determines the capacity $c(a)$ of each arc a of A. (Sometimes it is assumed that no arc enters vertex s and no arc leaves vertex t. Though this assumption would allow a simpler proof, we do not require it for the reason that the result is significantly more general this way. Still, if the reader finds it difficult to follow the argument, it might be convenient to consider the source-sink case and skip the irrelevant parts.) A *flow* of network (D, s, t, c) is a function $f : A \to \mathbb{R}$ such that capacity condition $0 \leq f(a) \leq c(a)$ holds for each arc a of A and each vertex v of D different from s and t satisfies the Kirchhoff law: $\sum_{uv \in A} f(uv) = \sum_{vu \in A} f(vu)$, that is, the amount of the incoming flow equals the amount of the outgoing flow for v. Note that there is no conceptual difference between s and t: both are ordinary vertices that are exempt from the Kirchhoff law. (It seems that many people do not realize this. The reason perhaps is that when we teach network flows, we used to emphasize that the role of s and t are so different: one is "the source" and the other is a "the sink". To convince the sceptic, it is illuminative to find a formula for the minimum value of an st flow in a network. It is not 0 in general.)

A *network with preferences* is a network (D, s, t, c) along with a preference order $\leq_v$ for each vertex v, such that $\leq_v$ is a linear order on the arcs that are incident to v. (Note that preference orders $\leq_s$ and $\leq_t$ do not play a role in the notion of stability. Moreover, we shall never have to compare an incoming and an outgoing arc of the same vertex, so we may think that for each vertex v there is a preference order on the incoming arcs and another one on the outgoing ones.) For a given network with preferences, it is convenient to think that vertices of D are "players" that trade with a certain product. An arc uv of D from player u to player v with capacity $c(uv)$ represents the possibility that player u can supply at most $c(uv)$ units of product to player v. A "trading scheme" is described by a flow f of the network, as for any two players u and v, flow $f(uv)$ determines the amount of product that u sells to v. Everybody in the market would like to trade as much as possible, that is, each player v strives to maximize the amount of flow through v. In particular, if flow f allows player v to receive some more flow (that is, there are products on the market that v can buy) and v can also send some more flow (i.e. some player would be happy to buy more products from v) then flow f does not correspond to a stable market situation.

Another instability occurs when $vw \leq_v vu$ (player v prefers to sell to w rather than to u) and flow f is such that w would be happy to buy more product from v (that is $f(vw) < c(vw)$ and w has some extra selling capacity), moreover $f(vu) > 0$ (v sells a positive amount of products to u). In this situation, v would send flow rather to w than to u, hence a stable market situation does not allow the above situation. A similar instability can be described if we talk about entering arcs instead of outgoing ones, that is, if we exchange the roles of buying and selling.

To formalize our concept of stability we need a few definitions. For a network (D, s, t, c) and flow f we say that arc a is *f-unsaturated* if $f(a) < c(a)$, that is, if it is possible to send some extra flow thorough P. A *blocking walk of flow f* is an alternating sequence of incident vertices and arcs $P = (v_1, a_1, v_2, a_2, \dots, a_{k-1}, v_k)$ such that all the following properties hold.

$$\text{arc } a_i \text{ points from } v_i \text{ to } v_{i+1} \text{ for } i = 1, 2, \dots, k-1 \text{ and} \tag{1}$$

$$\text{vertices } v_2, v_3, \dots, v_{k-1} \text{ are different from } s \text{ and } t \tag{2}$$

$$\text{each arc } a_i \text{ is } f\text{-unsaturated and} \tag{3}$$

$$v_1 \in \{s, t\} \text{ or there is an arc } a' = v_1 u \text{ such that } f(a') > 0 \text{ and } a_1 <_{v_1} a' \tag{4}$$

$$v_k \in \{s, t\} \text{ or there is an arc } a^* \text{ to } v_k \text{ such that } f(a^*) > 0 \text{ and } a_{k-1} <_{v_k} a^* . \tag{5}$$

So directed walk P is blocking if each player that corresponds to an inner vertex of P is happy and capable to increase the flow along P, moreover v_1 can send extra flow either because $v_1 = s$ or $v_1 = t$ is a terminal node or because v_1 may decrease the flow toward some vertex u that v_1 prefers less than v_2, and at last, v_k can receive some extra flow either because either $v_k \in \{s, t\}$ or v_k can refuse some flow arriving from w whom v_k ranks below v_{k-1}. (As we mentioned before, there is no difference between the roles of s and t in the network: none of them have to obey the Kirchhoff law and both of them can send or receive flow. If the reader is uncomfortable with the idea that the target node sends flow to the source then consider the case where no arc enters s and no arc leaves t. This assumption simplifies some of the proofs.) We say that an f-unsaturated path $P = (v_1, v_2, \dots, v_k)$ is *f-dominated at v_1* if (4) does not hold, and P is f-dominated at v_k if (5) does not hold.

A flow f of a network with preferences is *stable* if no blocking walk exists for f. In the *stable flow problem* we have given a network with preferences and our task is to find a stable flow if such exists.

A special case of the stable flow problem is the stable allocation problem of Baïou and Balinski [1]. The *stable allocation problem* is defined by finite disjoint sets W and F of workers and firms, a map $q : W \cup F \to \mathbb{R}$, a set E of edges between W and F along with a map $p : E \to \mathbb{R}$ and for each worker or firm $v \in W \cup F$ a linear order $<_v$ on those pairs of E that contain v. We shall refer to pairs of E as "edges" and hopefully it will not cause ambiguity. Quota $q(v)$ denotes the maximum of total assignment that worker or firm v can accept and capacity $p(wf)$ of edge $e = wf$ means the maximum allocation that worker w can be assigned to firm f along e. An *allocation* is a nonnegative map $g : E \to \mathbb{R}$ such that $g(e) \le p(e)$ holds for each $e \in E$ and for any $v \in W \cup F$ we have

$$g(v) := \sum_{vx \in E} g(vx) \le q(v) \ , \tag{6}$$

that is the total assignment $g(v)$ of player v cannot exceed quota $q(v)$ of v. If (6) holds with equality then we say that player v is *g-saturated*. An allocation is *stable* if for any edge wf of E at least one of the following properties hold:

$$g(wf) = p(wf) \text{(the particular employment is realized with full capacity)}, \quad (7)$$

$$\text{worker } w \text{ is } g\text{-saturated and } w \text{ does not prefer } f \text{ to any of his employers (we say that } wf \text{ is } \textit{g-dominated at } w\text{)}, \quad (8)$$

$$\text{firm } f \text{ is } g\text{-saturated and } f \text{ does not prefer } w \text{ to any of its employees (that is, edge } wf \text{ is } \textit{g-dominated at firm } f\text{)}. \quad (9)$$

If g_1 and g_2 are allocations and $w \in W$ is a worker then we say that *allocation* g_1 *dominates allocation* g_2 *for worker* w (in notation $g_1 \leq_w g_2$) if one of the following properties is true:

$$\text{either } g_1(wf) = g_2(wf) \text{ for each } f \in F \quad (10)$$

$$\text{or } \sum_{f' \in F} g_1(wf') = \sum_{f' \in F} g_2(wf') = q(w), \text{ and } g_1(wf) < g_2(wf) \text{ and } g_1(wf') > 0 \text{ implies that } wf' <_w wf. \quad (11)$$

That is, if w can freely choose his allocation from $\max(g_1, g_2)$ then w would choose g_1 either because g_1 and g_2 are identical for w or because w is saturated in both allocations and g_1 represents w's choice out of $\max(g_1, g_2)$. By exchanging the roles of workers and firms, one can define domination relation $\leq_f$ for any firm f, as well.

For any stable allocation problem, one can design a network (D, s, t, c) such that $V(D) = \{s, t\} \cup W \cup F$, $A(D) = \{sw : w \in W\} \cup \{ft : f \in F\} \cup \{wf : wf \in E\}$ and $c(sw) = q(w)$, $c(ft) = q(f)$ and $c(wf) = p(wf)$ for any worker w and firm f. That is, we consider the underlying bipartite graph, orient its edges from W to F, add new vertices s and t, with an arc from s to each worker-node and an arc from each firm-node to t, and capacities are given by the original edge-capacities and the corresponding quotas. Preference orders $<_v$ on the arcs incident to v are induced by the preference order on the corresponding edges incident to v, or, if there is no such edge, then it is a trivial linear order. It is straightforward to see from the definitions that g is a stable allocation if and only if there exists a stable flow f such that $g(e) = f(\boldsymbol{e})$ holds for each edge $e \in E$, where $\boldsymbol{e}$ is the arc that corresponds to edge e. The stable allocation problem was introduced by Baïou and Balinski as a certain "continuous" version of the stable marriage problem in [1]. It turned out that a natural extension of the deferred acceptance algorithm of Gale and Shapley [6] works for the stable allocation problem and the structure of stable allocations is similar to that of stable marriages. Beyond stating the existence of stable allocations, the theorem below describes some structural properties of them. The interested reader finds a proof based on Tarski's fixed point theorem in [5].

Theorem 1 (See Baïou and Balinski [1])
1. If stable allocation problem is described by W, F, E, p and q then there always exists a stable allocation g. Moreover, if p and q are integral, then there exists an integral stable allocation g.

2. If g_1 and g_2 are stable allocations and $v \in W \cup F$ then $g_1 \leq_v g_2$ or $g_2 \leq_v g_1$ holds.

3. Stable allocations have a natural lattice structure. I.e., if g_1 and g_2 are stable allocations then $g_1 \vee g_2$ and $g_1 \wedge g_2$ are stable allocations, where

$$(g_1 \vee g_2)(wf) = \begin{cases} g_1(wf) \text{ if } g_1 \leq_w g_2 \\ g_2(wf) \text{ if } g_2 \leq_w g_1 \end{cases} \quad \text{and} \tag{12}$$

$$(g_1 \wedge g_2)(wf) = \begin{cases} g_1(wf) \text{ if } g_1 \leq_f g_2 \\ g_2(wf) \text{ if } g_2 \leq_f g_1 \end{cases} \tag{13}$$

In other words, if workers choose from two stable allocations then we get another stable allocation, and this is also true for the firms' choices. Moreover, it is true that

$$(g_1 \vee g_2)(wf) = \begin{cases} g_1(wf) \text{ if } g_1 \geq_f g_2 \\ g_2(wf) \text{ if } g_2 \geq_f g_1 \end{cases} \quad \text{and} \tag{14}$$

$$(g_1 \wedge g_2)(wf) = \begin{cases} g_1(wf) \text{ if } g_1 \geq_w g_2 \\ g_2(wf) \text{ if } g_2 \geq_w g_1 \end{cases} \tag{15}$$

That is, in stable allocation $g_1 \vee g_2$ where each worker picks his better assignment, each firm receives the worse out of the two. Similarly, in $g_1 \wedge g_2$ the choice of the firms means the less preferred situation to the workers.

3 Stable Flows

Our goal in this section is to prove a generalization of Theorem 1. The "natural" approach to achieve this would be an appropriate generalization of the deferred acceptance algorithm of Gale and Shapley. The difficulty is that though the Gale-Shapley algorithm can handle quota function q, somehow it has problems with ensuring the Kirchhoff law.

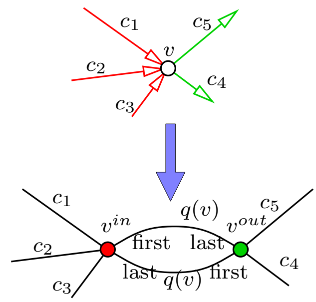
Theorem 2. *If network (D, s, t, c) and preference orders $<_v$ describe a stable flow problem then there always exists a stable flow f. If capacity function c is integral then there exists an integral stable flow.*

Note that it is possible to prove Theorem 2 by a mixture of the deferred acceptance algorithm and the augmenting path algorithm. That is, starting from s or from t, we follow "first choice walks" until they arrive to s or t and we augment along them with observing the capacity constraints. If a new path collides with an earlier one then some amount of flow is refused by the receiving vertex and we try to reroute the flow excess from the starting point of the refused arc. We have a stable flow as soon as we cannot find an augmenting path between the terminals.

Our proof of Theorem 2 follows a different approach for two reasons. On one hand, it seems that in the area of stable matchings neither the reduction of one problem to another one nor the use of graph terminology is routine. We demonstrate here that these methods may be fruitful. On the other hand, the "deferred augmentation" algorithm we sketched above does not give much information about the rich structure of stable flows that we shall deduce from the lattice property of stable allocations.

With the help of the given stable flow problem, we shall define a stable allocation problem. For each vertex v of D calculate

$$M(v) := \min\left(\sum_{xv\in A(D)} c(xv), \sum_{vx\in A(D)} c(vx)\right) ,$$



that is, $M(v)$ is the minimum of total capacity of those arcs of D that enter and leave v. So $M(v)$ is an upper bound on the amount of flow that can flow through vertex v. Choose $q(v) := M(v) + 1$. Construct graph G_D as follows. Split each vertex v of D into two distinct vertices v^{in} and v^{out}, and for each arc uv of D add edge $u^{out}v^{in}$ to G_D.

For each vertex v of D different from s and t add two parallel edges between v^{in} and v^{out}: to distinguish between them we will refer them as $v^{in}v^{out}$ and $v^{out}v^{in}$. Let $p(v^{in}v^{out}) = p(v^{out}v^{in}) := q(v)$, $p(u^{out}v^{in}) := c(uv)$ and $q(v^{in}) = q(v^{out}) := q(v)$. To finish the construction of the stable allocation problem, we need to fix a linear preference order for each vertex of G_D. For vertex v^{in} let $v^{in}v^{out}$ be the most preferred and $v^{out}v^{in}$ be the least preferred edge (if these edges are present), and the order of the other edges incident to v^{in} are coming from the preference order of v on the corresponding arcs. For vertex v^{out} the most preferred edge is $v^{out}v^{in}$ and the least preferred one is $v^{in}v^{out}$ (if it makes sense), and the other preferences are coming from $<_v$.

The proof of Theorem 2 is a consequence of the following Lemma that describes a close relationship between stable flows and stable allocations.

Lemma 1. *If network (D, s, t, c) and preference orders $<_v$ describe a stable flow problem then $f : A(D) \to \mathbb{R}$ is a stable flow if and only if there is a stable allocation g of G_D such that $f(uv) = g(u^{out}v^{in})$ holds for each arc uv of D.*

Proof. Assume first that g is a stable allocation in G_D. This means that none of the $v^{in}v^{out}$ edges is blocking, so either $g(v^{in}v^{out}) = p(v^{in}v^{out}) = q(v)$ or $v^{in}v^{out}$ must be g-dominated at v^{out}, hence v^{out} is assigned to $q(v^{out}) = q(v)$ amount of allocation. As $q(v)$ is more than the total capacity of arcs leaving v, $g(v^{in}v^{out}) > 0$ or $g(v^{out}v^{in}) > 0$ must hold. So v^{out} must have exactly $q(v)$ amount of allocation whenever $v^{in}v^{out}$ is present. An exchange of *in* and *out* shows that the presence of $v^{out}v^{in}$ implies that v^{in} has exactly $q(v^{in}) = q(v)$ allocation. These observations directly imply that the Kirchhoff law holds for f at each node different from s and t. The capacity condition is also trivial for f, hence f is a flow of D. Observe that by the choice of q, neither s nor t is g-saturated hence no edge is g-dominated at s or at t.

Assume that walk $P = (v_1, v_2, \ldots, v_k)$ blocks flow f. As P is f-unsaturated, each edge $v_i^{out}v_{i+1}^{in}$ of G_D must be g-dominated at v_i^{out} or at v_{i+1}^{in}. Walk P is blocking, hence either $v_1 \in \{s, t\}$, and hence $v_1^{out}v_2^{in}$ cannot be dominated at v_1 or there is a v_1u arc with positive flow value such that $v_1u > v_1v_2$. In both cases, edge $v_1^{out}v_2^{in}$ has to be g-dominated at v_2^{in}. It means that $g(v_2^{in}v_2^{out}) > 0$. As arc v_2v_3 is f-unsaturated, it follows that edge $v_2^{out}v_3^{in}$ must be g-dominated at v_3^{in}. This yields that $g(v_3^{in}v_3^{out}) > 0$. Again, arc v_3v_4 is f-unsaturated, hence edge $v_3^{out}v_4^{in}$ has to be g-dominated at v_4^{in}, and so on. At the end we get that $v_{k-1}^{out}v_k^{in}$ is g-dominated at v_k^{in}. If $v_k \in \{s, t\}$ then it is impossible as both these vertices are g-unsaturated. Otherwise by the blocking property of P there is an arc wv_k with positive flow and $v_{k-1}v_k <_{v_k} wv_k$, hence again, $v_{k-1}^{out}v_k^{in}$ cannot be g-dominated at v_k^{in}. The contradiction shows that no path can block f.

Assume now that f is a stable flow of D. We have to exhibit a stable allocation g of G_D such that f is the "restriction" of g. To determine g, our real task is to find the $g(v^{in}v^{out})$ and $g(v^{out}v^{in})$ values, as all other values of g are determined directly by f: $g(u^{out}v^{in}) = f(uv)$. The stable allocation we look for might not be unique. In what follows, we shall construct the *canonical representation* g_f of f.

Let S be the set of those vertices u of D such that there exists an f-unsaturated directed path $P = (v_1, v_2, \ldots, v_k = u)$ that is not f-dominated at v_1. As no path can block f, neither s, nor t belongs to S. To determine g_f, for each vertex $v \neq s, t$ allocate the remaining quota of v to $v^{in}v^{out}$ or to $v^{out}v^{in}$ depending on whether $v \in S$ or $v \notin S$ holds. More precisely, define

$$g_f(v^{in}v^{out}) = \begin{cases} q(v) - \sum_{x \in V(D)} f(vx) & \text{if } v \in S \\ 0 & \text{if } v \notin S \end{cases} \quad \text{and} \tag{16}$$

$$g_f(v^{out}v^{in}) = \begin{cases} q(v) - \sum_{x \in V(D)} f(xv) & \text{if } v \notin S \\ 0 & \text{if } v \in S \ . \end{cases} \tag{17}$$

By the definition of q, both $g_f(v^{in}v^{out})$ and $g_f(v^{out}v^{in})$ are nonnegative. If $v \in S$ then the amount of total allocation of v^{out} is $q(v) = q(v^{out})$ by (16), and for $v \notin S$ the amount of total allocation of v^{in} is $q(v) = q(v^{in})$ by (17). So if $v \neq s, t$ then the total allocation of v^{in} and v^{out} is $q(v)$ by the Kirchhoff law. The total allocations of s^{in}, s^{out} and t^{in}, t^{out} is less than $q(s)$ and $q(t)$ respectively, by the choice of q. That is, g_f is an allocation on G_D.

To justify the stability of g_f, we have to show that no blocking edge exists. We have seen earlier, that the presence of $v^{in}v^{out}$ in G_D means that v^{out} g-dominates $v^{in}v^{out}$. Similarly, each edge $v^{out}v^{in}$ is g_f-dominated at v^{in}. Assume now that $g_f(v^{out}u^{in}) < p(v^{out}u^{in}) = c(vu)$ holds.

If there is an f-unsaturated path P that is not f-dominated at its starting node and ends with arc vu then $u \in S$ by the definition of S, hence $g_f(u^{out}u^{in}) = 0$. Moreover, if some edge $w^{out}u^{in}$ with $v^{out}u^{in} <_{u^{in}} w^{out}u^{in}$ would have positive allocation then path P would block f, a contradiction. As u^{in} has $q(u^{in})$ amount of total allocation, edge $v^{out}u^{in}$ is g_f-dominated at u^{in}.

The last case is when any f-unsaturated path that ends with arc vu is f-dominated at its starting vertex. In particular, $v \notin S$, so $g_f(v^{in}v^{out}) = 0$.

Moreover, f-unsaturated path (v, u) must be f-dominated at v, hence $v \notin \{s, t\}$ and $v^{out}u^{in}$ is g_f-dominated at v^{out} as v^{out} has $q(v) = q(v^{out})$ amount of allocation. The conclusion is that $g := g_f$ is a stable allocation, just as we claimed.

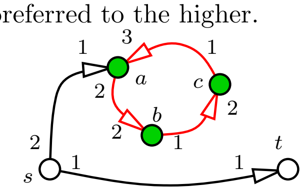
At this point, we are ready to prove our main result.

Proof (Proof of Theorem 2). There is a stable allocation for G_D by Theorem 1, hence there is a stable flow for D due to the first part of Theorem 1. If c is integral then $q(v)$ is an integer for each vertex v of D hence p is integral for G_D. The integrality property of stable allocations in the first part of Theorem 1 shows that there is an integral stable allocation g of G_D that describes an integral stable flow f of D.

At the end of this section let us point out a weakness of our stability concept. The motivation behind the notion is that we look for a flow that corresponds to an equilibrium situation where the players represented by the vertices of the network act in a selfish way. This equilibrium situation occurs if no coalition of the players can block the underlying flow f, and this blocking is defined by a certain f-unsaturated path (or cycle through s or t) along which the players are capable and prefer to increase the flow. However, in some sense an f-unsaturated cycle C per se causes instability because the players of C mutually agree to send some extra flow along C. So it is natural to define flow f of network (D, s, t, c) with preferences to be *completely stable* if f is stable and there exists no f-unsaturated cycle in D whatsoever. If f is a stable flow then we can "augment" along f-unsaturated cycles, and hence we can construct a flow $f' \geq f$ such that there no longer exists an f'-unsaturated cycle. But unfortunately flow f' might not be stable any more because we might have created a blocking walk by the cycle-augmentations.

In fact, there exist networks with preferences that do not have a completely stable flow. One example is on the figure: each arc has unit capacity, preferences are indicated around the vertices: lower rank is preferred to the higher.

As no arc leaves subset $U := \{a, b, c\}$ of the vertices, no flow can leave U, hence no flow enters U. In particular, arc sa has zero flow. If we assume indirectly that f is a completely stable flow then cycle abc cannot block, hence there must be a unit flow along it. But now path sa is blocking, a contradiction.



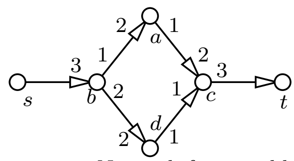
Stable flows have a blocking cycle

4 The Structure of Stable Flows

It is well-known about the stable marriage problem that in each stable marriage scheme, the same set of participants get married. That is, if someone does not get a marriage partner in some stable scheme then this very person remains single in each stable marriage schemes. A generalization of this is the rural hospital theorem of Roth [9] (see also Theorem 5.13 in [10]). It is about the college model, where instead of men we work with colleges, women correspond to students and each college has a quota on the maximum number of students. In the college admission problem, it is true that if a certain college c cannot fill up its quota in a stable admission scheme then c receives the same set of students in any stable admission scheme. (The phenomenon is named after the assignment problem of medical interns to hospitals.)

It seems that the rural hospital theorem cannot be generalized to the stable flow problem. It may happen in a network that a certain vertex transmits different amounts of flow in two stable flows.

An example is shown in the figure where each arc has unit capacity. There are two stable flows: one is along path *sbact* and the other follows path *sbdct*. So in one stable flow, vertex a transmits unit flow and no flow passes through a in the other one.



Network for a stable flow

There is however a consequence of the rural hospital theorem that can be generalized, namely, that the size of a stable matching is always the same. We have seen that the stable allocation problem is a special case of the stable flow problem, and from the construction it is apparent that the size of a stable matching (more precisely the total amount of assignments in a stable allocation) equals the value of the corresponding flow.

Theorem 3. *If network (D, s, t, c) and preference orders $<_v$ describe a stable flow problem and f_1 and f_2 are stable flows then the value of f_1 and f_2 are the same. More generally, $f_1(a) = f_2(a)$ for any arc of D that is incident to s (or to t).*

Proof. Lemma 1 implies that there exist stable allocations g_1 and g_2 of G_D that correspond to stable flows f_1 and f_2, respectively. The value of a flow is the net amount that leaves s in D, or, in G_D one can calculate it as the difference of total allocation of s^{out} and s^{in}. This means that the second part of the theorem implies the first one.

As there is no edge between s^{out} and s^{in}, the choice of $q(s)$ implies that both s^{out} and s^{in} are g_1-unsaturated. Hence property (11) can hold neither for s^{in} nor for s^{out}. But Theorem 1 implies that g_1 and g_2 are $\leq_{s^{out}}$ and $\leq_{s^{in}}$-comparable. So property (10) must be true for both flows g_1 and g_2 for vertices $v = s^{out}$ and $v = s^{in}$. This shows the second part of the Theorem for s. The argument for t is analogous to the above one.

As we have seen in Theorem 1, stable allocations have a lattice structure. Based on the connection of stable allocations and stable flows described in Lemma 1, we can prove that stable flows of a network with preferences also form a natural lattice. So assume that f is a stable flow in network $(D, s, t, c,)$ with preferences and let stable allocation g_f of G_D be the canonical representation of f as in the proof of Lemma 1.

Observe that any vertex $v \neq s, t$ of D, exactly one of $g_f(v^{in}v^{out})$ and $g_f(v^{out}v^{in})$ is positive by the choice of q and g_f. For stable flow f, we can classify the vertices of D different from s and t: v is an *f-vendor* if $g_f(v^{in}v^{out}) > 0$ and v is an *f-customer* if $g_f(v^{out}v^{in}) > 0$. If v is an f-vendor then no edge $v^{out}u^{in}$ can be g_f-dominated at v^{out} (as $g_f(v^{in}v^{out}) > 0$), hence player v sends as much flow to other vertices as much they accept. Similarly, if v is an f-customer then no edge $u^{out}v^{in}$ can be g_f-dominated at v^{out}, that is, player v receives as much flow as the others can supply her.

To explore the promised lattice structure of stable flows, let f_1 and f_2 two stable flows with canonical representations g_{f_1} and g_{f_2}, respectively. From Theorem 1 we know that stable allocations form a lattice, so $g_{f_1} \vee g_{f_2}$ and $g_{f_1} \wedge g_{f_2}$ are also stable allocations of G_D, and by Theorem 2, these stable allocations define stable flows $f_1 \vee f_2$ and $f_1 \wedge f_2$, respectively. How can we determine these latter flows directly, without the canonical representations? To answer this, we translate the lattice property of stable allocations on G_D to stable flows of D.

Theorem 3 shows that stable flows cannot differ on arcs incident to s or t, so on these arcs $f_1 \vee f_2$ and $f_1 \wedge f_2$ are determined. However, vertices different from s and t may have completely different situations in stable flows f_1 and f_2. The two colour classes of graph G_D are formed by the v^{in} and v^{out} type vertices, respectively. So, by Theorem 1, $g_{f_1} \vee g_{f_2}$ can be determined such that (say) each vertex v^{out} selects the better allocation and each vertex v^{in} receives the worse allocation out of the ones that g_{f_1} and g_{f_2} provides them. Similarly, for stable allocation $g_{f_1} \wedge g_{f_2}$ the "in"-type vertices choose according to their preferences and the "out"-type ones are left with the less preferred allocations. This means the following in the language of flows. If we want to construct $f_1 \vee f_2$ and v is a vertex different from s and t then either all arcs entering v will have the same flow in $f_1 \vee f_2$ as in f_1, or for all arcs a entering v we have $(f_1 \vee f_2)(a) = f_2(a)$ holds. A similar statement is true for the arcs leaving v. To determine which of the two alternatives is the right one, the following rules apply:

- If v is an f_1-vendor and an f_2-customer then v chooses f_2. If v is an f_2-vendor and an f_1-customer then v chooses f_1. That is, each vertex strives to be a customer.
- If v is an f_1-vendor and an f_2-vendor and v transmits more flow in f_1 than in f_2 (i.e. $0 < g_{f_1}(v^{in}v^{out}) < g_{f_2}(v^{in}v^{out})$) then v chooses f_1. That is, vendors prefer to sell more.
- If v is an f_1-customer and an f_2-customer and v transmits more flow in f_1 than in f_2 (i.e. $0 < g_{f_1}(v^{out}v^{in}) < g_{f_2}(v^{out}v^{in})$) then v chooses f_2. That is, customers prefer to buy less.
- Otherwise v is a customer in both f_1 and f_2 or v is a vendor in both flows and v transmits the same amount in both flows (i.e. $g_{f_1}(v^{out}v^{in}) = g_{f_2}(v^{out}v^{in})$ and $g_{f_1}(v^{in}v^{out}) = g_{f_2}(v^{in}v^{out})$). In this situation, v chooses the better "selling position" and gets the worse "buying position" out of stable flows f_1 and f_2.

Clearly, for the construction of $f_1 \wedge f_2$, one always has to choose the "other" options than the one that the above rules describe.

The lattice structure of stable flows defines a partial order on stable flows: $f_1 \preceq f_2$ if and only if $f_1 \vee f_2 = f_2$ holds, or equivalently, if $f_1 \wedge f_2 = f_1$ is true. By to the above rules, this means that each f_1-customer v is an f_2-customer, such that v buys at least as much in f_1 as in f_2. Each f_2-vendor u is an f_1-vendor and u sells at most as much in f_1 as in f_2. If w plays the same role (vendor or customer) in both flows and transmits the same amount then v prefers the selling position of f_2 and the buying position of f_1.

Acknowledgment. The author kindly acknowledges the support of the EGRES.

References

1. Baïou, M., Balinski, M.: The stable allocation (or ordinal transportation) problem. Math. Oper. Res. 27(3), 485–503 (2002)
2. Cechlárová, K., Fleiner, T.: On a generalization of the stable roommates problem. ACM Trans. Algorithms 1(1), 143–156 (2005)
3. Dean, B.C., Munshi, S.: Faster algorithms for stable allocation problems. In: Proceedings of the MATCH-UP (Matching Under Preferences) Workshop at ICALP 2008, Reykjavik, pp. 133–144 (2008)
4. Fleiner, T.: A fixed point approach to stable matchings and some applications. Mathematics of Operations Research 28(1), 103–126 (2003)
5. Fleiner, T.: On stable matchings and flows. Technical Report TR-2009-11, Egerváry Research Group, Budapest (2009), `http://www.cs.elte.hu/egres`
6. Gale, D., Shapley, L.S.: College admissions and stability of marriage. Amer. Math. Monthly 69(1), 9–15 (1962)
7. Knuth, D.E.: Stable marriage and its relation to other combinatorial problems. American Mathematical Society, Providence (1997); An introduction to the mathematical analysis of algorithms, Translated from the French by Martin Goldstein and revised by the author
8. Ostrovsky, M.: Stability in supply chain networks. American Economic Review 98(3), 897–923 (2006)
9. Roth, A.E.: On the allocation of residents to rural hospitals: a general property of two-sided matching markets. Econometrica 54(2), 425–427 (1986)
10. Roth, A.E., Oliveria Sotomayor, M.A.: Two-sided matching. Cambridge University Press, Cambridge (1990); A study in game-theoretic modeling and analysis, With a foreword by Robert Aumann
11. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific J. of Math. 5, 285–310 (1955)