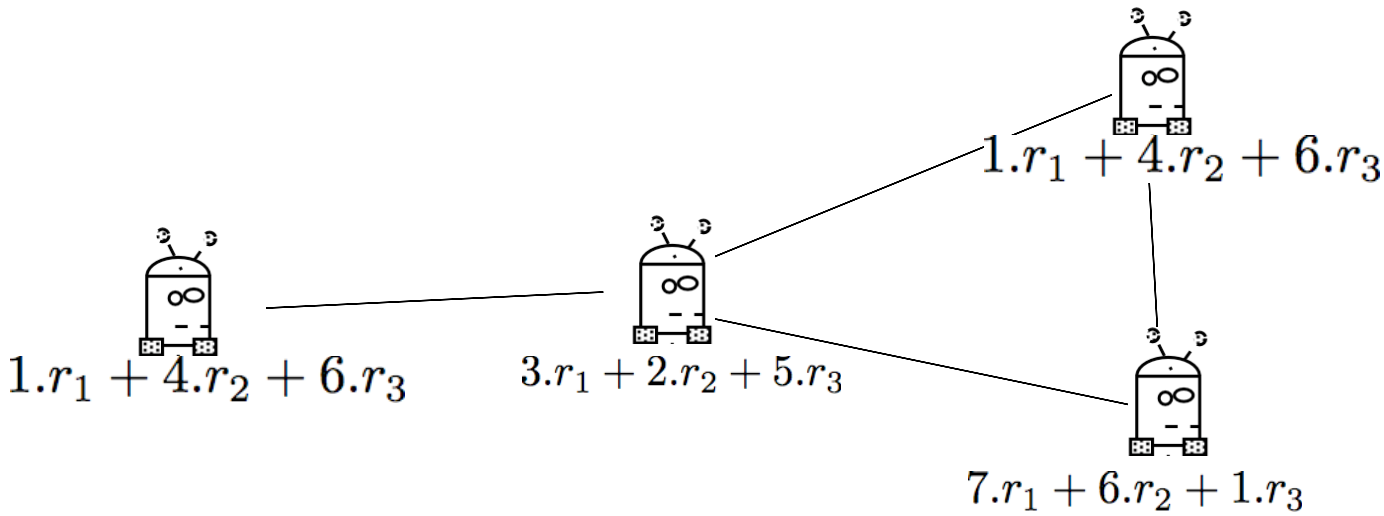
MARA on Graphs

Yann Chevaleyre,
joint work with Nicolas Maudet
& Ulle Endriss

3rd MARA-GetTogether

Setting

- Similar to Nicolas' tutorial
  - Non-divisible, non shareable resources
  - Agents have utility function, with no externalities
  - The question is how to allocate efficiently (w.r.t the *utilitarian social welfare* $\sum u_i$)
- But: agents can only negotiate with their neighbours.

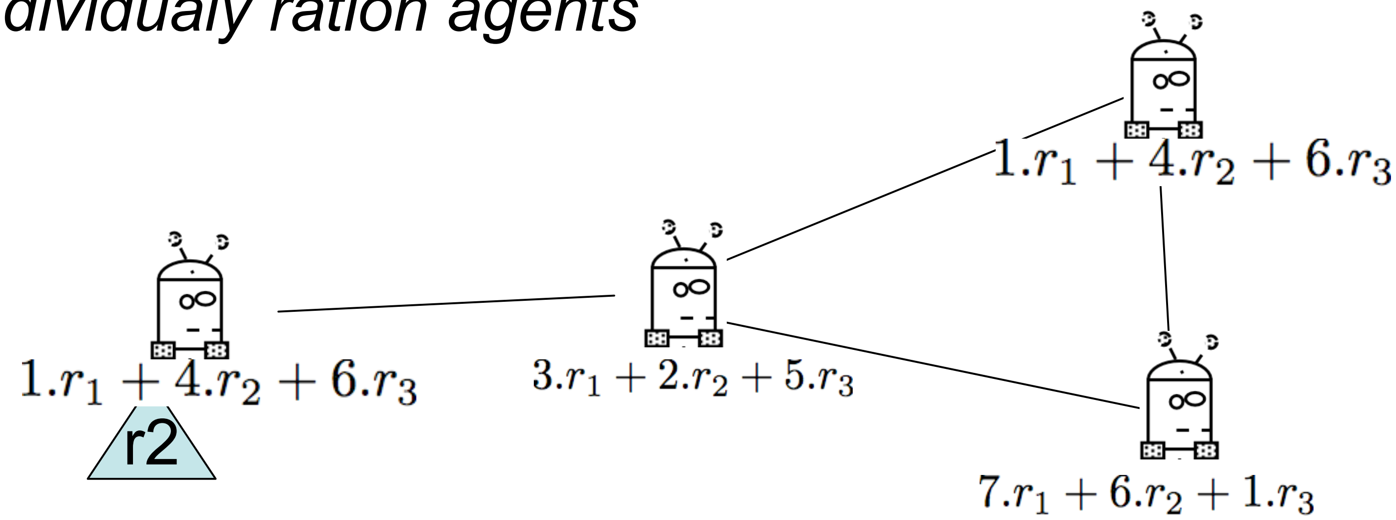


Outline of this talks

1. **Miopic Agents**
   – Will optimal allocation be reached ? How far from optimal ? What is the dynamic of resources on the graph ?
2. Non-Miopic/Learning Agents
   – Although agents know nothing about other non-neighbour agents, is it possible to do better than miopic ?

Graphs induce Sub-optimal outcomes

- Even in simple settings (additive utilities), optimal allocation is no more guaranteed.
- If the graph was complete, optimal allocation would be reached   (« Bottleneck effect »)
- To overcome this, we would need *non-myopic/non individualy ration agents*

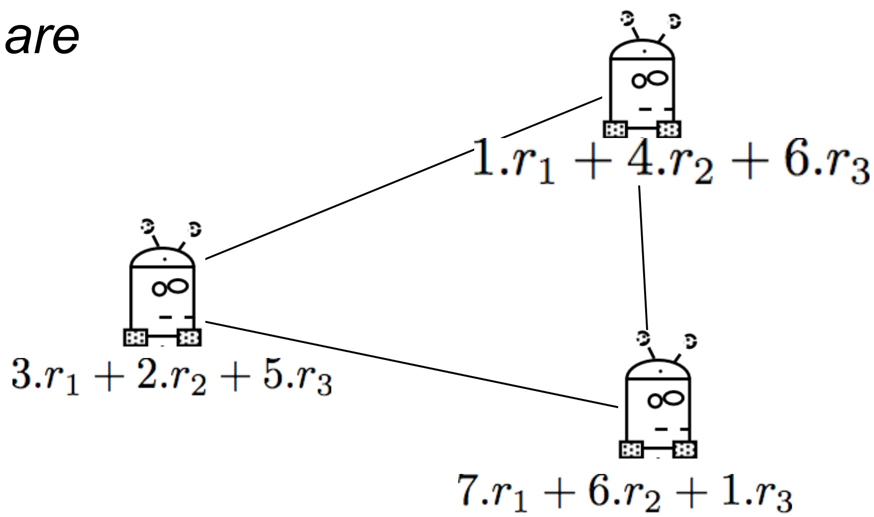


Our goal

- Find a way to caracterize the bottleneck effect, with parameters of the graph
- Study the number of « moves » of a resource in a graph, and relate to the sw
- Find a « realistic » set of assumptions under which this can be computed.

Setting/Assumptions

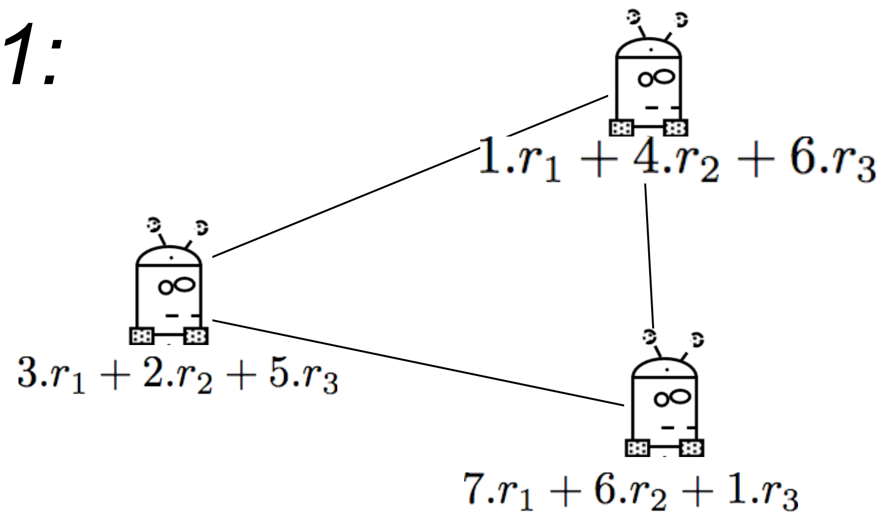
- Additive utilities
  simpler setting to analyse, but: we expect our results to hold for arbitrary utilities
- Utilities drawn from an unknown distribution $\mathcal{D}$
  Unrealistic: equivalently, agents are placed randomly on the graph, and cannot change their placement the way they want.



Trajectory of a resource

- Which path can it take ?

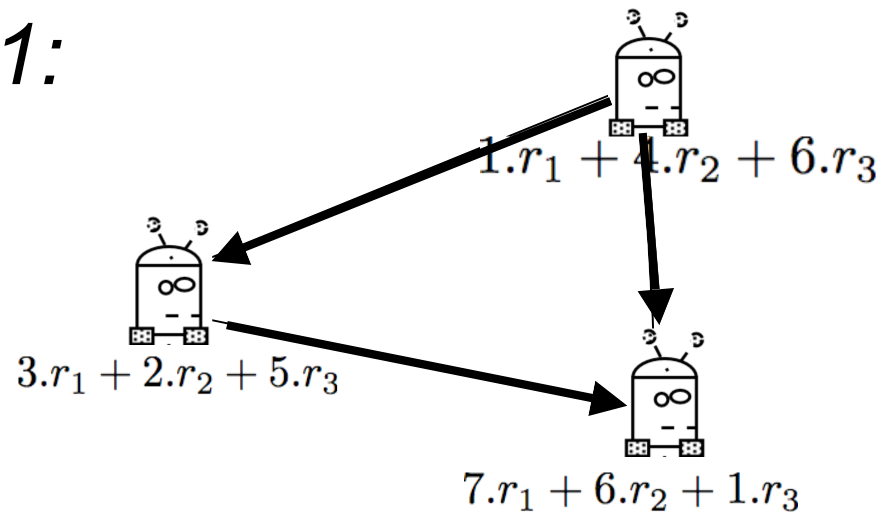
For e.g., r1:



Trajectory of a resource

- Which path can it take?

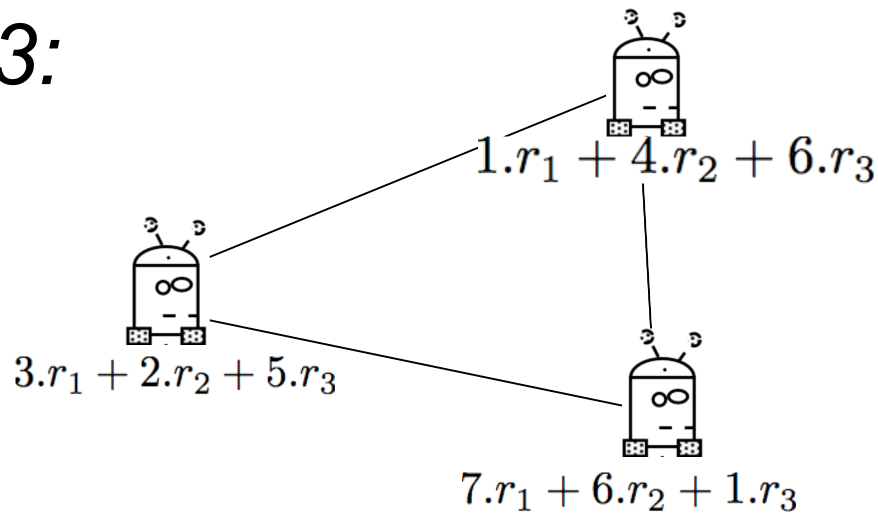
For e.g., r1:



Trajectory of a resource

- Which path can it take?

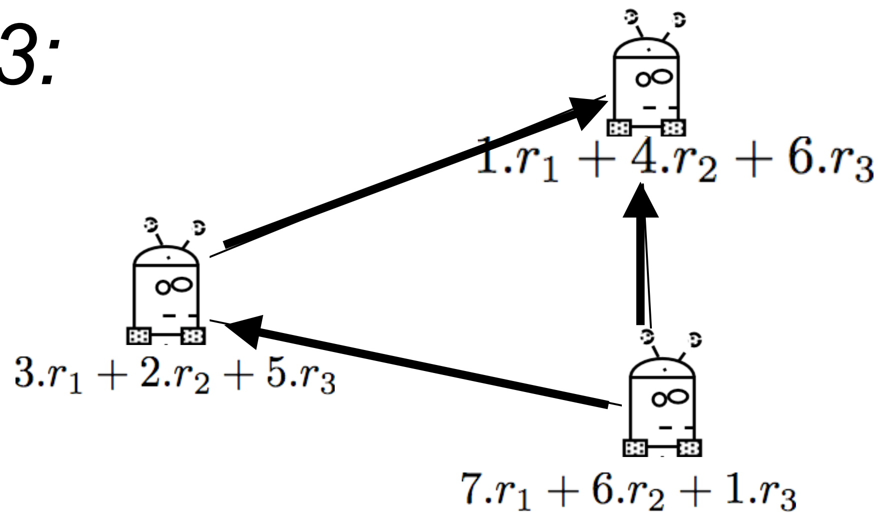
For e.g., r3:



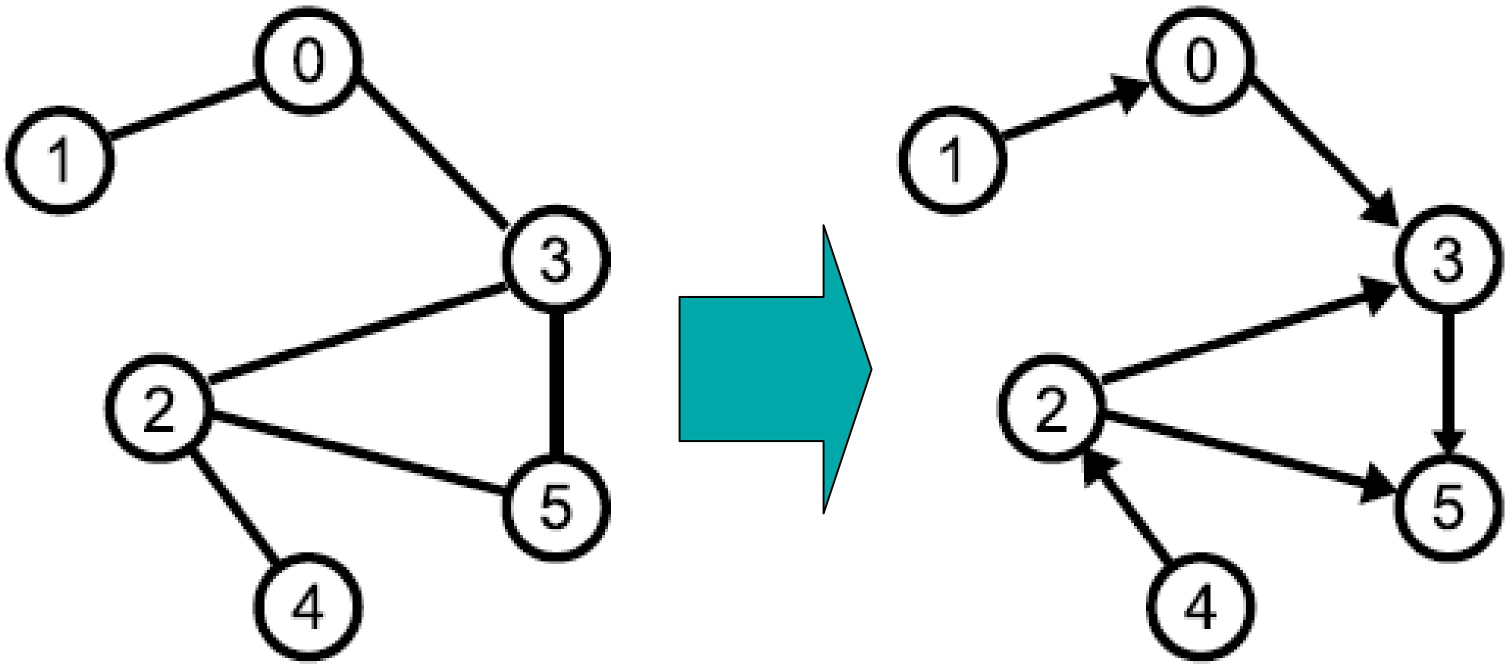
Trajectory of a resource

- Which path can it take?

For e.g., r3:



Utilities -> digraph

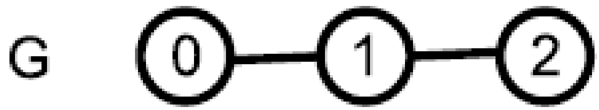


Trajectory of a resource

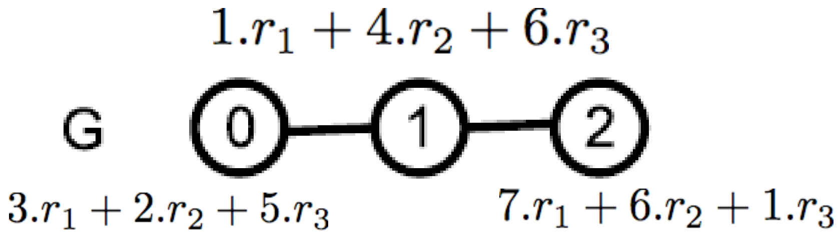
- When utilities are modular, trajectories are **independant**
- With the initial allocation, the directed graph contains **all the information** to compute the trajectory of *r*.
- **Goal** : estimate the number of steps accross the graph made by each resource.

Expected trajectory length on chains (1/4)

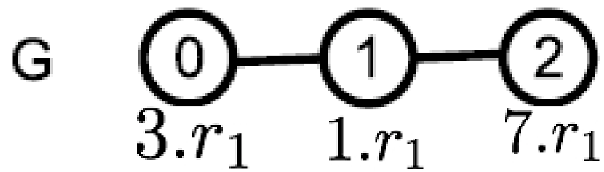
- Consider a graph with three agents 1,2,3



- Suppose their utilities are drawn randomly

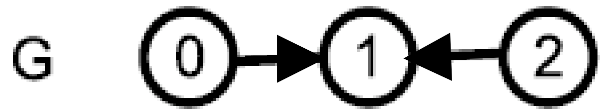


- Focus on a single resource



- This induces an order among agents and a digraph

$$1 \prec 0 \prec 2$$

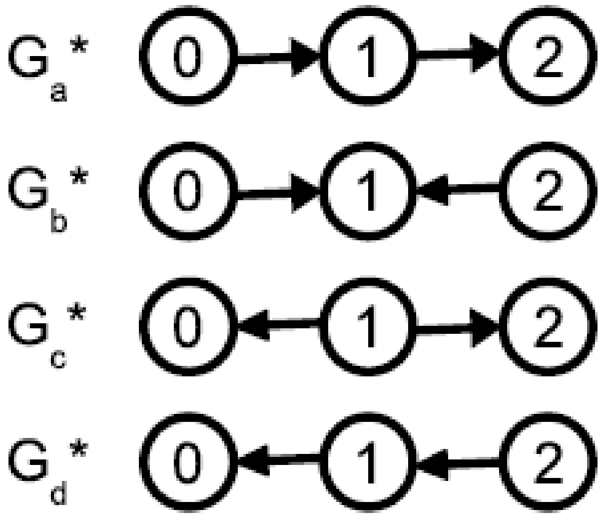


Expected trajectory length on chains (2/4)

- Utilities are drawn randomly from $\mathcal{D}$
- This implies that *all orders are equiprobable*

$$Pr[1 \prec 2 \prec 3] = Pr[1 \prec 3 \prec 2] = Pr[2 \prec 1 \prec 3] = \ldots = \frac{1}{3!}$$

- **but not all digraphs !!!**

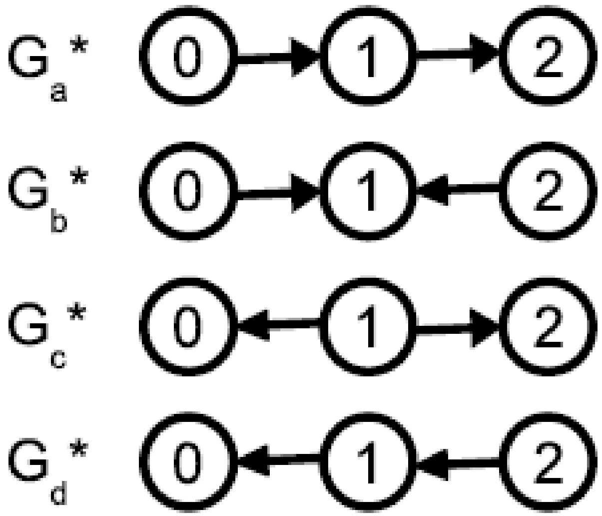


Expected trajectory length on chains (2/4)

- Utilities are drawn randomly from $\mathcal{D}$
- This implies that *all orders are equiprobable*

$$Pr[1 \prec 2 \prec 3] = Pr[1 \prec 3 \prec 2] = Pr[2 \prec 1 \prec 3] = \ldots = \frac{1}{3!}$$

- **but not all digraphs !!!**



$0 \prec 1 \prec 2$

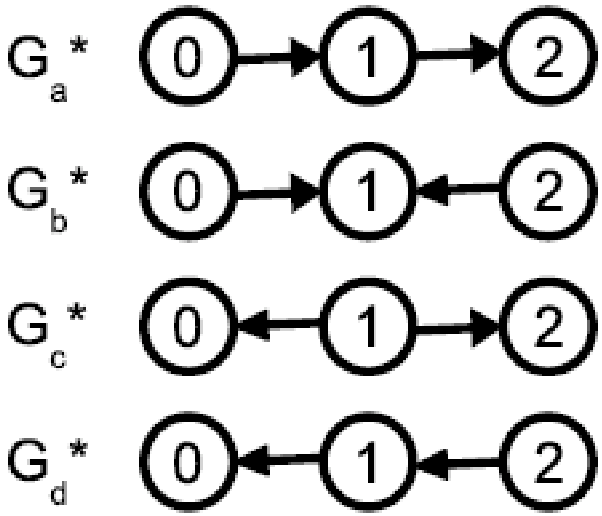
$0 \prec 2 \prec 1$
$2 \prec 0 \prec 1$

Expected trajectory length on chains (3/4)

- Utilities are drawn randomly from $\mathcal{D}$
- This implies that *all orders are equiprobable*

$$Pr[1 \prec 2 \prec 3] = Pr[1 \prec 3 \prec 2] = Pr[2 \prec 1 \prec 3] = \ldots = \frac{1}{3!}$$

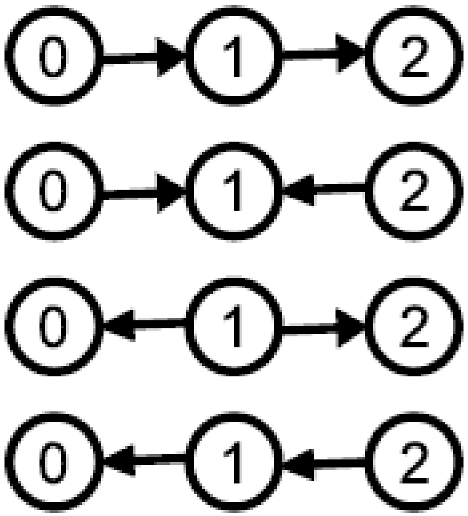
- **but not all digraphs !!!**



G_a^*	$0 \prec 1 \prec 2$	Pr=1/6
G_b^*	$0 \prec 2 \prec 1$ $2 \prec 0 \prec 1$	Pr=2/6
G_c^*		Pr=2/6
G_d^*		Pr=1/6

Expected trajectory length on chains (4/4)

- Suppose resource r_1 is located on agent *0*.
- Compute trajectory of each digraph
- Compute length of expected trajectory



G_a^*	$0 \prec 1 \prec 2$	Pr=1/6 , len=2
G_b^*	$0 \prec 2 \prec 1$ $2 \prec 0 \prec 1$	Pr=2/6 , len=1
G_c^*	$1 \prec 2 \prec 0$ $1 \prec 0 \prec 2$	Pr=2/6 , len=0
G_d^*	$2 \prec 1 \prec 0$	Pr=1/6 , len=0

E[len]=2/3

Average Length of a walk in any graph of bounded degree δ

$$P[walk\, len_0 \;=\; k] \;\leq \frac{\delta}{\delta+1} \times \prod_{t=1}^{k-1} \left(\frac{\delta-1}{\delta+t}\right)$$

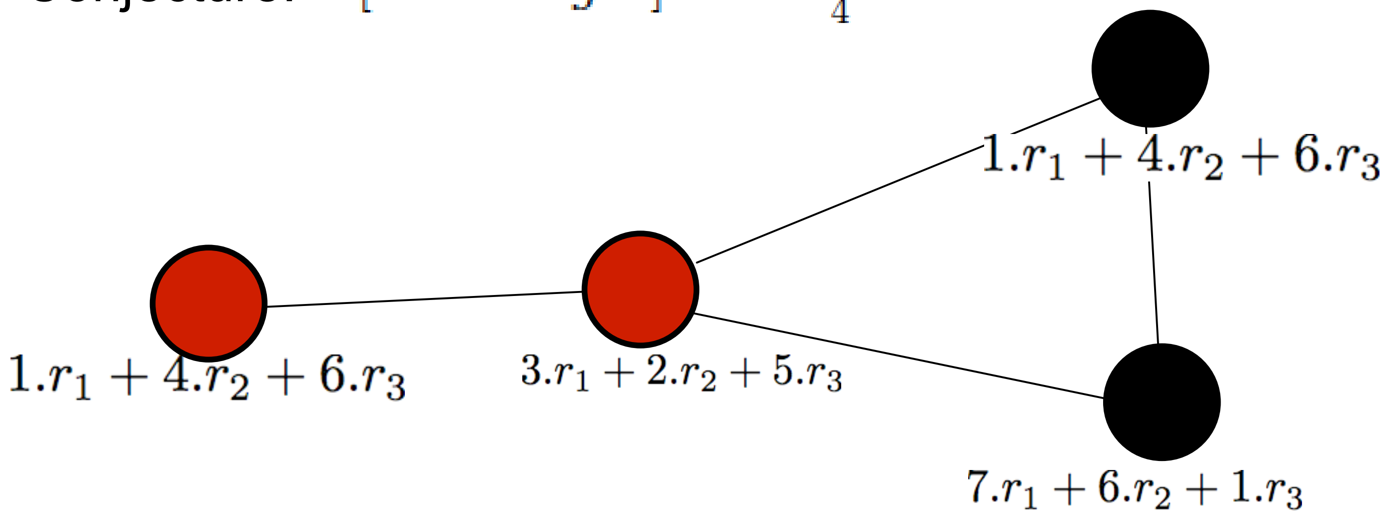
$$E[walk\, length] \;\leq\; \frac{\delta^2 - 1}{4}$$

Corrolary: If coefficients of utilities are distributed uniformly on [0,α] we get:

$$E[sw_{final}] \leq m\alpha \times (1 - 2^{-\frac{\delta^2+3}{4}})$$

Removing assumptions

- Addivity of utilities
  - Conjecture : trajectory length is approximately the same
- Independance of distribution of agents
  - There are 2 categories of individual (e.g. red & white) caracterized by two different distributions. Each agent can choose to be one of those
  - Conjecture: $E[walk\,length] \leq 2\,\frac{\delta^2 - 1}{4}$



Conclusion

- Assuming conjectures, result is quite « general »
- Better bounds to be found
  - Bound could be much tighter than $O(d^2)$
  - bounds based on the degree distribution.
- Except for graphs with high degree (small world, complete graphs, expander graphs), resources do not move a lot.
- Many other types of sw can be estimated with this method.

Outline of this talks

1. Miopic Agents
   – Will optimal allocation be reached ? How far from optimal ? What is the dynamic of resources on the graph ?
2. **Non-Miopic/Learning Agents**
   – Although agents know nothing about other non-neighbour agents, is it possible to do better than miopic ?

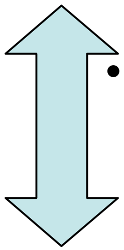
MARA on Graphs : finding opt allocation

- With central authority
  - Global optimization
    - Finding the opt allocation w.r.t. a criterion
- Without central authority
  - Local optimization/learning, depending on the agents knowledge

From optimization to learning

– Assume at each time step, each agent can propose a transaction with one of its neighbors.
– Local optimization/learning, depending on the agents knowledge (privacy issues)

optimization

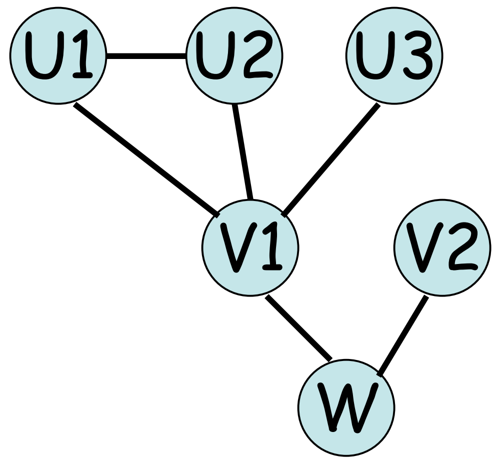


- Agents know everything (graph+utilities+allocation)
  Agents know the graph only
  Agents know nothing except the identity of their neighbor

learning

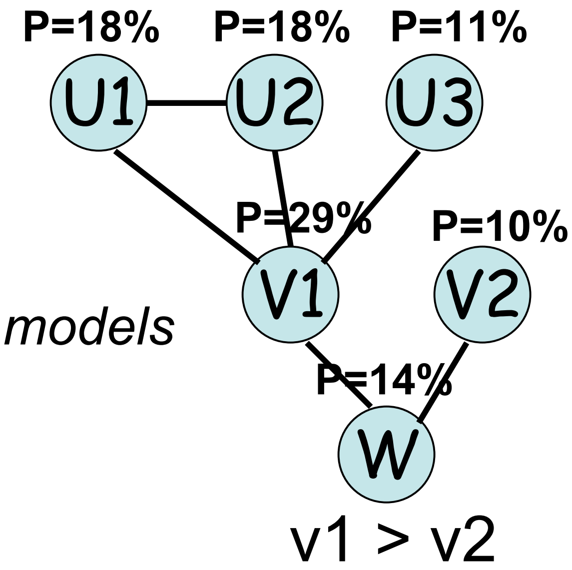
Knowing the graph…what can we do ?

- No knowledge about:
  - Current allocation (except own goods)
  - Utilities
- With which neighbor should agents trade ?
- Assume resources travel freely on the graph, and randomly
- Then, for w, v1 > v2



Knowing the graph…what can we do ?

- **Assumption:** resources travel freely on the graph and randomly, what is the prob that *r* is on v ?



- **Related to**:
  - network flow problems
  - *Stationary distributions in markov models*
  - Spectral graph theory

Reasoning with very partial information: Multiagent Learning

- Mal Learning:
  « given that an agent has no control/knowledge over its opponent, how should it act ? »
- Mainly Economic litterature / game theory
  [Fudenberg,Leving]

Reasoning with very partial information Multiagent Learning - Main aspects

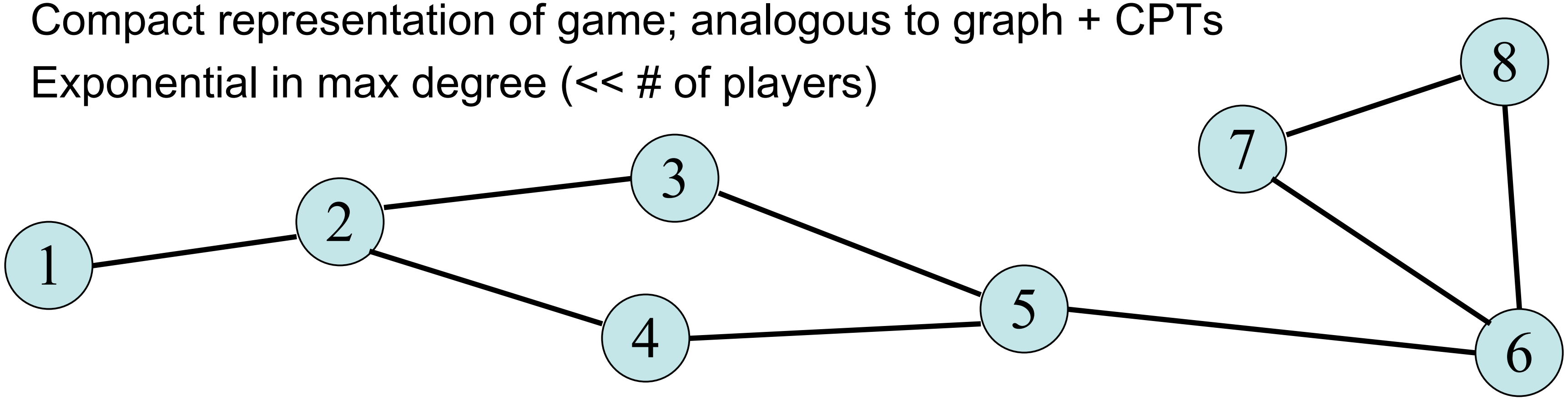
- **Information available to learner**:
  - *The full matrix*
  - *Payoffs of actions taken by others*
  - *Payoffs of our actions only (partial monitoring)+actions of others*
  - *Our payoff* only
- **Define Criteria**
  - *Rationality*. (best response against a stationary opponent)
  - *Convergence*. (nash in self-play)
- **Define possible States/actions**

Our setting in MAL

- **Types of agents**
  - Altruistic, maximizing sw (team game)
  - Selfish (general sum game)
- **From MARA to games**:
  - State = Allocation
  - Actions = selling r to a for price x, buying r to b
    or just: *trade with x*
- **Modeling rewards**:
  - Independant learners (no interactions)
  - Graphical games (interaction between neighbors only)
  - Repeated game (no states)
  - Stochastic games (each state has its matrix game)

Graphical Games

- Undirected graph *G* capturing local (strategic) interactions
- Each player represented by a vertex
- *N_i(G) :* neighbors of i in *G* (includes i)
- Assume: Payoffs expressible as *M_i(**a'**),* where ***a'*** over only *N_i(G)*
- Graphical game: *(G,{M'_i})*
- Compact representation of game; analogous to graph + CPTs
- Exponential in max degree (<< # of players)



- Computation of correlated equilibria : sparse LP [kearns]
- Learning in a cooperative setting [guestring'02]

over-simplified settings

- **Independant learners (no interactions)**
  - Define States. e.g. state=owned resources.
    Actions = « trade with a », « trade with b »..
  - WPL [AAMAS'07]
  - Wolf-PHC [IJCAI'01]
  - Coin [NIPS'99]
- Suppose single negotiation process
  => not enough time to learn state space. What can be done ?

  Independant learners without states

  - Multi-armed bandit algorithms (no state)
    - Can converge to nash in zero-sum game
    - Minimizes regret in general sum game
    - E.g. ε-greedy algorithm

Conclusion

- Learn quickly with bandits
- Learn slowly but accurately with stochastic (graphical) games
- In fully cooperative setting (non-selfish), many efficient learning algorithms