No Justified Complaints: A Bottleneck-Based Fair Resource Sharing

Danny Dolev, Dror Feitelson, Joe Halpern,
Nati Linial, and Raz Kupferman

iAGT May '11

The basic setup:

N *users* are sharing m *resources*.

The basic setup:

N *users* are sharing m *resources*.

- There is a unit supply of each resource (a harmless normalization).

The basic setup:

N *users* are sharing m *resources*.

- There is a unit supply of each resource (a harmless normalization).
- Each user i has an entitlement $e_i > 0$ with $\sum_{i=1...N} e_i = 1$. (More on what these entitlements are - below).

The basic setup (contd.)

Requests: For every i, j the number $r_{ij} \geq 0$ is the quantity of resource j that the i-th user is requesting.

The basic setup (contd.)

Requests: For every i, j the number $r_{ij} \geq 0$ is the quantity of resource j that the i-th user is requesting.
No exchanges: In serving user i the proportions between the numbers $\{r_{ij}\}_{j=1,\ldots,m}$ must be respected.

The basic setup (contd.)

Requests: For every i, j the number $r_{ij} \geq 0$ is the quantity of resource j that the i-th user is requesting.

No exchanges: In serving user i the proportions between the numbers $\{r_{ij}\}_{j=1,\dots,m}$ must be respected.

In words: Users are not willing (are not permitted?) to substitute one resource for another.

The basic setup (contd.)

Requests: For every i, j the number $r_{ij} \geq 0$ is the quantity of resource j that the i-th user is requesting.

No exchanges: In serving user i the proportions between the numbers $\{r_{ij}\}_{j=1,\dots,m}$ must be respected. In words: Users are not willing (are not permitted?) to substitute one resource for another.

Consequently, for every i we need to select some $x_i > 0$ and give user i exactly $x_i r_{ij}$ of resource j, for every j.

The basic setup (contd.)

Requests: For every i, j the number $r_{ij} \geq 0$ is the quantity of resource j that the i-th user is requesting.

No exchanges: In serving user i the proportions between the numbers $\{r_{ij}\}_{j=1,\ldots,m}$ must be respected. In words: Users are not willing (are not permitted?) to substitute one resource for another.

Consequently, for every i we need to select some $x_i > 0$ and give user i exactly $x_i r_{ij}$ of resource j, for every j.

The fruit-salad metaphor.

So, we are seeking positive real numbers $x_i > 0$ for $i = 1, \ldots, N$. These numbers determine the resource allocation:
For every $N \geq i \geq 1, m \geq j \geq 1$, player i is to receive $x_i r_{ij}$ units of resource j.

So, we are seeking positive real numbers $x_i > 0$ for $i = 1, \ldots, N$. These numbers determine the resource allocation:
For every $N \geq i \geq 1, m \geq j \geq 1$, player i is to receive $x_i r_{ij}$ units of resource j.
The obvious *feasibility* condition says that we must not exceed the unit capacity of the various resources. Namely, for every resource j

$$\sum_i x_i r_{ij} \leq 1$$

So, we are seeking positive real numbers $x_i > 0$ for $i = 1, \ldots, N$. These numbers determine the resource allocation:
For every $N \geq i \geq 1, m \geq j \geq 1$, player i is to receive $x_i r_{ij}$ units of resource j.
The obvious *feasibility* condition says that we must not exceed the unit capacity of the various resources. Namely, for every resource j

$$\sum_i x_i r_{ij} \leq 1$$

In matrix notation:

$$x > 0 \qquad xR \leq 1$$

What are those entitlements?

These numbers might represent various things:

- A user's share in the computers' installation.

What are those entitlements?

These numbers might represent various things:

- A user's share in the computers' installation.
- An externally defined level of priority.

What are those entitlements?

These numbers might represent various things:

- A user's share in the computers' installation.
- An externally defined level of priority.
- Alternatively, the number of users is actually $M \gg N$ and all users are treated equally, i.e. all entitlements are $\frac{1}{M}$. However, there are only N *types* of users and e_i is the fraction of users of type i.

Bottlenecks

For a given request matrix R and a feasible $x > 0$ (i.e. $xR \leq 1$), we say that resource j is a bottleneck if it is consumed in full, i.e., if

$$(xR)_j = 1.$$

Bottleneck-based fairness criterion

Given a request matrix R, an entitlement vector e and a feasible x, let J be the set of bottleneck resources. We say that x is fair if for every user i, either

- $x_i = 1$ (so that user i is receiving his request in full), or

Bottleneck-based fairness criterion

Given a request matrix R, an entitlement vector e and a feasible x, let J be the set of bottleneck resources. We say that x is fair if for every user i, either

- $x_i = 1$ (so that user i is receiving his request in full), or
- There is some bottleneck resource $j \in J$ for which $x_i r_{ij} \geq e_i$.

Bottleneck-based fairness means that there are No Justified Complaints

The criterion says that if a user is receiving less than his total request, then:

Bottleneck-based fairness means that there are No Justified Complaints

The criterion says that if a user is receiving less than his total request, then:
There is a resource on which user i is receiving at least his entitlement.

Bottleneck-based fairness means that there are No Justified Complaints

The criterion says that if a user is receiving less than his total request, then:
There is a resource on which user i is receiving at least his entitlement. Moreover, this is a bottleneck resource, so increasing i-th share will necessarily hurt other users. In other words, player i has no grounds for a justified complaint.

This is *not* a linear program

For a fixed set J of bottleneck resources, a fair and feasible x can be found by solving a linear program. Consequently, the whole problem is solvable in time $\exp(O(m))$ where m is the number of resources.

This is *not* a linear program

For a fixed set J of bottleneck resources, a fair and feasible x can be found by solving a linear program. Consequently, the whole problem is solvable in time $\exp(O(m))$ where m is the number of resources. Without knowledge of J, the problem cannot be solved using LP.

This is *not* a linear program

For a fixed set J of bottleneck resources, a fair and feasible x can be found by solving a linear program. Consequently, the whole problem is solvable in time $\exp(O(m))$ where m is the number of resources. Without knowledge of J, the problem cannot be solved using LP.

A little more on this - at the end of the talk.

An example

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

An example

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

All $e_i = 1/4$.

An example

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

All $e_i = 1/4$.
$x = (1/4, 1/4, 3/8, 3/8)$ is feasible and fair with
$J = \{1, 2\}$,

An example

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

All $e_i = 1/4$.
$x = (1/4, 1/4, 3/8, 3/8)$ is feasible and fair with $J = \{1, 2\}$,
Same for $x' = (3/8, 3/8, 1/4, 1/4)$, with $J = \{3, 4\}$.

An example

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

All $e_i = 1/4$.
$x = (1/4, 1/4, 3/8, 3/8)$ is feasible and fair with $J = \{1, 2\}$,
Same for $x' = (3/8, 3/8, 1/4, 1/4)$, with $J = \{3, 4\}$.
Their average $\frac{x+x'}{2} = (5/16, 5/16, 5/16, 5/16)$ is feasible but not fair.

Main result

Theorem

For every matrix R with $1 \geq r_{ij} \geq 0$ for all i, j

Main result

Theorem

For every matrix R *with* $1 \geq r_{ij} \geq 0$ *for all* i, j *and for every* $e > 0$ *with* $\sum e_i = 1$,

Main result

Theorem

For every matrix R *with* $1 \geq r_{ij} \geq 0$ *for all* i, j *and for every* $e > 0$ *with* $\sum e_i = 1$, *there exists a vector* $x > 0$ *such that* $xR \leq 1$

Main result

Theorem

For every matrix R with $1 \geq r_{ij} \geq 0$ for all i, j and for every $e > 0$ with $\sum e_i = 1$, there exists a vector $x > 0$ such that $xR \leq 1$ so that for every i there is a $j \in J$ with

$$x_i r_{ij} \geq e_i$$

where $J = \{j | (xR)_j = 1\}$.

A little geometric insight

The constraints $x > 0$ and $xR \leq 1$ define a convex polytope P that resides in the positive quadrant of the N-dimensional space $\mathbb{R}^N$.

A little geometric insight

The constraints $x > 0$ and $xR \leq 1$ define a convex polytope P that resides in the positive quadrant of the N-dimensional space $\mathbb{R}^N$.

Corresponding to each constraint is a hyperplane that constitutes a facet of P.

A little geometric insight

The constraints $x > 0$ and $xR \leq 1$ define a convex polytope P that resides in the positive quadrant of the N-dimensional space $\mathbb{R}^N$.
Corresponding to each constraint is a hyperplane that constitutes a facet of P. (Actually, this is a bit inaccurate, and some resource may be defining a redundant inequality, but in this case we pre-process the problem and eliminate any such resources).

A little geometric insight (contd.)

The feasibility of x is equivalent to $x \in P$.

A little geometric insight (contd.)

The feasibility of x is equivalent to $x \in P$.
In order for x to be fair it is necessary that x belongs to $\partial^+ P$, the upper boundary of P.

A little geometric insight (contd.)

The feasibility of x is equivalent to $x \in P$.
In order for x to be fair it is necessary that x belongs to $\partial^+ P$, the upper boundary of P.
Recall that the set J of bottleneck resources depends on x. Actually it is defined by the set of facets on which x resides. (In every vertex of P several facets meet).

So what is it that complicates matters?

The boundary ∂P may, in general, be a complicated combinatorial object (the so-called *face lattice* of the polytope P). We do not have very good methods of searching through it.

Continuous mathematics to the rescue?

The domain P that's defined by $x > 0$ and $xR \leq 1$ is convex and *closed down* (since the matrix R is nonnegative).

Continuous mathematics to the rescue?

The domain P that's defined by $x > 0$ and $xR \leq 1$ is convex and *closed down* (since the matrix R is nonnegative).
If it's the combinatorial complexity that complicates the situation, let us try to simplify matters by considering, rather than P, another domain Q in the first quadrant of $\mathbb{R}^N$ that is

Continuous mathematics to the rescue?

The domain P that's defined by $x > 0$ and $xR \leq 1$ is convex and *closed down* (since the matrix R is nonnegative).

If it's the combinatorial complexity that complicates the situation, let us try to simplify matters by considering, rather than P, another domain Q in the first quadrant of $\mathbb{R}^N$ that is

- Convex

Continuous mathematics to the rescue?

The domain P that's defined by $x > 0$ and $xR \leq 1$ is convex and *closed down* (since the matrix R is nonnegative).
If it's the combinatorial complexity that complicates the situation, let us try to simplify matters by considering, rather than P, another domain Q in the first quadrant of $\mathbb{R}^N$ that is

- Convex
- Closed down

Continuous mathematics to the rescue?

The domain P that's defined by $x > 0$ and $xR \leq 1$ is convex and *closed down* (since the matrix R is nonnegative).
If it's the combinatorial complexity that complicates the situation, let us try to simplify matters by considering, rather than P, another domain Q in the first quadrant of $\mathbb{R}^N$ that is

- Convex
- Closed down
- Has a smooth upper boundary $\partial^+ Q$

But how?

- How can you even describe such a smooth domain?

But how?

- How can you even describe such a smooth domain?
- How do you test feasibility?

But how?

- How can you even describe such a smooth domain?
- How do you test feasibility?
- What form does fairness take in this context?

But how?

- How can you even describe such a smooth domain?
- How do you test feasibility?
- What form does fairness take in this context?

Note that in the original setup feasibility can be reformulated as $x > 0$ and $\Phi(x) \leq 1$, where

But how?

- How can you even describe such a smooth domain?
- How do you test feasibility?
- What form does fairness take in this context?

Note that in the original setup feasibility can be reformulated as $x > 0$ and $\Phi(x) \leq 1$, where $\Phi(x) := \max_j (xR)_j$.

Dealing with a smooth domain Q

We define Q as $\{x \in \mathbb{R}^N | f(x) \leq 1, x > 0\}$,

Dealing with a smooth domain Q

We define Q as $\{x \in \mathbb{R}^N | f(x) \leq 1, x > 0\}$, where f is

- Concave

Dealing with a smooth domain Q

We define Q as $\{x \in \mathbb{R}^N | f(x) \leq 1, x > 0\}$, where f is

- Concave
- Positive in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$

Dealing with a smooth domain Q

We define Q as $\{x \in \mathbb{R}^N | f(x) \leq 1, x > 0\}$, where f is

- Concave
- Positive in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$
- Monotone decreasing in each variable in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$

Dealing with a smooth domain Q

We define Q as $\{x \in \mathbb{R}^N | f(x) \leq 1, x > 0\}$, where f is

- Concave
- Positive in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$
- Monotone decreasing in each variable in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$
- Sufficiently smooth

Dealing with a smooth domain Q

We define Q as $\{x \in \mathbb{R}^N | f(x) \leq 1, x > 0\}$, where f is

- Concave
- Positive in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$
- Monotone decreasing in each variable in the vicinity of the origin in the first quadrant of $\mathbb{R}^N$
- Sufficiently smooth

In simple words: Q should look like P, but have a smooth upper boundary.

Working with a smooth domain

Testing feasibility of x is easy. We only need to verify that $x > 0$ and that $f(x) \leq 1$.

Working with a smooth domain

Testing feasibility of x is easy. We only need to verify that $x > 0$ and that $f(x) \leq 1$.
But what about the bottleneck condition?

Working with a smooth domain

Testing feasibility of x is easy. We only need to verify that $x > 0$ and that $f(x) \leq 1$.
But what about the bottleneck condition?
This is exactly where we gain:

Working with a smooth domain

Testing feasibility of x is easy. We only need to verify that $x > 0$ and that $f(x) \leq 1$.
But what about the bottleneck condition?
This is exactly where we gain: Membership in Q is defined by (uncountably many) inequalities, one per each hyperplane that is tangent to $\partial^+ Q$. In particular, every $x \in \partial^+ Q$, is automatically feasible.

In a smooth world

Let us express the hyperplane H that's tangent to $\partial^+ Q$ at x as $H = \{y \in \mathbb{R}^N | < a, y >= 1\}$.

In a smooth world

Let us express the hyperplane H that's tangent to $\partial^+ Q$ at x as $H = \{y \in \mathbb{R}^N | < a, y >= 1\}$. Here a is (the appropriately normalized) normal to the surface $\partial^+ Q$ at x.

In a smooth world

Let us express the hyperplane H that's tangent to $\partial^+ Q$ at x as $H = \{y \in \mathbb{R}^N | < a, y >= 1\}$.
Here a is (the appropriately normalized) normal to the surface $\partial^+ Q$ at x.
Every point $z \in Q$ satisfies $< a, z > \leq 1$, (z lies *below* the hyperplane H).

In a smooth world

Let us express the hyperplane H that's tangent to $\partial^+ Q$ at x as $H = \{y \in \mathbb{R}^N | < a, y >= 1\}$.
Here a is (the appropriately normalized) normal to the surface $\partial^+ Q$ at x.
Every point $z \in Q$ satisfies $< a, z > \leq 1$, (z lies *below* the hyperplane H). Among Q's points x is uniquely determined by the condition $< a, x >= 1$.

In a smooth world (contd.)

In words, the case of a smooth domain can be viewed as a variation of the original problem:

In a smooth world (contd.)

In words, the case of a smooth domain can be viewed as a variation of the original problem: There are uncountably many resources, one per each tangent plane of Q.

In a smooth world (contd.)

In words, the case of a smooth domain can be viewed as a variation of the original problem: There are uncountably many resources, one per each tangent plane of Q. For a feasible solution in the interior of Q, there are no bottleneck resources.

In a smooth world (contd.)

In words, the case of a smooth domain can be viewed as a variation of the original problem: There are uncountably many resources, one per each tangent plane of Q. For a feasible solution in the interior of Q, there are no bottleneck resources. However, at a boundary point $x \in \partial^+ Q$, there is exactly one bottleneck resource.

In a smooth world (contd.)

In words, the case of a smooth domain can be viewed as a variation of the original problem: There are uncountably many resources, one per each tangent plane of Q. For a feasible solution in the interior of Q, there are no bottleneck resources. However, at a boundary point $x \in \partial^+ Q$, there is exactly one bottleneck resource. Namely, the one corresponding to the tangent plane of Q at x.

In a smooth world (contd.)

In words, the case of a smooth domain can be viewed as a variation of the original problem: There are uncountably many resources, one per each tangent plane of Q. For a feasible solution in the interior of Q, there are no bottleneck resources. However, at a boundary point $x \in \partial^{+}Q$, there is exactly one bottleneck resource. Namely, the one corresponding to the tangent plane of Q at x.

Actually, I am cheating here a little - I have shoved under the rug the possibility of $x_i = 1$, but it's not a very serious lie.

What have we gained?

The combinatorial complexity is no longer an issue.

What have we gained?

The combinatorial complexity is no longer an issue. Moreover, fairness has become an equation rather than an inequality.

What have we gained?

The combinatorial complexity is no longer an issue. Moreover, fairness has become an equation rather than an inequality.
Now there is a unique bottleneck resource - the one corresponding to the tangent to $\partial^+ Q$ at x.

What have we gained?

The combinatorial complexity is no longer an issue. Moreover, fairness has become an equation rather than an inequality.
Now there is a unique bottleneck resource - the one corresponding to the tangent to $\partial^+ Q$ at x.Therefore the fairness condition reads

$$\forall i = 1, \ldots, N \qquad x_i a_i \geq e_i$$

What have we gained?

The combinatorial complexity is no longer an issue. Moreover, fairness has become an equation rather than an inequality.
Now there is a unique bottleneck resource - the one corresponding to the tangent to $\partial^+ Q$ at x.Therefore the fairness condition reads

$$\forall i = 1, \ldots, N \quad x_i a_i \geq e_i$$

where a is the normalized normal to $\partial^+ Q$ at x and e_i are the entitlements.

What have we gained? (contd.)

Summing over all i this becomes

$$< a, x > \geq \sum e_i .$$

What have we gained? (contd.)

Summing over all i this becomes

$$< a, x > \geq \sum e_i .$$

But, as we know $< a, x > = 1$ and $\sum e_i = 1$ as well.

What have we gained? (contd.)

Summing over all i this becomes

$$< a, x > \geq \sum e_i.$$

But, as we know $< a, x >= 1$ and $\sum e_i = 1$ as well. It follows that we must have equalities throughout.

What have we gained? (contd.)

Summing over all i this becomes

$$< a, x > \geq \sum e_i.$$

But, as we know $< a, x > = 1$ and $\sum e_i = 1$ as well. It follows that we must have equalities throughout. In words, what we are seeking is a vector $x > 0$ with $f(x) = 1$ and

$$\forall i \quad a_i x_i = c \cdot e_i$$

for some $c > 0$. Here a is a vector normal to $\partial^+ Q$ at x and e_i are the entitlements.

What f should we choose?

The most obvious choice of f (and thus of the domain Q) is

$$f(x) = -\sum_j \log(1 - (xR)_j)$$

It is positive near the origin of the first quadrant of $\mathbb{R}^N$.

What f should we choose?

The most obvious choice of f (and thus of the domain Q) is

$$f(x) = -\sum_j \log(1 - (xR)_j)$$

It is positive near the origin of the first quadrant of $\mathbb{R}^N$. It tends to ∞ as we approach the boundary of P (where some of the inequalities $(xR)_j \leq 1$ become equalities).

The general approach

We do not define Q right away. For every $t > 0$ we consider the domain

$$Q_t = \{x > 0 | f(x) \leq t\}$$

The general approach

We do not define Q right away. For every $t > 0$ we consider the domain

$$Q_t = \{x > 0 | f(x) \leq t\}$$

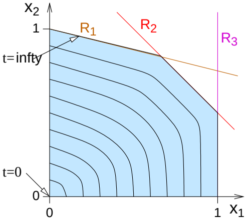


Figure: *Illustration of level-sets of f from $t = 0$ to $t = \infty$.*

Putting everything together

On the boundary ∂^+ of the domain Q_t we find (using standard material from the theory of ordinary differential equations) a point x such that there is a constant $c > 0$ for which

$$\forall i \quad n_i x_i = c \cdot e_i$$

where n is the (l_2-normalized) normal at x to the surface $f(x) = t$.

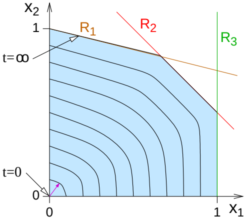
Putting everything together

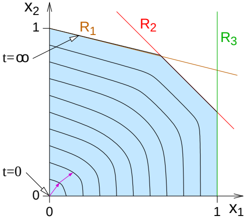
On the boundary ∂^+ of the domain Q_t we find (using standard material from the theory of ordinary differential equations) a point x such that there is a constant $c > 0$ for which

$$\forall i \quad n_i x_i = c \cdot e_i$$

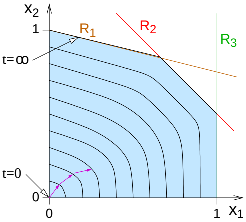
where n is the (l_2-normalized) normal at x to the surface $f(x) = t$.
Finally we pass to a limit as $t \to \infty$.

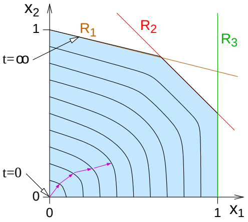
Solving the differential equation

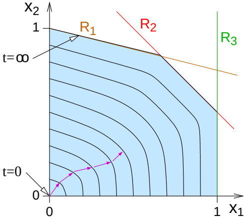




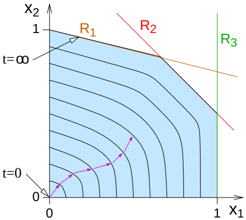
x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1

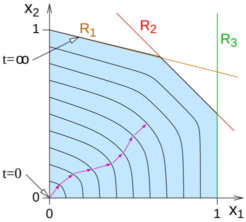

x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1


x_2
1
R_1
R_2
R_3
t=∞
t=0
0
0
1
x_1

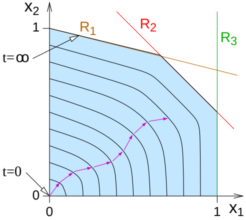


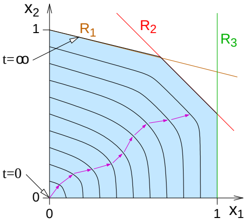
x_2
1
R_1
R_2
R_3
t=∞
t=0
0
0
1
x_1


x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1

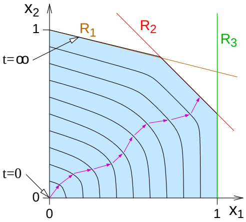


x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1


x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1



x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1


x2
1
R1
R2
R3
t=∞
t=0
0
0
1
x1

What is still missing?

We still do not know how to turn the numerical solution of the ordinary differential equation into a polytime algorithm. (I am quite certain that this can be done and should not be too difficult).

What is still missing?

We still do not know how to turn the numerical solution of the ordinary differential equation into a polytime algorithm. (I am quite certain that this can be done and should not be too difficult). What do the solutions look like? As mentioned above, for each fixed J (set of bottleneck resources), the problem becomes a Linear Program. In particular, for this reason, in the generic case, the solution set is a discrete set of points. We can consider various criteria to tell which of these solutions is more suitable.

Some more unresolved issues

Specifically, can we guarantee the existence of a fair solution with a high utilization of resources? To answer this question positively, we certainly have to limit the feasible solution which we consider acceptable (*nearly fair* in some sense).

Some more unresolved issues

Specifically, can we guarantee the existence of a fair solution with a high utilization of resources? To answer this question positively, we certainly have to limit the feasible solution which we consider acceptable (*nearly fair* in some sense).
Recall that the choice of f in our solution was quite arbitrary. Perhaps some clever choice of f can help.

That's all, folks