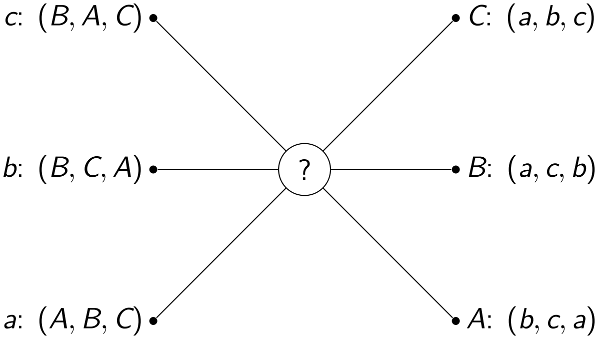
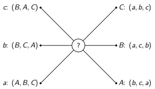
Matching Games, Routing Games and Cooperative Game Theory

Vince Knight

2024-03-04 until 2024-03-29


Altes Amtsgericht
1906
Swakopmund


c: (B, A, C)
b: (B, C, A)
a: (A, B, C)
?
C: (a, b, c)
B: (a, c, b)
A: (b, c, a)



Let us start with a slightly different type of game.

The set of small letters on the left wants to be matched with the sets of small letter on the right.

Each of their preference is indicated in the brackets. For example *a* would prefer to be matched with *A* over *B* over *C*.

Can you identify a potential matching?

Definition

A matching game of size N is defined by two disjoint sets S and R or suitors and reviewers of size N. Associated to each element of S and R is a preference list:

$$f : S \to R^N \text{ and } g : R \to S^N$$

Definition

A matching game of size N is defined by two disjoint sets S and R or suitors and reviewers of size N. Associated to each element of S and R is a preference list:

$$f : S \to R^N \text{ and } g : R \to S^N$$

So this is a formal definition of the situation we had above. We had $N = 3$.

$$S = \{a, b, c\}$$

and

$$R = \{A, B, C\}$$

Definition

A matching M is any bijection between S and R. If $s \in S$ and $r \in R$ are matched by M we denote:

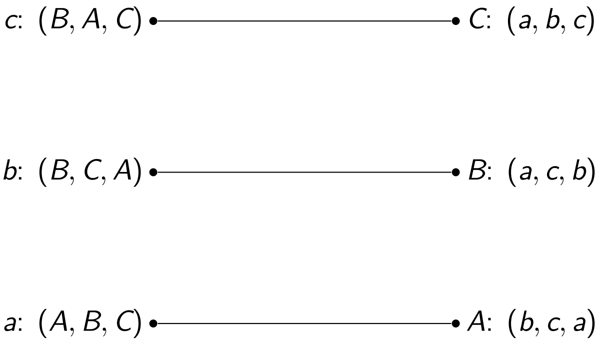
$$M(s) = r$$

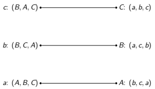
Definition

A matching M is any bijection between S and R. If $s \in S$ and $r \in R$ are matched by M we denote:

$$M(s) = r$$

Here is our formal definition of what we are looking for: a matching.


c: (B, A, C)
C: (a, b, c)
b: (B, C, A)
B: (a, c, b)
a: (A, B, C)
A: (b, c, a)



Here is a matching.
If you look closely you might notice that there is some instability here.
c would prefer to be matched with B over their current matching **and** B would prefer to be matched with c over their current matching.

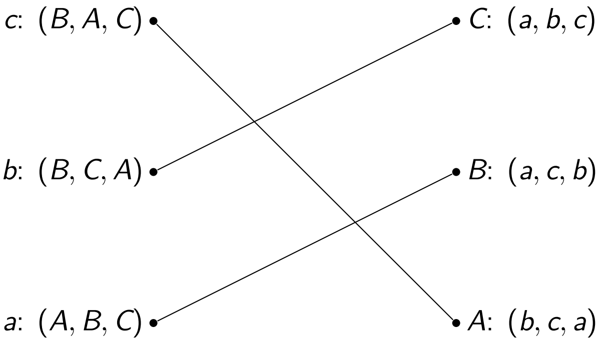
Definition

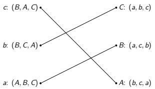
A pair (s, r) is said to **block** a matching M if $M(s) \neq r$ but s prefers r to $M(r)$ and r prefers s to $M^{-1}(r)$.

Definition

A pair (s, r) is said to **block** a matching M if $M(s) \neq r$ but s prefers r to $M(r)$ and r prefers s to $M^{-1}(r)$.

In the case of the proposed matching (c, B) is a blocking pair.


c: (B, A, C)
b: (B, C, A)
a: (A, B, C)
C: (a, b, c)
B: (a, c, b)
A: (b, c, a)



This is a stable matching.

Can we always find one?


Lloyd Shapley[1] (1923 - 2016) and **David Gale** (1921 - 2008)
By Bengt Nyman - Flickr: IMG_4826, CC BY 2.0,
https://commons.wikimedia.org/w/index.php?curid=23083494
By George M. Bergman, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=103840310

[1] David Gale and Lloyd S Shapley. "College admissions and the stability of marriage". In: *The American Mathematical Monthly* 69.1 (1962), pp. 9–15.

Lloyd Shapley[1] (1923 - 2016) and **David Gale** (1921 - 2008)

[1] David Gale and Lloyd S Shapley. "College admissions and the stability of marriage". In: *The American Mathematical Monthly* 69.1 (1962), pp. 9–15.

The answer is yes and Shapley and Gale published the Gale Shapley algorithm.

Definition

1. Assign every $s \in S$ and $r \in R$ to be unmatched
2. Pick some unmatched $s \in S$, let r be the top of s's preference list:
    1. If r is unmatched set $M(s) = r$
    2. If r is matched:
        1. If r prefers s to $M^{-1}(r)$ then set $M(r) = s$
        2. Otherwise s remains unmatched and remove r from s's preference list.
3. Repeat step 2 until all $s \in S$ are matched.

Definition

1. Assign every $s \in S$ and $r \in R$ to be unmatched
2. Pick some unmatched $s \in S$, let r be the top of s's preference list:
    1. If r is unmatched set $M(s) = r$
    2. If r is matched:
        1. If r prefers s to $M^{-1}(r)$ then set $M(r) = s$
        2. Otherwise s remains unmatched and remove r from s's preference list.
3. Repeat step 2 until all $s \in S$ are matched.

This is the algorithm that they define and in fact there are number of theoretic results that hold for this algorithm. For example:

Theorem

All possible executions of the Gale-Shapley algorithm yield the same stable matching and in this stable matching every suitor has the best possible partner in any stable matching.

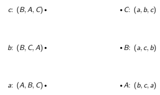
Theorem

In a suitor-optimal stable matching each reviewer has the worst possible matching.

c: (B, A, C) • • C: (a, b, c)

b: (B, C, A) • • B: (a, c, b)

a: (A, B, C) • • A: (b, c, a)

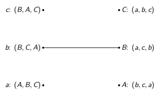


We pick b and as all the reviewers are unmatched set $M(b) = B$.

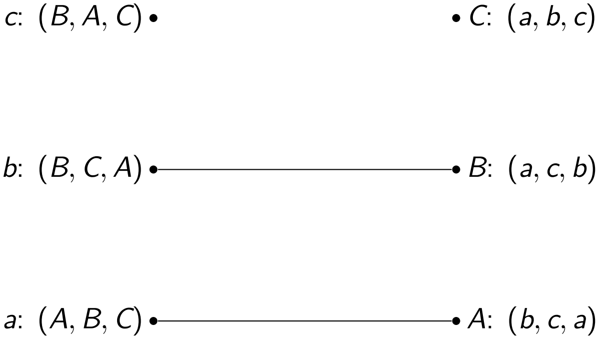
c: (B, A, C) •   • C: (a, b, c)

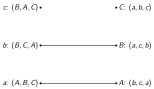
b: (B, C, A) •   • B: (a, c, b)

a: (A, B, C) •   • A: (b, c, a)

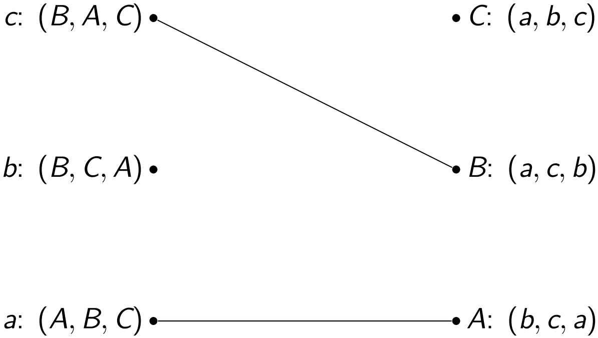


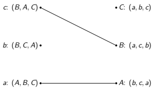
We pick a and as A is unmatched set $M(a) = A$.


c: (B, A, C)
C: (a, b, c)
b: (B, C, A)
B: (a, c, b)
a: (A, B, C)
A: (b, c, a)

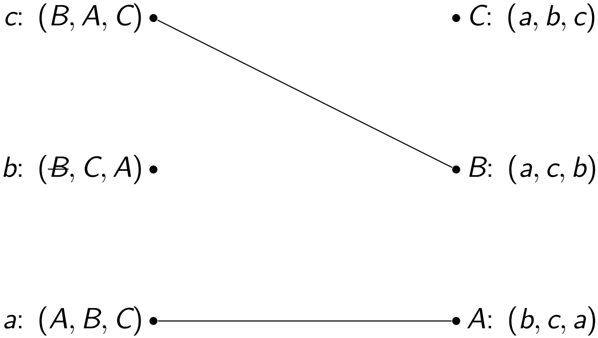


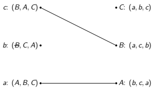
We pick c and b is matched but prefers c to $M^{-1}(B) = b$, we set $M(c) = B$.


c: (B, A, C)
C: (a, b, c)
b: (B, C, A)
B: (a, c, b)
a: (A, B, C)
A: (b, c, a)

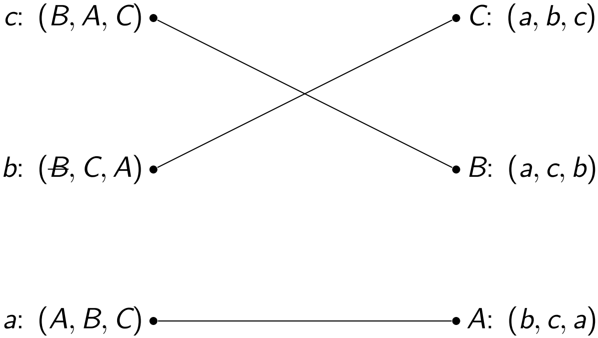


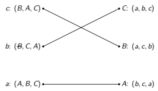
We pick b and as B is matched but prefers c to b we cross out B from b's preferences:


c: (B, A, C)
C: (a, b, c)
b: $(\cancel{B}, C, A)$
B: (a, c, b)
a: (A, B, C)
A: (b, c, a)

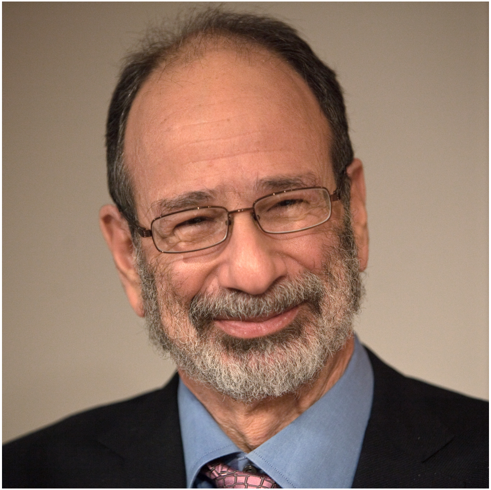


We pick b again and set $M(b) = C$.


c: (B, A, C)
C: (a, b, c)
b: $(\cancel{B}, C, A)$
B: (a, c, b)
a: (A, B, C)
A: (b, c, a)

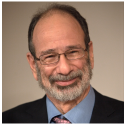


And this final matching is in fact stable.



Alvin Roth[2] (1951 -)

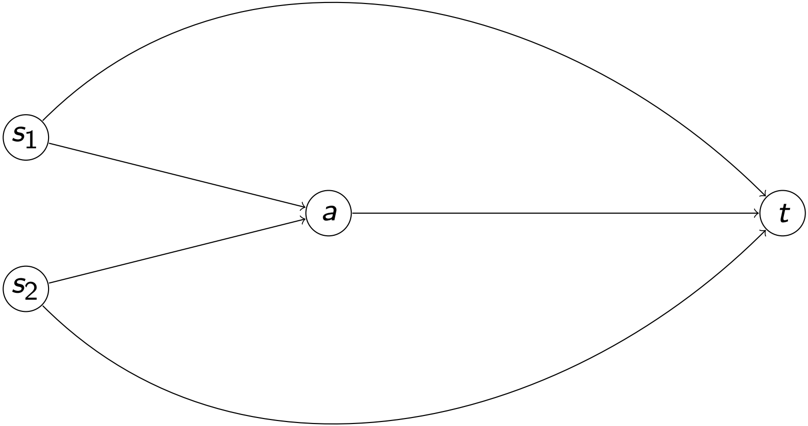
[2] Alvin E Roth and Marilda Sotomayor. "Two-sided matching". In: *Handbook of game theory with economic applications* 1 (1992), pp. 485–541.

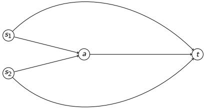


Alvin Roth[2] (1951 -)

[2] Alvin E Roth and Marilda Sotomayor. "Two-sided matching". In: *Handbook of game theory with economic applications* 1 (1992), pp. 485–541.

Alvin Roth here actually shared the nobel Prize in economics with Shapley. He used matching games to design markets for organ exchange as well as school choice and medical internship.


s_1
s_2
a
t



Here is another type of game. Imagine having an infinite amount of traffic that needs to get from the two sources here s_1, s_2 to the sink t. If of the roads (arcs on this network) have different amounts of congestion. Perhaps one of them is a road that can handle more traffic efficiently which makes it popular. Which in turn makes it slower.

This is what is called a routing game.

Definition

A **routing game** (G, r, c) is defined on a graph $G = (V, E)$ with a defined set of sources s_i and sinks t_i.

- Each source-sink pair corresponds to a set of traffic (also called a commodity) r_i that must travel along the edges of G from s_i to t_i.
- Every edge e of G has associated to it a nonnegative, continuous and nondecreasing cost function (also called latency function) c_e.

Definition

A **routing game** (G, r, c) is defined on a graph $G = (V, E)$ with a defined set of sources s_i and sinks t_i.

- Each source-sink pair corresponds to a set of traffic (also called a commodity) r_i that must travel along the edges of G from s_i to t_i.
- Every edge e of G has associated to it a nonnegative, continuous and nondecreasing cost function (also called latency function) c_e.

This is the formal definition of the routing game.

Definition

For a routing game (G, r, c) we define the optimal flow f^* as the solution to the following optimisation problem:

Minimise:

$$\sum_{e \in E} c_e(f_e) f_e$$

:

Subject to:

$$\sum_{P \in \mathcal{P}_i} f_P = r_i \qquad \text{for all } i$$

$$f_e = \sum_{P \in \mathcal{P} \text{ if } e \in P} f_P \qquad \text{for all } e \in E$$

$$f_P \geq 0$$

Definition

For a routing game (G, r, c) we define the optimal flow f^* as the solution to the following optimisation problem:

Minimise:

$$\sum_{e\in E} c_e(f_e)f_e$$

:

Subject to:

$$\begin{aligned} \sum_{P\in\mathcal{P}_i} f_P &= r_i && \text{for all } i \\ f_e &= \sum_{P\in\mathcal{P} \text{ if } e\in P} f_P && \text{for all } e \in E \\ f_P &\geq 0 \end{aligned}$$

This is the formal definition of the optimal flow. Importantly, this is actually an easy problem mathematically (it's a convex optimisation problem).

For a routing game (G, r, c) a flow $\tilde{f}$ is called a **Nash flow** if and only if for every commodity i and any two paths $P_1, P_2 \in \mathcal{P}_i$ such that $f_{P_1} > 0$ then:

Definition

If c is a differentiable cost function then we define the **marginal cost** function c^* as:

$$c^* = \frac{d}{dx}(xc(x))$$

Definition

If c is a differentiable cost function then we define the **marginal cost** function c^* as:

$$c^* = \frac{d}{dx}(xc(x))$$

This however is the definition of a Nash flow.
It is saying that if a particular path is used by traffic that that path must have minimal cost.

This implies that the equilibrium is when all traffic is routed along minimal costing paths. In other words: no one has a reason to deviate.

Theorem

For a routing game (G, r, c) the Nash flow $\tilde{f}$ is the solution to the following optimisation problem:

Minimise:

$$\Phi(f) = \sum_{e \in E} \int_0^{f_e} c_e(x) dx$$

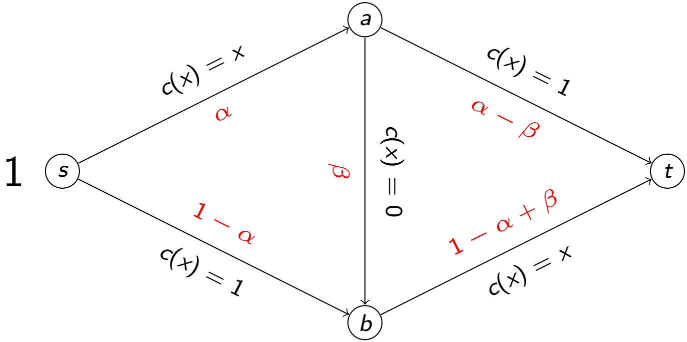
:

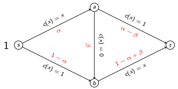
Subject to:

$$\sum_{P \in \mathcal{P}_i} f_P = r_i \qquad \text{for all } i$$

$$f_e = \sum_{P \in \mathcal{P} \text{ if } e \in P} f_P \qquad \text{for all } e \in E$$

$$f_P \geq 0$$


1
s
a
b
t
c(x) = x
α
c(x) = 1
α − β
β
c(x) = 0
1 − α
c(x) = 1
1 − α + β
c(x) = x

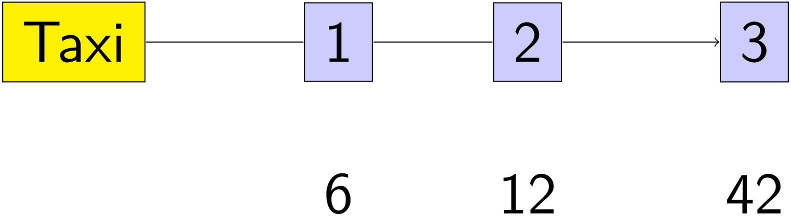


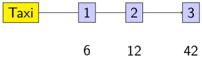
This is called Braess' Paradox.

The optimal flow in this case is for half the flow to go along the top and half the flow to go along the bottom.

The Paradox here is that by the addition of a resources (the super road) the congestion increases.

There are real life examples of this: for example the Big Dig in Boston.


Taxi
1
2
3
6
12
42



Now for another type of game: how should these three people share a taxi fare?

They have agreed to cooperate and now they want to know how to share their cost.

Definition

A **characteristic function game** G is given by a pair (N, v) where N is the number of players and $v : 2^{[N]} \to \mathbb{R}$.

Definition

A **characteristic function game** G is given by a pair (N, v) where N is the number of players and $v : 2^{[M]} \to \mathbb{R}$.

Here is the characteristic function game for the taxi fare: how much would the fare be for each coalition of players:

$$v(C) = \begin{cases} 6, & \text{if } C = \{1\} \\ 12, & \text{if } C = \{2\} \\ 42, & \text{if } C = \{3\} \\ 12, & \text{if } C = \{1,2\} \\ 42, & \text{if } C = \{1,3\} \\ 42, & \text{if } C = \{2,3\} \\ 42, & \text{if } C = \{1,2,3\} \end{cases}$$

- Efficiency;
- Null player;
- Symmetry;
- Additivity.

- Efficiency;
- Null player;
- Symmetry;
- Additivity.

Is there a way to share out the fare in a way that matches these four axiomatic properties:

- The entire fare is paid.
- People who are not in the taxi do not pay.
- If people go the same distance they pay the same amount.
- It doesn't matter which taxi is taken.

Lloyd Shapley[3] (1923 - 2016)
By Bengt Nyman - Flickr: IMG_4826, CC BY 2.0,
https://commons.wikimedia.org/w/index.php?curid=23083494

[3] Lloyd S Shapley. "Notes on the n-person game—ii: The value of an n-person game". In: (1951).

Lloyd Shapley[3] (1923 - 2016)
By Bengt Nyman - Flickr: IMG_4826, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=23083494

[3] Lloyd S Shapley. "Notes on the n-person game—ii: The value of an n-person game". In: (1951).

The answer is yes: Lloyd Shapley (him again) came up with an approach to do this.
It is now referred to as the Shapley value.

Given $G = (N, v)$ the **Shapley value** of player i is denoted by $\phi_i(G)$ and given by:

$$\phi_i(G) = \frac{1}{N!} \sum_{\pi \in \Pi_n} \Delta_\pi^G(i)$$

where:

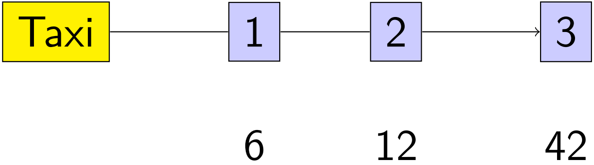
$$S_\pi(i) = \{j \in [N] \mid \pi(j) < \pi(i)\}$$

and

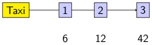
$$\Delta_\pi^G(i) = v(S_\pi(i) \cup i) - v(S_\pi(i))$$

The Shapley value is the average vector that corresponds to the marginal contributions of each player in any given permutation of the players.

Imagine letting people out of the taxi in random orders and they each pay what is owed at the time.



π	1	2	3
$1, 2, 3$	6	6	30
$1, 3, 2$	6	0	36
$2, 1, 3$	0	12	30
$2, 3, 1$	0	12	30
$3, 1, 2$	0	0	42
$3, 2, 1$	0	0	42
Average	2	5	35



π	1	2	3
1, 2, 3	6	6	30
1, 3, 2	6	0	36
2, 1, 3	0	12	30
2, 3, 1	0	12	30
3, 1, 2	0	0	42
3, 2, 1	0	0	42
Average	2	5	35

Here is how that calculation looks here.

We consider each permutation of the players and work out what they pay in that order.

Explainable AI[4]

Model	R^2
$y = c_1x_1$	0.075
$y = c_2x_2$	0.086
$y = c_3x_3$	0.629
$y = c_1x_1 + c_2x_2$	0.163
$y = c_1x_1 + c_3x_3$	0.63
$y = c_2x_2 + c_3x_3$	0.906
$y = c_1x_1 + c_2x_2 + c_3x_3$	0.907

$$\phi(G) = (0.0383, 0.1818, 0.6868)$$

[4] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

Explainable AI[4]

Model	R^2
$y = c_1x_1$	0.075
$y = c_2x_2$	0.086
$y = c_3x_3$	0.629
$y = c_1x_1 + c_2x_2$	0.163
$y = c_1x_1 + c_3x_3$	0.63
$y = c_2x_2 + c_3x_3$	0.906
$y = c_1x_1 + c_2x_2 + c_3x_3$	0.907

$$\phi(G) = (0.0383, 0.1818, 0.6868)$$

[4] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

Here is another example application.

In this simple version of 'AI' (it's just fitting a linear function to some data) we can try to see the effect each variable makes. What explains the contribution more?

R^2 is a measure of explained variability in a given model.

The Shapley value can be used here to see how best to attribute the total R^2.

daffidwilde.github.io/matching/

coopgt.readthedocs.io

knightva@cardiff.ac.uk

daffidwilde.github.io/matching/
coopgt.readthedocs.io
knightva@cardiff.ac.uk

Here are some examples of good pieces of software with good documentation for all of the above (except for the routing games.)