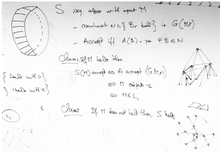
Integrating Theoretical Algorithmic Ideas in Empirical Biological Study



Amos Korman

In collaboration with Ofer Feinerman (Weizmann Institute)

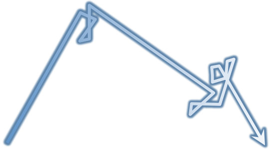
Outline

Classical scientific frameworks in biology

Experimental framework:

1. Preprocessing stage: observe and analyze
2. "**Guess**" a mathematical model
3. Data analysis: tune the parameters

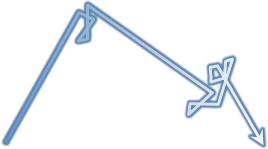
Example: the *Albatross* (Nature 1996, 2007)



$\Pr(l = d) \approx 1/d^{\alpha}$

The *Albatross* is performing a *Lévy flight*

Example: the *Albatross* (Nature 1996, 2007)



$\Pr(l = d) \approx 1/d^{\alpha}$

The *Albatross* is performing a *Lévy flight*

What is α? do statistics on experiments and obtain e.g., $\alpha = 2$

Theoretical framework:

1. **Guess** an abstract mathematical model (loosely representing reality)
2. **Analyze** the model

Theoretical framework:

1. **Guess** an abstract mathematical model (loosely representing reality)
2. **Analyze** the model
   - Find parameters maximizing a utility function

Theoretical framework:

1. **Guess** an abstract mathematical model
   (loosely representing reality)
2. **Analyze** the model
   - Find parameters maximizing a utility function

     Example: if you perform a Lévy flight search under some certain food distribution then $\alpha = 2$ is optimal [Viswanathan et al. Nature 1999]

Theoretical framework:

1. **Guess** an abstract mathematical model
   (loosely representing reality)
2. **Analyze** the model
   - Find parameters maximizing a utility function

     Example: if you perform a Lévy flight search under some certain food distribution then $\alpha = 2$ is optimal [Viswanathan et al. Nature 1999]

   - "Explain" a known phenomena

     Example: Kleinberg's analysis of the greedy routing algorithm in small world networks "explains" Milgram's experiment [Nature 2000]

Outline

An algorithmic perspective

Recently, CS theoreticians have tried to contribute from an algorithmic perspective [Alon, Chazelle, Kleinberg, Papadimitriou, Valiant, etc.].

An algorithmic perspective

Recently, CS theoreticians have tried to contribute from an algorithmic perspective [Alon, Chazelle, Kleinberg, Papadimitriou, Valiant, etc.].

Guiding principle

Algorithms' people are good at:

An algorithmic perspective

Recently, CS theoreticians have tried to contribute from an algorithmic perspective [Alon, Chazelle, Kleinberg, Papadimitriou, Valiant, etc.].

Guiding principle

Algorithms' people are good at:

1. Formulating sophisticated **guesses** (algorithms)
2. **Analyzing** the algorithms

Algorithmic perspective in classical frameworks

Experimental framework:

1. Preprocessing stage: observe and analyze
2. **Guess** a mathematical model [Afek et al., Science'11]
3. Data analysis: tune the parameters

Theoretical framework:

1. **Guess** a mathematical model
2. **Analyze** the model
   - Maximize a utility function [Papadimitriou et al., PNAS 2008]
   - Explain a known phenomena [Kleinberg, Nature, 2000]

Algorithmic perspective in classical frameworks

Experimental framework:

1. Preprocessing stage: observe and analyze
2. **Guess** a mathematical model [Afek et al., Science'11]
3. Data analysis: tune the parameters

Theoretical framework:

1. **Guess** a mathematical model
2. **Analyze** the model
    - Maximize a utility function [Papadimitriou et al., PNAS 2008]
    - Explain a known phenomena [Kleinberg, Nature, 2000]

Can an algorithmic perspective contribute otherwise?

Tradeoffs: Connections between parameters

Tradeoffs: Connections between parameters

A big challenge: reduce the parameter space

Solution in **physics**: obtain equation (or connection) between parameters. E.g., $E = MC^2$, $\Delta U = Q + W$, etc.

Tradeoffs: Connections between parameters

A big challenge: reduce the parameter space

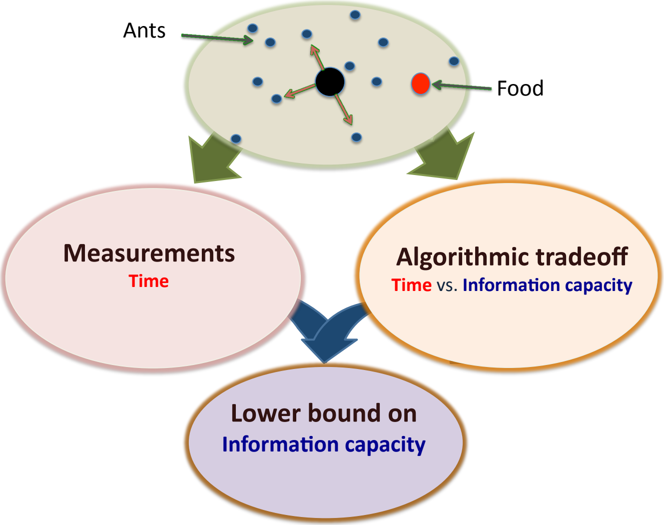
Solution in **physics**: obtain equation (or connection) between parameters. E.g., $E = MC^2$, $\Delta U = Q + W$, etc.

What about biology?

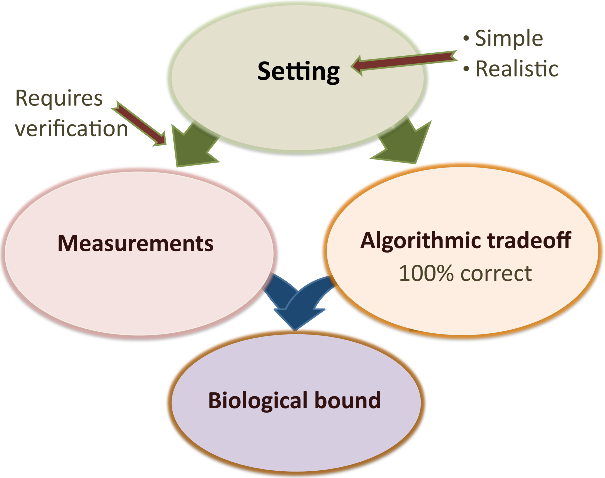
- 1st solution: borrow connections from physics
- We propose: obtain connections between parameters using an **algorithmic approach.**
  Tradeoffs: use lower bounds from CS to show that, e.g., any algorithm that runs in time T must use x amount of resources ($x > f(T)$).

Outline

Connecting parameters using an algorithmic perspective



Remarks: simplified experimental verifications



◦ Tradeoffs are invariant of the algorithm $\Longrightarrow$ Instead of verifying setting+algorithm, only need to verify the setting!

A proof of concept

This talk

- Introduce the model (semi-realistic)
- Discuss the theoretical tradeoffs
- Experimental part: on-going

A proof of concept

This talk

- Introduce the model (semi-realistic)
- Discuss the theoretical tradeoffs
- Experimental part: on-going

Remark

The work is not complete. This presentation is a ***proof of concept***

Outline

Inspiration: the Cataglyphis niger and Honey bee

The *Cataglyphis niger:*

Inspiration: the Cataglyphis niger and Honey bee

The *Cataglyphis niger:*

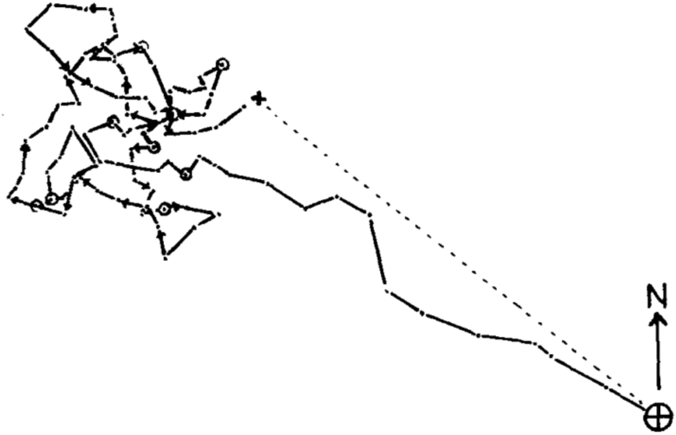
- Desert ant– does not leave traces, more individual

Inspiration: the Cataglyphis niger and Honey bee

The *Cataglyphis niger:*

- Desert ant– does not leave traces, more individual
- Relatively smart– big brain, good navigation abilities

Good distance and location estimations [Wehner et al.]

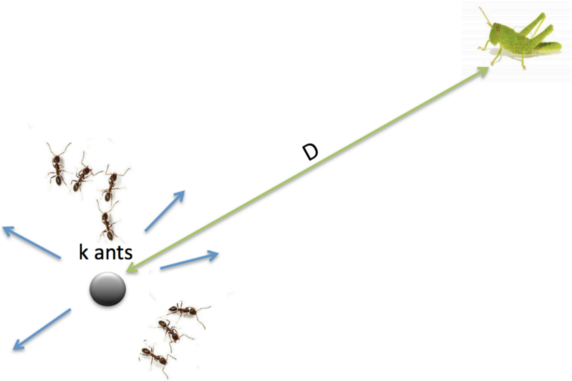


Goal: find nearby treasures fast

Reasons for proximity

- Increasing the rate of food collection in case a large quantity of food is found [Orians and Pearson, 1979],
- Decreasing predation risk [Krebs, 1980],
- The ease of navigating back after collecting the food using familiar landmarks [Collett et al., 1992], etc.

Central place foraging

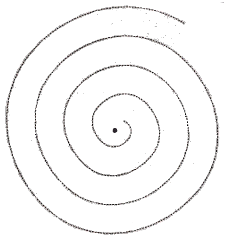


- Goal: find nearby treasures fast (biologically motivated)
- No communication once out of the nest
- Grid network: the *visual radius* determines the grid resolution
- **Fact:** The expected running time is $\Omega(D + D^2/k)$

Searching with one ant ($k = 1$)

An optimal algorithm

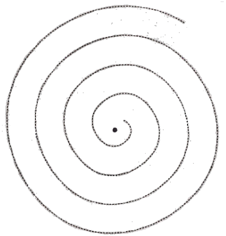
Perform a *spiral* search from the nest (takes $O(D^2)$ time).



Searching with one ant ($k = 1$)

An optimal algorithm

Perform a *spiral* search from the nest (takes $O(D^2)$ time).



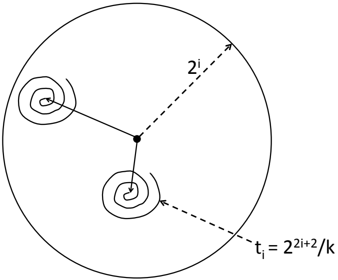
Random walk

Not efficient: expected time to visit any given node is ∞.

Optimal algorithm (PODC 2012) [Feinerman, Korman, Lotker, Sereni]

Lemma

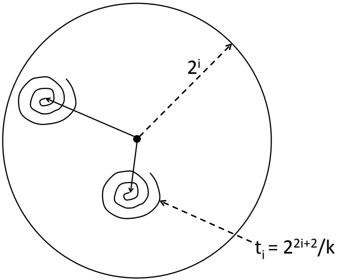
If agents know the value of k then there exists an optimal algorithm running in time $O(D + D^2/k)$



Optimal algorithm (PODC 2012) [Feinerman, Korman, Lotker, Sereni]

Lemma

If agents know the value of k then there exists an optimal algorithm running in time $O(D + D^2/k)$



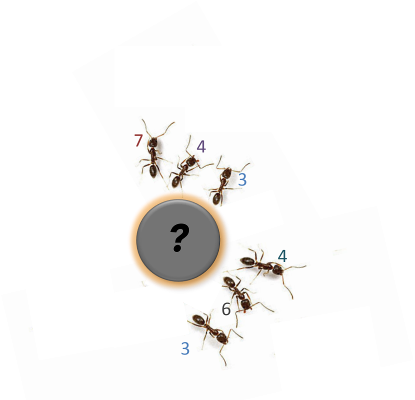
Questions: Is it necessary to know k? How much initial information is necessary?

Outline

What is the amount of information that agents need initially?

Probabilistic centralized oracle

Given k agents, oracle assigns each agent i an *advice* A_i



Information theoretic approach

Advice complexity

Given k agents, the advice complexity $f(k)$ is the maximum $\#$bits used for representing the advice of an agent

Information theoretic approach

Advice complexity

Given k agents, the advice complexity $f(k)$ is the maximum $\#$bits used for representing the advice of an agent

State complexity

Note, a lower bound f on the advice complexity implies a lower bound of 2^f on the $\#$ of possible advices (states) when coming out of the nest

Main theorem [Feinerman and Korman, DISC 2012]

Theorem

For every $0 < \epsilon \leq 1$, *whatever algorithm ants use:*
if the search time is $\leq \log^{1-\epsilon} k \cdot (D + D^2/k)$ *then the advice complexity is* $\epsilon \log \log k - O(1)$

Main theorem [Feinerman and Korman, DISC 2012]

Theorem

For every $0 < \epsilon \leq 1$*, whatever algorithm ants use:*
if the search time is $\leq \log^{1-\epsilon} k \cdot (D + D^2/k)$ *then the advice complexity is* $\epsilon \log\log k - O(1)$

Corollary

If time is $T \leq \log^{1-\epsilon} k \cdot (D + D^2/k)$ *then number of states when coming out of the nest is* $S = \Omega(\log^{\epsilon} k)$

Main theorem [Feinerman and Korman, DISC 2012]

Theorem

For every $0 < \epsilon \leq 1$*, whatever algorithm ants use:*
if the search time is $\leq \log^{1-\epsilon} k \cdot (D + D^2/k)$ *then the advice complexity is* $\epsilon \log\log k - O(1)$

Corollary

If time is $T \leq \log^{1-\epsilon} k \cdot (D + D^2/k)$ *then number of states when coming out of the nest is* $S = \Omega(\log^{\epsilon} k)$

Remarks

- Results are asymptotically tight
- Hidden constants are small

A novel scientific framework?

Combine the theoretical lower bound with an experiment on living ants

A novel scientific framework?

Combine the theoretical lower bound with an experiment on living ants

1. Measure the search time – approximate T as a function of k and D (relatively easy)

A novel scientific framework?

Combine the theoretical lower bound with an experiment on living ants

1. Measure the search time - approximate T as a function of k and D (relatively easy)
2. If the search time $T < \log^{1-\epsilon} k \cdot (D + D^2/k)$ then the number of states of ants when coming out of the nest is $\Omega(\log^{\epsilon} k)$

Outline

Conclusions and future work

- This work is a proof of concept for a novel scientific framework

Conclusions and future work

- This work is a proof of concept for a novel scientific framework
- To fully illustrate it there is a need for experimental work. This will undoubtedly require some tuning in model and theoretical results

Conclusions and future work

- This work is a proof of concept for a novel scientific framework
- To fully illustrate it there is a need for experimental work. This will undoubtedly require some tuning in model and theoretical results
- The framework can be applied to other biological contexts. What about bacteria? tradeoffs between efficiency and communication?

Conclusions and future work

- This work is a proof of concept for a novel scientific framework
- To fully illustrate it there is a need for experimental work. This will undoubtedly require some tuning in model and theoretical results
- The framework can be applied to other biological contexts. What about bacteria? tradeoffs between efficiency and communication?

Thanks!

Simplified proof (for a weaker version of the lower bound)

Lemma

If running time is $o(\log k \cdot (D + D^2/k))$ then advice > 0.

Simplified proof (for a weaker version of the lower bound)

Lemma

If running time is $o(\log k \cdot (D + D^2/k))$ then advice > 0.

Proof

- Assume all agents start with the same data-structure no matter what is k

Simplified proof (for a weaker version of the lower bound)

Lemma

If running time is $o(\log k \cdot (D + D^2/k))$ then advice > 0.

Proof

- Assume all agents start with the same data-structure no matter what is k
- Assume running in time is $(D + \frac{D^2}{k}) \cdot \phi(k)$ (and $\phi(\cdot)$ is non-decreasing). I.e., the expected time to visit u is $T_u \leq (d(u,s) + \frac{d(u,s)^2}{k}) \cdot \phi(k)$.

Simplified proof (for a weaker version of the lower bound)

Lemma

If running time is $o(\log k \cdot (D + D^2/k))$ then advice > 0.

Proof

- Assume all agents start with the same data-structure no matter what is k
- Assume running in time is $(D + \frac{D^2}{k}) \cdot \phi(k)$ (and $\phi(\cdot)$ is non-decreasing).
  I.e., the expected time to visit u is $T_u \leq (d(u,s) + \frac{d(u,s)^2}{k}) \cdot \phi(k)$.
- Fix W (upper bound on $\#$ agents)

Simplified proof (for a weaker version of the lower bound)

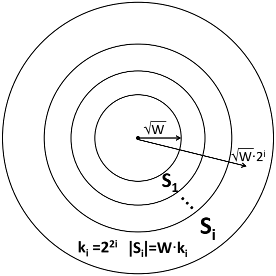
Lemma

If running time is $o(\log k \cdot (D + D^2/k))$ then advice > 0.

Proof

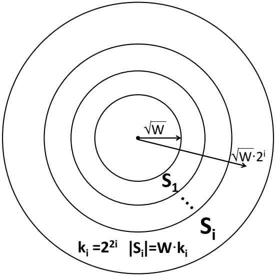
- Assume all agents start with the same data-structure no matter what is k
- Assume running in time is $(D + \frac{D^2}{k}) \cdot \phi(k)$ (and $\phi(\cdot)$ is non-decreasing). I.e., the expected time to visit u is $T_u \leq (d(u,s) + \frac{d(u,s)^2}{k}) \cdot \phi(k)$.
- Fix W (upper bound on $\#$ agents)
- Structure of proof: we show that by time $T = 2W \cdot \phi(W)$, an ant is expected to visit many nodes: $\approx W \cdot \log(W)$. Since she can visit at most one node in 1 time unit, it follows that we cannot have $\phi(W) = o(\log W)$.

Simplified proof (cont.)



- Fix $i = 1, 2, \cdots, \frac{\log W}{2} - 1$, and consider
  $S_i := \{u \mid \sqrt{W} \cdot 2^{i-1} < d(u,s) \leq \sqrt{W} \cdot 2^i\}$. Note, $|S_i| \approx W \cdot 2^{2i}$

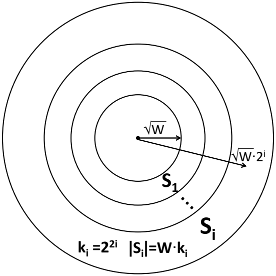
Simplified proof (cont.)



- Fix $i = 1, 2, \cdots, \frac{\log W}{2} - 1$, and consider
  $S_i := \{u \mid \sqrt{W} \cdot 2^{i-1} < d(u, s) \leq \sqrt{W} \cdot 2^i\}$. Note, $|S_i| \approx W \cdot 2^{2i}$

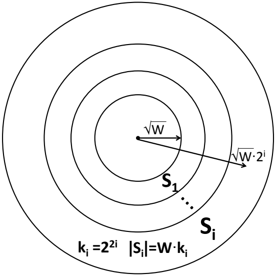
- Assume now that $k_i = 2^{2i}$. So, $|S_i| \approx W \cdot k_i$. Note that $k_i < W$.

Simplified proof (cont.)



- Fix $i = 1, 2, \cdots, \frac{\log W}{2} - 1$, and consider
  $S_i := \{u \mid \sqrt{W} \cdot 2^{i-1} < d(u,s) \leq \sqrt{W} \cdot 2^i\}$. Note, $|S_i| \approx W \cdot 2^{2i}$
- Assume now that $k_i = 2^{2i}$. So, $|S_i| \approx W \cdot k_i$. Note that $k_i < W$.
- Moreover, $k_i = 2^{i+1} \cdot 2^{i-1} \leq \sqrt{W} \cdot 2^{i-1}$. I.e., $k_i \leq d(u,s)$, $\forall u \in S_i$.

Simplified proof (cont.)



- Fix $i = 1, 2, \cdots, \frac{\log W}{2} - 1$, and consider
  $$S_i := \{u \mid \sqrt{W} \cdot 2^{i-1} < d(u,s) \leq \sqrt{W} \cdot 2^i\}. \quad \text{Note, } |S_i| \approx W \cdot 2^{2i}$$
- Assume now that $k_i = 2^{2i}$. So, $|S_i| \approx W \cdot k_i$. Note that $k_i < W$.
- Moreover, $k_i = 2^{i+1} \cdot 2^{i-1} \leq \sqrt{W} \cdot 2^{i-1}$. I.e., $k_i \leq d(u,s)$, $\forall u \in S_i$.
- Therefore,
  $$T_u \leq (d(u,s) + \frac{d(u,s)^2}{k_i}) \cdot \phi(k_i) \leq 2 \cdot \frac{d(u,s)^2}{k_i} \cdot \phi(k_i) < 2W \cdot \phi(W) = T.$$

Simplified proof (cont.)

- So, probability of visiting $u \in S_i$ by time $2T$ is at least $1/2$.

Simplified proof (cont.)

- So, probability of visiting $u \in S_i$ by time $2T$ is at least $1/2$.
- Thus, the expected number of nodes in S_i that all agents visit by time $2T$ is roughly $|S_i| \approx W \cdot k_i$. Hence, the expected number of nodes in S_i that one agent visits by time $2T$ is $|S_i|/k_i \approx W$.

Simplified proof (cont.)

- So, probability of visiting $u \in S_i$ by time $2T$ is at least $1/2$.
- Thus, the expected number of nodes in S_i that all agents visit by time $2T$ is roughly $|S_i| \approx W \cdot k_i$. Hence, the expected number of nodes in S_i that one agent visits by time $2T$ is $|S_i|/k_i \approx W$.
- Observe, this holds $\forall i \in [1, \frac{\log W}{2})$.

Simplified proof (cont.)

- So, probability of visiting $u \in S_i$ by time $2T$ is at least $1/2$.
- Thus, the expected number of nodes in S_i that all agents visit by time $2T$ is roughly $|S_i| \approx W \cdot k_i$. Hence, the expected number of nodes in S_i that one agent visits by time $2T$ is $|S_i|/k_i \approx W$.
- Observe, this holds $\forall i \in [1, \frac{\log W}{2})$.
- Hence, the expected number of nodes that a single agent visits by time $2T$ is $\approx W \cdot \log W$. As $T \approx W \cdot \phi(W)$, this implies that we cannot have $\phi(W) = o(\log W)$. □