Negotiation

José M Vidal

Department of Computer Science and Engineering University of South Carolina.

March 3, 2010

Abstract

We describe automated negotiation as it applies to multiagent systems. Chapter 6.

Why Negotiate?

- Coordinate selfish interests.
- Aggregate distributed conflicting knowledge.
- Solve characteristic form games and more complex versions.

Why Negotiate?

- Coordinate selfish interests.
- Aggregate distributed conflicting knowledge.
- Solve characteristic form games and more complex versions.
- For example: NASA missions, capitol hill?

Bargaining Problem

- $u_i : \Delta \rightarrow \Re$ where Δ is the set of deals.
- δ^- is the no-deal deal.
- Assume that for all agents $u_i(\delta^-) = 0$

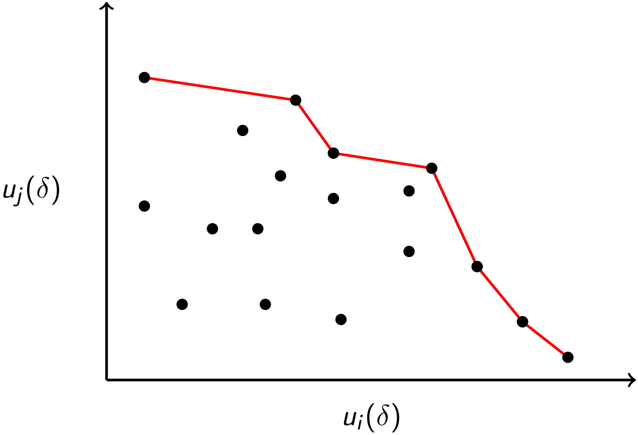
Pareto

Definition (Pareto optimal)

A deal δ is Pareto optimal if there is no other deal such that everyone prefers it over δ. That is, there is no δ' such that

$$\forall_i u_i(\delta') > u_i(\delta).$$

Pareto Frontier



What do we want?

- We will want a Pareto deal, but which one?

What do we want?

- We will want a Pareto deal, but which one?
- Idea: Come up with some requirements first then see if a solution that meets those requirements exists.

Independence from Units

Definition (Independence of utility units)

A negotiation protocol is independent of utility units if when given U it chooses δ and when given $U' = \{(\beta_1 u_1, \ldots, \beta_I u_I) : u \in U\}$ it chooses δ' where

$$\forall_i \; u_i(\delta') = \beta_i u_i(\delta).$$

Symmetry

Definition (Symmetry)

A negotiation protocol is symmetric if the solution remains the same as long as the set of utility functions U is the same, regardless of which agent has which utility.

Individual Rationality

Definition (Individual rationality)

A deal δ is individually rational if

$$\forall_i \; u_i(\delta) \geq u_i(\delta^-).$$

Which means that $u_i(\delta) \geq 0$ since we will be assuming that $u_i(\delta^-) = 0$. A deal is individually rational if all the agents prefer it over not reaching an agreement.

Independence from Irrelevant Alternatives

Definition (Independece of irrelavant alternatives)

A negotiation protocol is independent of irrelevant alternatives if it is true that when given Δ it chooses δ and when given $\Delta' \subset \Delta$ where $\delta \in \Delta'$ it again chooses δ, assuming U stays constant.

That is, a protocol is independent of irrelevant alternative is the deal it chooses does not change after we remove a deal that lost. Only removal of the winning deal changes the deal the protocol chooses.

Egalitarian

$$\delta = \arg\max_{\delta' \in E} \sum_i u_i(\delta')$$

where E is the set of all deals where all agents receive the same utility, namely

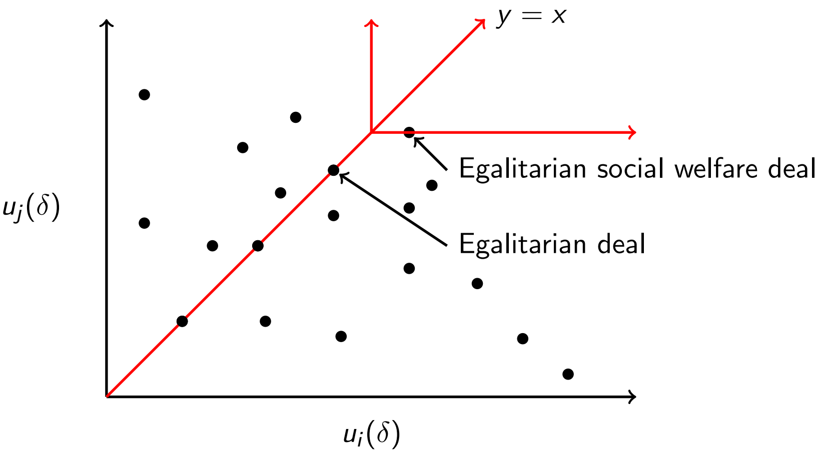
$$E = \{\delta \mid \forall_{i,j} u_i(\delta) = u_j(\delta)\}.$$

Egalitarian Social Welfare

Find the closest approximation:

$$\delta = \arg\max_{\delta} \min_{i} u_i(\delta)$$

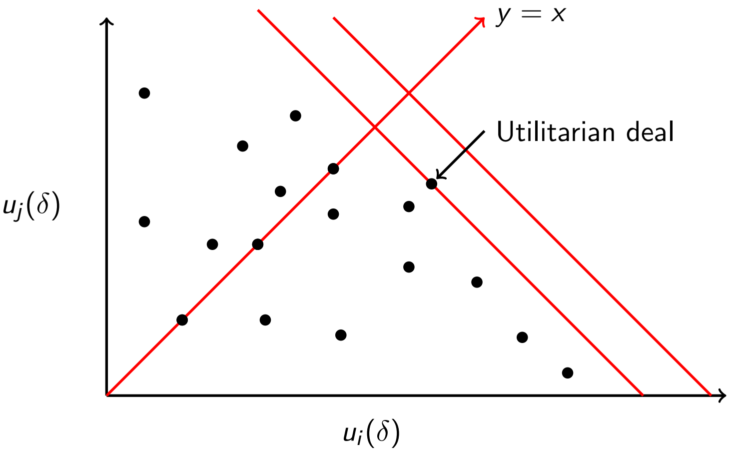
Egalitarian



Utilitarian Solution

$$\delta = \arg\max \sum_i u_i(\delta).$$

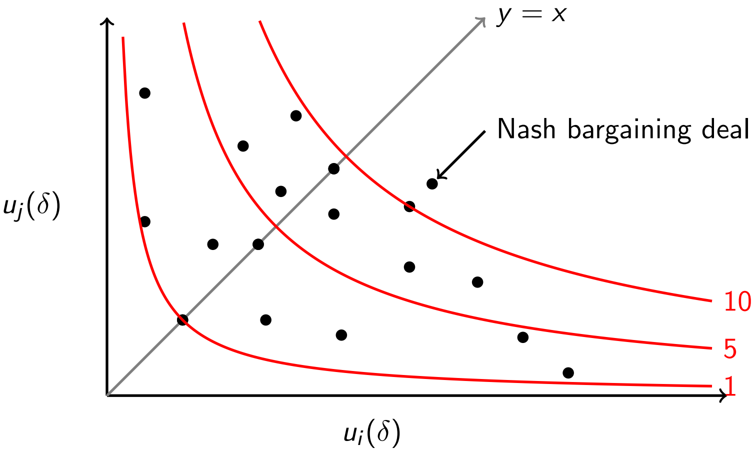
Utilitarian Solution



Nash Bargaining Solution

$$\delta = \arg\max_{\delta'} \prod u_i(\delta').$$

Nash Bargaining Solution



Nice Nash

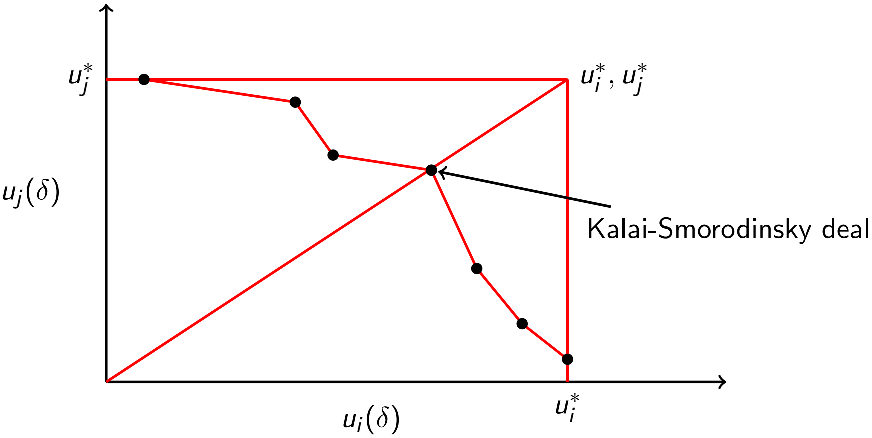
Nash bargaining solution is the only one that satisfies:

1. Pareto efficient
2. Independent of utility units
3. Independent of irrelevant alternatives
4. Symmetric

Kalai-Smorodinsky

- Let u_i^* be the maximum utility that i could get from the set of all deals in the Pareto frontier.
- Then, find the deal that lies in the line between the point δ^- and the point (u_i^*, u_j^*)

Kalai-Smorodinsky



Strategic Solutions

Idea:

1. Formalize the bargaining process
2. Assume rational agents
3. Determine their equilibrium strategies for their bargaining process.

Rubinstein's Alternating Offers

1. Two agents i and j
2. At each time step t one agent proposes a deal δ
3. The other can either accept or reject δ
4. Utilities decrease over time $u_i = \lambda_i^t u_i(\delta)$

Theorem (Alternating Offers Bargaining Strategy)

The Rubinstein's alternating offers game where the agents have complimentary linear utilities ($u_i(\delta) = \delta$ and $u_j(\delta) = 1 - u_i(\delta)$) has a unique subgame perfect equilibrium strategy where

- *agent i proposes a deal*

$$\delta_i^* = \frac{1 - \lambda_j}{1 - \lambda_i \lambda_j}$$

  and accepts the offer δ_j from j only if $u_i(\delta_j) \leq u_i(\delta_j^*)$,

- *agent j proposes a deal*

$$\delta_j^* = \frac{1 - \lambda_i}{1 - \lambda_i \lambda_j}$$

  and accepts the offer δ_i from i only if $u_j(\delta_i) \leq u_j(\delta_i^*)$.

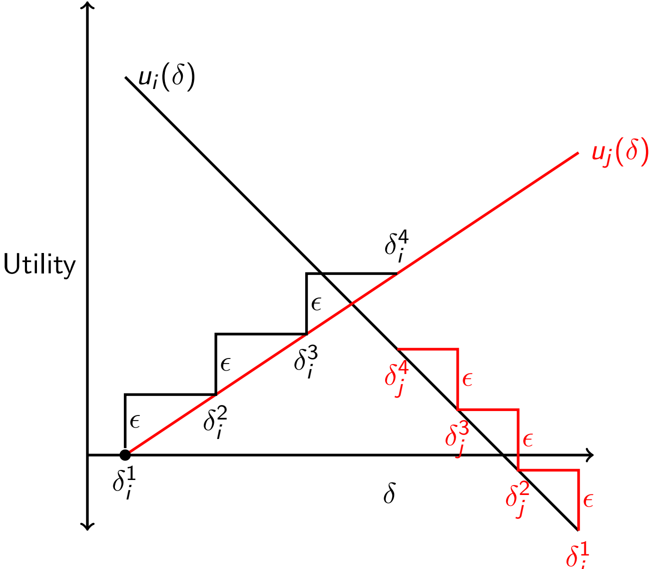
Alternating Offers Strategy

The theorem tells us that the best strategy for these agents is propose a bid on the first time step which will be accepted by the other agent.

MONOTONIC-CONCESSION

1 $\delta_i \leftarrow \arg\max_\delta u_i(\delta)$
2 Propose δ_i
3 Receive δ_j proposal
4 **if** $u_i(\delta_j) \geq u_i(\delta_i)$
5   **then** Accept δ_j
6   **else** $\delta_i \leftarrow \delta'_i$ such that $u_j(\delta'_i) \geq \epsilon + u_j(\delta_i)$ and $u_i(\delta'_i) \geq u_i(\delta^-)$
7 **goto** 2

Monotonic Concession



Monotonic Concession Summary

- Slow
- Agents know others' utility functions
- Tricky last step: both might want other's offer

Zeuthen Strategy

1. Propose my best deal.
2. Let willingness to risk conflict for i be the utility i loses by accepting j's offer divided by the utility i loses by not conceding and causing conflict. That is:

$$risk_i = \frac{u_i(\delta_i) - u_i(\delta_j)}{u_i(\delta_i)}$$

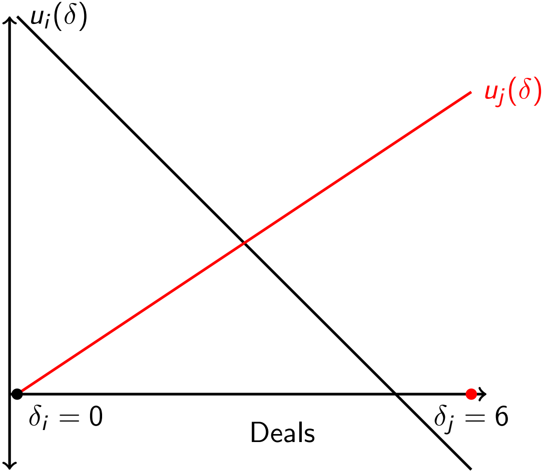
3. If $risk_i < risk_j$ then I must concede *just enough* so that in the next round I do not have to concede again.

Zeuthen Strategy

ZEUTHEN-MONOTONIC-CONCESSION

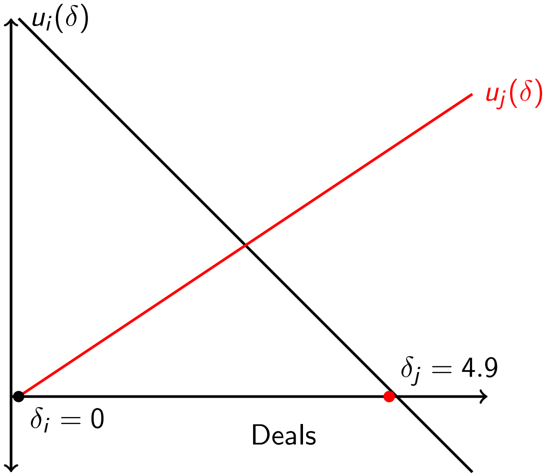
1 $\delta_i \leftarrow \arg\max_\delta u_i(\delta)$
2 Propose δ_i
3 Receive δ_j proposal
4 **if** $u_i(\delta_j) \geq u_i(\delta_i)$
5   **then** Accept δ_j
6 $\text{risk}_i \leftarrow \frac{u_i(\delta_i) - u_i(\delta_j)}{u_i(\delta_i)}$
7 $\text{risk}_j \leftarrow \frac{u_j(\delta_j) - u_j(\delta_i)}{u_j(\delta_j)}$
8 **if** $\text{risk}_i < \text{risk}_j$
9   **then** $\delta_i \leftarrow \delta_i'$ such that $\text{risk}_i(\delta_i') > \text{risk}_j$
10   **goto** 2
11 **goto** 3

Zeuthen Strategy



$u_i(\delta) = 5 - \delta,$
$u_j(\delta) = \frac{2}{3}\delta$
$\delta = \{0 \ldots 6\}$
$\delta_i = 0, \delta_j = 6$
$risk_i = \frac{5-(-1)}{5} = \frac{6}{5},$
$risk_j = \frac{4-0}{4} = 1$

Zeuthen Strategy



$u_i(\delta) = 5 - \delta$,
$u_j(\delta) = \frac{2}{3}\delta$
$\delta = \{0 \ldots 6\}$
$\delta_i = 0, \delta_j = 6$
$risk_i = \frac{5-(-1)}{5} = \frac{6}{5}$,
$risk_j = \frac{4-0}{4} = 1$ j must concede, more than 1.
$\delta_j < 5$

Zeuthen Characteristics

- It is not guaranteed to maximize social welfare.
- It is guaranteed to terminate, and any agreement it reaches will be individually rational and Pareto optimal.
- It is also in Nash equilibrium–if the other guy is using it then you have nothing to gain by not using it. Allows agents to publish their strategy.
- But, sometimes risks are equal.
- Requires agents to know eachother's utility functions.

One Step Protocol

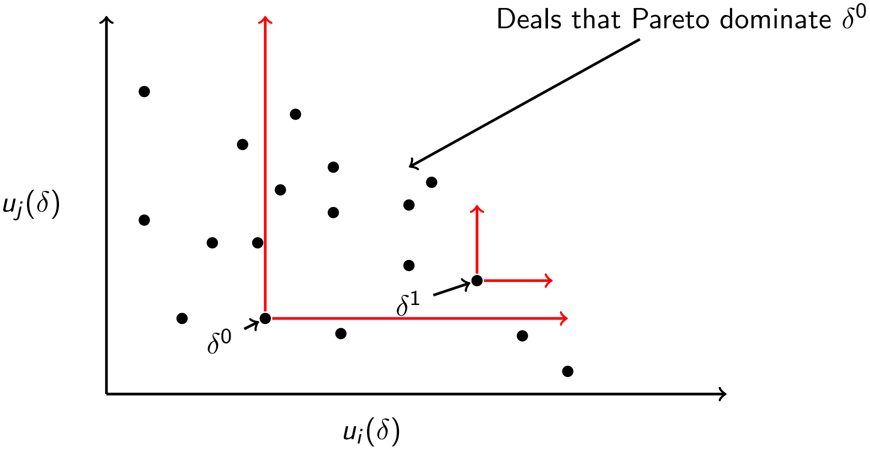
ONE-STEP-NEGOTIATION

$E \leftarrow \{\delta \mid \forall_{\delta'} u_i(\delta)u_j(\delta) \geq u_i(\delta')u_j(\delta')\}$
$\delta_i \leftarrow \arg\max_{\delta \in E} u_i(\delta)$
Propose δ_i
Receive δ_j
if $u_i(\delta_j)u_j(\delta_j) < u_i(\delta_i)u_j(\delta_i)$
    then Report error, j is not following strategy.
Coordinate with j to choose randomly between δ_i and δ_j.

One Step Protocol

- Algorithm is in Nash equilibrium.

Hill Climbing



Ad-hoc Negotiation Strategies

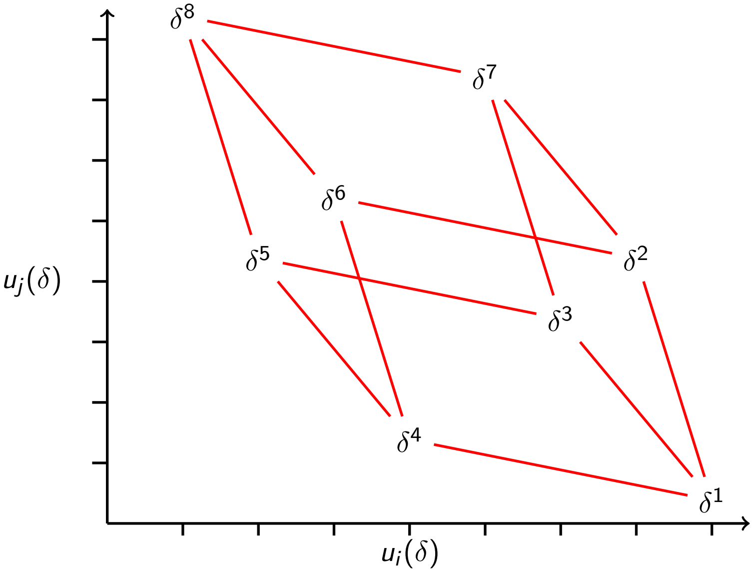
- A linear discounts utility linearly.
- A conceder concedes a lot initially.
- An impatient demands a lot initially.

Task Allocation Problem

- The task allocation problem consists of:
  - T: tasks
  - A: agents
  - $c_i : s \longrightarrow \Re$ cost that i incurs in carrying out tasks $s \subseteq T$.
  - δ represents allocation of tasks to agents.
  - δ^- is initial allocation
- The cost function is monotonic.
- The cost of doing nothing is 0.

Task Allocation Problem

δ	$s_i(\delta)$	$s_j(\delta)$	$c_i(\delta)$	$c_j(\delta)$	$u_i(\delta)$	$u_j(\delta)$
δ^1	$\emptyset$	$\{t_1, t_2, t_3\}$	0	8	8	0
δ^2	$\{t_1\}$	$\{t_2, t_3\}$	1	4	7	4
δ^3	$\{t_2\}$	$\{t_1, t_3\}$	2	5	6	3
δ^4	$\{t_3\}$	$\{t_1, t_2\}$	4	7	4	1
δ^5	$\{t_2, t_3\}$	$\{t_1\}$	6	4	2	4
δ^6	$\{t_1, t_3\}$	$\{t_2\}$	5	3	3	5
δ^7	$\{t_1, t_2\}$	$\{t_3\}$	3	1	5	7
δ^8	$\{t_1, t_2, t_3\}$	$\emptyset$	7	0	1	8


δ^8
δ^7
δ^6
δ^5
δ^2
δ^3
δ^4
δ^1
$u_j(\delta)$
$u_i(\delta)$

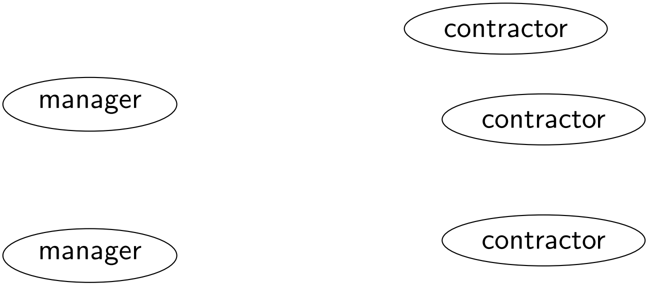
Payments

1. Enable more deals by allowing payments.

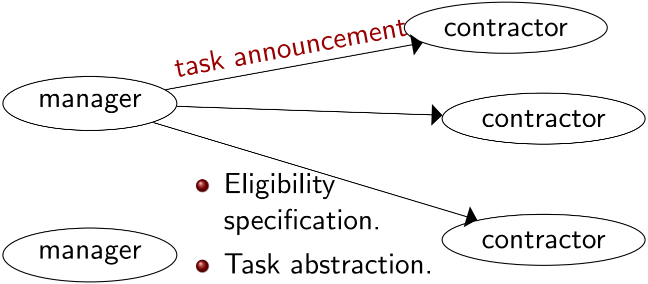
Payments

1. Enable more deals by allowing payments.
2. This was the idea behind the original contract net protocol (Smith and Davis, 1981).

Contract Net Protocol

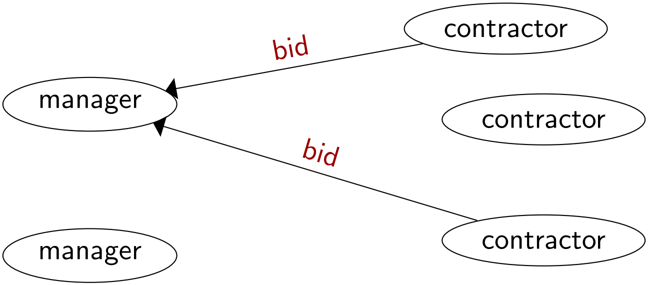


Contract Net Protocol

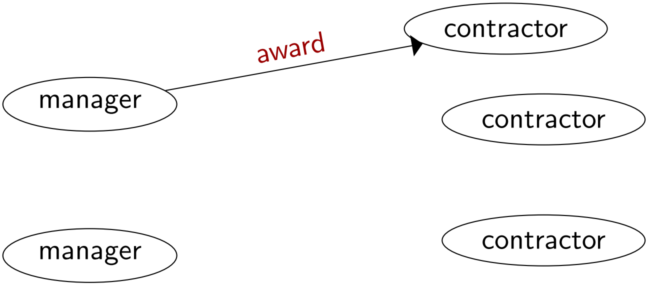


- Eligibility specification.
- Task abstraction.
- Bid specification.
- Expiration time.

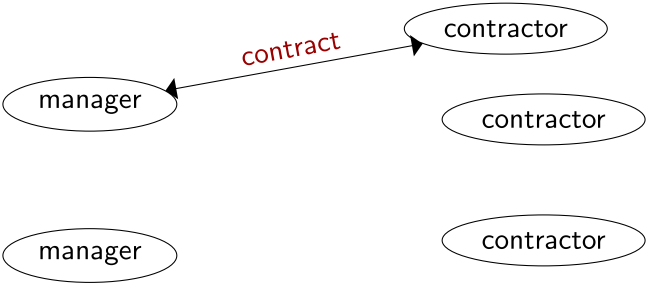
Contract Net Protocol



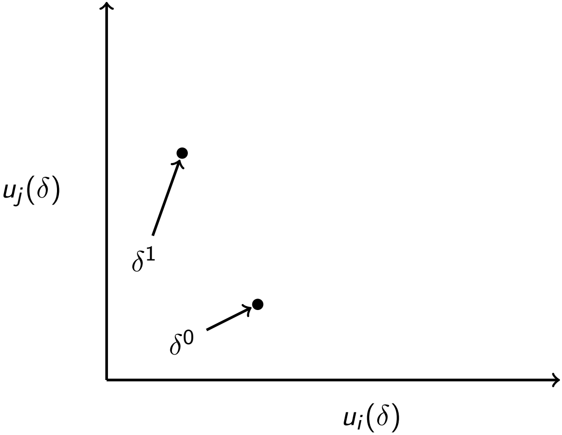
Contract Net Protocol



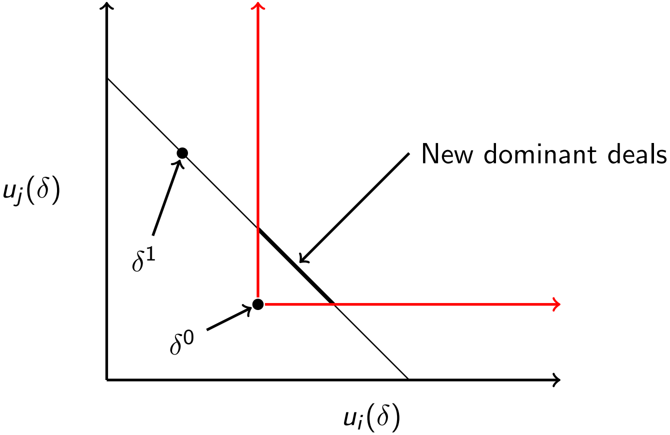
Contract Net Protocol



Payments Create Deals



Payments Create Deals



Additive Cost Functions

More formally,

Definition

A function $c(s)$ is an additive cost function if for all $s \subseteq T$ it is true that

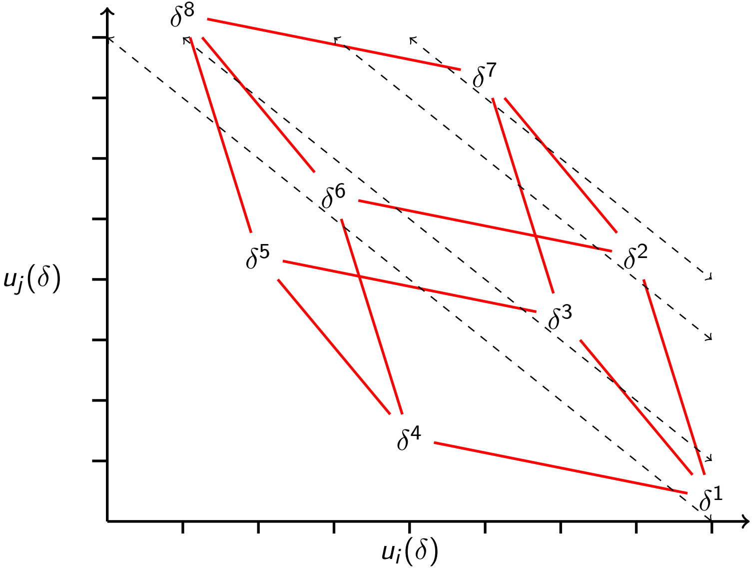
$$c(s) = \sum_{t \in s} c(t).$$

They are easier to analyze.

Additive + Payments

Theorem

In a task allocation problem with an additive cost function where we only allow exchange of one task at a time, any protocol that allows payments and always moves to dominant deals will eventually converge to the utilitarian solution .


δ^8
δ^7
δ^6
δ^5
δ^2
δ^3
δ^4
δ^1
$u_j(\delta)$
$u_i(\delta)$

Arbitrary Cost Functions

- In general, not much we can say.

Arbitrary Cost Functions

- In general, not much we can say.
- If any deal can be reached from any other deal (fully connected) then hill climbing will again reach the utilitarian solution.

Lying About Tasks

Possible Lies

- Not tell others about some tasks I have.
- Make up tasks and hope I end up having to do them.
- Make up tasks and create them if needed.

Assume known final deal. For example, Nash bargaining solution.

Task Creation Example

δ	$s_i(\delta)$	$s_j(\delta)$	$u_i(\delta)$	$u_j(\delta)$
δ^1	$\emptyset$	$\{t_1\}$	1	3
δ^2	$\{t_1\}$	$\emptyset$	2	1

Task Creation Example

δ	$s_i(\delta)$	$s_j(\delta)$	$u_i(\delta)$	$u_j(\delta)$
δ^1	$\emptyset$	$\{t_1\}$	1	3
δ^2	$\{t_1\}$	$\emptyset$	2	1

Create phony t_2.

Task Creation Example

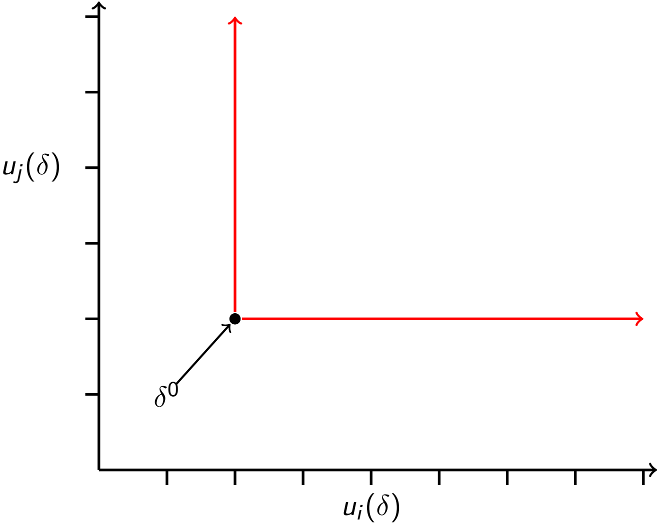
δ	$s_i(\delta)$	$s_j(\delta)$	$u_i(\delta)$	$u_j(\delta)$
δ^1	$\emptyset$	$\{t_1\}$	1	3
δ^2	$\{t_1\}$	$\emptyset$	2	1

Create phony t_2.

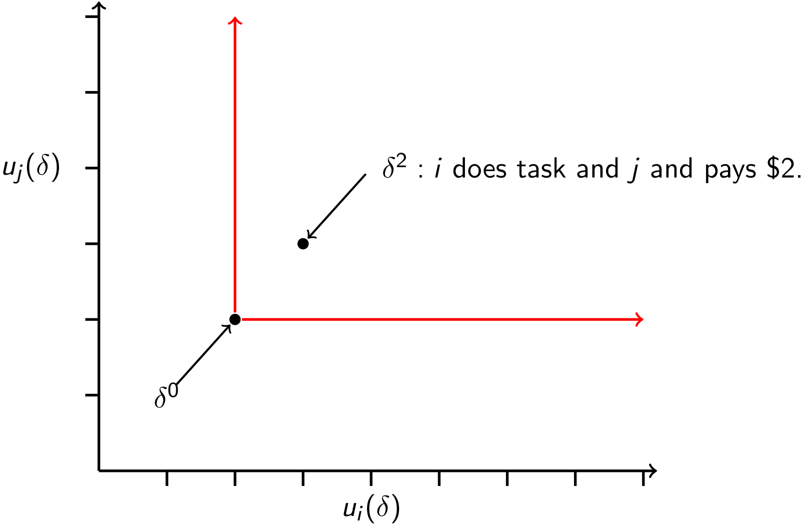
δ	$s_i(\delta)$	$s_j(\delta)$	$u_i(\delta)$	$u_j(\delta)$
δ^1	$\emptyset$	$\{t_1, t_2\}$	1	5
δ^2	$\{t_1\}$	$\{t_2\}$	2	3
δ^3	$\{t_2\}$	$\{t_1\}$	2	3
δ^4	$\{t_1, t_2\}$	$\emptyset$	8	1

Contracts

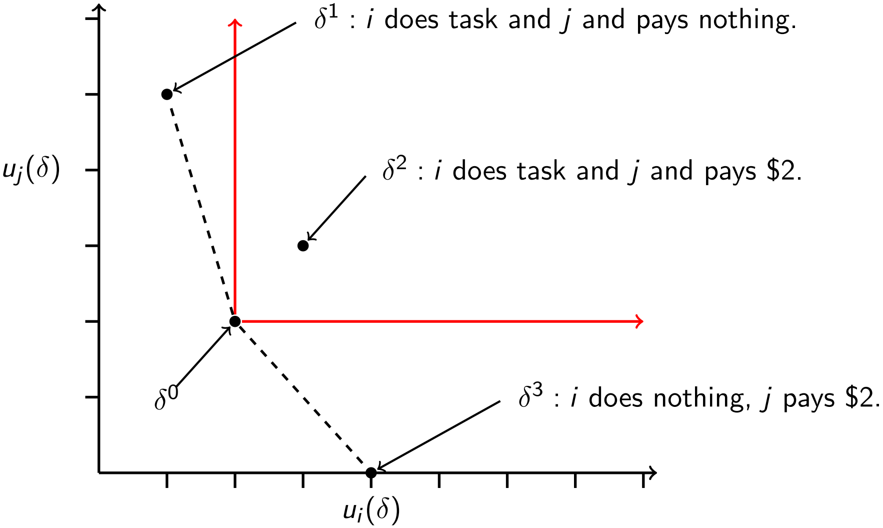
- Agents might want to de-commit on a contract.



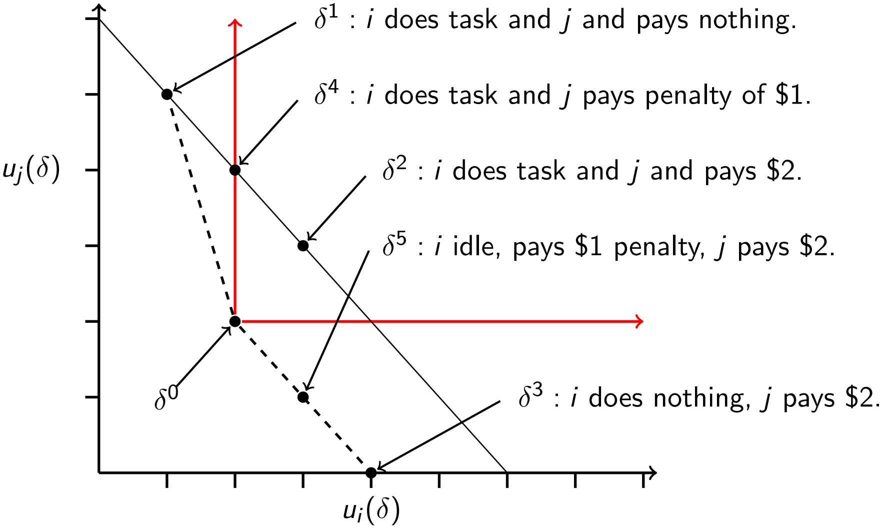
δ^0: j does task and i is idle.



δ^0: j does task and i is idle.



δ^0: j does task and i is idle.



δ^0: j does task and i is idle.

Contract Penalties

- Penalties reduce risks.

Contract Penalties

- Penalties reduce risks.
- But, if we can enforce penalties, why not just enforce original contracts?

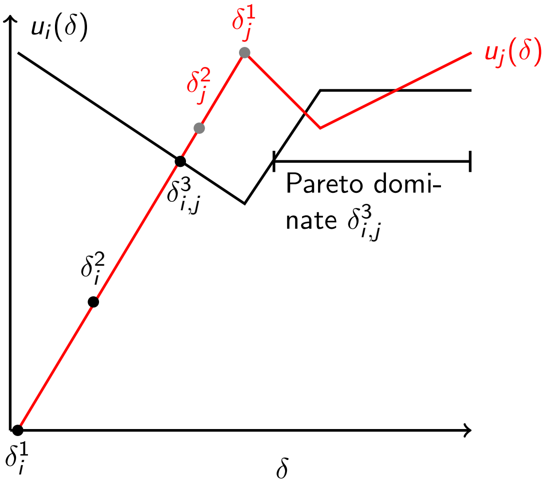
Complex Deals

- A multi-dimensional deal is composed of a set of variables $x_1, x_2, \ldots, x_n$ with domains $D_1, D_2, \ldots D_n$.
- $u_i(\delta)$
- Or, $u_i(\delta) = c_1 u_i^1(x_1) + c_2 u_i^2(x_2) + \cdots + c_n u_i^n(x_n)$

Complex Deals

- A multi-dimensional deal is composed of a set of variables $x_1, x_2, \ldots, x_n$ with domains $D_1, D_2, \ldots D_n$.
- $u_i(\delta)$
- Or, $u_i(\delta) = c_1 u_i^1(x_1) + c_2 u_i^2(x_2) + \cdots + c_n u_i^n(x_n)$
- Yes, this is a constraint optimization problem! But now agents do not own the variables.

Convergence



Negotiation with Mediator

ANNEALING-MEDIATOR

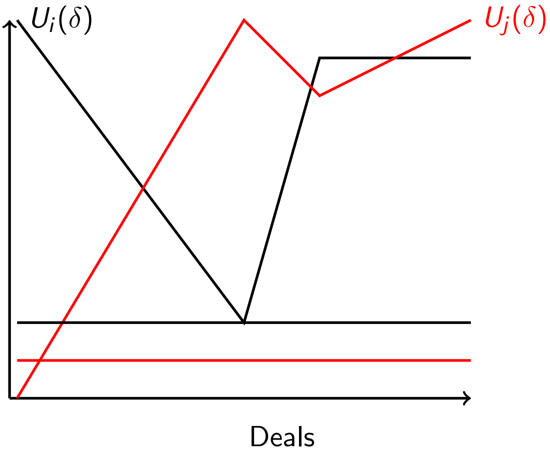
Generate random deal δ.
$\delta_{\text{accepted}} \leftarrow \delta$
Present δ to agents.
if both accept
    then $\delta_{\text{accepted}} \leftarrow \delta$
        $\delta \leftarrow \text{mutate}(\delta)$
        goto 3
if one or more reject
    then $\delta \leftarrow \text{mutate}(\delta_{\text{accepted}})$
goto 3

Hill Climbers and Annealers

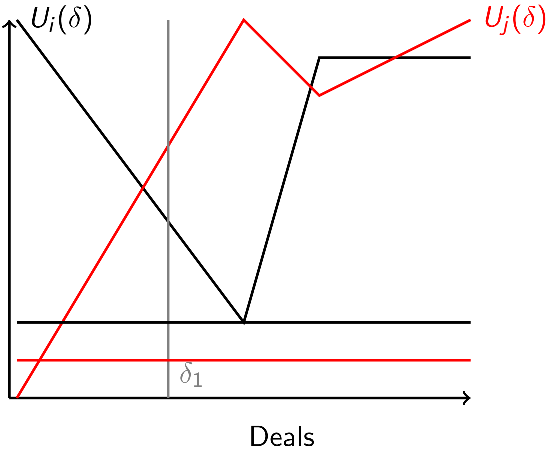
Hill Climber Accepts a deal only if it has utility higher than its reservation price $u_i(\delta^-)$ and higher than that of the last deal it accepted. That is, it monotonically increases it reservation price as it accepts deals with higher utility.

Annealer Use a simulated annealing algorithm. That is, they maintain a temperature T and accept deals worse than the last accepted deal with probability $\max(1, e^{-\frac{\Delta U}{T}})$, where ΔU is the utility change between the contracts.

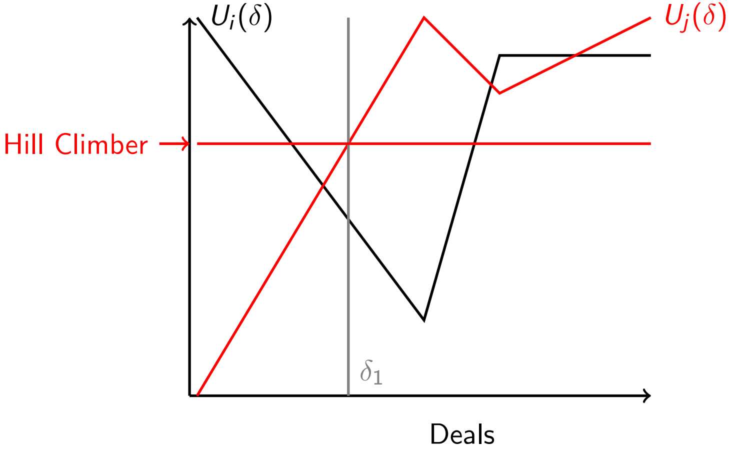
Hill-Climbers and Annealers



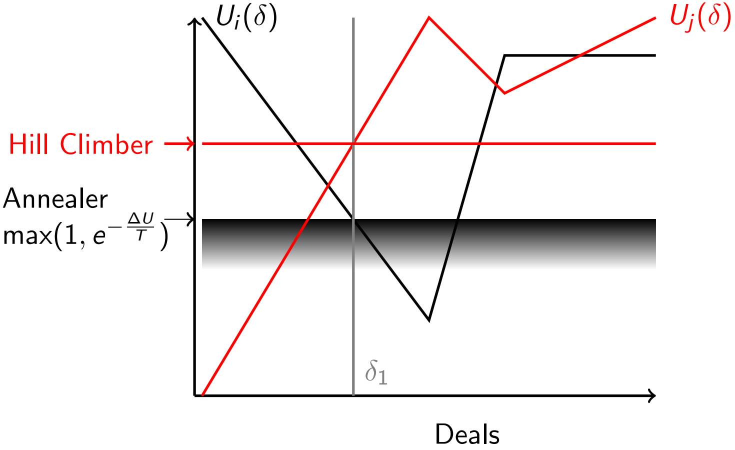
Hill-Climbers and Annealers



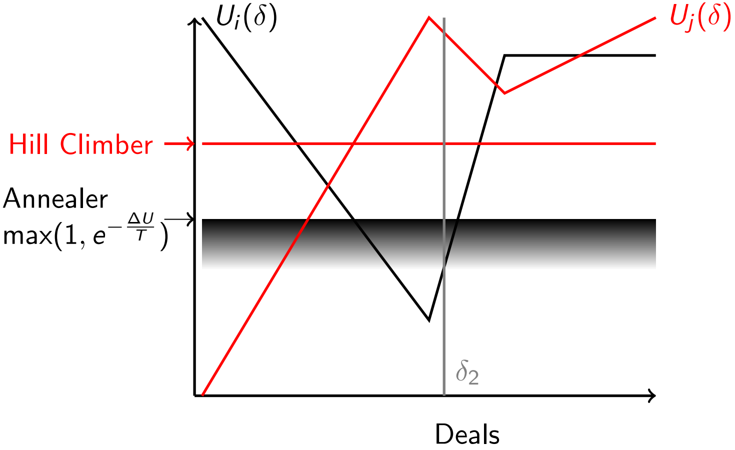
Hill-Climbers and Annealers



Hill-Climbers and Annealers



Hill-Climbers and Annealers



Prisoner's Dilemma, again!

	Hill Climber	Annealer
Hill Climber	.73, .74	.99, .51
Annealer	.51, .99	.84, .84

Adding Tit-for-Tat

	Hill Climber	Annealer	T4T
Hill Climber	400, 400	700, 180	500, 340
Annealer	180, 700	550, 550	550, 550
T4T	340, 500	550, 550	550, 550

Argument-Based Negotiations

- Critique
- Counter-proposal
- Justify
- Persuade
- There are also threats, rewards, and appeals.

Argument-Based Negotiations

- Critique the proposal.

  A: I propose that you provide me with service X under conditions P.

  B: I am happy with the price of X but the delivery date is too late.

  A: I propose that I will provide you with service Y if you provide me with X.

  B: I don't want Y.
- Counter-proposal
- Justify
- Persuade
- There are also threats, rewards, and appeals.

Argument-Based Negotiations

- Critique
- Counter-proposal

  A: I propose that you provide me with service X.
  B: I propose that I provide you with service X if you provide me with service Z.
  A: I propose that I provide you with service Y if you provide me with service X.
  B: I propose that I provide you with service X if you provide me with service Z.
- Justify
- Persuade
- There are also threats, rewards, and appeals.

Argument-Based Negotiations

- Critique
- Counter-proposal
- Justify his reason for adopting a particular negotiation stance.

  A: I don't have the software for delivering service X.
- Persuade
- There are also threats, rewards, and appeals.

Argument-Based Negotiations

- Critique
- Counter-proposal
- Justify
- Persuade the other agent to change its negotiation stance.

  A: Service X is much better than you think, look at this report.
- There are also threats, rewards, and appeals.

Argument-Based Negotiations

- Critique
- Counter-proposal
- Justify
- Persuade
- There are also threats, rewards, and appeals.

These techniques help

- build model of opponent's utility function,
- eliminate whole sets of deals,
- change the other agent's utility function,
- change my utility function.

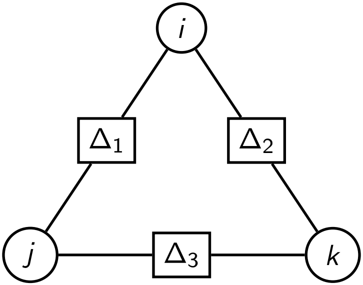
Negotiation Networks

Definition

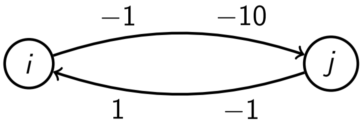
A negotiation network problem involves a set of agents A and set of sets of deals. Each set of deals Δ_i involves only a subset of agents $\Delta_i^a \subseteq A$ and always includes the no-deal deal δ^-. A solution $\vec{\delta}$ to the problem is a set of deals, one from each Δ_i set, such that all the deals that each agent is involved in are compatible with each other. We thus define

$$c_i(\delta, \delta') = \begin{cases} 1 & \text{if } \delta \text{ and } \delta' \text{ are compatible} \\ 0 & \text{otherwise} \end{cases}$$

Negotiation Network

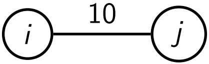


Network Exchange Theory

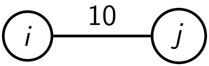


The coercion network.

Equi-Resistance



Equi-Resistance



i's resistance to payment p is given by

$$r_i = \frac{p_i^{max} - p_i}{p_i - p_i^{con}}$$

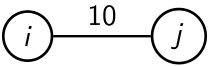
where

$$p_i^{max} = \text{Maximum } i \text{ could get, } 10$$

and

$$p_i^{con} = \text{Conflict deal, } 0$$

Equi-Resistance

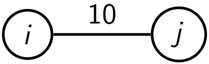


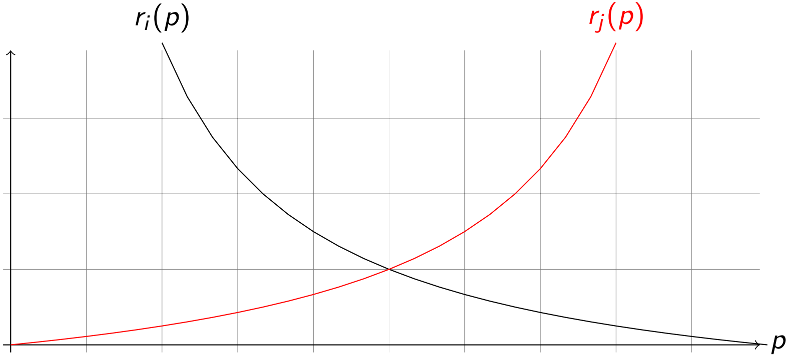
NET tells us that exchange happens at equi-resistance:

$$r_i = \frac{p_i^{max} - p_i}{p_i - p_i^{con}} = \frac{p_j^{max} - p_j}{p_j - p_j^{con}} = r_j.$$

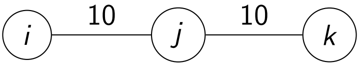
We can represent this graphically by simply replacing p_j with $10 - p_i$ in j's resistance equation r_j and plotting the two curves r_i and r_j. The point at which the curves cross is the point of exchange.

Equi-Resistance

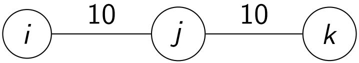


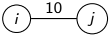
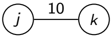


Iterated Equi-Resistance

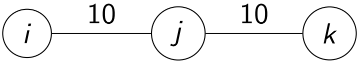


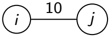
Iterated Equi-Resistance

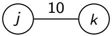


1. Apply Equi-resistance to  .
2. Apply Equi-resistance to  .
3. Repeat until quiescence.

Iterated Equi-Resistance

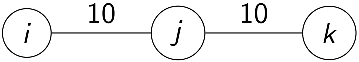


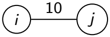
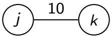
1. Apply Equi-resistance to 

. Gives us $p_j = 5$.
2. Apply Equi-resistance to 

.
3. Repeat until quiescence.

Iterated Equi-Resistance



1. Apply Equi-resistance to  . Gives us $p_j = 5$.
2. Apply Equi-resistance to  . Let $p_j^{con} = 5$ and apply equi-resistance again.
3. Repeat until quiescence.

NET Limitations

- Only tested on small networks.
- Multiple equilibriums.
- Might never settle down.
- Still, viable descriptive solution.