

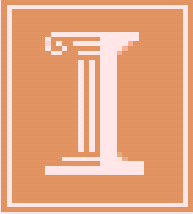
# On Scaling Multi-Agent Task Reallocation using a Market-Based Approach

**Rajesh Karmani, Timo Latvala,\* and Gul Agha**

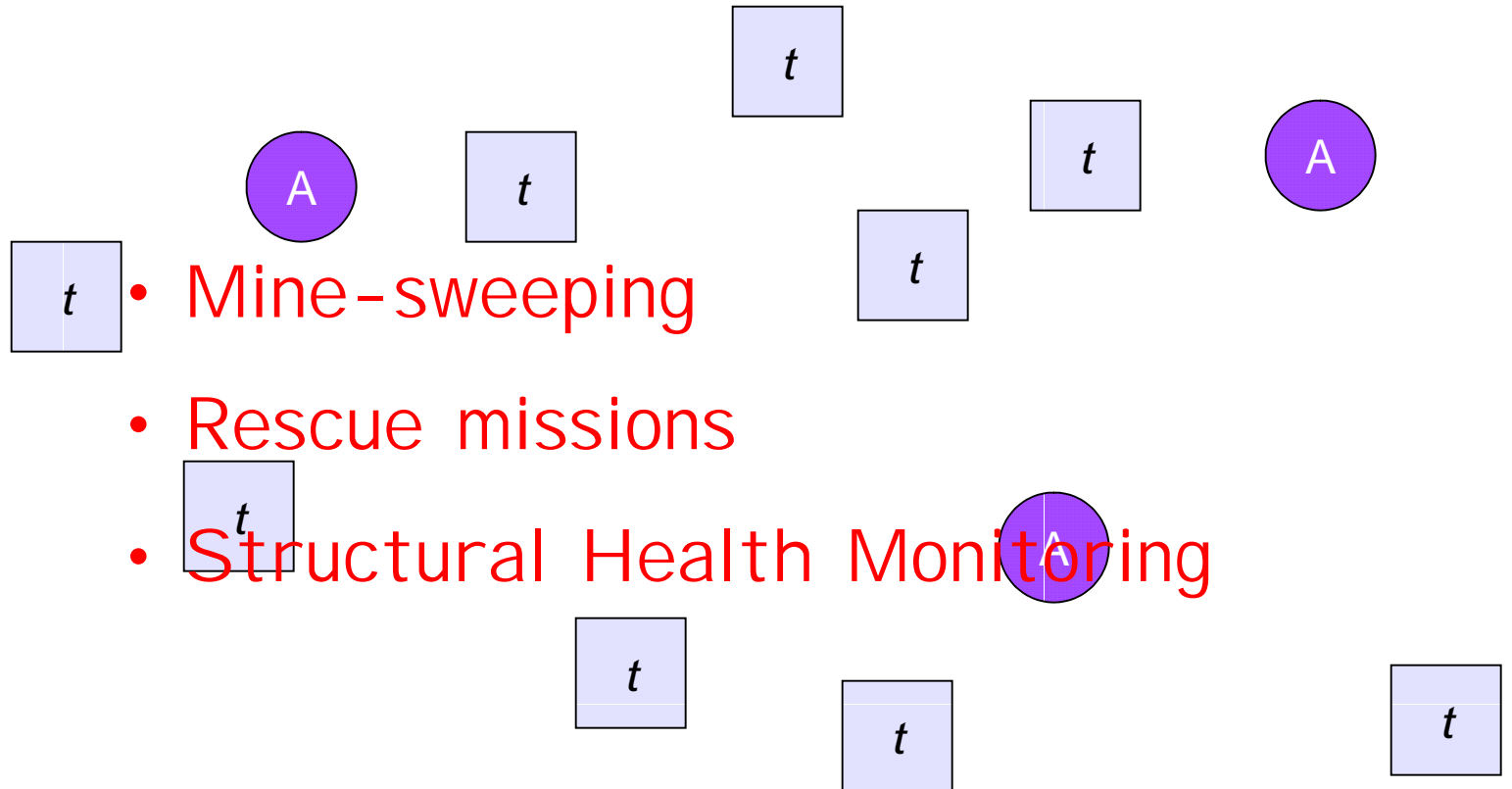
Open Systems Laboratory, UIUC  
<http://osl.cs.uiuc.edu>

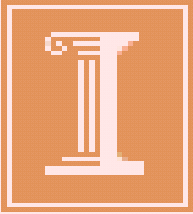
*\*Current Affiliation: Space Systems Finland Ltd.*

SASO 2007, Boston



# Problem Scenario

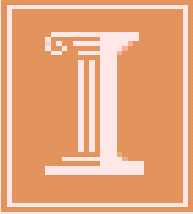




# Problem Definition



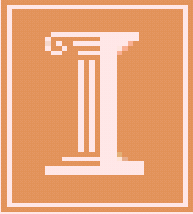
- *Goal:* Reach a globally efficient allocation of tasks to agents
- *Constraints:*
  - Agents exhibit self-interested behavior
  - Agents are computationally challenged
  - Large number of agents and tasks
    - Say,  $m=200$  agents and  $n=1000$  tasks



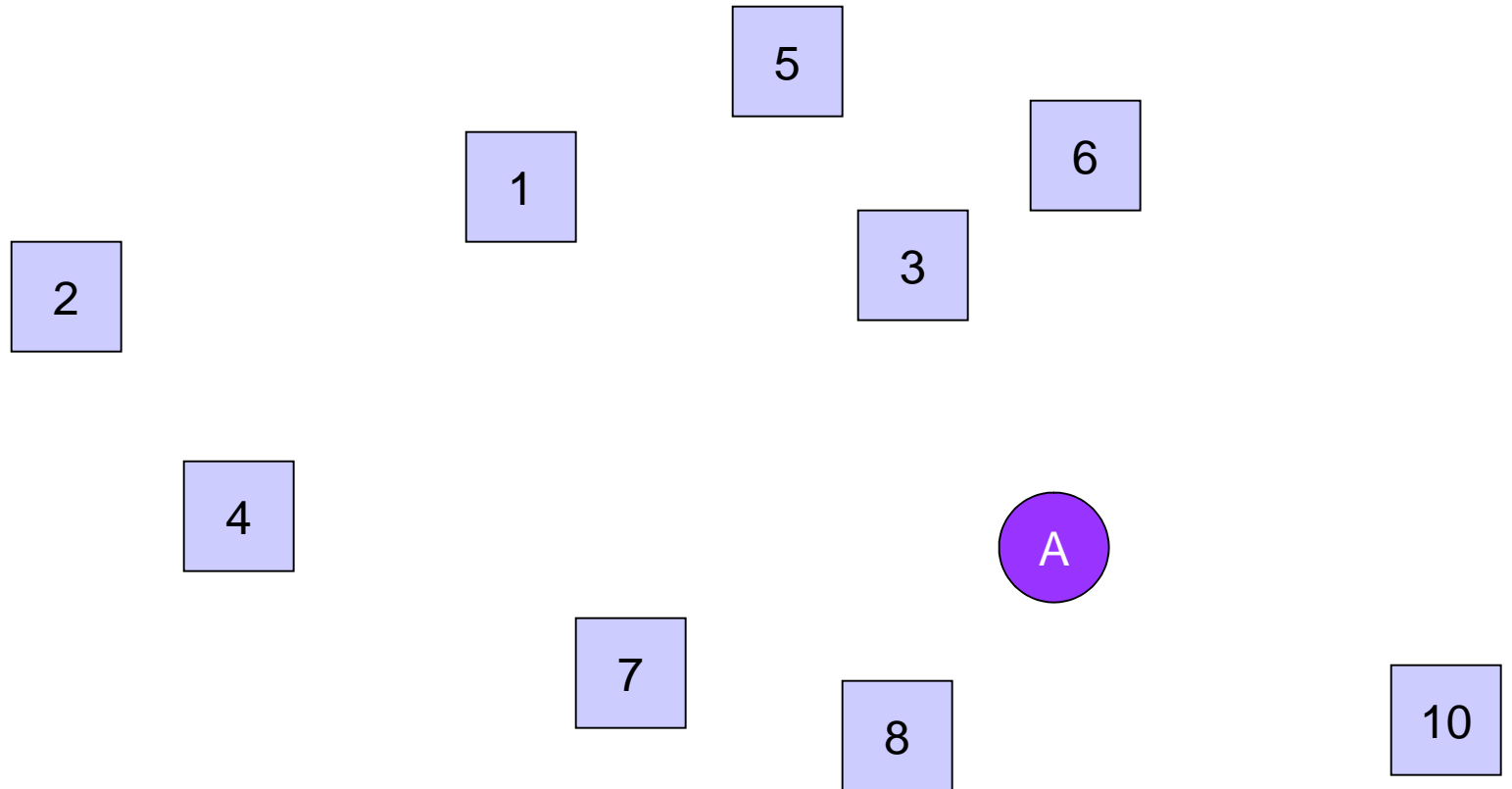
# Assumptions

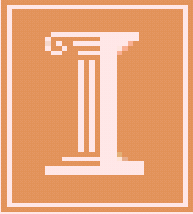


- Single-task agents (one at a time)
- Single-agent tasks
- Time-extended assignment
  - Agents plan for tours of tasks
  - Requires solving Traveling Salesman Problem (TSP)
- Static targets in a  $k$ -dimensional space
- Reliable communication
- Limited fuel for the tour (a constraint)

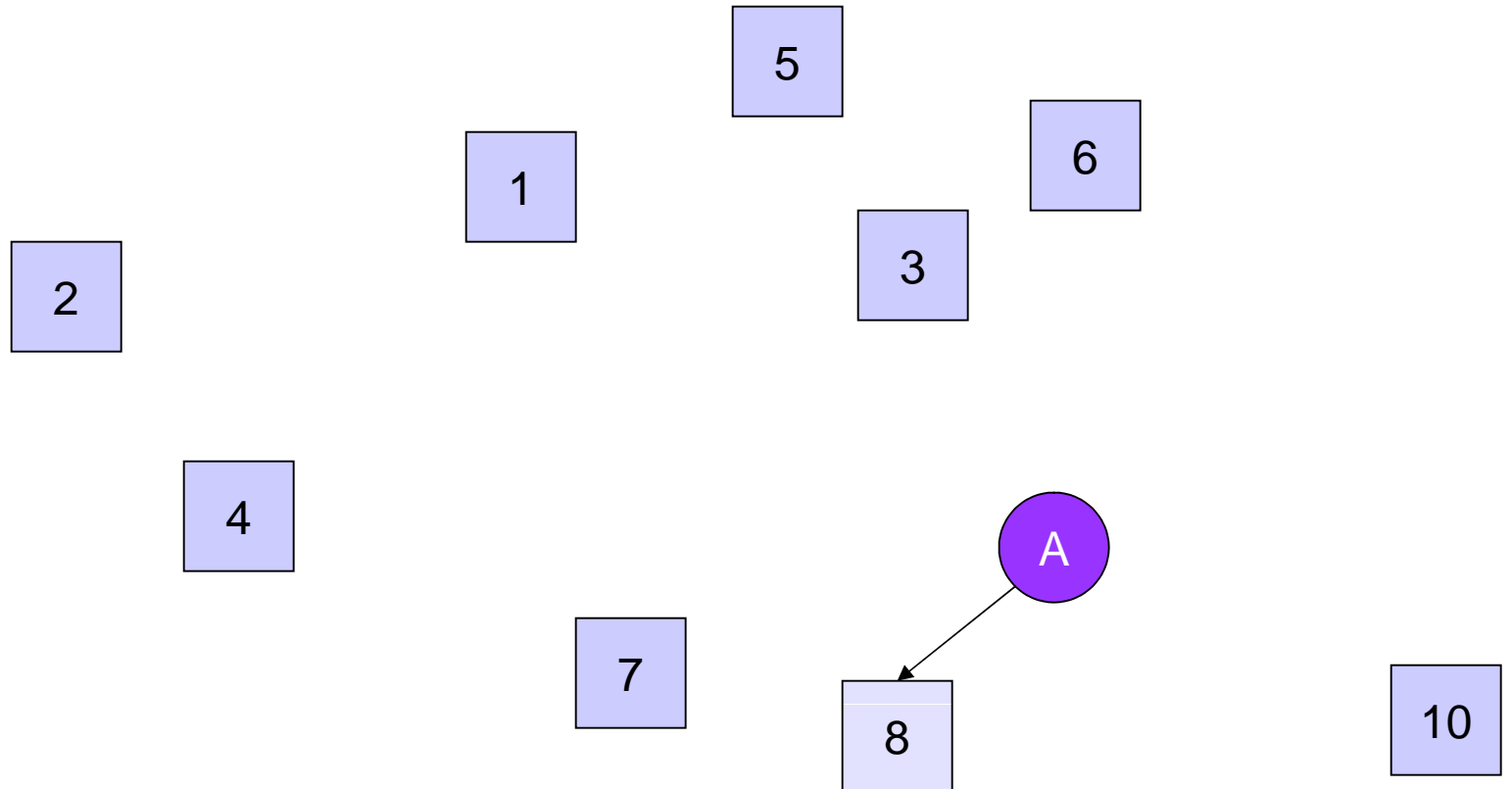


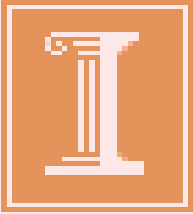
# Traveling Salesman Problem



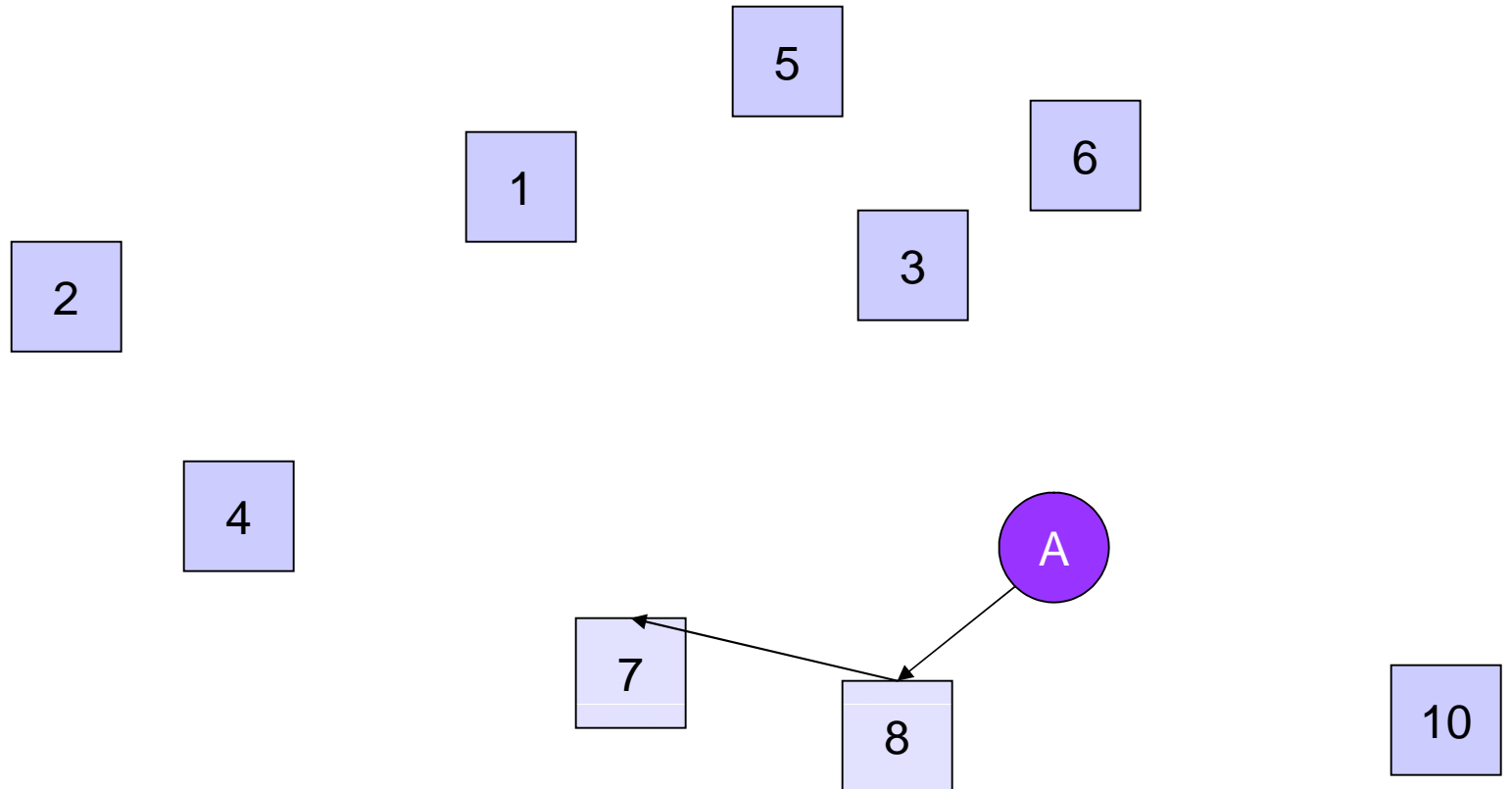


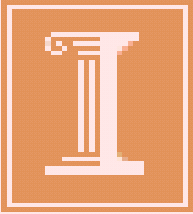
# Traveling Salesman Problem



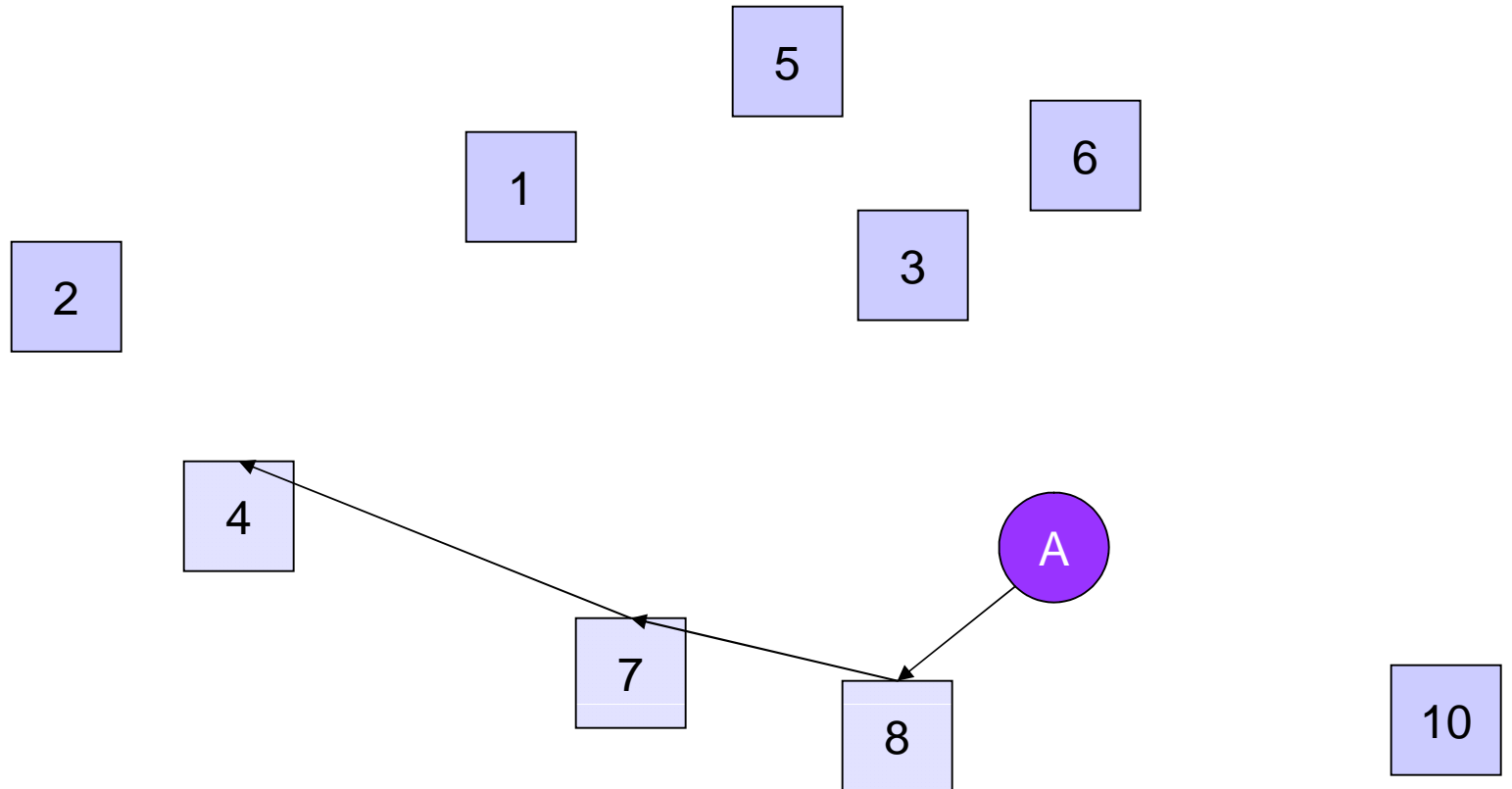


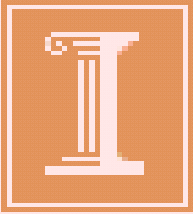
# Traveling Salesman Problem



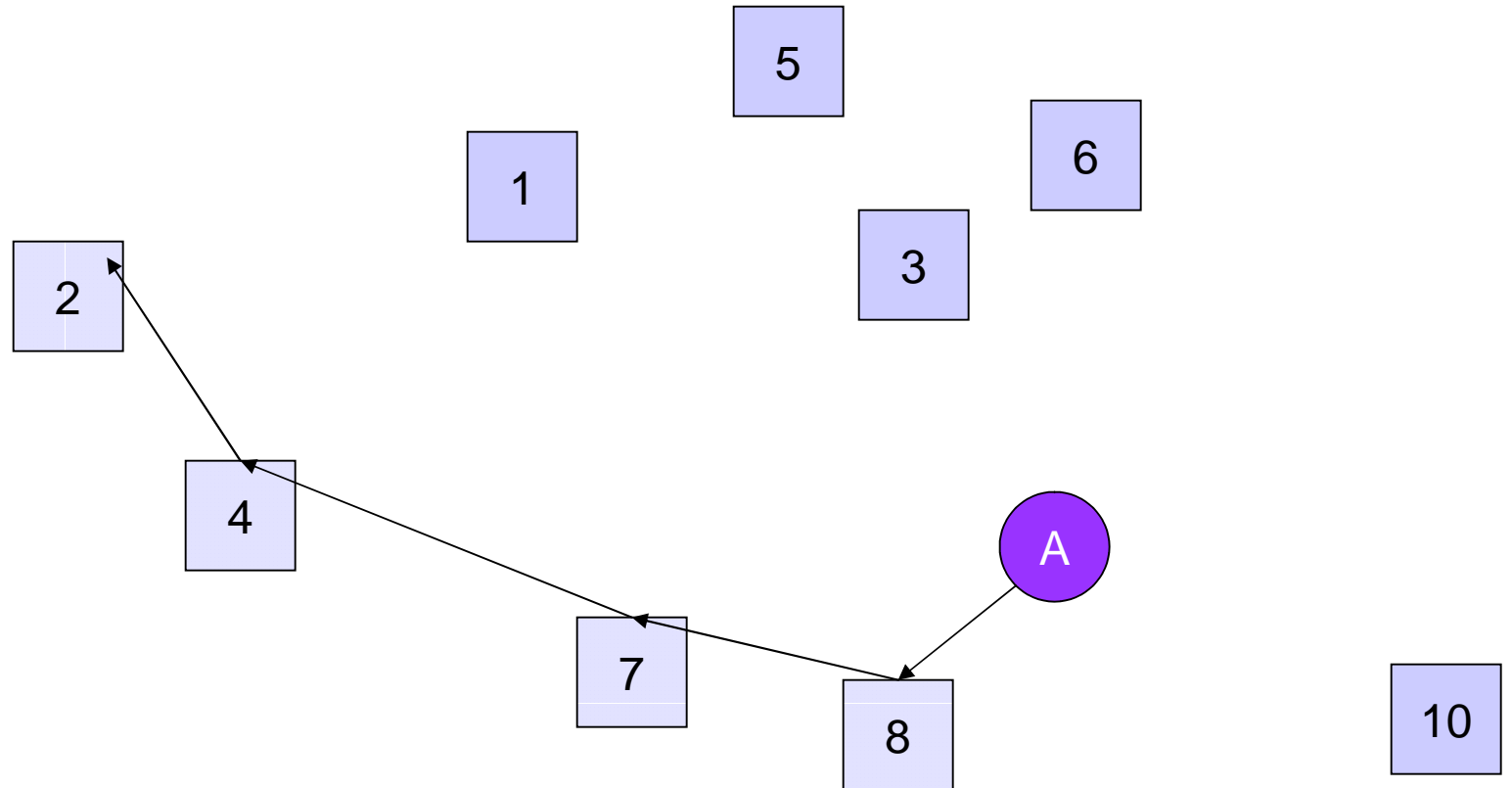


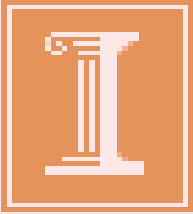
# Traveling Salesman Problem



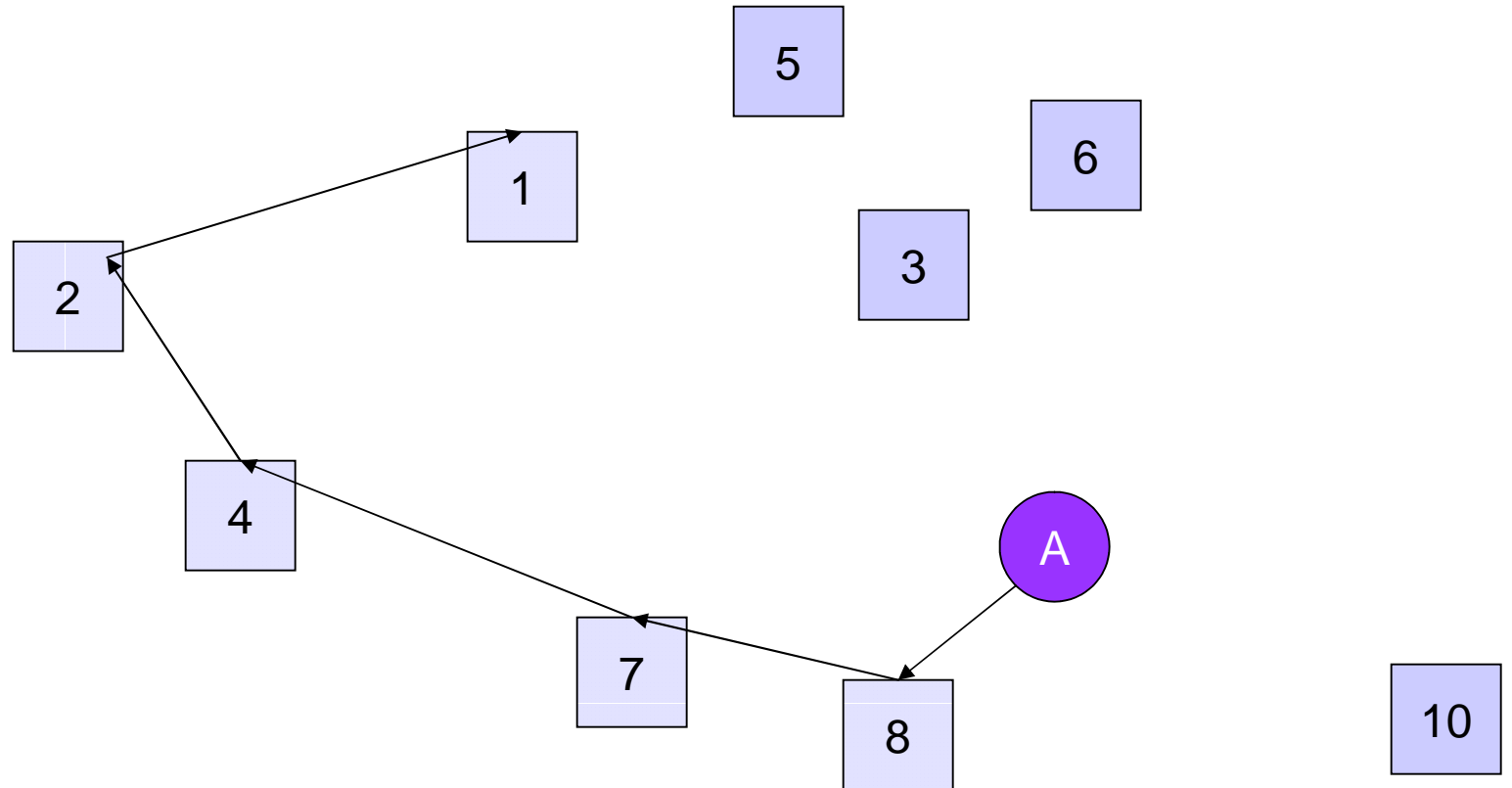


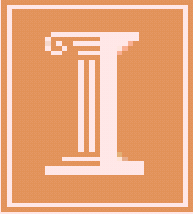
# Traveling Salesman Problem



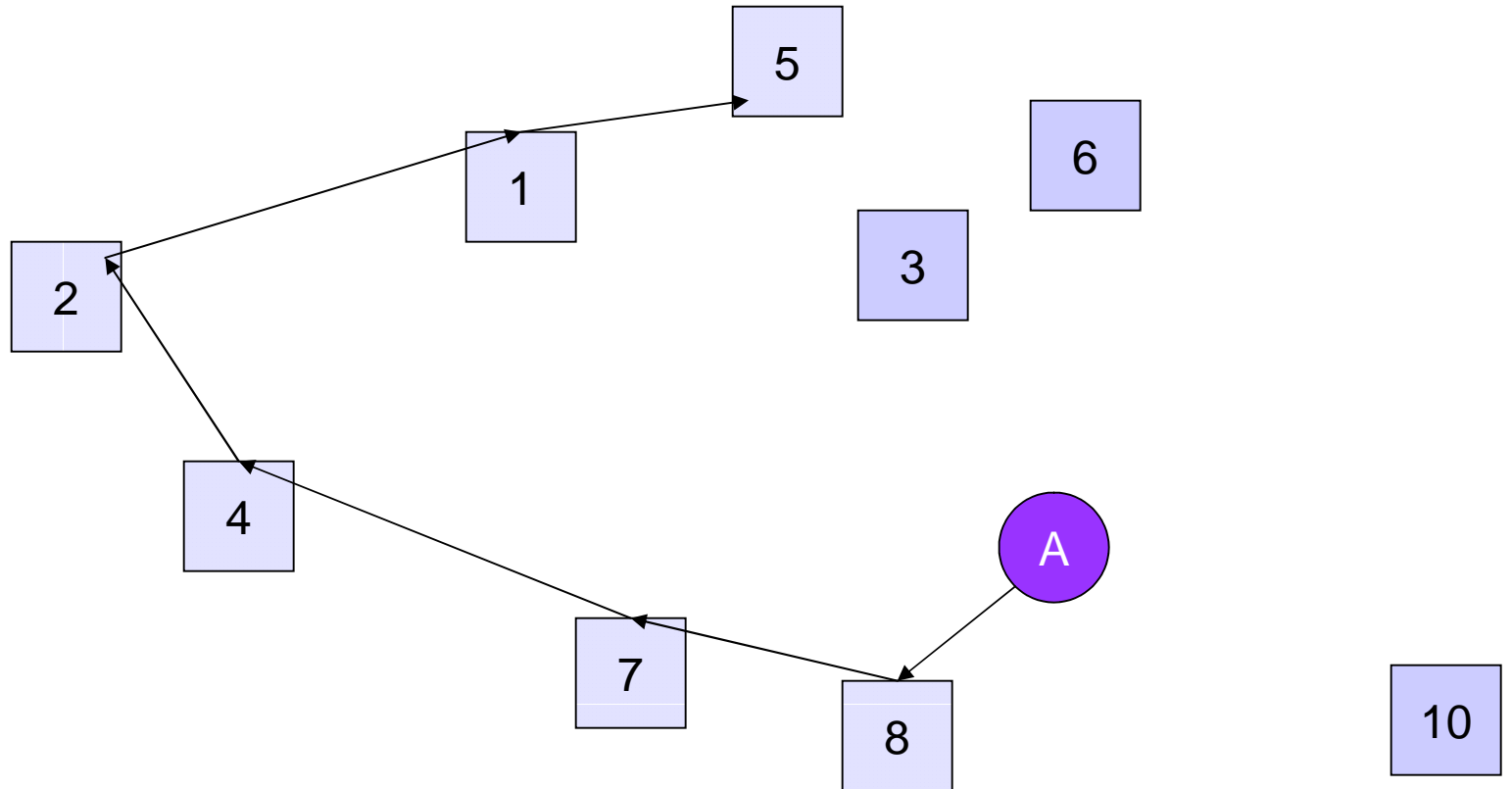


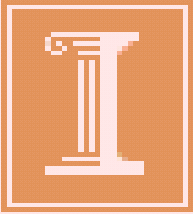
# Traveling Salesman Problem



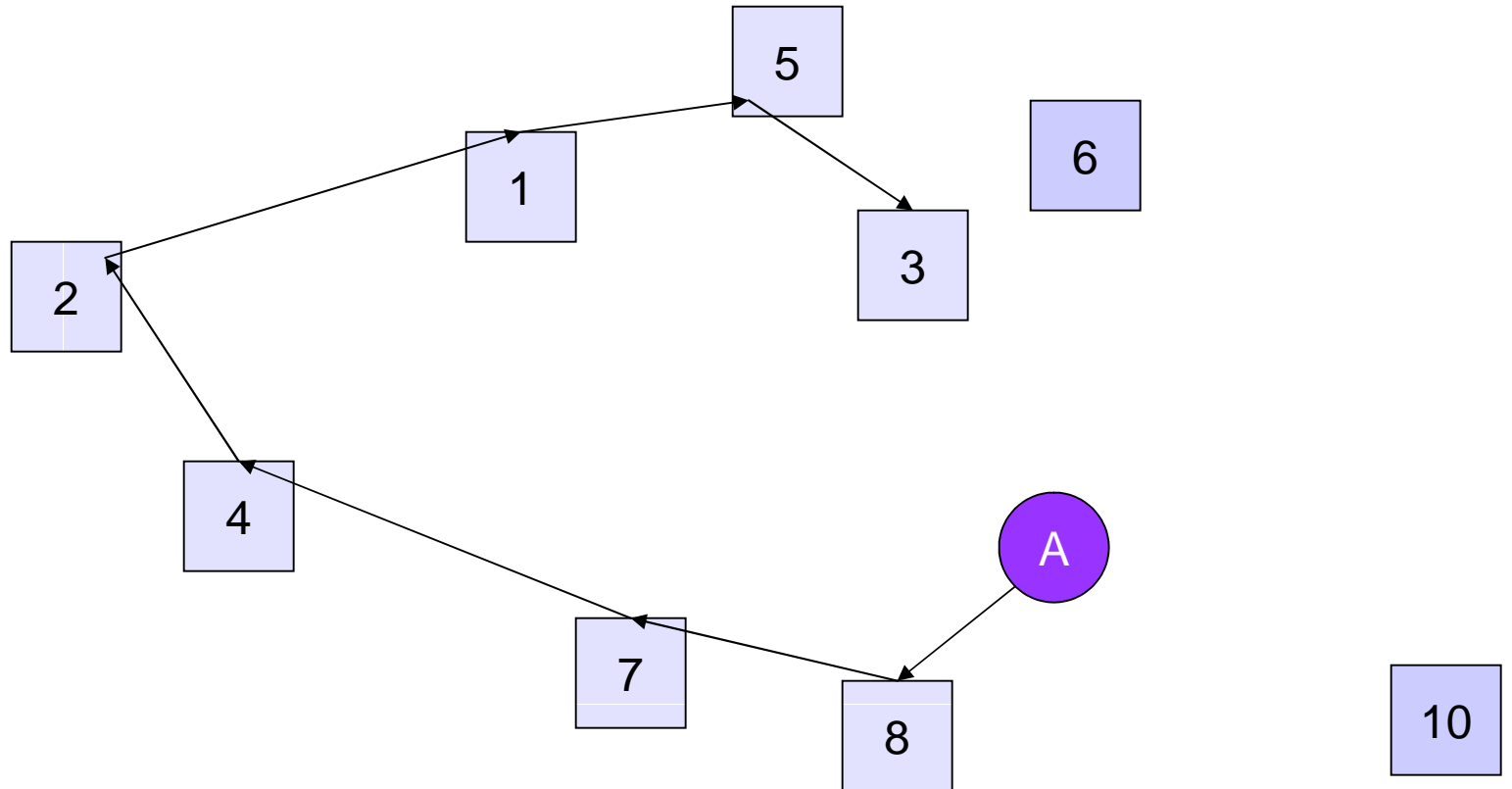


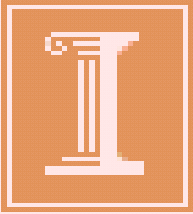
# Traveling Salesman Problem



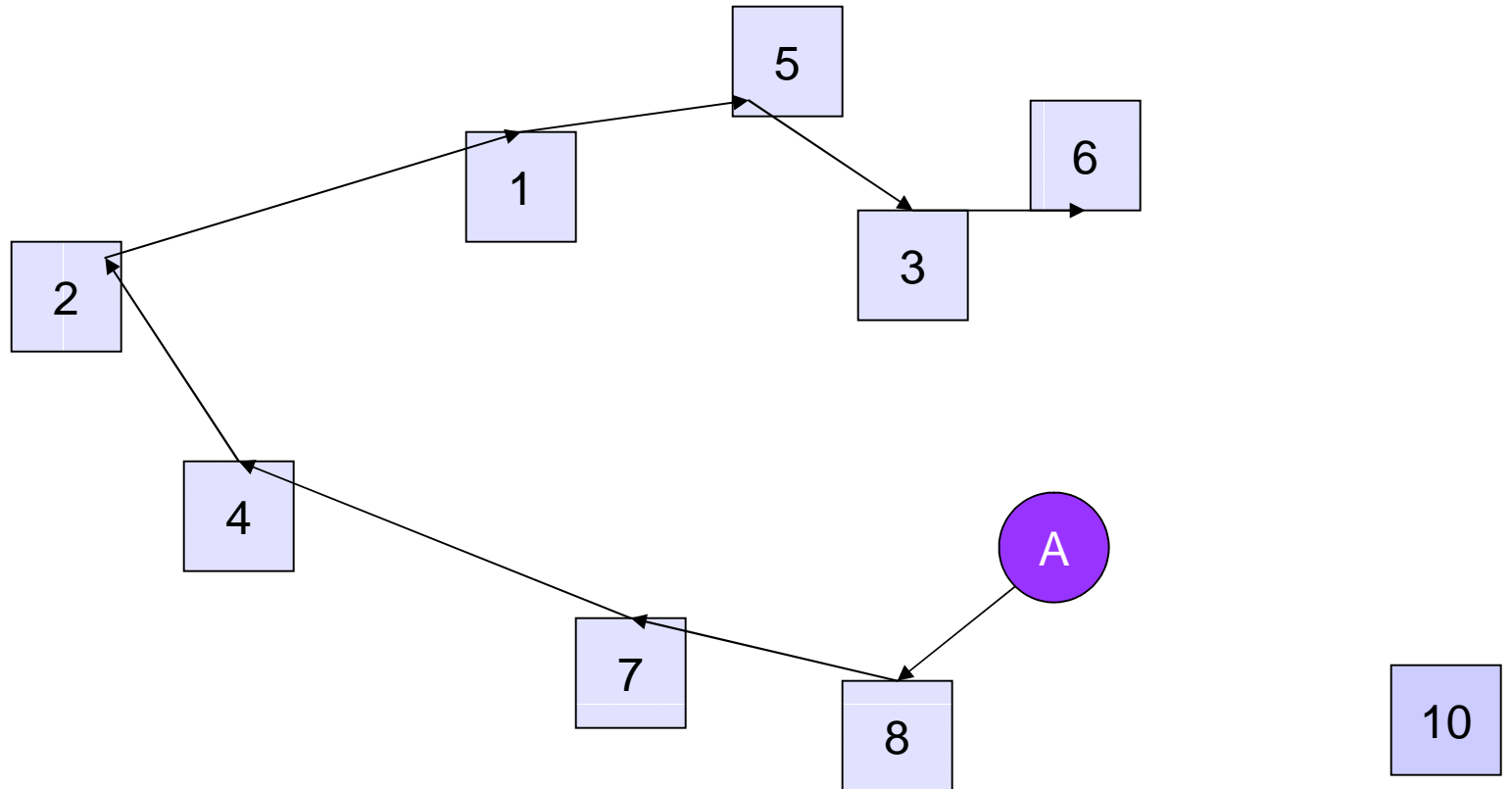


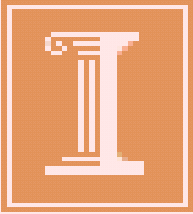
# Traveling Salesman Problem



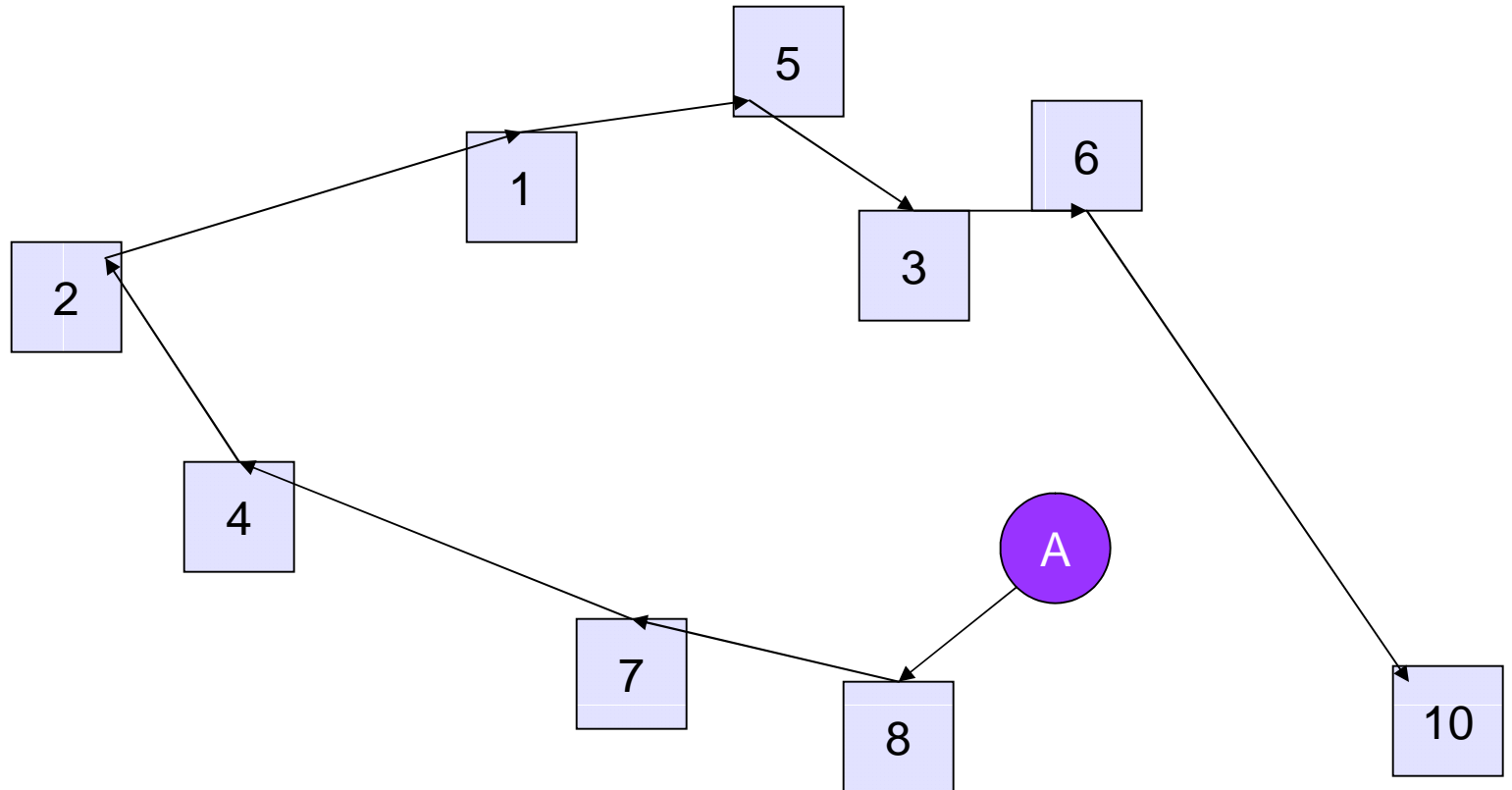


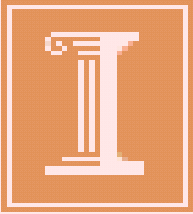
# Traveling Salesman Problem



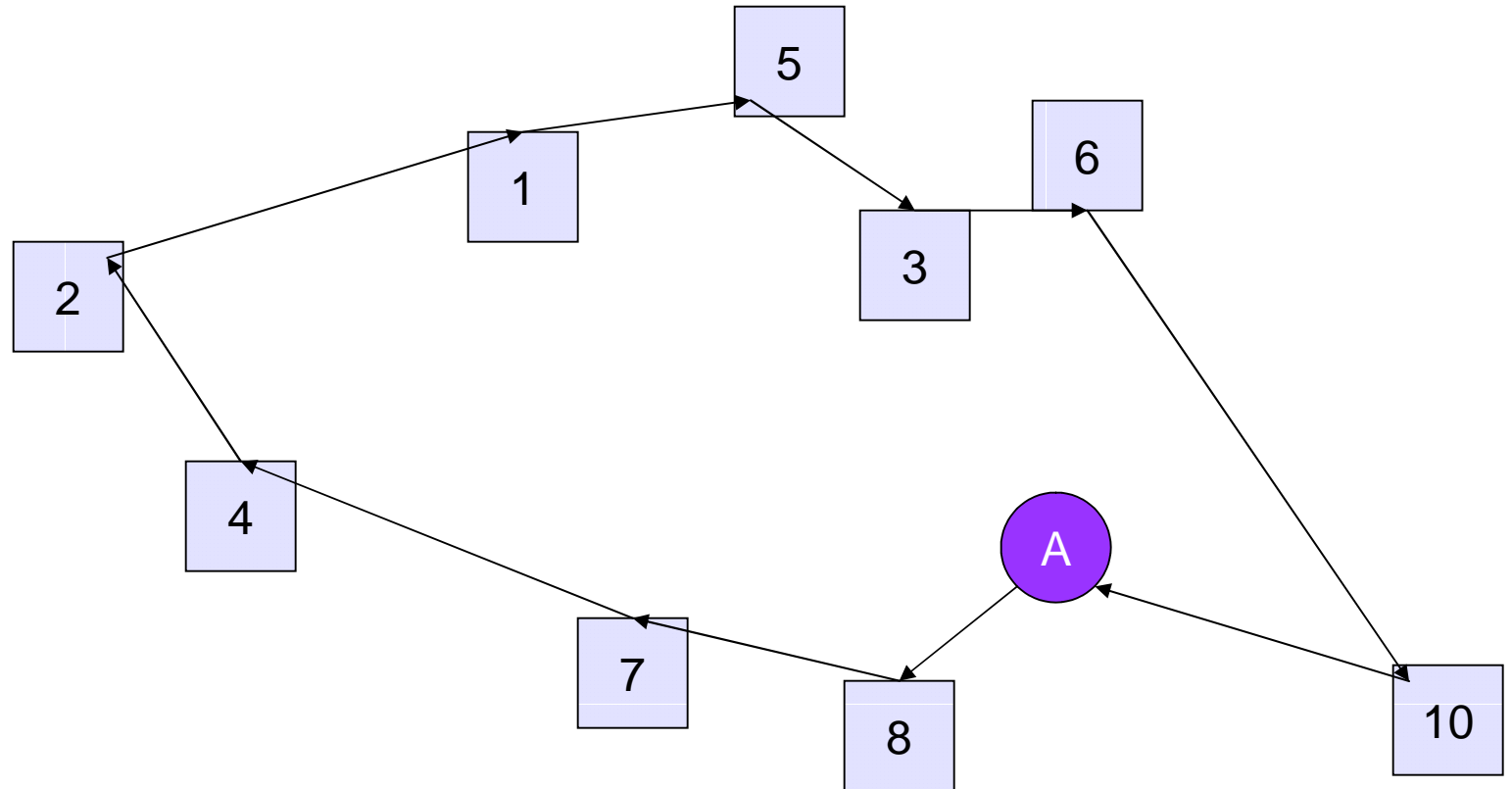


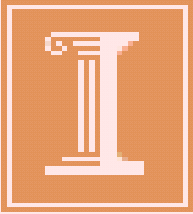
# Traveling Salesman Problem



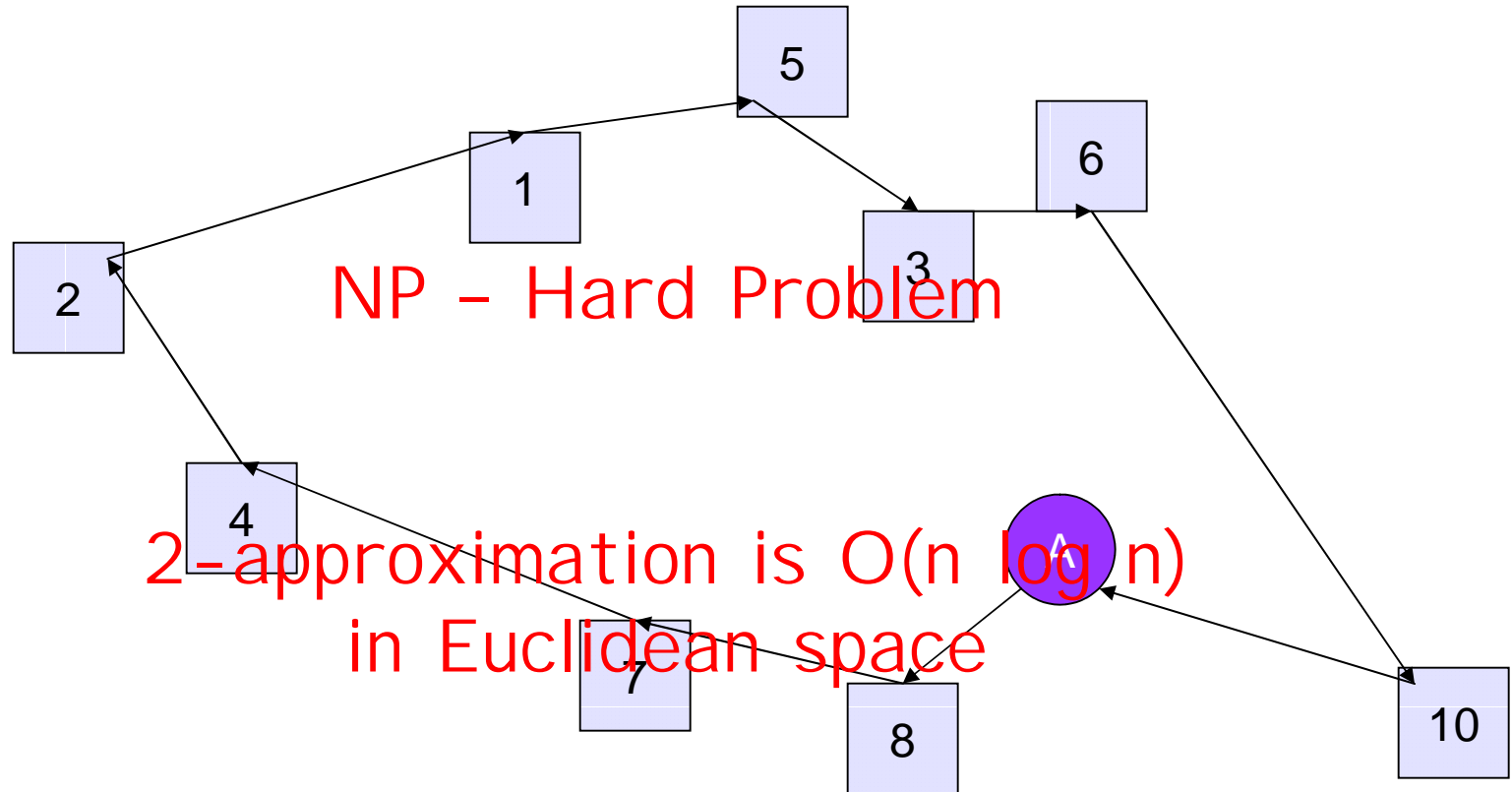


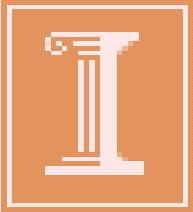
# Traveling Salesman Problem



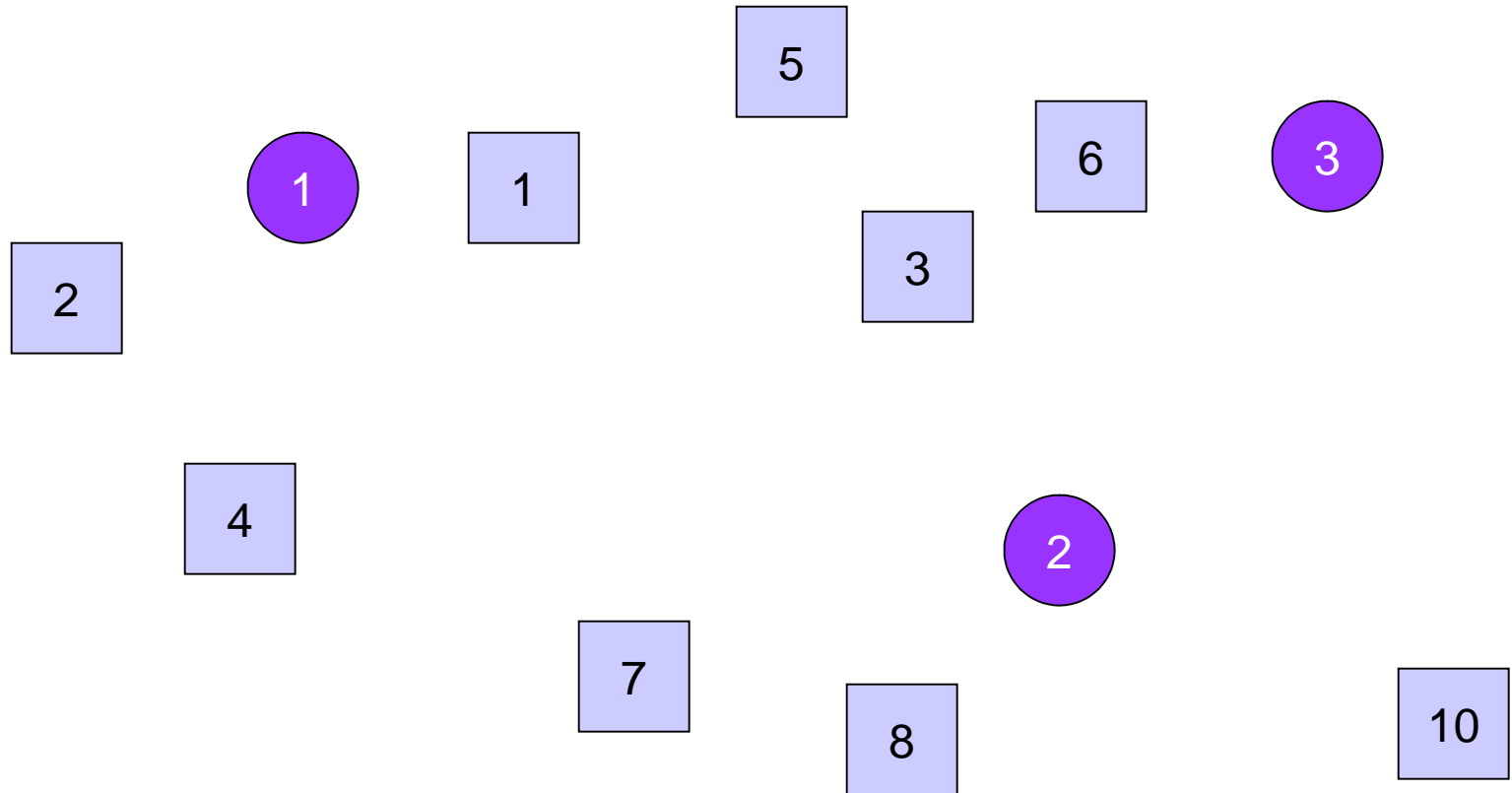


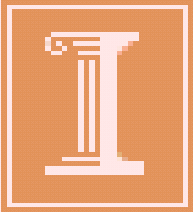
# Traveling Salesman Problem



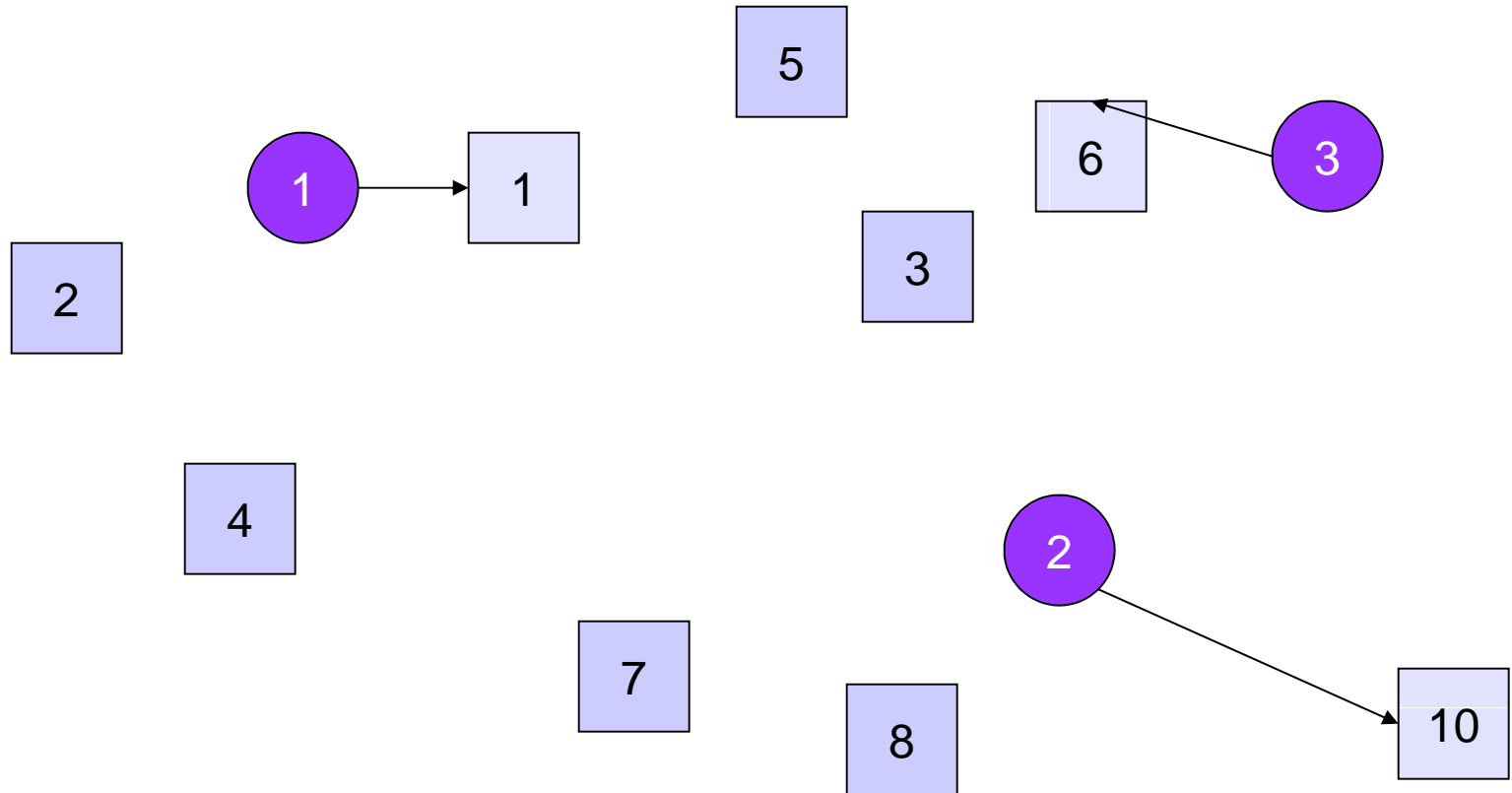


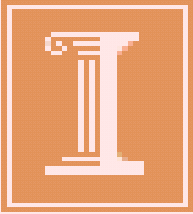
# Multi-Depot Traveling Salesman Problem



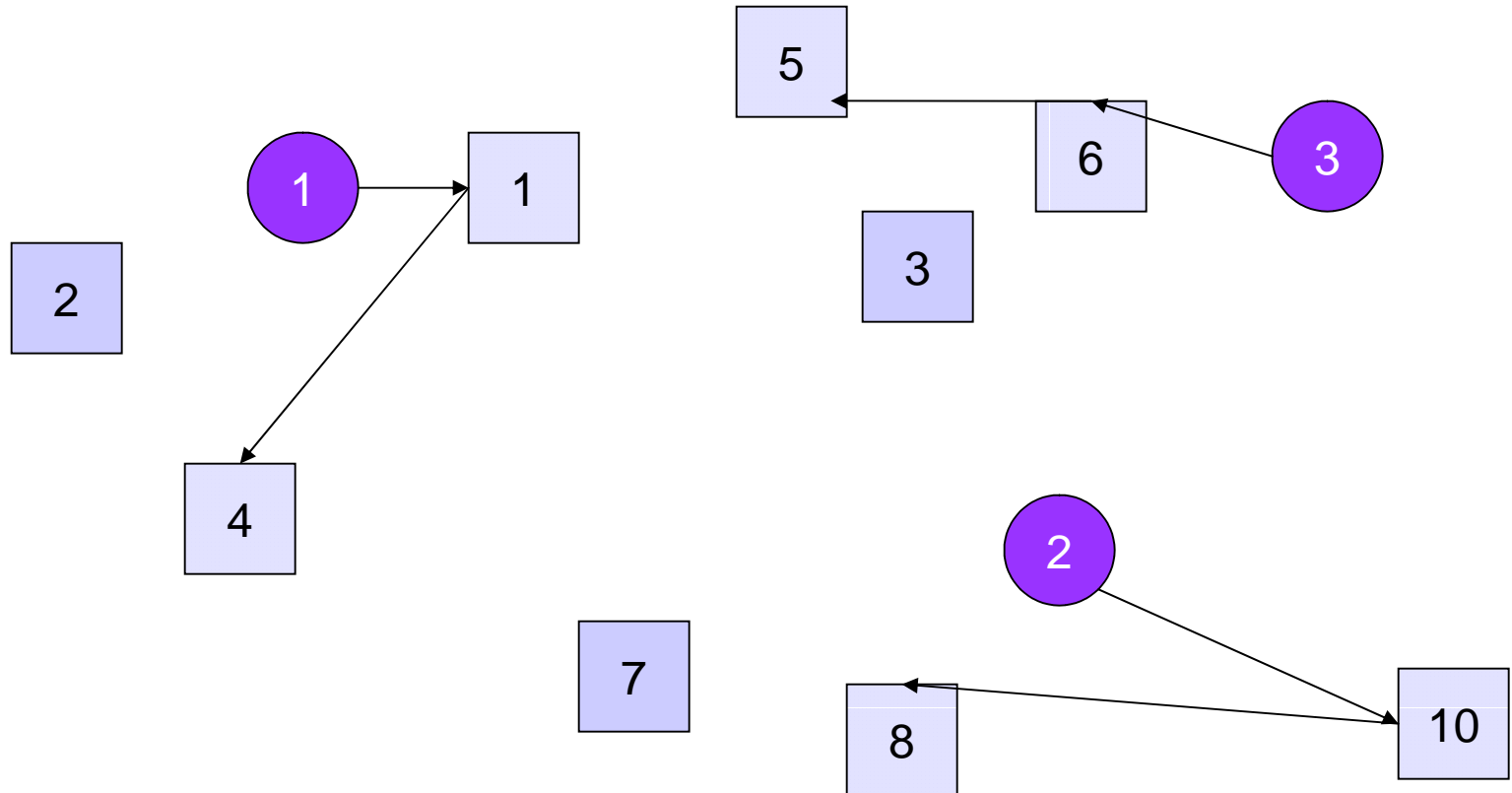


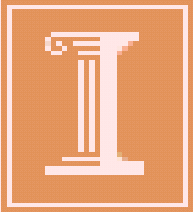
# Multi-Depot Traveling Salesman Problem



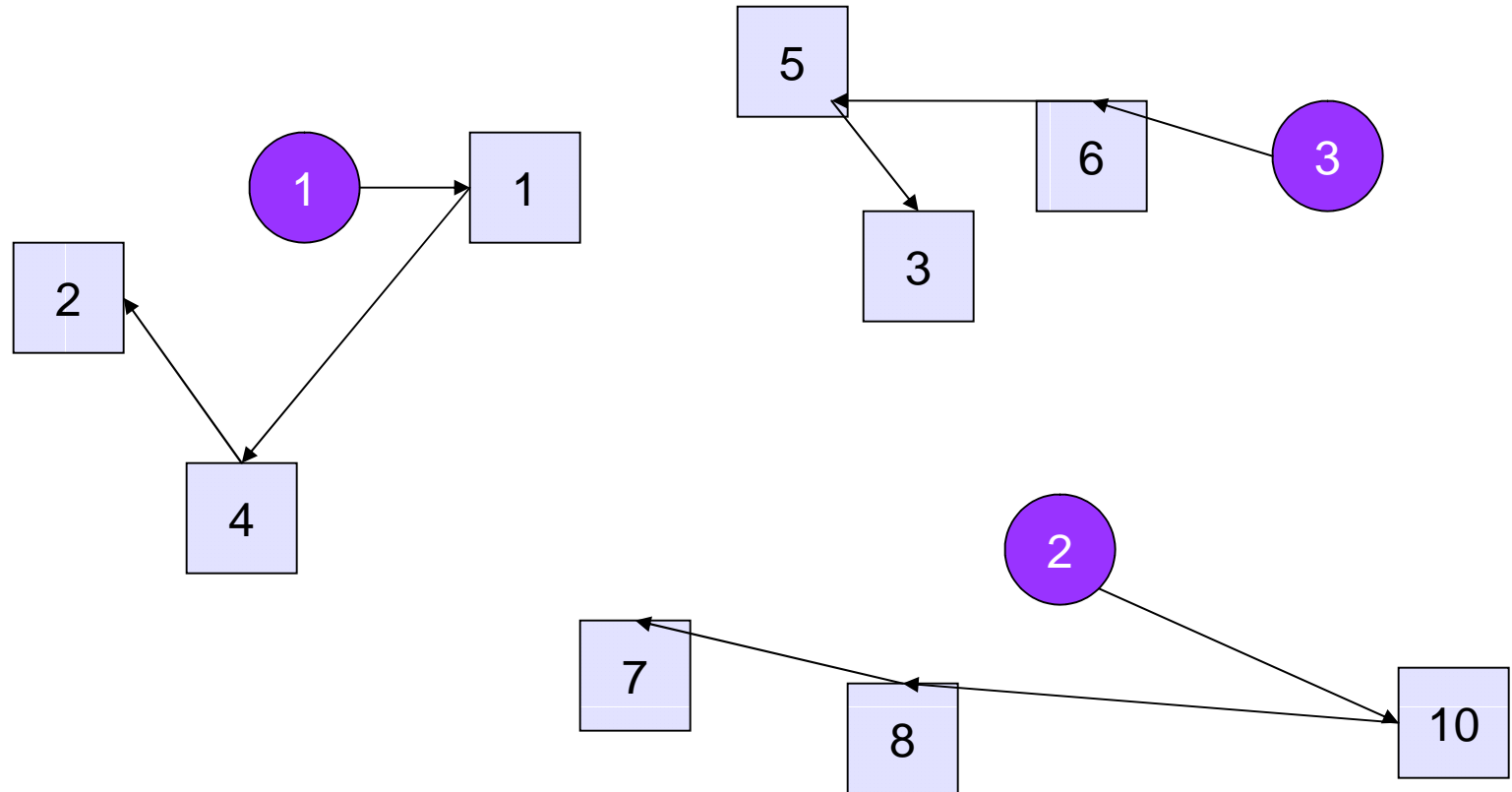


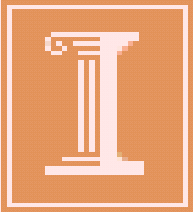
# Multi-Depot Traveling Salesman Problem



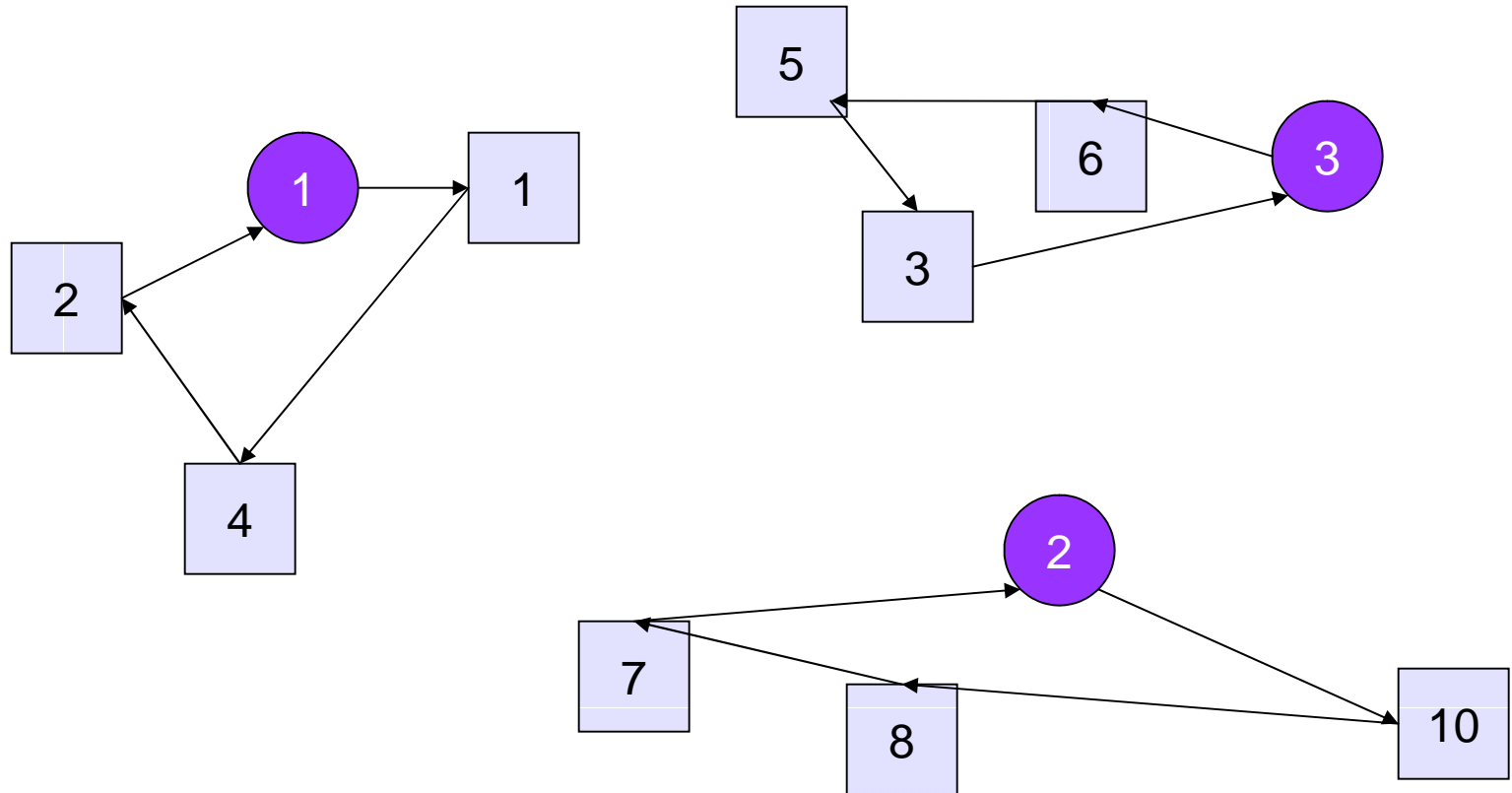


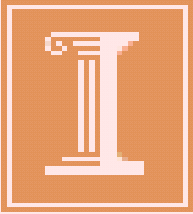
# Multi-Depot Traveling Salesman Problem





# Multi-Depot Traveling Salesman Problem





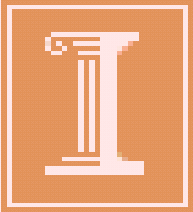
# First thoughts... Centralized, may be?



- Naïve centralized solution (brute force)
- $m^n$  task allocations
- computes  $m \cdot m^n$  TSPs ( $m \cdot 2^n$  with caching)

**Cost =  $O(m^{n+1} n \log n)$**

**Broadcast the solution!!**

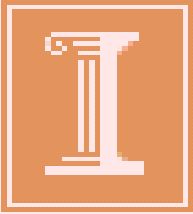


# First thoughts... Distributed, lets see!



- Agents determine the best tour for themselves
  - computing  $(2^n - 1)$  TSPs
- Cost =  $O(2^n n \log n)$

Total cost for  $m$  agents =  $O(m 2^n n \log n)$   
Still have to negotiate towards their best tour

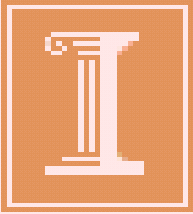


# Other approaches



- Combinatorial auctions
  - Optimal solution
  - Bid computation is exponential
  - Clearance is *NP-Complete*
- Other auction mechanism
  - Bounds on solution quality [Lagoudakis et al., IROS '04]

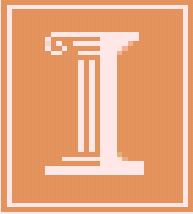
Marginal cost based reallocation  
negotiation to improve solutions



# What hinders scalability?



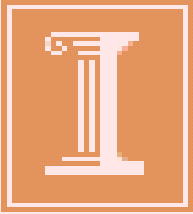
- Evaluating potential contracts involve expensive TSP computations
- Options?
  - 2-approximation TSP algorithm
  - Not scalable,  $O(10^4)$  for each contract



# What hinders scalability?

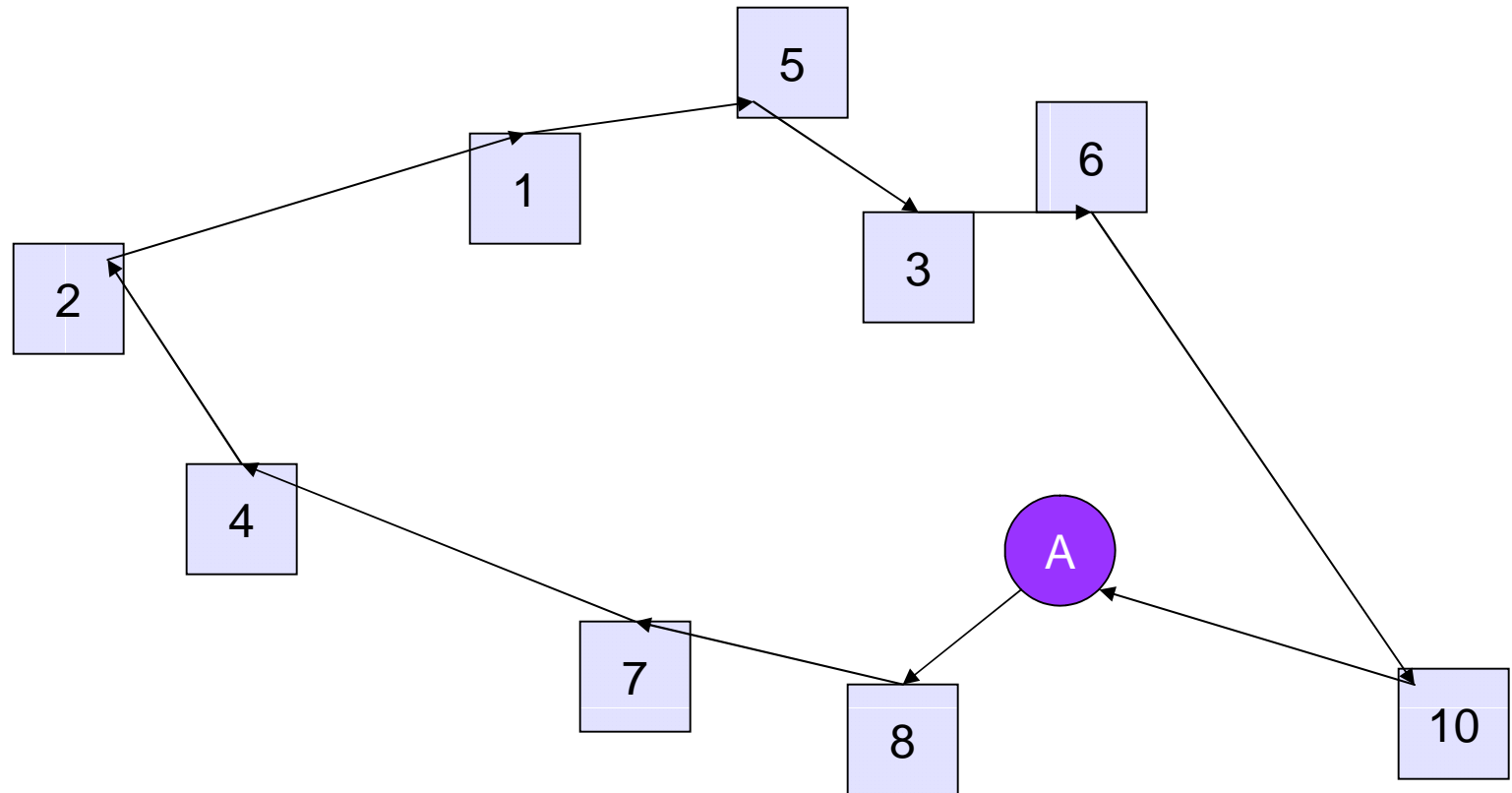


- Evaluating potential contracts involve expensive TSP computations
- Options?
  - Insertion heuristics
  - Cost is  $O(n^2)$

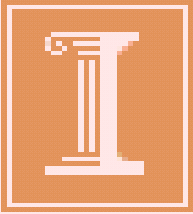


# Universal TSP Algorithm

[Jia et al, STOC '05]

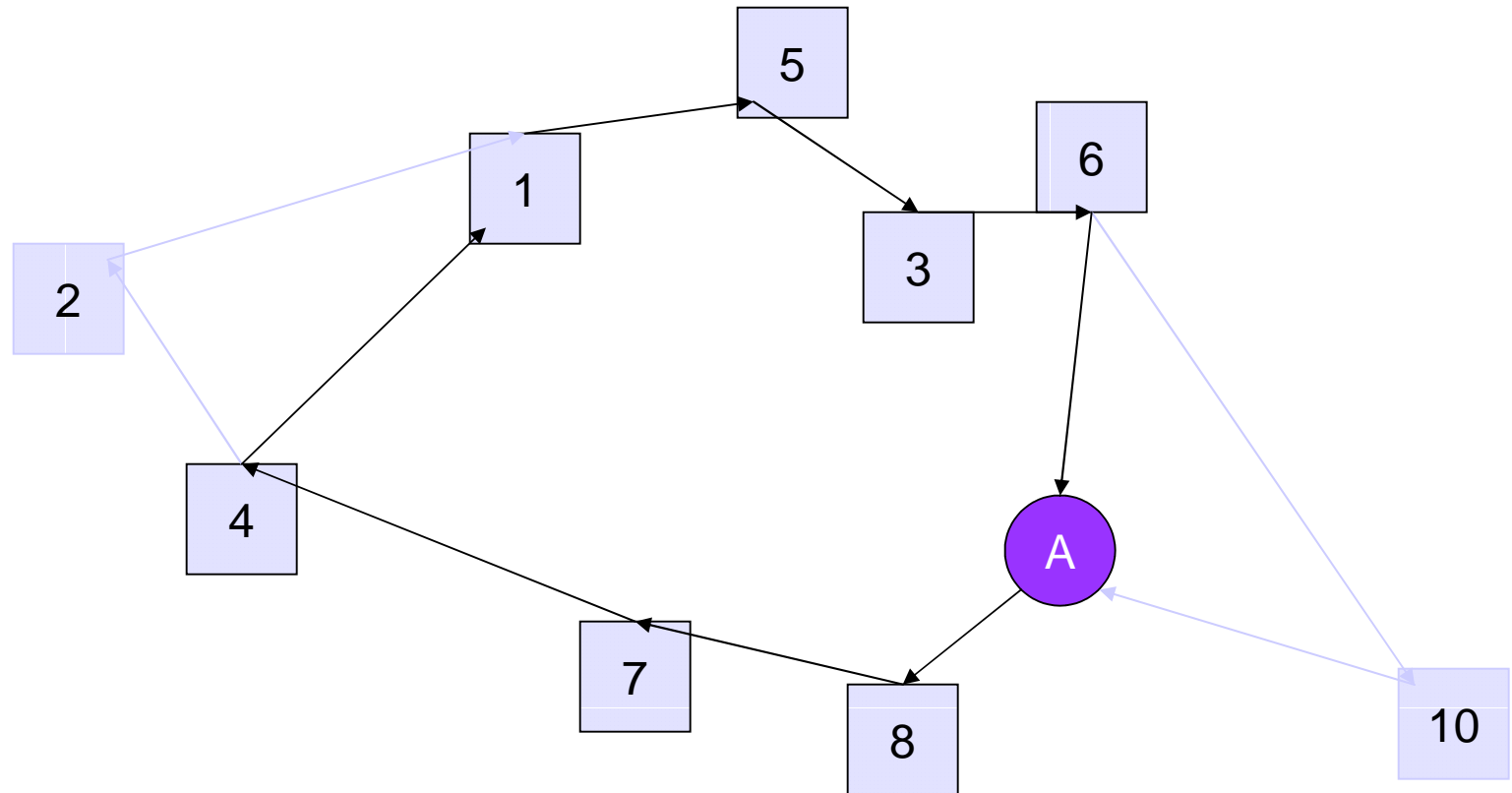


Polynomial time algorithm

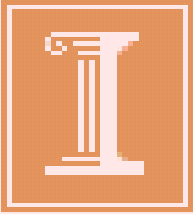


# Universal TSP Algorithm

[Jia et al, STOC '05]

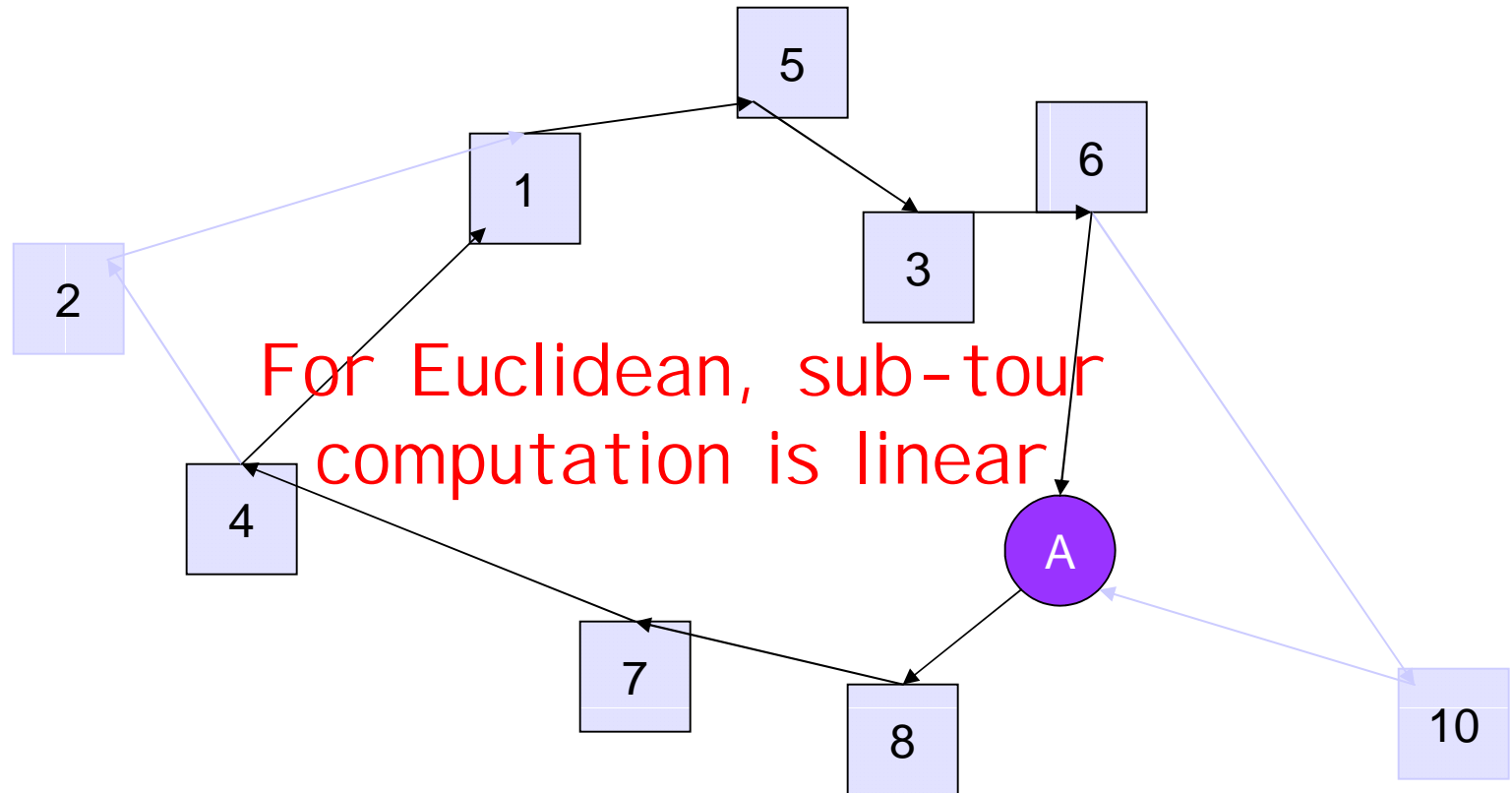


Polynomial time algorithm

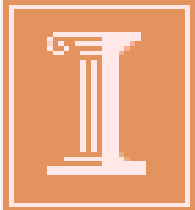


# Universal TSP Algorithm

[Jia et al, STOC '05]



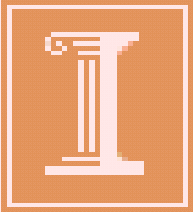
Polynomial time algorithm



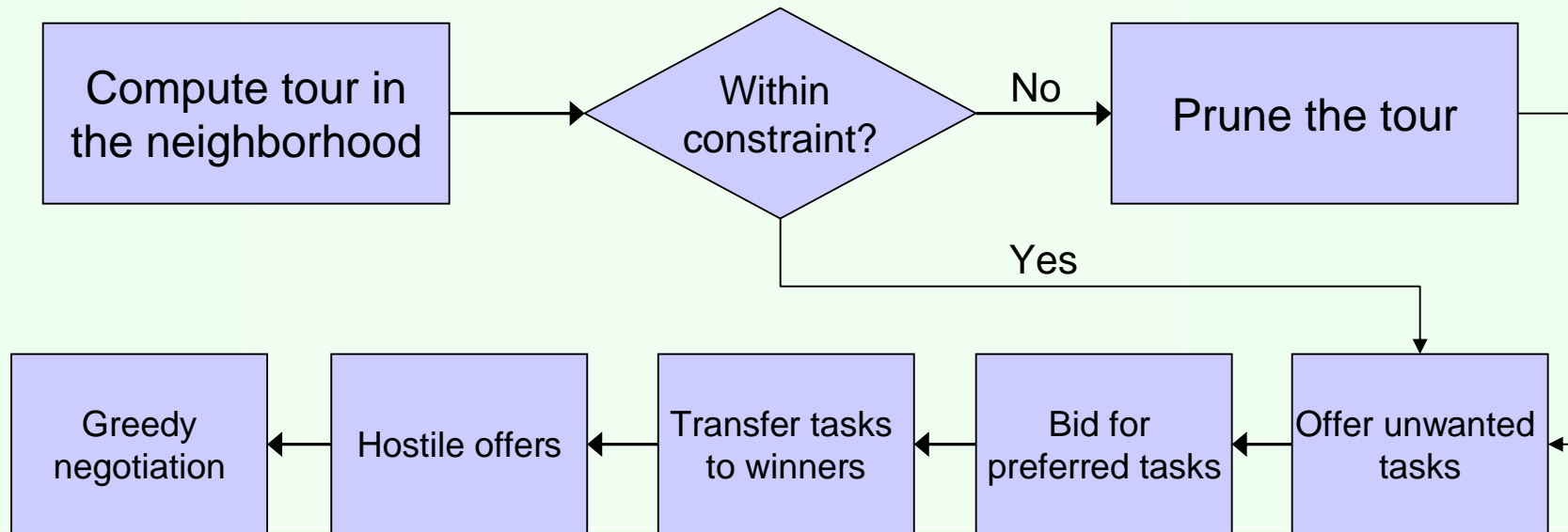
## Market-based Approach with Look ahead Agents (MALA)

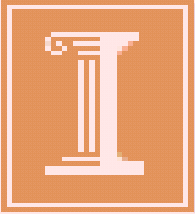


- Agents compute the best tour for themselves using *modified* Universal TSP
- Negotiate their way from the initial allocation  $R$  to their preferred allocation  $T$
- Complete the tour
- **Caveat: Not all agents would be able to reach their target allocation (due to conflicting preferences)**

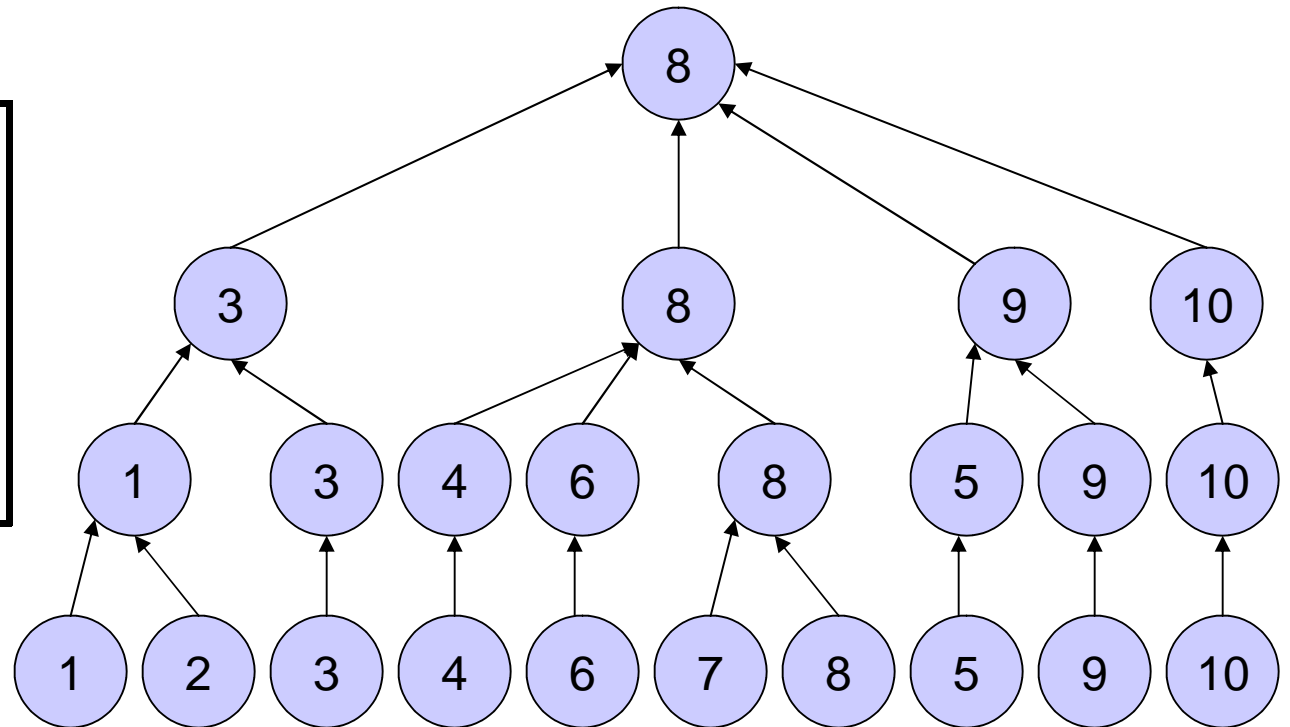
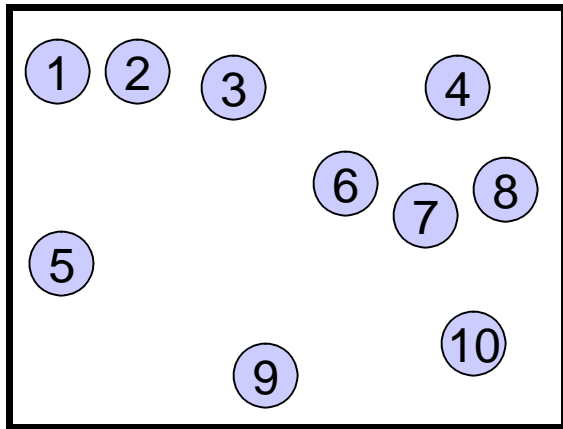


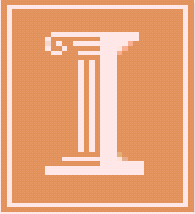
# Overview of the Approach



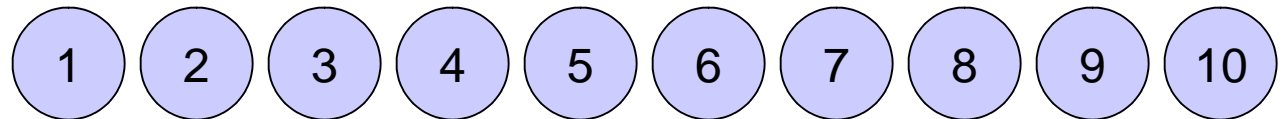
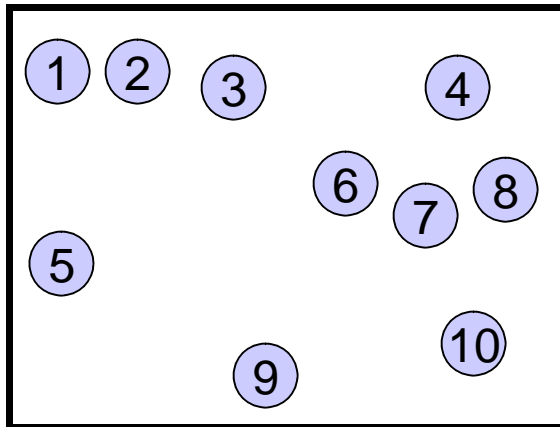


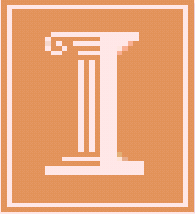
# U-TSP: Tree Data Structure





# U-TSP: Construction of Tree

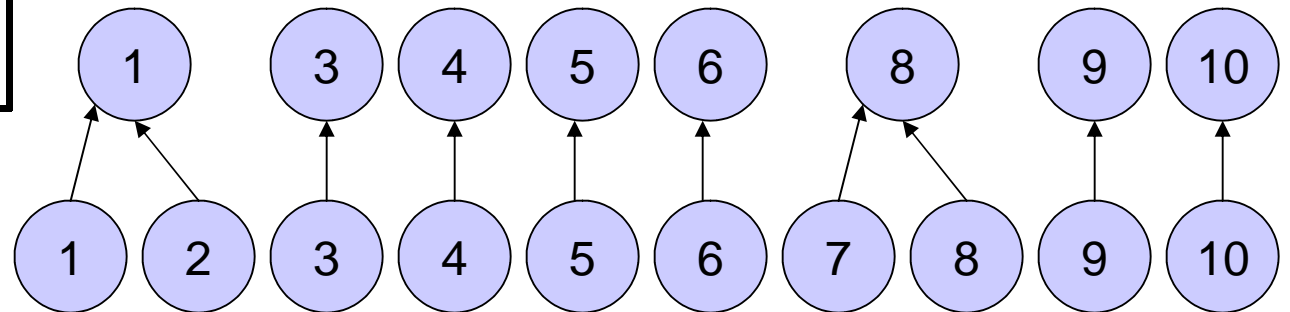


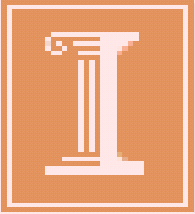


# U-TSP: Construction of Tree

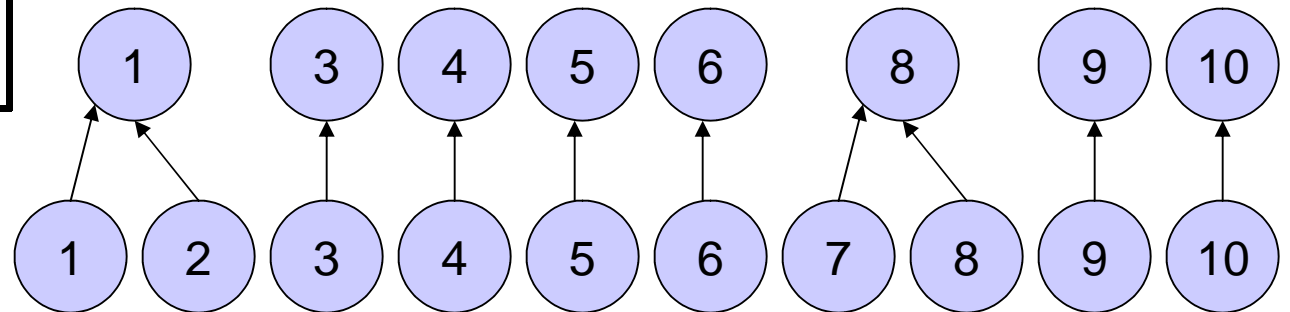
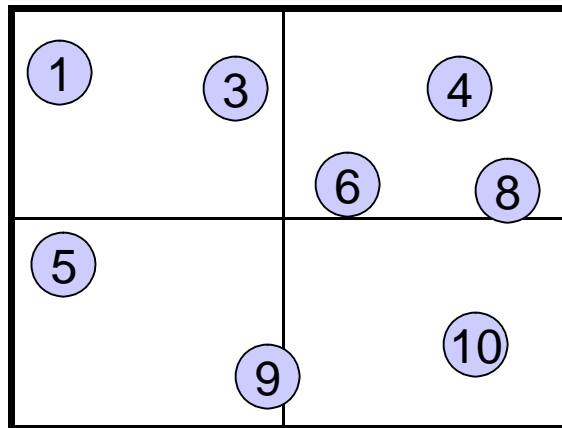


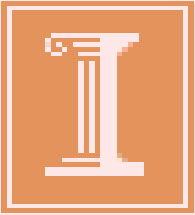
1 2	3	4
5	6	7 8
	9	10



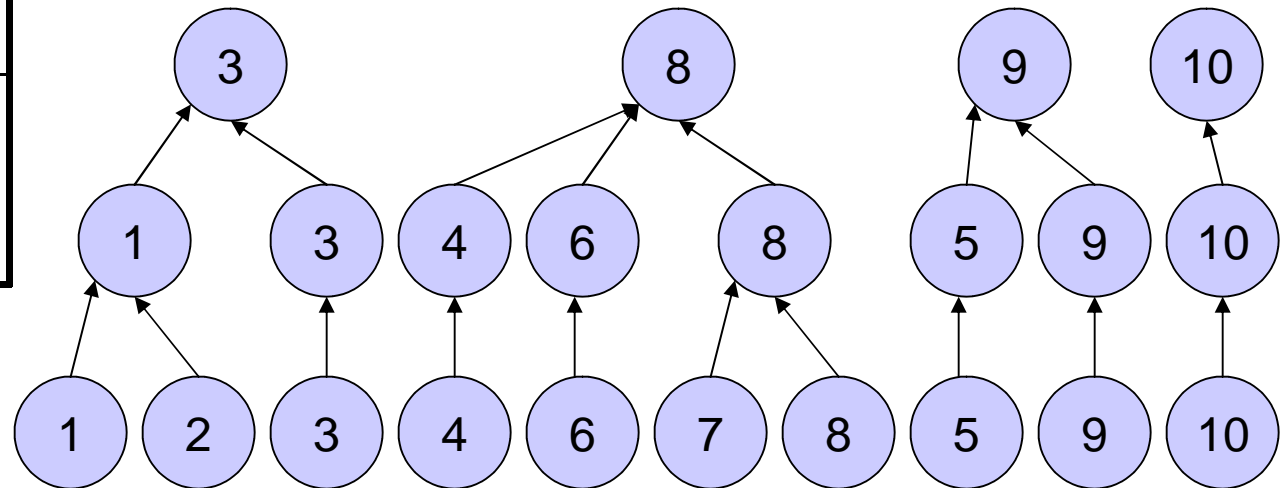
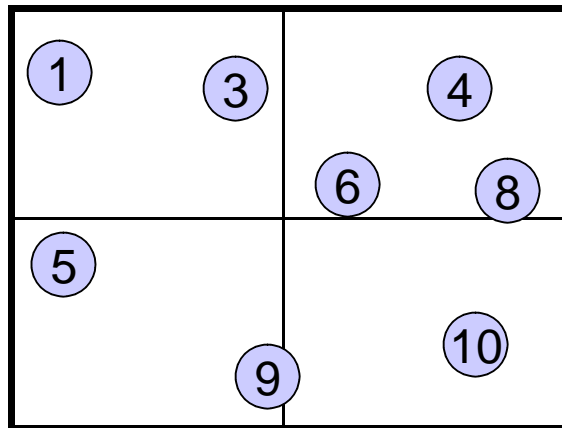


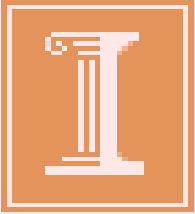
# U-TSP: Construction of Tree



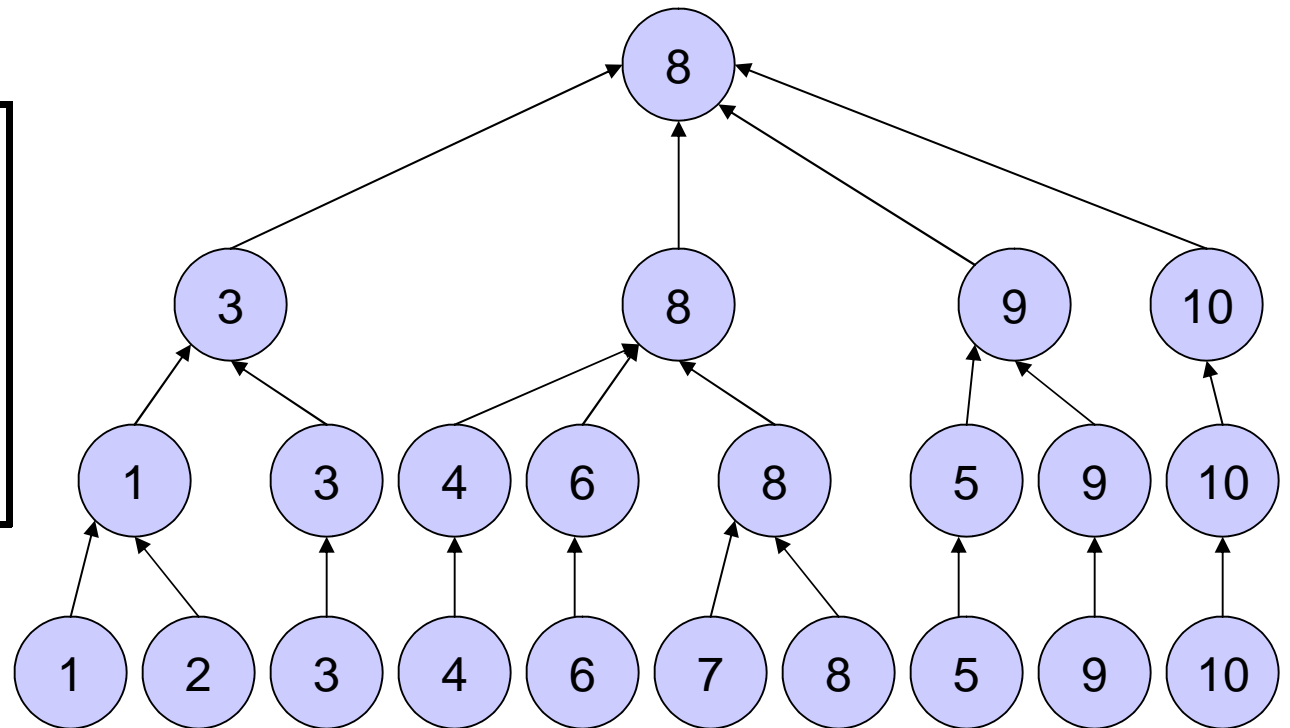
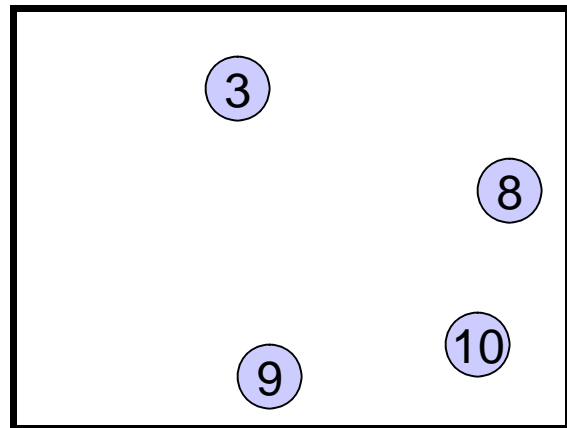


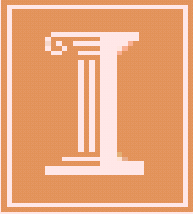
# U-TSP: Construction of Tree



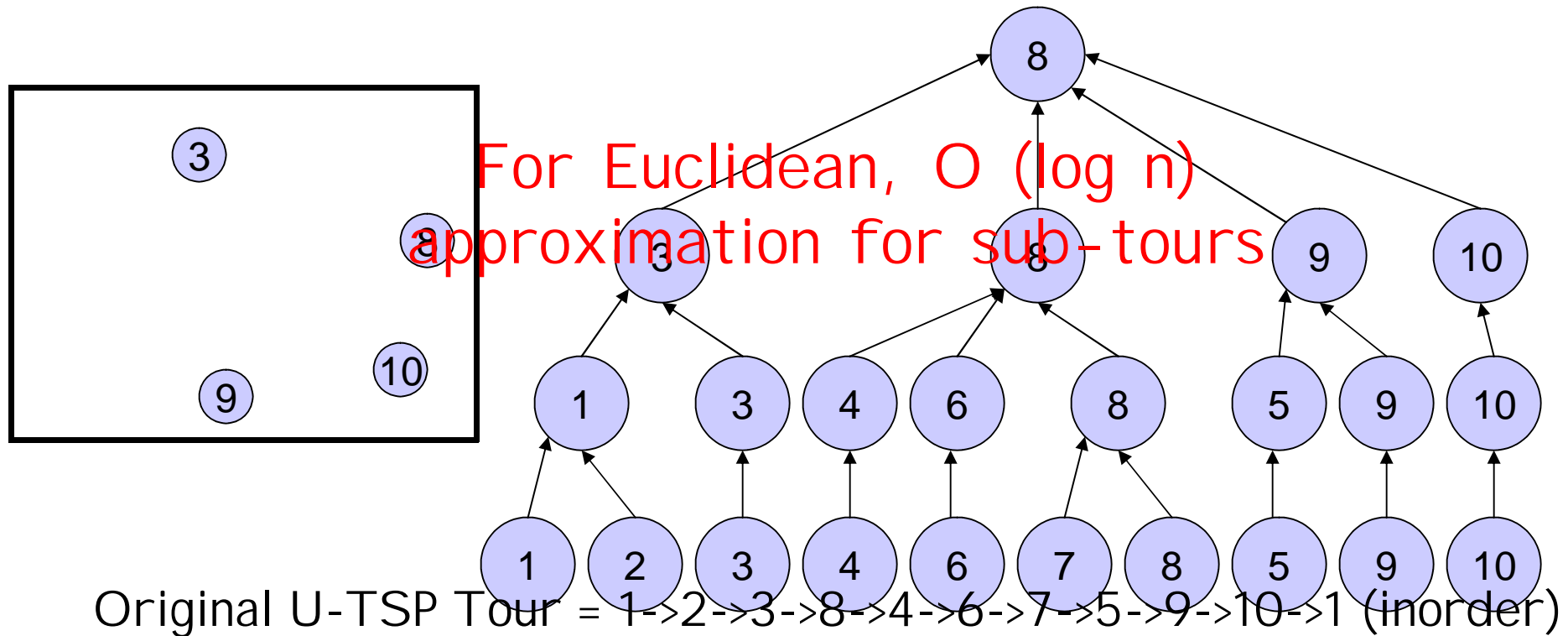


# U-TSP: Construction of Tree





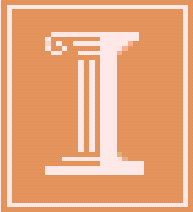
# U-TSP: Construction of Tree



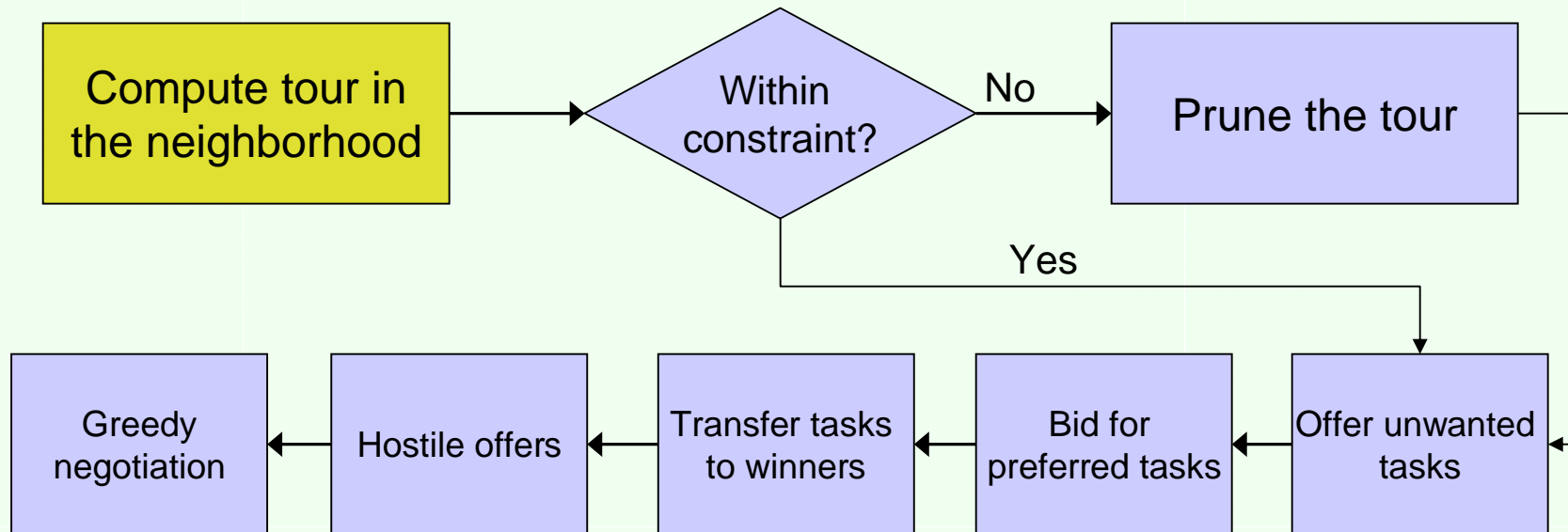
Original U-TSP Tour = 1->2->3->8->4->6->7->5->9->10->1 (inorder)

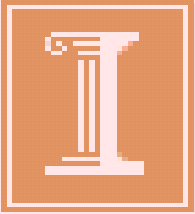
2-Approx Tour = 1->2->3->4->7->8->10->9->6->5->1 (assume)

Modified U-TSP Tour = 1->2->3->10->4->7->8->6->9->5->1



# Overview of the Approach





# Illustration: Pruning Tour

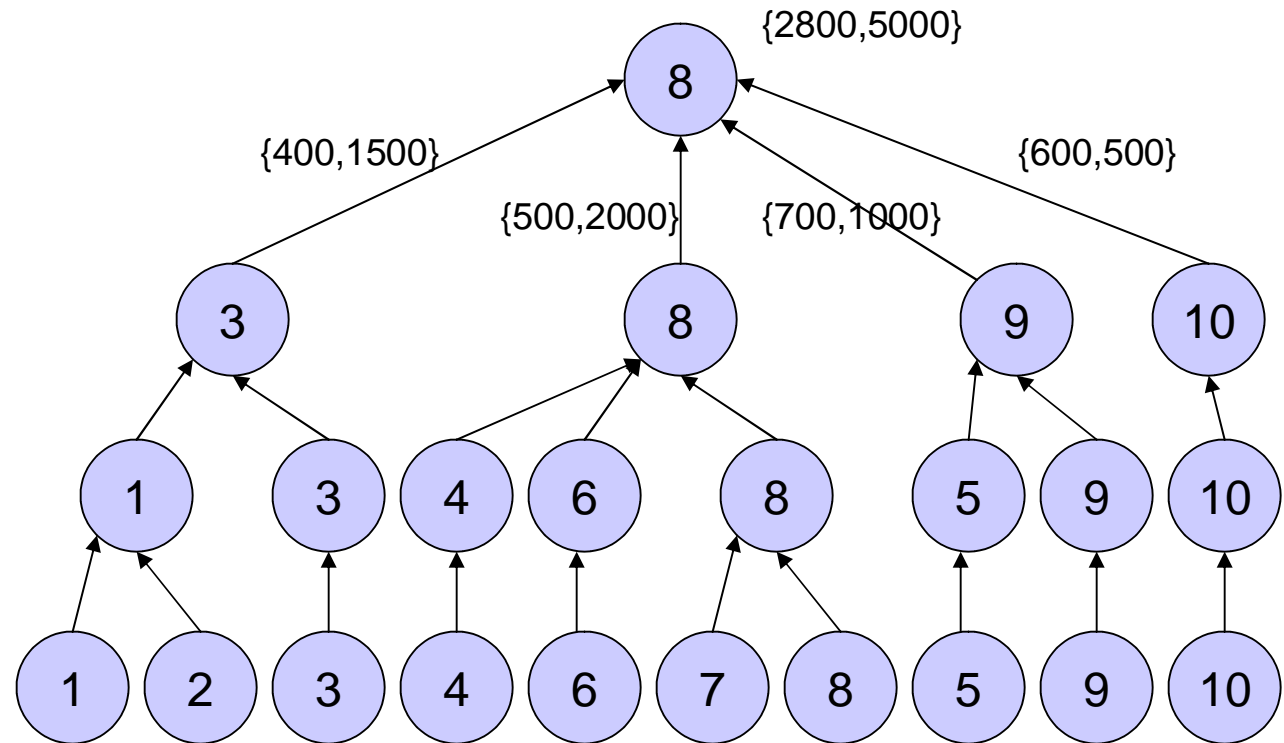


Fuel Budget: 2000

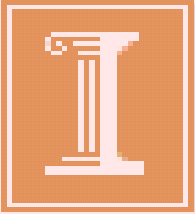
Utility per task: 500

Branch label  $\{x,y\} =$

$\{\text{Marginal Cost, Utility}\}$



Modified U-TSP Tour = 1->2->3->10->4->7->8->6->9->5->1



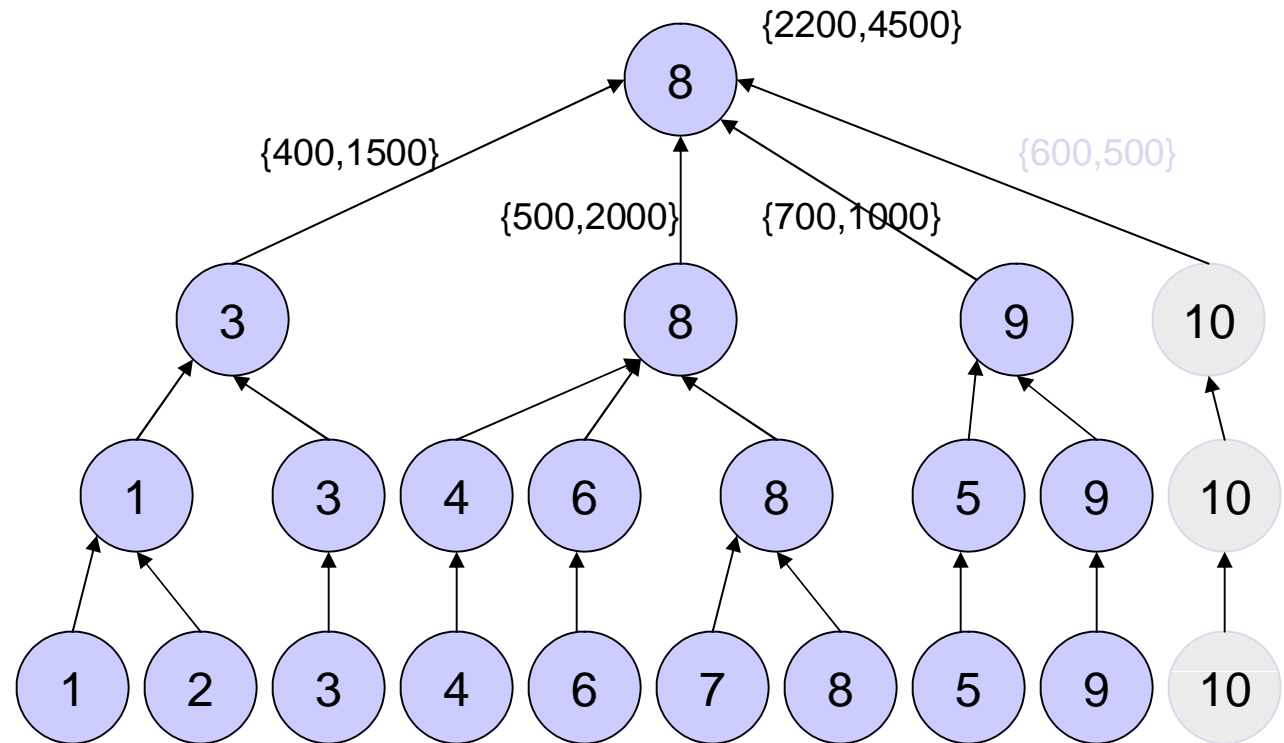
# Illustration: Pruning Tour



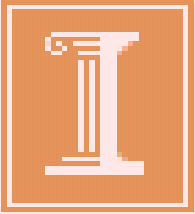
Fuel Budget: 2000

Utility per task: 500

Branch label  $\{x,y\} =$   
 $\{\text{Marginal Cost, Utility}\}$



Modified U-TSP Tour = 1->2->3->10->4->7->8->6->9->5->1



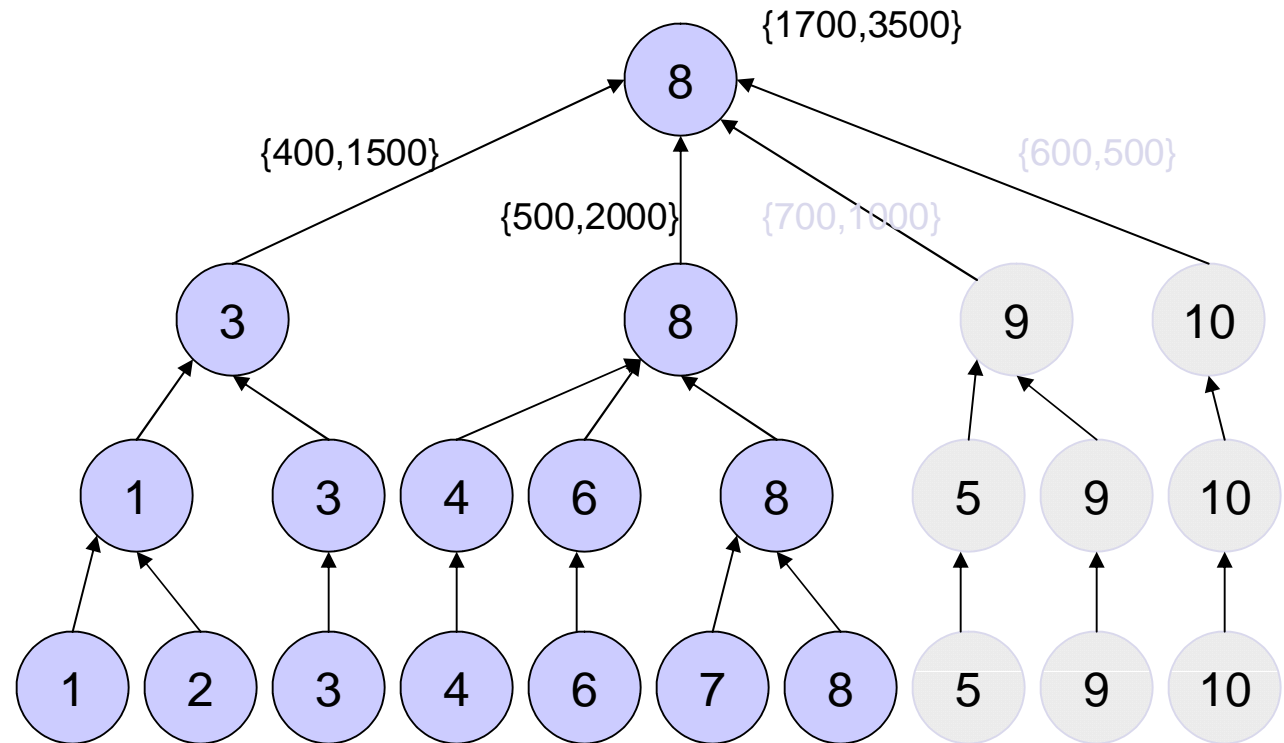
# Illustration: Pruning Tour



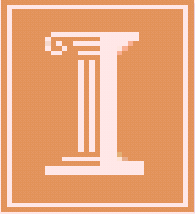
Fuel Budget: 2000

Utility per task: 500

Branch label  $\{x,y\} =$   
 $\{\text{Marginal Cost, Utility}\}$



Modified U-TSP Tour = 1->2->3->10->4->7->8->6->9->5->1



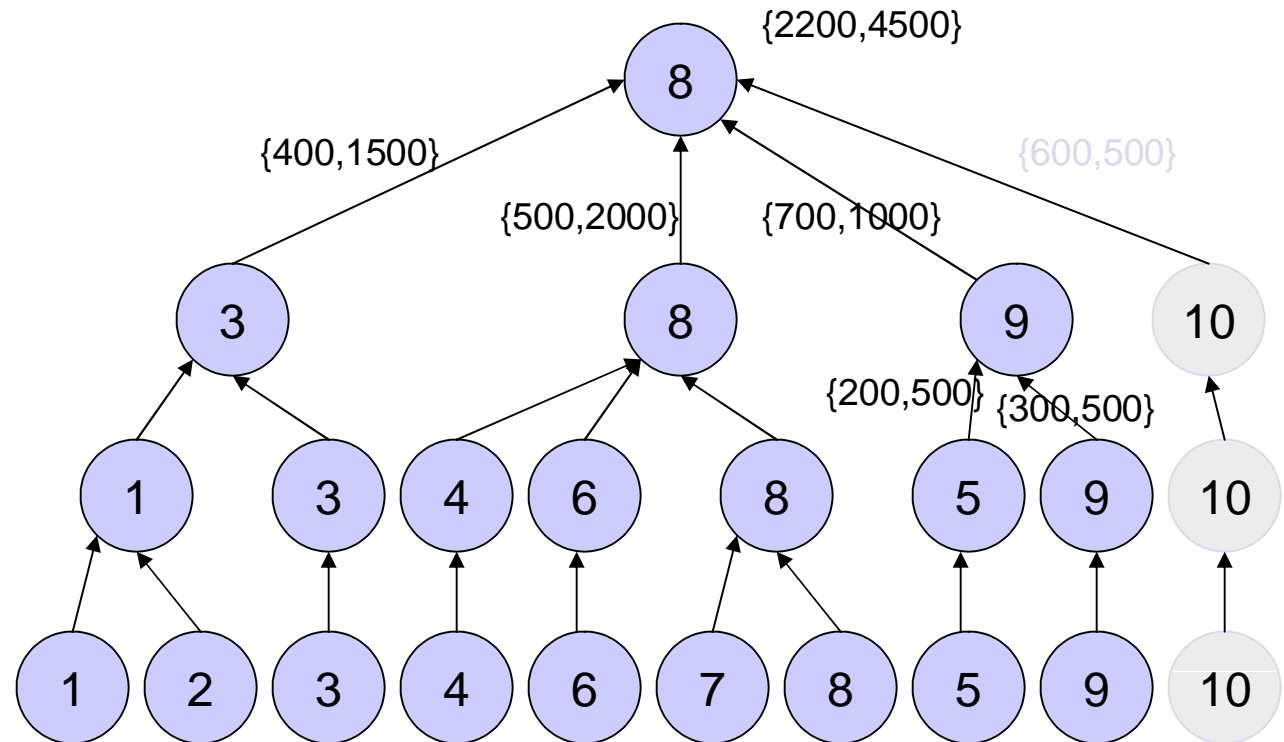
# Illustration: Pruning Tour



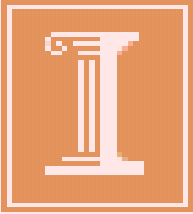
Fuel Budget: 2000

Utility per task: 500

Branch label  $\{x,y\} =$   
 $\{\text{Marginal Cost, Utility}\}$



Modified U-TSP Tour = 1->2->3->10->4->7->8->6->9->5->1



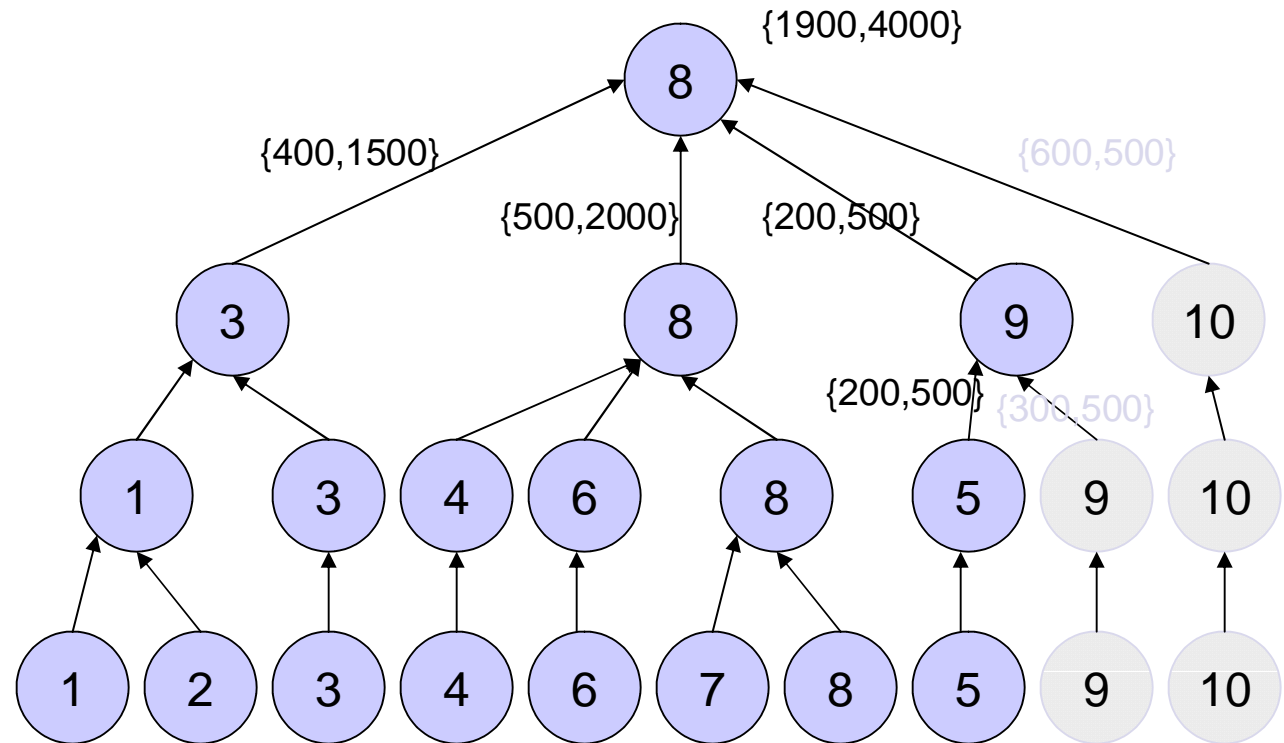
# Illustration: Pruning Tour



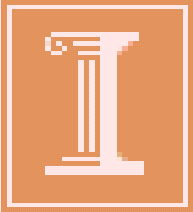
Fuel Budget: 2000

Utility per task: 500

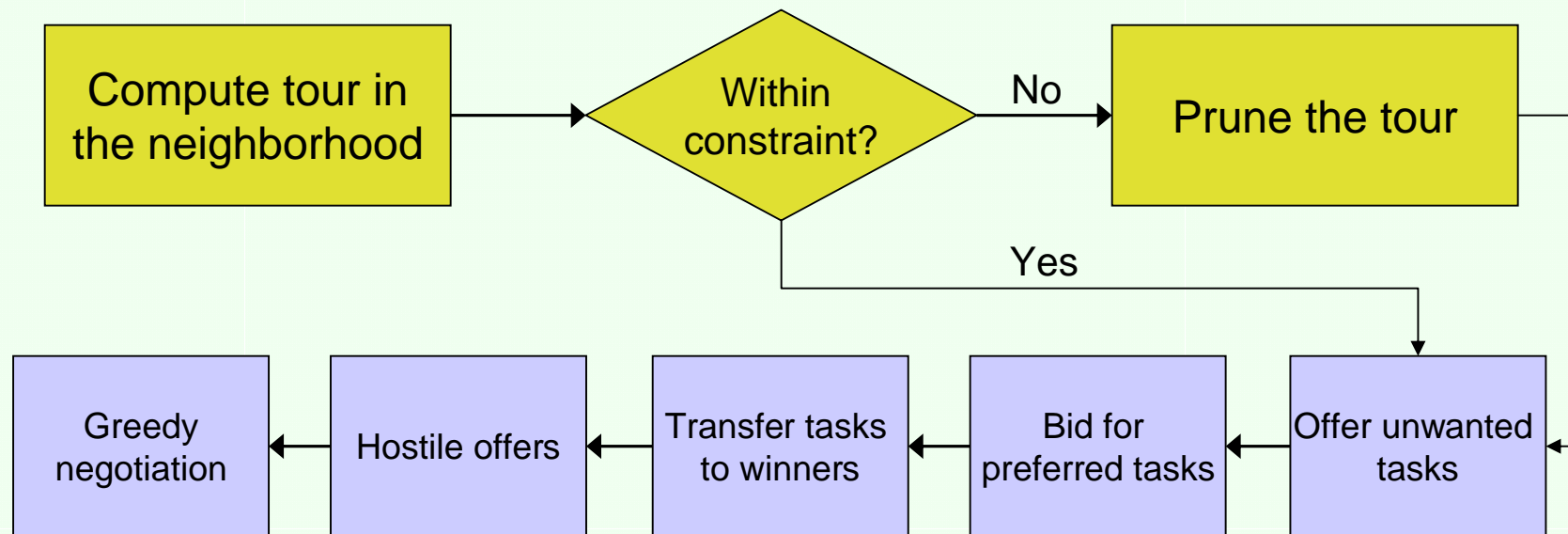
Branch label  $\{x,y\} =$   
 $\{\text{Marginal Cost, Utility}\}$



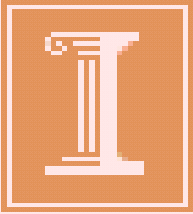
Modified U-TSP Tour = 1->2->3->10->4->7->8->6->9->5->1



# Overview of the Approach



## Negotiation Phase

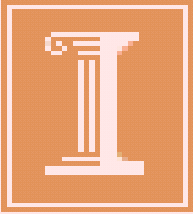


# Negotiation Contracts



Sandholm classification [AAAI '98]

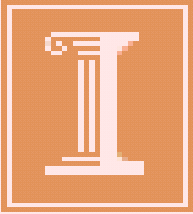
- ✔ *O-Contracts*: Move a task from one agent to another
- *S-Contracts*: Swap two tasks between two agents
- *C-Contracts*: Move a cluster of tasks from one agent to another
- *M-Contracts*: Multi-agent contracts
  
- Side-payment possible with all contracts



# Market Mechanism



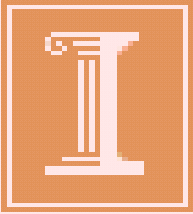
- Each task carries utility,  $u$
- Un-serviced tasks incur penalty,  $p$
- Types of messages:
  - Offer
  - Bid
  - Accept



# Negotiation Strategy



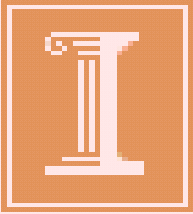
- Offer all tasks in  $(R \setminus T)$  (unwanted tasks) in the market
  - If only one agent wants a task, offer it for free
  - Otherwise, initiate an auction
- Try to grab all tasks in  $(T \setminus R)$  from the market, if available



# Local optimum state



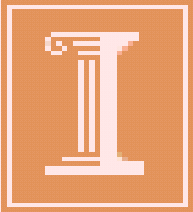
- For each task  $t$ , one of the three holds:
  - Case 1: Only one agent prefers  $t$  and  $t$  is assigned to it.
  - Case 2: No agent prefers  $t$  but it is assigned to one of them.
  - Case 3: Multiple agents prefer  $t$  but one of them owns it
- Case 1 is great! 😊
- Case 2 is bad. Owner incurs penalty  $p$  ☹️
- For case 3, agents send Want To Buy message



# Task States



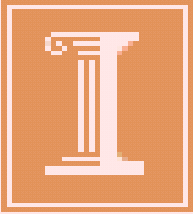
	Owner wants	Owner doesn't want
No agent wants	<b>Impossible</b>	Owner incurs penalty $p$
One agent wants	Owner retains	Transfer for free
Many agents want	Auction	Auction



# Task States



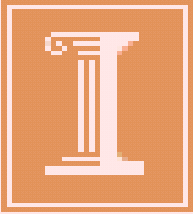
	Owner wants	Owner doesn't want
No agent wants	<b>Impossible</b>	Owner incurs penalty $p$
One agent wants	Owner retains	
Many agents want	Owner retains	



# Simulation Settings



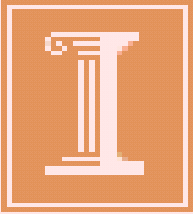
- Custom-designed synchronous simulator
- 1730 problem instances for different combinations of number of agents, number of tasks and fuel constraint
- Compared MALA to:
  - centralized O-contracts based approach
  - TraderBots approach



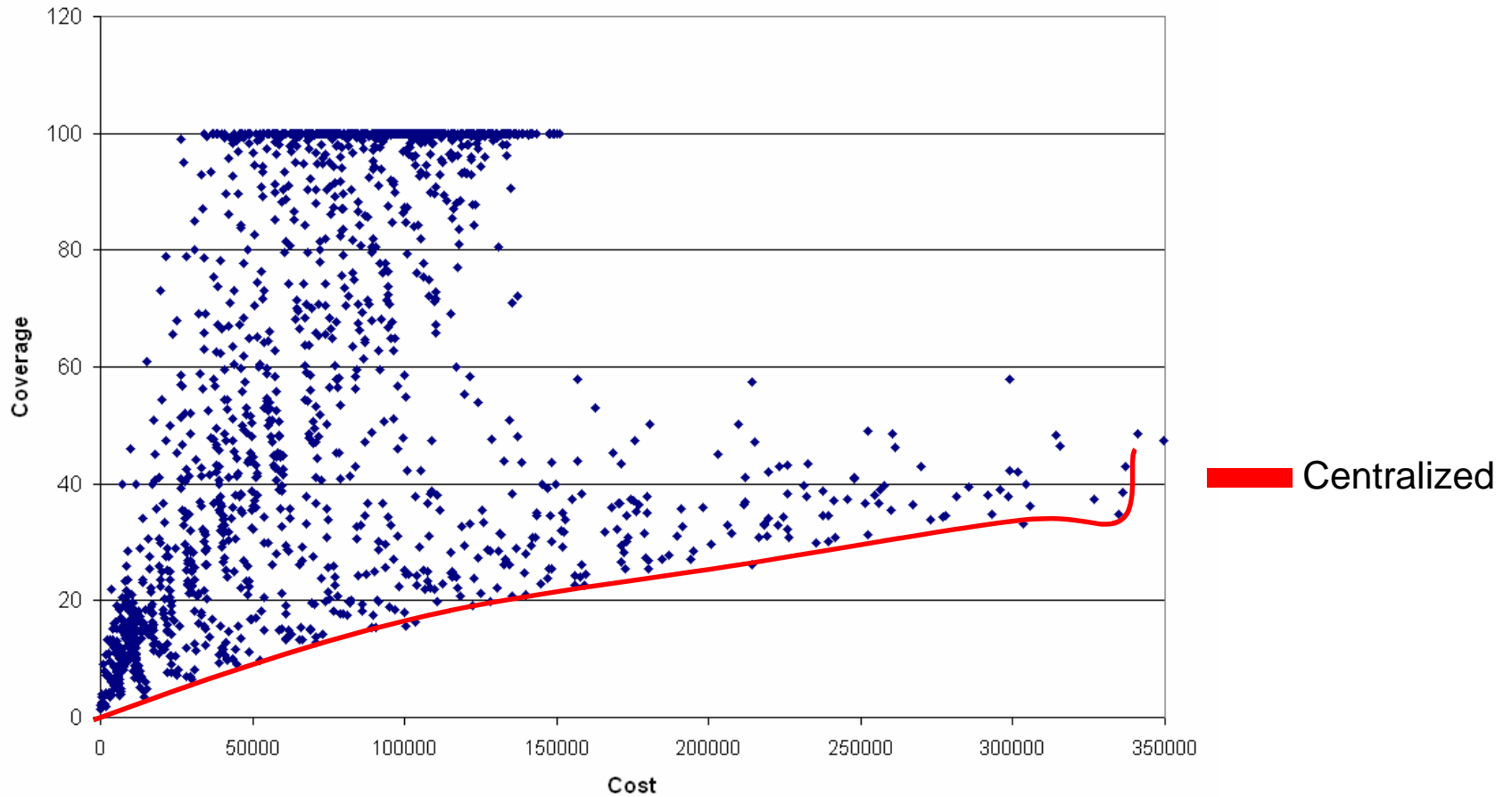
# Evaluation Metrics

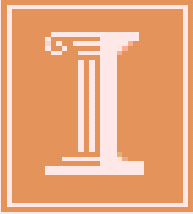


- Coverage
  - Tasks “correctly” allocated to agents as a fraction of total number of tasks
- Cost
  - Total distance covered by all agents
- Number of Messages
- Number of Broadcasts

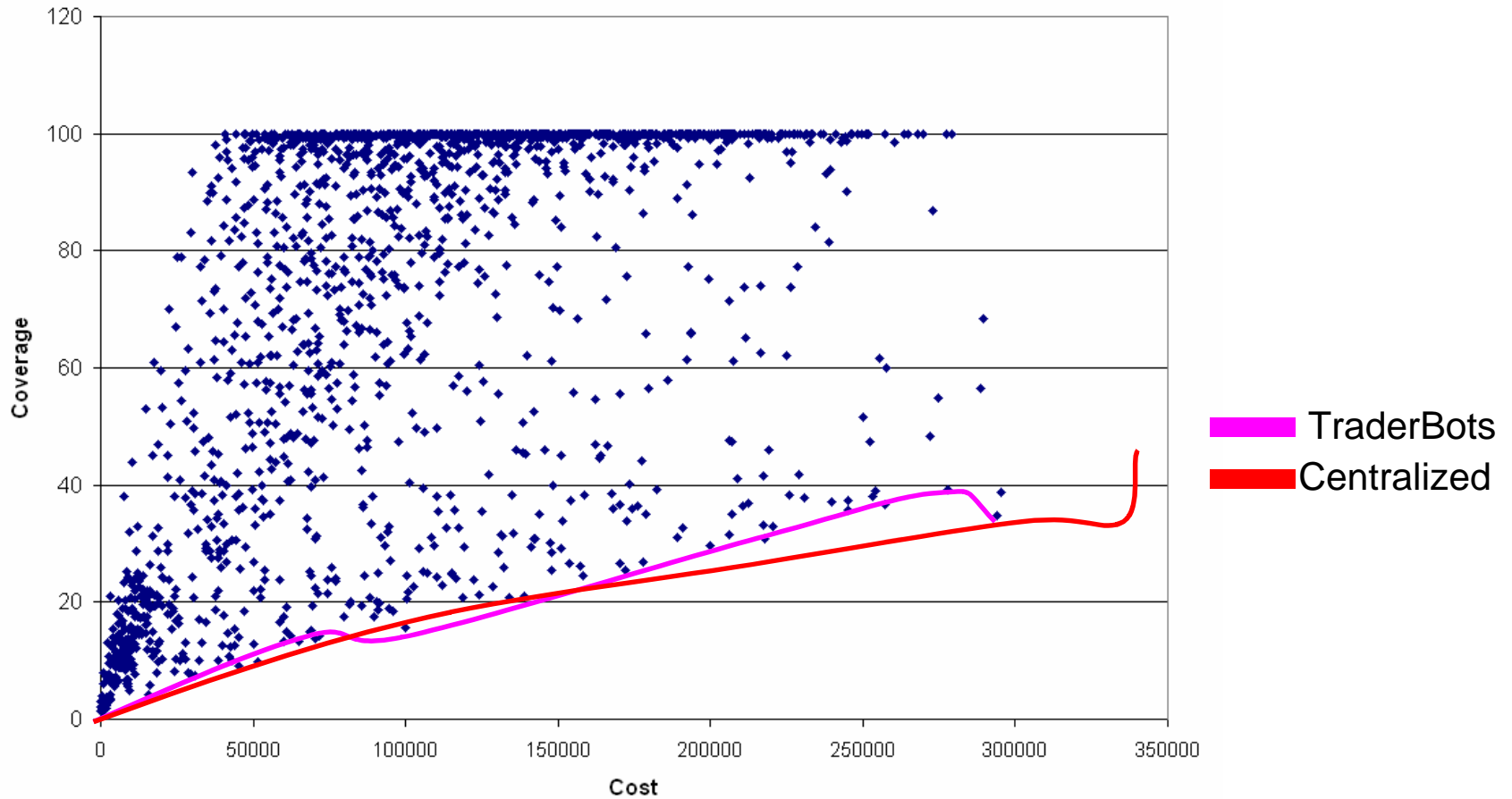


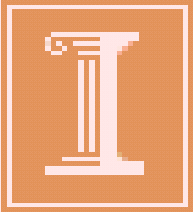
# Coverage vs Cost (Centralized approach)



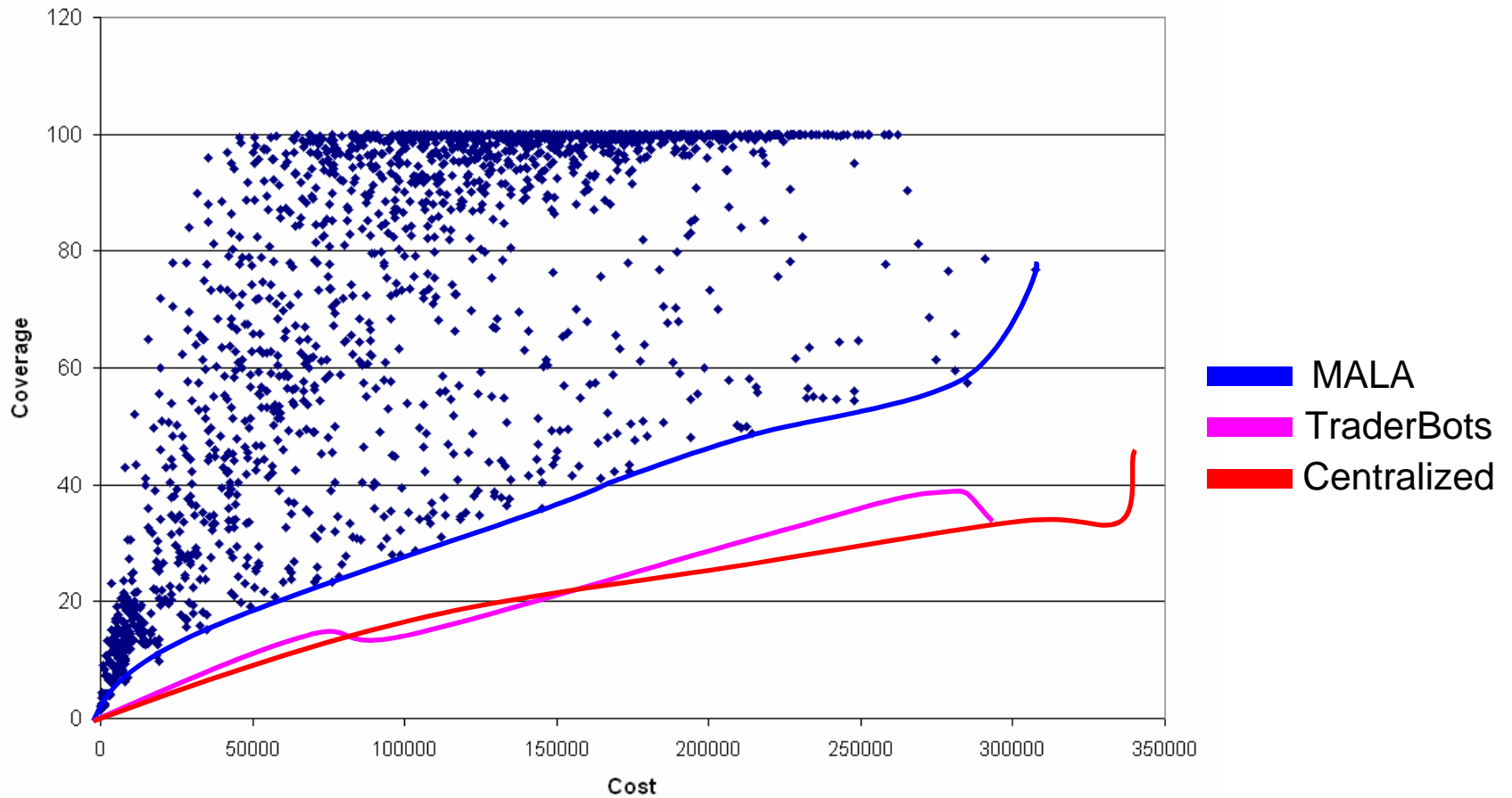


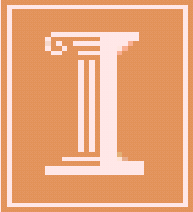
# Coverage vs Cost (TraderBots)





# Coverage vs Cost (MALA)

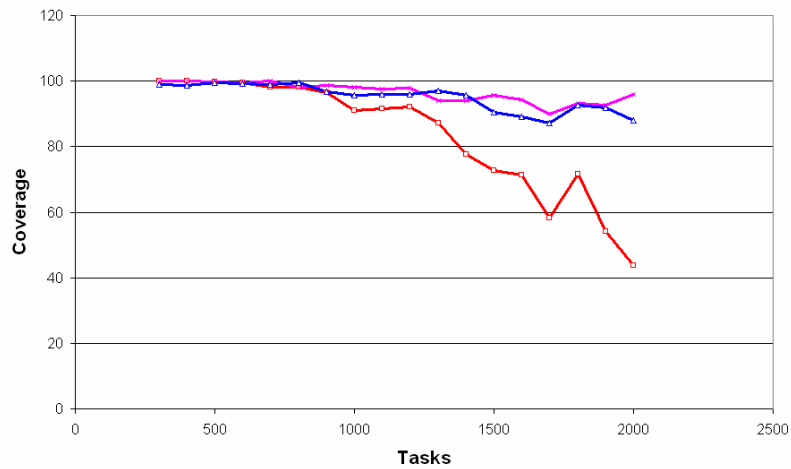




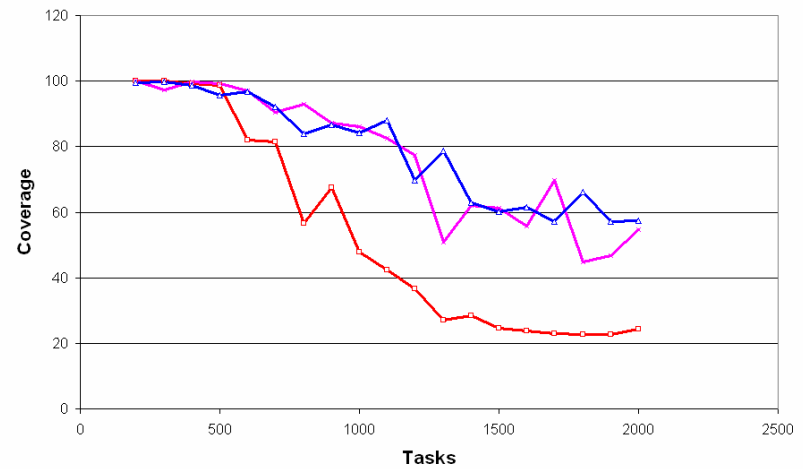
# Coverage vs Number of Tasks



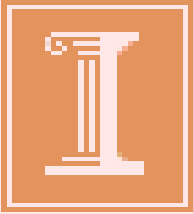
100 Agents - 1100 Fuel Constraint



60 Agents - 1300 Fuel Constraint



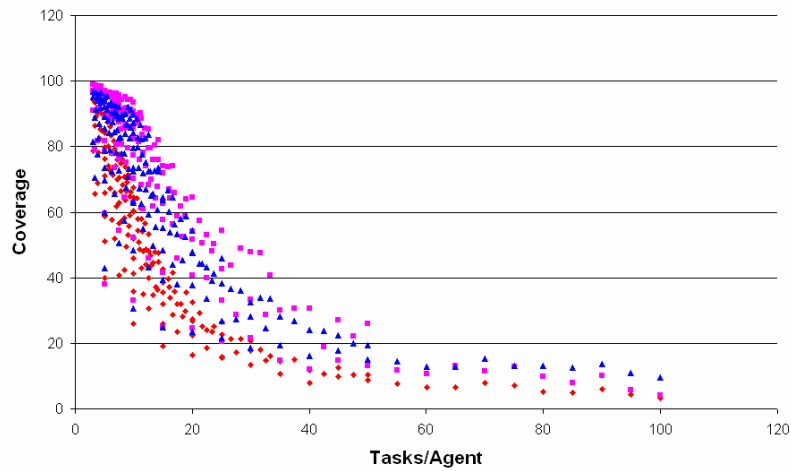
- █ MALA
- █ TraderBots
- █ Centralized



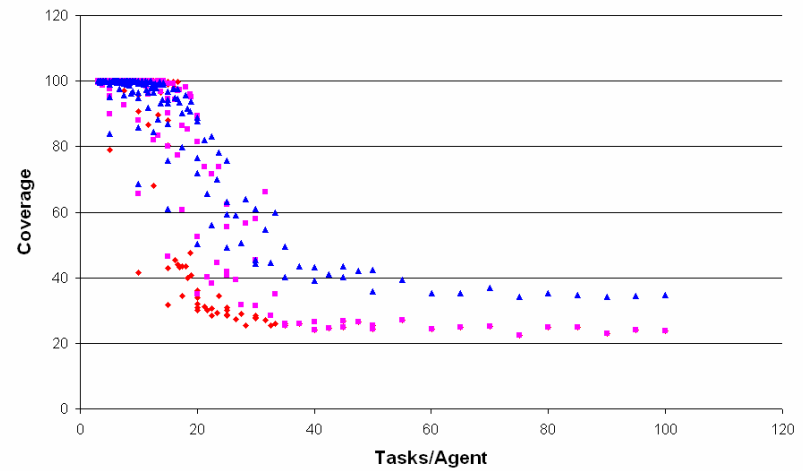
# Coverage vs Density



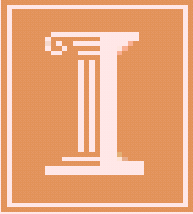
500 Units Fuel Constraint



1500 Units Fuel Constraint



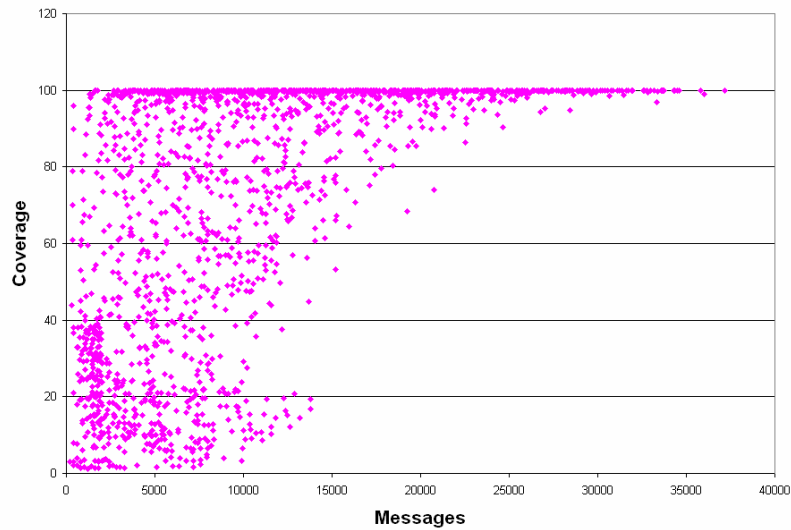
- MALA
- TraderBots
- Centralized



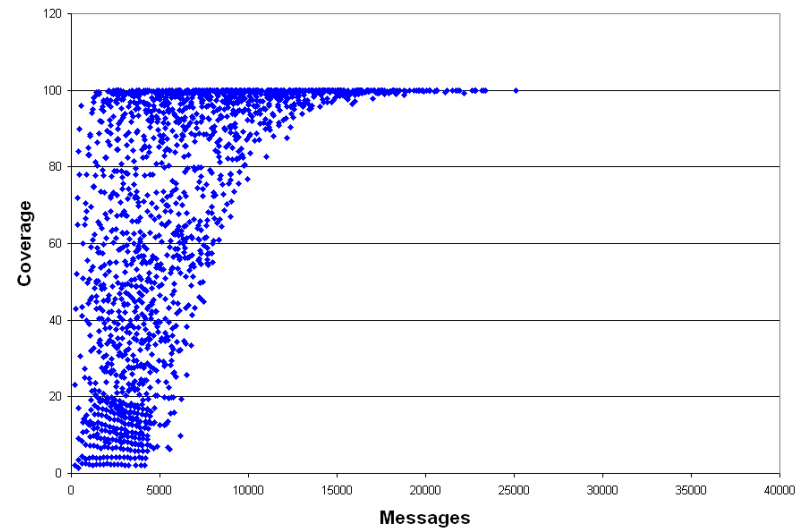
# Coverage vs Messages



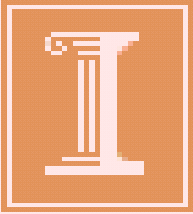
**TraderBots**



**MALA**



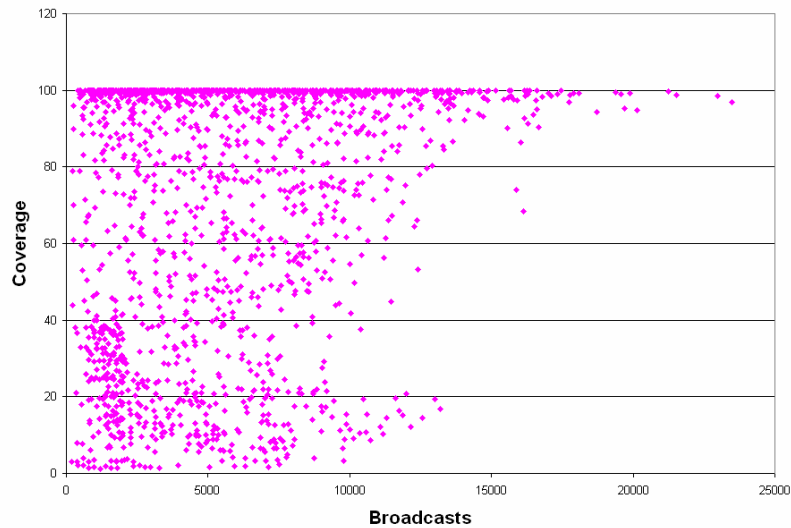
Paired difference:  $t=42$ ,  $p < 0.001$



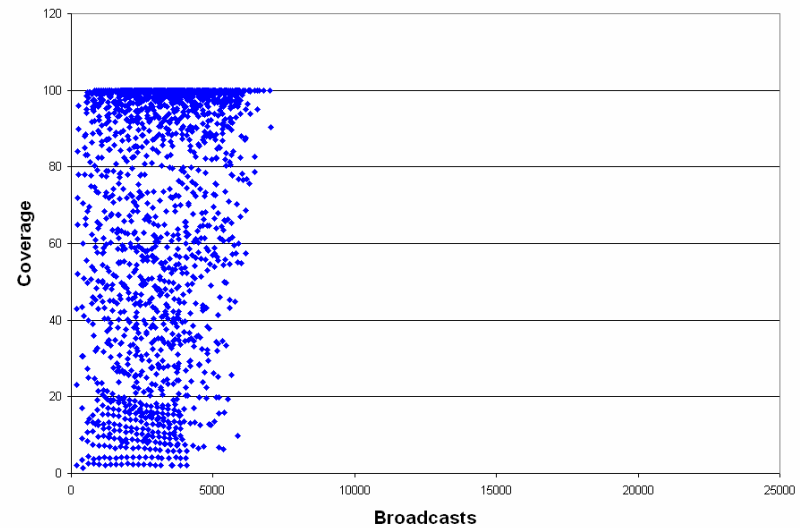
# Coverage vs Broadcasts



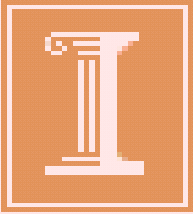
**TraderBots**



**MALA**



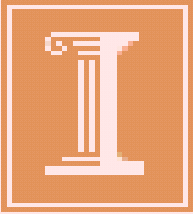
Paired difference:  $t=37, p < 0.001$



# Still Not Optimal?



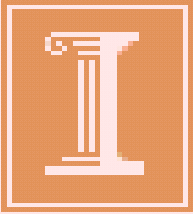
- The look ahead is approximate
- The preference is unknown to other agents and multiple agents may have conflicting preferences



# Future Directions



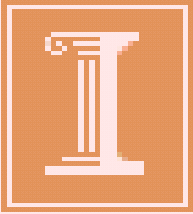
- Relaxing the assumptions
- Reduce communication especially broadcasts
- Designing adaptive negotiation strategy
- Boils down to finding the best method to know other agents' preferences



## Future Directions (contd...)



- Leaders among agents may
  - Enable opportunistic centralization [Dias et al., IROS '02]
  - Provide a better allocation for a small group
  - Resolve constraints and dependencies among tasks
- Agents use learning to improve bidding strategy and reduce communication



Thanks!



Questions?

Feedback?