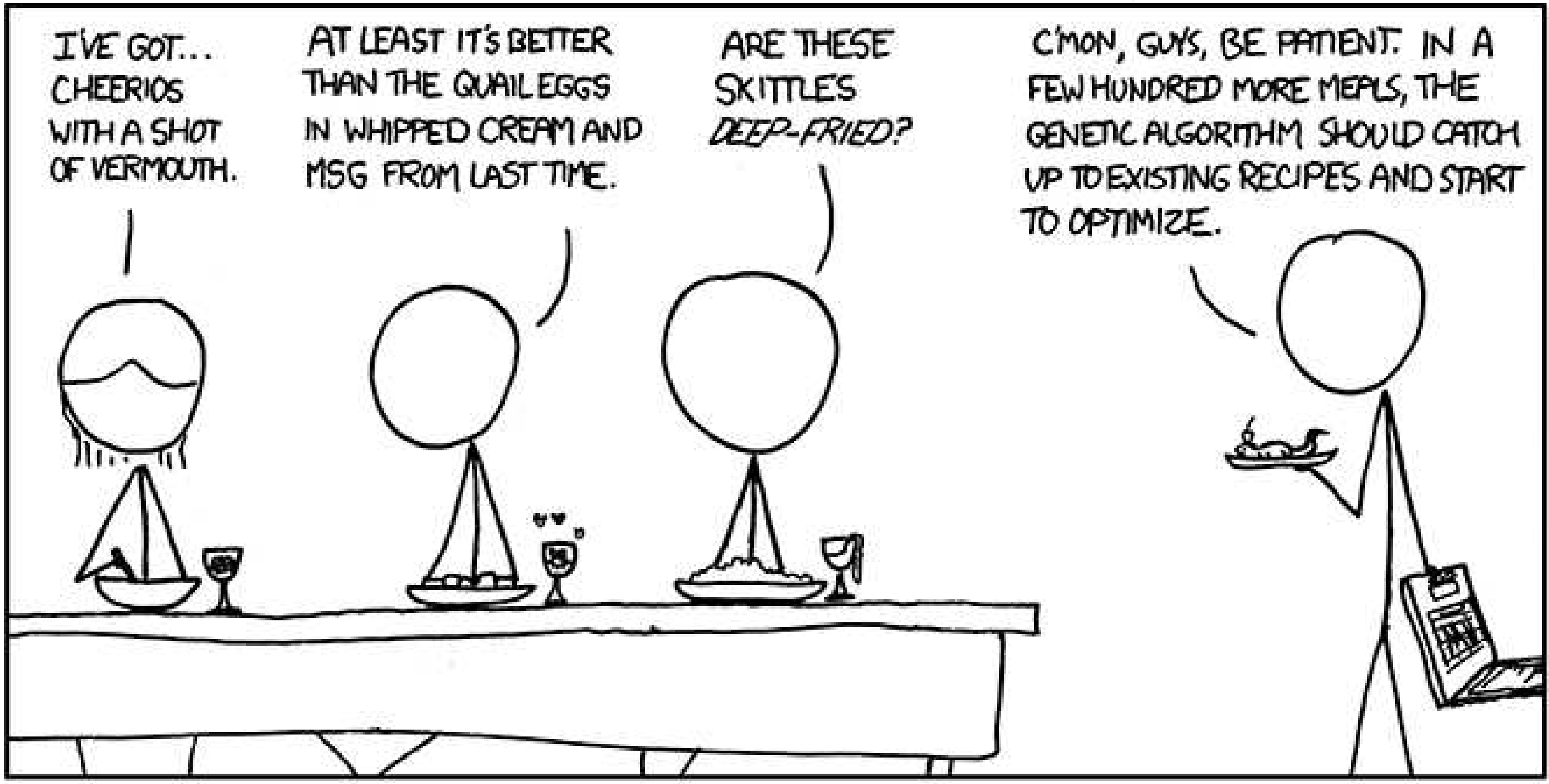
Pre-talk food for thought: computational biomimicry



WE'VE DECIDED TO DROP THE CS DEPARTMENT FROM OUR WEEKLY DINNER PARTY HOSTING ROTATION.[1]

---

[1] Compliments to XKCD: http://xkcd.com/720/.

Engineering Serendipity: Design and Analysis of Optimal Task-processing Agents

---

Presented in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Theodore (Ted) P. Pavlic, B.S., M.S. – The Ohio State University

Department of Electrical and Computer Engineering

Monday, August 9, 2010, 2:30 PM

Dissertation Committee: Dr. Kevin M. Passino (Advisor, ECE), Dr. Andrea Serrani (ECE),
Dr. Atilla Eryilmaz (ECE), Dr. David Blau (GS Rep., Economics)

Overview

Introduction

Solitary optimal task-processing agents in biology and engineering

Cooperative task processing

MultiIFD: Distributed gradient descent for constrained optimization

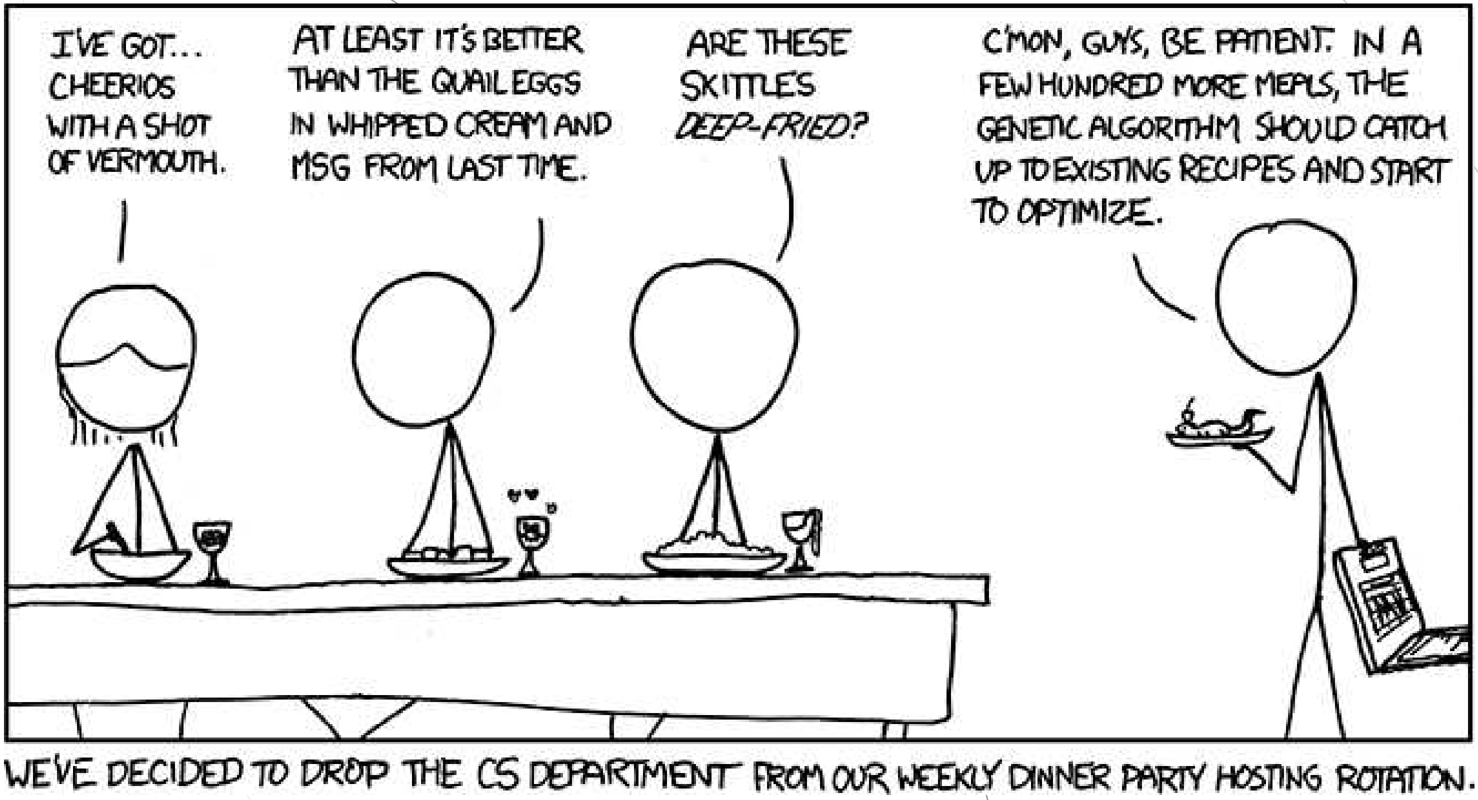
Closing remarks

Future directions*

---

*Omitted for brevity

Manufacturing Serendipity



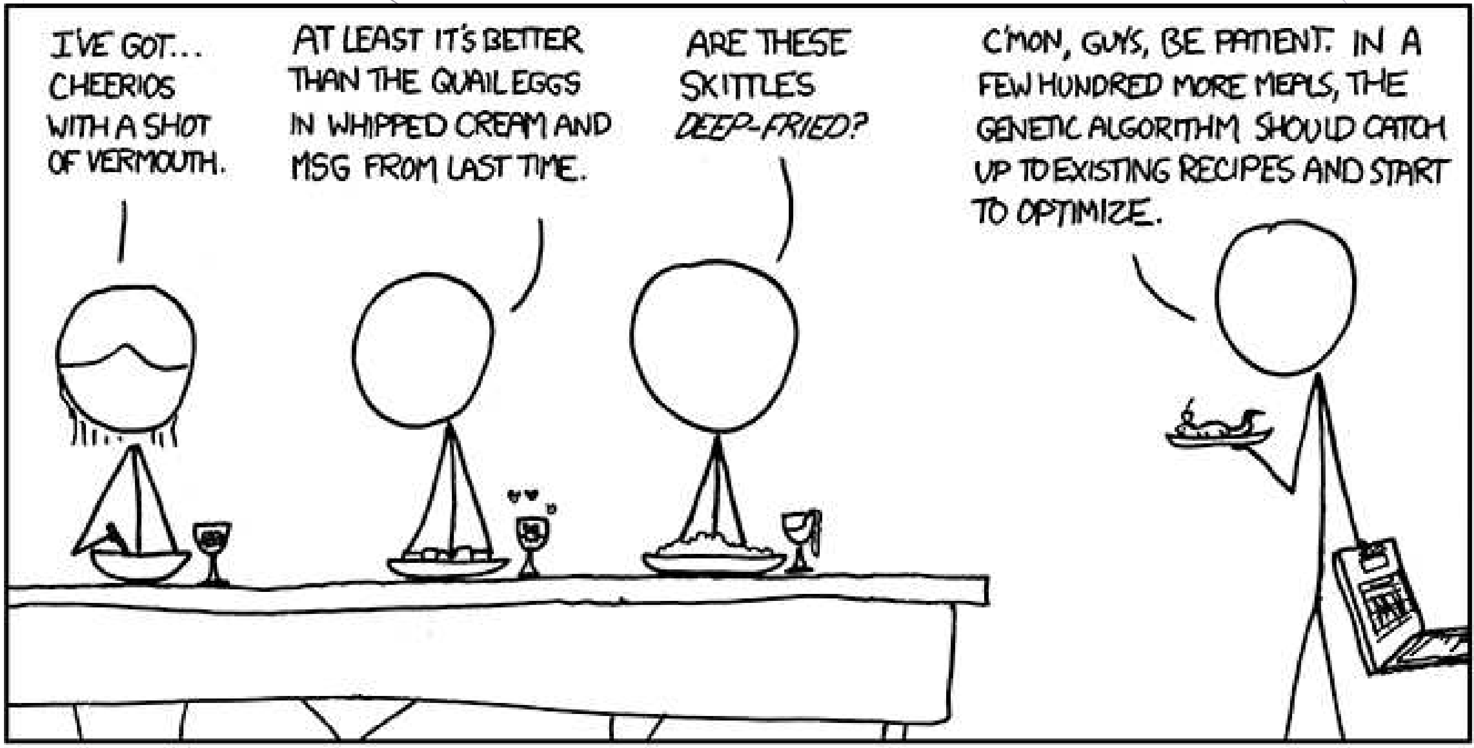
[1]

- Nakrani and Tovey (2007): honeybees and Internet server allocation

---

[1] Compliments to XKCD: http://xkcd.com/720/.

Manufacturing Serendipity



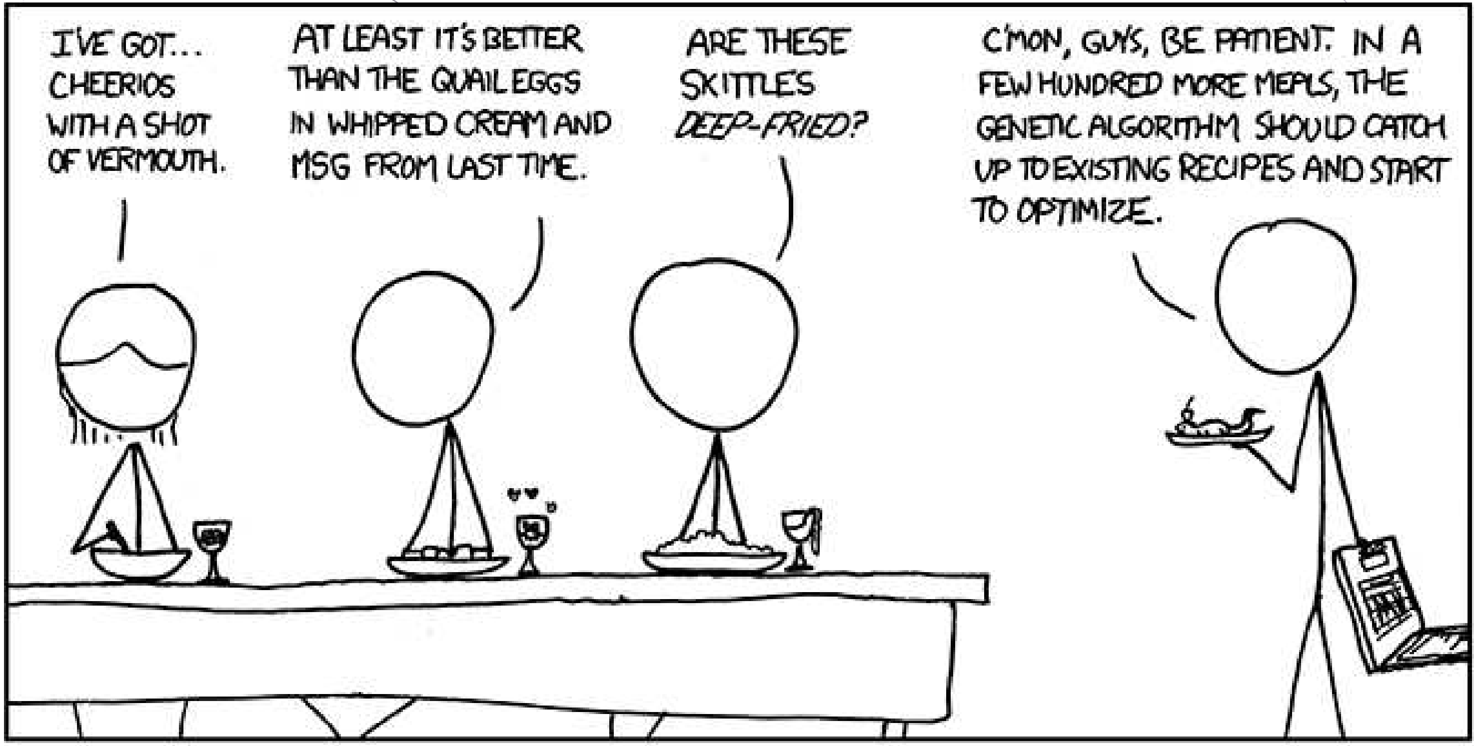
WE'VE DECIDED TO DROP THE CS DEPARTMENT FROM OUR WEEKLY DINNER PARTY HOSTING ROTATION.[1]

- Craig Tovey: "manufacture serendipity"

---

[1] Compliments to XKCD: http://xkcd.com/720/.

Engineering~~Manufacturing~~ Serendipity



WE'VE DECIDED TO DROP THE CS DEPARTMENT FROM OUR WEEKLY DINNER PARTY HOSTING ROTATION.[1]

- This dissertation: serendipity catalyst

---

[1] Compliments to XKCD: http://xkcd.com/720/.

Solitary optimal task-processing agents in biology and engineering

- Unified framework (Pavlic and Passino 2010c)
- Impulsiveness explained* (Pavlic and Passino 2010a)
- Optimal sunk-cost effect (Pavlic and Passino 2010b)

---

*Omitted for brevity

Foraging theory for autonomous vehicles

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

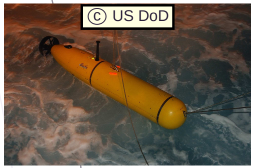
- Homomorphism: solitary foragers ↦ autonomous vehicles

Bobwhite quail (Gendron and Staddon 1983)

↧

MQ-8 Fire Scout (Northrop Grumman)

or



BPAUV (Bluefin)

Foraging theory for autonomous vehicles

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

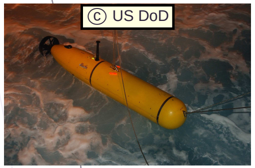
- Homomorphism: solitary foragers $\mapsto$ autonomous vehicles
  - ☐ Diversity of tasks (grasshoppers, enemy vehicles, probable underwater mines): $n \in \mathbb{N}$ types

Bobwhite quail (Gendron and Staddon 1983)

↧

MQ-8 Fire Scout (Northrop Grumman)

or



BPAUV (Bluefin)

Foraging theory for autonomous vehicles

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

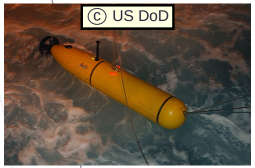
- Homomorphism: solitary foragers $\mapsto$ autonomous vehicles
  - ☐ Diversity of tasks (grasshoppers, enemy vehicles, probable underwater mines): $n \in \mathbb{N}$ types
  - ☐ Tasks of type $i \in \{1, 2, \dots, n\}$ have average value $g_i(\tau_i)$ for τ_i average time processing
    - Darwinian fitness surrogate (e.g., calories)
    - Economic value (e.g., dollars of profit)
    - Design preference (e.g., threat level)

Bobwhite quail (Gendron and Staddon 1983)

↧

MQ-8 Fire Scout (Northrop Grumman)

or



BPAUV (Bluefin)

Foraging theory for autonomous vehicles

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

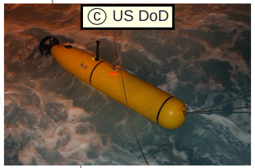
- Homomorphism: solitary foragers $\mapsto$ autonomous vehicles
  - ☐ Diversity of tasks (grasshoppers, enemy vehicles, probable underwater mines): $n \in \mathbb{N}$ types
  - ☐ Tasks of type $i \in \{1, 2, \ldots, n\}$ have average value $g_i(\tau_i)$ for τ_i average time processing
    - Darwinian fitness surrogate (e.g., calories)
    - Economic value (e.g., dollars of profit)
    - Design preference (e.g., threat level)
  - ☐ Opportunity cost: ignore some tasks

Bobwhite quail (Gendron and Staddon 1983)

↧

MQ-8 Fire Scout (Northrop Grumman)

or



BPAUV (Bluefin)

Foraging theory for autonomous vehicles
(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Homomorphism: solitary foragers ↦ autonomous vehicles
  - Diversity of tasks (grasshoppers, enemy vehicles, probable underwater mines): $n \in \mathbb{N}$ types
  - Tasks of type $i \in \{1, 2, \dots, n\}$ have average value $g_i(\tau_i)$ for τ_i average time processing
    - Darwinian fitness surrogate (e.g., calories)
    - Economic value (e.g., dollars of profit)
    - Design preference (e.g., threat level)
  - Opportunity cost: ignore some tasks
  - Rate maximization (MVT) for long runs
    - Prey model ↦ Task choice
    - Patch model ↦ Processing-time choice

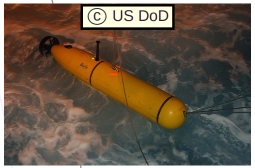

© Les Howard

Bobwhite quail (Gendron and Staddon 1983)

↧


MQ-8 Fire Scout (Northrop Grumman)

or

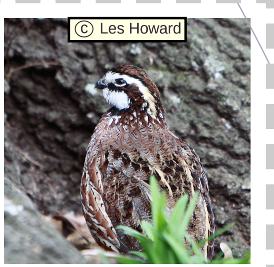

© US DoD

BPAUV (Bluefin)

Optimal task processing for generalized solitary agents

(Pavlic and Passino 2010c)

- Equivalence class: ~~Homomorphism:~~ solitary foragers $\Longleftrightarrow$ autonomous vehicles
  - ☐ Diversity of tasks (grasshoppers, enemy vehicles, probable underwater mines): $n \in \mathbb{N}$ types
  - ☐ Tasks of type $i \in \{1, 2, \ldots, n\}$ have average value $g_i(\tau_i)$ for τ_i average time processing
    - Darwinian fitness surrogate (e.g., calories)
    - Economic value (e.g., dollars of profit)
    - Design preference (e.g., threat level)
  - ☐ Opportunity cost: ignore some tasks
  - ☐ Rate maximization (MVT) for long runs
    - Prey model $\Longleftrightarrow$ Task choice
    - Patch model $\Longleftrightarrow$ Processing-time choice
  - ☐ More general statements available

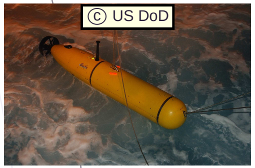

© Les Howard

Bobwhite quail
(Gendron and Staddon 1983)

⇕

MQ-8 Fire Scout
(Northrop Grumman)

⇕


© US DoD

BPAUV
(Bluefin)

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching
- Vehicle goes through i.i.d. cycles of searching and processing
  - ☐ $\bar{G}$: average per-encounter gain
  - ☐ $\bar{T}$: average per-encounter search and processing time
  - ☐ $\mathcal{G}(t)$: Markov renewal–reward process for accumulated gain

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching
- Long runtime $\implies$ maximize rate of return (i.e., gain ↑ & cycles ↑)

$$\operatorname*{aslim}_{t\to\infty} \frac{\mathcal{G}(t)}{t} = \frac{\bar{G}}{\bar{T}} = \frac{-c^s + \sum\limits_{i=1}^{n} \lambda_i p_i g_i}{1 + \sum\limits_{i=1}^{n} \lambda_i p_i \tau_i} \triangleq R(\vec{p})$$

Maximum rate $R(\vec{p}^*)$ is an *opportunity cost*; it represents the minimum gain from an activity to justify its use of time.

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching
- In general, $p_i \in [0,1]$, but

$$\frac{\partial R(\vec{p})}{\partial p_i} = \frac{\lambda_i g_i \left(1 + \sum\limits_{j=1}^{n} \lambda_j p_j \tau_j\right) - \lambda_i \tau_i \left(-c^s + \sum\limits_{j=1}^{n} \lambda_j p_j g_j\right)}{\left(1 + \sum\limits_{i=1}^{n} \lambda_i p_i \tau_i\right)^2}$$

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching
- So KKT reveals optimization is 2^n combinatorial:

$$\frac{\partial R(\vec{p})}{\partial p_i} = \frac{\boxed{\nabla_i = 0} \rightarrow \lambda_i g_i \left(1 + \sum_{\substack{j=1\\ j\neq i}}^{n} \lambda_j p_j \tau_j\right) - \lambda_i \tau_i \left(-c^s + \sum_{\substack{j=1\\ j\neq i}}^{n} \lambda_j p_j g_j\right)}{\left(1 + \sum_{i=1}^{n} \lambda_i p_i \tau_i\right)^2 \leftarrow \boxed{> 0}}$$

Property called the *zero–one rule* because $\exists \vec{p}^* : p_i^* \in \{0, 1\}$.

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching
- Classical prey *ranking* refines 2^n search to $n + 1$ search:

$$\overbrace{\frac{g_1}{\tau_1} > \frac{g_2}{\tau_2} > \cdots > \frac{g_{k^*}}{\tau_{k^*}}}^{\text{Processed types } (p_i^* = 1)} > \overbrace{\frac{-c^s + \sum_{i=1}^{k^*} \lambda_i g_i}{1 + \sum_{i=1}^{k^*} \lambda_i \tau_i}}^{\text{Optimal rate } R(\vec{p}^*)} > \overbrace{\frac{g_{k^*+1}}{\tau_{k^*+1}} > \cdots > \frac{g_n}{\tau_n}}^{\text{Ignored types } (p_i^* = 0)}$$

where optimal $p_i^* = [i \leq k^*]$ with $k^* \in \{0, 1, \ldots, n\}$.

Prey model for $n \in \mathbb{N}$ task types

(Andrews et al. 2004; Charnov 1973; Quijano et al. 2006; Stephens and Krebs 1986)

- Autonomous vehicle faces n-way merged Poisson process
  - ☐ λ_i: encounter rate for task of type i
  - ☐ $(g_i \triangleq g_i(\tau_i), \tau_i)$: mean (value, time) per type-i processing
  - ☐ p_i: probability that type-i task is processed (decision variable)
  - ☐ c^s: cost per-unit-time of searching
- Classical prey *ranking* refines 2^n search to $n+1$ search:

P

$$\frac{g_1}{\tau_1}$$

whe

Behavioral heuristic for encounter ℓ at time $t(\ell)$

$$p_{i(\ell)} = \left[\frac{g_{i(\ell)}}{\tau_{i(\ell)}} > \frac{\mathcal{G}(t(\ell))}{t(\ell)} \right] \qquad \text{(Iverson bracket)}$$

can calculate rate-maximizing prey choice in real time without sorting and searching (Pavlic and Passino 2010a).

Foreshadowing

Advantage-to-disadvantage optimization for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of tasks
  - ☐ $p_i \in [p_i^-, p_i^+] \subseteq [0, 1]$: decision variable
  - ☐ $\tau_i \in [\tau_i^-, \tau_i^+] \subseteq \overline{R}_{\geq 0}$: decision variable
  - ☐ $a_i, d_i : [\tau_i^-, \tau_i^+] \mapsto \mathbb{R}$: type i (dis)advantage $a_i(\tau_i)$ ($d_i(\tau_i)$)
  - ☐ $a, d \in \mathbb{R}$: background environmental (dis)advantage

Advantage-to-disadvantage optimization for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of tasks
  - ☐ $p_i \in [p_i^-, p_i^+] \subseteq [0,1]$: decision variable
  - ☐ $\tau_i \in [\tau_i^-, \tau_i^+] \subseteq \overline{R}_{\geq 0}$: decision variable
  - ☐ $a_i, d_i : [\tau_i^-, \tau_i^+] \mapsto \mathbb{R}$: type i (dis)advantage $a_i(\tau_i)$ ($d_i(\tau_i)$)
  - ☐ $a, d \in \mathbb{R}$: background environmental (dis)advantage
- Generalized advantage-to-disadvantage objective:

$$\text{maximize} \quad J(\vec{p}, \vec{\tau}) \triangleq \frac{a + \sum_{i=1}^{n} p_i a_i(\tau_i)}{d + \sum_{i=1}^{n} p_i d_i(\tau_i)}$$

Form generalized prey/patch algorithms for special $\{a, a_1, \ldots, a_n, d, d_1, \ldots, d_n\}$ cases.

Generalized prey model for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of tasks
  - $p_i \in [p_i^-, p_i^+] \subseteq [0,1]$: decision variable
  - $\tau_i \in [\tau_i^-, \tau_i^+] \subseteq \overline{R}_{\geq 0}$: decision variable
  - $a_i, d_i : [\tau_i^-, \tau_i^+] \mapsto \mathbb{R}$: type i (dis)advantage $a_i(\tau_i)$ ($d_i(\tau_i)$)
  - $a, d \in \mathbb{R}$: background environmental (dis)advantage
- Generalized prey algorithm: $d_i(\tau_i) \equiv d_i \neq 0$ non-zero constant

Optimal $\tau_i^* = \operatorname*{arg\,max}_{\tau_i \in [\tau_i^-, \tau_i^+]} \dfrac{a_i(\tau_i)}{d_i(\tau_i)}$ (*Max profitability*)

Optimal $p_i^* \in \{p_i^-, p_i^+\}$ (*Extreme-preference rule*)

for each type $i \in \{1, 2, \ldots, n\}$.

Generalized prey model for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of tasks
  - ☐ $p_i \in [p_i^-, p_i^+] \subseteq [0,1]$: decision variable
  - ☐ $\tau_i \in [\tau_i^-, \tau_i^+] \subseteq \overline{R}_{\geq 0}$: decision variable
  - ☐ $a_i, d_i : [\tau_i^-, \tau_i^+] \mapsto \mathbb{R}$: type i (dis)advantage $a_i(\tau_i)$ ($d_i(\tau_i)$)
  - ☐ $a, d \in \mathbb{R}$: background environmental (dis)advantage

- *Generalized profitability* ranking ($(n+1)$ search):

$$\overbrace{\frac{a_1(\tau_1^*)}{d_1(\tau_1^*)} > \cdots > \frac{a_{k^*}(\tau_{k^*}^*)}{d_{k^*}(\tau_{k^*}^*)}}^{\text{More-preferred types } (p_i^* = p_i^+)} > \overbrace{\frac{a + \sum\limits_{i=1}^{n} p_i^{k^*} a_i(\tau_i^*)}{d + \sum\limits_{i=1}^{n} p_i^{k^*} d_i(\tau_i^*)}}^{\text{Optimal } J(\vec{p}^*, \vec{\tau}^*)} > \overbrace{\frac{a_{k^*+1}(\tau_{k^*+1}^*)}{d_{k^*+1}(\tau_{k^*+1}^*)} > \cdots > \frac{a_n(\tau_n^*)}{d_n(\tau_n^*)}}^{\text{Less-preferred types } (p_i^* = p_i^-)}$$

where $p_i^k \triangleq [i \leq k]p_i^+ + [i > k]p_i^-$ and $k^* \in \{0, 1, \ldots, n\}$.

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of lumped tasks
  - $\tau_i^* = \tau_i^- = \tau_i^+$ for all $i \in \{1, 2, \ldots, n\}$
  - $[p_i^-, p_i^+] = [0, 1]$ for all $i \in \{1, 2, \ldots, n\}$

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of lumped tasks
  - ☐ $\tau_i^* = \tau_i^- = \tau_i^+$ for all $i \in \{1, 2, \dots, n\}$
  - ☐ $[p_i^-, p_i^+] = [0, 1]$ for all $i \in \{1, 2, \dots, n\}$
- Payload only supports $N \in \mathbb{N}$ tasks serviced
  - ☐ N packages (food, artillery) to deploy
  - ☐ N eggs to oviposit (e.g., parasitoid oviposition)

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of lumped tasks
  - $\tau_i^* = \tau_i^- = \tau_i^+$ for all $i \in \{1, 2, \dots, n\}$
  - $[p_i^-, p_i^+] = [0, 1]$ for all $i \in \{1, 2, \dots, n\}$
- Payload only supports $N \in \mathbb{N}$ tasks serviced
  - N packages (food, artillery) to deploy
  - N eggs to oviposit (e.g., parasitoid oviposition)
- Objective: Accumulate $G^T \in \mathbb{R}$ value by end-of-life
  - Threshold for mission to be considered success
  - Threshold for genes proliferation/survival to next foraging bout

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Generalized autonomous agent faces $n \in \mathbb{N}$ types of lumped tasks
  - ☐ $\tau_i^* = \tau_i^- = \tau_i^+$ for all $i \in \{1, 2, \dots, n\}$
  - ☐ $[p_i^-, p_i^+] = [0, 1]$ for all $i \in \{1, 2, \dots, n\}$
- Payload only supports $N \in \mathbb{N}$ tasks serviced
  - ☐ N packages (food, artillery) to deploy
  - ☐ N eggs to oviposit (e.g., parasitoid oviposition)
- Objective: Accumulate $G^T \in \mathbb{R}$ value by end-of-life
  - ☐ Threshold for mission to be considered success
  - ☐ Threshold for genes proliferation/survival to next foraging bout
- Rate maximization assumptions not valid for this case

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Static alternative to stochastic dynamic programming:

$$\text{maximize} \quad J(\vec{p},\vec{\tau}) \triangleq \frac{\mathrm{E}(\mathcal{G}(T^N)) - G^T}{\mathrm{E}(T^N)} \qquad \text{(“Excess rate”)}$$

$$= \frac{-c^s + \sum\limits_{i=1}^{n} \lambda_i p_i \left(g_i(\tau_i) - \frac{G^T}{N} \right)}{1 + \sum\limits_{i=1}^{n} \lambda_i p_i \tau_i}$$

where $T^N \triangleq$ (time after N^{th} processed task).

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

- Static alternative to stochastic dynamic programming:

$$\text{maximize} \quad J(\vec{p},\vec{\tau}) \triangleq \frac{\mathrm{E}(\mathcal{G}(T^N)) - G^T}{\mathrm{E}(T^N)} \qquad \text{(“Excess rate”)}$$

$$= \frac{-c^s + \sum\limits_{i=1}^{n} \lambda_i p_i \left(g_i(\tau_i) - \frac{G^T}{N} \right)}{1 + \sum\limits_{i=1}^{n} \lambda_i p_i \tau_i}$$

where $T^N \triangleq$ (time after N^{th} processed task).

- Generalized profitability for $i \in \{1, 2, \ldots, n\}$:

$$\frac{a_i(\tau_i^*)}{d_i(\tau_i^*)} \triangleq \frac{g_i(\tau_i^*) - \frac{G^T}{N}}{\tau_i^*}$$

Ranking depends on success threshold G^T (matches SDP).

$N \in \mathbb{N}$ event finite-lifetime case for $n \in \mathbb{N}$ task types

(Pavlic and Passino 2010c)

$n = 5$ task types, $N = 300$ tasks per mission, 100 Monte Carlo samples (mean ± SEM)

		Take all	Classical	Excess	estClassical	estExcess
$G^T = 9000$	$\mathcal{G}(T^N)$:	16565 ± 30	10946 ± 16	20473 ± 25	11218 ± 128	18119 ± 38
	$\geq G^T$:	100%	100%	100%	98%	100%
	T^N:	11119 ± 42	4391 ± 8	9227 ± 13	4567 ± 63	11668 ± 43
$G^T = 13500$	$\mathcal{G}(T^N)$:	16642 ± 33	10958 ± 16	25153 ± 11	11270 ± 103	18647 ± 44
	$\geq G^T$**:**	**100%**	**0%**	**100%**	**5%**	**100%**
	T^N:	11158 ± 38	4393 ± 8	15645 ± 42	4586 ± 50	12779 ± 46
$G^T = 16500$	$\mathcal{G}(T^N)$:	16546 ± 34	10993 ± 16	25141 ± 14	10965 ± 91	18796 ± 39
	$\geq G^T$**:**	**55%**	**0%**	**100%**	**0%**	**100%**
	T^N:	11092 ± 40	4421 ± 8	15605 ± 53	4440 ± 43	13120 ± 44

$(\lambda_1, g_1, \tau_1) = (0.5, 30, 10)$, $(\lambda_2, g_2, \tau_2) = (0.25, 50, 20)$, $(\lambda_3, g_3, \tau_3) = (0.4, 80, 35)$, $(\lambda_4, g_4, \tau_4) = (0.1, 100, 110)$, $(\lambda_5, g_5, \tau_5) = (0.8, 55, 50)$, $c^s = 0.1$

- Take high gain only: 29700 (36000 **time**)
- Take high profitability only: 8940 (3600 time)
- Take high excess profitability only: 23925 (11250 **time**)

Ecological rationality: operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

---

*Omitted for brevity

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Laboratory impulsiveness (Ainslie 1974; Bateson and Kacelnik 1996; Bradshaw and Szabadi 1992; Green et al. 1981; McDiarmid and Rilling 1965; Rachlin and Green 1972; Siegel and Rachlin 1995; Snyderman 1983; Stephens and Anderson 2001)

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Laboratory impulsiveness
  - ☐ Using starvation, animals are trained to use a Skinner box
  - ☐ Repeat mutually exclusive binary-choice trials (at low weight)

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Laboratory impulsiveness
  - ☐ Using starvation, animals are trained to use a Skinner box
  - ☐ Repeat mutually exclusive binary-choice trials (at low weight)
- What can be inferred about Skinner box results?
  - ☐ Violates assumption that simultaneous encounters occur with probability zero (Poisson assumption)
  - ☐ Mutually exclusive choice unlikely when prey is immobile
  - ☐ Impulsiveness vanishes for patch decision (Stephens et al. 2004)
  - ☐ Attention (Monterosso and Ainslie 1999; Siegel and Rachlin 1995)

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Laboratory impulsiveness
  - Using starvation, animals are trained to use a Skinner box
  - Repeat mutually exclusive binary-choice trials (at low weight)
- What can be inferred about Skinner box results?
  - Violates assumption that simultaneous encounters occur with probability zero (Poisson assumption)
  - Mutually exclusive choice unlikely when prey is immobile
  - Impulsiveness vanishes for patch decision (Stephens et al. 2004)
  - Attention (Monterosso and Ainslie 1999; Siegel and Rachlin 1995)
- Skinner trials are worst-case scenario for a robot
  - Predisposes robots to underestimate (adds suboptimal eq.)

Ecological rationality: Operant laboratory impulsiveness*

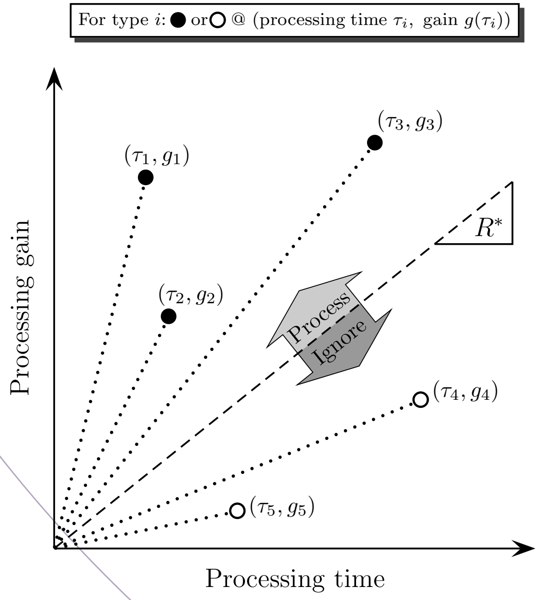
(Pavlic and Passino 2010a)

- Laboratory impulsiveness
  - ☐ Using starvation, animals are trained to use a Skinner box
  - ☐ Repeat mutually exclusive binary-choice trials (at low weight)
- What can be inferred about Skinner box results?
  - ☐ Violates assumption that simultaneous encounters occur with probability zero (Poisson assumption)
  - ☐ Mutually exclusive choice unlikely when prey is immobile
  - ☐ Impulsiveness vanishes for patch decision (Stephens et al. 2004)
  - ☐ Attention (Monterosso and Ainslie 1999; Siegel and Rachlin 1995)
- Skinner trials are worst-case scenario for an animal?
  - ☐ Predisposes animals to underestimate? (adds suboptimal eq.)

Ecological rationality: Operant laboratory impulsiveness*

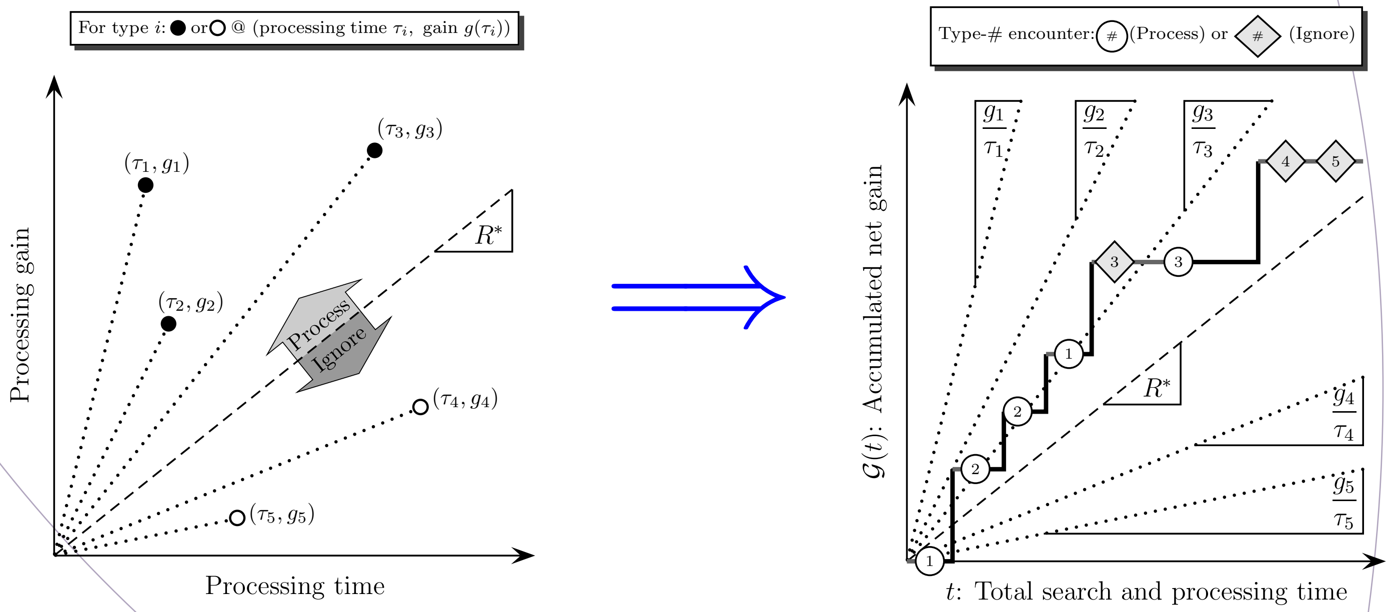
(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:



Ecological rationality: Operant laboratory impulsiveness*
(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:

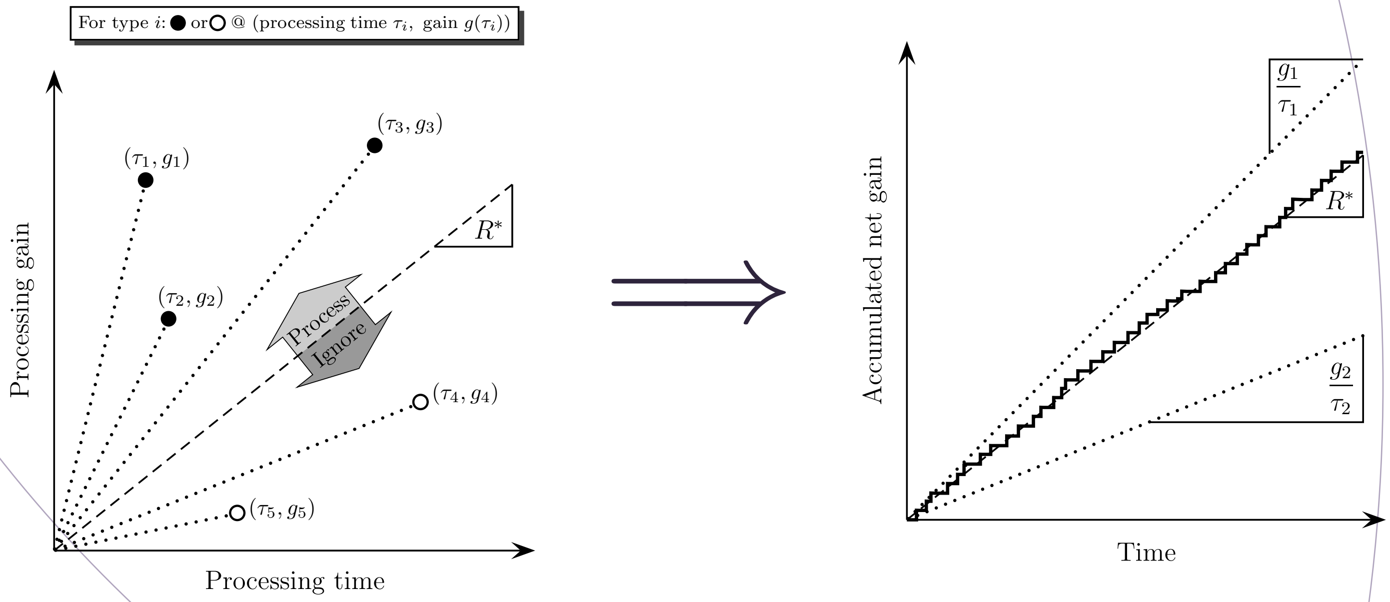


Process encounter k *when* $g_{i(k)}/\tau_{i(k)} > \mathcal{G}(t(k))/t(k)$

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:



Process encounter k *when* $g_{i(k)}/\tau_{i(k)} > \mathcal{G}(t(k))/t(k)$

- Rule (even with mistakes) is optimal facing Poisson encounters (i.e., simultaneous w.p.0)

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:

☐ Digestive rate constraints (b_i: prey bulk) (Hirakawa 1995):

$$\frac{\sum\limits_{i=1}^{n} \lambda_i p_i b_i}{1 + \sum\limits_{i=1}^{n} \lambda_i p_i \tau_i} \le B \quad \overset{\text{KKT}}{\Longrightarrow} \quad \begin{array}{r} p_1^* = 1 \\ \vdots \\ p_{k^*-1}^* = 1 \\ p_{k^*}^* \in [0, 1] \end{array}$$

Partial Preferences (rank by g_i/b_i)

Digression

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:

☐ Ecological–physiological hybrid method (Whelan and Brown 2005):

$$\text{Asymptotic gut constraint} \quad \Longleftrightarrow \quad \text{Rank by } \frac{g_i}{\tau_i + \tau_i^b}$$

Digression

Ecological rationality: Operant laboratory impulsiveness*

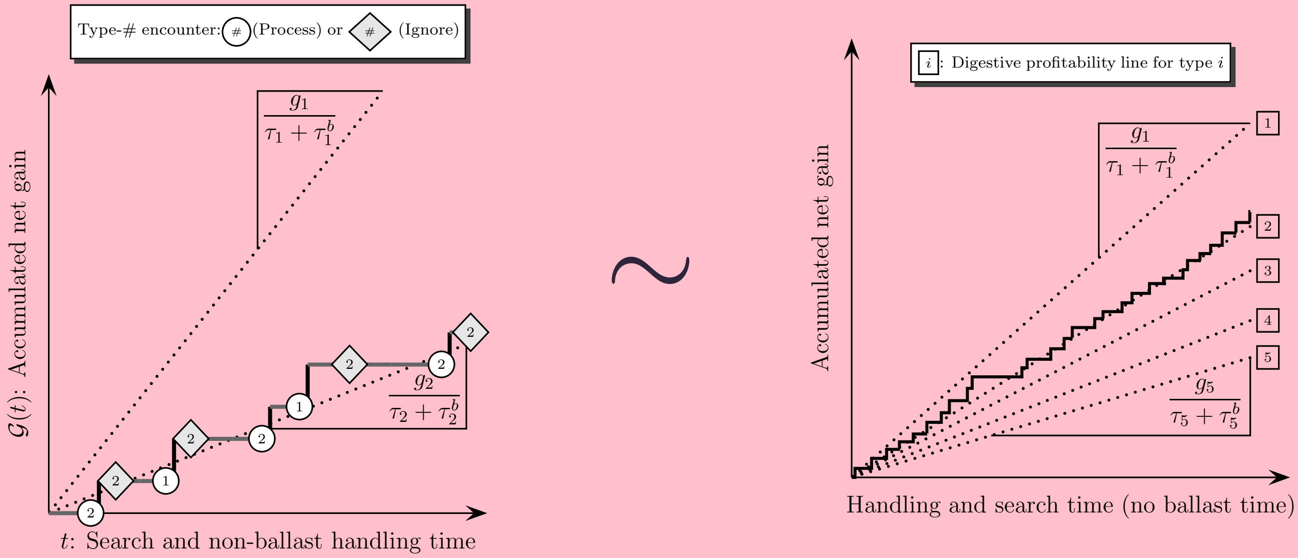
(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:

☐ Ecological–physiological hybrid method (Whelan and Brown 2005):

$$\text{Asymptotic gut constraint} \iff \text{Rank by } \frac{g_i}{\tau_i + \tau_i^b}$$

☐ *Process encounter k when $g_{i(k)}/(\tau_{i(k)} + \tau_{i(k)}^b) > \mathcal{G}(t(k))/t(k)$*



Digression

Ecological rationality: Operant laboratory impulsiveness*

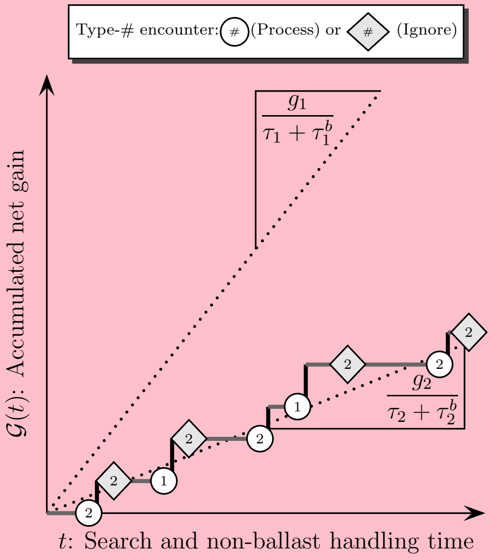
(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:

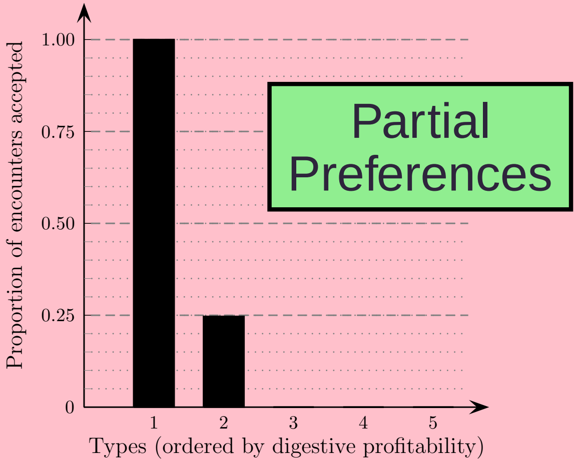
☐ Ecological–physiological hybrid method (Whelan and Brown 2005):

$$\text{Asymptotic gut constraint} \quad \iff \quad \text{Rank by } \frac{g_i}{\tau_i + \tau_i^b}$$

☐ *Process encounter* k *when* $g_{i(k)}/(\tau_{i(k)} + \tau_{i(k)}^b) > \mathcal{G}(t(k))/t(k)$



$\sim$



Partial Preferences

Digression

Ecological rationality: Operant laboratory impulsiveness*

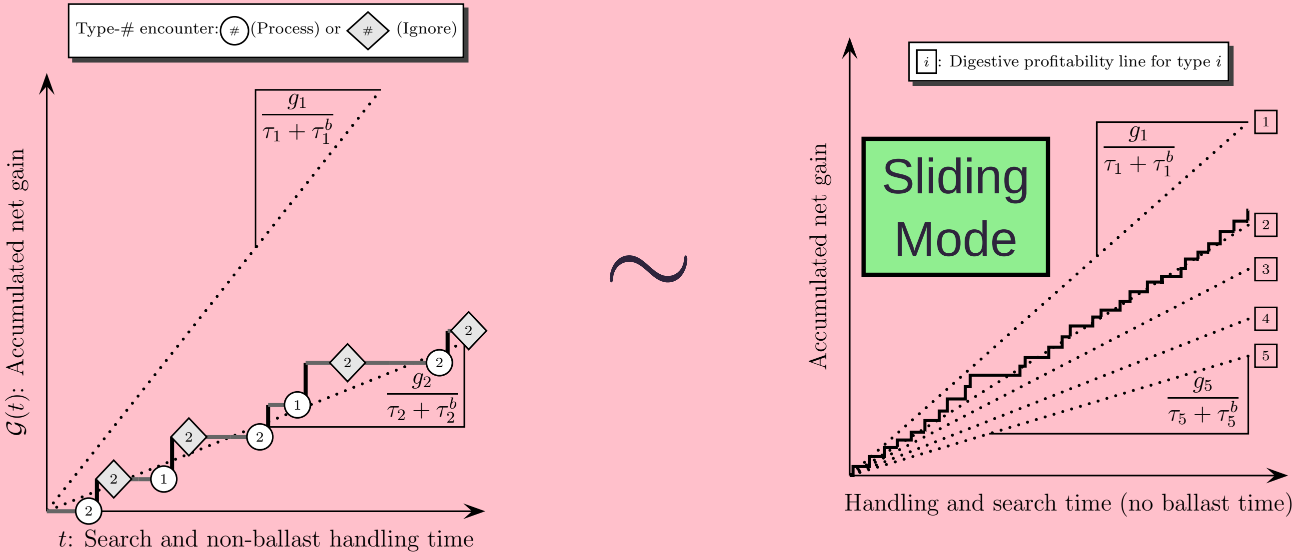
(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:

☐ Ecological–physiological hybrid method (Whelan and Brown 2005):

$$\text{Asymptotic gut constraint} \iff \text{Rank by } \frac{g_i}{\tau_i + \tau_i^b}$$

☐ *Process encounter k when $g_{i(k)}/(\tau_{i(k)} + \tau_{i(k)}^b) > \mathcal{G}(t(k))/t(k)$*

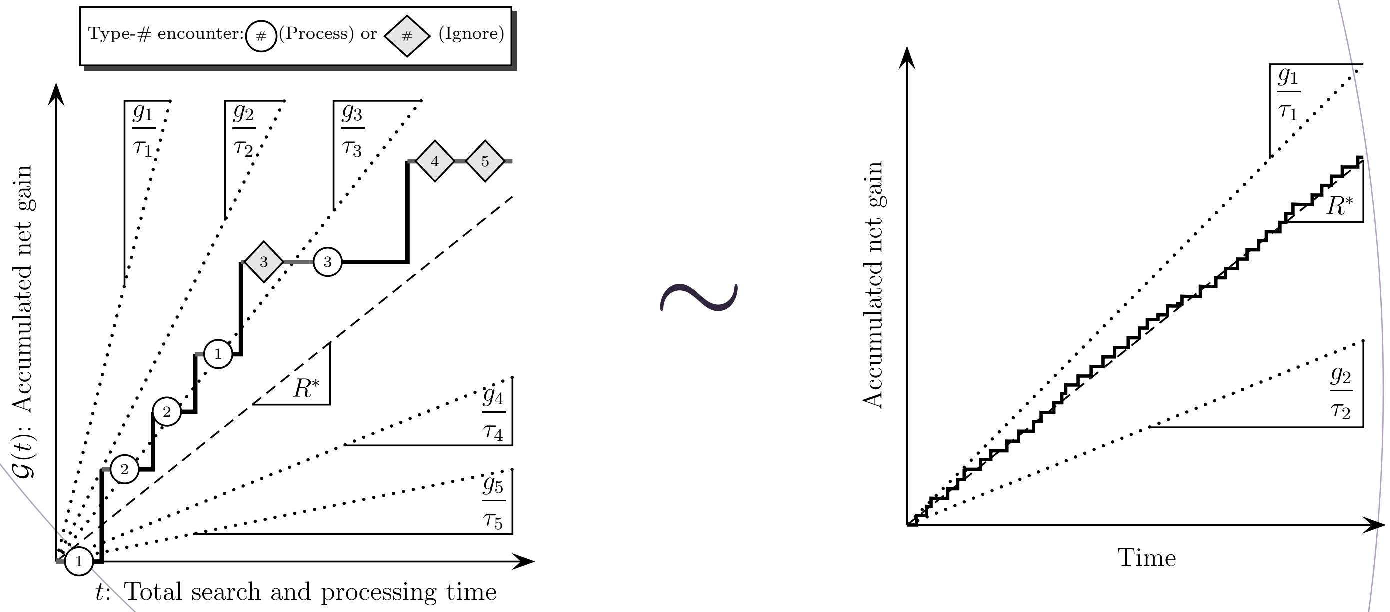


Digression

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:



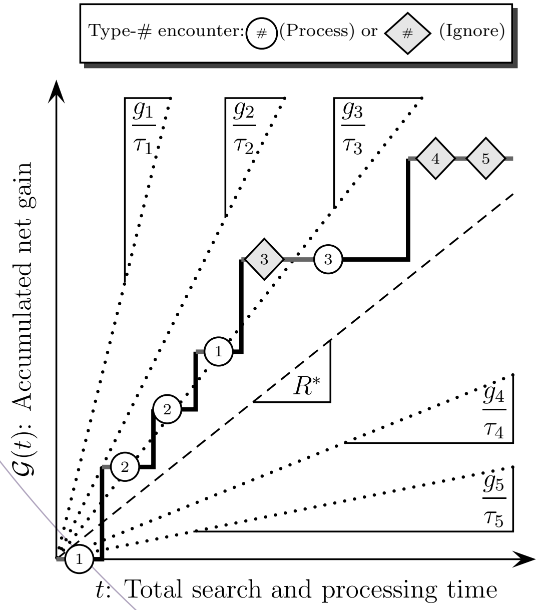
Process encounter k *when* $g_{i(k)}/\tau_{i(k)} > \mathcal{G}(t(k))/t(k)$

- Attention: simultaneous encounter (w.p.0) $\implies$ low time first

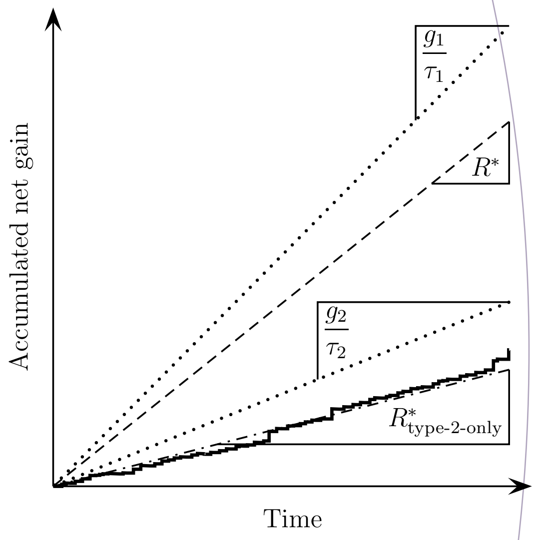
Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:



$\sim$



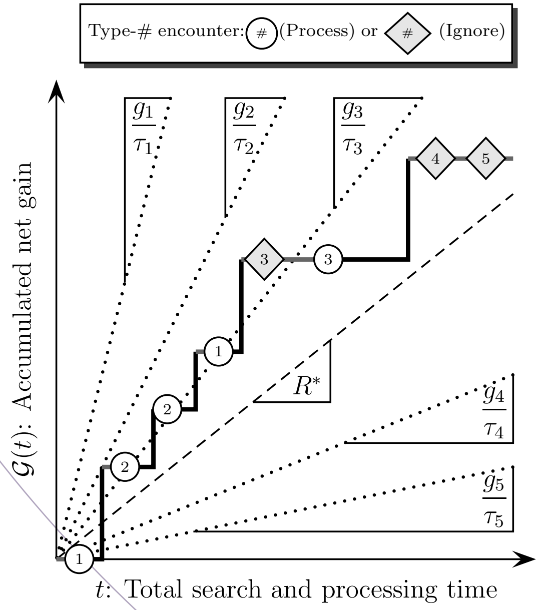
Process encounter k *when* $g_{i(k)}/\tau_{i(k)} > \mathcal{G}(t(k))/t(k)$

- Attention: simultaneous encounter (w.p.1) $\Longrightarrow$ low time first

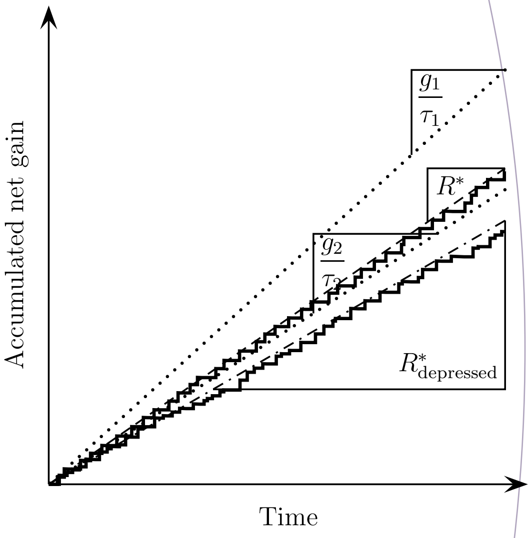
Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:



$\sim$



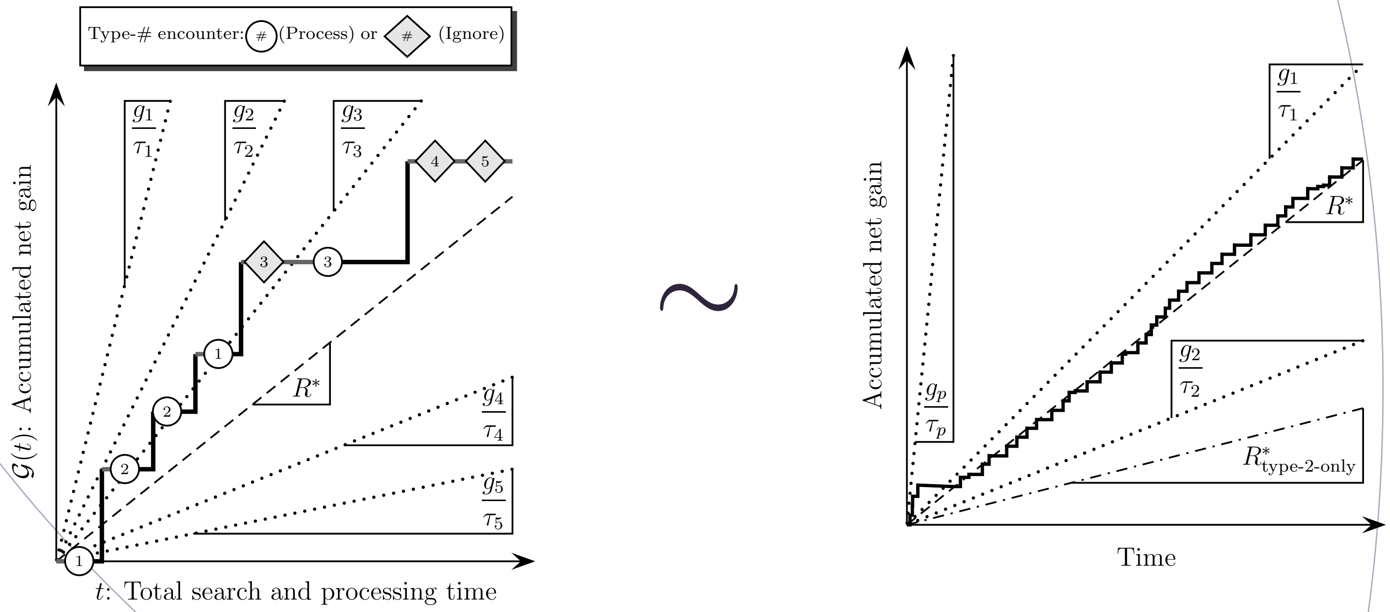
Process encounter k *when* $g_{i(k)}/\tau_{i(k)} > \mathcal{G}(t(k))/t(k)$

- Attention: simultaneous encounter (w.p.1) $\Longrightarrow$ either first
  - ☐ Lucky runs accumulate high initial estimate
    - Lucky forager specializes; unlucky forager generalizes

Ecological rationality: Operant laboratory impulsiveness*

(Pavlic and Passino 2010a)

- Graphical description of optimal prey choice:



Process encounter k when $g_{i(k)}/\tau_{i(k)} > \mathcal{G}(t(k))/t(k)$

- Attention: simultaneous encounter (w.p.1) $\Longrightarrow$ low time first
  - ☐ Rescue optimality with early *ad libitum* feeding

Ecological rationality: sunk-cost effects and long patch residence times

(Pavlic and Passino 2010b)

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
  - In shallow water, swans feeding on tubers can “head dip”
  - In deep water, they must “up end,” which requires more energy
  - Nolet et al. find no theoretical justification for longer times at the more energetic tasks

Ⓒ Gemma Longman

Ⓒ Derek Tsang

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
    - ☐ In shallow water, swans feeding on tubers can “head dip”
    - ☐ In deep water, they must “up end,” which requires more energy
    - ☐ Nolet et al. find no theoretical justification for longer times at the more energetic tasks
    - ☐ Other sunk cost/Concorde effects (Arkes and Blumer 1985; Arkes and Ayton 1999; Dawkins and Carlisle 1976; Kanodia et al. 1989; Staw 1981)

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

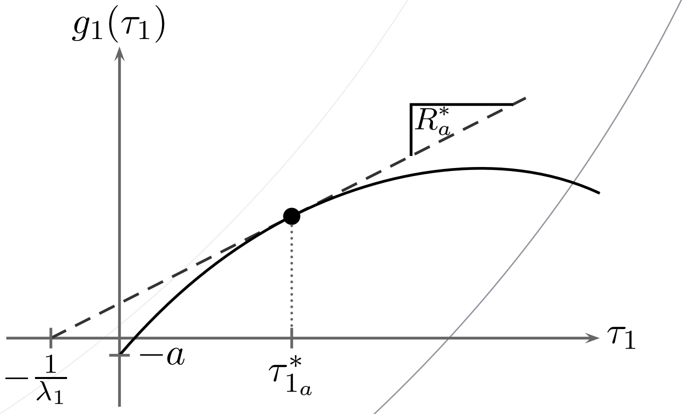
- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
  - ☐ In shallow water, swans feeding on tubers can "head dip"
  - ☐ In deep water, they must "up end," which requires more energy
  - ☐ Nolet et al. find no theoretical justification for longer times at the more energetic tasks
  - ☐ Other sunk cost/Concorde effects (Arkes and Blumer 1985; Arkes and Ayton 1999; Dawkins and Carlisle 1976; Kanodia et al. 1989; Staw 1981)
- Sunk-cost observations are consistent with rate maximization when patch entry costs are modeled (unconventional).

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
- Sunk-cost observations are consistent with rate maximization when patch entry costs are modeled (unconventional). For $n = 1$,

$$R(\tau_1) = \frac{g_1(\tau_1)}{\frac{1}{\lambda_1} + \tau_1} \quad \text{where} \quad \{a < b < c\} \triangleq |g_1(0) < 0|$$

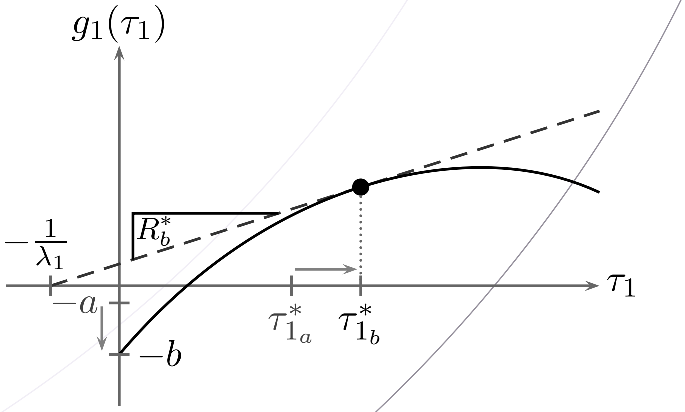


Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
- Sunk-cost observations are consistent with rate maximization when patch entry costs are modeled (unconventional). For $n = 1$,

$$R(\tau_1) = \frac{g_1(\tau_1)}{\frac{1}{\lambda_1} + \tau_1} \quad \text{where} \quad \{a < b < c\} \triangleq |g_1(0) < 0|$$



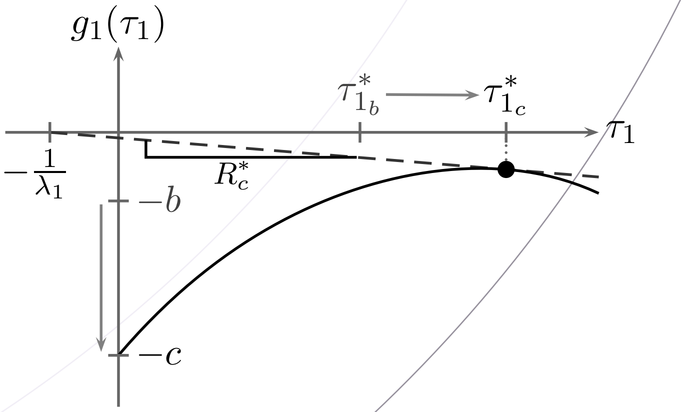
Due to entry costs, searching is a less desirable task.

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
- Sunk-cost observations are consistent with rate maximization when patch entry costs are modeled (unconventional). For $n = 1$,

$$R(\tau_1) = \frac{g_1(\tau_1)}{\frac{1}{\lambda_1} + \tau_1} \quad \text{where} \quad \{a < b < c\} \triangleq |g_1(0) < 0|$$



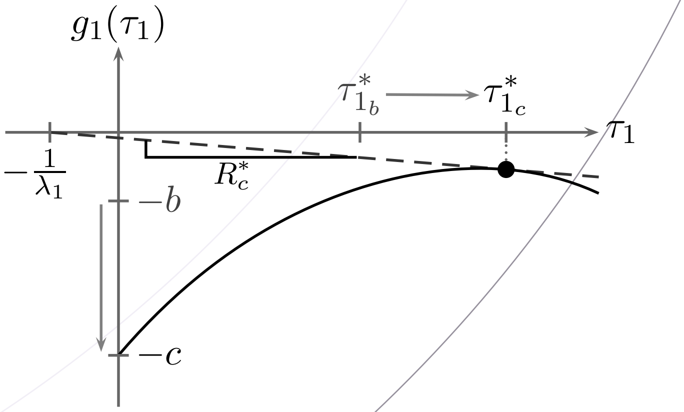
Due to entry costs, searching is a less desirable task.

Ecological rationality: Sunk costs and long patch times

(Pavlic and Passino 2010b)

- Nolet et al. (2001) are unable to explain spatial differences in tundra swan foraging
- Sunk-cost observations are consistent with rate maximization when patch entry costs are modeled (unconventional). For $n = 1$,

$$R(\tau_1) = \frac{g_1(\tau_1)}{\frac{1}{\lambda_1} + \tau_1} \quad \text{where} \quad \{a < b < c\} \triangleq |g_1(0) < 0|$$



Due to entry costs, searching is a less desirable task.

- May explain some overstaying as well (Nonacs 2001)

Distributed task processing

- Cooperative task processing (Pavlic and Passino 2010d)
    - Separable constraints (Cartesian product)
    - Parallel Nash equilibrium solver
- MultiIFD constrained gradient descent
    - Polyhedral constraint set
    - Distributed Pareto equilibrium solver

Cooperative task processing

(Pavlic and Passino 2010d)

Cute slide

ⓒ Josh Laverty

ⓒ Rennett Stowe

ⓒ David Blaikie

Example TPN: Cooperative breeding~~Cute slide~~

Ⓒ Josh Laverty

Ⓒ Rennett Stowe

Ⓒ David Blaikie

Cooperative breeding
(Hamilton and Taborsky 2005)

© Josh Laverty

© Rennett Stowe

© Chris J. Fry

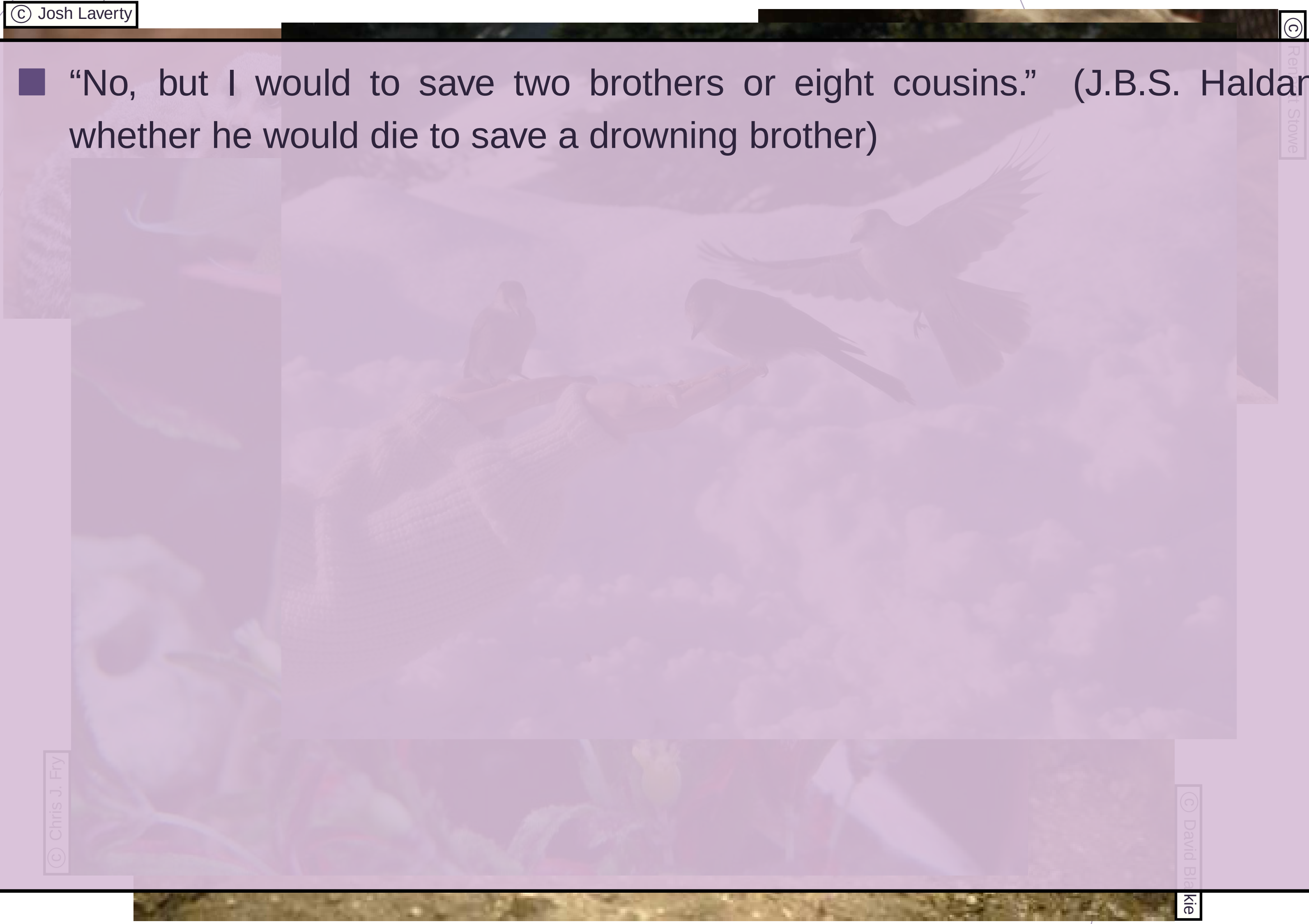
© David Blaikie

Cooperative breeding

(Waite and Strickland 1997)

Evolution of cooperation

(Axelrod 1984; Hamilton 1964; Nowak 2006; Ohtsuki et al. 2006; Trivers 1971)



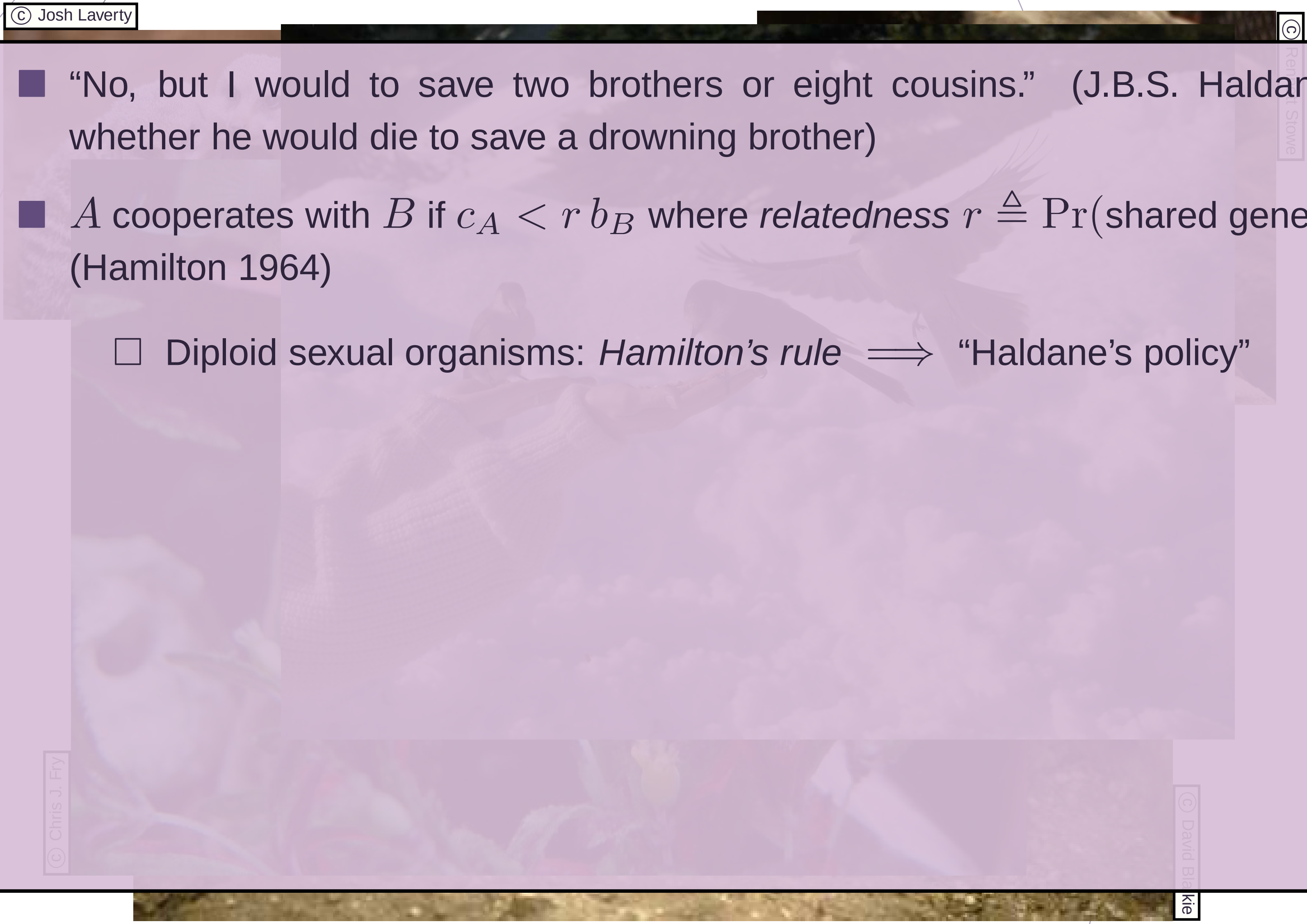
© Josh Laverty

© Chris J. Fry

- “No, but I would to save two brothers or eight cousins.” (J.B.S. Haldane; whether he would die to save a drowning brother)

Evolution of cooperation

(Axelrod 1984; Hamilton 1964; Nowak 2006; Ohtsuki et al. 2006; Trivers 1971)



ⓒ Josh Laverty

ⓒ Rene   t Stowe

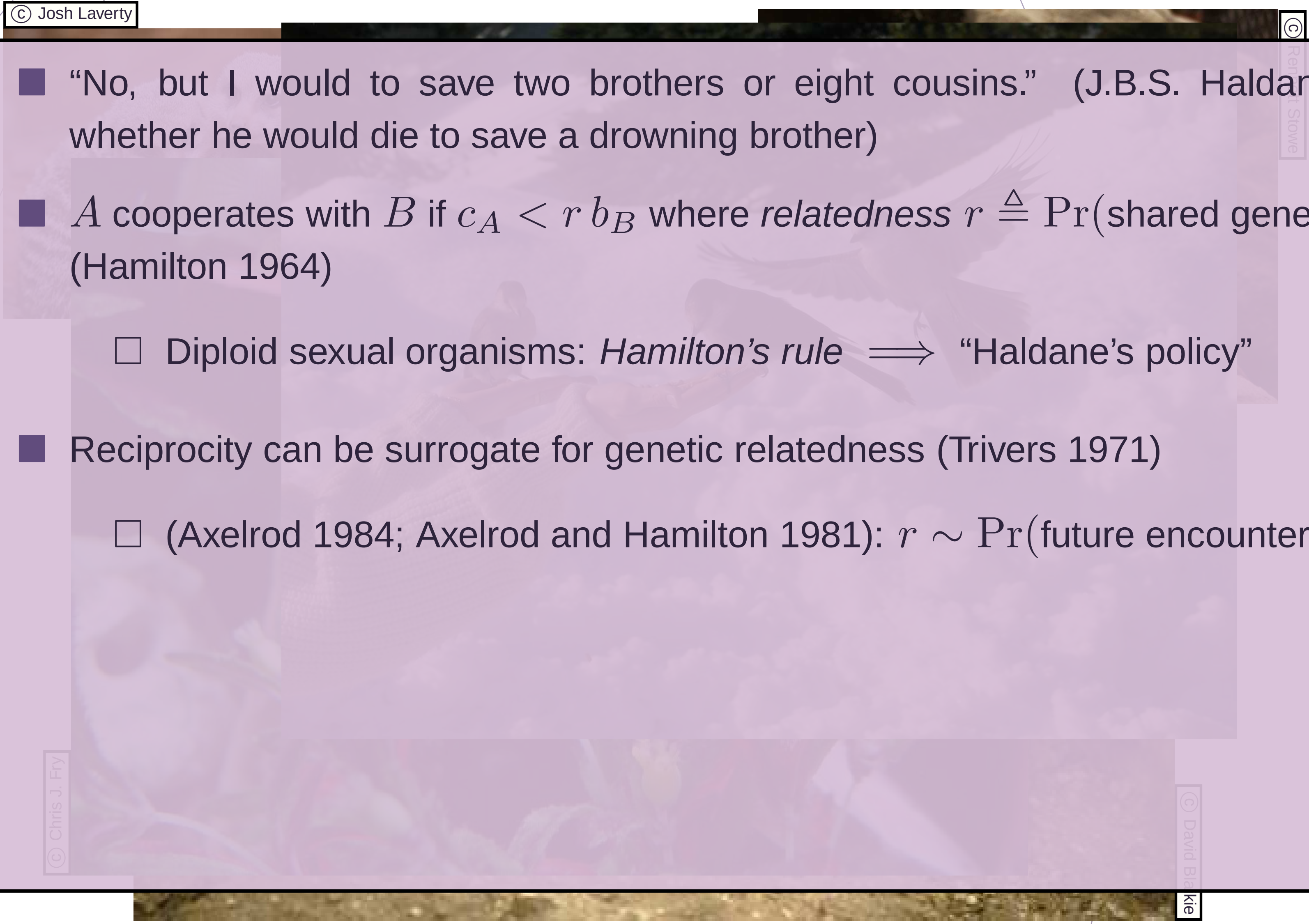
ⓒ Chris J. Fry

ⓒ David Bla   kie

- "No, but I would to save two brothers or eight cousins." (J.B.S. Haldane; whether he would die to save a drowning brother)
- A cooperates with B if $c_A < r\, b_B$ where *relatedness* $r \triangleq \Pr(\text{shared gene})$ (Hamilton 1964)
  - ☐ Diploid sexual organisms: *Hamilton's rule* $\Longrightarrow$ "Haldane's policy"

Evolution of cooperation

(Axelrod 1984; Hamilton 1964; Nowak 2006; Ohtsuki et al. 2006; Trivers 1971)



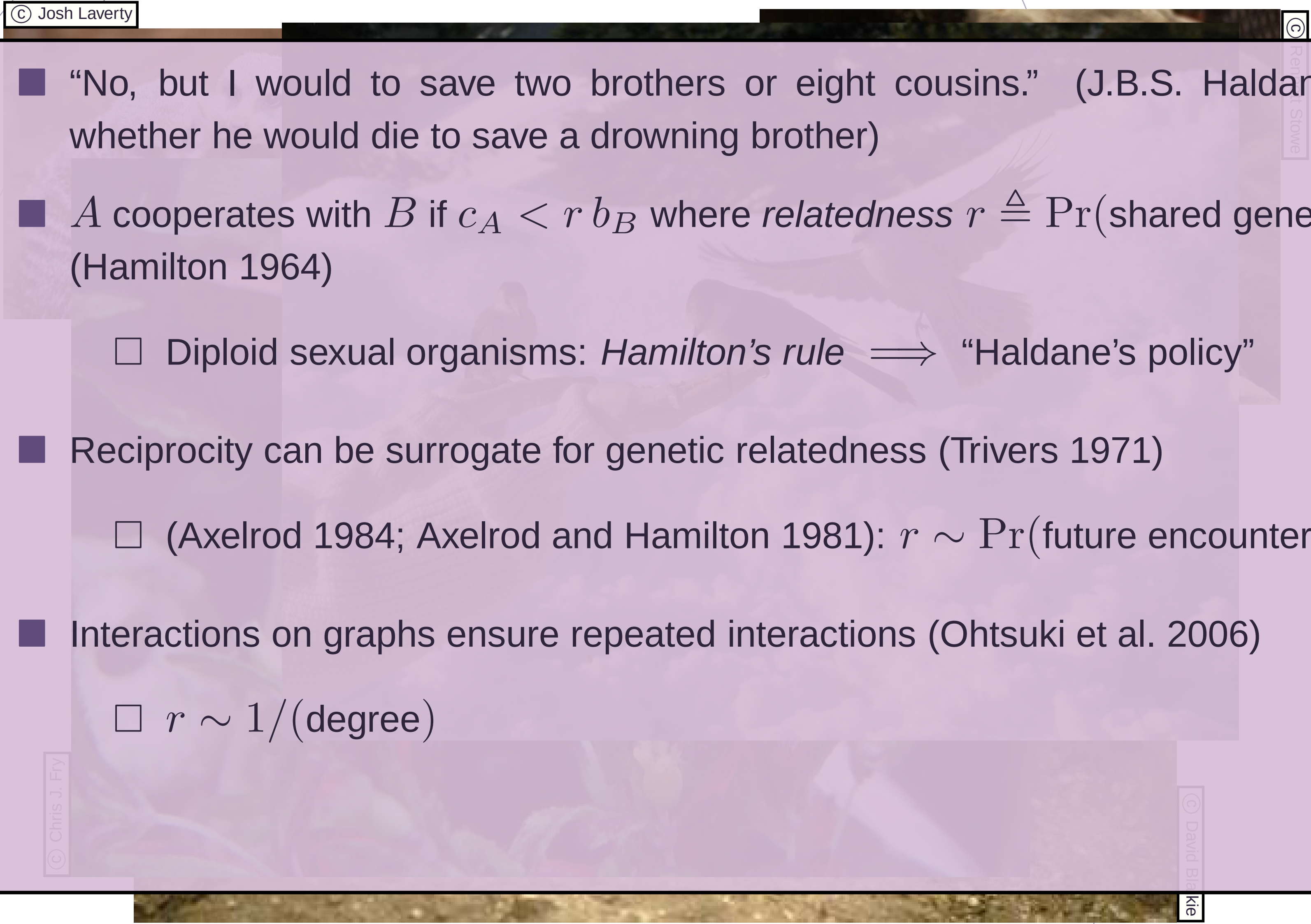
ⓒ Josh Laverty

ⓒ Chris J. Fry

- "No, but I would to save two brothers or eight cousins." (J.B.S. Haldane; whether he would die to save a drowning brother)
- A cooperates with B if $c_A < r\, b_B$ where *relatedness* $r \triangleq \Pr(\text{shared gene})$ (Hamilton 1964)
  - ☐ Diploid sexual organisms: *Hamilton's rule* $\Longrightarrow$ "Haldane's policy"
- Reciprocity can be surrogate for genetic relatedness (Trivers 1971)
  - ☐ (Axelrod 1984; Axelrod and Hamilton 1981): $r \sim \Pr(\text{future encounter})$

Evolution of cooperation

(Axelrod 1984; Hamilton 1964; Nowak 2006; Ohtsuki et al. 2006; Trivers 1971)



© Josh Laverty

© Rennett Stowe

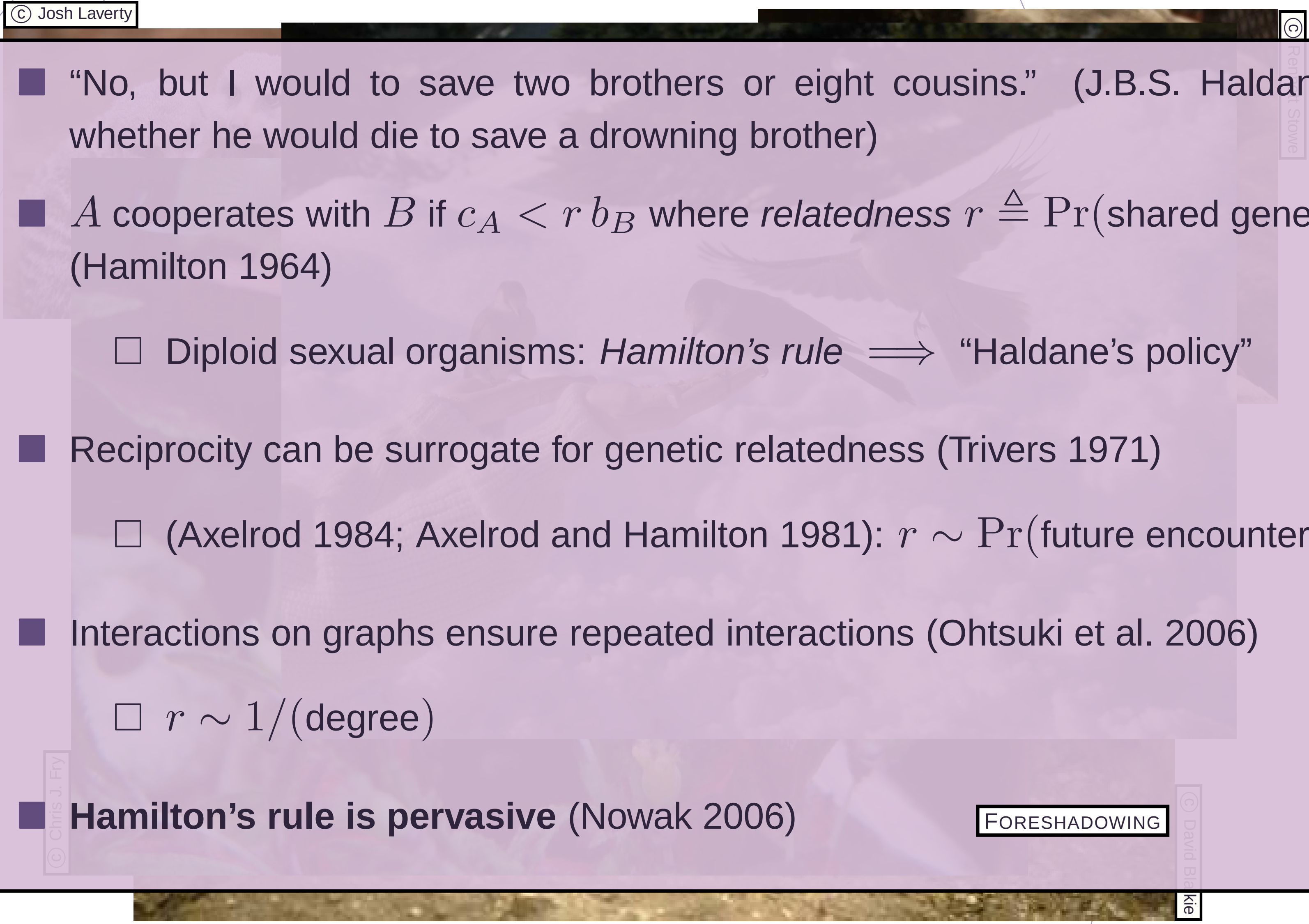
© Chris J. Fry

© David Blakie

- "No, but I would to save two brothers or eight cousins." (J.B.S. Haldane; whether he would die to save a drowning brother)
- A cooperates with B if $c_A < r\, b_B$ where *relatedness* $r \triangleq \Pr(\text{shared gene})$ (Hamilton 1964)
    - ☐ Diploid sexual organisms: *Hamilton's rule* $\Longrightarrow$ "Haldane's policy"
- Reciprocity can be surrogate for genetic relatedness (Trivers 1971)
    - ☐ (Axelrod 1984; Axelrod and Hamilton 1981): $r \sim \Pr(\text{future encounter})$
- Interactions on graphs ensure repeated interactions (Ohtsuki et al. 2006)
    - ☐ $r \sim 1/(\text{degree})$

Evolution of cooperation

(Axelrod 1984; Hamilton 1964; Nowak 2006; Ohtsuki et al. 2006; Trivers 1971)

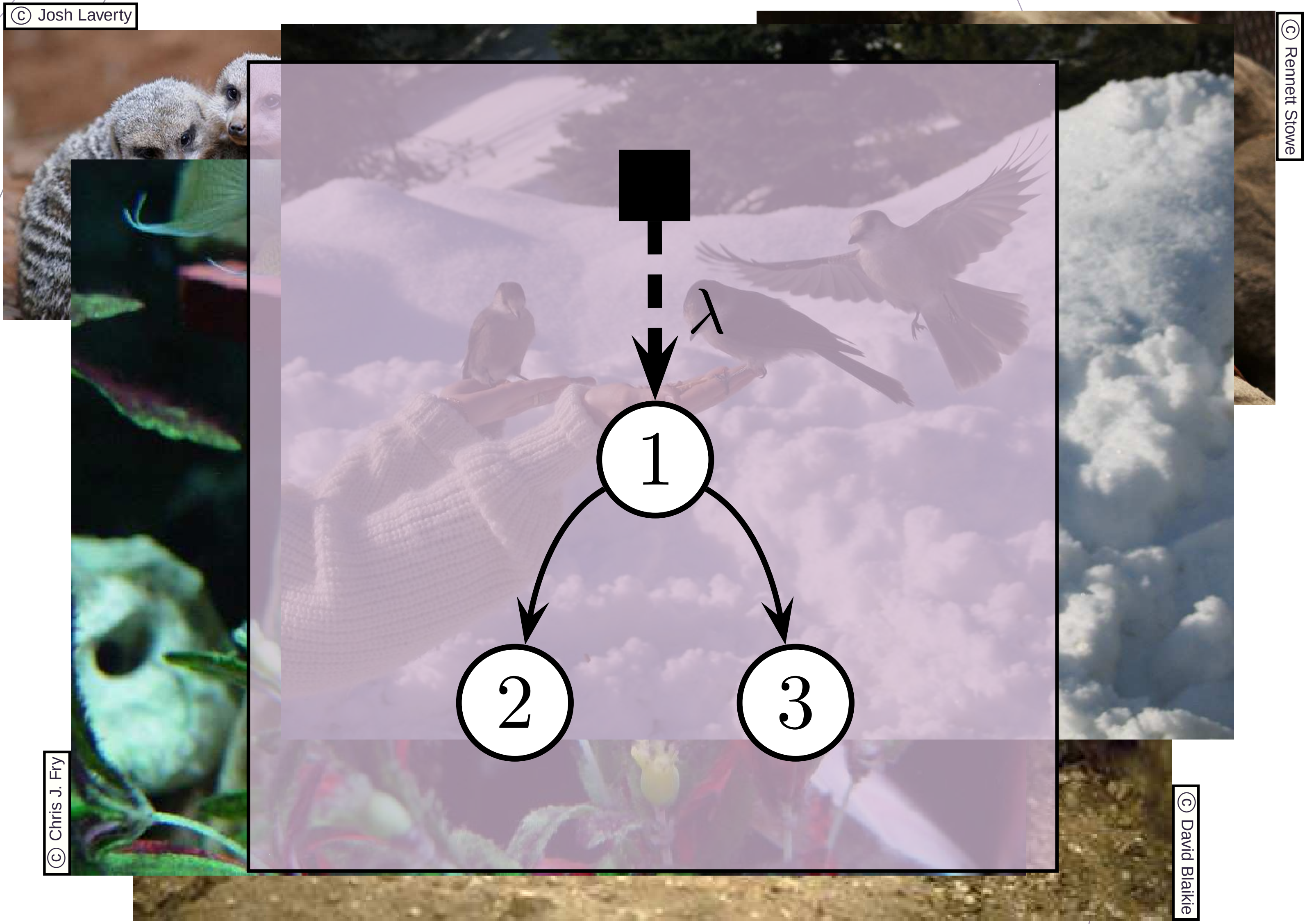


- “No, but I would to save two brothers or eight cousins.” (J.B.S. Haldane; whether he would die to save a drowning brother)
- A cooperates with B if $c_A < r\, b_B$ where *relatedness* $r \triangleq \Pr(\text{shared gene})$ (Hamilton 1964)
  - ☐ Diploid sexual organisms: *Hamilton’s rule* $\Longrightarrow$ “Haldane’s policy”
- Reciprocity can be surrogate for genetic relatedness (Trivers 1971)
  - ☐ (Axelrod 1984; Axelrod and Hamilton 1981): $r \sim \Pr(\text{future encounter})$
- Interactions on graphs ensure repeated interactions (Ohtsuki et al. 2006)
  - ☐ $r \sim 1/(\text{degree})$
- **Hamilton’s rule is pervasive** (Nowak 2006)

FORESHADOWING

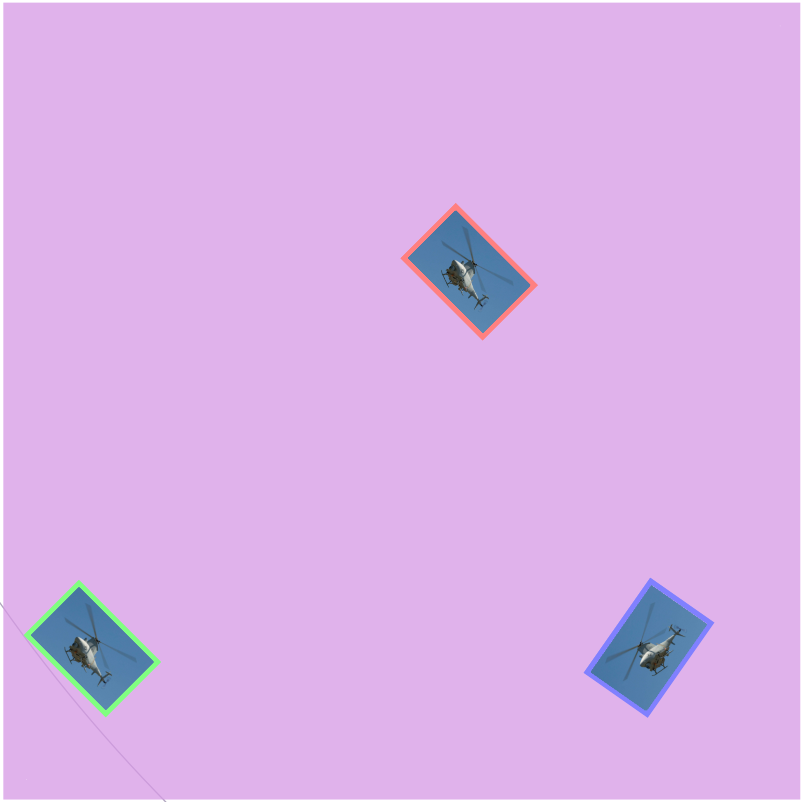
Example TPN: Cooperative breeding

(Pavlic and Passino 2010d)



Cooperative patrol

(Finke and Passino 2007; Finke et al. 2006; Gil et al. 2008)

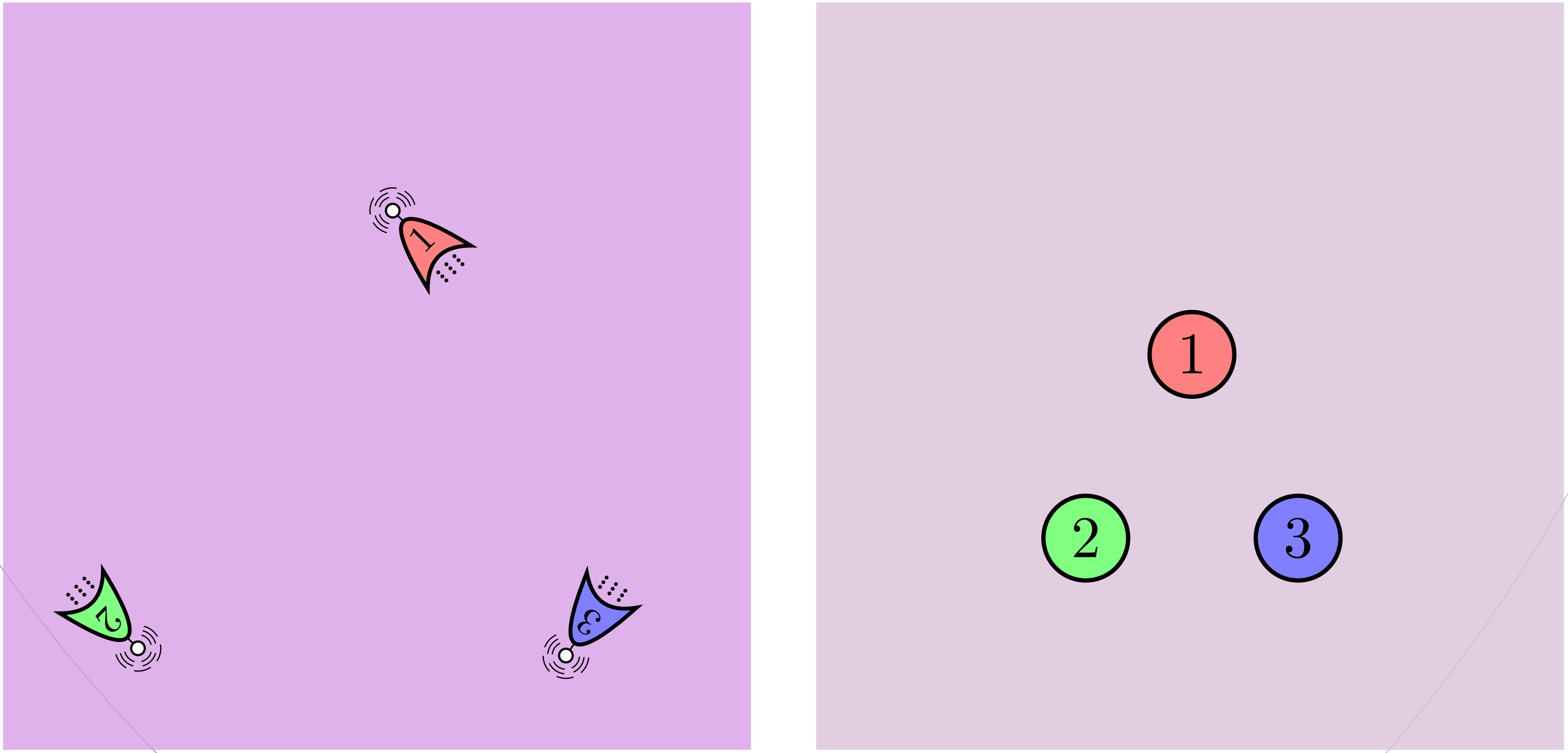


Example AAV Application:

Three MQ-8 Firescouts on patrol

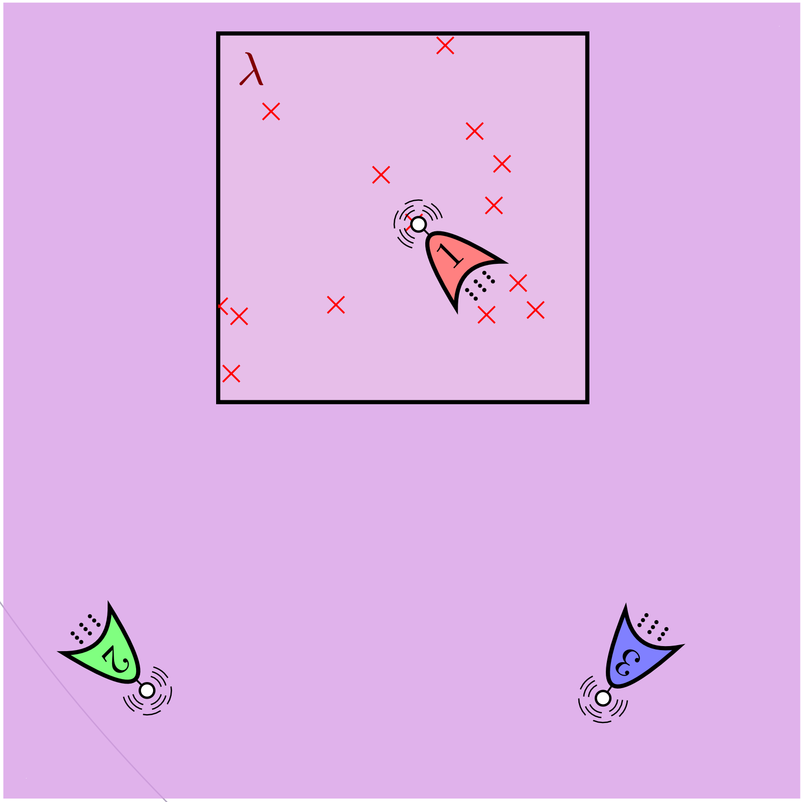
Example TPN: Cooperative patrol

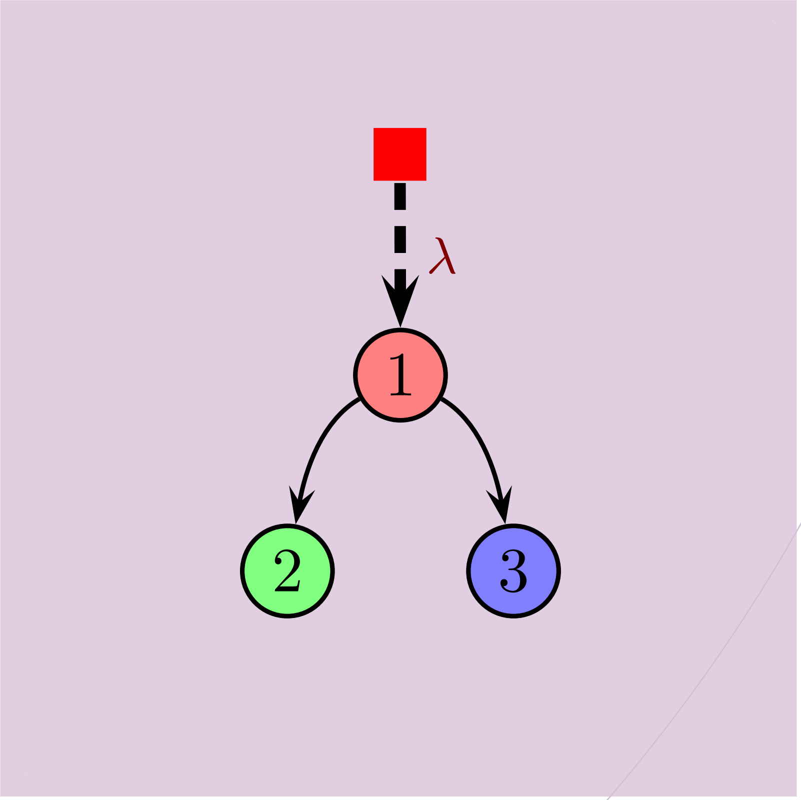
(Pavlic and Passino 2010d)



Example TPN: Cooperative patrol

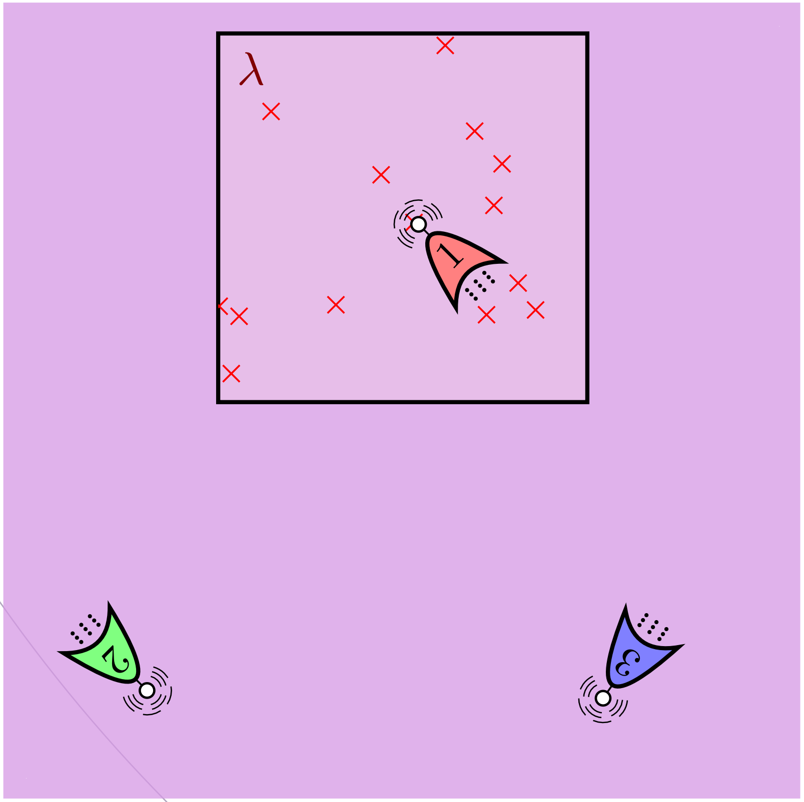
(Pavlic and Passino 2010d)

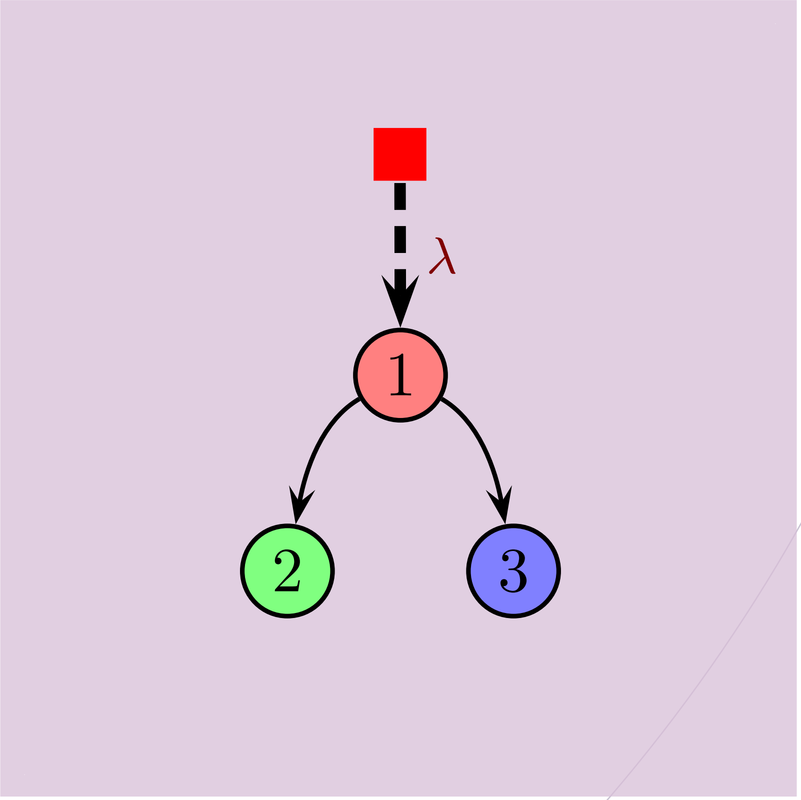




Example TPN: Cooperative patrol

(Pavlic and Passino 2010d)

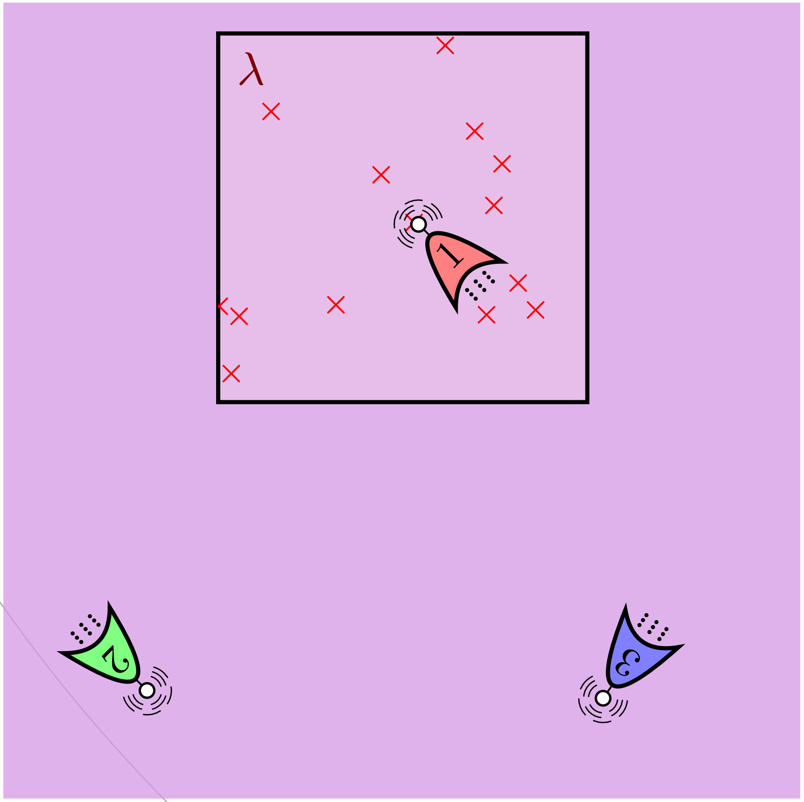


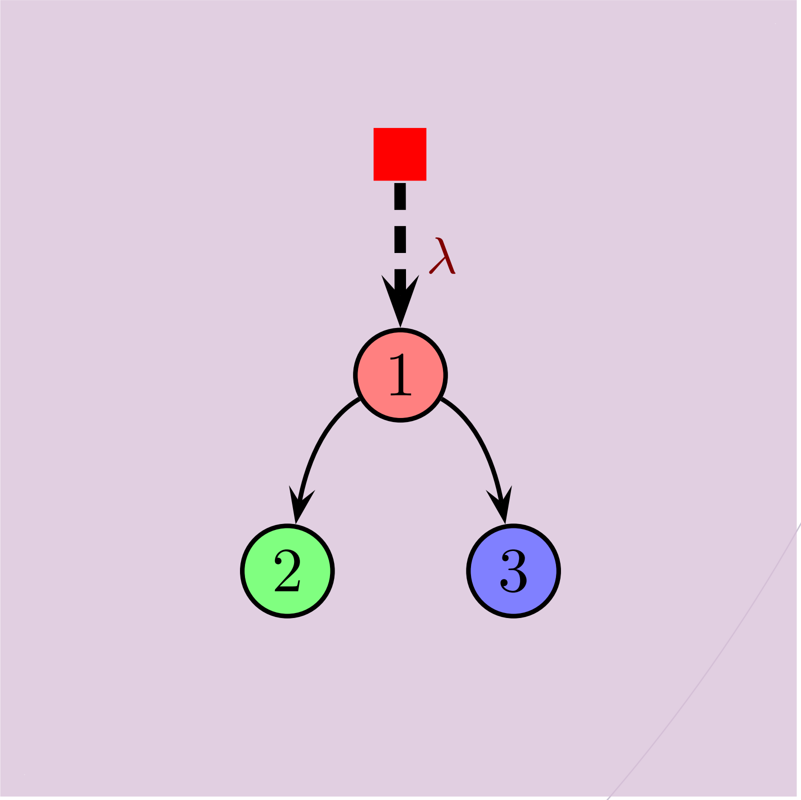


- $\sim$Cooperative breeding: breeder 1 and helpers 2 and 3

Example TPN: Cooperative patrol

(Pavlic and Passino 2010d)

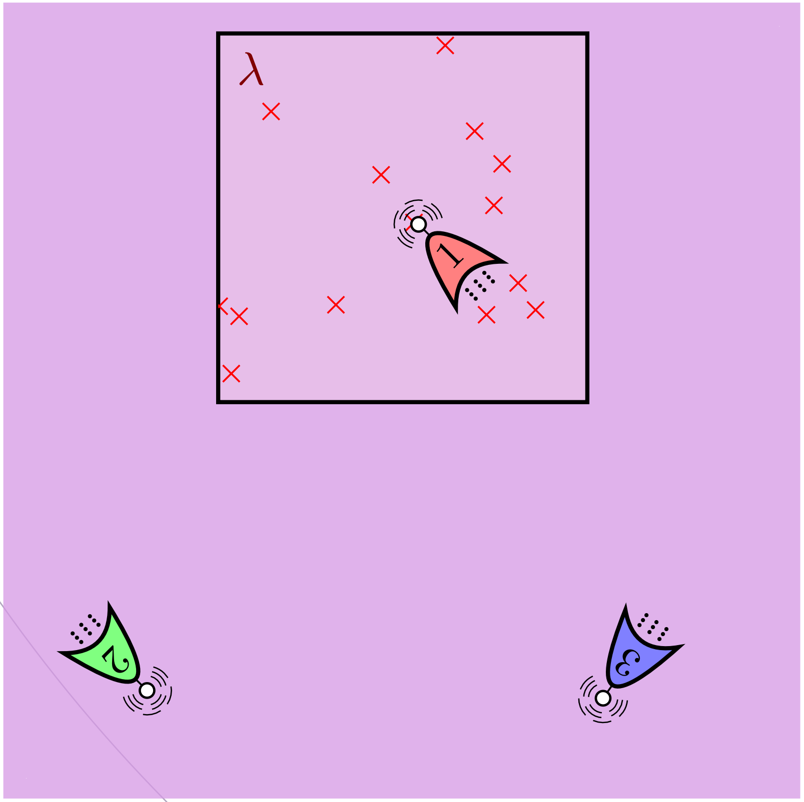


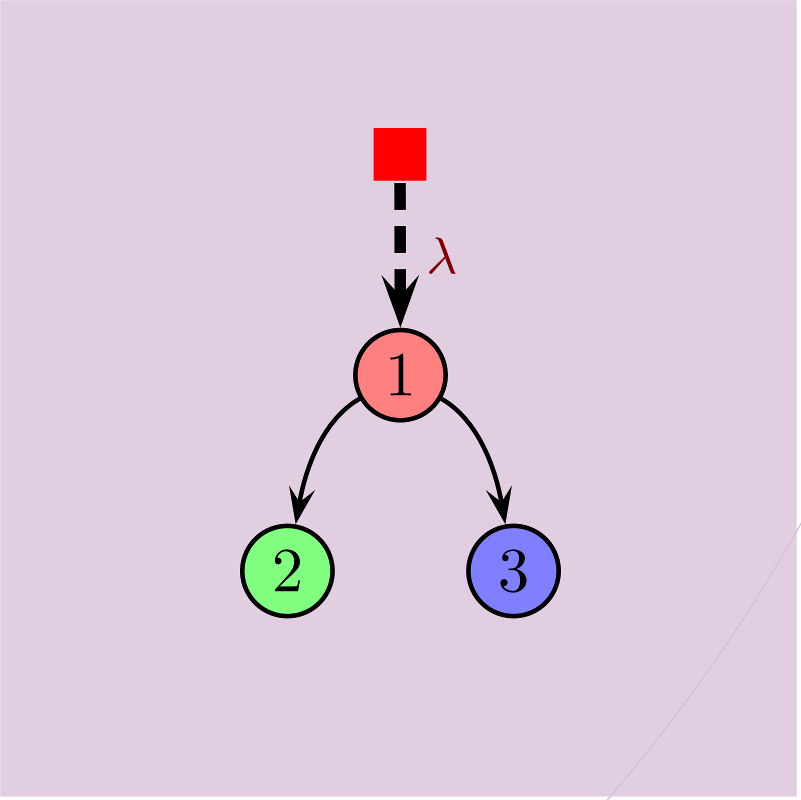


- ∼Social foraging: *producer* 1 and *scroungers* 2 and 3 (Giraldeau and Caraco 2000; Stephens and Krebs 1986)

Example TPN: Cooperative patrol

(Pavlic and Passino 2010d)

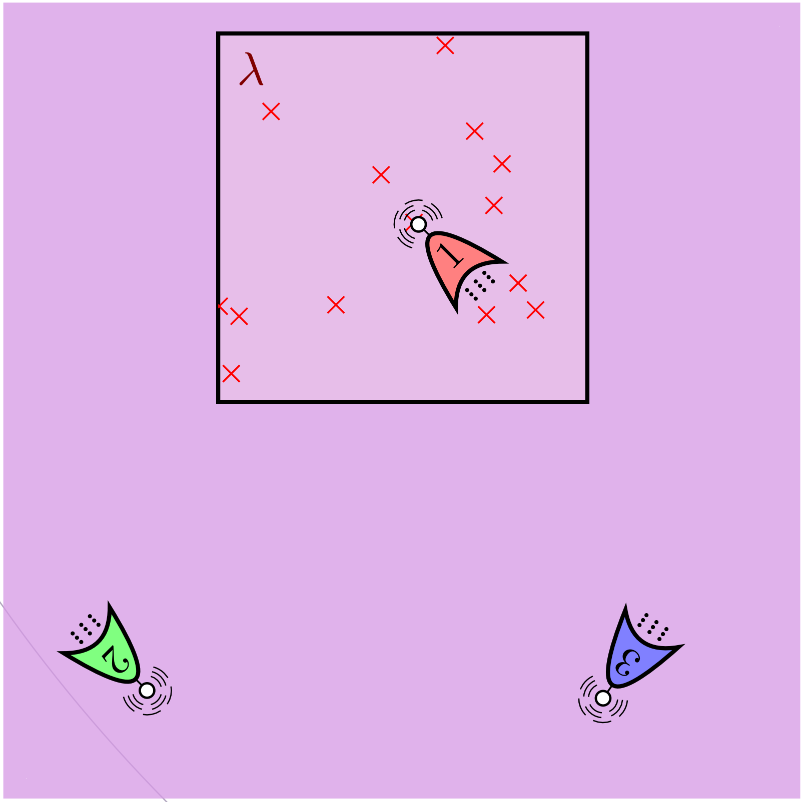


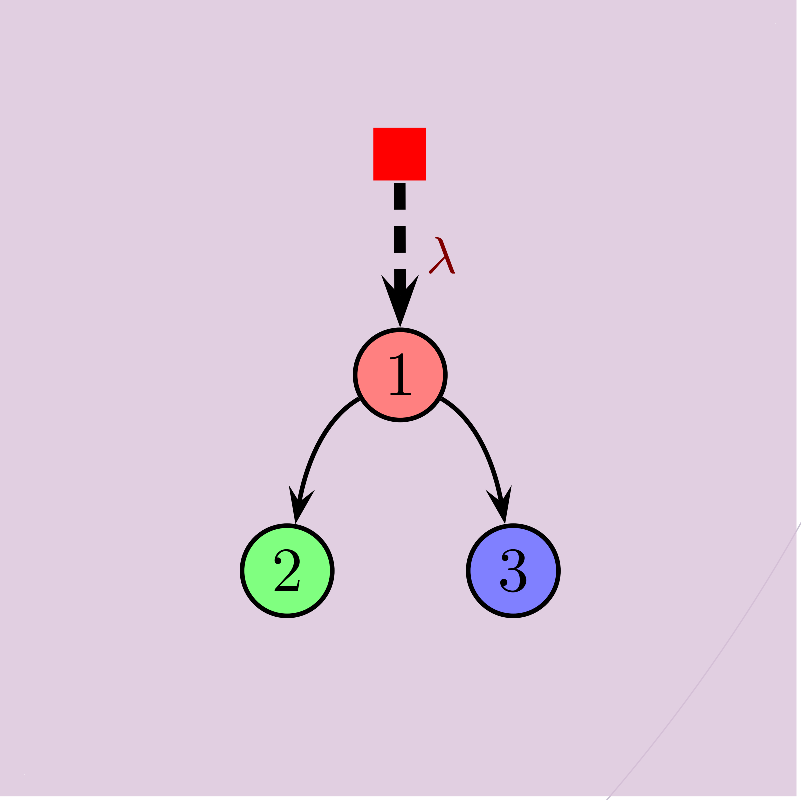


- Task-processing network (TPN): *conveyor* 1 and *cooperators* 2 and 3

Example TPN: Cooperative patrol

(Pavlic and Passino 2010d)

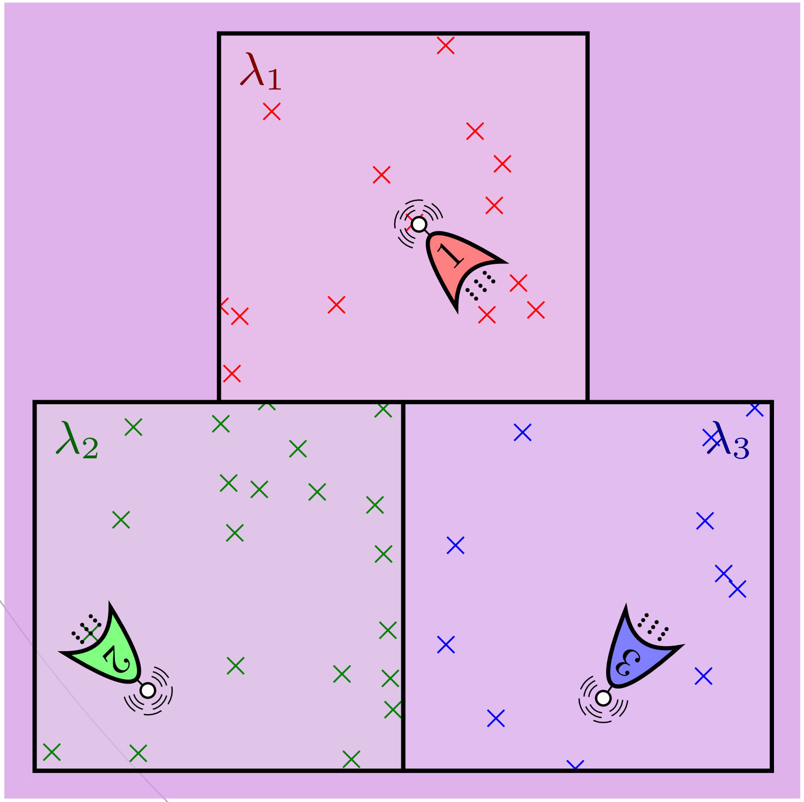


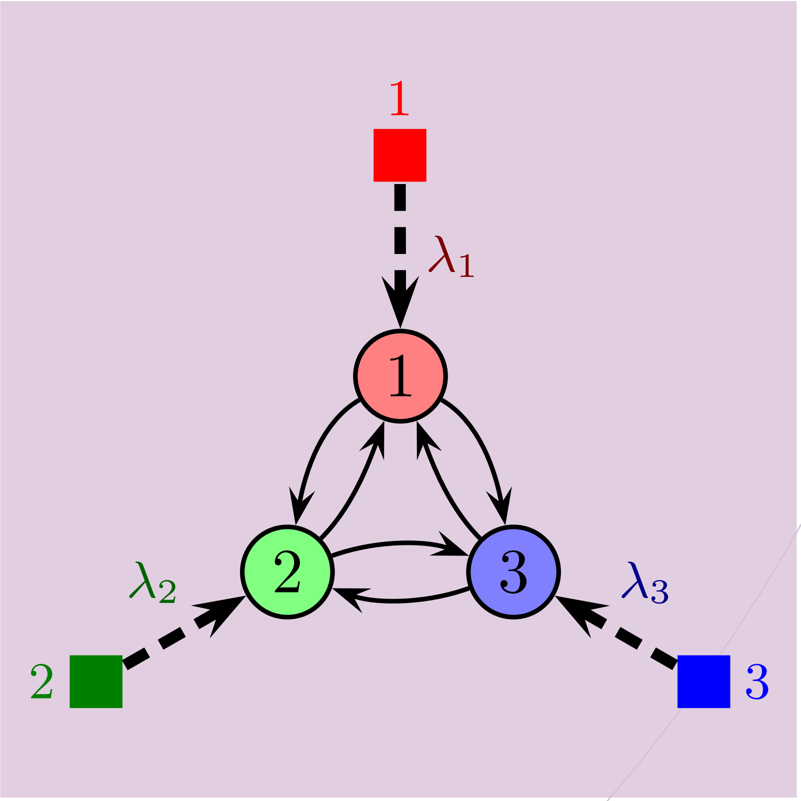


- Task-processing network (TPN): *conveyor* $\color{red}{1}$ and $\underbrace{\textit{cooperators}}_{\text{W/C}}$ $\color{green}{2}$ and $\color{blue}{3}$

Example TPN: Cooperative patrol

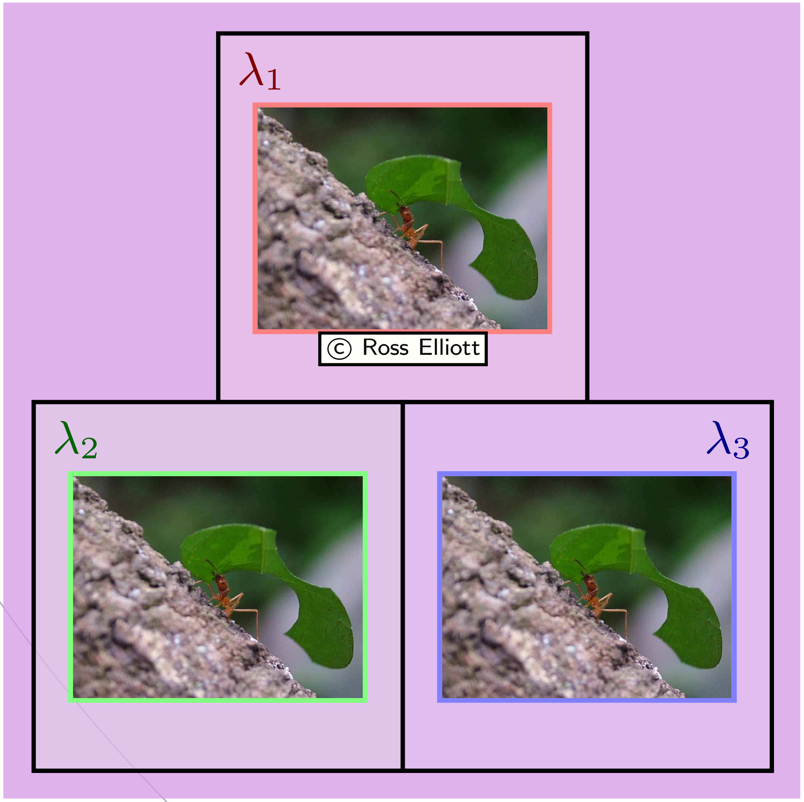
(Pavlic and Passino 2010d)

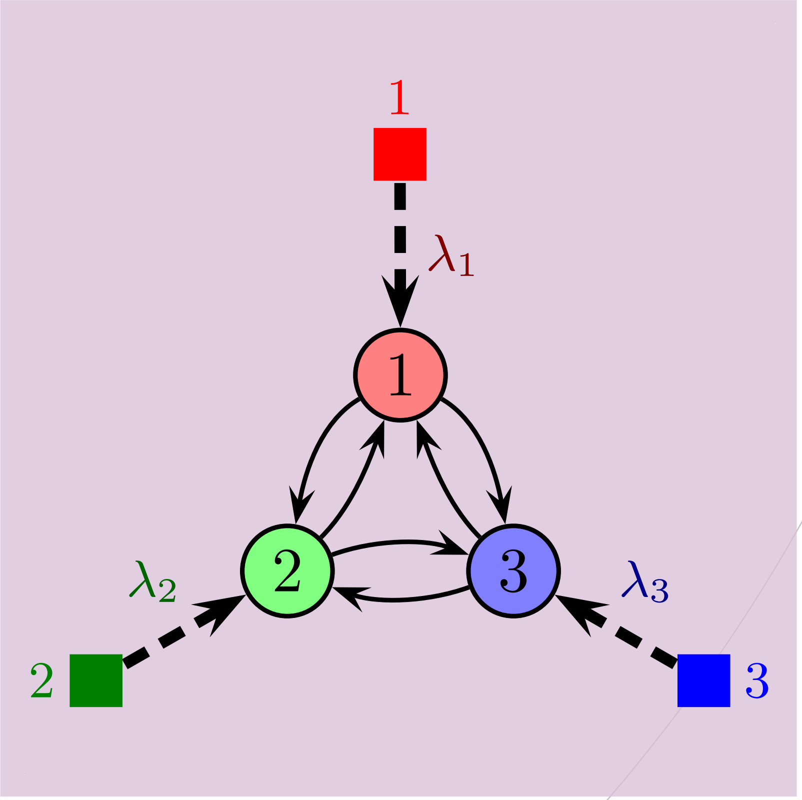




- Each can be both conveyor and cooperator simultaneously

Example TPN: Eusocial breeders?

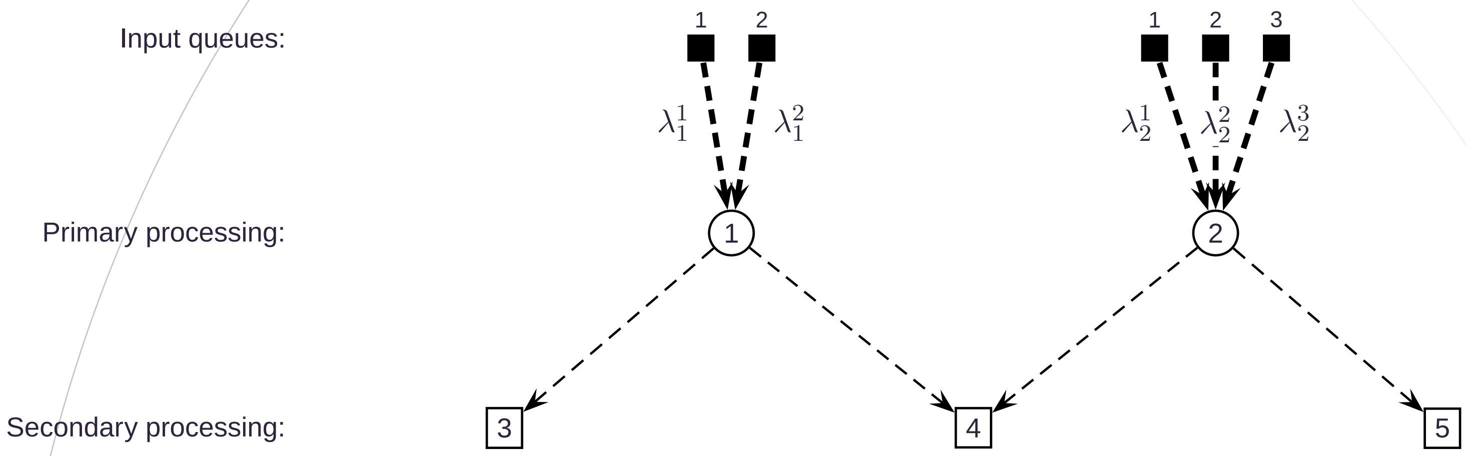




- Each can be both conveyor and cooperator simultaneously (e.g., eusocial polygyny/pleometrosis?)

Flexible manufacturing system

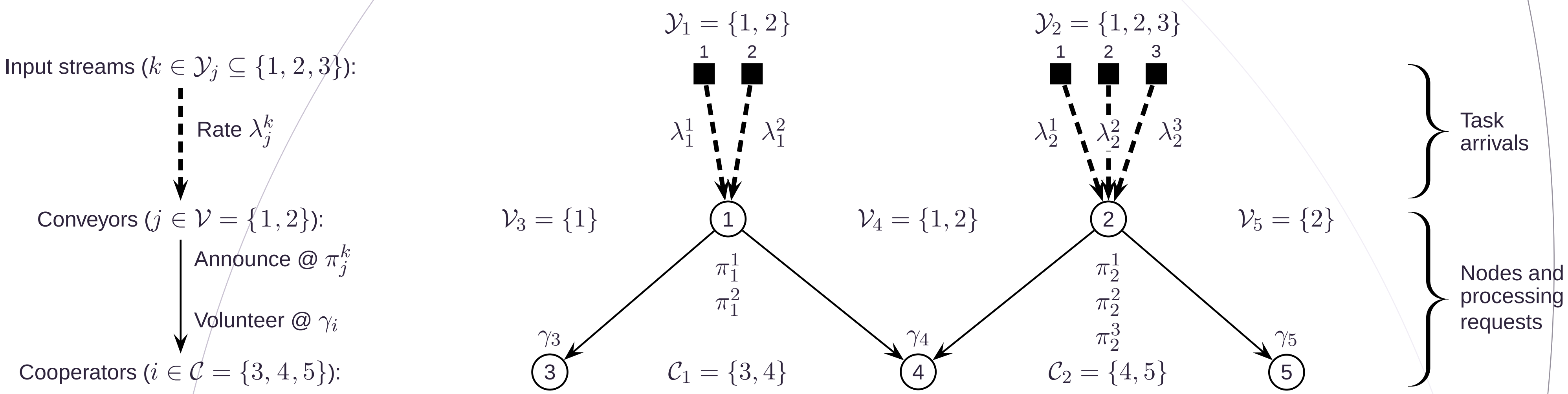
(Cruz 1991; Perkins and Kumar 1989)



Flexible manufacturing system (FMS)

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)



Input streams ($k \in \mathcal{Y}_j \subseteq \{1,2,3\}$):

Rate λ_j^k

Conveyors ($j \in \mathcal{V} = \{1,2\}$):

Announce @ π_j^k

Volunteer @ γ_i

Cooperators ($i \in \mathcal{C} = \{3,4,5\}$):

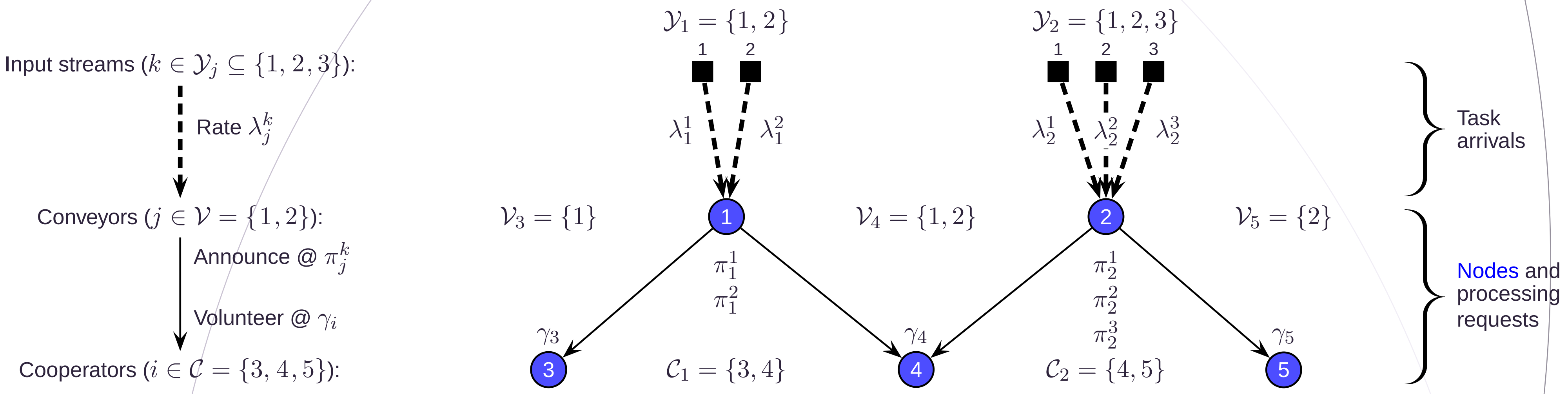
Task arrivals

Nodes and processing requests

Flexible manufacturing system (FMS)

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)

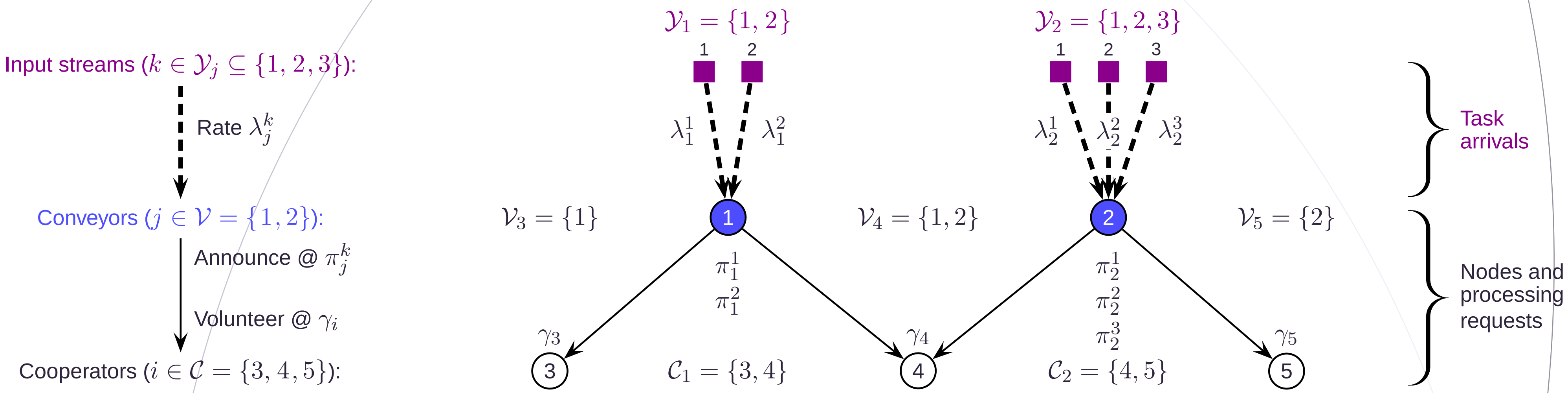


Flexible manufacturing system (FMS)

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)

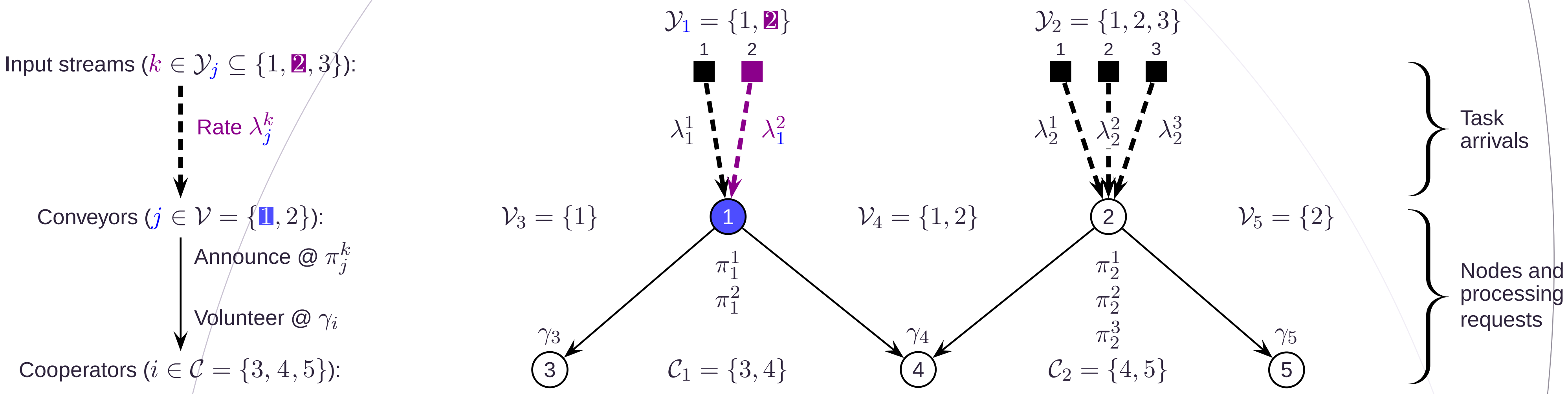


Flexible manufacturing system (FMS)

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents
- *Conveyor* $j \in \mathcal{V} \subseteq \mathcal{A}$ has $|\mathcal{Y}_j \subset \mathbb{N}| > 0$ types of incoming task orders

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)

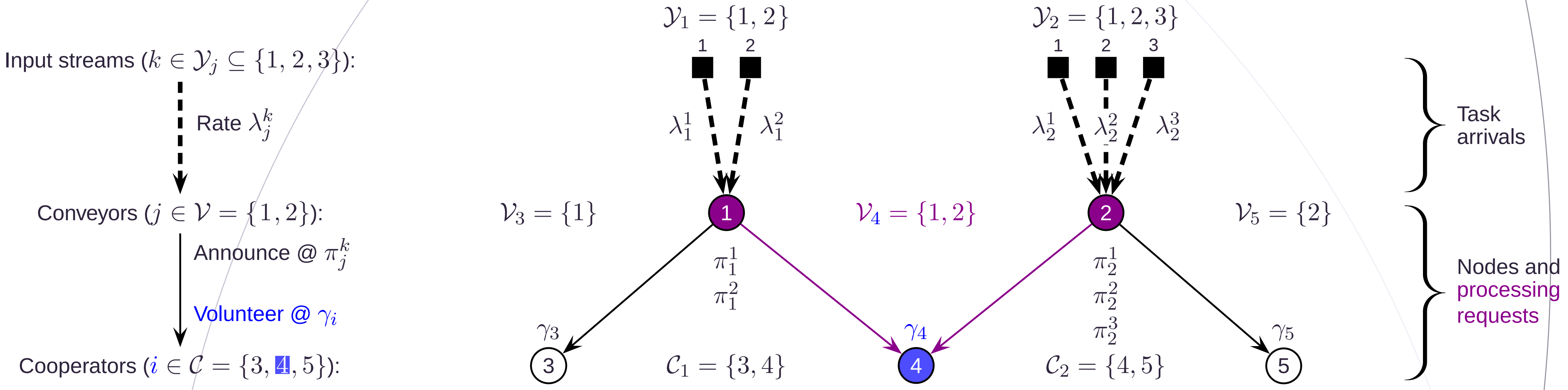


Flexible manufacturing system (FMS)

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents
- *Conveyor* $j \in \mathcal{V} \subseteq \mathcal{A}$ has $|\mathcal{Y}_j \subset \mathbb{N}| > 0$ types of incoming task orders
- Task orders of type $k \in \mathcal{Y}_j$ arrive at conveyor j at rate $\lambda_j^k \in \mathbb{R}_{\geq 0}$

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)

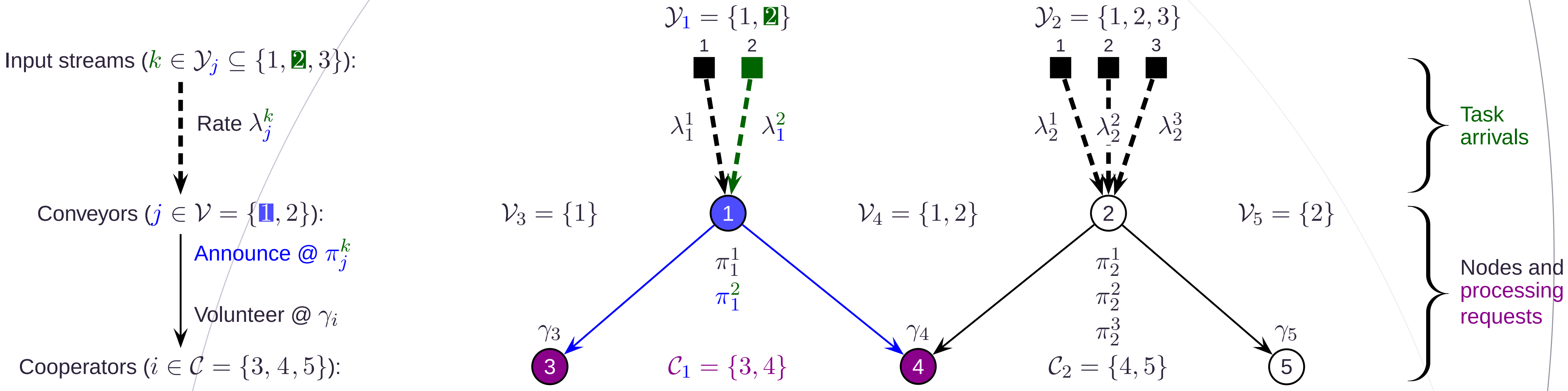


Flexible manufacturing system (FMS)

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents
- *Conveyor* $j \in \mathcal{V} \subseteq \mathcal{A}$ has $|\mathcal{Y}_j \subset \mathbb{N}| > 0$ types of incoming task orders
- Task orders of type $k \in \mathcal{Y}_j$ arrive at conveyor j at rate $\lambda_j^k \in \mathbb{R}_{\geq 0}$
- *Cooperator* $i \in \mathcal{C} \subseteq \mathcal{A}$ responds to $\gamma_i \in [0,1]$ broadcasts from conveyors $\mathcal{V}_i \subseteq \mathcal{V}$

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)



Input streams ($k \in \mathcal{Y}_j \subseteq \{1, \boxed{2}, 3\}$):

Rate λ_j^k

Conveyors ($j \in \mathcal{V} = \{\boxed{1}, 2\}$):

Announce @ π_j^k

Volunteer @ γ_i

Cooperators ($i \in \mathcal{C} = \{3, 4, 5\}$):

$\mathcal{Y}_1 = \{1, \boxed{2}\}$

$\mathcal{Y}_2 = \{1, 2, 3\}$

1 2 1 2 3

λ_1^1 λ_1^2 λ_2^1 λ_2^2 λ_2^3

$\mathcal{V}_3 = \{1\}$ 1 $\mathcal{V}_4 = \{1, 2\}$ 2 $\mathcal{V}_5 = \{2\}$

π_1^1 π_1^2

π_2^1 π_2^2 π_2^3

γ_3 γ_4 γ_5

3 $\mathcal{C}_1 = \{3, 4\}$ 4 $\mathcal{C}_2 = \{4, 5\}$ 5

Task arrivals

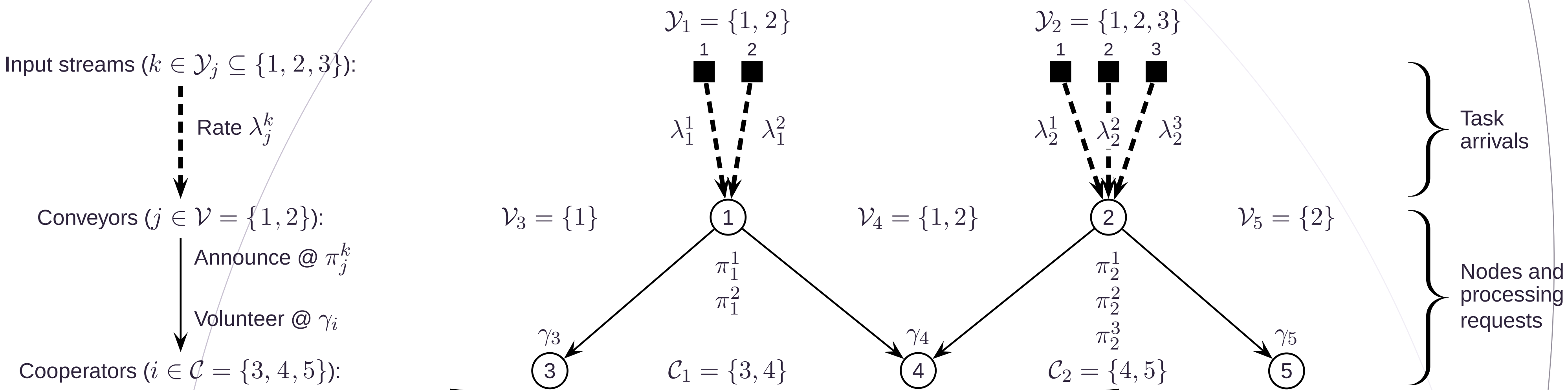
Nodes and processing requests

Flexible manufacturing system (FMS)

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents
- *Conveyor* $j \in \mathcal{V} \subseteq \mathcal{A}$ has $|\mathcal{Y}_j \subset \mathbb{N}| > 0$ types of incoming task orders
- Task orders of type $k \in \mathcal{Y}_j$ arrive at conveyor j at rate $\lambda_j^k \in \mathbb{R}_{\geq 0}$
- *Cooperator* $i \in \mathcal{C} \subseteq \mathcal{A}$ responds to $\gamma_i \in [0, 1]$ broadcasts from conveyors $\mathcal{V}_i \subseteq \mathcal{V}$
- Conveyor j broadcasts $\pi_j^k \in [0, 1]$ type-k arrivals to cooperators $\mathcal{C}_i \subseteq \mathcal{C}$

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)



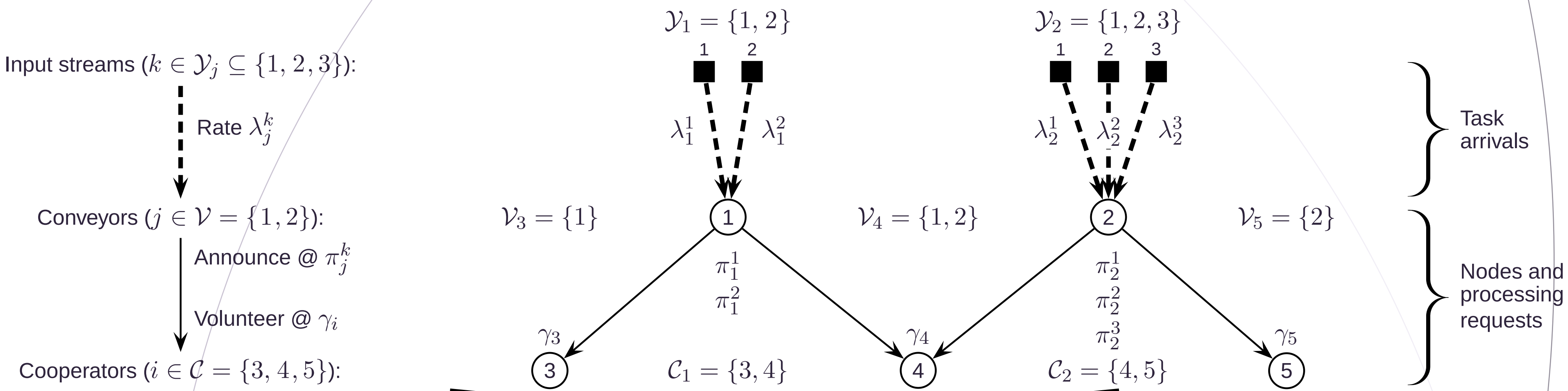
~~Flexible manufacturing system (FMS)~~

Cooperative breeders?

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents
- *Conveyor* $j \in \mathcal{V} \subseteq \mathcal{A}$ has $|\mathcal{Y}_j \subset \mathbb{N}| > 0$ types of incoming task orders
- Task orders of type $k \in \mathcal{Y}_j$ arrive at conveyor j at rate $\lambda_j^k \in \mathbb{R}_{\geq 0}$
- *Cooperator* $i \in \mathcal{C} \subseteq \mathcal{A}$ responds to $\gamma_i \in [0,1]$ broadcasts from conveyors $\mathcal{V}_i \subseteq \mathcal{V}$
- Conveyor j broadcasts $\pi_j^k \in [0,1]$ type-k arrivals to cooperators $\mathcal{C}_j \subseteq \mathcal{C}$

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)



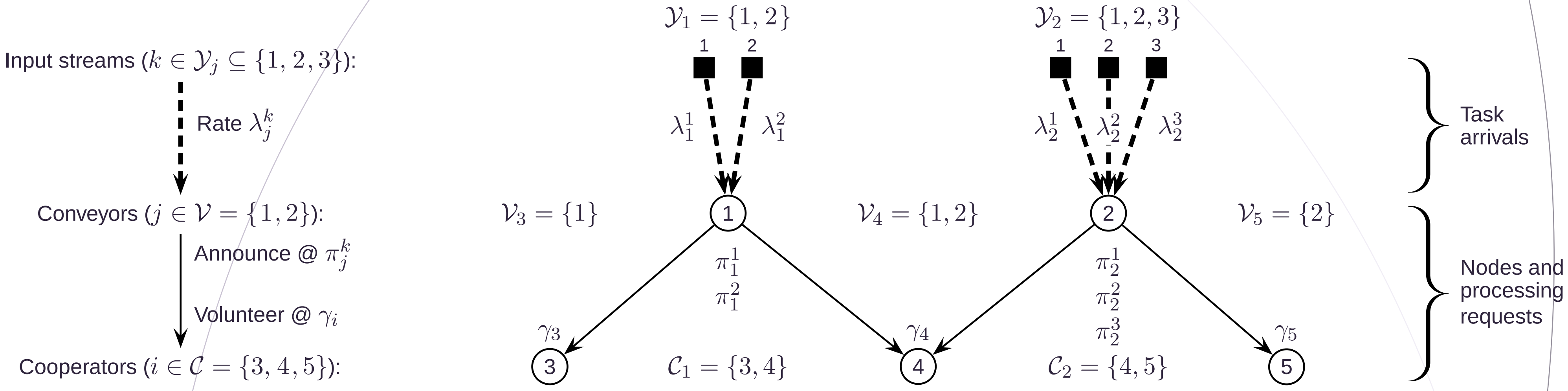
~~Flexible manufacturing system (FMS)~~

Coffee shop?

- $\mathcal{A} \subset \mathbb{N}$: Task-processing agents
- *Conveyor* $j \in \mathcal{V} \subseteq \mathcal{A}$ has $|\mathcal{Y}_j \subset \mathbb{N}| > 0$ types of incoming task orders
- Task orders of type $k \in \mathcal{Y}_j$ arrive at conveyor j at rate $\lambda_j^k \in \mathbb{R}_{\geq 0}$
- *Cooperator* $i \in \mathcal{C} \subseteq \mathcal{A}$ responds to $\gamma_i \in [0, 1]$ broadcasts from conveyors $\mathcal{V}_i \subseteq \mathcal{V}$
- Conveyor j broadcasts $\pi_j^k \in [0, 1]$ type-k arrivals to cooperators $\mathcal{C}_j \subseteq \mathcal{C}$

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)



Flexible manufacturing system (FMS)

$\subseteq \mathcal{V}$

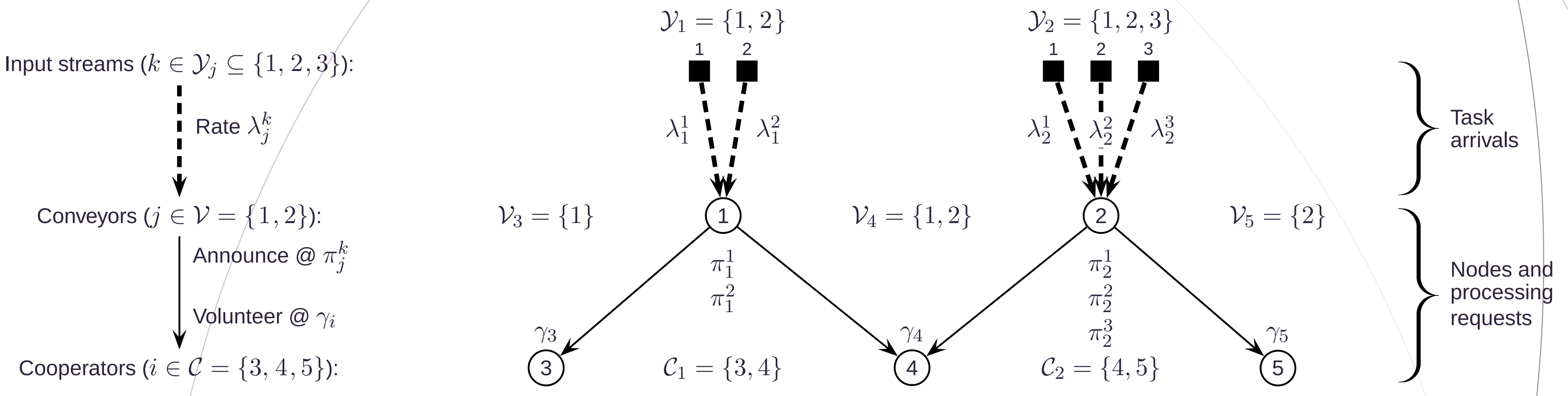
Similar models exist for network congestion control (Altman et al. 2005a,b; Buttyán and Hubaux 2003; Shakkottai et al. 2006).

- Messages move from source to destination;
  each message is **broadcasted to all** adjacent nodes.
- Intermediate nodes choose whether to **pass or drop** each message.
- Distributed policy is designed so Nash equilibrium solution is non-trivial.

Optimal multi-hop message-passing networks

Example TPN: Flexible manufacturing system

(Pavlic and Passino 2010d)



Flexible manufacturing system (FMS)

Similar models exist for network congestion control (Altman et al. 2005a,b; Buttyán and Hubaux 2003; Shakkottai et al. 2006).

Not ideal for **task processing**.

- Messages move from source to destination;
  each message is **broadcasted to all** adjacent nodes.
- Intermediate nodes choose whether to **pass or drop** each message.
- Distributed policy is designed so Nash equilibrium solution is non-trivial.

Optimal multi-hop message-passing networks

Cooperation game

Preliminaries*

- Let
  - ☐ $\mathcal{I} \subseteq \mathcal{A} \subset \mathbb{N}$: finite index set
  - ☐ $\Omega \triangleq \{\gamma_i\}_{i \in \mathcal{I}}$: indexed family with $\gamma_i \in [0,1]$ for each $i \in \mathcal{I}$

  For $g, h \in \mathbb{N}$ and $\Gamma \subseteq \mathcal{I}$, define SOBP and SOMS so

$$\mathrm{SOBP}_g(\Gamma) \triangleq \sum_{\ell=0}^{|\Gamma|} \frac{1}{g+\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}|=\ell}} \left(\left(\prod_{i \in \mathcal{C}} \gamma_i \right) \left(\prod_{k \in \Gamma - \mathcal{C}} (1-\gamma_k) \right) \right)$$

$$\mathrm{SOMS}_h(\Gamma) \triangleq \sum_{\ell=0}^{|\Gamma|} (-1)^\ell \frac{1}{h+\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}|=\ell}} \left(\prod_{i \in \mathcal{C}} \gamma_i \right).$$

- Properties of SOBP and SOMS provide bounds for convergence analysis (i.e., Lyapunov/non-deterministic set stability).

Cooperation game
Preliminaries*

- For $g, h \in \mathbb{N}$ and $\Gamma \subseteq \mathcal{I}$,

$$\mathrm{SOBP}_g(\Gamma) \triangleq \sum_{\ell=0}^{|\Gamma|} \frac{1}{g+\ell} \sum_{\substack{\mathcal{C}\subseteq\Gamma\\ |\mathcal{C}|=\ell}} \left(\left(\prod_{i\in\mathcal{C}} \gamma_i \right) \left(\prod_{k\in\Gamma-\mathcal{C}} (1-\gamma_k) \right) \right)$$

- For $\Gamma \subseteq \mathcal{A}$, $\mathrm{SOBP}_1(\{i,k,\ell\} - \{i\})$ is

$$(1-\gamma_k)(1-\gamma_\ell) + \frac{1}{2}\gamma_k(1-\gamma_\ell) + \frac{1}{2}\gamma_\ell(1-\gamma_k) + \frac{1}{3}\gamma_k\gamma_\ell$$

(i.e., *sum of binomial products*). For conveyor $j \in \mathcal{V}$ and cooperator $i \in \mathcal{C}_j = \{i,k,\ell\}$, $\mathrm{SOBP}_1(\{i,k,\ell\} - \{i\})$ is the probability that i is chosen to process an advertised task from $j \in \mathcal{V}_i$ (given that it volunteered).

Cooperation game
Preliminaries*

- For $g, h \in \mathbb{N}$ and $\Gamma \subseteq \mathcal{I}$,

$$\mathrm{SOBP}_g(\Gamma) \triangleq \sum_{\ell=0}^{|\Gamma|} \frac{1}{g+\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}|=\ell}} \left(\left(\prod_{i \in \mathcal{C}} \gamma_i \right) \left(\prod_{k \in \Gamma - \mathcal{C}} (1-\gamma_k) \right) \right)$$

- For $\Gamma \subseteq \mathcal{A}$, $\mathrm{SOBP}_1(\{i,k,\ell\} - \{i\})$ is

$$(1-\gamma_k)(1-\gamma_\ell) + \frac{1}{2}\gamma_k(1-\gamma_\ell) + \frac{1}{2}\gamma_\ell(1-\gamma_k) + \frac{1}{3}\gamma_k\gamma_\ell$$

(i.e., *sum of binomial products*). For conveyor $j \in \mathcal{V}$ and cooperator $i \in \mathcal{C}_j = \{i,k,\ell\}$, $\mathrm{SOBP}_1(\{i,k,\ell\} - \{i\})$ is the probability that i is chosen to process an advertised task from $j \in \mathcal{V}_i$ (given that it volunteered).

- SOMS gives slope and curvature information about SOBP.

Cooperation game

Agent utility function: rate of gain

For cooperator $i \in \mathcal{C}$, its local rate of gain

$$U_i(\vec{\gamma}) \triangleq \overbrace{b_i + \underbrace{\left(1 - \prod_{j \in \mathcal{C}_i} (1 - \gamma_j)\right)}_{\Pr(\text{Volunteer from } \mathcal{C}_i \,|\, \text{Advertisement from } i)} r_i}^{\text{Conveyor part — constant with respect to } \gamma_i} + \underbrace{\gamma_i \sum_{j \in \mathcal{V}_i} \Big(\qquad - \overbrace{\mathrm{SOBP}_1(\mathcal{C}_j - \{i\})}^{\Pr(i \text{ awarded task from } j \,|\, i \text{ volunteers})} c_{ij} \Big)}_{\text{Cooperator part}}$$

Costs and benefits of local processing on $i \in \mathcal{V}$:

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

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

Costs and benefits to $i \in \mathcal{C}$ for volunteering for tasks exported from $j \in \mathcal{V}_i$:

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

Cooperation game
Agent utility function: rate of gain

For cooperator $i \in \mathcal{C}$, its local rate of gain

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

Costs and benefits of local processing on $i \in \mathcal{V}$:

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

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

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

Costs and benefits to $i \in \mathcal{C}$ for volunteering for tasks exported from $j \in \mathcal{V}_i$:

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

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

Fictitious payment functions added as stabilizing controls (“quantity” $Q_i \triangleq \sum_{j\in\mathcal{C}_i} \gamma_j$).

Cooperation game
Agent utility function: rate of gain

Decreasing-cost externality

For cooperator $i \in \mathcal{C}$, its local rate of gain

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

Costs and benefits of local processing on $i \in \mathcal{V}$:

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

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

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

Cournot oligopolies on a graph

Costs and benefits to $i \in \mathcal{C}$ for volunteering for tasks exported from $j \in \mathcal{V}_i$:

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

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

Fictitious payment functions added as stabilizing controls (“quantity” $Q_i \triangleq \sum_{j\in\mathcal{C}_i} \gamma_j$).

Nash equilibrium
Existence, uniqueness, and asynchronous convergence

- Totally asynchronous parallel computation of $\vec{\gamma}^*$ by local gradient ascent

Nash equilibrium
Existence, uniqueness, and asynchronous convergence

- Totally asynchronous parallel computation of $\vec{\gamma}^*$ by local gradient ascent
  - ☐ Agents iterate asynchronously.
  - ☐ Each agent operates on a possibly outdated copy of $\vec{\gamma}$.
  - ☐ Asynchronous system is described by difference inclusion.

Nash equilibrium
Existence, uniqueness, and asynchronous convergence

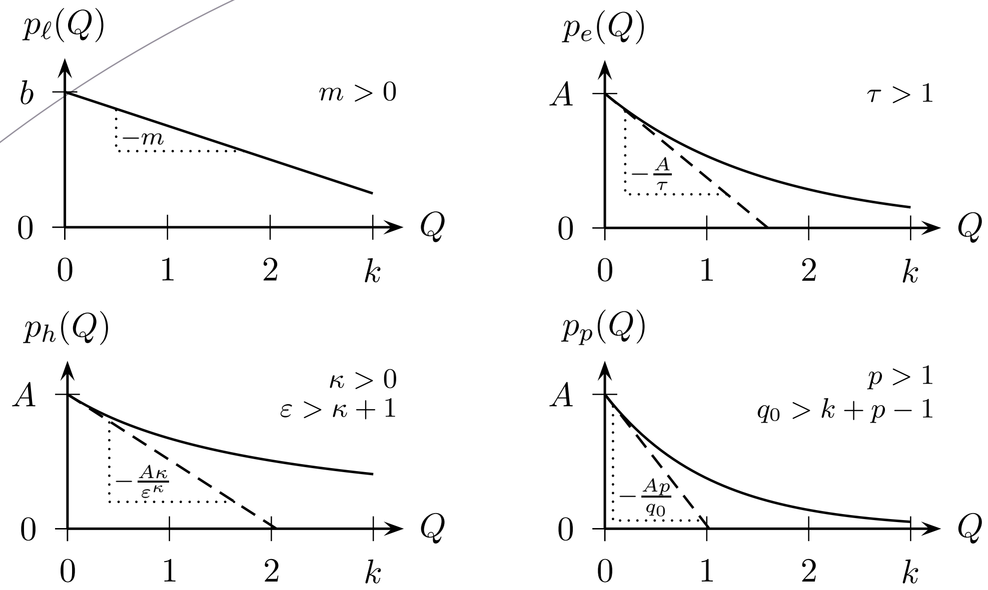
- Totally asynchronous parallel computation of $\vec{\gamma}^*$ by local gradient ascent
  - ☐ Agents iterate asynchronously.
  - ☐ Each agent operates on a possibly outdated copy of $\vec{\gamma}$.
  - ☐ Asynchronous system is described by difference inclusion.
  - ☐ It is sufficient to show synchronous transition mapping is a contraction with respect to maximum norm $(\|\vec{\gamma}\|_\infty \triangleq \max_{i \in \mathcal{C}}\{|\gamma_i|\})$.
  - ☐ A unique equilibrium exists and is asymptotically stable.

Nash equilibrium
Existence, uniqueness, and asynchronous convergence

- Totally asynchronous parallel computation of $\vec{\gamma}^*$ by local gradient ascent
  - ☐ Agents iterate asynchronously.
  - ☐ Each agent operates on a possibly outdated copy of $\vec{\gamma}$.
  - ☐ Asynchronous system is described by difference inclusion.
  - ☐ It is sufficient to show synchronous transition mapping is a contraction with respect to maximum norm ($\|\vec{\gamma}\|_\infty \triangleq \max_{i \in \mathcal{C}}\{|\gamma_i|\}$).
  - ☐ A unique equilibrium exists and is asymptotically stable.
- Constraints on topology and payment functions ensure contraction.

Payment and topological constraints

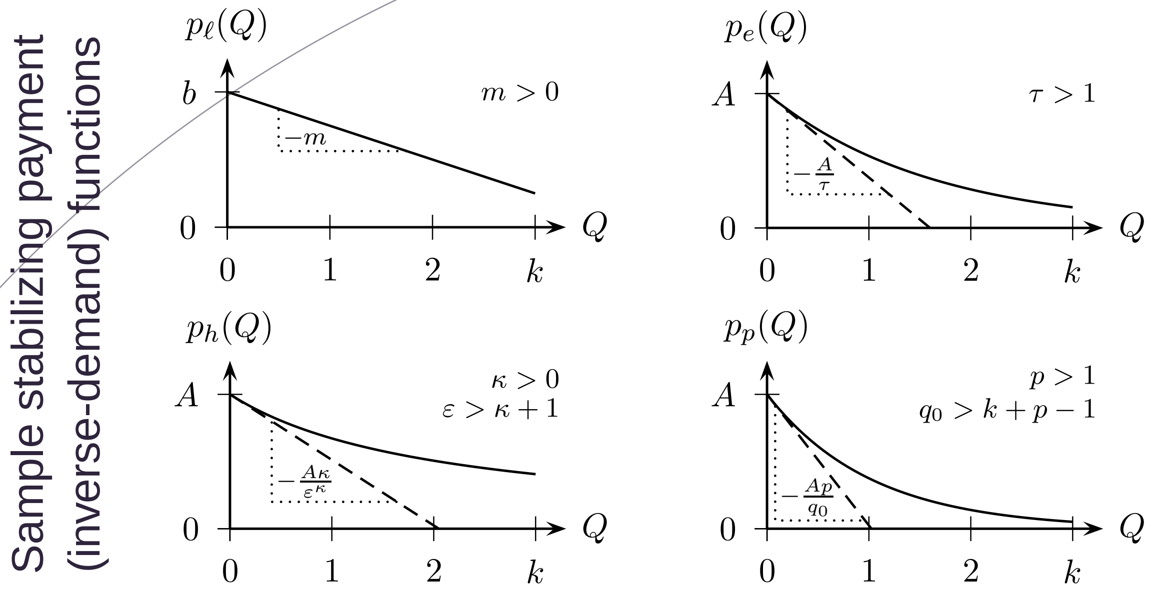
Definitions



Sample stabilizing payment (inverse-demand) functions

- For $k \in \mathbb{N}$, $p : [0, k] \mapsto \mathbb{R}$ is a stabilizing payment function if
  - ☐ $p'(Q) \triangleq \mathrm{d}p(Q)/\mathrm{d}Q < 0$ for all $Q \in [0, k]$.
  - ☐ $p''(Q) \triangleq \mathrm{d}^2 p(Q)/\mathrm{d}Q^2 > 0$ for all $Q \in [0, k]$.
  - ☐ $\gamma p''(Q) \leq -p'(Q)$ for all $Q \in [\gamma, k - (1 - \gamma)]$ with $\gamma \in [0, 1]$.

Payment and topological constraints
Definitions


Sample stabilizing payment (inverse-demand) functions

- For $k \in \mathbb{N}$, $p : [0, k] \mapsto \mathbb{R}$ is a stabilizing payment function if
    - $p'(Q) \triangleq \mathrm{d}p(Q)/\mathrm{d}Q < 0$ for all $Q \in [0, k]$.
    - $p''(Q) \triangleq \mathrm{d}^2 p(Q)/\mathrm{d}Q^2 > 0$ for all $Q \in [0, k]$.
    - $\gamma p''(Q) \leq -p'(Q)$ for all $Q \in [\gamma, k - (1 - \gamma)]$ with $\gamma \in [0, 1]$.
- For $k \in \{0, 1, \dots, |\mathcal{C}|\}$, a conveyor $j \in \mathcal{V}$ is called a k*-conveyor* if it has k outgoing connections (i.e., $|\mathcal{C}_j| = k$).

Payment and topological constraints
Assumptions

- Assume that:

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

Assumptions

- Assume that:

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; each conveyor is a k-conveyor where $k \in \{0, 1, 2, 3\}$).

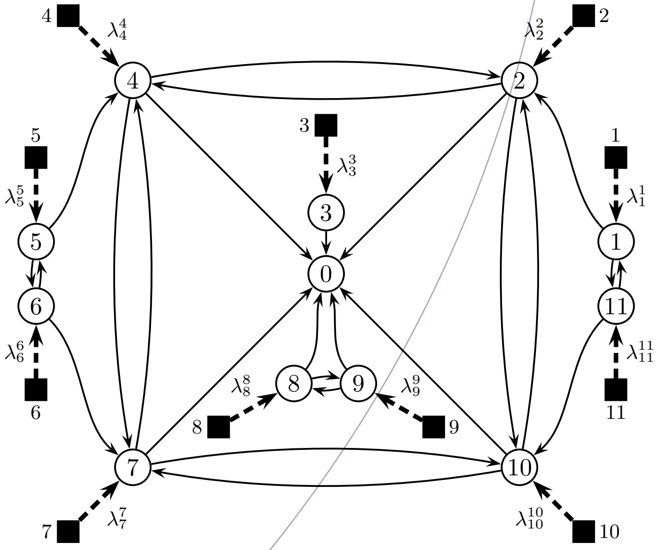
Payment and topological constraints
Assumptions

- Assume that:

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; each conveyor is a k-conveyor where $k \in \{0, 1, 2, 3\}$).
3. For cooperator $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, if j is a 3-conveyor (i.e., $|\mathcal{C}_j| = 3$), then there must be some other conveyor $k \in \mathcal{V}_i$ that is a 2-conveyor (i.e., $|\mathcal{C}_k| = 2$).

Payment and topological constraints
Assumptions

- Assume that:
  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; each conveyor is a k-conveyor where $k \in \{0, 1, 2, 3\}$).
  3. For cooperator $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, if j is a 3-conveyor (i.e., $|\mathcal{C}_j| = 3$), then there must be some other conveyor $k \in \mathcal{V}_i$ that is a 2-conveyor (i.e., $|\mathcal{C}_k| = 2$).



Algorithm for totally asynchronous gradient ascent

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

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

Projected gradient ascent

Algorithm for totally asynchronous gradient ascent

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

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

where

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

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

Algorithm for totally asynchronous gradient ascent

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

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

where

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

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

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

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

Algorithm for totally asynchronous gradient ascent

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

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

where

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

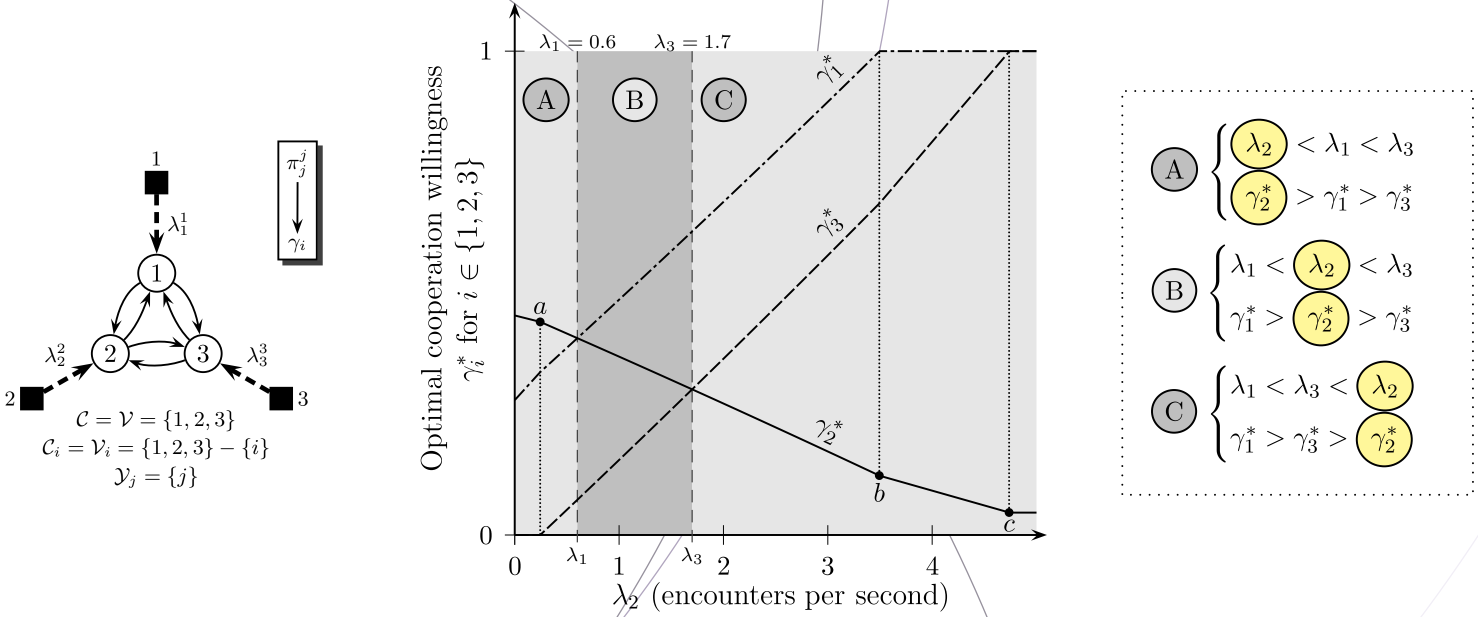
for all $\vec{\gamma} \in [0,1]^n$. If

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

∼ Hamilton's rule on networks

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

Results
Cyclic feedback

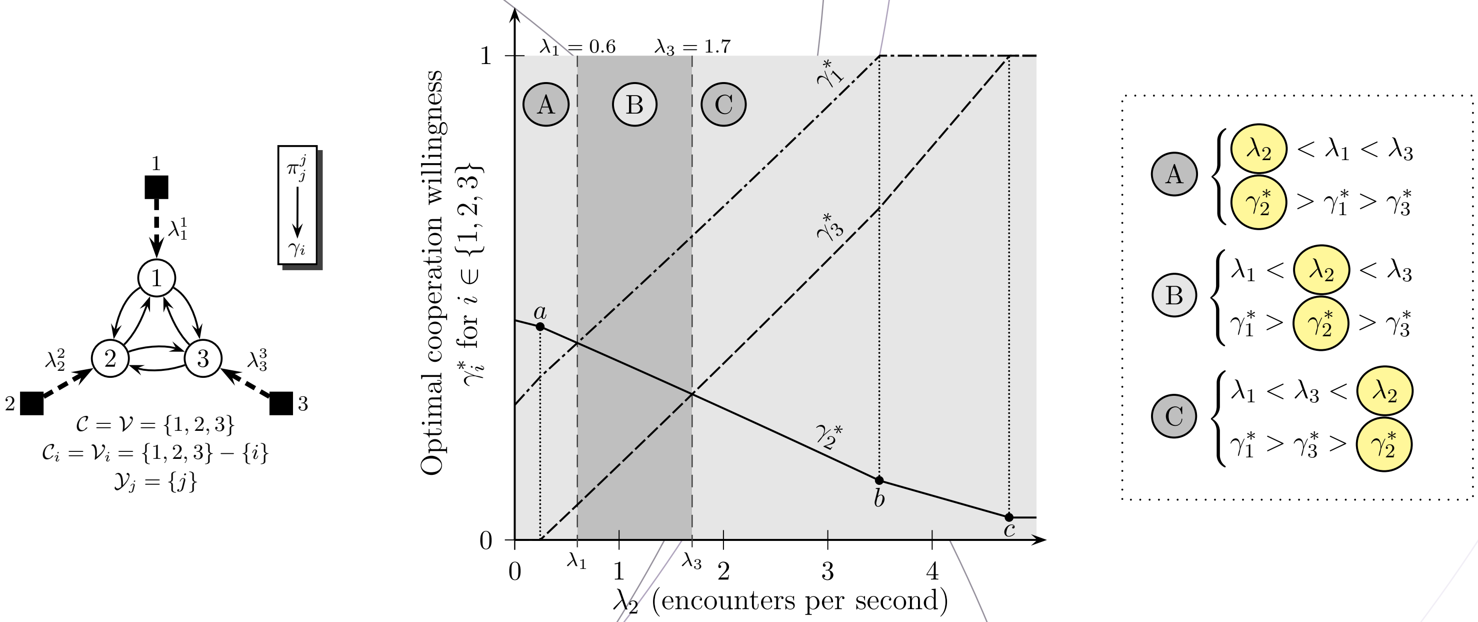


Equilibrium in AAV patrol scenario

- Simulation converges to predicted Nash equilibrium

Results

Cyclic feedback

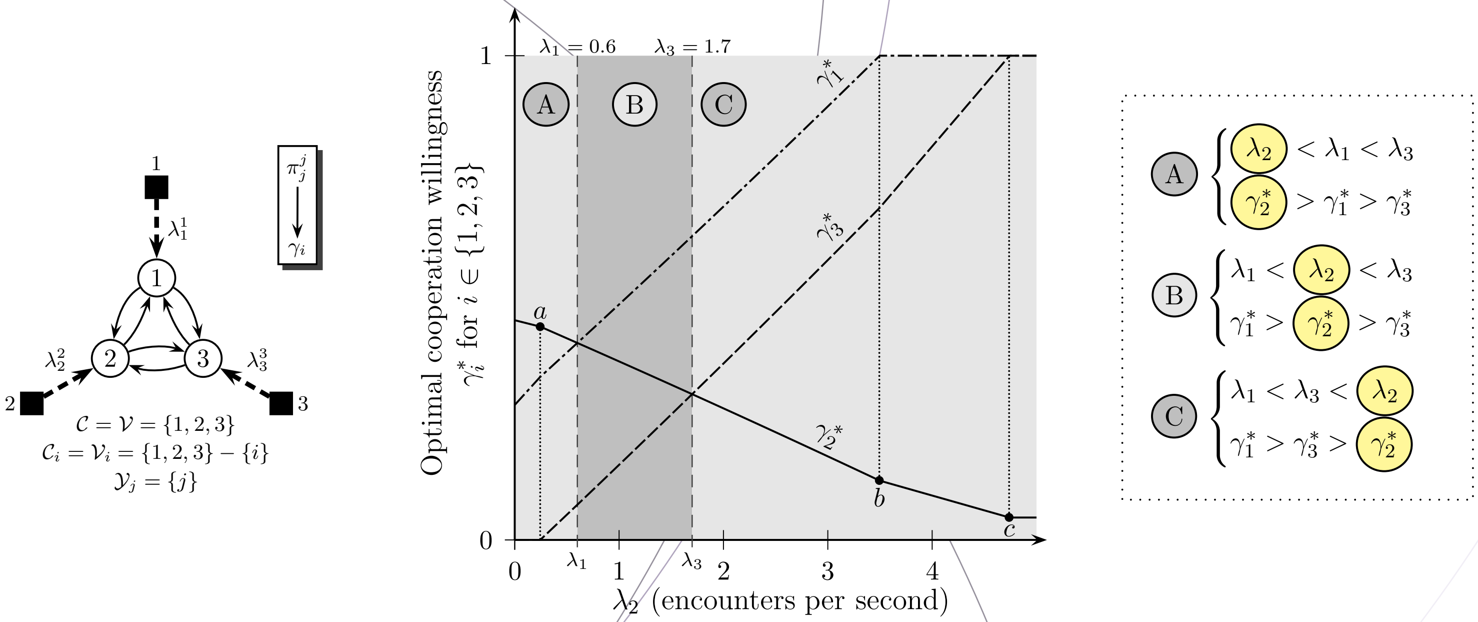


Equilibrium in AAV patrol scenario

- Simulation 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

Results

Cyclic feedback



Equilibrium in AAV patrol scenario

- Simulation 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
- Emergence due to market coupling from network cycles

MultiIFD: Distributed gradient descent for constrained optimization

- IFD, power dispatch, and nutrient constraints
- MultiIFD for intelligent lighting

Focal optimization problem

- Let $m, n \in \mathbb{N}$ and $\mathcal{X} \subseteq \mathbb{R}^n$ equipped with product topology.
- For each $j \in \{1, 2, \dots, m\}$, $\vec{a}_j \in \mathbb{R}^n$ and $c_j \in \mathbb{R}$.
- The focal optimization problem:

$$\begin{aligned} \text{minimize } & F(\vec{x}) \\ \text{subject to } & A\vec{x} \geq \vec{c} \end{aligned}$$

where $A \triangleq [\vec{a}_1, \vec{a}_2, \dots, \vec{a}_m]^\top$ and $\vec{c} \triangleq [c_1, c_2, \dots, c_m]^\top$.

Focal optimization problem

- Let $m, n \in \mathbb{N}$ and $\mathcal{X} \subseteq \mathbb{R}^n$ equipped with product topology.
- For each $j \in \{1, 2, \dots, m\}$, $\vec{a}_j \in \mathbb{R}^n$ and $c_j \in \mathbb{R}$.
- The focal optimization problem:

$$\begin{aligned} \text{minimize } & F(\vec{x}) \\ \text{subject to } & A\vec{x} \geq \vec{c} \end{aligned}$$

where $A \triangleq [\vec{a}_1, \vec{a}_2, \dots, \vec{a}_m]^\top$ and $\vec{c} \triangleq [c_1, c_2, \dots, c_m]^\top$.

- To show: Interdisciplinary connections.

Focal optimization problem

- Let $m, n \in \mathbb{N}$ and $\mathcal{X} \subseteq \mathbb{R}^n$ equipped with product topology.
- For each $j \in \{1, 2, \ldots, m\}$, $\vec{a}_j \in \mathbb{R}^n$ and $c_j \in \mathbb{R}$.
- The focal optimization problem:

$$\begin{aligned} \text{minimize } & F(\vec{x}) \\ \text{subject to } & A\vec{x} \geq \vec{c} \end{aligned}$$

where $A \triangleq [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_m]^\top$ and $\vec{c} \triangleq [c_1, c_2, \ldots, c_m]^\top$.

- To show: Interdisciplinary connections.
- Distributed parallel solvers for this problem are not trivial.
  - ☐ Usually parallelizable dual-space methods are effectively centralized.
  - ☐ To show: Amenable to parallelization in primal space (matches eusocial insects?).

Social foraging: the ideal free distribution (IFD)

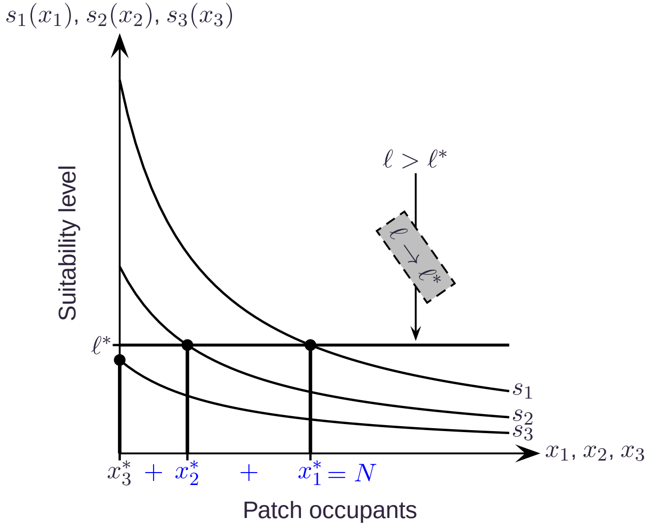
(Fretwell 1972; Fretwell and Lucas 1969; Stephens et al. 2007)

- Habitat in which animals self allocate according to IFD:
  - ☐ $N \in \mathbb{N}$ foragers *free* to move among $n \in \mathbb{N}$ locations.
  - ☐ The *ideal* forager knows the suitability $s_i(x_i)$ of each location $i \in \{1, 2, \dots, n\}$ with $x_i \in [0, N]$ occupants.
  - ☐ Suitabilities are monotonically decreasing.
  - ☐ Sufficiently small number of foragers continuously move away from lower suitability toward higher suitability.

Social foraging: the ideal free distribution (IFD)

(Fretwell 1972; Fretwell and Lucas 1969; Stephens et al. 2007)

- Graphical IFD with equilibrium suitability $\ell^* \in \mathbb{R}_{>0}$:



Social foraging: the ideal free distribution (IFD)

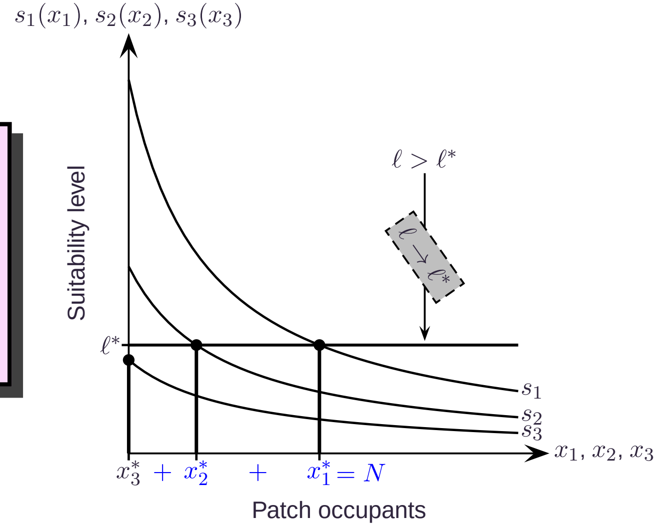
- Graphical IFD with equilibrium suitability $\ell^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$s_i(x_i^*) = \ell^*$$

or

$$s_i(0) < \ell^* \text{ and } x_i^* = 0$$



- Matches KKT characterization of solution to:

$$\text{maximize } \sum_{i=1}^{n} \int_0^{x_i} s_i(\tau)\, \mathrm{d}\tau$$

$$\text{subject to } x_1 + x_2 + \cdots + x_n = N$$

Social foraging: the ideal free distribution (IFD)

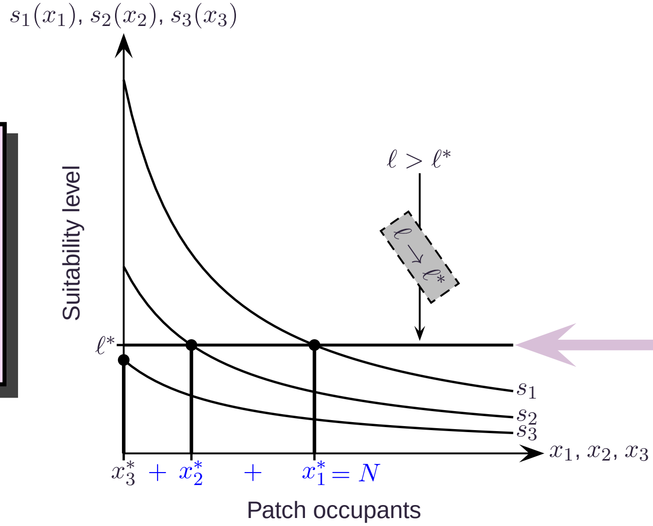
- Graphical IFD with equilibrium suitability $\ell^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$s_i(x_i^*) = \ell^*$$

or

$$s_i(0) < \ell^* \text{ and } x_i^* = 0$$



$\ell^* > 0$,
and so
an *inequality*
constraint is active

- Matches KKT characterization of solution to:

$$\text{maximize} \sum_{i=1}^{n} \int_0^{x_i} s_i(\tau) \, \mathrm{d}\tau$$

$$\text{subject to } x_1 + x_2 + \cdots + x_n \leq N$$

Social foraging: the ideal free distribution (IFD)

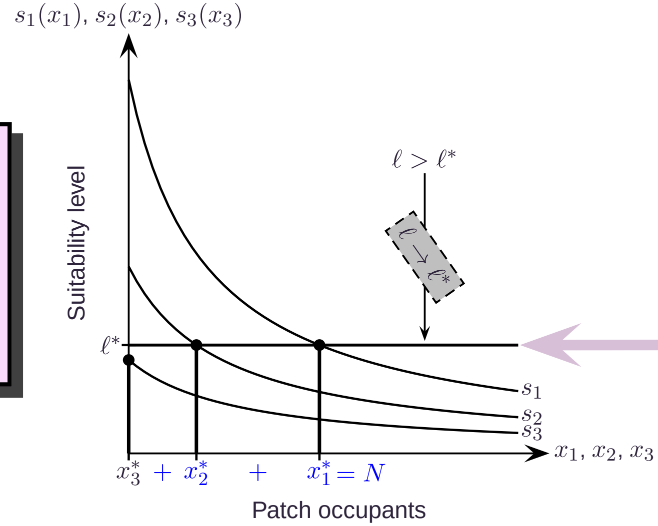
- Graphical IFD with equilibrium suitability $\ell^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$s_i(x_i^*) = \ell^*$$

or

$$s_i(0) < \ell^* \text{ and } x_i^* = 0$$



$\ell^* > 0$,
and so
an *inequality*
constraint is active

- Matches KKT characterization of solution to:

$$\text{maximize } \sum_{i=1}^{n} G_i(x_i)$$

$$\text{subject to } x_1 + x_2 + \cdots + x_n \leq N$$

Eusocial insects maximizing colony gain with up to N foragers?

Social foraging: the ideal free distribution (IFD)

- Economic formulation:
  - ☐ For $i \in \{1, 2, \ldots, n\}$ and $x_i \in \mathbb{R}_{\geq 0}$, let *price* $p_i(x_i) \triangleq 1/s_i(x_i)$.

Social foraging: the ideal free distribution (IFD)

- Economic formulation:
  - ☐ For $i \in \{1, 2, \dots, n\}$ and $x_i \in \mathbb{R}_{\geq 0}$, let *price* $p_i(x_i) \triangleq 1/s_i(x_i)$.
  - ☐ Prices are monotonically increasing.

Social foraging: the ideal free distribution (IFD)

- Economic formulation:
  - For $i \in \{1, 2, \dots, n\}$ and $x_i \in \mathbb{R}_{\geq 0}$, let *price* $p_i(x_i) \triangleq 1/s_i(x_i)$.
  - Prices are monotonically increasing.
- Foragers seeking highest suitability $\iff$ foragers seeking lowest price.

Social foraging: the ideal free distribution (IFD)

- Economic formulation:
  - ☐ For $i \in \{1, 2, \dots, n\}$ and $x_i \in \mathbb{R}_{\geq 0}$, let *price* $p_i(x_i) \triangleq 1/s_i(x_i)$.
  - ☐ Prices are monotonically increasing.
- Foragers seeking highest suitability $\iff$ foragers seeking lowest price.
- Price IFD with market price $\lambda^* \triangleq 1/\ell^*$:

$$\frac{1}{s_i(x_i^*)} = \boxed{p_i(x_i^*) = \lambda^*} = \frac{1}{\ell^*}$$

or

$$\frac{1}{s_i(0)} = \boxed{p_i(0) > \lambda^*} = \frac{1}{\ell^*} \quad \text{and} \quad x_i^* = 0$$

Social foraging: the ideal free distribution (IFD)

- Economic formulation:
  - ☐ For $i \in \{1, 2, \ldots, n\}$ and $x_i \in \mathbb{R}_{\geq 0}$, let *price* $p_i(x_i) \triangleq 1/s_i(x_i)$.
  - ☐ Prices are monotonically increasing.
- Foragers seeking highest suitability $\iff$ foragers seeking lowest price.
- Price IFD with market price $\lambda^* \triangleq 1/\ell^*$:

Occupied patch at market price.

Otherwise, entry price is too high.

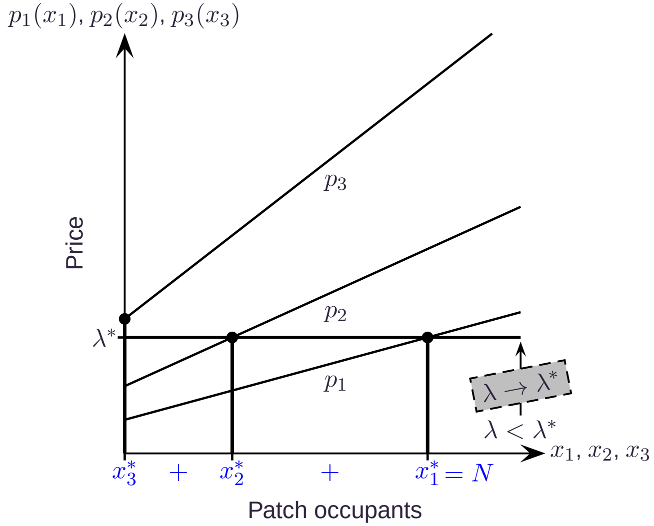
$$\frac{1}{s_i(x_i^*)} = \boxed{p_i(x_i^*) = \lambda^*} = \frac{1}{\ell *}$$

or

$$\frac{1}{s_i(0)} = \boxed{p_i(0) > \lambda^*} = \frac{1}{\ell *} \quad \text{and} \quad x_i^* = 0$$

Social foraging: the ideal free distribution (IFD)

- Graphical IFD with equilibrium price $\lambda^* \in \mathbb{R}_{>0}$:



Social foraging: the ideal free distribution (IFD)

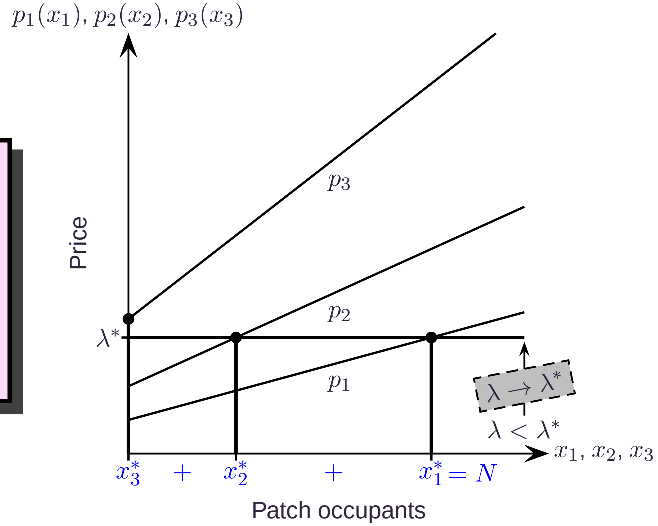
- Graphical IFD with equilibrium price $\lambda^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$p_i(x_i^*) = \lambda^*$$

or

$p_i(0) > \lambda^*$ and $x_i^* = 0$



- Matches KKT characterization of solution to:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \int_0^{x_i} p_i(\tau)\,\mathrm{d}\tau \\ \text{subject to} \quad & x_1 + x_2 + \cdots + x_n = N \end{aligned}$$

Social foraging: the ideal free distribution (IFD)

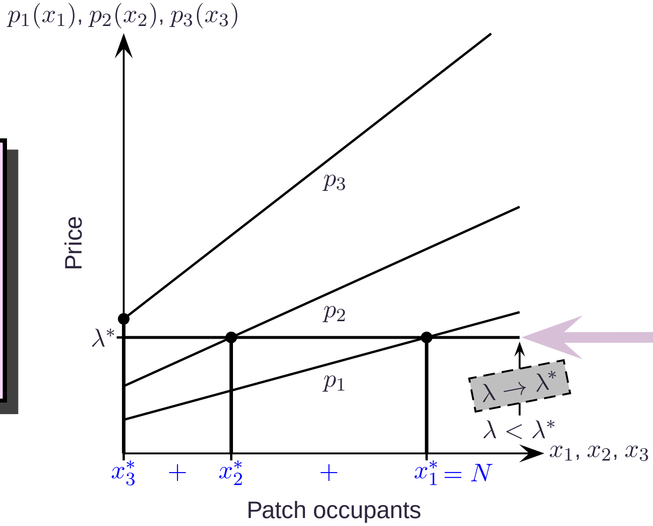
- Graphical IFD with equilibrium price $\lambda^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$p_i(x_i^*) = \lambda^*$$

or

$$p_i(0) > \lambda^* \text{ and } x_i^* = 0$$



$\lambda^* > 0,$
and so
an *inequality*
constraint is active

- Matches KKT characterization of solution to:

$$\text{minimize} \quad \sum_{i=1}^{n} \int_0^{x_i} p_i(\tau)\,\mathrm{d}\tau$$

$$\text{subject to} \quad x_1 + x_2 + \cdots + x_n \geq N$$

Social foraging: the ideal free distribution (IFD)

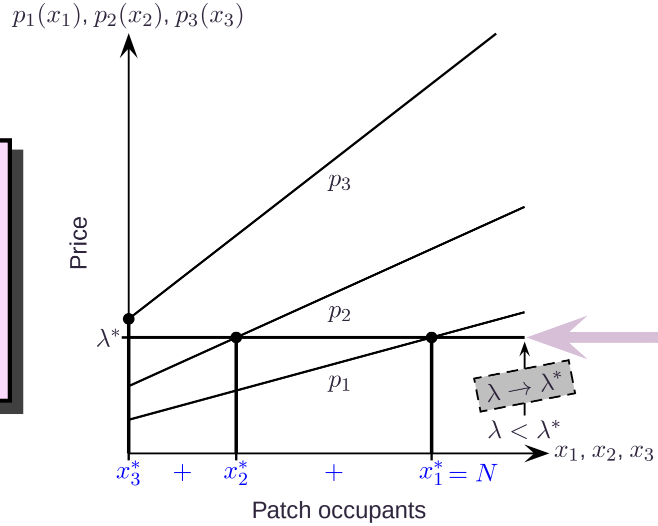
- Graphical IFD with equilibrium price $\lambda^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$p_i(x_i^*) = \lambda^*$$

or

$$p_i(0) > \lambda^* \text{ and } x_i^* = 0$$



$\lambda^* > 0$,
and so
an *inequality*
constraint is active

- Matches KKT characterization of solution to:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} C_i(x_i) \\ \text{subject to} \quad & x_1 + x_2 + \cdots + x_n \geq N \end{aligned}$$

Eusocial insects minimizing colony cost with at least N foragers?

Social foraging: the ideal free distribution (IFD)

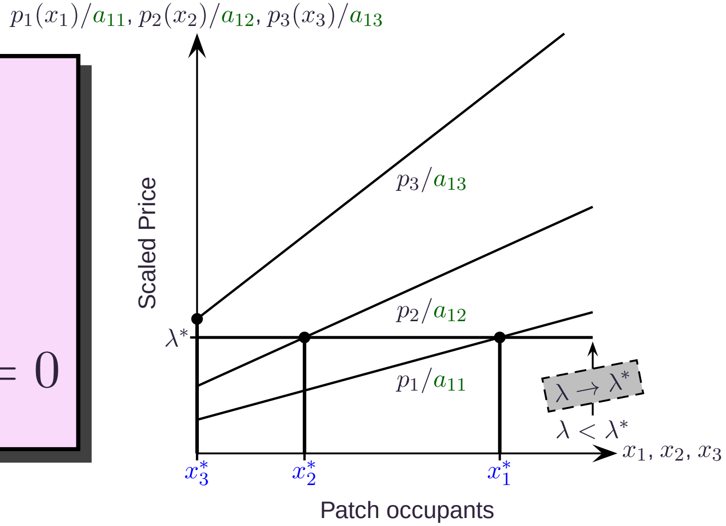
- Graphical IFD with equilibrium price $\lambda^* \in \mathbb{R}_{>0}$:

<ins>KKT conditions:</ins>

$$\frac{p_i(x_i^*)}{a_{1i}} = \lambda^*$$

or

$$\frac{p_i(0)}{a_{1i}} > \lambda^* \text{ and } x_i^* = 0$$



- Matches KKT characterization of solution to:

$$\text{minimize} \quad F(\vec{x}) \qquad \text{(e.g., } x_1 + \cdots + x_n\text{)}$$

$$\text{subject to} \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq c_1$$

Eusocial insects meeting nutrient constraint at lowest cost (e.g., total number of foragers)?

Social foraging: the ideal free distribution (IFD)

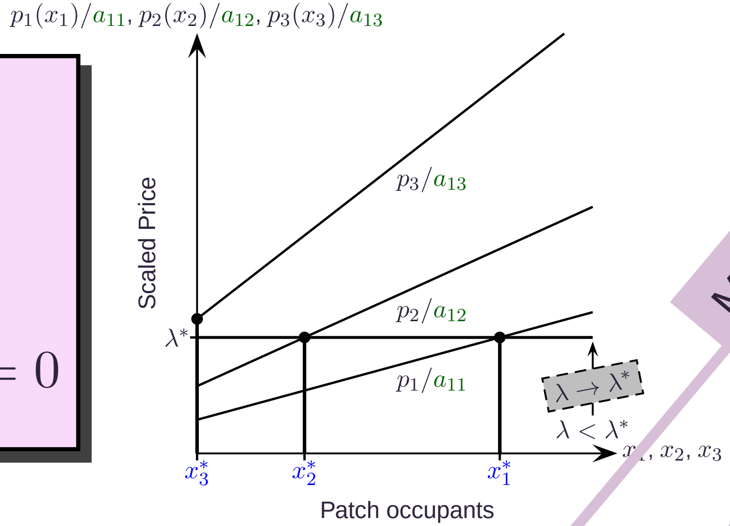
- Graphical IFD with equilibrium price $\lambda^* \in \mathbb{R}_{>0}$:

KKT conditions:

$$\frac{p_i(x_i^*)}{a_{1i}} = \lambda^*$$

or

$$\frac{p_i(0)}{a_{1i}} > \lambda^* \text{ and } x_i^* = 0$$



Matches focal problem with $m = 1$.

- Matches KKT characterization of solution to:

$$\text{minimize} \quad F(\vec{x}) \qquad \text{(e.g., } x_1 + \cdots + x_n\text{)}$$

$$\text{subject to} \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq c_1$$

Eusocial insects meeting nutrient constraint at lowest cost (e.g., total number of foragers)?

Social foraging: the ideal free distribution (IFD)

- For example, if $F(\vec{x}) = \|\vec{x}\|_2^2 = x_1^2 + \cdots + x_n^2$, then $x_i^* > 0$ for all $i \in \{1, 2, \ldots, n\}$; in particular,

$$x_i^* = c_1 \frac{a_{1i}}{{a_{11}}^2 + {a_{12}}^2 + \cdots + {a_{1n}}^2}$$

$$\frac{p_i(x_i^*)}{a_{1i}} = \frac{\nabla_i F(x_i^*)}{a_{1i}} = \frac{2c_1}{{a_{11}}^2 + {a_{12}}^2 + \cdots + {a_{1n}}^2} = \lambda^*$$

where price $p_i(x_i) = 2x_i$.

Equilibrium distribution for

$$F(\vec{x}) = \|\vec{x}\|_1 = x_1 + x_2 + \cdots + x_n$$

allocates all foragers to patch

$$\arg\max\{a_{1i} : i \in \{1, 2, \ldots, n\}\}.$$

It may be valuable to increase spread of distribution.

Social foraging: the ideal free distribution (IFD)

- For example, if $F(\vec{x}) = \|\vec{x}\|_2^2 = x_1^2 + \cdots + x_n^2$, then $x_i^* > 0$ for all $i \in \{1, 2, \ldots, n\}$; in particular,

$$x_i^* = c_1 \frac{a_{1i}}{a_{11}{}^2 + a_{12}{}^2 + \cdots + a_{1n}{}^2}$$

$$\frac{p_i(x_i^*)}{a_{1i}} = \frac{\nabla_i F(x_i^*)}{a_{1i}} = \frac{2c_1}{a_{11}{}^2 + a_{12}{}^2 + \cdots + a_{1n}{}^2} = \lambda^*$$

where price $p_i(x_i) = 2x_i$.

- Equilibrium distribution matches classical IFD with

$$s_i(x_i) = \frac{a_{1i}}{2x_i} \quad \text{and} \quad N = c_1 \frac{a_{11} + a_{12} + \cdots + a_{1n}}{a_{11}{}^2 + a_{12}{}^2 + \cdots + a_{1n}{}^2} \quad \text{and} \quad \ell^* = \frac{1}{\lambda^*}$$

for $i \in \{1, 2, \ldots, n\}$.

Social foraging: the ideal free distribution (IFD)

- For example, if $F(\vec{x}) = \|\vec{x}\|_2^2 = x_1^2 + \cdots + x_n^2$, then $x_i^* > 0$ for all $i \in \{1, 2, \dots, n\}$; in particular,

$$x_i^* = c_1 \frac{a_{1i}}{{a_{11}}^2 + {a_{12}}^2 + \cdots + {a_{1n}}^2}$$

$$\frac{p_i(x_i^*)}{a_{1i}} = \frac{\nabla_i F(x_i^*)}{a_{1i}} = \frac{2c_1}{{a_{11}}^2 + {a_{12}}^2 + \cdots + {a_{1n}}^2} = \lambda^*$$

  where price $p_i(x_i) = 2x_i$.

- Equilibrium distribution matches classical IFD with

$$s_i(x_i) = \frac{a_{1i}}{2x_i} \quad \text{and} \quad N = c_1 \frac{a_{11} + a_{12} + \cdots + a_{1n}}{{a_{11}}^2 + {a_{12}}^2 + \cdots + {a_{1n}}^2} \quad \text{and} \quad \ell^* = \frac{1}{\lambda^*}$$

  for $i \in \{1, 2, \dots, n\}$.

- So nutrient-constrained cost-minimizing IFD modulates necessary N with constraints and environment.

Social foraging: the ideal free distribution (IFD)

- Where there's one nutrient, there may be others. Let $m > 1$:

$$\begin{array}{rl} \text{minimize} & F(\vec{x}) \\ \text{subject to} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq c_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq c_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq c_m \end{array}$$

Hence, the IFD with multiple nutrient constraints is the focal optimization problem.

Social foraging: the ideal free distribution (IFD)

- Where there's one nutrient, there may be others. Let $m > 1$:

$$\begin{array}{rr} \text{minimize} & F(\vec{x}) \\ \text{subject to} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq c_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq c_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq c_m \end{array}$$

  Hence, the IFD with multiple nutrient constraints is the focal optimization problem.

- KKT does not imply uniform suitability/price equilibrium. For $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$, gradient oblique to $\vec{a}_j$:

$$\nabla_i F(\vec{x}^*) = \overbrace{\lambda_1^* a_{1i} + \lambda_2^* a_{2i} + \cdots + \lambda_m^* a_{mi}}^{\text{Active constraint support}} + \overbrace{\mu_i^* - \nu_i^*}^{\text{Truncation support}}$$

Social foraging: the ideal free distribution (IFD)

- Where there's one nutrient, there may be others. Let $m > 1$:

$$\begin{aligned} \text{minimize} \quad & F(\vec{x}) \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq c_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq c_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq c_m \end{aligned}$$

Hence, the IFD with multiple nutrient constraints is the focal op

Impact on observations of foraging distributions?

- KKT does not imply uniform suitability/price equilibrium. For $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$, gradient oblique to $\vec{a}_j$:

$$\nabla_i F(\vec{x}^*) = \overbrace{\lambda_1^* a_{1i} + \lambda_2^* a_{2i} + \cdots + \lambda_m^* a_{mi}}^{\text{Active constraint support}} + \overbrace{\mu_i^* - \nu_i^*}^{\text{Truncation support}}$$

Economic power dispatch

(Bergen and Vittal 2000)

- Classic problem in distributed power generation:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} C_i(P_i) \\ \text{subject to} \quad & P_1 + P_2 + \cdots + P_n = P_D \end{aligned}$$

for $n \in \mathbb{N}$ where generator $i \in \{1, 2, \dots, n\}$ contributes P_i power to P_D power demanded at a generator cost of $C_i(P_i)$.

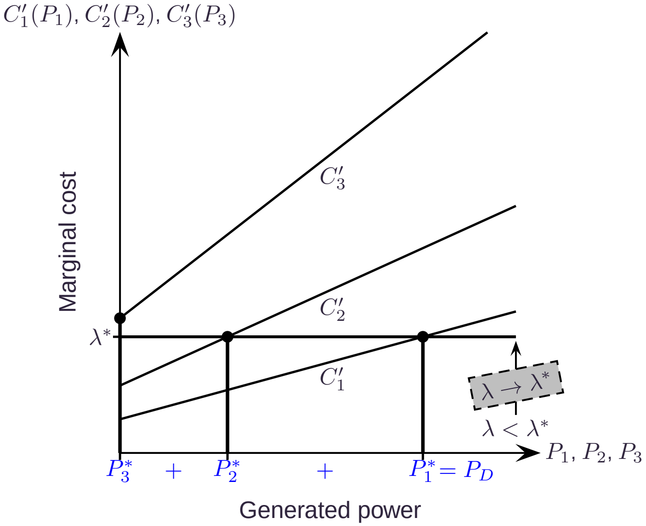
Economic power dispatch
(Bergen and Vittal 2000)

- Classic problem in distributed power generation:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} C_i(P_i) \\ \text{subject to} \quad & P_1 + P_2 + \cdots + P_n = P_D \end{aligned}$$

for $n \in \mathbb{N}$ where generator $i \in \{1, 2, \ldots, n\}$ contributes P_i power to P_D power demanded at a generator cost of $C_i(P_i)$.

- Graphical solution described by Bergen and Vittal (2000):



Economic power dispatch

(Bergen and Vittal 2000)

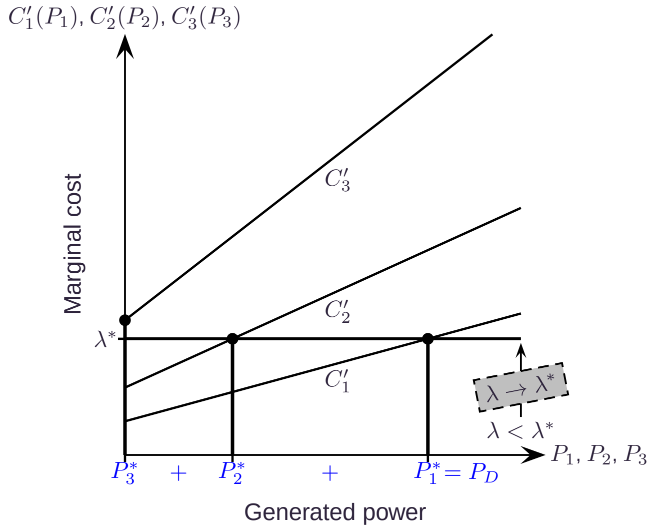
- Classic problem in distributed power generation:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} C_i(P_i) \\ \text{subject to} \quad & P_1 + P_2 + \cdots + P_n = P_D \end{aligned}$$

  for $n \in \mathbb{N}$ where generator $i \in \{1, 2, \ldots, n\}$ contributes P_i power to P_D power demanded at a generator cost of $C_i(P_i)$.

- Graphical solution described by Bergen and Vittal (2000):

Identical to single-constraint price-minimization IFD



Economic power dispatch

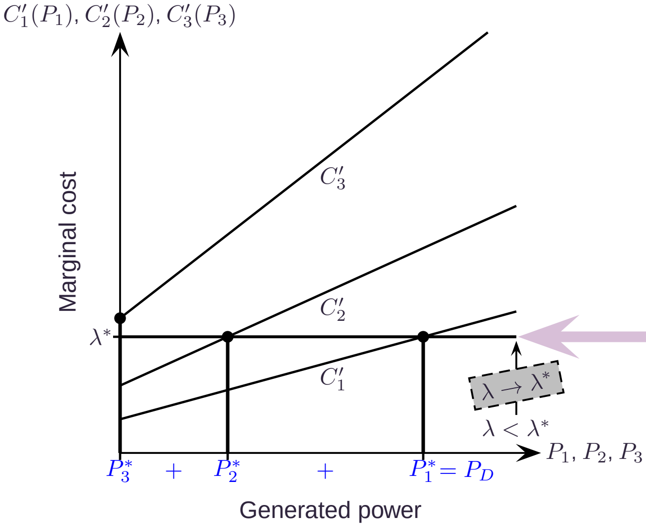
- Classic problem in distributed power generation:

$$\text{minimize} \quad \sum_{i=1}^{n} C_i(P_i)$$

$$\text{subject to} \quad P_1 + P_2 + \cdots + P_n \geq P_D$$

for $n \in \mathbb{N}$ where generator $i \in \{1, 2, \ldots, n\}$ contributes P_i power to P_D power demanded at a generator cost of $C_i(P_i)$.

- Graphical solution described by Bergen and Vittal (2000).



Identical to single-constraint price-minimization IFD

$\lambda^* > 0$, and so an *inequality* constraint is active

Economic power dispatch

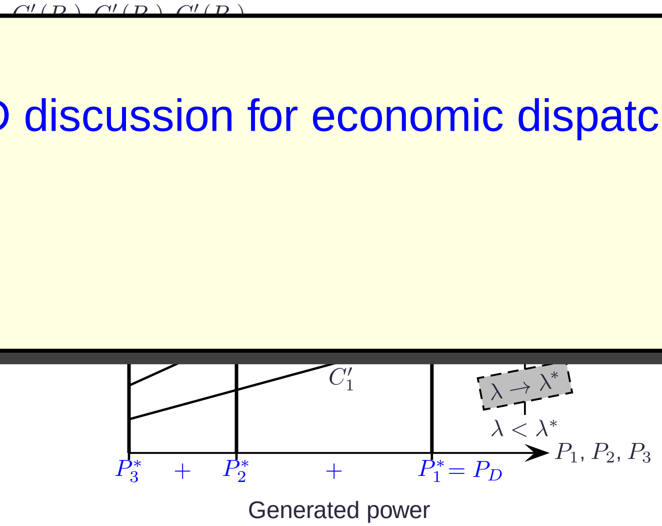
- Classic problem in distributed power generation:

$$\text{minimize} \quad \sum_{i=1}^{n} C_i(P_i)$$

$$\text{subject to} \quad P_1 + P_2 + \cdots + P_n \geq P_D$$

for $n \in \mathbb{N}$ where generator $i \in \{1, 2, \dots, n\}$ contributes P_i power to P_D power demanded at a generator cost of $C_i(P_i)$.

- Graphical solution described by Bergen and Vittal (2000).



si price-minimization IFD

an *inequality* constraint is active

(repeat IFD discussion for economic dispatch problem)

Economic power dispatch

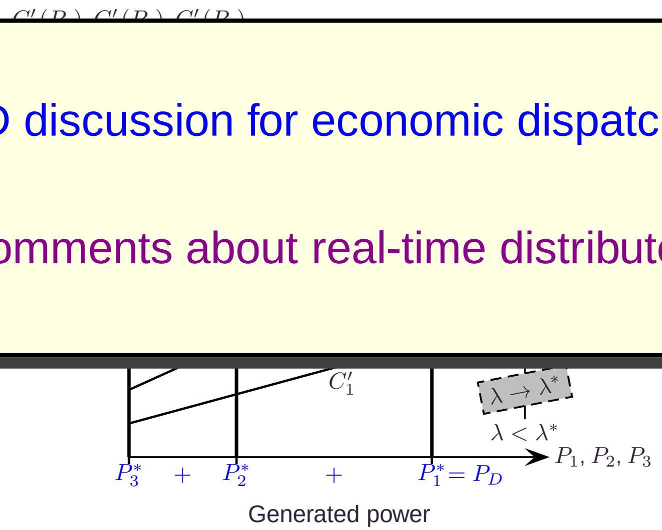
- Classic problem in distributed power generation:

$$\text{minimize} \quad \sum_{i=1}^{n} C_i(P_i)$$

$$\text{subject to} \quad P_1 + P_2 + \cdots + P_n \geq P_D$$

for $n \in \mathbb{N}$ where generator $i \in \{1, 2, \ldots, n\}$ contributes P_i power to P_D power demanded at a generator cost of $C_i(P_i)$.

- Graphical solution described by Bergen and Vittal (2000).



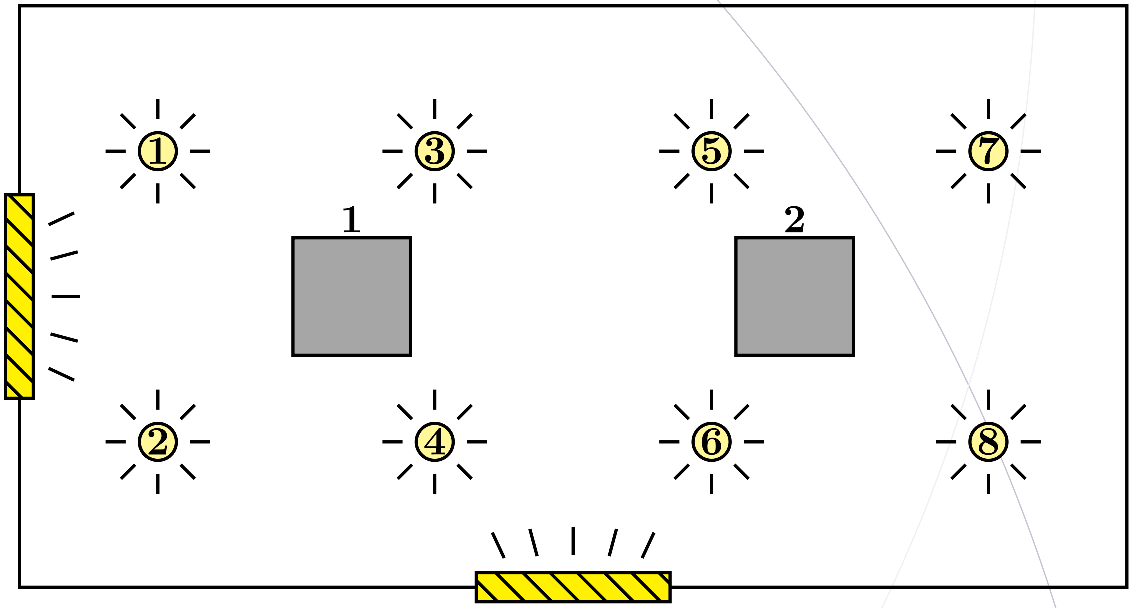
(repeat IFD discussion for economic dispatch problem)

[augment with comments about real-time distributed optimization]

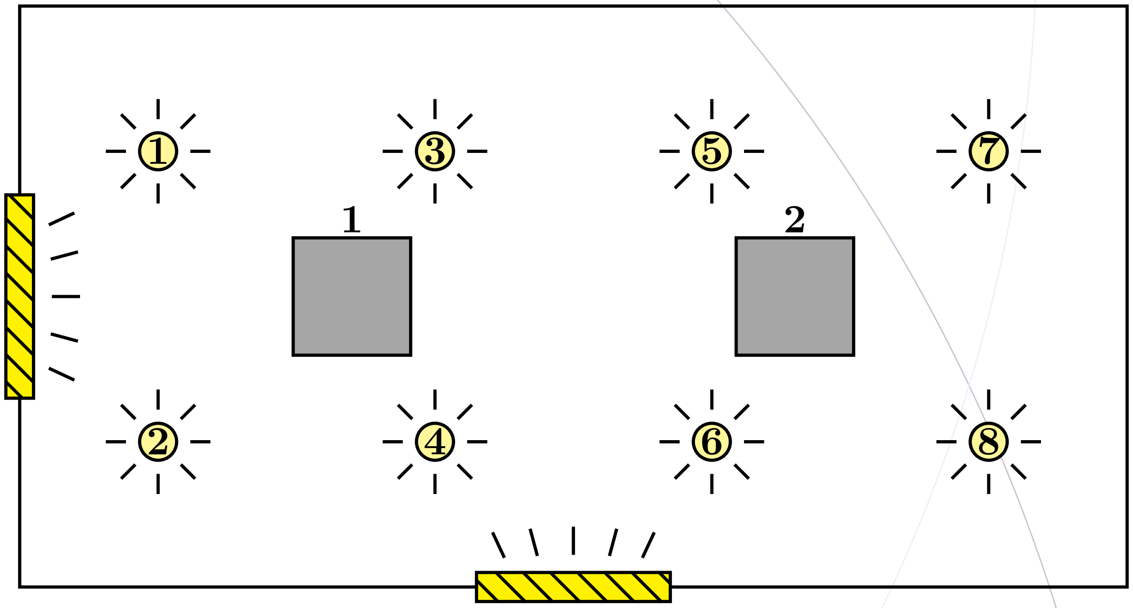
si
price-minimization IFD

an *inequality*
constraint is active

Intelligent lights



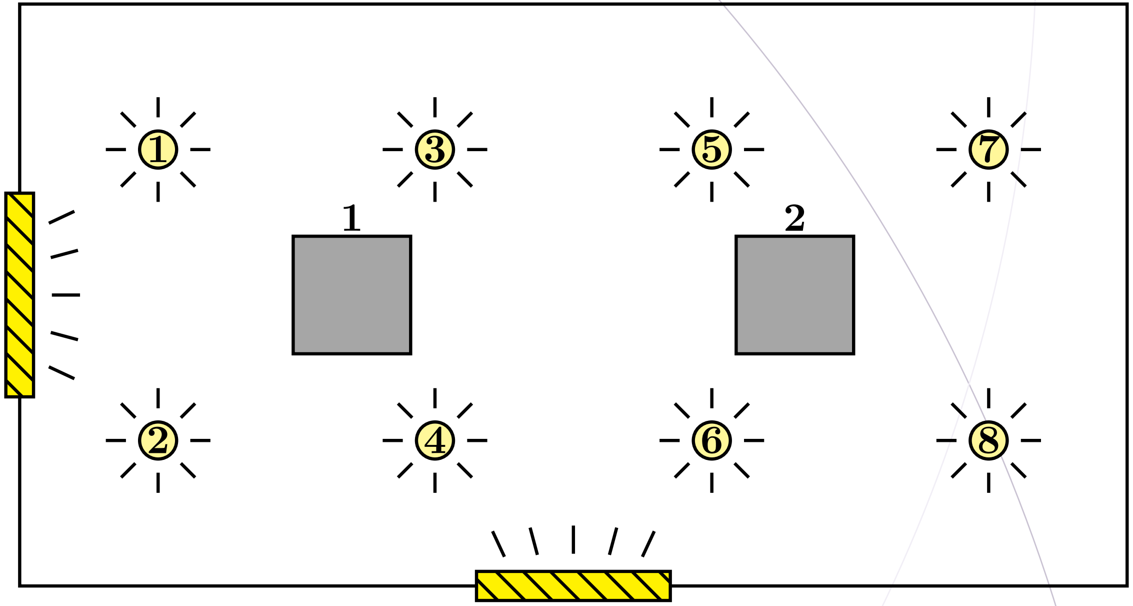
Intelligent lights
(e.g., Wen and Agogino 2008)



- There are $n \in \mathbb{N}$ lights and $m \in \mathbb{N}$ sensors.

Intelligent lights

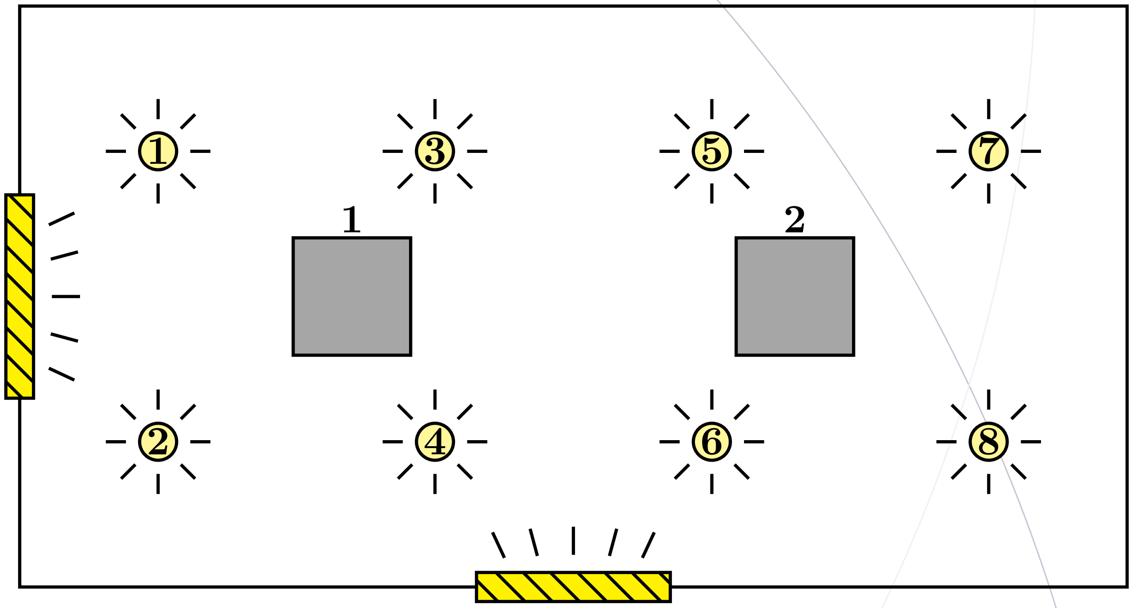
(e.g., Wen and Agogino 2008)



- There are $n \in \mathbb{N}$ lights and $m \in \mathbb{N}$ sensors.
- For each $i \in \{1, 2, \ldots, n\}$, x_i is control signal for i^{th} light.

Intelligent lights

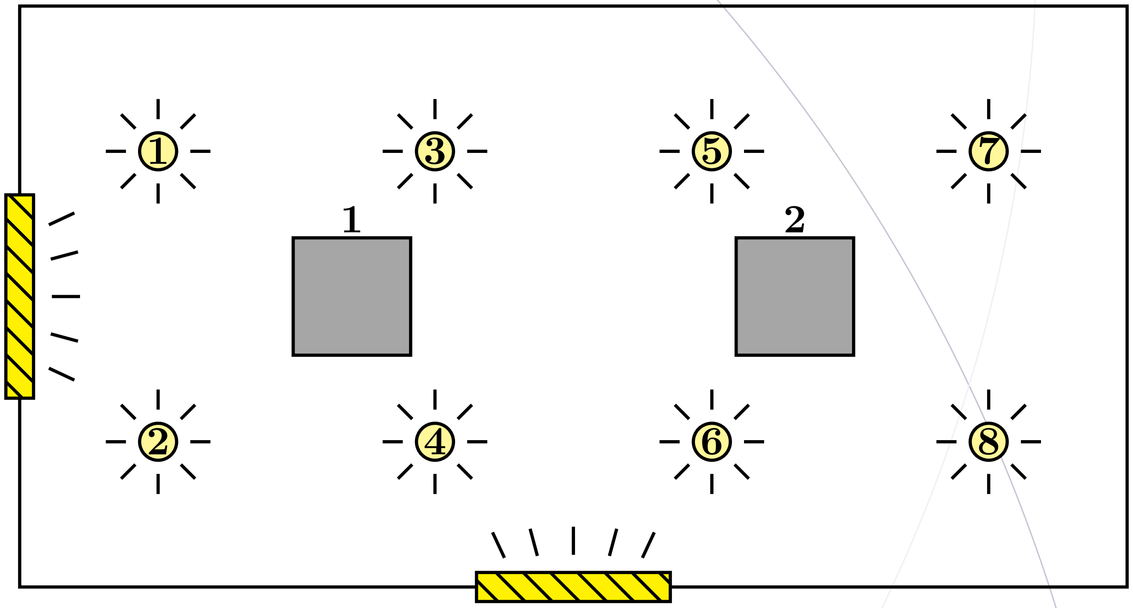
(e.g., Wen and Agogino 2008)



- There are $n \in \mathbb{N}$ lights and $m \in \mathbb{N}$ sensors.
- For each $i \in \{1, 2, \ldots, n\}$, x_i is control signal for i^{th} light.
- Lighting-control–photosensor-reading maps are approximately linear.

Intelligent lights

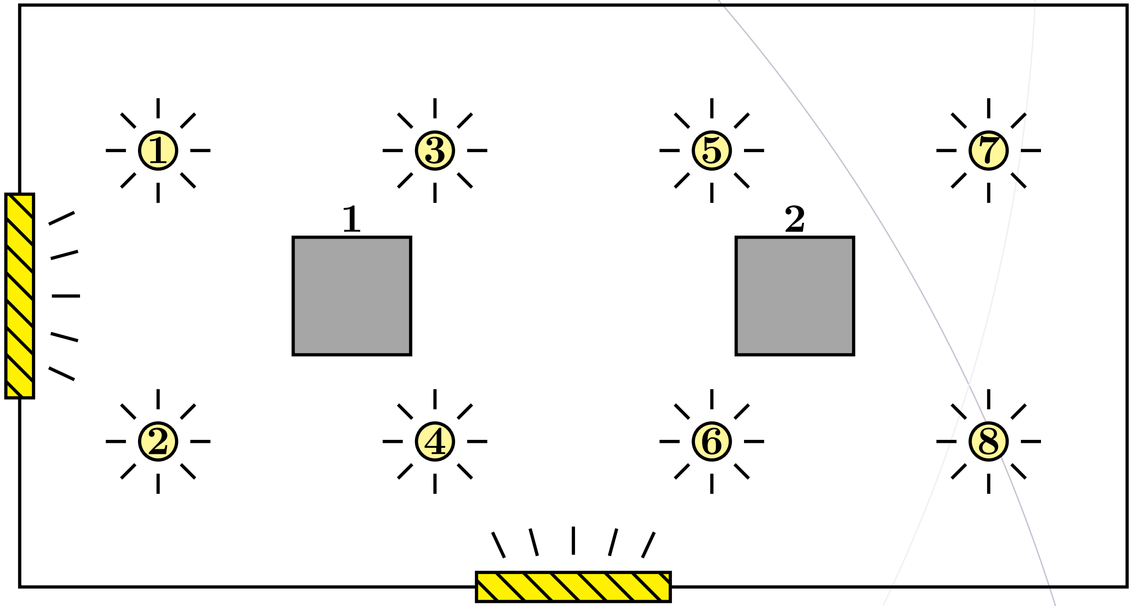
(e.g., Wen and Agogino 2008)



- There are $n \in \mathbb{N}$ lights and $m \in \mathbb{N}$ sensors.
- For each $i \in \{1, 2, \ldots, n\}$, x_i is control signal for i^{th} light.
- Lighting-control–photosensor-reading maps are approximately linear.
- Slow disturbance sources (e.g., windows) exist and can be harvested.

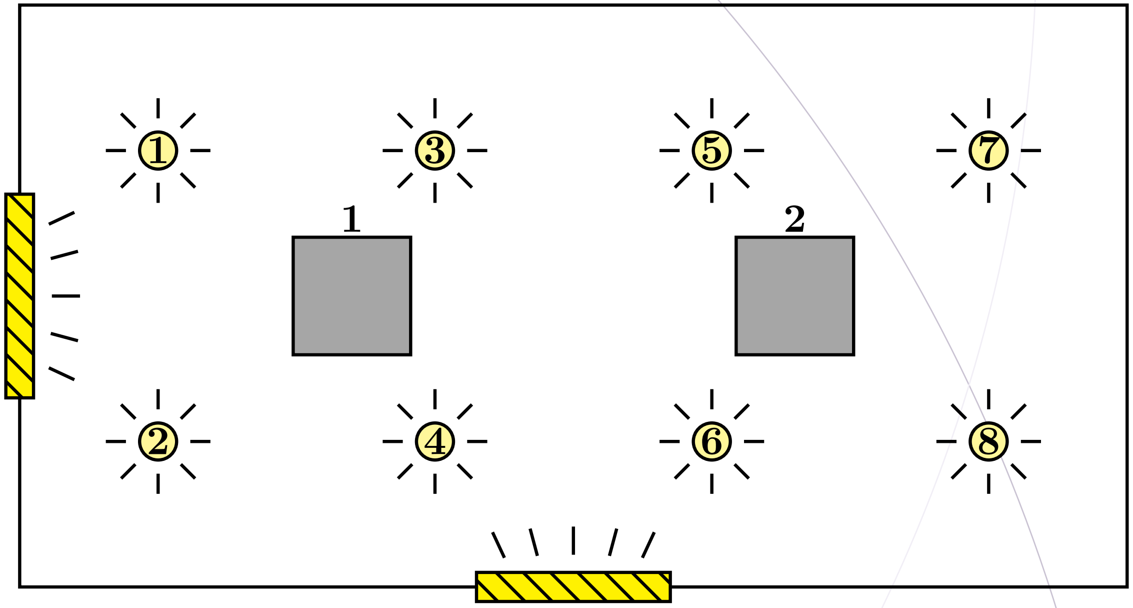
Intelligent lights

(e.g., Wen and Agogino 2008)



- There are $n \in \mathbb{N}$ lights and $m \in \mathbb{N}$ sensors.
- For each $i \in \{1, 2, \ldots, n\}$, x_i is control signal for i^{th} light.
- Lighting-control–photosensor-reading maps are approximately linear.
- Slow disturbance sources (e.g., windows) exist and can be harvested.
- Meet constraints at each sensor using reduced power.

Intelligent lights



Again:

$$\begin{aligned} \text{minimize} \quad & F(\vec{x}) \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq c_1 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq c_m \end{aligned}$$

Each sensor is a nutrient; each light is a food patch.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.
- Distributed approach here for convex F with monotonic ∇F:

Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.
- Distributed approach here for convex F with monotonic ∇F:
  - Real-time solutions are allowed to violate constraints.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.
- Distributed approach here for convex F with monotonic ∇F:
  - ☐ Real-time solutions are allowed to violate constraints.
  - ☐ $n \in \mathbb{N}$ agents act independently to reduce x_i for each $i \in \{1, 2, \ldots, n\}$ (monotonicity $\implies$ reduced cost).

Distributed solver

Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.
- Distributed approach here for convex F with monotonic ∇F:
  - ☐ Real-time solutions are allowed to violate constraints.
  - ☐ $n \in \mathbb{N}$ agents act independently to reduce x_i for each $i \in \{1, 2, \dots, n\}$ (monotonicity $\implies$ reduced cost).
  - ☐ $m \in \mathbb{N}$ independent agents provide response surfaces to support constrained equilibrium.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.
- Distributed approach here for convex F with monotonic ∇F:
  - ☐ Real-time solutions are allowed to violate constraints.
  - ☐ $n \in \mathbb{N}$ agents act independently to reduce x_i for each $i \in \{1, 2, \dots, n\}$ (monotonicity $\implies$ reduced cost).
  - ☐ $m \in \mathbb{N}$ independent agents provide response surfaces to support constrained equilibrium.
  - ☐ Interaction between agents on constraint causes trajectories to slide along constraint boundary toward equilibrium.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destroyed in dual space. Sequential iterations yield no parallelization.
- Distributed approach here for convex F with monotonic ∇F:
  - ☐ Real-time solutions are allowed to violate constraints.
  - ☐ $n \in \mathbb{N}$ agents act independently to reduce x_i for each $i \in \{1, 2, \dots, n\}$ (monotonicity $\implies$ reduced cost).
  - ☐ $m \in \mathbb{N}$ independent agents provide response surfaces to support constrained equilibrium.
  - ☐ Interaction between agents on constraint causes trajectories to slide along constraint boundary toward equilibrium.
  - ☐ Trajectories reach invariant bounded neighborhood of optimum $\vec{x}^*$.

Distributed solver
Motivation

- Especially for intelligent light case, *distributed* solvers are desired.
- Constraint set $\{\vec{x} \in \mathcal{X} : A\vec{x} \geq \vec{c}\}$ is non-separable polyhedron in general.
- Dual problem has separable constraint set $\mathbb{R}^m$, but sparsity in A is often destr o parallelization.

System state $\vec{x}$ is stigmergic memory.

- Distributed approach here for convex F with monotonic ∇F:
  - ☐ Real-time solutions are allowed to violate constraints.
  - ☐ $n \in \mathbb{N}$ agents act independently to reduce x_i for each $i \in \{1, 2, \dots, n\}$ (monotonicity $\implies$ reduced cost).
  - ☐ $m \in \mathbb{N}$ independent agents provide response surfaces to support constrained equilibrium.
  - ☐ Interaction between agents on constraint causes trajectories to slide along constraint boundary toward equilibrium.
  - ☐ Trajectories reach invariant bounded neighborhood of optimum $\vec{x}^*$.

Distributed solver
Some algorithm details

- A *MultiIFD* discrete-time realization with sufficiently small parameter δ:
  - ☐ For each $i \in \{1, 2, \dots, n\}$,

$$x_i^+ = x_i - \delta.$$

  - ☐ For each $j \in \{1, 2, \dots, m\}$,

$$\vec{x}^+ = \vec{x} + \begin{cases} \sigma_j \vec{v}_j & \text{if } \vec{a}_j^\top \vec{x} \leq c_j \\ 0 & \text{otherwise} \end{cases}$$

where MultiIFD direction

$$\vec{v}_j = \left[\frac{a_{j1}}{\nabla_1 F(\vec{x})}, \quad \frac{a_{j2}}{\nabla_2 F(\vec{x})}, \quad \dots, \quad \frac{a_{jn}}{\nabla_n F(\vec{x})} \right]^\top$$

and

$$\sigma_j = \frac{c_j - \vec{a}_j^\top \vec{x}}{\vec{a}_j^\top \vec{v}_j}.$$

Distributed solver
Some algorithm details

- A *MultiIFD* discrete-time realization with sufficiently small parameter δ:
  - ☐ For each $i \in \{1, 2, \dots, n\}$,

    "Animals return from patches" →
    $$x_i^+ = x_i - \delta.$$
  - ☐ For each $j \in \{1, 2, \dots, m\}$,

    $$\vec{x}^+ = \vec{x} + \begin{cases} \sigma_j \vec{v}_j & \text{if } \vec{a}_j^\top \vec{x} \leq c_j \\ 0 & \text{otherwise} \end{cases}$$

    where MultiIFD direction

    $$\vec{v}_j = \left[\frac{a_{j1}}{\nabla_1 F(\vec{x})}, \quad \frac{a_{j2}}{\nabla_2 F(\vec{x})}, \quad \dots, \quad \frac{a_{jn}}{\nabla_n F(\vec{x})} \right]^\top$$

    and

    $$\sigma_j = \frac{c_j - \vec{a}_j^\top \vec{x}}{\vec{a}_j^\top \vec{v}_j}.$$

Distributed solver
Some algorithm details

- A *MultiIFD* discrete-time realization with sufficiently small parameter δ:
  - For each $i \in \{1, 2, \ldots, n\}$,

    "Animals return from patches" → $x_i^+ = x_i - \delta.$
  - For each $j \in \{1, 2, \ldots, m\}$,

    "Foragers dispatched for a nutrient" → $\vec{x}^+ = \vec{x} + \begin{cases} \sigma_j \vec{v}_j & \text{if } \vec{a}_j^\top \vec{x} \leq c_j \\ 0 & \text{otherwise} \end{cases}$

    where MultiIFD direction

$$\vec{v}_j = \left[\frac{a_{j1}}{\nabla_1 F(\vec{x})}, \quad \frac{a_{j2}}{\nabla_2 F(\vec{x})}, \quad \ldots, \quad \frac{a_{jn}}{\nabla_n F(\vec{x})} \right]^\top$$

    and

$$\sigma_j = \frac{c_j - \vec{a}_j^\top \vec{x}}{\vec{a}_j^\top \vec{v}_j}.$$

Distributed solver
Some algorithm details

- A *MultiIFD* discrete-time realization with sufficiently small parameter δ:
  - ☐ For each $i \in \{1, 2, \ldots, n\}$,

“Animals return from patches” →

$$x_i^+ = x_i - \delta.$$

  - ☐ For each $j \in \{1, 2, \ldots, m\}$,

“Foragers dispatched for a nutrient” →

$$\vec{x}^+ = \vec{x} + \begin{cases} \sigma_j \vec{v}_j & \text{if } \vec{a}_j^\top \vec{x} \leq c_j \\ 0 & \text{otherwise} \end{cases}$$

where MultiIFD direction

Dispatch according to n *suitabilities* →

$$\vec{v}_j = \left[\frac{a_{j1}}{\nabla_1 F(\vec{x})}, \quad \frac{a_{j2}}{\nabla_2 F(\vec{x})}, \quad \ldots, \quad \frac{a_{jn}}{\nabla_n F(\vec{x})} \right]^\top$$

and

$$\sigma_j = \frac{c_j - \vec{a}_j^\top \vec{x}}{\vec{a}_j^\top \vec{v}_j}.$$

Distributed solver
Some algorithm details

- A *MultiIFD* discrete-time realization with sufficiently small parameter δ:
  - For each $i \in \{1, 2, \dots, n\}$,

    "Animals return from patches"

    $$x_i^+ = x_i - \delta.$$

  - For each $j \in \{1, 2, \dots, m\}$,

    "Foragers dispatched for a nutrient"

    $$\vec{x}^+ = \vec{x} + \begin{cases} \sigma_j \vec{v}_j & \text{if } \vec{a}_j^\top \vec{x} \leq c_j \\ 0 & \text{otherwise} \end{cases}$$

    where MultiIFD direction

    Dispatch according to n *suitabilities*

    $$\vec{v}_j = \left[\frac{a_{j1}}{\nabla_1 F(\vec{x})}, \quad \frac{a_{j2}}{\nabla_2 F(\vec{x})}, \quad \dots, \quad \frac{a_{jn}}{\nabla_n F(\vec{x})} \right]^\top$$

    and

    Reach constraint surface

    $$\sigma_j = \frac{c_j - \vec{a}_j^\top \vec{x}}{\vec{a}_j^\top \vec{v}_j}.$$

Simulation results

- Theoretical analysis predicts invariant hypercorners.

Simulation results

- Theoretical analysis predicts invariant hypercorners.
  - For single-constraint case, system enters invariant corners of hypercube with vertex anchored in constraint hyperplane.

Simulation results

- Theoretical analysis predicts invariant hypercorners.
    - ☐ For single-constraint case, system enters invariant corners of hypercube with vertex anchored in constraint hyperplane.
    - ☐ Hypercube slides along constraint toward fixed hypercorners.

Simulation results

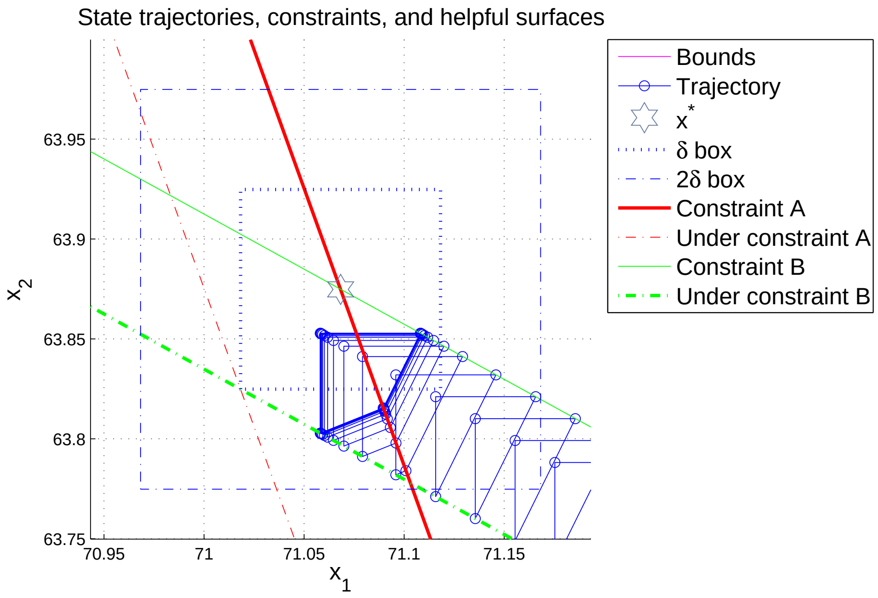
- Theoretical analysis predicts invariant hypercorners.
  - For single-constraint case, system enters invariant corners of hypercube with vertex anchored in constraint hyperplane.
  - Hypercube slides along constraint toward fixed hypercorners.
  - Ultimate error bounds from by fixed set location (2δ for $\| \cdot \|_2^2$).

Simulation results

- Theoretical analysis predicts invariant hypercorners.
  - ☐ For single-constraint case, system enters invariant corners of hypercube with vertex anchored in constraint hyperplane.
  - ☐ Hypercube slides along constraint toward fixed hypercorners.
  - ☐ Ultimate error bounds from by fixed set location (2δ for $||\cdot||_2^2$).
- For multiple active constraints, hypercubes interact:

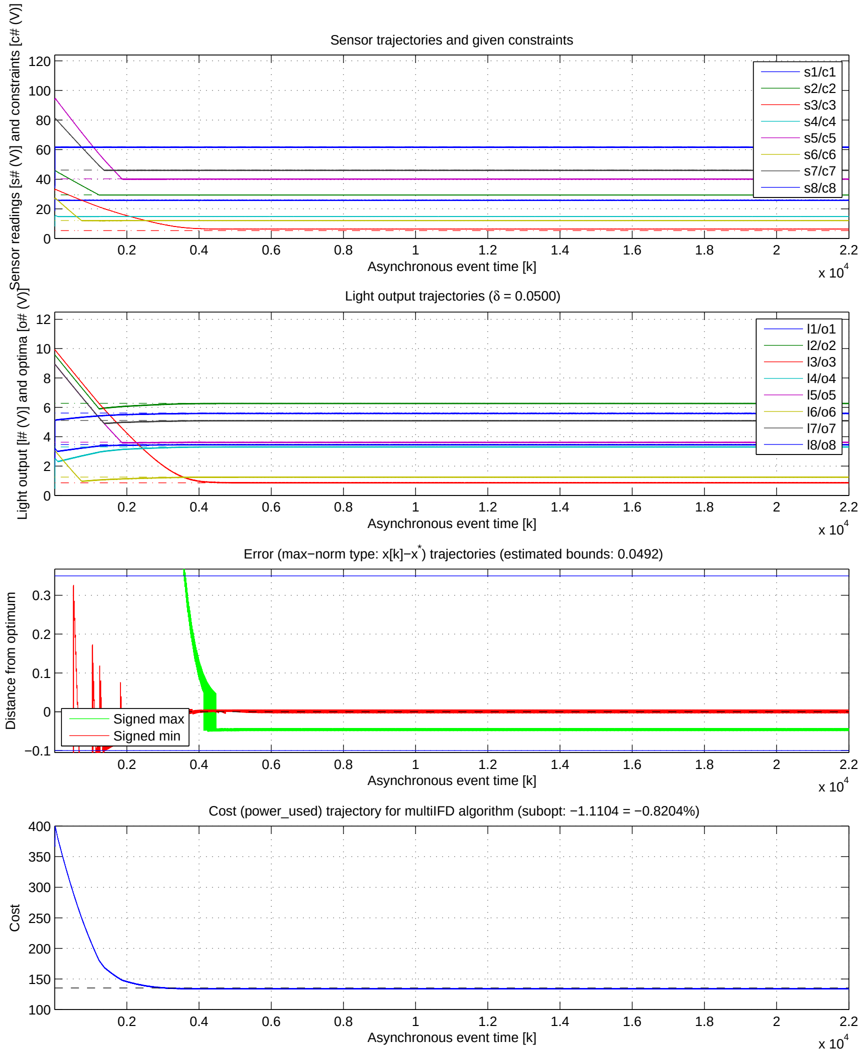
Simulation results

- Theoretical analysis predicts invariant hypercorners.
  - For single-constraint case, system enters invariant corners of hypercube with vertex anchored in constraint hyperplane.
  - Hypercube slides along constraint toward fixed hypercorners.
  - Ultimate error bounds from by fixed set location (2δ for $\|\cdot\|_2^2$).
- For multiple active constraints, hypercubes interact:

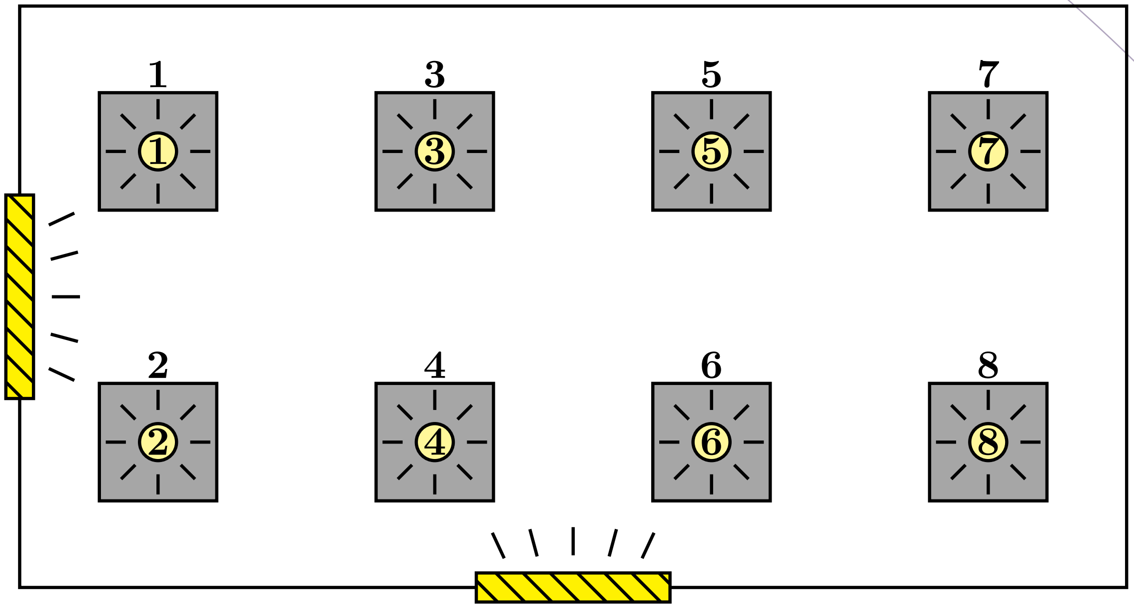


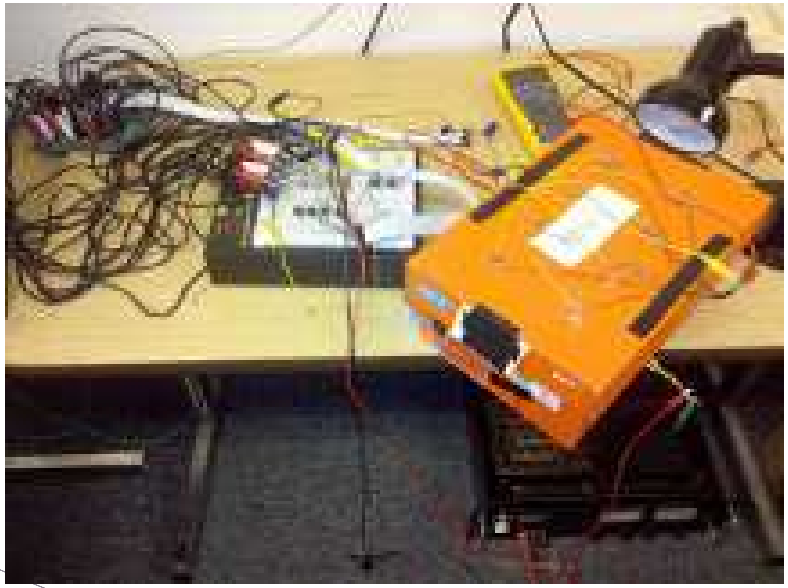
Simulation results

- Richer example:

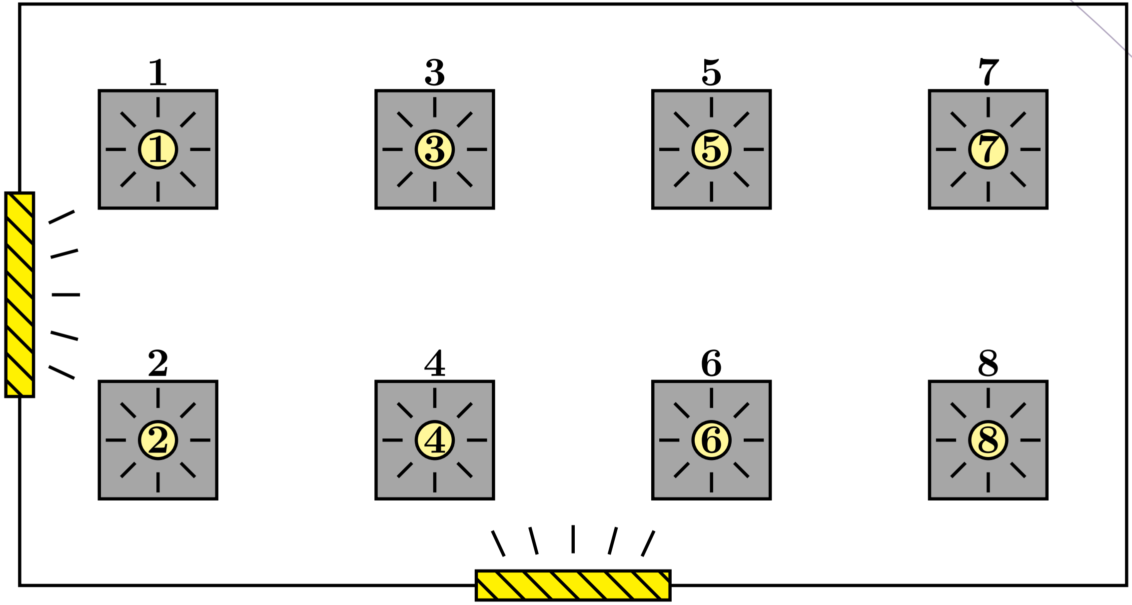


Experimental results
Intelligent light testbed

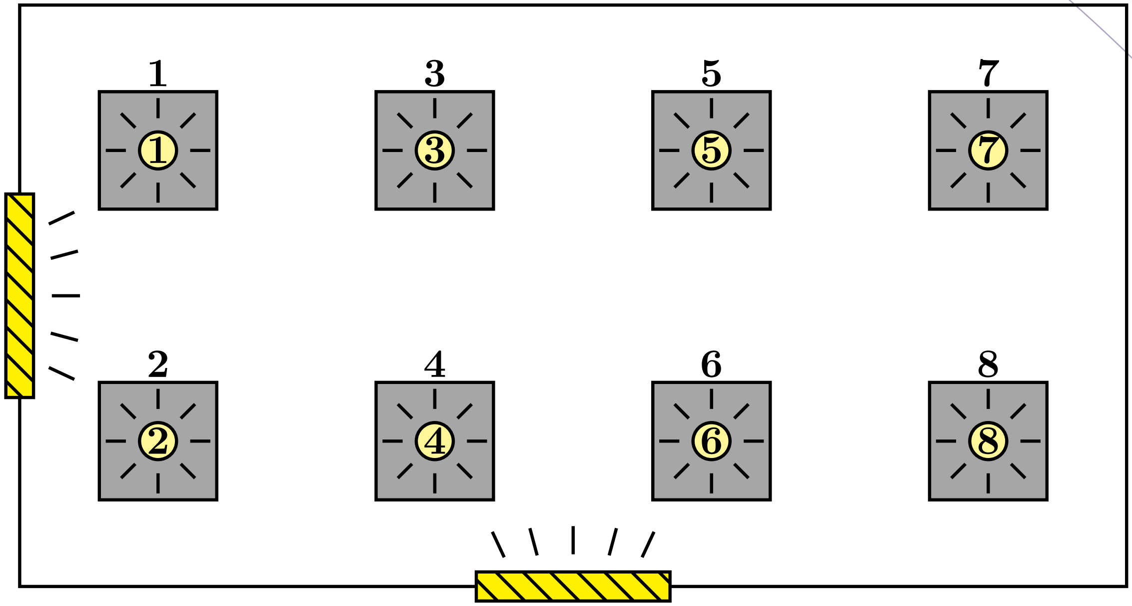




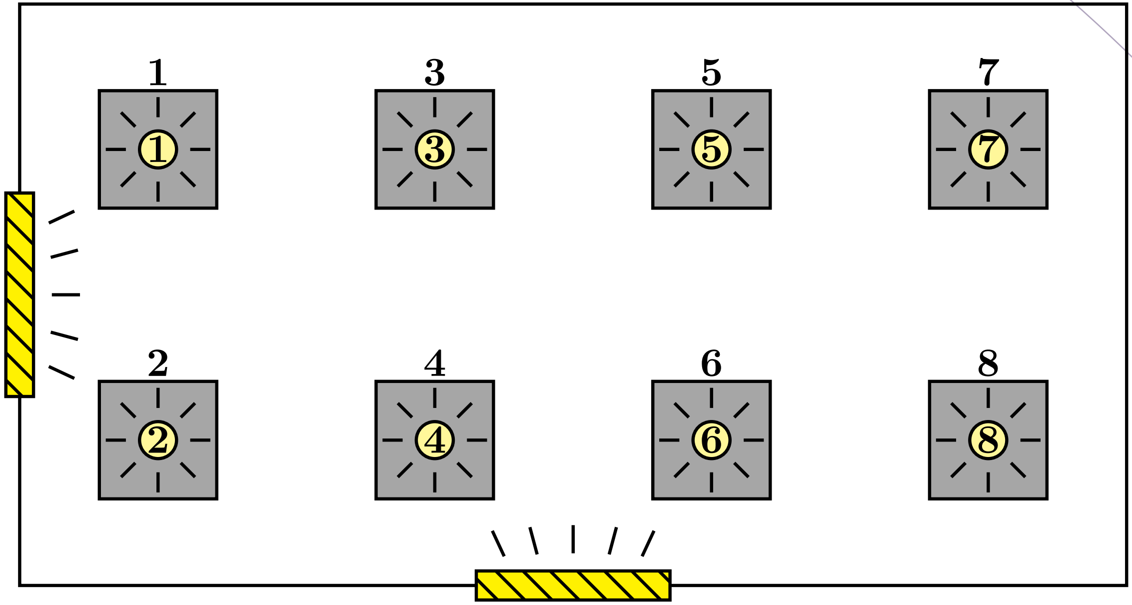
Experimental results
Intelligent light testbed

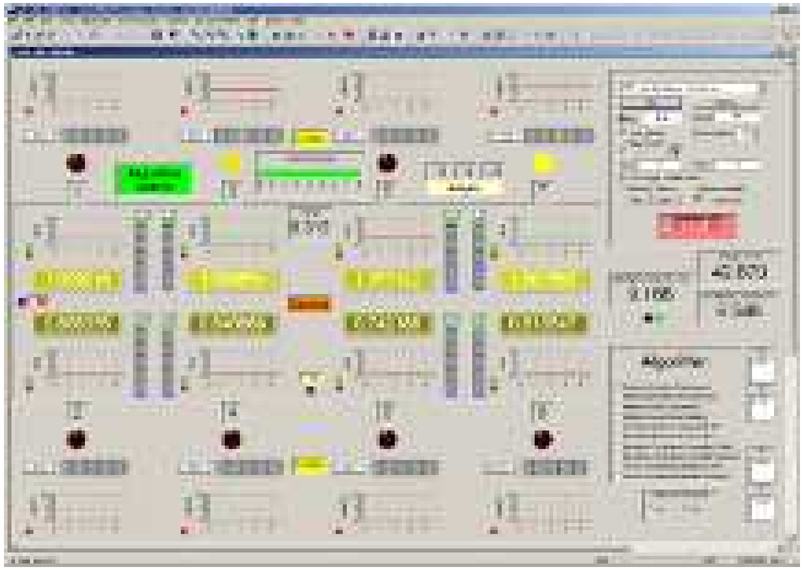


Experimental results
Intelligent light testbed

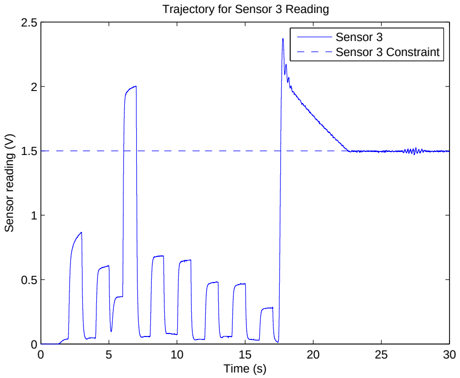


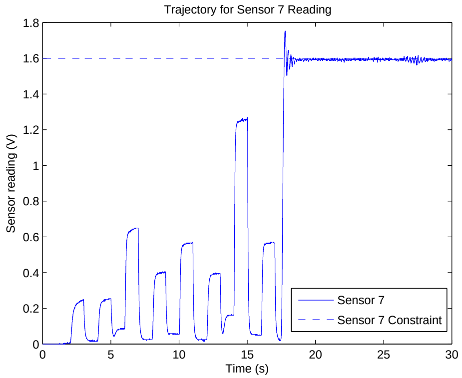
Experimental results
Intelligent light testbed

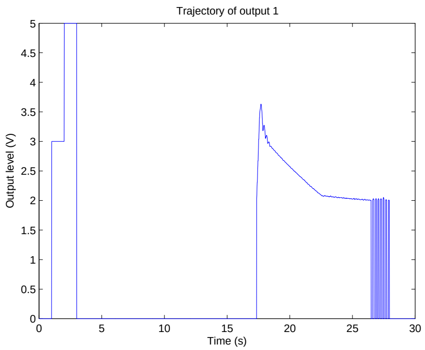


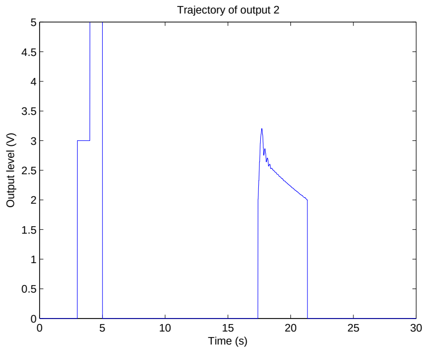


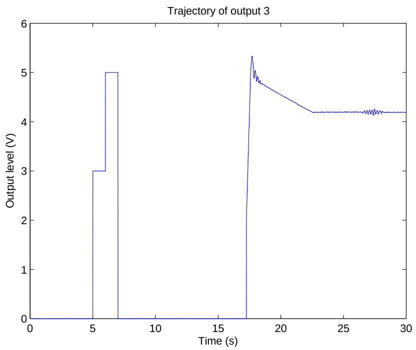
Experimental results
Distributed solver with automatic commissioning

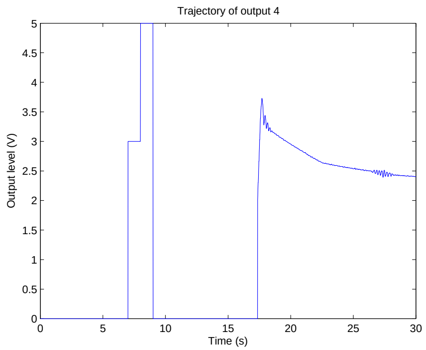

Trajectory for Sensor 3 Reading

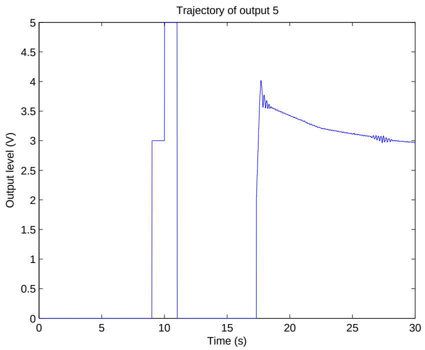

Trajectory for Sensor 7 Reading

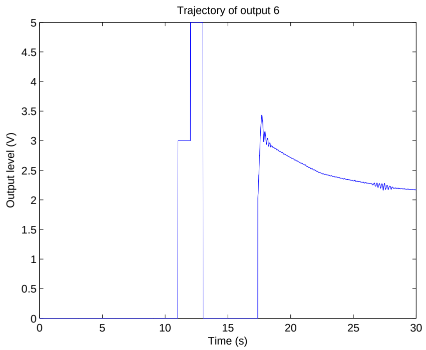

Trajectory of output 1

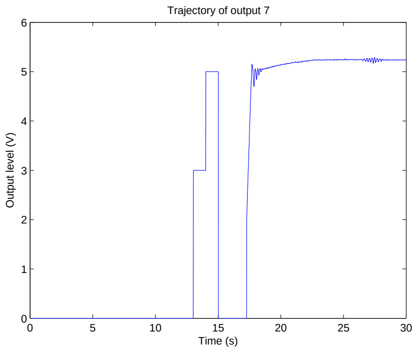

Trajectory of output 2

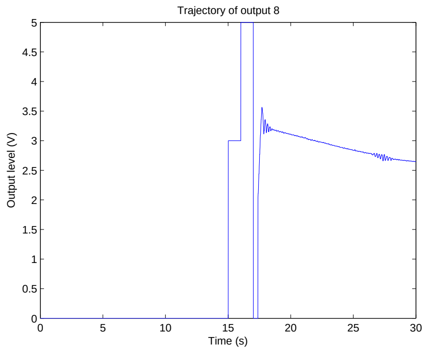

Trajectory of output 3


Trajectory of output 4


Trajectory of output 5

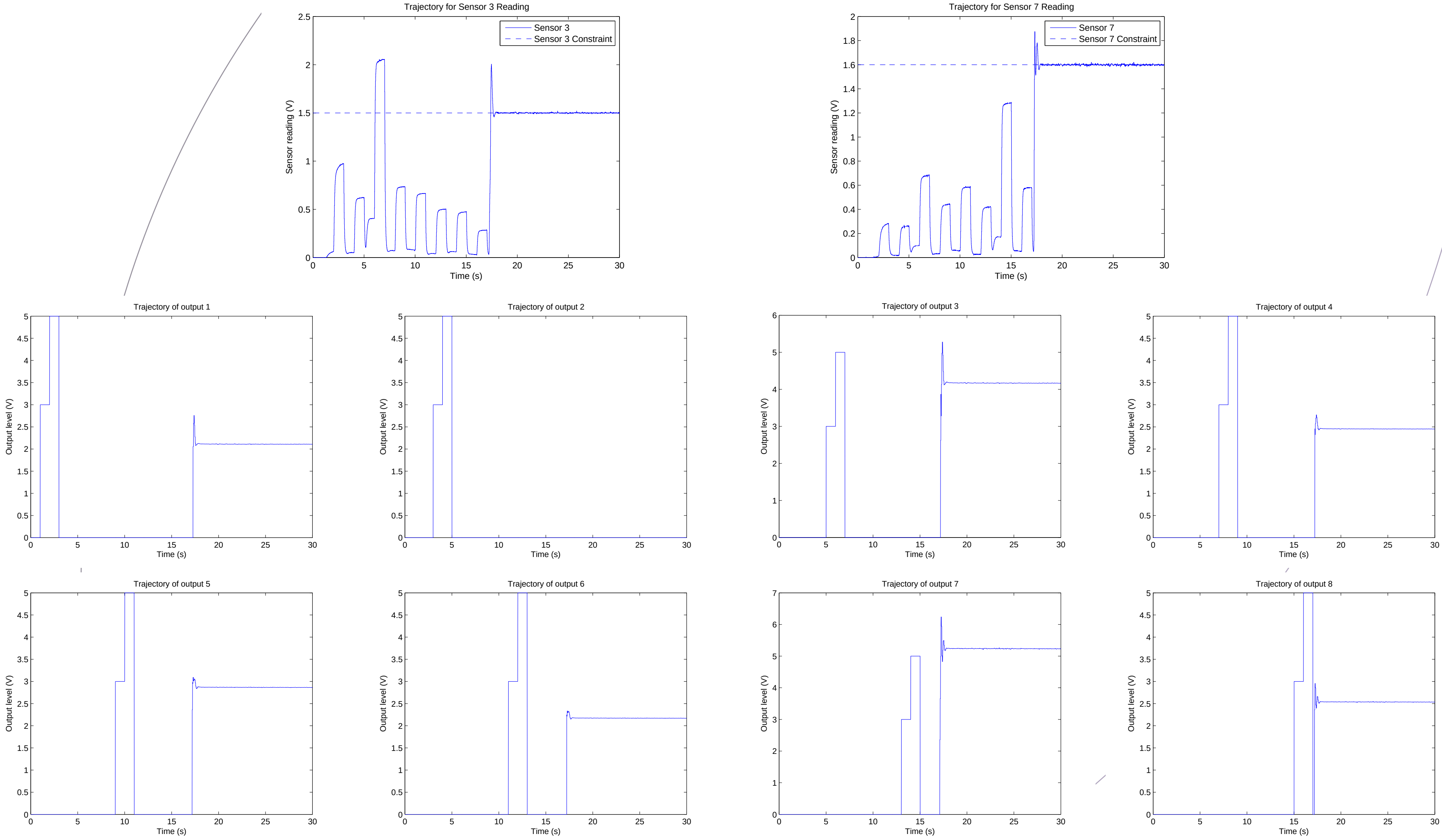

Trajectory of output 6


Trajectory of output 7


Trajectory of output 8

Experimental results

Centralized solver with automatic commissioning



Closing remarks

Contributions

Solitary task-processing agents

- Developed solitary-agent optimization framework.

Contributions
Solitary task-processing agents

- Developed solitary-agent optimization framework.
  - Generalizes classical optimal foraging theory.

Contributions
Solitary task-processing agents

- Developed solitary-agent optimization framework.
  - Generalizes classical optimal foraging theory.
  - Generates foraging-like behaviors that perform better under realistic assumptions.

Contributions
Solitary task-processing agents

- Developed solitary-agent optimization framework.
  - Generalizes classical optimal foraging theory.
  - Generates foraging-like behaviors that perform better under realistic assumptions.
- Used solitary-agent framework for engineering and biological applications.

Contributions
Solitary task-processing agents

- Developed solitary-agent optimization framework.
  - ☐ Generalizes classical optimal foraging theory.
  - ☐ Generates foraging-like behaviors that perform better under realistic assumptions.
- Used solitary-agent framework for engineering and biological applications.
  - ☐ Autonomous agent design: static behavior that adjusts to reduce shortfall likelihood in finite-event scenario.

Contributions

Solitary task-processing agents

- Developed solitary-agent optimization framework.
  - ☐ Generalizes classical optimal foraging theory.
  - ☐ Generates foraging-like behaviors that perform better under realistic assumptions.
- Used solitary-agent framework for engineering and biological applications.
  - ☐ Autonomous agent design: static behavior that adjusts to reduce shortfall likelihood in finite-event scenario.
  - ☐ Ecological rationality: impulsiveness explained as residue of operant laboratory*.

Contributions
Solitary task-processing agents

- Developed solitary-agent optimization framework.
  - ☐ Generalizes classical optimal foraging theory.
  - ☐ Generates foraging-like behaviors that perform better under realistic assumptions.
- Used solitary-agent framework for engineering and biological applications.
  - ☐ Autonomous agent design: static behavior that adjusts to reduce shortfall likelihood in finite-event scenario.
  - ☐ Ecological rationality: impulsiveness explained as residue of operant laboratory*.
  - ☐ Ecological rationality: sunk-cost effect showed to be rate maximizer with entry costs.

Contributions
Cooperative task-processing networks

- Developed framework for cooperative task processing on a network.

Contributions
Cooperative task-processing networks

- Developed framework for cooperative task processing on a network.
  - ☐ Candidate model for cooperative breeding, social foraging, cooperative patrol, flexible manufacturing systems, coffee shops, and others.

Contributions
Cooperative task-processing networks

- Developed framework for cooperative task processing on a network.
  - Candidate model for cooperative breeding, social foraging, cooperative patrol, flexible manufacturing systems, coffee shops, and others.
  - Fictitious economy generates Nash equilibrium solutions with non-trivial task sharing.

Contributions
Cooperative task-processing networks

- Developed framework for cooperative task processing on a network.
  - ☐ Candidate model for cooperative breeding, social foraging, cooperative patrol, flexible manufacturing systems, coffee shops, and others.
  - ☐ Fictitious economy generates Nash equilibrium solutions with non-trivial task sharing.
  - ☐ Totally asynchronous distributed solver converges to unique Nash equilibrium under topological and payment constraints.

Contributions
Cooperative task-processing networks

- Developed framework for cooperative task processing on a network.
  - ☐ Candidate model for cooperative breeding, social foraging, cooperative patrol, flexible manufacturing systems, coffee shops, and others.
  - ☐ Fictitious economy generates Nash equilibrium solutions with non-trivial task sharing.
    - Topological cycles lead to patterns qualitatively similar to load balancing.
  - ☐ Totally asynchronous distributed solver converges to unique Nash equilibrium under topological and payment constraints.

Contributions
Cooperative task-processing networks

- Developed framework for cooperative task processing on a network.
  - Candidate model for cooperative breeding, social foraging, cooperative patrol, flexible manufacturing systems, coffee shops, and others.
  - Fictitious economy generates Nash equilibrium solutions with non-trivial task sharing.
    - Topological cycles lead to patterns qualitatively similar to load balancing.
  - Totally asynchronous distributed solver converges to unique Nash equilibrium under topological and payment constraints.
    - Convergence conditions similar to Hamilton's rule on networks.

Contributions

Distributed solver for optimization under constraints

Contributions

Distributed solver for optimization under constraints

- New formulation of IFD as cost minimizer under nutrient constraints.

Contributions

Distributed solver for optimization under constraints

- New formulation of IFD as cost minimizer under nutrient constraints.
- Related IFD, economic power dispatch, and intelligent lights.

Contributions

Distributed solver for optimization under constraints

- New formulation of IFD as cost minimizer under nutrient constraints.
- Related IFD, economic power dispatch, and intelligent lights.
- Developed distributed solver for non-linear optimization program with constraints.

Contributions

Distributed solver for optimization under constraints

- New formulation of IFD as cost minimizer under nutrient constraints.
- Related IFD, economic power dispatch, and intelligent lights.
- Developed distributed solver for non-linear optimization program with constraints.
- Further developed tabletop intelligent lights testbed.

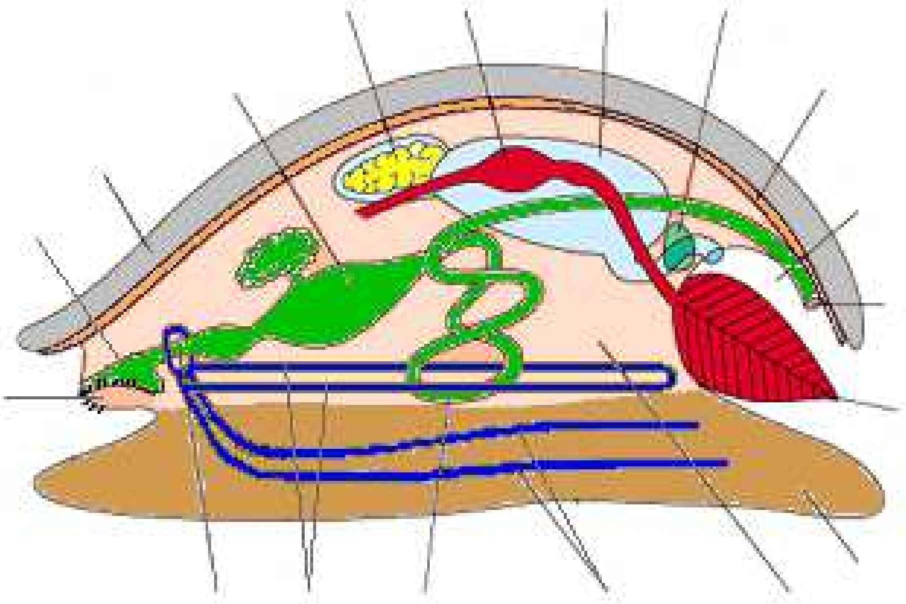
Contributions

Distributed solver for optimization under constraints

- New formulation of IFD as cost minimizer under nutrient constraints.
- Related IFD, economic power dispatch, and intelligent lights.
- Developed distributed solver for non-linear optimization program with constraints.
- Further developed tabletop intelligent lights testbed.
- Verified distributed solver matches centralized solver performance on experimental testbed.

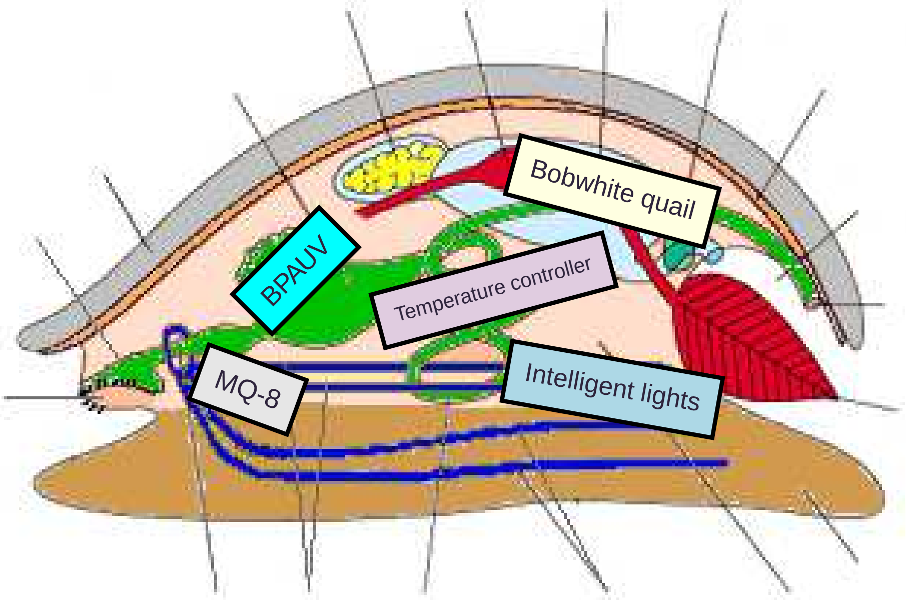
Bio-inspiration: Generalized mollusc*

(Anderson 2001; Ruppert et al. 2004)



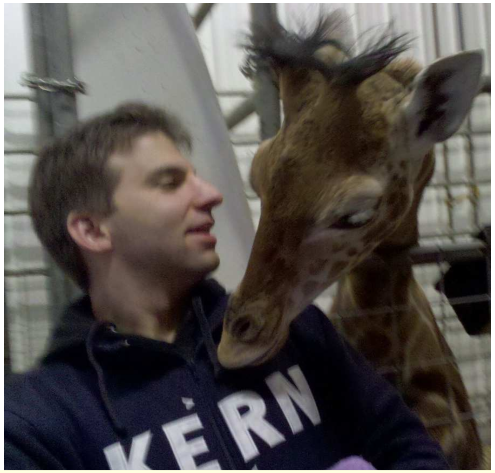
- **Hypothetical** generalized mollusc: Lots of molluscs under one shell

Generalized ~~mollusc~~models* (Pavlic 2010)?



- **Hypothetical** generalized mollusc: Lots of molluscs under one shell
- **Abstract** optimal task-processing agent: Lots of agents under unified behavioral framework

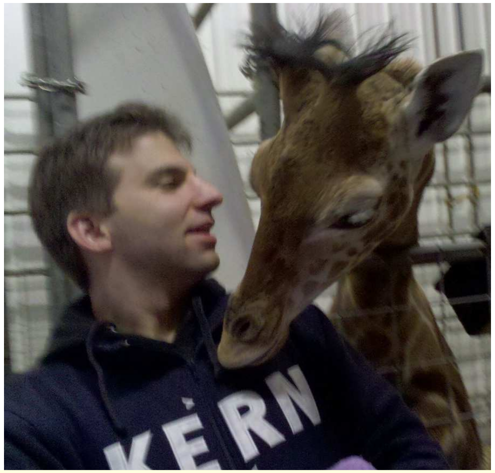
Thanks!



(*interdisciplinary research in engineering*: sticking your neck out to work with animals)

- Thank you!

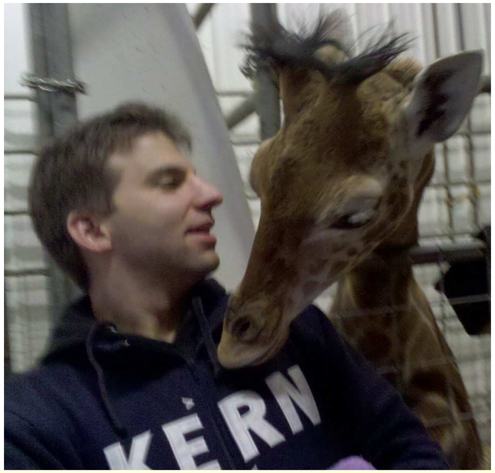
Thanks!



(*interdisciplinary research in engineering*: sticking your neck out to work with animals)

- Thank you!
- Helpful People: Kevin M. Passino, Thomas A. Waite, Ian M. Hamilton

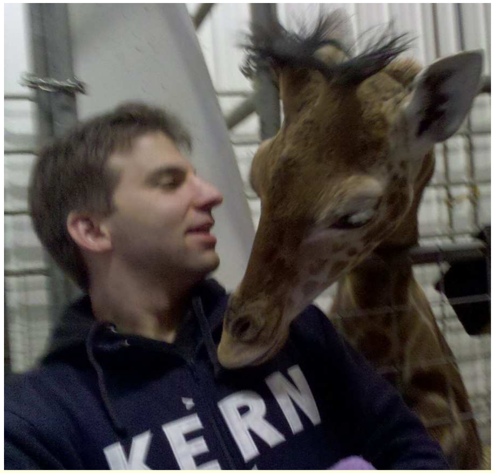
Thanks!



(*interdisciplinary research in engineering*: sticking your neck out to work with animals)

- Thank you!
- Helpful People: Kevin M. Passino, Thomas A. Waite, Ian M. Hamilton
- Reading Committee: Andrea Serrani, Atilla Eryilmaz, David Blau

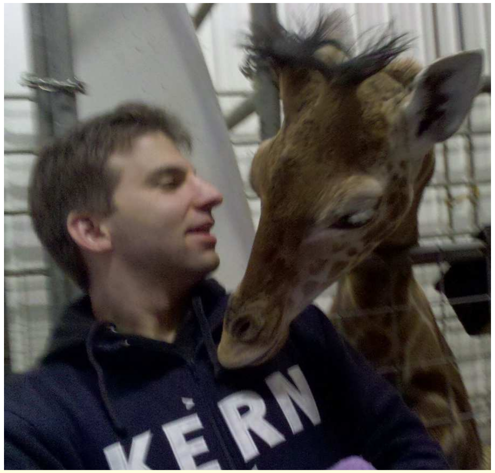
Thanks!



(*interdisciplinary research in engineering*: sticking your neck out to work with animals)

- Thank you!
- Helpful People: Kevin M. Passino, Thomas A. Waite, Ian M. Hamilton
- Reading Committee: Andrea Serrani, Atilla Eryilmaz, David Blau
- Funding Sources: AFRL, AFOSR, OSU, NSF

Thanks! Questions?



(*interdisciplinary research in engineering*: sticking your neck out to work with animals)

- Thank you!
- Helpful People: Kevin M. Passino, Thomas A. Waite, Ian M. Hamilton
- Reading Committee: Andrea Serrani, Atilla Eryilmaz, David Blau
- Funding Sources: AFRL, AFOSR, OSU, NSF
- Questions?

Works cited

Theodore P. Pavlic and Kevin M. Passino. Generalizing foraging theory for analysis and design. *International Journal of Robotics Research*, 2010c. To appear in special Stochastic Robotics issue

Theodore P. Pavlic and Kevin M. Passino. When rate maximization is impulsive. *Behavioral Ecology and Sociobiology*, 64(8):1255–1265, August 2010a. doi:10.1007/s00265-010-0940-1

Theodore P. Pavlic and Kevin M. Passino. The sunk-cost effect as an optimal rate-maximizing behavior. *Acta Biotheoretica*, 2010b. doi:10.1007/s10441-010-9107-8. In press

Theodore P. Pavlic and Kevin M. Passino. Cooperative task processing. *IEEE Transactions on Automatic Control*, 2010d. Submitted

Future directions*

- Solitary task-processing agents
- Cooperative task processing networks
- MultiIFD gradient descent
- Future future directions*

---

* Omitted for brevity

Future directions: solitary task-processing agents[*]

---

[*] Omitted for brevity

Solitary task-processing agents

- Incorporate speed choice into advantage-to-disadvantage framework (details follow)
- Investigate using post-modern portfolio theory (PMPT) and stochastic dominance to revitalize risk-sensitivity theory.

Foraging theory for speed choice*

- Vehicle speed choice is very similar to cryptic prey problem described by Gendron and Staddon (1983)
  - ☐ *Ceteris paribus*, encounter rate increases with search speed
  - ☐ Search cost increases with search speed
  - ☐ Detection mistakes may vary with speed
  - ☐ Non-trivial speed–prey choice coupling
    - Prey $\Longrightarrow$ speed $\Longrightarrow$ rate $\Longrightarrow$ prey

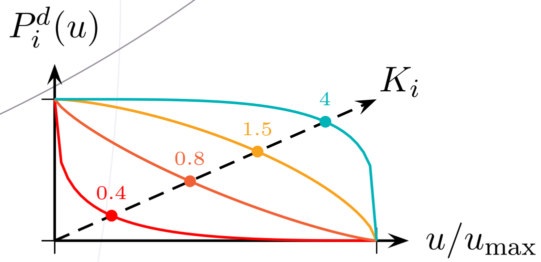
Bobwhite quail (Gendron and Staddon 1983)

Foraging theory for speed choice*

- Vehicle speed choice is very similar to cryptic prey problem described by Gendron and Staddon (1983)
  - ☐ *Ceteris paribus*, encounter rate increases with search speed
  - ☐ Search cost increases with search speed
  - ☐ Detection mistakes may vary with speed
  - ☐ Non-trivial speed–prey choice coupling
    - Prey $\Longrightarrow$ speed $\Longrightarrow$ rate $\Longrightarrow$ prey

Bobwhite quail (Gendron and Staddon 1983)

- To match bobwhite quail observations, Gendron and Staddon choose detection function $P_i^d(u) \triangleq (1 - (u/u_{\text{max}})^{K_i})^{1/K_i}$ that maps search speed $u \in [0, u_{\text{max}}]$ to detection probability P_i^d for tasks of type i with conspicuousness $K_i \in [0, \infty)$.
  - ☐ No analytical tractability
  - ☐ Chose $n = 2$ for simulation (1983)
  - ☐ P_i^d is strange at bounds (1 and 0)



On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- Speed $u \in [u_{\text{min}}, u_{\text{max}}] \subset [0, \infty)$ influences each encounter rate

$$\lambda_i(u) = u D_i P_i^d(u)$$

where D_i is the linear density in the population

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

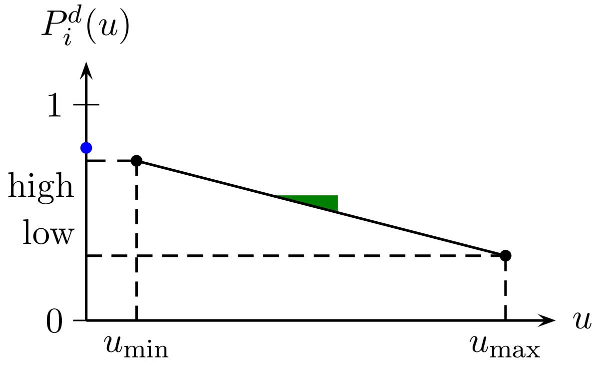
(Pavlic and Passino 2009)

- Speed $u \in [u_{\text{min}}, u_{\text{max}}] \subset [0, \infty)$ influences each encounter rate

$$\lambda_i(u) = u D_i P_i^d(u)$$

where D_i is the linear density in the population

- Detection function is linear interpolation of probability bounds



$$P_i^d(u) = P_i^\ell u + P_i^a$$

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

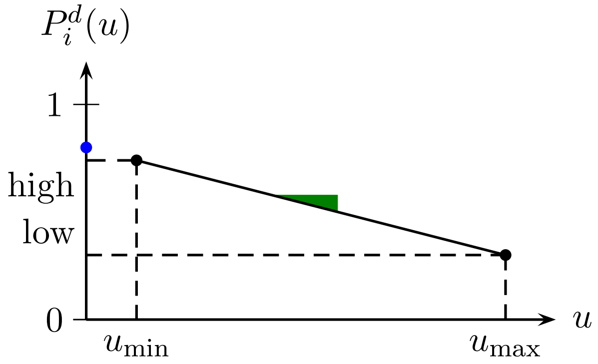
(Pavlic and Passino 2009)

- Speed $u \in [u_{\text{min}}, u_{\text{max}}] \subset [0, \infty)$ influences each encounter rate

$$\lambda_i(u) = uD_iP_i^d(u)$$

where D_i is the linear density in the population

- Detection function is linear interpolation of probability bounds



$$P_i^d(u) = P_i^\ell u + P_i^a$$

- Search cost is also assumed to be affine function

$$c^s(u) = c_\ell^s u + c_a^s$$

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

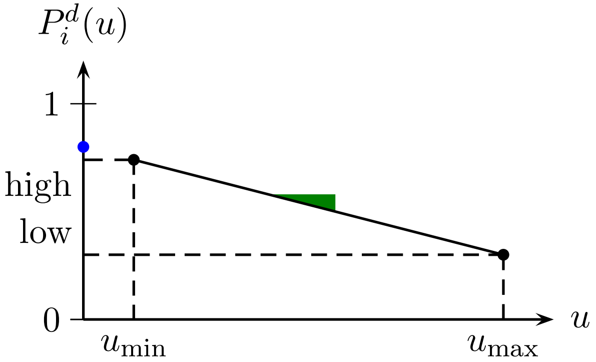
(Pavlic and Passino 2009)

- Speed $u \in [u_{\text{min}}, u_{\text{max}}] \subset [0, \infty)$ influences each encounter rate

$$\lambda_i(u) = u D_i P_i^d(u)$$

where D_i is the linear density in the population

- Detection function is linear interpolation of probability bounds



$$P_i^d(u) = P_i^\ell u + P_i^a$$

- *[Processing costs can be modeled in a similar way]*

$$c_i(u) = c_i^\ell u + c_i^a$$

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- After regrouping, new objective function

$$R(\vec{p},u) = \frac{G_2(\vec{p})u^2 + G_1(\vec{p})u + G_0(\vec{q})}{T_2(\vec{p})u^2 + T_1(\vec{p})u + 1}$$

where coefficients

$$G_2(\vec{p}) \triangleq \sum_{i=1}^{n} D_i p_i g_i P_i^\ell$$

$$G_1(\vec{p}) \triangleq \sum_{i=1}^{n} D_i p_i P_i^a g_i - c_\ell^s$$

$$G_0(\vec{p}) \triangleq -c_a^s$$

$$T_2(\vec{p}) \triangleq \sum_{i=1}^{n} p_i \tau_i D_i P_i^\ell$$

$$T_1(\vec{p}) \triangleq \sum_{i=1}^{n} p_i \tau_i D_i P_i^a$$

are constant with respect to u (i.e., biquadratic ratio)

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- After regrouping, new objective function

$$R(\vec{p}, u) = \frac{G_2(\vec{p})u^2 + G_1(\vec{p})u + G_0(\vec{q})}{T_2(\vec{p})u^2 + T_1(\vec{p})u + 1}$$

where coefficients

$$G_2(\vec{p}) \triangleq \sum_{i=1}^{n} D_i p_i g_i P_i^\ell \qquad T_2(\vec{p}) \triangleq \sum_{i=1}^{n} p_i \tau_i D_i P_i^\ell$$

$$G_1(\vec{p}) \triangleq \sum_{i=1}^{n} D_i p_i P_i^a g_i - c_\ell^s \qquad T_1(\vec{p}) \triangleq \sum_{i=1}^{n} p_i \tau_i D_i P_i^a$$

$$G_0(\vec{p}) \triangleq -c_a^s$$

are constant with respect to u (i.e., biquadratic ratio)

- Find optimal u^* for each $\vec{p}^*$ candidate ($n+1$ total)

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- Because biquadratic objective, for each $\vec{p}^*$ candidate,

$$\frac{\partial R(u)}{\partial u} = \frac{(G_2T_1 - G_1T_2)u^2 + 2(G_2 - G_0T_2)u + (G_1 - G_0T_1)}{\left(T_2u^2 + T_1u + 1\right)^2}$$

By KKT, if quadratic numerator root $u^* \in [u_{\text{min}}, u_{\text{max}}]$, then u^* is optimal speed; otherwise, optimal speed $u^* \in \{u_{\text{min}}, u_{\text{max}}\}$ based on sign of numerator

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- Because biquadratic objective, for each $\vec{p}^*$ candidate,

$$\frac{\partial R(u)}{\partial u} = \frac{(G_2T_1 - G_1T_2)u^2 + 2(G_2 - G_0T_2)u + (G_1 - G_0T_1)}{\left(T_2u^2 + T_1u + 1\right)^2}$$

By KKT, if quadratic numerator root $u^* \in [u_{\text{min}}, u_{\text{max}}]$, then u^* is optimal speed; otherwise, optimal speed $u^* \in \{u_{\text{min}}, u_{\text{max}}\}$ based on sign of numerator

- Implement $(n+1)$-search algorithm on-line if D_i density estimates available (Pavlic and Passino 2009, Dubins' car AAV simulations with speed filtering)

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- Because biquadratic objective, for each $\vec{p}^*$ candidate,

$$\frac{\partial R(u)}{\partial u} = \frac{(G_2T_1 - G_1T_2)u^2 + 2(G_2 - G_0T_2)u + (G_1 - G_0T_1)}{\left(T_2u^2 + T_1u + 1\right)^2}$$

By KKT, if quadratic numerator root $u^* \in [u_{\text{min}}, u_{\text{max}}]$, then u^* is optimal speed; otherwise, optimal speed $u^* \in \{u_{\text{min}}, u_{\text{max}}\}$ based on sign of numerator

- Implement $(n+1)$-search algorithm on-line if D_i density estimates available (Pavlic and Passino 2009, Dubins' car AAV simulations with speed filtering)
- Non-trivial to guarantee convergence of density estimates on-line
  - Estimation process $\Longrightarrow$ ~~type-II~~ **type-III** functional response

On-line prey–speed choice for $n \in \mathbb{N}$: Effects of speed*

(Pavlic and Passino 2009)

- Because biquadratic objective, for each $\vec{p}^*$ candidate,

$$\frac{\partial R(u)}{\partial u} = \frac{(G_2T_1 - G_1T_2)u^2 + 2(G_2 - G_0T_2)u + (G_1 - G_0T_1)}{\left(T_2u^2 + T_1u + 1\right)^2}$$

By KKT, if quadratic numerator root $u^* \in [u_{\text{min}}, u_{\text{max}}]$, then u^* is
optin based
on si

Future direction:

Augment advantage-to-disadvantage framework with parameter representing speed.

- Implement $(n+1)$ search algorithm on-line if D_i density estimates available (Pavlic and Passino 2009, Dubins' car AAV simulations with speed filtering)
- Non-trivial to guarantee convergence of density estimates on-line
  - ☐ Estimation process $\Longrightarrow$ ~~type-II~~ **type-III** functional response

Future directions: cooperative task-processing networks[*]

---

[*] Omitted for brevity

Future directions: cooperative task-processing networks

- Optimal forwarding tendencies
- Simultaneous forwarding and volunteering tendencies
- Tendencies that vary across neighbors
- Incorporate processing time
- Reciprocity

Future directions: MultiIFD gradient descent*

---

*Omitted for brevity

Future directions: MultiIFD gradient descent

- Proof for multiple-constraint case
- Proof for looser timing
- Non-linear (but likely convex) constraints
    - Methods for automatic commissioning
- Compare performance to other conventional intelligent lighting algorithms
- Exploration of implementation in non-lighting applications

Future future directions*

---

* Omitted for brevity

Future future directions*

- Current–voltage/predator–prey analogy
  - Tunnel diode limit cycles
  - Extend circuit theory into ecological analysis
- Tree dynamics
  - Game theoretic analysis of tree distributions
  - Tree growth as dynamic system
- Game theoretic analysis of energy efficiency bait-and-switch
  - Future pricing plans charge per service rather than per kW-hour
  - Consumer incentive to upgrade to high-efficiency devices assumes long-term payoff
  - Long-term payoff vanishes when pricing model changes
  - Few now know of pricing changes