Biomimicry of Bacterial Foraging
for Distributed Optimization and Control

Kevin M. Passino[1]
Presented by: Alexander Van de Kleut[2]

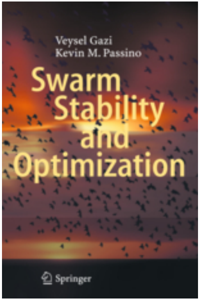
[1]The Ohio State University
Electrical and Computer Engineering

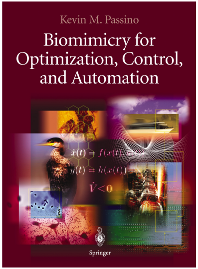
[2]University of Waterloo
Centre for Theoretical Neuroscience

IEEE Control Systems Magazine, 2002

Table of Contents

About the Author





About the Author

Fuzzy control — 3599
KM Passino, S Yurkovich, M Reinfrank
Addison-wesley 42, 15-21, 1998

Biomimicry of bacterial foraging for distributed optimization and control — 3023
KM Passino
IEEE control systems magazine 22 (3), 52-67, 2002

Stability analysis of swarms — 1125
V Gazi, KM Passino
IEEE transactions on automatic control 48 (4), 692-697, 2003

Stable adaptive control using fuzzy systems and neural networks — 728
JT Spooner, KM Passino
IEEE Transactions on Fuzzy Systems 4 (3), 339-359, 1996

Stability analysis of social foraging swarms — 710
V Gazi, KM Passino
IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 …, 2004

Foraging

Foraging

- searching for nutrients
- avoiding noxious stimuli (toxins, predators, etc)

Social Foraging

- increases likelihood of finding nutrients
- better detection and protection from noxious stimuli
- gains can offset cost of food competition

Foraging as Optimization

How can we view foraging as an Optimization Process?

- We have some parameters θ and a loss function $J(\theta)$ that we want to minimize

Foraging as Optimization

How can we view foraging as an Optimization Process?

- We have some parameters θ and a loss function $J(\theta)$ that we want to minimize
- θ can represent the position of an organism in its environment

Foraging as Optimization

How can we view foraging as an Optimization Process?

- We have some parameters θ and a loss function $J(\theta)$ that we want to minimize
- θ can represent the position of an organism in its environment
- J can represent the concentration of nutrients and noxious stimuli
  - smaller values of J = more nutrients, less noxious stimuli
  - higher values of J = more noxious stimuli, less nutrients

Foraging as Optimization

How can we view foraging as an Optimization Process?

- We have some parameters θ and a loss function $J(\theta)$ that we want to minimize
- θ can represent the position of an organism in its environment
- J can represent the concentration of nutrients and noxious stimuli
  - smaller values of J = more nutrients, less noxious stimuli
  - higher values of J = more noxious stimuli, less nutrients
- In general, J and θ can be arbitrary
  - $\theta \in \mathbb{R}^p$
  - $J : \mathbb{R}^p \to \mathbb{R}$

E. coli

- Model organism

E. coli

- Model organism
  - Highly studied

E. coli

- Model organism
  - Highly studied
  - Well-characterized foraging behaviour

E. coli

- Model organism
  - Highly studied
  - Well-characterized foraging behaviour
  - Probably won't feel bad about simplifying its behaviour

E. coli

- Model organism
  - Highly studied
  - Well-characterized foraging behaviour
  - Probably won't feel bad about simplifying its behaviour
- Social organism

E. coli

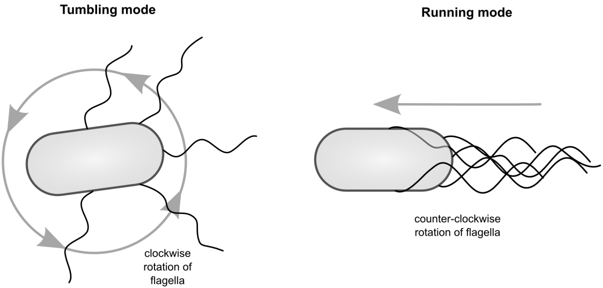
- Model organism
  - Highly studied
  - Well-characterized foraging behaviour
  - Probably won't feel bad about simplifying its behaviour
- Social organism
  - Secretes signals to attract others nearby

E. coli

- Model organism
  - Highly studied
  - Well-characterized foraging behaviour
  - Probably won't feel bad about simplifying its behaviour
- Social organism
  - Secretes signals to attract others nearby
  - Encourages "swarming" or "clumping"

E. Coli Behaviour

- Swims using left-handed helical flagella ("propellers")
  - **Tumble**: flagella all rotate clockwise → pull on cell in all directions → random movement
  - **Run**: flagella all rotate counterclockwise → flagella form a bundle → push on cell in one direction → directed movement



E. Coli Behaviour

- If during a tumble E. Coli swims down a nutrient concentration gradient:
  - Prolongs time spent on a run
  - Continues moving in the same direction
- Otherwise:
  - Tends to switch to a tumble (search for more)
  - Moves randomly which searching for more nutrient gradients to exploit

Algorithm for a Single Bacterium

$\theta \sim \mathcal{U}^p(\min, \max)$
for $j \leftarrow 1 \ldots N_c$ **do**:
    $\Delta \sim \mathcal{U}^p(-1, 1)$
    $J_{\text{last}} \leftarrow J(\theta)$
    $\theta \leftarrow \theta + C \frac{\Delta}{\|\Delta\|}$
    for $m \leftarrow 1 \ldots N_s$ **do**:
        if $J(\theta) < J_{\text{last}}$ **then**:
            $J_{\text{last}} \leftarrow J(\theta)$
            $\theta \leftarrow \theta + C \frac{\Delta}{\|\Delta\|}$
        else
            $m \leftarrow N_s$

Algorithm for a Colony

for $i \leftarrow 1 \ldots S$ **do**:
    $\theta_i \sim \mathcal{U}^p(\min, \max)$
for $j \leftarrow 1 \ldots N_c$ **do**:
    for $i \leftarrow 1 \ldots S$ **do**:
        $\Delta_i \sim \mathcal{U}^p(-1, 1)$
        $J_{\text{last}} \leftarrow J(\theta_i)$
        $\theta_i \leftarrow \theta_i + C \frac{\Delta_i}{\|\Delta_i\|}$
        for $m \leftarrow 1 \ldots N_s$ **do**:
            if $J(\theta_i) < J_{\text{last}}$ **then**:
                $J_{\text{last}} \leftarrow J(\theta_i)$
                $\theta_i \leftarrow \theta_i + C \frac{\Delta_i}{\|\Delta_i\|}$
            else
                $m \leftarrow N_s$

E. Coli Swarming

Algorithm for a Colony with Swarming

for $i \leftarrow 1 \ldots S$ **do**:
    $\theta_i \sim \mathcal{U}^p(\min, \max)$
for $j \leftarrow 1 \ldots N_c$ **do**:
    for $i \leftarrow 1 \ldots S$ **do**:
        $\Delta_i \sim \mathcal{U}^p(-1, 1)$
        $J_{\text{last}} \leftarrow J(\theta_i) + J_{cc}(\theta_i)$
        $\theta_i \leftarrow \theta_i + C(i)\frac{\Delta_i}{\|\Delta_i\|}$
        for $m \leftarrow 1 \ldots N_s$ **do**:
            if $J(\theta_i) < J_{\text{last}}$ **then**:
                $J_{\text{last}} \leftarrow J(\theta_i) + J_{cc}(\theta_i)$
                $\theta_i \leftarrow \theta_i + C(i)\frac{\Delta_i}{\|\Delta_i\|}$
            else
                $m \leftarrow N_s$