Exploration, robustness and optimality of network routing algorithms which employ "ant-like" mobile agents

Andre Costa

ARC Centre of Excellence for Mathematics and Statistics of Complex Systems

University of Melbourne

Collaborators: Peter Taylor, Nigel Bean

AUSTRALIAN RESEARCH COUNCIL
Centre of Excellence for Mathematics
and Statistics of Complex Systems

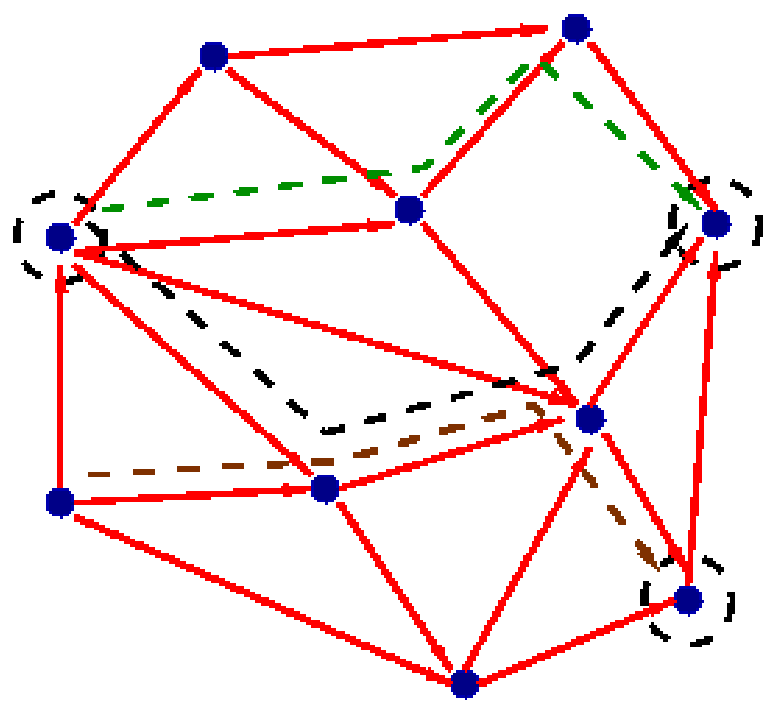
Multi-Agent Systems and Machine Learning Symposium, November 2004

Outline

- Telecommunications networks, routing and optimality
- Biological ants and collective problem solving
- Ant-based routing algorithms
- Analytic modelling
- Ants, Markov decision problems, reinforcement learning and game theory
- Exploration, robustness and optimality (or lack thereof !)

The Network Environment

- packet-based system
- multiple origin-destination node pairs
- routing control $\longrightarrow$ objective: to achieve "optimal routing"



Inspiration from Nature: Biological Ants

Ants

- deposit chemical pheromone as they travel
- tend to follow trails with highest pheromone concentration
- sometimes explore (follow trails with low or zero concentration)

indirect communication between ants mediated via pheremone

The “swarm” has the potential to carry out collective problem-solving (example: double-bridge experiment)

Analogies

How do we create an artificial ant system for a telecommunications network ?

natural	**artificial**
ants	information packets
trails	network links
trail intersections	network nodes
chemical pheromone on trail	probabilistic weights for link choice
pheremone deposition	weight increment
pheremone evaporation	weight decrement

Biological versus artificial ants

- Reinforcement of shortest path in double bridge experiment is a result of ant/pheromone dynamics in the (initial) “transient” period of the experiment, and is highly dependent on initial conditions.
- Can enhance artificial ants with behaviours and properties which overcome such limitations - for example, artificial ants perform differential reinforcement of paths based on delay measurement “after the fact”.

A typical ant-based routing system

The network:

- A set of nodes $\mathcal{S}$,
- connected by the set $\mathcal{A}$ of directed links.
- Denote by $\mathcal{N}_i$ the set of neighbour nodes of node i.

A typical ant-based routing system

The routing algorithm components:

At every node i, the following values are maintained for every destination node d:

1. A set of **trip time estimates** Q_{ijd}, for all neighbouring nodes $j \in \mathcal{N}_i$, where Q_{ijd} constitutes an estimate of the trip time, or delay, associated with travelling from node i to d using the outgoing link (i, j).

2. A set of **ant routing probabilities** ϕ_{ijd}, for all neighbouring nodes $j \in \mathcal{N}_i$, where ϕ_{ijd} is the probability that an *ant* at node i, with destination d, selects the outgoing link (i, j).

3. A set of **data routing probabilities** ψ_{ijd}, for all neighbouring nodes $j \in \mathcal{N}_i$, where ψ_{ijd} is the probability that a *data packet* at node i, with destination d, is routed via the outgoing link (i, j).

A typical ant-based routing system

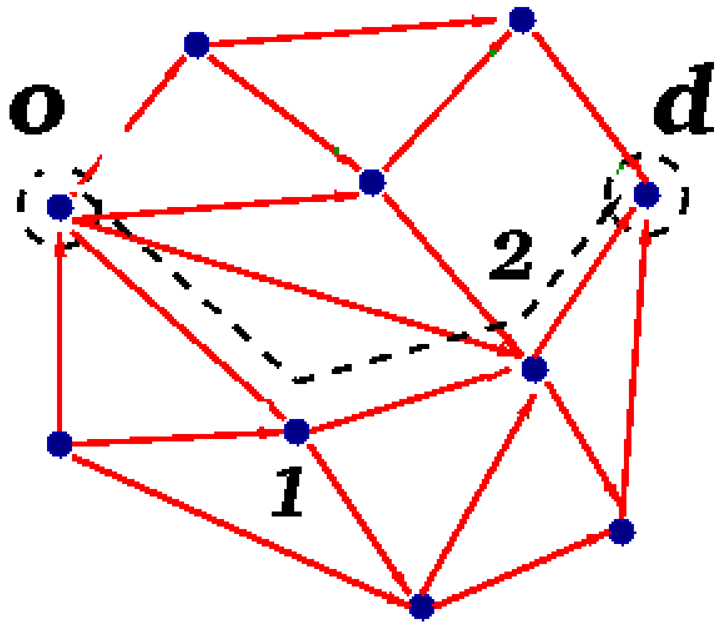
- Ants and data packets share the same network, but are routed according to different sets of routing probabilities.
- Data packets are *passive*, and are routed through the network as usual.
- Ants *actively* measure trip times and feed this information back into the routing tables by updating the trip time estimates.

A typical ant-based routing system

Behaviour and functionality of ants:

- Ants are regularly created at all nodes and sent to all possible destination nodes.
- An ant with origin node o and destination node d is routed according to the ant routing probabilities, until d is reached.
- On its forward journey, the ant measures and stores trip time information.
- Once d is reached, it re-traces its path back to o.
- The ant updates the appropriate trip time estimates maintained at each of the nodes on its path.

Example: sample path



Forward path : $\{o, 1, 2, d\}$

Backward path : update $Q_{2,d,d}$, $Q_{1,2,d}$, $Q_{o,1,d}$

Updates are functions of forward trip time measurements: $q_{2,d,d}$, $q_{1,2,d}$, $q_{o,1,d}$

A typical ant-based routing system

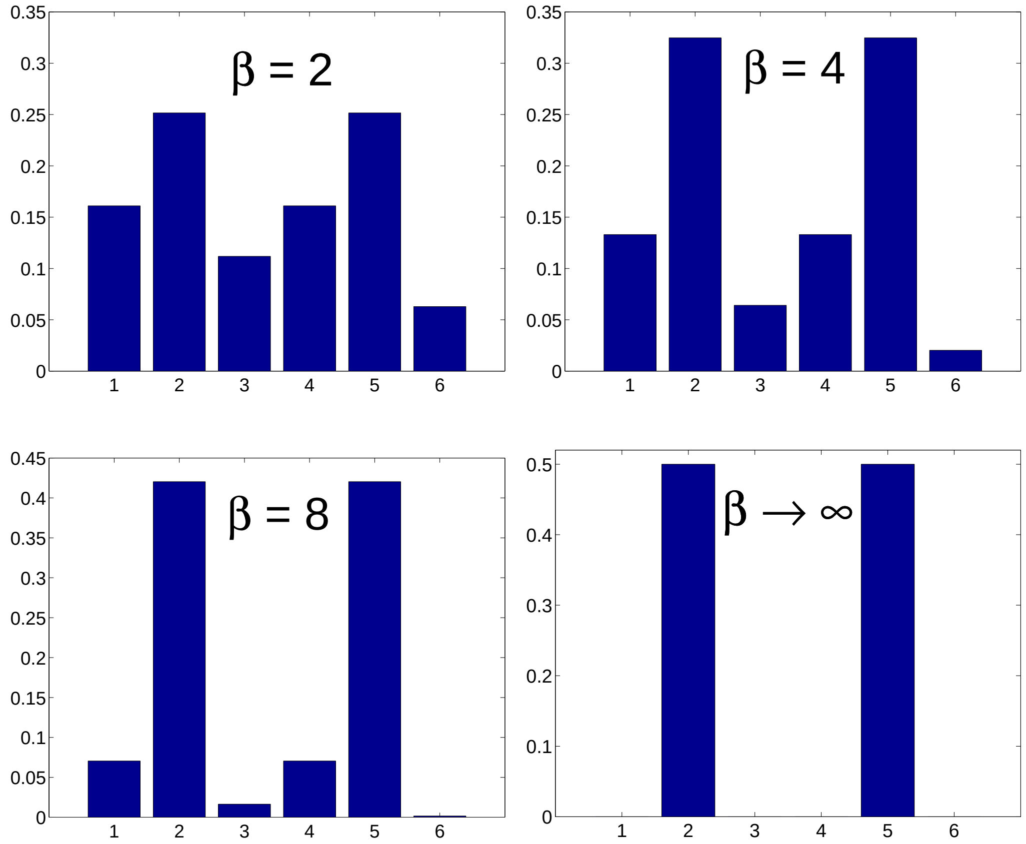
The ant and data routing probabilities are updated as follows:

$$\begin{aligned} \phi_{ijd} &:= Z_i \left(\frac{1}{Q_{ijd}} \right)^{\beta} \\ \psi_{ijd} &:= \hat{Z}_i \left(\frac{1}{Q_{ijd}} \right)^{\sigma}, \end{aligned}$$

where Z_i and $\hat{Z}_i$ are normalising constants, and

- $\beta > 0$ is an “exploration” parameter
- $\sigma > 0$ is a “load-balancing” parameter
- Typically, choose $\beta < \sigma$

$j \in \mathcal{N}_i$	1	<u>2</u>	3	4	<u>5</u>	6
Q_{ij}	*2.5*	<u>*2*</u>	*3*	*2.5*	<u>*2*</u>	*4*



Exploration

In the network context, exploration can be characterized as the “probing” of a number of possible routes, in order to obtain information about them (even if they are not currently in use).

Indeed, all “on-line” learning algorithms require an exploration mechanism. Otherwise,

- cannot adapt to changes in network conditions
- convergence to an optimal solution can be heavily reliant on initial conditions

$\boxed{\text{Exploration} \longrightarrow \text{reduce possibility of convergence to sub-optimal solutions}}$

What are the consequences of using randomized routing “polices” as a way of achieving exploration ?

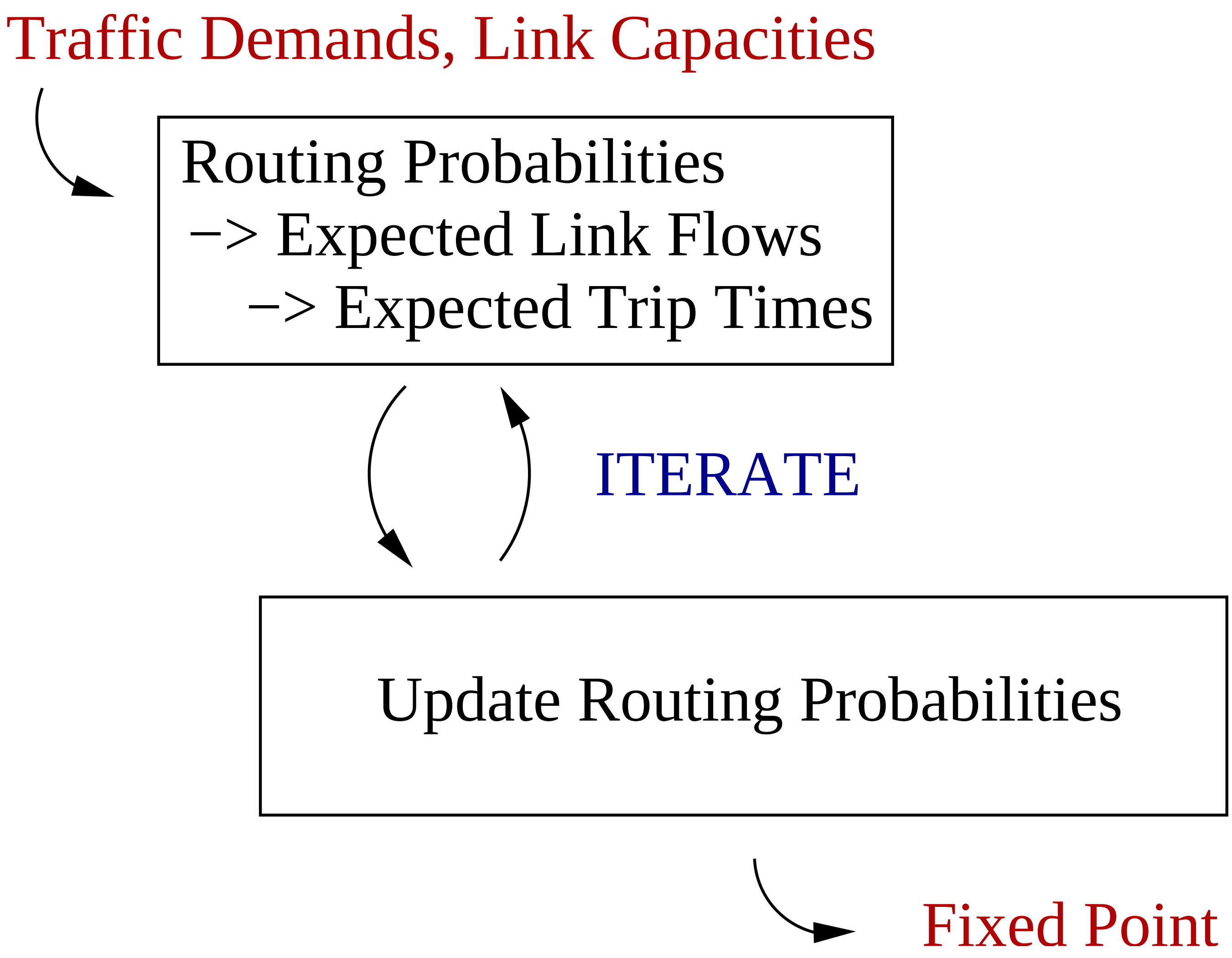
Literature: simulation-based studies

- A number of implementations and variations of ant-based routing algorithms have been proposed (1995 - present).
- All studies have focused on simulation experiments.
- Some of these studies indicate that ant-based routing algorithms have desirable transient adaptive properties, in response to
  - sudden or gradual changes in traffic demands
  - isolated node or link failures

Our aim: to gain develop analytic models and theoretical approaches to the study of ant-based routing algorithms.

Focus: equilibrium (steady-state) behaviour.

Analytic Model



Analysis

We gain insight into the ant-based algorithm by considering two cases

1. Absence of data traffic, queueing delays negligible, ants experience only fixed transmission delays
   - optimality $\longrightarrow$ a standard shortest path problem
   - can be analysed as a Markov decision problem
   - highlights fundamental aspects (and limitations) of current ant-based routing systems
2. Presence of data traffic, queueing delays dominate the dynamics of the system
   - optimality $\longrightarrow$ system or user optima
   - can be analysed using game theory/constrained nonlinear optimisation
   - yields insight into the load-balancing ability of ant-based routing algorithms

Analysis - fixed link delays

No data traffic demands, ants experience only fixed link transmission delays

$$\text{fixed delay on link } (i, j) = r_{ij}.$$

As a point of reference, consider the "optimal Q-values"

$$Q^*_{ij} = r_{ij} + J^*_j$$

where J^*_j is the (optimal) shortest path cost associated with reaching the destination node d from node j.

A shortest path from any node to d is constructed by selecting at each node i the outgoing link which satisfies

$$\arg \min_{(i,l):l \in \mathcal{N}_i} \{Q^*_{il}\}$$

until the destination is reached.

Analysis - fixed link delays

A fixed point of the analytic model has the property that

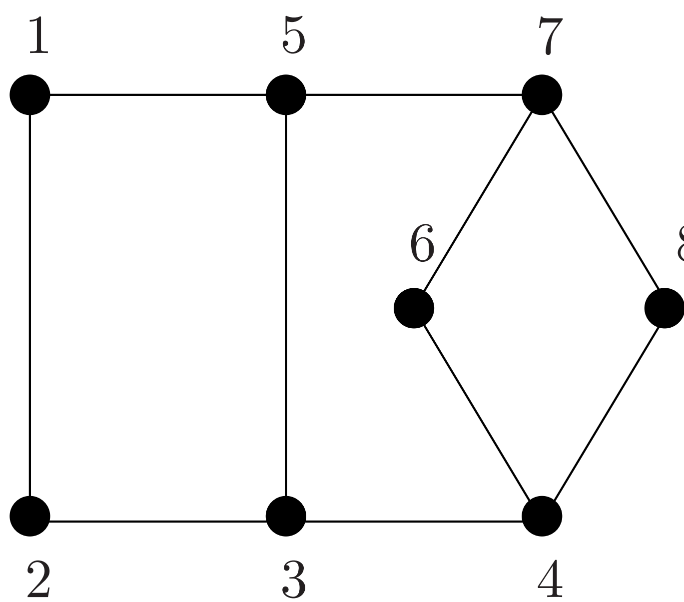
$$Q_{ij} \geq Q^*_{ij},$$

with strict inequality for at least one link.

This is actually not surprising, but does it matter ?

Analysis - fixed link costs

Example: consider origin node = 1, destination node = 8



Link delays $r_{ij} = 1$ for all links (i, j)

Analysis - fixed link delays

A fixed point of the analytic model has the property that

$$Q_{ij} \geq Q^*_{ij},$$

with strict inequality for at least one link.

This is not surprising, but does it matter ?

Yes !

It is not always possible to construct a shortest path from all nodes to the destination by simply selecting outgoing links which satisfy

$$\arg \min_{(i,l):l \in \mathcal{N}_i} \{Q_{ij}\}.$$

Analysis - fixed link delays

Inherent sub-optimality in the ant-based system, due to the fact that

Ants employ the same policy for exploration as they do for decision-making

In the language of control theory, the dual tasks of system identification and control are coupled, and that there is a tradeoff between these tasks.

(Later - this tradeoff can be eliminated by better design !)

Ants and Markov decision problems

Gain insights into the effect of exploration by considering:

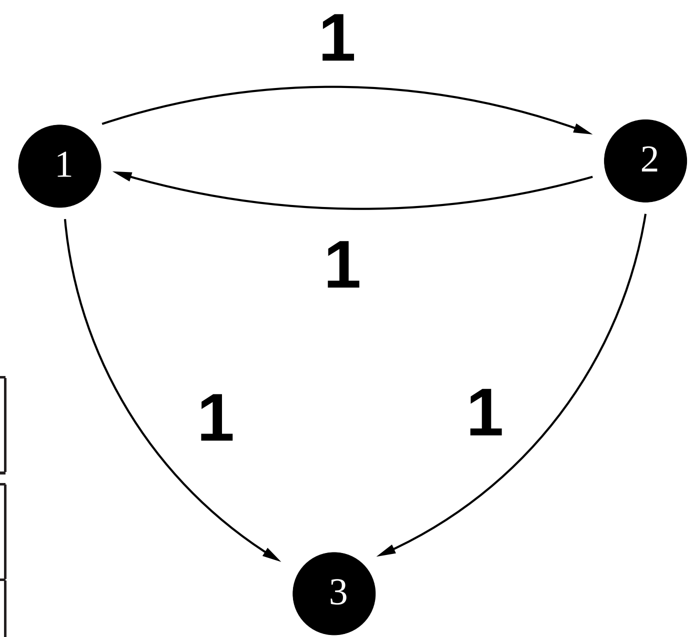
- Underlying problem is a deterministic shortest path MDP
- Ant-based algorithm is an “online” learning algorithm for solving the MDP (c.f. reinforcement learning)

It can be shown that:

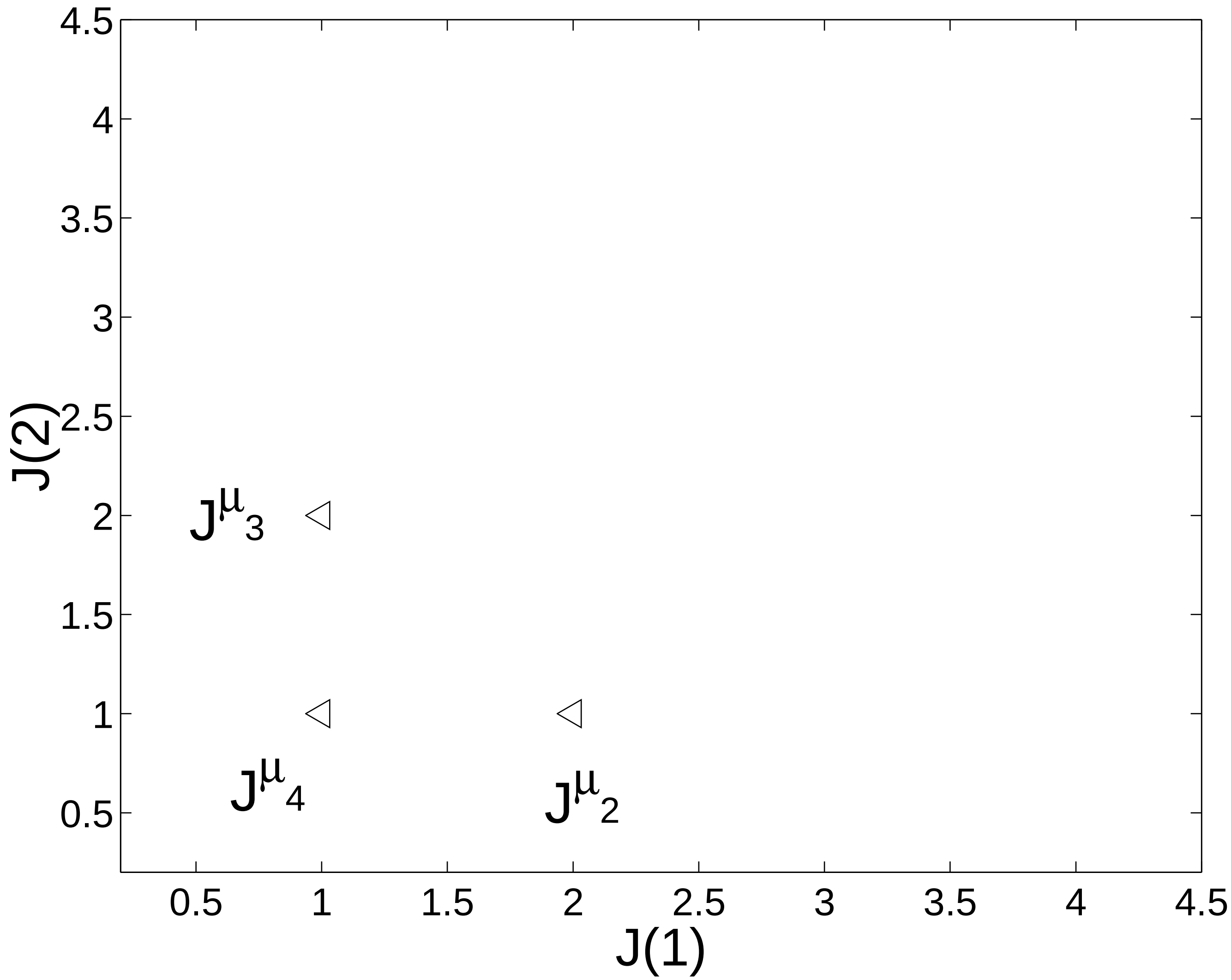
1. Exploration via policy randomization effectively *modifies* the MDP being solved
2. The modified MDP may have a different optimal policy to the original
3. This can lead to “exploration-induced error”

A deterministic shortest path (DSP) problem

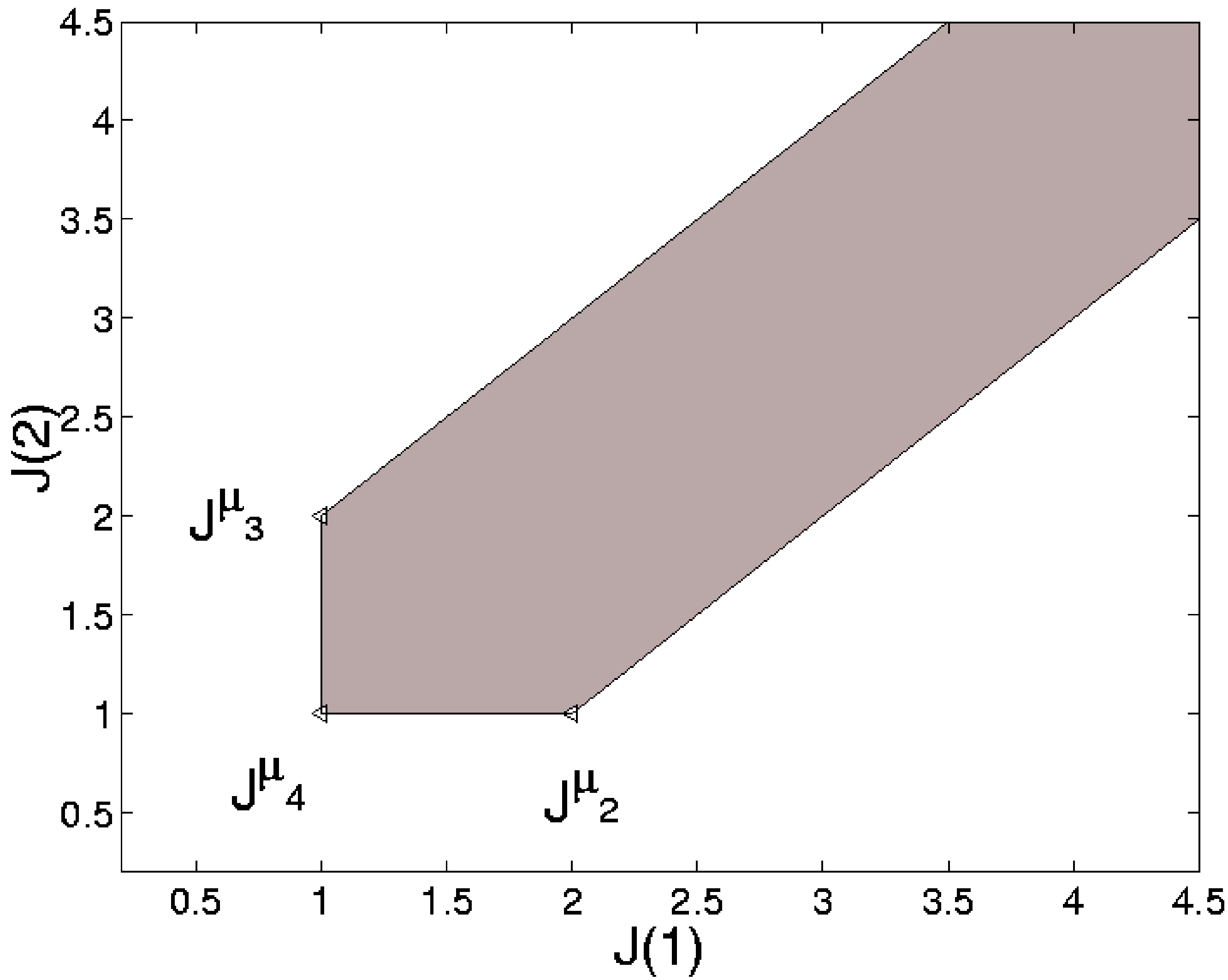
k	μ_k	J^{μ_k}
1	(2, 1)	(∞, ∞)
2	(2, 3)	(2, 1)
3	(3, 1)	(1, 2)
4	(3, 3)	(1, 1)



Points: costs of policies μ_2, μ_3, μ_4 applied to DSP problem



Region: costs of arbitrary randomized polices applied to DSP problem



Recall that in the ant-based routing algorithm, we had randomized ant routing policies given by

$$\phi_{ij} \propto \left(\frac{1}{Q_{ij}}\right)^{\beta}.$$

In the case of finite fixed link delays and $\beta < \infty$, we therefore have

$$\phi_{ij} > 0$$

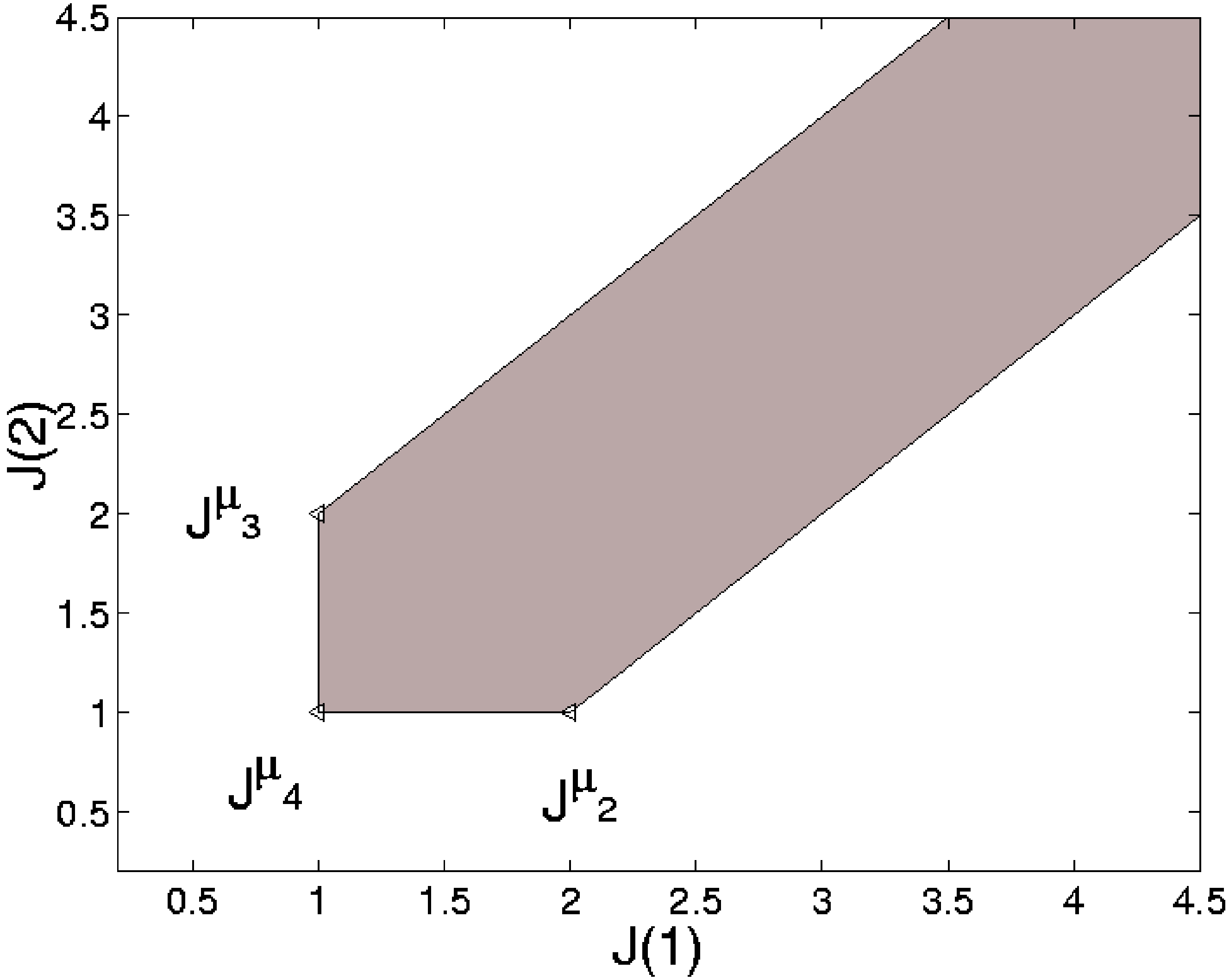
for all links (i, j).

Suppose in particular that

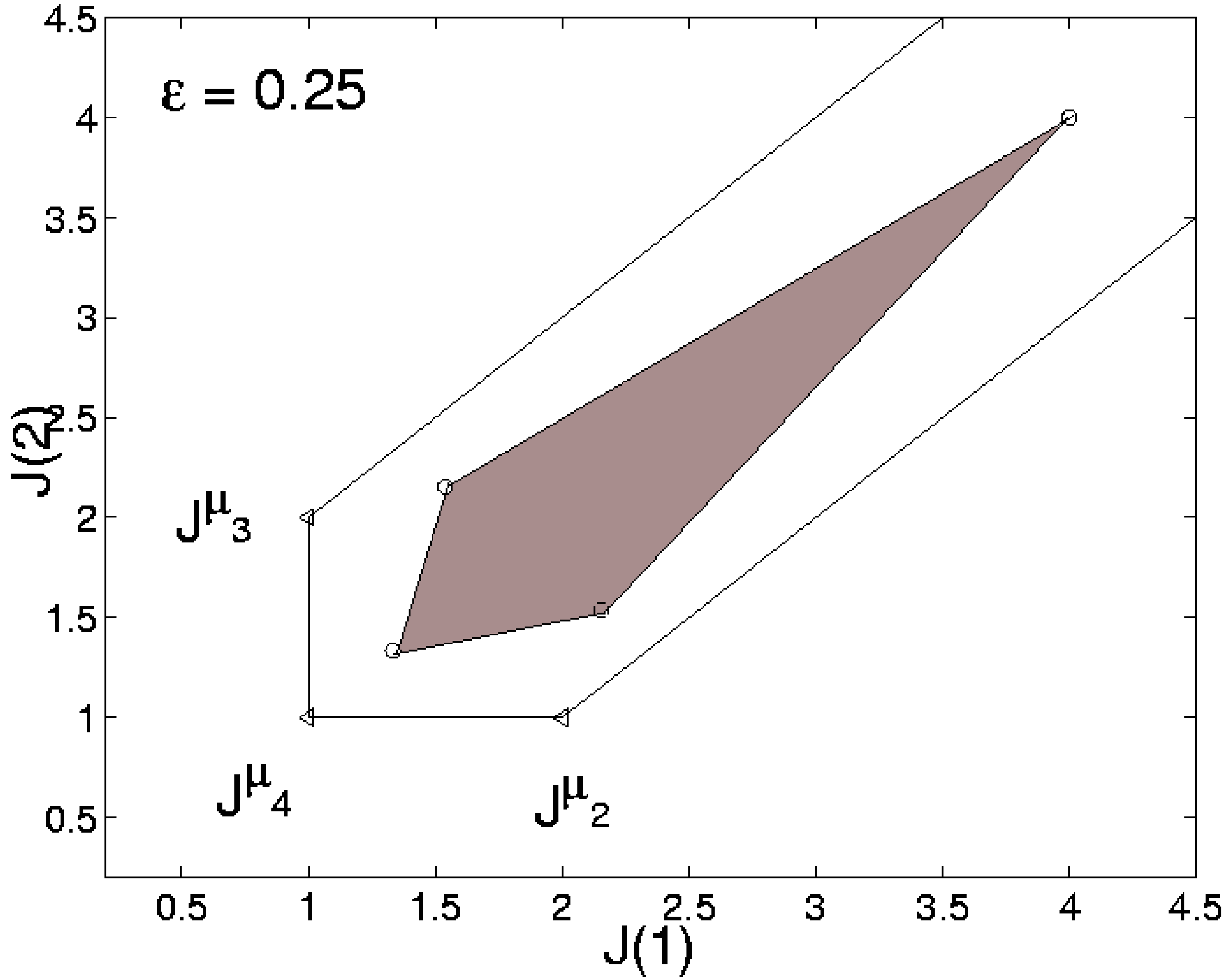
$$\phi_{ij} \geq \epsilon,$$

where $\epsilon > 0$.

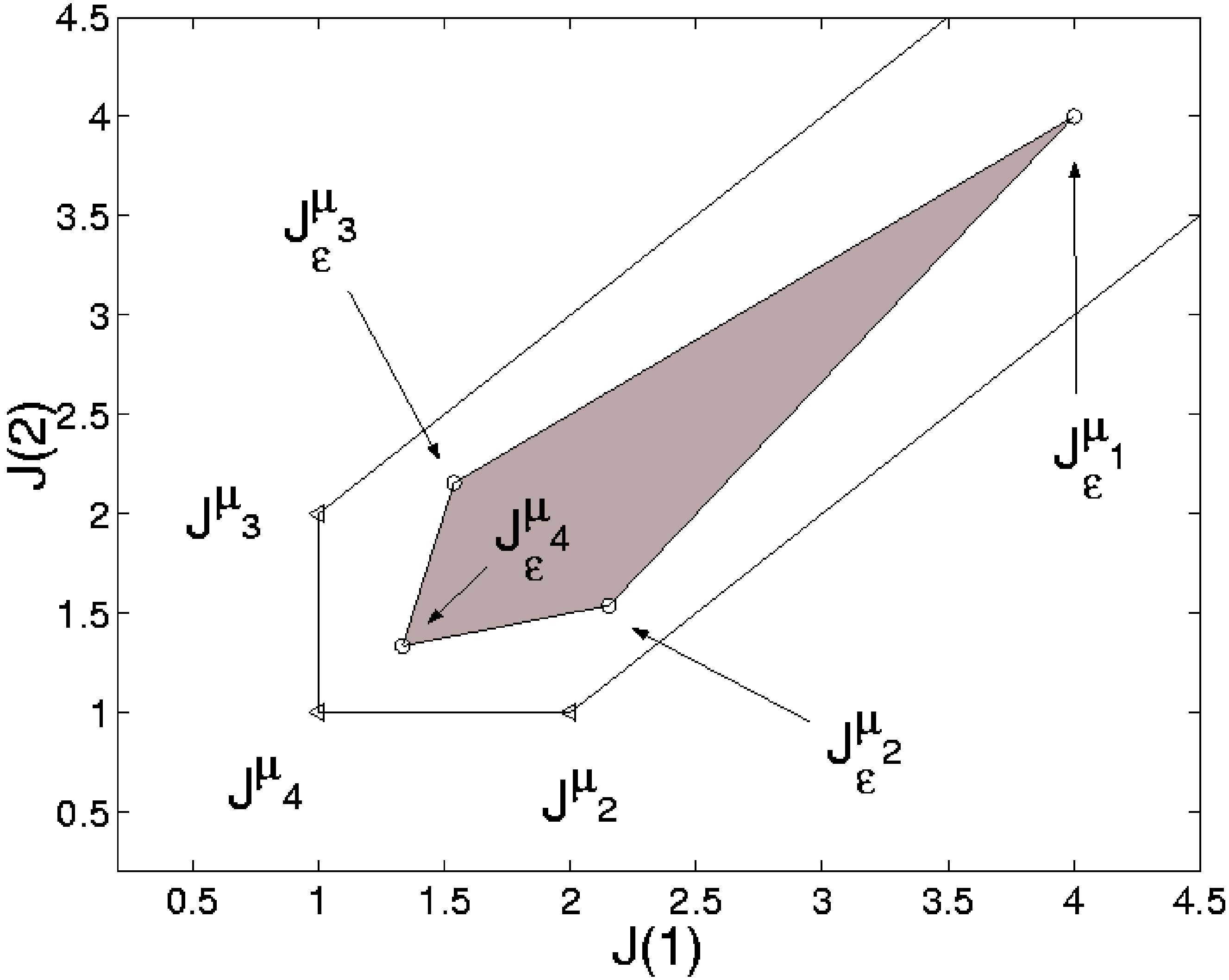
Region: costs of arbitrary randomized polices applied to DSP problem



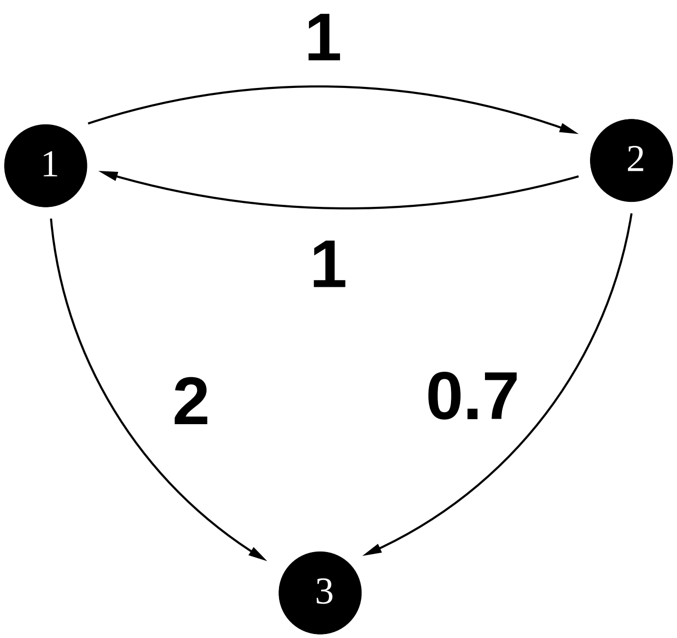
Region: costs of randomized policies subject to $\phi_{ij} \geq \epsilon$



Region: costs of arbitrary randomized polices applied to "ϵ-modified" MDP

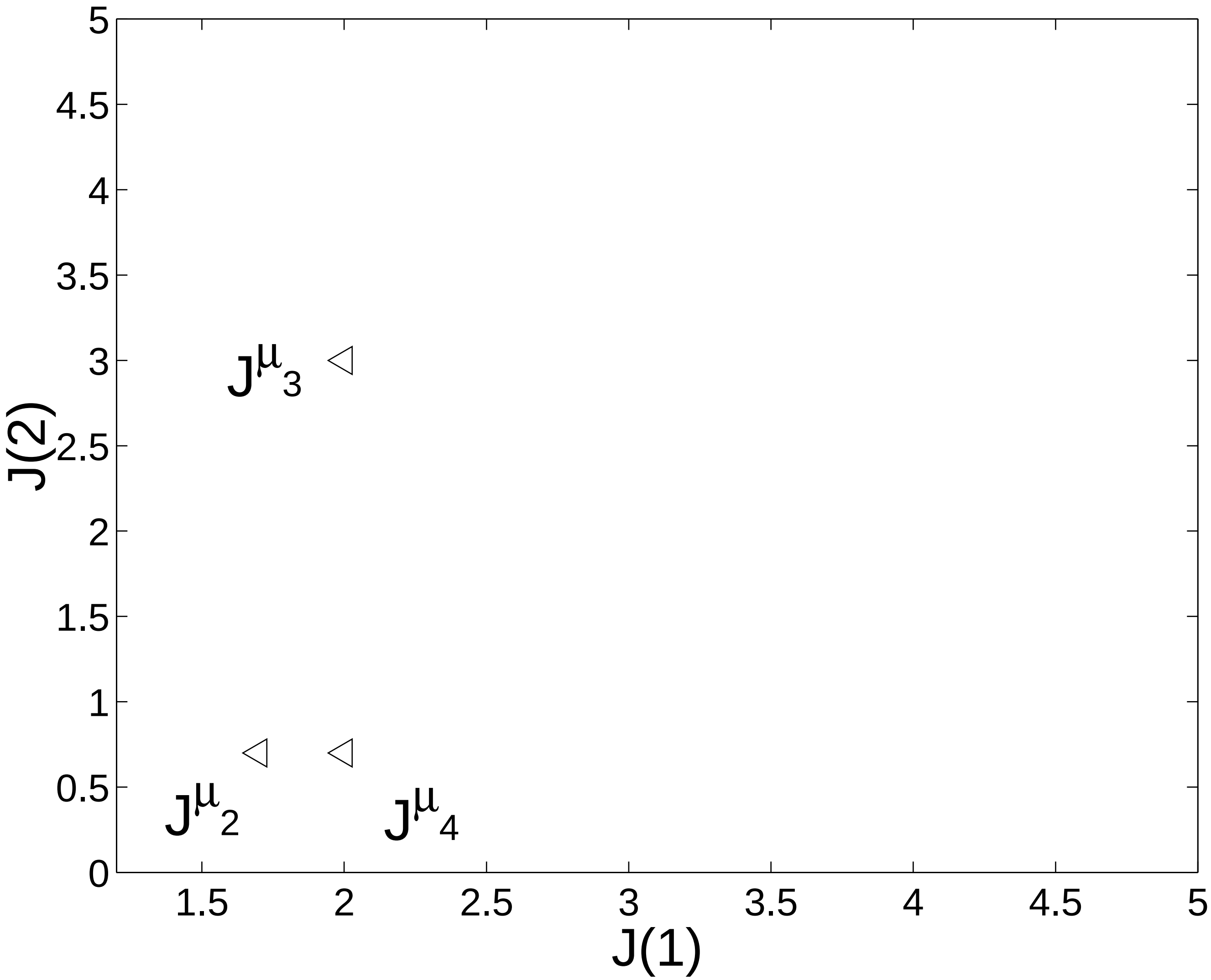


Example 2: Another deterministic shortest path problem

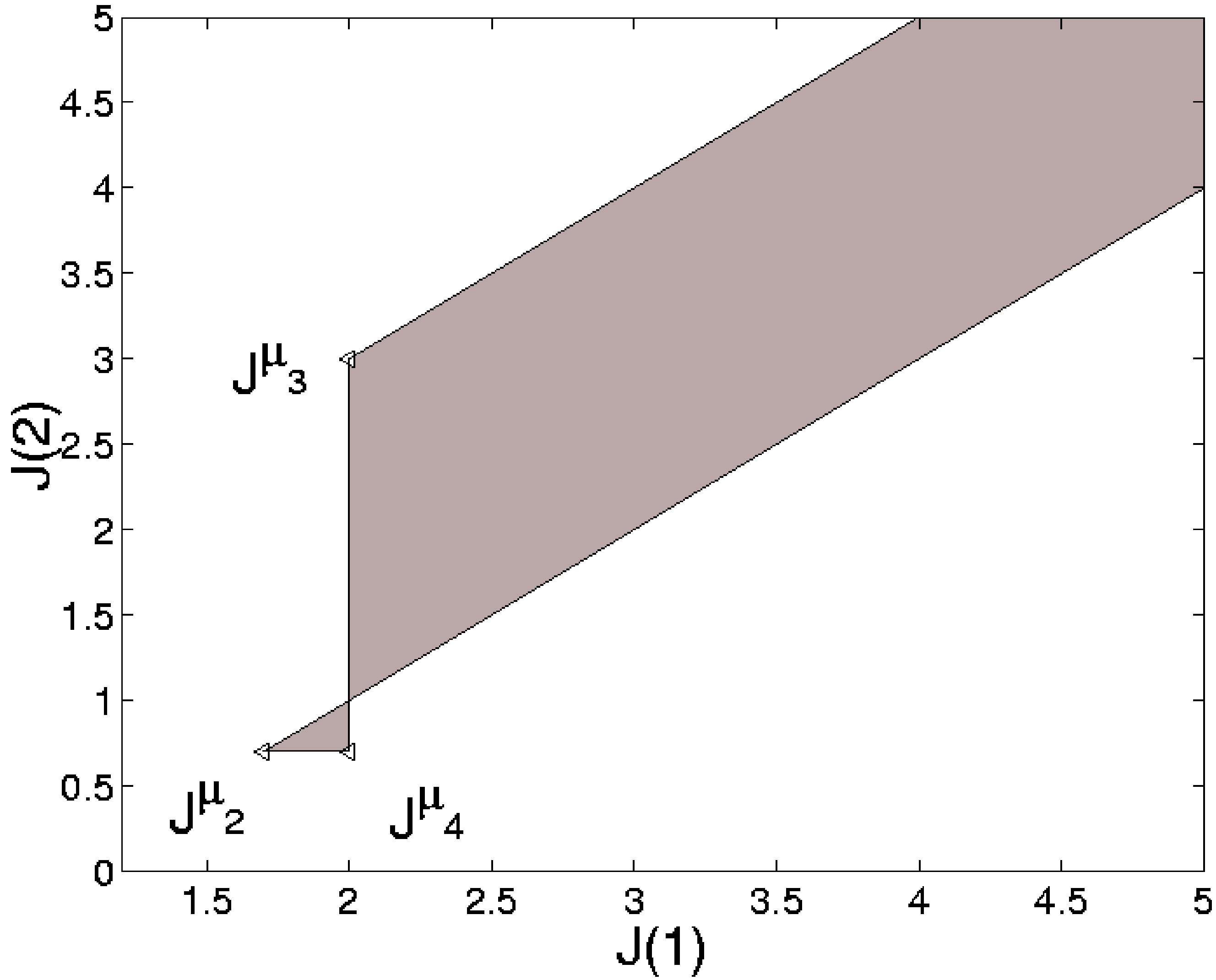


k	μ_k	J^{μ_k}
1	(2, 1)	(∞, ∞)
2	(2, 3)	(1.7, 0.7)
3	(3, 1)	(2, 3)
4	(3, 3)	(2, 0.7)

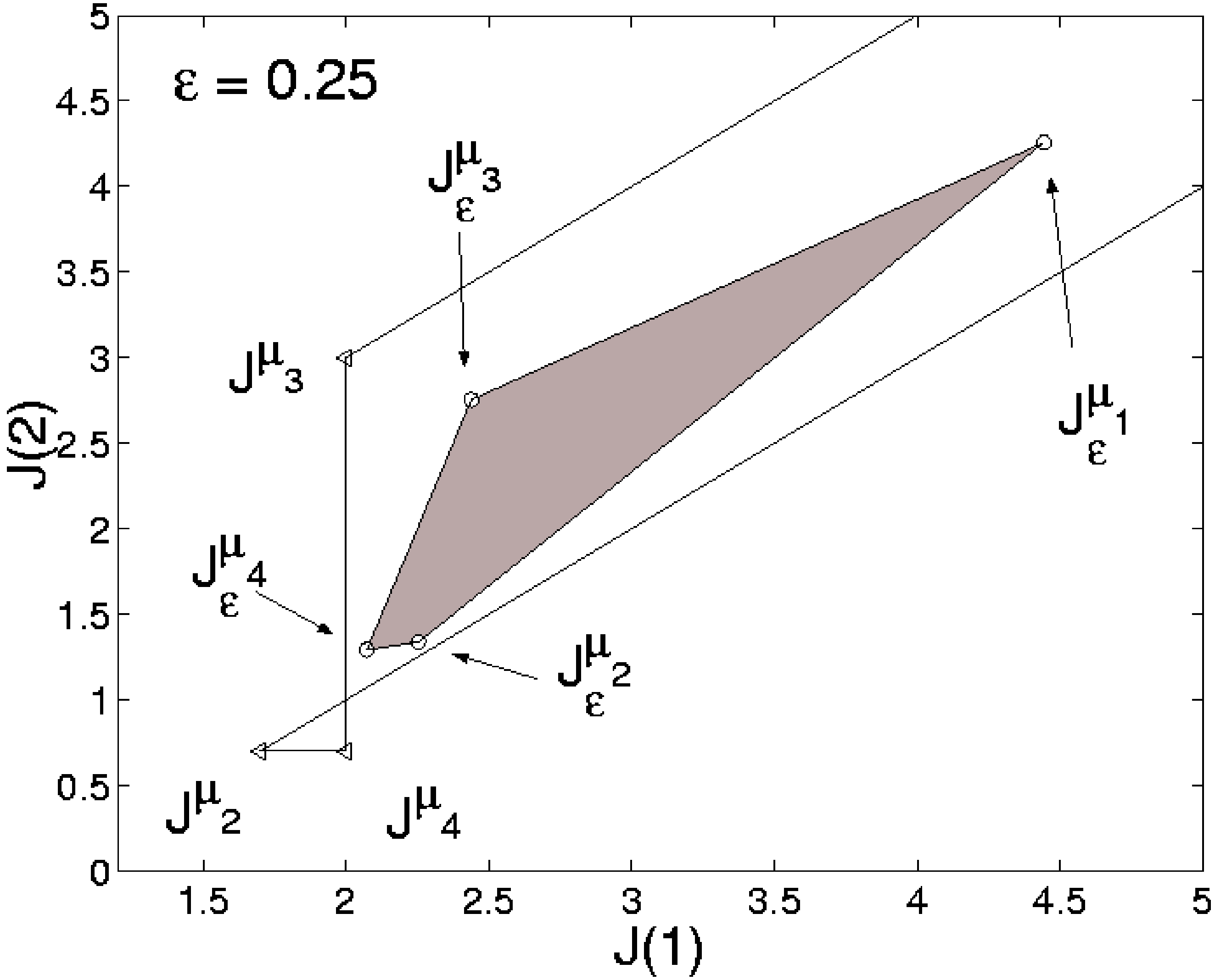
Points: costs of policies μ_2, μ_3, μ_4 applied to DSP problem



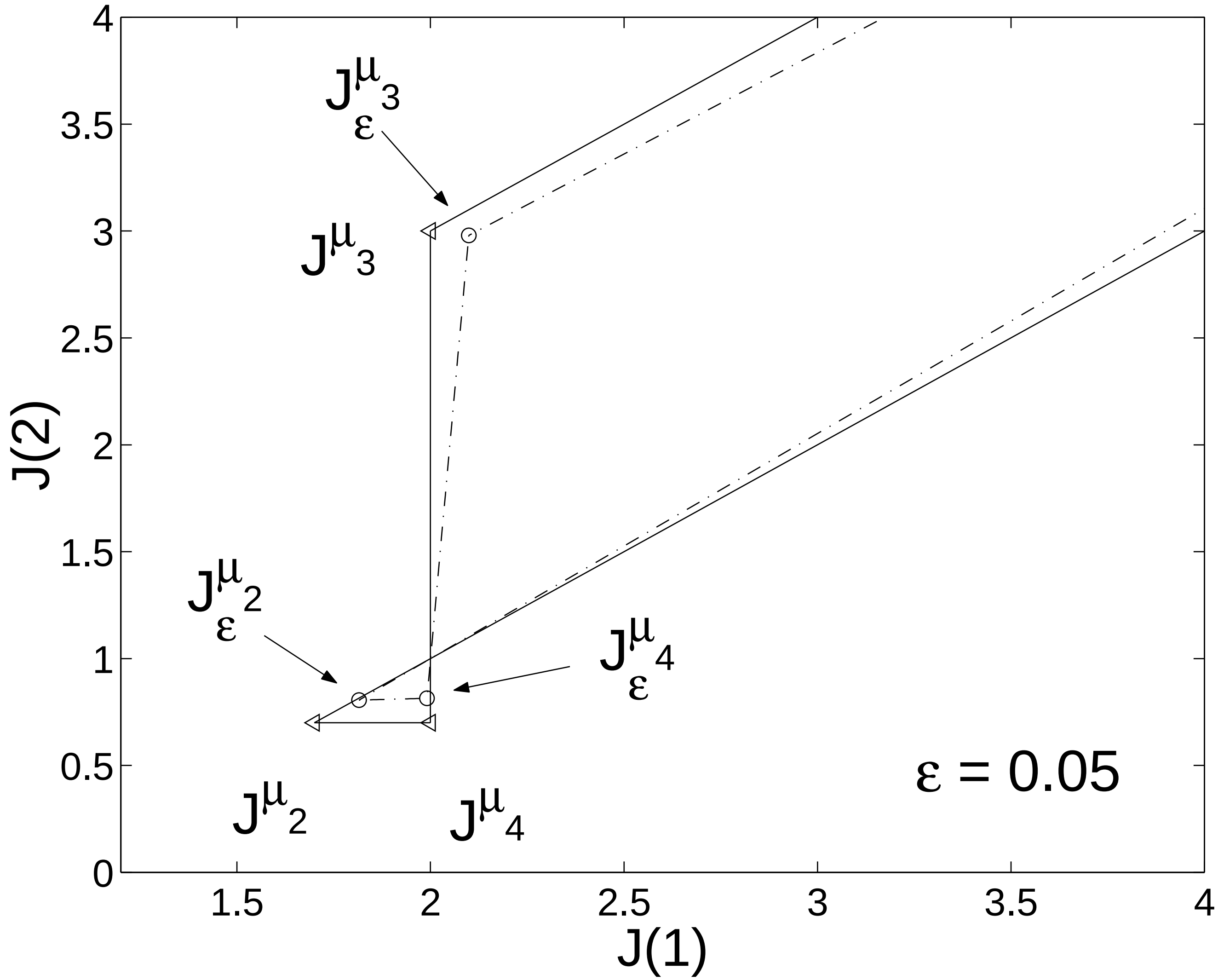
Region: costs of arbitrary polices applied to DSP problem



Region: costs of arbitrary randomized polices applied to “ϵ-modified” MDP



Reduced exploration level



Summary of insights from MDP analysis

1. Exploration via policy randomization effectively *modifies* the MDP that the ant-based algorithm is attempting to solve
2. The modified MDP may have a different optimal policy to the original
3. This can lead to an error in the identification of the optimal policy by the ants/learning agents.

Implications for ant-based routing

- Can try and set exploration level “sufficiently small”...
- ...but difficult to establish this level “a priori”
- Better alternative: de-couple the mechanisms for exploration and decision-making

Restrict exploration to first hop

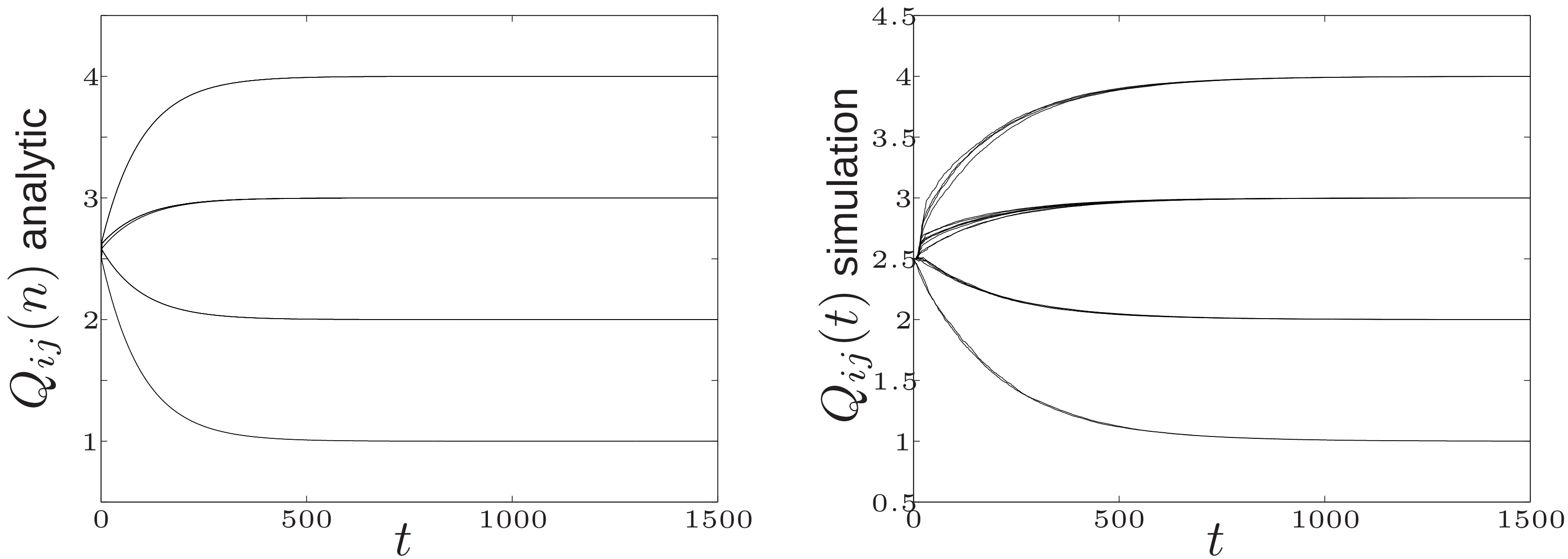
De-couple the mechanisms for exploration and “exploitation” by restricting exploration to ants’ first hop decision.

“Exploit” links which have minimum delay estimates for all subsequent decisions until d is reached (greedy routing).

Theorem : (analytic model) provided ants explore on their first hop,

$$\lim_{\beta \to \infty} Q_{ij} = Q^*_{ij}.$$

Proof: based on proof of the policy iteration algorithm (dynamic programming).



link delays $r_{ij} = 1$ for all links (i, j), $\beta \to \infty$

	ant algorithm	optimal		ant algorithm	optimal
ϕ_{12}	0	0	Q_{12}	4	4
ϕ_{18}	1	1	Q_{18}	3	3

Analysis - flow-dependent link delays

Introducing data traffic demands leads to flow-dependent link delays.

$$\begin{aligned} \text{expected link delay} &= \text{expected queueing delay} + \text{fixed transmission delay} \\ &= d(f_{ij}(n)) + r_{ij} \end{aligned}$$

For example,

$$d(f_{ij}(n)) = \begin{cases} \frac{1}{(C_{ij} - f_{ij}(n))} & \text{if} \quad f_{ij}(n) < C_{ij}, \\ \infty & \text{if} \quad f_{ij}(n) \geq C_{ij}. \end{cases}$$

where C_{ij} is the service rate parameter of link (i, j).

Analysis - flow-dependent link delays

- Data traffic introduces a strong coupling between the traffic routing policy, and the delays that are experienced by ants and data packets on each link.
- Multiple traffic streams (different origin-destination node pairs) effectively "compete" for finite shared resources (link capacities).

How to evaluate a given routing policy ?

- A particular routing policy may be beneficial to one traffic stream but detrimental to another. This leads to the notion of "users", user optimisation and user-equilibria
- Alternatively, an average system-wide performance measure can be used to evaluate a given routing policy, thus leading to the notion of a system optimum.

Analysis - flow-dependent link delays

Appropriate optimisation concepts

- System optimization (minimize total average flow-weighted delay)
- User optimization
  - Nash equilibria (users = traffic stream)
  - Wardrop equilibria (users = individual packets)

The Wardrop equilibrium arises as a special limiting case of the more general Nash equilibrium:

[number of users $\longrightarrow \infty$, total user demand remains constant, each user's decisions has negligible impact on the decisions of other users]

Traffic optimization and equilibrium

The analysis of traffic equilibria on networks originated with the work of Wardrop (1952), which developed a means for analysing and characterizing vehicle traffic flows on road networks.

Wardrop’s First Principle can be stated equivalently in the following three ways:

> “The travel times on all *used* paths between an origin and a destination point are equal, and less than those which would be experienced by a single vehicle on any unused path”

or

> “No traveler can improve his travel time by unilaterally changing routes”

or

> “Every traveler follows the minimum travel time path”.

Analysis - flow-dependent link delays

Replace “traveler” and “vehicle” with “packet” in the above definitions, we have

Definition of Wardrop equilibrium:

“*A data traffic routing policy* Ψ *corresponds to a Wardrop equilibrium if no packet can unilaterally decrease its trip time from its origin to the destination by following a policy that is different to* Ψ”.

The Wardrop equilibrium was later shown to be a special case of the Nash equilibrium, thus establishing a connection between the study of traffic equilibria on networks and game theory.

Analysis - flow-dependent link delays

It turns out that system optima are not (systematically) attainable by ant-based routing algorithms, because

- ants perform delay (trip time) measurements, not marginal delay measurements.

Also, Nash equilibria are not (systematically) attainable by ant-based routing algorithms, because

- "stateless" routing - each node routes packets according their destination node, not according to their node of origin.

However, it turns out that ant-based routing algorithms are not incompatible with Wardrop equilibria.

Analysis - flow-dependent link delays

Our studies using the analytic model demonstrate that the “heuristic” routing policies produced by the ant-based routing algorithm are perturbed Wardrop equilibria.

These perturbations result in sub-optimal performance (with respect to both system and individual packet-based performance measures).

This is due to inherent coupling between the tasks of network exploration and exploitation.

As before, routing can be made more efficient by de-coupling the mechanisms which perform these tasks.

Analysis - flow-dependent link delays

Subject to the following modifications, ant-based routing algorithms are able to attain Wardrop equilibria, which constitute a form of packet-based optimisation

- ants perform exploration when selecting their first hop node (guarantees exploration)
- ants follow the current data routing policy for all subsequent link selections (ants “see” same delays as data packets)
- allow data routing probabilities to take the values 0 and 1 if necessary.

Summary of Results

1. Using randomized policies as an exploration mechanism:

Flow-dependent link delays	$\longrightarrow$	deviations from Wardrop equilibria
	analogous to	
Fixed link delays	$\longrightarrow$	deviations from shortest paths

2. Restricting exploration to agents' first hop (then follow data routing policy):

Flow-dependent link delays	$\longrightarrow$	algorithm finds Wardrop equilibria
	analogous to	
Fixed link delays	$\longrightarrow$	algorithm finds shortest paths

Discussion

- In current ant-based routing algorithms, optimality is traded for some degree of robustness
- This tradeoff can be eliminated by de-coupling exploration from data traffic routing policy (exploitation)
- Concurrence and asynchronism in the real system introduce additional convergence issues that we have not addressed.