

Cooperative Task Processing: A Framework

Ted Pavlic

January 19, 2010

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation:

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

- ◆ Agents are compelled to optimize a global good

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

- ◆ Agents are compelled to optimize a global good
- ◆ Designs stifle emergent behavior

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

- ◆ Agents are compelled to optimize a global good
- ◆ Designs stifle emergent behavior
- ◆ Complex designs can have unrealistic communication/shared information requirements

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

- ◆ Agents are compelled to optimize a global good
- ◆ Designs stifle emergent behavior
- ◆ Complex designs can have unrealistic communication/shared information requirements
- ◆ Looks nothing like the cooperation of interest to biologists and sociologists

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

- ◆ Agents are compelled to optimize a global good
- ◆ Designs stifle emergent behavior
- ◆ Complex designs can have unrealistic communication/shared information requirements
- ◆ Looks nothing like the cooperation of interest to biologists and sociologists

■ Innovation:

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Motivation: **Cooperative control is boring**

- ◆ Agents are compelled to optimize a global good
- ◆ Designs stifle emergent behavior
- ◆ Complex designs can have unrealistic communication/shared information requirements
- ◆ Looks nothing like the cooperation of interest to biologists and sociologists

■ Innovation: **Framework that introduces interesting cooperation to control engineers**

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Toward a working definition

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Toward a working definition

◆ A cooperative act benefits another

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Toward a working definition
 - ◆ A cooperative act benefits another
 - ◆ No surprise that agents with global utility function cooperate

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Toward a working definition

- ◆ A cooperative act benefits another
- ◆ No surprise that agents with global utility function cooperate
- ◆ No surprise that agents with local utility functions cooperate when remote benefit is a byproduct

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Toward a working definition
 - ◆ A cooperative act benefits another
 - ◆ No surprise that agents with global utility function cooperate
 - ◆ No surprise that agents with local utility functions cooperate when remote benefit is a byproduct
 - ◆ Altruistic (**interesting**) case: Benefit to another at apparent cost to self

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Toward a working definition
 - ◆ A cooperative act benefits another
 - ◆ No surprise that agents with global utility function cooperate
 - ◆ No surprise that agents with local utility functions cooperate when remote benefit is a byproduct
 - ◆ Altruistic (**interesting**) case: Benefit to another at apparent cost to self

- So *altruism* is interesting case of cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
 - ◆ “No, but I would to save two brothers or eight cousins.” (J.B.S. Haldane, in response to whether he would die to save a drowning brother)

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
 - ◆ “No, but I would to save two brothers or eight cousins.” (J.B.S. Haldane, in response to whether he would die to save a drowning brother)
 - ◆ Explains altruism among relatives but not friends or worse

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness
 - ◆ Axelrod developed protocols of reciprocity that cooperate when future encounters are certain

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness
 - ◆ Axelrod developed protocols of reciprocity that cooperate when future encounters are certain
 - ◆ Axelrod's protocols observed in nature by many (e.g., Milinski's sticklebacks, Dugatkin's guppies)

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness
- Nowak et al. show that cooperation emerges via birth–death processes on networks

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness
- Nowak et al. show that cooperation emerges via birth–death processes on networks
 - ◆ **Non-random assortment** can favor cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness
- Nowak et al. show that cooperation emerges via birth–death processes on networks
 - ◆ **Non-random assortment** can favor cooperation
 - ◆ Cooperation thrives when average number of neighbors is low (i.e., when future is tightly bound to others)

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Hamilton's rule: Cooperation is beneficial when $c/b < r$ (r : **relatedness**—function of distance on family tree)
- Trivers suggested that future **reciprocity** can be a surrogate for relatedness
- Nowak et al. show that cooperation emerges via birth–death processes on networks
- Nowak et al. also show that in all cases, *relatedness* can be defined so that Hamilton's c/b rule holds
 - ◆ $c/b < r, c/b < w, c/b < 1/k$

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
 - ◆ Techniques typically used to model noise or parameter variations (i.e., competing player whose interests are not necessarily aligned)

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
 - ◆ Techniques typically used to model noise or parameter variations (i.e., competing player whose interests are not necessarily aligned)
 - ◆ Methods also used to model behaviors of human agents interacting with the system

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
 - ◆ Techniques typically used to model noise or parameter variations (i.e., competing player whose interests are not necessarily aligned)
 - ◆ Methods also used to model behaviors of human agents interacting with the system
 - ◆ Ad hoc multi-hop networks (Altman et al., Hubaux et al.) choose to forward packets at cost to local bandwidth/power, but packets are not tasks

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz
 - ◆ Flexible manufacturing system, network components \implies bounded queues/burstiness

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz
 - ◆ Flexible manufacturing system, network components \implies bounded queues/burstiness
 - ◆ Behaviors are static

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz
- So we combine task-processing networks with non-cooperative game theory

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz
- So we combine task-processing networks with non-cooperative game theory
 - ◆ Study distributed agent-level behaviors that converge to competitive (Nash) equilibrium

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz
- So we combine task-processing networks with non-cooperative game theory
 - ◆ Study distributed agent-level behaviors that converge to competitive (Nash) equilibrium
 - ◆ Behaviors rely on little coordination

Engineering Interesting Cooperation

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Realm of non-cooperative/competitive game theory
- Task-processing networks described by Perkins and Kumar/Cruz
- So we combine task-processing networks with non-cooperative game theory
 - ◆ Study distributed agent-level behaviors that converge to competitive (Nash) equilibrium
 - ◆ Behaviors rely on little coordination
 - ◆ Competitive equilibrium respects both local and global utility

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

- $\mathcal{A} \subset \mathbb{N}$: Set of *task-processing agents*
- $\mathcal{P} \subseteq \{(i, j) \in \mathcal{A}^2 : i \neq j\}$: Directed arcs connecting distinct agents
- $\mathcal{V}_i \triangleq \{j \in \mathcal{A} : (j, i) \in \mathcal{P}\}$: Set of *conveyors* for each $i \in \mathcal{A}$
- $\mathcal{C}_i \triangleq \{j \in \mathcal{A} : (i, j) \in \mathcal{P}\}$: Set of *cooperators* for each $i \in \mathcal{A}$
- $\mathcal{V} \triangleq \{j \in \mathcal{A} : \mathcal{C}_j \neq \emptyset\}$: Set of all conveyors
- $\mathcal{C} \triangleq \{i \in \mathcal{A} : \mathcal{V}_i \neq \emptyset\}$: Set of all cooperators
- $\mathcal{Y}_i \subset \mathbb{N}$: Possibly empty set of *task types* that arrive at conveyor $i \in \mathcal{A}$
- $\lambda_j^k \in \mathbb{R}_{>0}$: Encounter rate of type- k tasks at agent $j \in \mathcal{A}$
- $\pi_j^k \in [0, 1]$: Probability that conveyor $j \in \mathcal{A}$ advertises an incoming k -type task to its connected cooperators \mathcal{C}_j
- $\gamma_i \in [0, 1]$: Probability that cooperator $i \in \mathcal{A}$ volunteers for advertised task from one of its connected conveyors \mathcal{V}_i

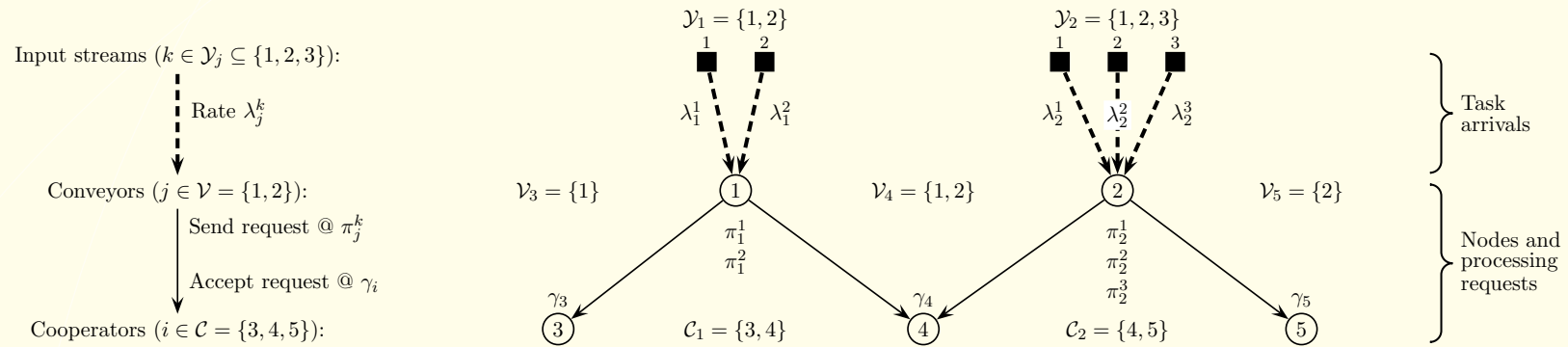


Figure 1: Flexible manufacturing system (FMS)

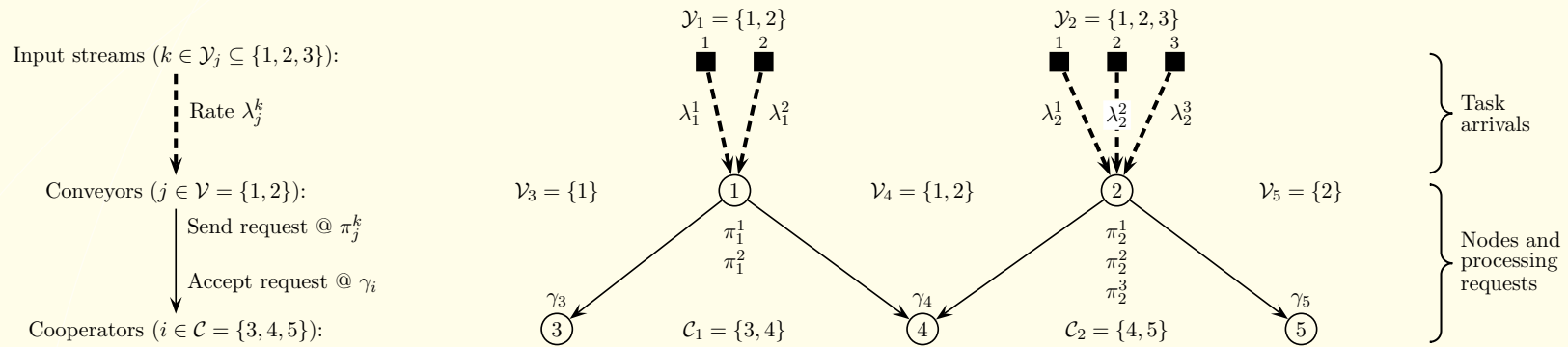


Figure 1: Flexible manufacturing system (FMS)

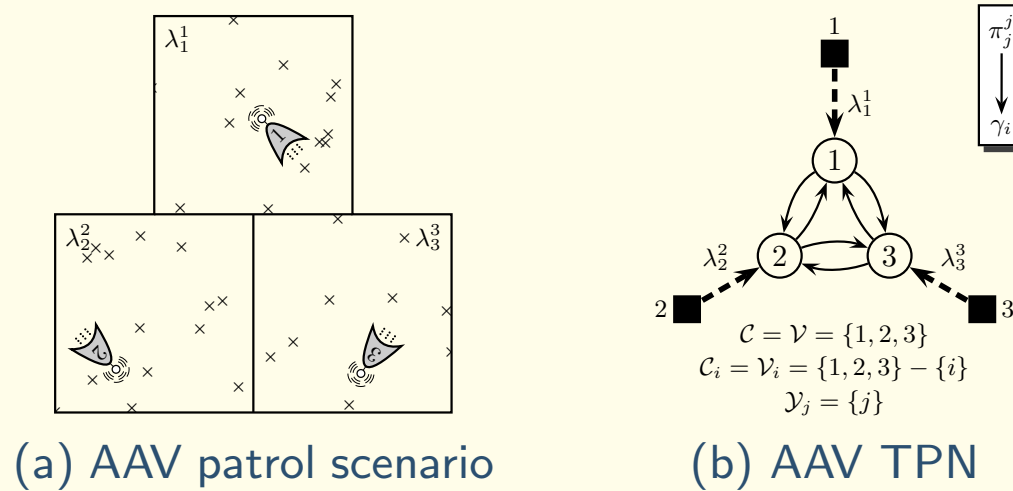


Figure 2: AAV patrol scenario

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

Cooperation Game

$$U_i(\underline{\gamma}) \triangleq \underbrace{b_i + \left(1 - \prod_{j \in \mathcal{C}_i} (1 - \gamma_j)\right) r_i - Q_i p_i(Q_i)}_{\text{Conveyor part — constant with respect to } \gamma_i} + \underbrace{\gamma_i \sum_{j \in \mathcal{V}_i} (p_{ij}(Q_j) - \text{SOBP}_1(\mathcal{C}_j - \{i\}) c_{ij})}_{\text{Pr}(i \text{ awarded task from } j | i \text{ volunteers})}$$

$\text{Pr}(\text{Volunteer from } \mathcal{C}_i | \text{Advertisement from } i)$
Cooperator part — γ_i and Q_j vary with γ_i

where

$$b_i \triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k (b_i^k - c_i^k),$$

$$r_i \triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \pi_i^k (r_i^k - (b_i^k - c_i^k)),$$

$$p_i(Q_i) \triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \pi_i^k p_i^k(Q_i),$$

are the costs and benefits of local processing on $i \in \mathcal{V}$,

and

$$c_{ij} \triangleq \sum_{k \in \mathcal{Y}_j} \lambda_j^k \pi_j^k c_{ij}^k,$$

$$p_{ij}(Q_j) \triangleq \sum_{k \in \mathcal{Y}_j} \lambda_j^k \pi_j^k q_{ij}^k p_j^k(Q_j).$$

are the costs and benefits to $i \in \mathcal{C}$ for volunteering for tasks exported from $j \in \mathcal{V}_i$.

$$U_i(\underline{\gamma}) \triangleq \underbrace{b_i + \left(1 - \prod_{j \in \mathcal{C}_i} (1 - \gamma_j)\right) r_i - Q_i p_i(Q_i)}_{\text{Conveyor part — constant with respect to } \gamma_i} + \underbrace{\gamma_i \sum_{j \in \mathcal{V}_i} (p_{ij}(Q_j) - \text{SOBP}_1(\mathcal{C}_j - \{i\}) c_{ij})}_{\text{Pr}(i \text{ awarded task from } j | i \text{ volunteers})}$$

Pr(Volunteer from \mathcal{C}_i | Advertisement from i)
Cooperator part — γ_i and Q_j vary with γ_i

where

$$b_i \triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k (b_i^k - c_i^k),$$

$$r_i \triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \pi_i^k (r_i^k - (b_i^k - c_i^k)),$$

$$p_i(Q_i) \triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \pi_i^k p_i^k(Q_i),$$

are the costs and benefits of local processing on $i \in \mathcal{V}$,

and

$$c_{ij} \triangleq \sum_{k \in \mathcal{Y}_j} \lambda_j^k \pi_j^k c_{ij}^k,$$

$$p_{ij}(Q_j) \triangleq \sum_{k \in \mathcal{Y}_j} \lambda_j^k \pi_j^k q_{ij}^k p_j^k(Q_j).$$

are the costs and benefits to $i \in \mathcal{C}$ for volunteering for tasks exported from $j \in \mathcal{V}_i$.

TPN version 1: Fictitious payment functions added as stabilizing controls.

Introduction

Task-processing
network

Cooperation Game

**Asynchronous
Convergence to
Cooperation**

Simulation Results

Closing Remarks

Asynchronous Convergence to Cooperation

Assume that (Payment and topological constraints):

Introduction

Task-processing
network

Cooperation Game

**Asynchronous
Convergence to
Cooperation**

Simulation Results

Closing Remarks

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

Assume that (Payment and topological constraints):

1. For all $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, p_{ij} is a stabilizing payment function.

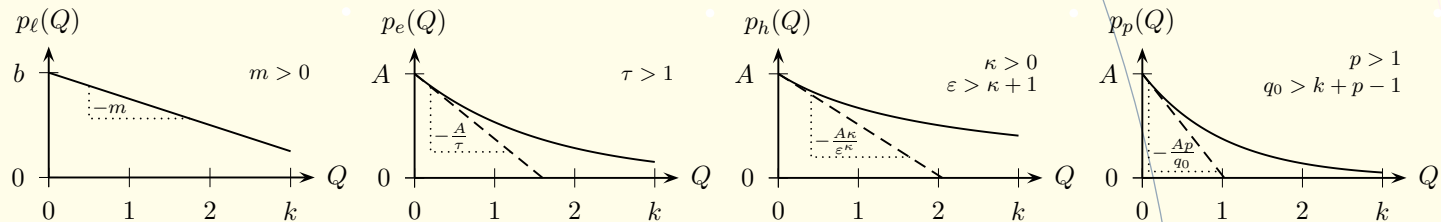


Figure 3: Sample stabilizing payment functions

Assume that (Payment and topological constraints):

1. For all $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, p_{ij} is a stabilizing payment function.
2. For all $j \in \mathcal{V}$, $|\mathcal{C}_j| \leq 3$ (i.e., no conveyor can have more than 3 outgoing links to cooperators).

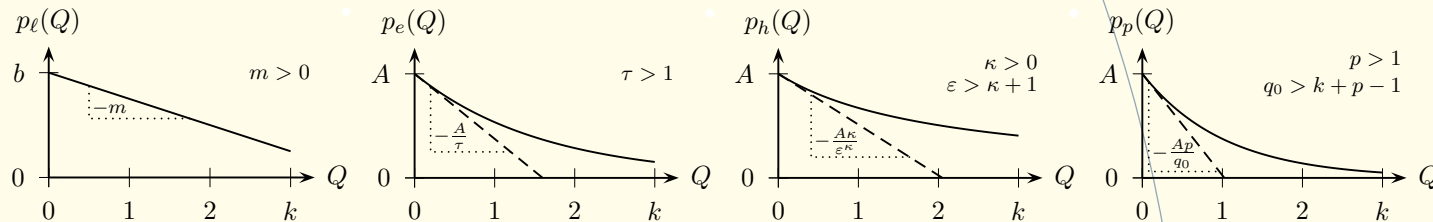


Figure 3: Sample stabilizing payment functions

Assume that (Payment and topological constraints):

1. For all $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, p_{ij} is a stabilizing payment function.
2. For all $j \in \mathcal{V}$, $|\mathcal{C}_j| \leq 3$ (i.e., no conveyor can have more than 3 outgoing links to cooperators).
3. For $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, if j is a 3-conveyor, then there must be some $k \in \mathcal{V}_i$ that is a 2-conveyor.

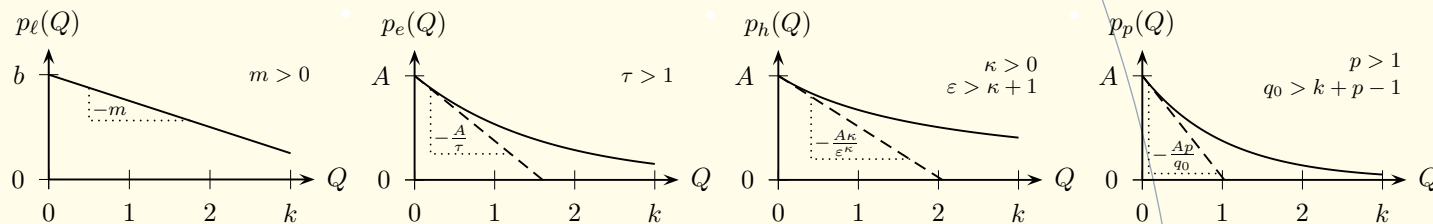


Figure 3: Sample stabilizing payment functions

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

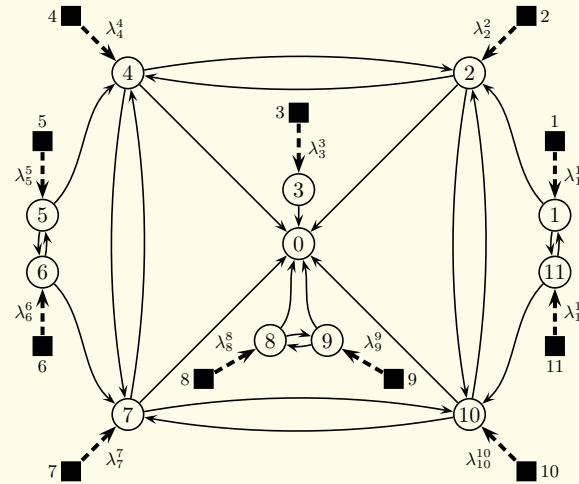


Figure 4: Rich yet stable task-processing network.

- “Pills” stabilize problematic areas by focussing attention.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

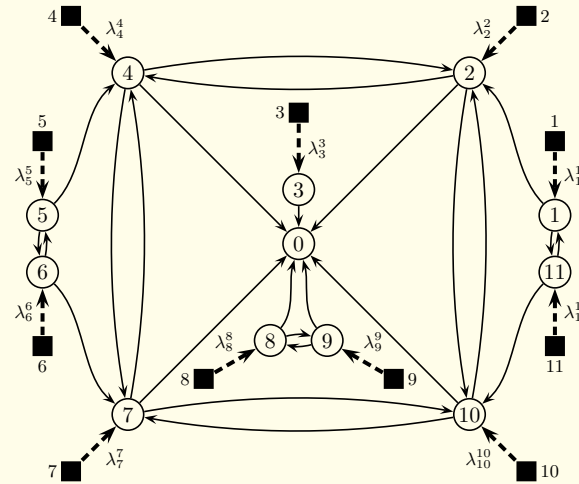


Figure 4: Rich yet stable task-processing network.

- “Pills” stabilize problematic areas by focussing attention.
- Future research direction : Stable network *motifs*.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

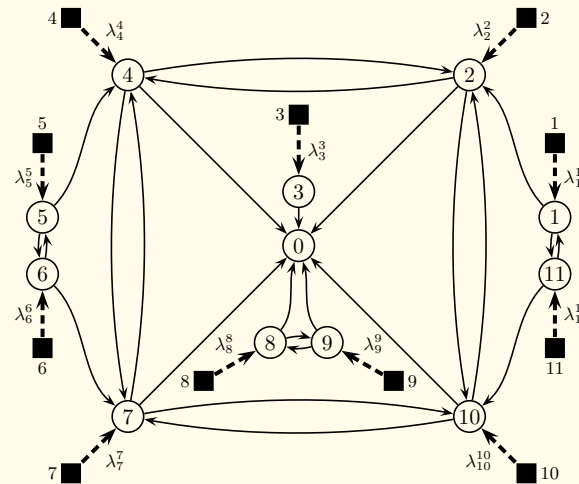


Figure 4: Rich yet stable task-processing network.

- “Pills” stabilize problematic areas by focussing attention.
- Future research direction (for someone else): Stable network *motifs*.

Totally Asynchronous Algorithm

Define $T : [0, 1]^n \mapsto [0, 1]^n$ by $T(\underline{\gamma}) \triangleq (T_1(\underline{\gamma}), T_2(\underline{\gamma}), \dots, T_n(\underline{\gamma}))$ where, for each $i \in \mathcal{C}$,

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\}$$

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

Totally Asynchronous Algorithm

Define $T : [0, 1]^n \mapsto [0, 1]^n$ by $T(\underline{\gamma}) \triangleq (T_1(\underline{\gamma}), T_2(\underline{\gamma}), \dots, T_n(\underline{\gamma}))$
where, for each $i \in \mathcal{C}$,

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\}$$

(i.e., gradient ascent)

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

Totally Asynchronous Algorithm

Define $T : [0, 1]^n \mapsto [0, 1]^n$ by $T(\underline{\gamma}) \triangleq (T_1(\underline{\gamma}), T_2(\underline{\gamma}), \dots, T_n(\underline{\gamma}))$ where, for each $i \in \mathcal{C}$,

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\}$$

(i.e., gradient ascent), where

$$\frac{1}{\sigma_i} \geq 2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)|$$

for all $\underline{\gamma} \in [0, 1]^n$.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

Totally Asynchronous Algorithm

Define $T : [0, 1]^n \mapsto [0, 1]^n$ by $T(\underline{\gamma}) \triangleq (T_1(\underline{\gamma}), T_2(\underline{\gamma}), \dots, T_n(\underline{\gamma}))$ where, for each $i \in \mathcal{C}$,

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\}$$

(i.e., gradient ascent), where

$$\frac{1}{\sigma_i} \geq 2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)|$$

for all $\underline{\gamma} \in [0, 1]^n$. If

$$\min_{j \in \mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)| > \left(|\mathcal{V}_i| - \frac{1}{2}\right) \max_{j \in \mathcal{V}_i} |c_{ij}|, \quad \text{for all } i \in \mathcal{C},$$

then the totally asynchronous distributed iteration (TADI) sequence $\{\underline{\gamma}(t)\}$ generated with mapping T and the outdated estimate sequence $\{\underline{\gamma}^i(t)\}$ for all $i \in \mathcal{C}$ each converge to the unique Nash equilibrium of the cooperation game.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

Totally Asynchronous Algorithm

Define $T : [0, 1]^n \mapsto [0, 1]^n$ by $T(\underline{\gamma}) \triangleq (T_1(\underline{\gamma}), T_2(\underline{\gamma}), \dots, T_n(\underline{\gamma}))$ where, for each $i \in \mathcal{C}$,

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\}$$

(i.e., gradient ascent), where

$$\frac{1}{\sigma_i} \geq 2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)|$$

for all $\underline{\gamma} \in [0, 1]^n$. If (\propto Hamilton's rule on networks)

$$\overbrace{\min_{j \in \mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)|}^{\text{Benefit}} > \overbrace{\left(|\mathcal{V}_i| - \frac{1}{2}\right)}^{\text{Relatedness}} \overbrace{\max_{j \in \mathcal{V}_i} |c_{ij}|}^{\text{Cost}}, \quad \text{for all } i \in \mathcal{C},$$

then the totally asynchronous distributed iteration (TADI) sequence $\{\underline{\gamma}(t)\}$ generated with mapping T and the outdated estimate sequence $\{\underline{\gamma}^i(t)\}$ for all $i \in \mathcal{C}$ each converge to the unique Nash equilibrium of the cooperation game.

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

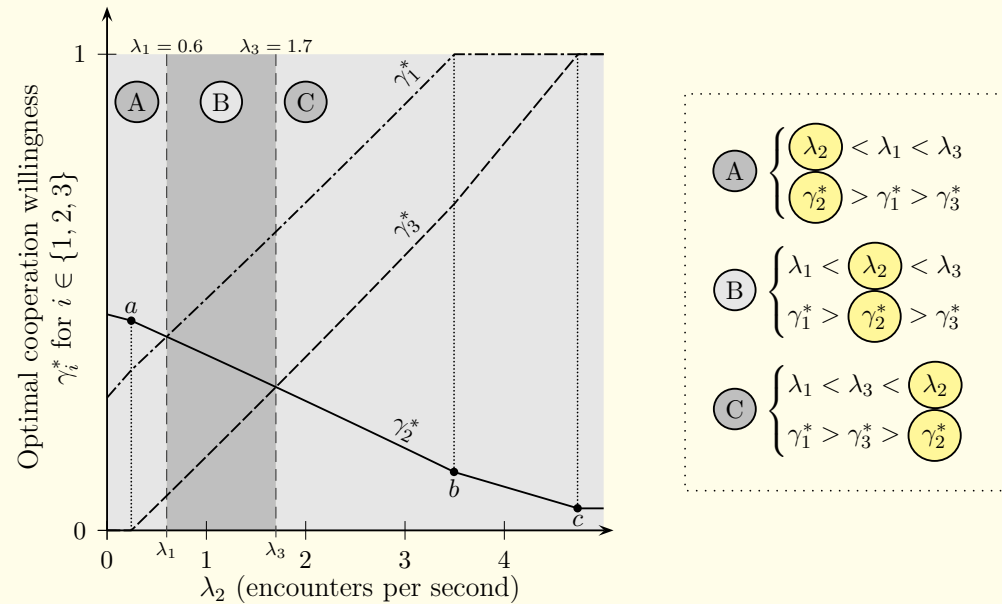


Figure 5: Simulation of AAV patrol scenario.

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

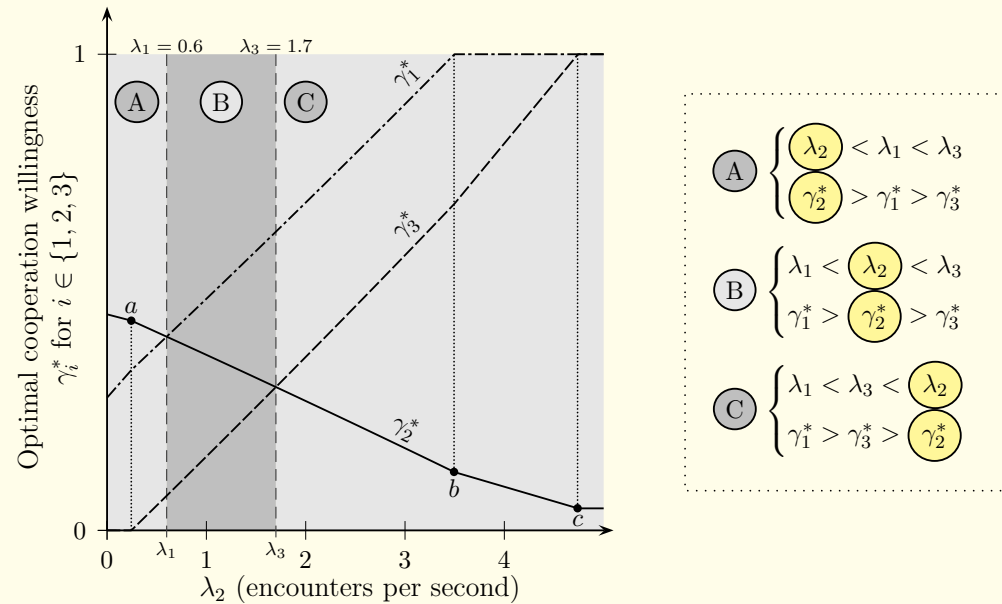


Figure 5: Simulation of AAV patrol scenario.

- Converges to predicted Nash equilibrium.

Introduction

Task-processing network

Cooperation Game

Asynchronous Convergence to Cooperation

Simulation Results

Closing Remarks

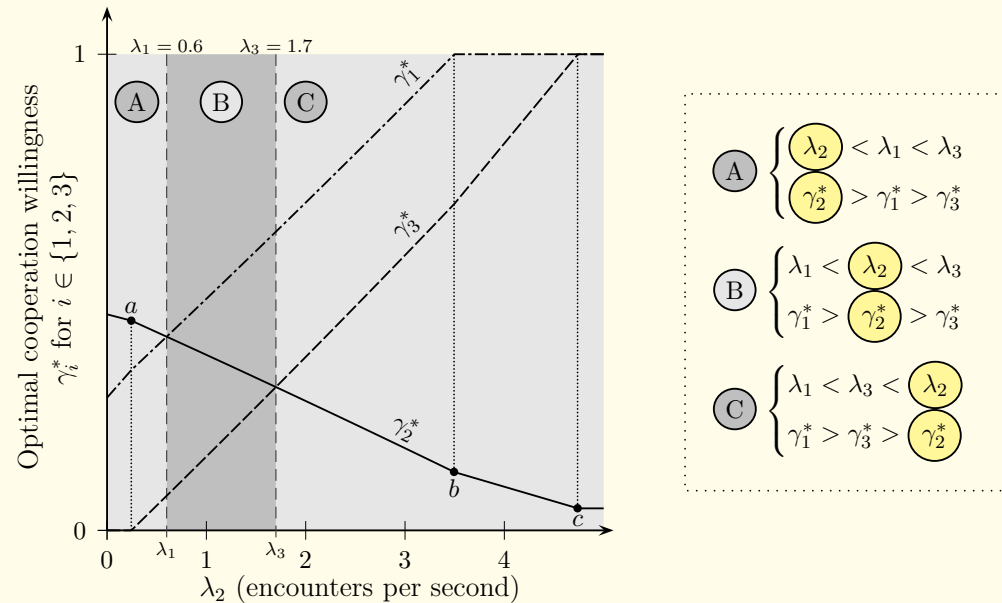


Figure 5: Simulation of AAV patrol scenario.

- Converges to predicted Nash equilibrium.
- Increases in one encounter rate (e.g., λ_2) cause equilibrium shift so neighbors (e.g., 1 and 3) help more and agent (e.g., 2) helps less.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Future directions:

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Future directions:

- ◆ Germ of a framework; lots more to generalize.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Future directions:

- ◆ Germ of a framework; lots more to generalize.
- ◆ Lots of information still needed to be broadcasted (or known a priori); an improvement would approximate individual gradients in a stable way.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

■ Future directions:

- ◆ Germ of a framework; lots more to generalize.
- ◆ Lots of information still needed to be broadcasted (or known a priori); an improvement would approximate individual gradients in a stable way.
- ◆ Does not capitalize on reciprocity. Reciprocal behaviors should not need payment for stability nor much communication.

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Future directions:
 - ◆ Germ of a framework; lots more to generalize.
 - ◆ Lots of information still needed to be broadcasted (or known a priori); an improvement would approximate individual gradients in a stable way.
 - ◆ Does not capitalize on reciprocity. Reciprocal behaviors should not need payment for stability nor much communication.
- Acknowledgments: Tom Waite, Ian Hamilton, Andrea Serrani, Jose B. Cruz, Jr., Atilla Eryilmaz, Bertsekas and Tsitsiklis

Introduction

Task-processing
network

Cooperation Game

Asynchronous
Convergence to
Cooperation

Simulation Results

Closing Remarks

- Future directions:
 - ◆ Germ of a framework; lots more to generalize.
 - ◆ Lots of information still needed to be broadcasted (or known a priori); an improvement would approximate individual gradients in a stable way.
 - ◆ Does not capitalize on reciprocity. Reciprocal behaviors should not need payment for stability nor much communication.

- Acknowledgments: Tom Waite, Ian Hamilton, Andrea Serrani, Jose B. Cruz, Jr., Atilla Eryilmaz, Bertsekas and Tsitsiklis

- **Thanks!** Questions?