A Generic Mean Field Model for Optimization in Large-scale Stochastic Systems and Applications in Scheduling

Nicolas Gast Bruno Gaujal

Grenoble University

Knoxville, May 13th-15th 2009

Introduction

Mean field has been introduced by physicists to study systems of interacting objects. For example, the movement of particles in the air:

First solution: the microscopic description

The system is represented by the states of each particle.

- Many equations for each possible collision: impossible to solve exactly.

Second solution (better!): macroscopic equations

We are interested by the average behavior of the system:

- The system is described by its temperature.
- Deterministic equation.

- The transition from microscopic description to macroscopic equations is called the mean field approximation.

Mean Field in Computer Science

More recently, Mean field has been used to analyze performance of communication systems. The objects are the users in the system. For example:

- Performance of TCP [Baccelli, McDonald, Reynier [02]]
- Reputation Systems [Le Boudec et al. [07]]
- 802.11 [Bordenave, McDonald, Proutière [05]]
- ...

In many example, it can be shown that when the number of users grows, the average behavior of the system becomes deterministic.

Mean Field in Computer Science

More recently, Mean field has been used to analyze performance of communication systems. The objects are the users in the system. For example:

- Performance of TCP [Baccelli, McDonald, Reynier [02]]
- Reputation Systems [Le Boudec et al. [07]]
- 802.11 [Bordenave, McDonald, Proutière [05]]
- ...

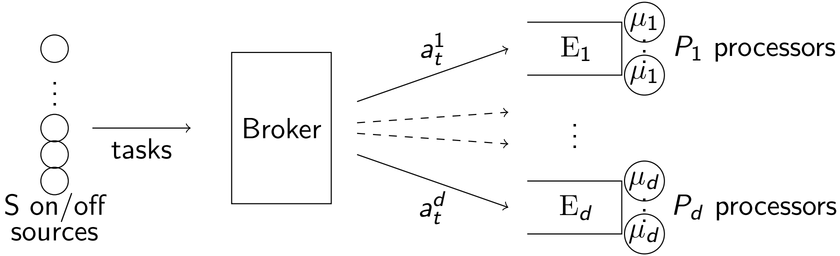
In many example, it can be shown that when the number of users grows, the average behavior of the system becomes deterministic.

Aim of this talk

- Show that mean field can also be used for optimization problem.
- Study a general framwork for wich we can prove the results.

Example of mean field model

Example – Consider the following brokering problem:

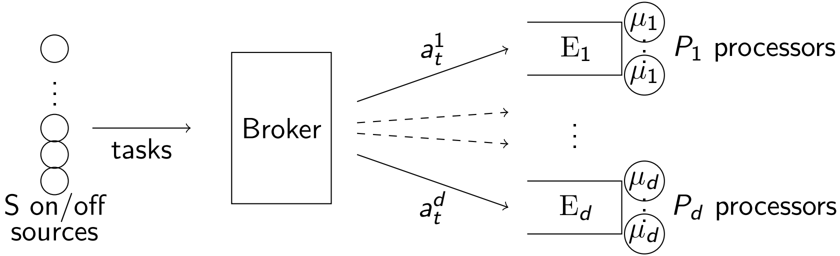


Stochastic system

- Objects are sources+Processors: There are $S + P_1 + \cdots + P_d$ objects
- The state of an object is *active* or *inactive* (random)
- Evolution of state is markovian

Example of mean field model

Example – Consider the following brokering problem:



Stochastic system

- Objects are sources+Processors: There are $S + P_1 + \cdots + P_d$ objects
- The state of an object is *active* or *inactive* (random)
- Evolution of state is markovian

Mean field limit

We scale S and P_i by N.
We are interested in:

- (number of tasks sent)$/N$.
- (available processors in cluster i)$/N$.

Main result

Stochastic system N objects.

Compute optimal policy (hard)

Optimal system

Remark: the purpose of this talk is not to solve the previous example but to study a general framework for optimization in stochastic systems.

Main result

Stochastic system N objects. —— $N \to \infty$ ——> Mean Field Limit

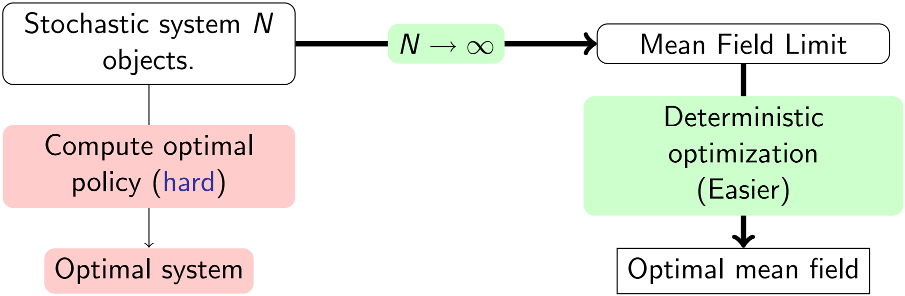
Compute optimal policy (hard)

↓

Optimal system

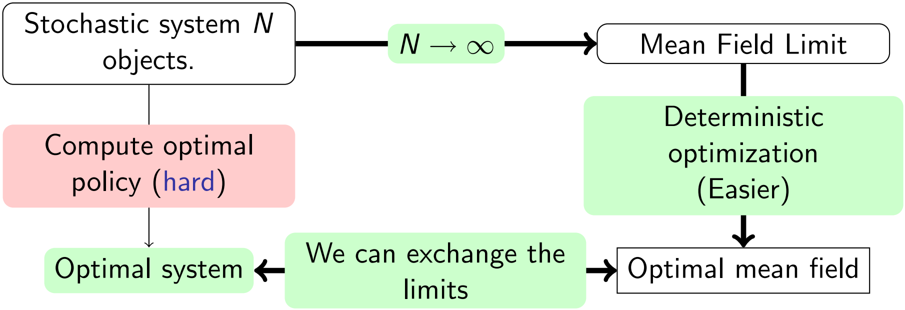
1. Compute the mean field limit (constructive definition).

Main result



1. Compute the mean field limit (constructive definition).
2. Solve the deterministic problem.

Main result



Our results

The optimal stochastic system also converges.

More precisely, when N grows:

1. The optimal reward converges.
2. The optimal policy also converges.
3. The speed of convergence is $O(\sqrt{N})$ (CLT theorem).

1. Theoretical Results

2. A (simple) example

3. Conclusion

An optimization problem

- N objects evolving in a finite state space.
- Environment $E(t)$ at time t ($E(t) \in \mathbb{R}^d$)

$$\begin{bmatrix} X_1(0) \\ \vdots \\ X_N(0) \\ E(0) \end{bmatrix} \xrightarrow{\text{random}(0)} \begin{bmatrix} X_1(1) \\ \vdots \\ X_N(1) \\ E(1) \end{bmatrix} \xrightarrow{\text{random}(1)} \cdots\cdots \xrightarrow{\text{random}(T-1)} \begin{bmatrix} X_1(T) \\ \vdots \\ X_N(T) \\ E(T) \end{bmatrix}$$

An optimization problem

- N objects evolving in a finite state space.
- Environment $E(t)$ at time t ($E(t) \in \mathbb{R}^d$)

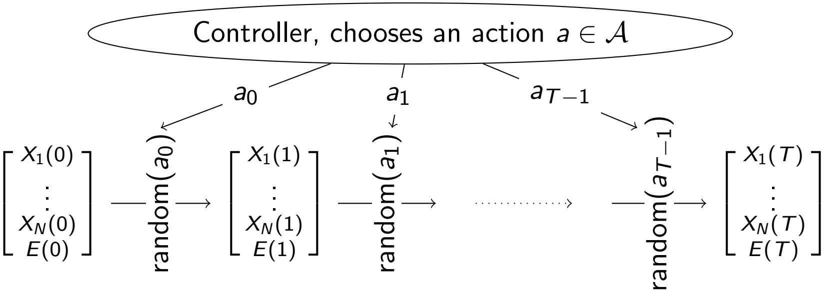
$$\begin{bmatrix} X_1(0) \\ \vdots \\ X_N(0) \\ E(0) \end{bmatrix} \xrightarrow{\text{random}(0)} \begin{bmatrix} X_1(1) \\ \vdots \\ X_N(1) \\ E(1) \end{bmatrix} \xrightarrow{\text{random}(1)} \cdots\cdots \xrightarrow{\text{random}(T-1)} \begin{bmatrix} X_1(T) \\ \vdots \\ X_N(T) \\ E(T) \end{bmatrix}$$

Mean field assumption

We define the Population mix $\mathrm{M}(t)$ – The i^{th} component $(\mathrm{M}(t))_i$ is the proportion of objects in state i at time t.

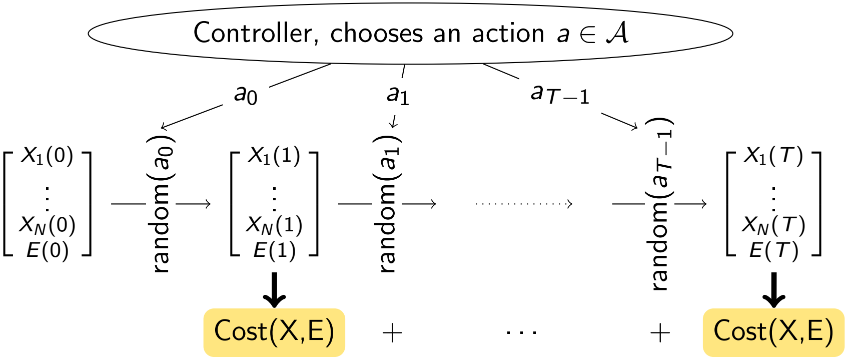
1. $E(t+1)$ only depends on the population mix $\mathrm{M}(t)$.
2. The evolution of an object depends on $E(t)$ but is independent of the other objects.

An optimization problem



The controller can change the dynamics of the system.

An optimization problem



The controller can change the dynamics of the system.

Goal

Find a policy to maximize:

- finite-time expected cost *or*
- expected discounted cost

Technical assumption:

- Mean field evolution
- Action set compact
- Continuous parameters

Optimal cost convergence

- V^{*N} – optimal cost for the system of size N.
- v^* – optimal cost for the deterministic limit.
- $a_0^* a_1^* a_2^* \dots$ – optimal actions for the deterministic limit.

Theorem (Convergence of the optimal cost)

Under technical assumptions, for both discounted and finite-time cost:

$$\begin{array}{rcl} \lim_{N\to\infty} V^{*N} & = & v^* \\ \lim_{N\to\infty} V^{*N} & = & V^N_{a_0^* a_1^* a_2^* \dots} \end{array} \qquad \text{(a.s.)}$$

In particular, this shows that:

- Optimal cost converges
- Static policy a^* is asymptotically optimal

A central limit-like theorem

The convergence speed is in $O(1/\sqrt{N})$:

Theorem (CLT for the evolution of objects)

Under technical assumptions, if the actions taken by the controller are fixed, then there exists a Gaussian variable G_t s.t:

$$\sqrt{N}\left(\mathrm{M}_t^N - m_t\right) \xrightarrow{\text{Law}} G_t$$

The covariance of G_t can be effectively computed.

A central limit-like theorem

The convergence speed is in $O(1/\sqrt{N})$:

Theorem (CLT for the evolution of objects)

Under technical assumptions, if the actions taken by the controller are fixed, then there exists a Gaussian variable G_t s.t:

$$\sqrt{N}\left(\mathrm{M}_t^N - m_t\right) \xrightarrow{\text{Law}} G_t$$

The covariance of G_t can be effectively computed.

Theorem (CLT for cost)

Under technical assumptions, when N goes to infinity:

$$\begin{aligned} \sqrt{N}\left|V_T^{*N} - V_{a^*}^N\right| &\leq_{\text{st}} \beta + \gamma \|G_0\|_\infty \\ \sqrt{N}\left|V_T^{*N} - v_T^*\right| &\leq_{\text{st}} \beta' + \gamma' \|G_0\|_\infty \end{aligned}$$

A simple resource allocation problem

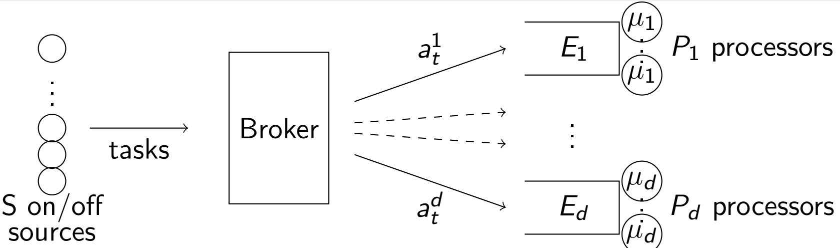
Aim of the example

- Illustrate the framework by a concrete example
- When does $N (= S + P_1 + \cdots + P_d)$ becomes large enough for the approximation to apply?

A simple resource allocation problem

Aim of the example

- Illustrate the framework by a concrete example
- When does $N(= S + P_1 + \cdots + P_d)$ becomes large enough for the approximation to apply?



Stochastic arrivals

Stochastic availability: failure,...

- Optimize the total completion time $= \sum_{t=0}^{T} \sum_{i=1}^{d} E_i(t)$.

Optimal policy: stochastic and limit case

The stochastic system is hard to solve

1. This problem is a multidimensional restless bandit problem
   - Known to be hard
   - Existence of heuristics (Index policies)
2. In practice in such systems [EGEE]
   - Use of heuristics (JSQ)

Optimal policy: stochastic and limit case

The stochastic system is hard to solve

1. This problem is a multidimensional restless bandit problem
   - Known to be hard
   - Existence of heuristics (Index policies)
2. In practice in such systems [EGEE]
   - Use of heuristics (JSQ)

Using our framework: compute optimal mean field

The problem becomes:

- Find an allocation to minimize the idle time of processors.
- All variable are in $\mathbb{R}^d$.
- The optimal policy can be computed by a greedy algorithm.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I I						
Queue 2							
Queue 3	I I						
Optimal allocation							

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0				
	I	P0	P0				
	P0						
Queue 2			P0				
Queue 3	I	P0					
	I	P0					
		P0					
Optimal allocation	5						
	1						
	3						

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0				
	I	P0	P0				
	P0						
Queue 2			P0				
Queue 3	I	P0	P1				
	I	P0					
		P0					
Optimal allocation	5	.					
	1	.					
	3	1					

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0	P3			
	I	P0	P0				
	P0						
Queue 2			P0				
Queue 3	I	P0	P1				
	I	P0					
		P0					
Optimal allocation	5	.	.	1			
	1	.	.	.			
	3	1	.	.			

- Grey = “off” processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0	P3	P4	P4	
	I	P0	P0		P4		
	P0				P4		
					P4		
Queue 2			P0				
Queue 3	I	P0	P1		P4		
	I	P0			P4		
		P0					
Optimal allocation	5	.	.	1	5		
	1	.	.	.	.		
	3	1	.	.	2		

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0	P3	P4	P4	
	I	P0	P0		P4	P5	
	P0				P4		
					P4		
Queue 2			P0			P5	
						P5	
Queue 3	I	P0	P1		P4	P5	
	I	P0			P4	P5	
		P0				P5	
Optimal allocation	5	.	.	1	5	1	
	1	.	.	.	.	2	
	3	1	.	.	2	3	

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0	P3	P4	P4	P6
	I	P0	P0		P4	P5	
	P0				P4		
					P4		
Queue 2			P0			P5	P6
						P5	
Queue 3	I	P0	P1		P4	P5	P6
	I	P0			P4	P5	P6
		P0				P5	
Optimal allocation	5	.	.	1	5	1	1
	1	.	.	.	.	2	1
	3	1	.	.	2	3	2

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.

Computing the optimal policy π^*

Time t	0	1	2	3	4	5	6
Arrival of packets	9	1	0	1	7	6	6
Queue 1	I	P0	P0	P3	P4	P4	P6
	I	P0	P0		P4	P5	
	P0				P4		
					P4		
Queue 2			P0			P5	P6
						P5	
Queue 3	I	P0	P1		P4	P5	P6
	I	P0			P4	P5	P6
		P0				P5	
Optimal allocation	5	.	.	1	5	1	1+2
	1	.	.	.	.	2	1
	3	1	.	.	2	3	2

- Grey = "off" processors.
- I : initial packets.
- P0: packets arrived at time 0.
- 2 packets remains at the end.

Numerical example

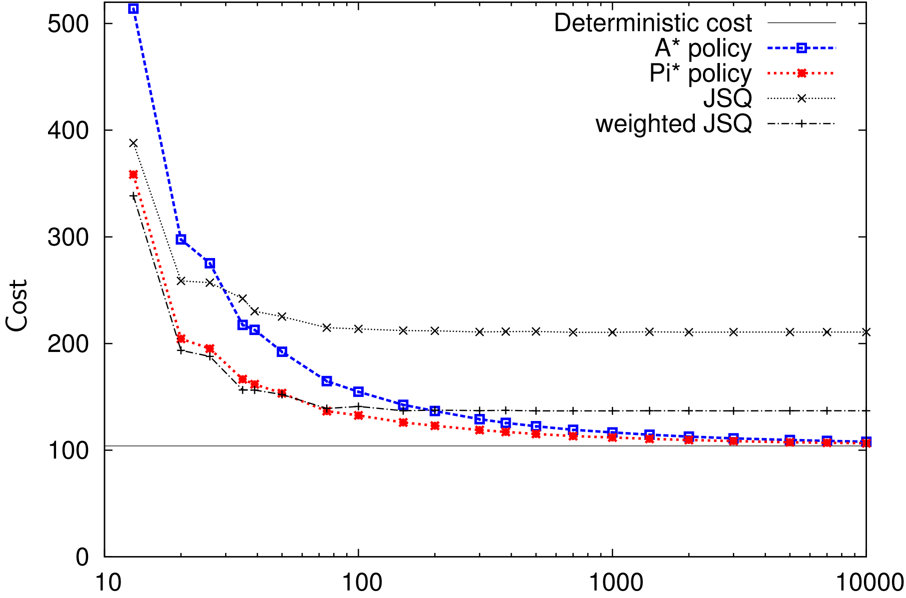
This provides two policies for the initial stochastic system.

- π^* : at t, we apply $\pi^*_t(\mathrm{M}^N_t, \mathrm{E}^N_t)$ – adaptive policy.
- a^* : we apply $a^*_t \stackrel{\mathrm{def}}{=} \pi^*_t(m_t, e_t)$ – static policy.

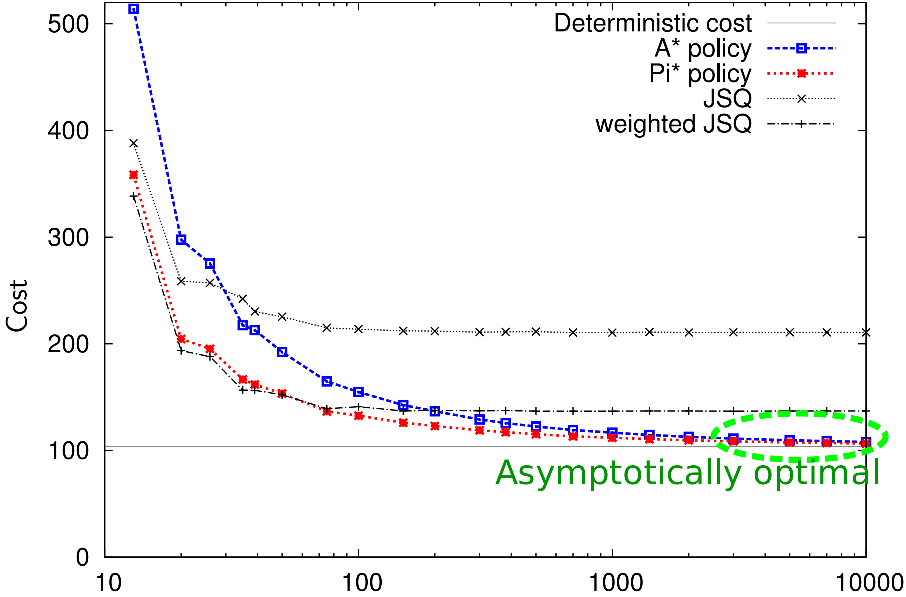
We want to compare

- V^{*N} – optimal cost for the system of size N
- $V^N_{a^*}$ – cost when applying a^*
- $V^N_{\pi^*}$ – cost when applying π^*
- V^N_{JSQ} – cost of Join Shortest Queue.
- v^* – cost of the deterministic limit.

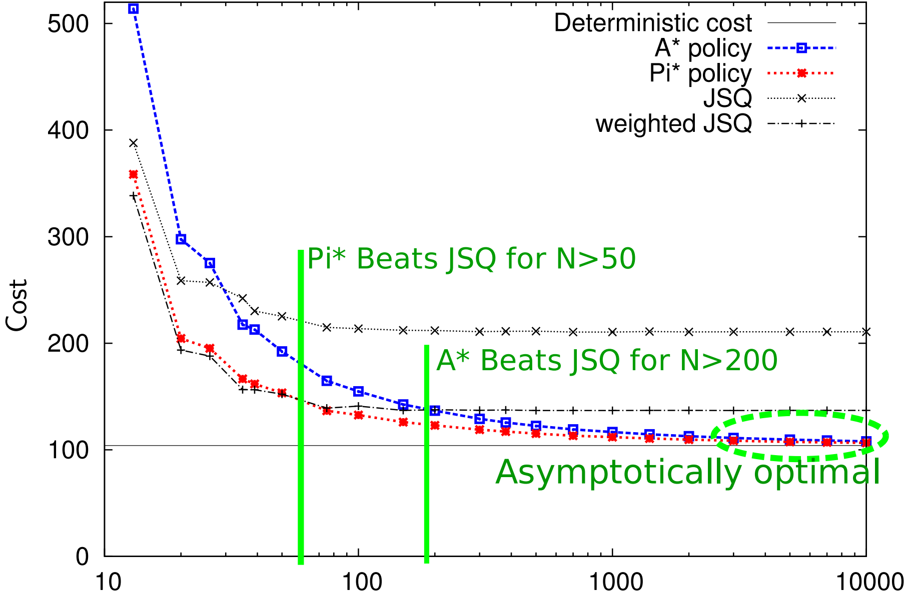
Cost convergence



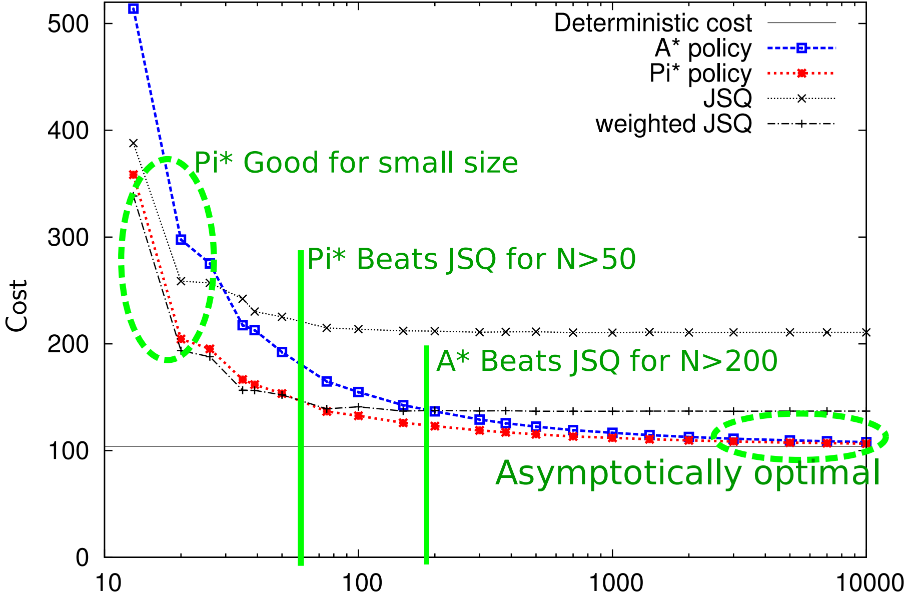
Cost convergence



Cost convergence

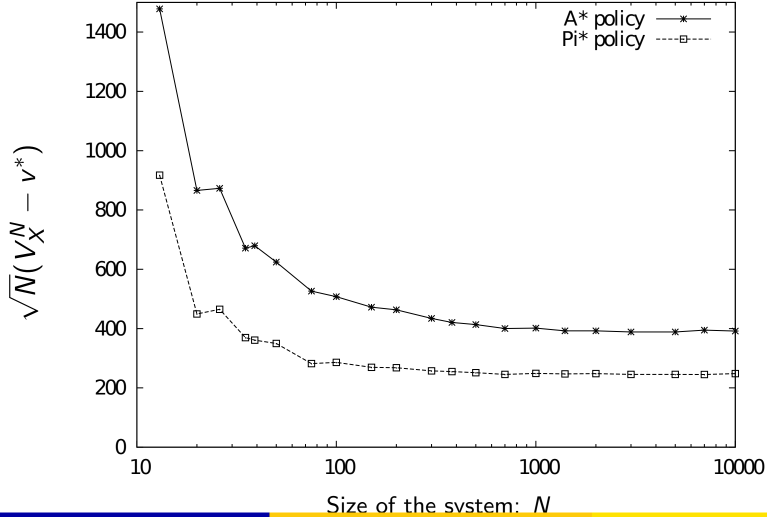


Cost convergence



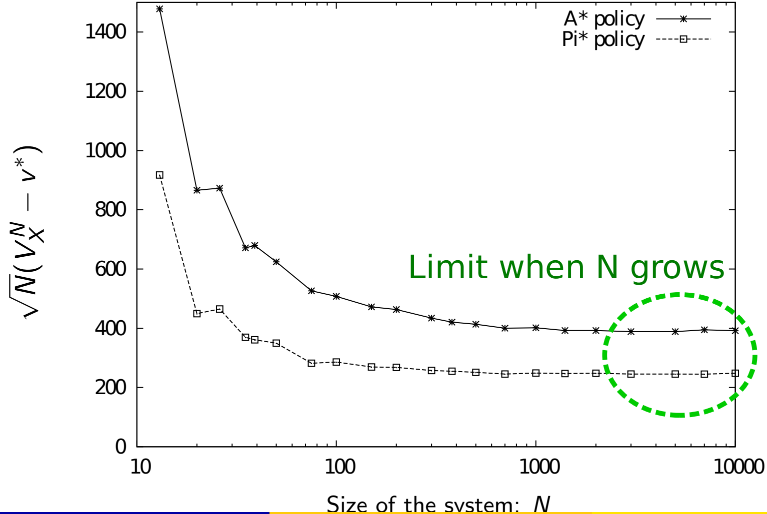
Speed of convergence – central limit theorem

Plot of $\sqrt{N}(V^N_{\pi^*} - v^*)$ and $\sqrt{N}(V^N_{a^*} - v^*)$



Speed of convergence – central limit theorem

Plot of $\sqrt{N}(V_{\pi^*}^N - v^*)$ and $\sqrt{N}(V_{a^*}^N - v^*)$



1. Theoretical Results

2. A (simple) example

3. **Conclusion**

Conclusion

- Optimal policy of the deterministic limit is asymptotically optimal.
- Works for low values of N (≈ 100 in the example).

To apply this in practice, there are three cases (from best to worse):

1. We can solve the deterministic limit:
   - apply a^* or π^*.
2. Design an approximation algorithm for the deterministic system:
   - also an approximation (asymptotically) for stochastic problem.
3. Use brute force computation:
   - $v^*_{t\dots T}(m,e) = C(m,e) + \sup_a v^*_{t+1\dots T}(\phi_a(m,e))$
   - Compared to the random case, there is no expectation to compute.

Conclusion

- Optimal policy of the deterministic limit is asymptotically optimal.
- Works for low values of N (≈ 100 in the example).

To apply this in practice, there are three cases (from best to worse):

1. We can solve the deterministic limit:
   - apply a^* or π^*.
2. Design an approximation algorithm for the deterministic system:
   - also an approximation (asymptotically) for stochastic problem.
3. Use brute force computation:
   - $v^*_{t\dots T}(m,e) = C(m,e) + \sup_a v^*_{t+1\dots T}(\phi_a(m,e))$
   - Compared to the random case, there is no expectation to compute.

- In general, the stochastic case is impossible to solve and this problem is usually addressed using restricted classes of policies:
  - With limited information, Static/Adaptative, ...

  We showed that this distinction asymptotically collapses.

References

- Paper corresponding to this talk:
  - Gast N., Gaujal B. – *A Mean Field Approach for Optimization in Particles Systems and Applications* – RR 6877, `http://mescal.imag.fr/membres/nicolas.gast/`
- Mean field models:
  - Le Boudec, McDonald, Mudinger – *A Generic Mean Field Convergence Result for Systems of Interacting Objects* – QEST 2007
  - Le Boudec, Benaïm – *A Class Of Mean Field Interaction Models for Computer and Communication Systems* – Performance Evaluation
- Infinite horizon results:
  - Borkar – *Stochastic Approximation: A Dynamical Systems Viewpoint* – Cambridge University Press 2008

Thanks

Thank you for your attention.