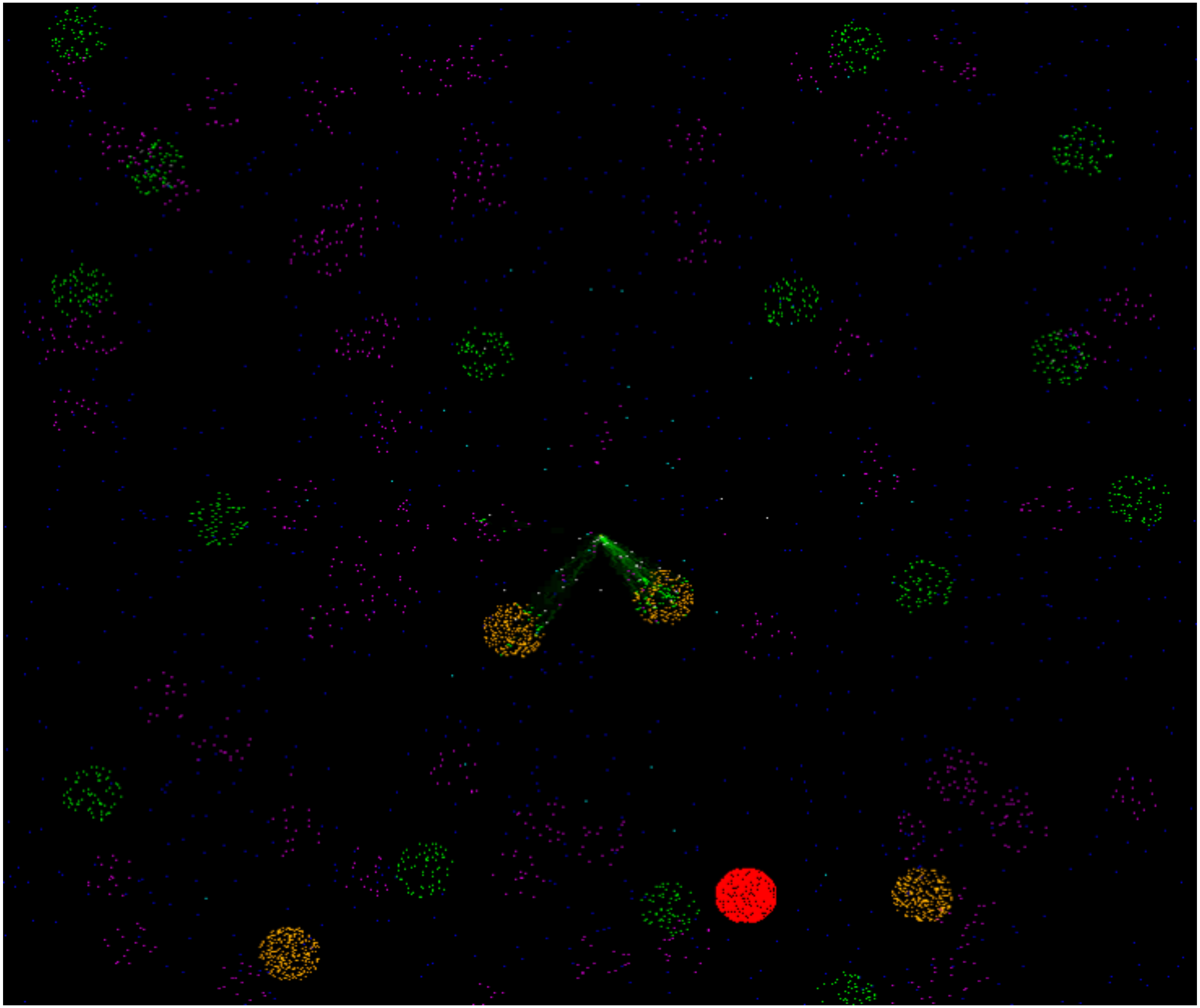
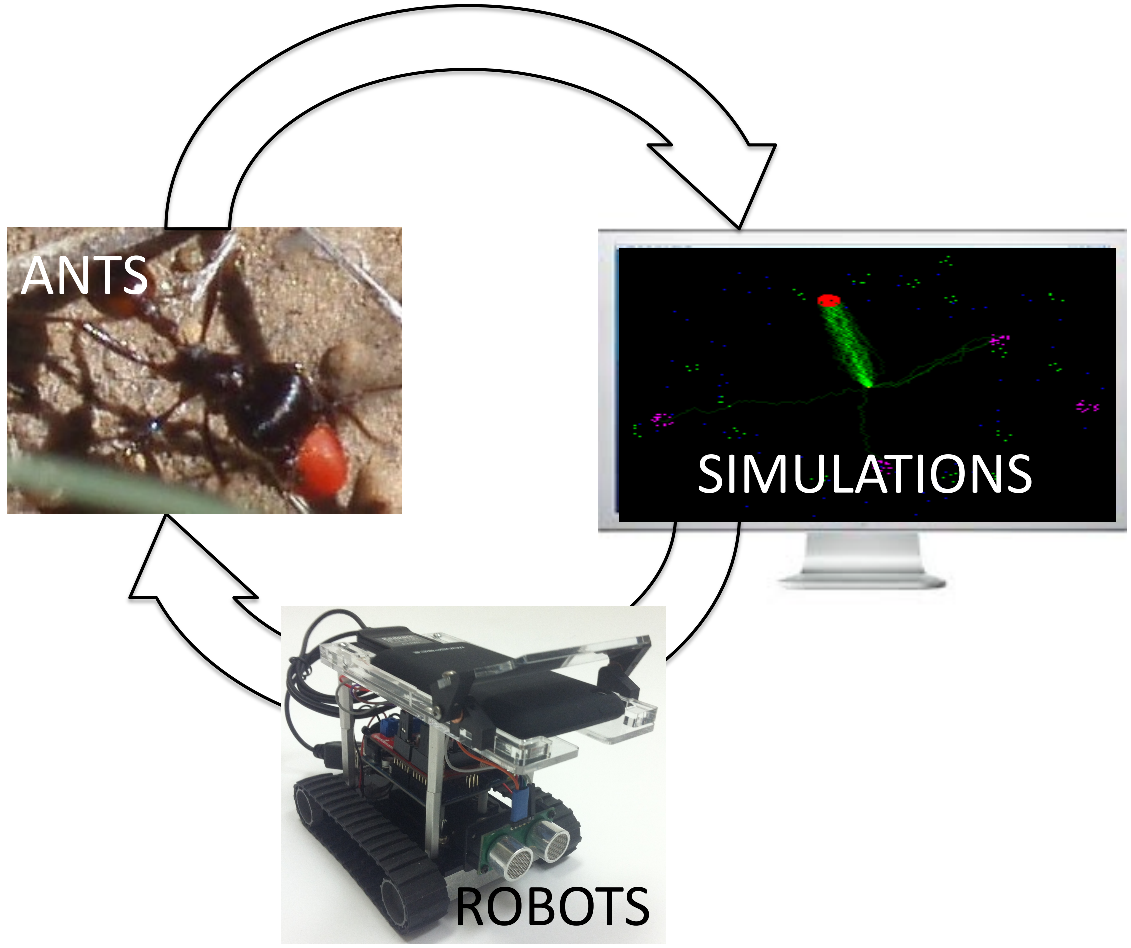


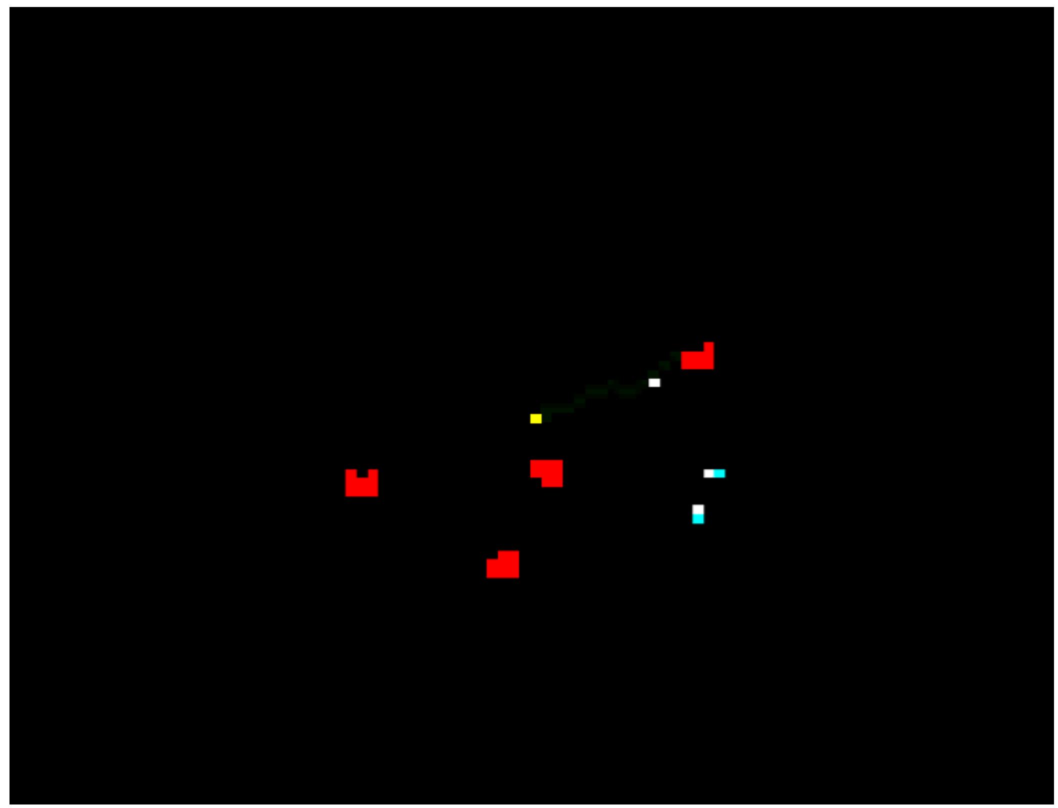

ANTS
SIMULATIONS
ROBOTS

Change size, speed, #, to match robots

Compare simulations to real robots

Change simulation so GA evolves
good parameters for real robots

How much do error & noise matter?



Scalability

~20 Ants

Competition

"The foreign policy of ants can be summed up as follows: restless aggression, territorial conquest, and genocidal annihilation of neighboring colonies whenever possible.

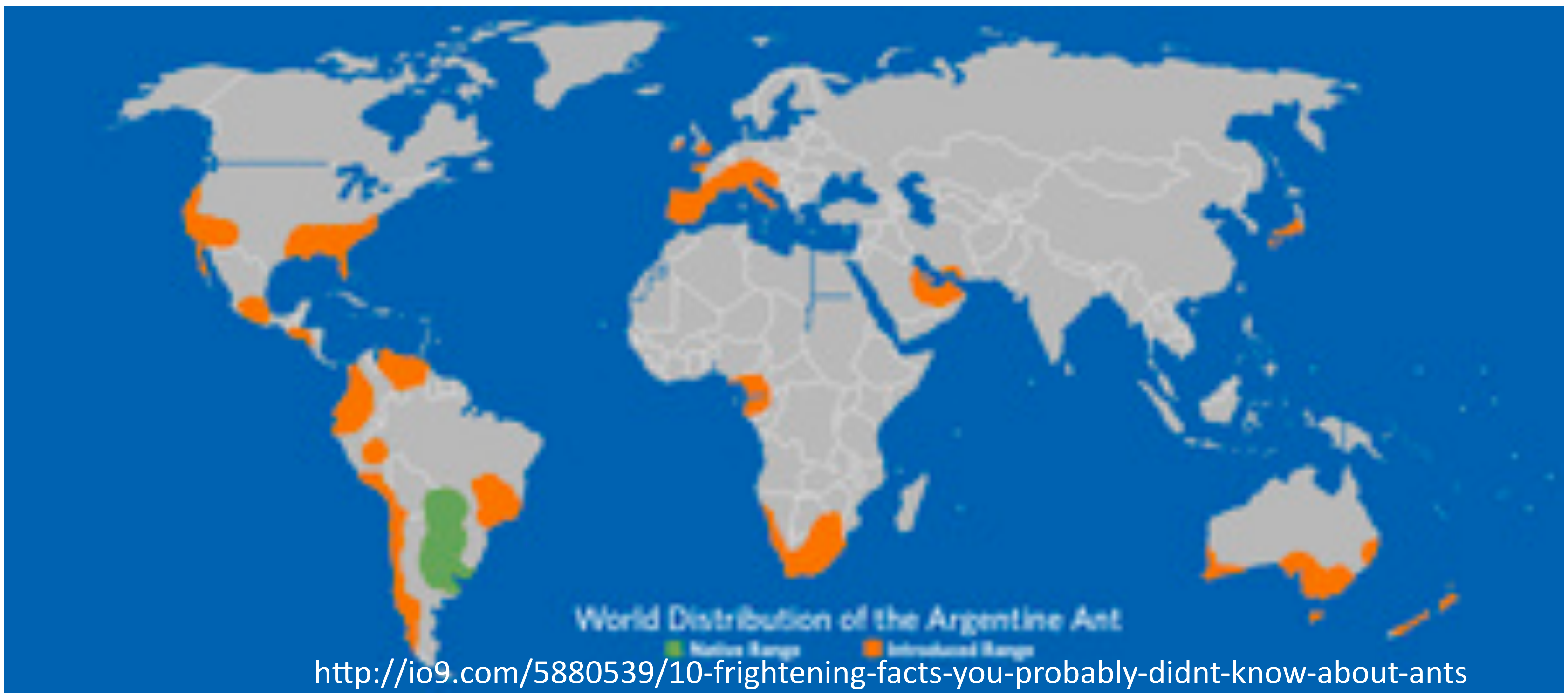
If ants had nuclear weapons, they would probably end the world in a week."

-Bert Holldobler & Edward O. Wilson, Journey to the Ants

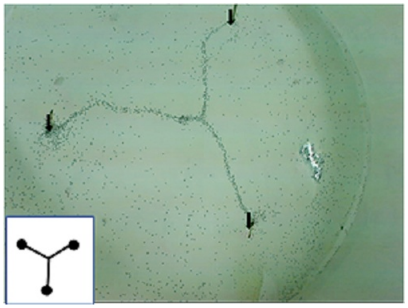
Overdispersed Nests © Alex Wild

Cooperation

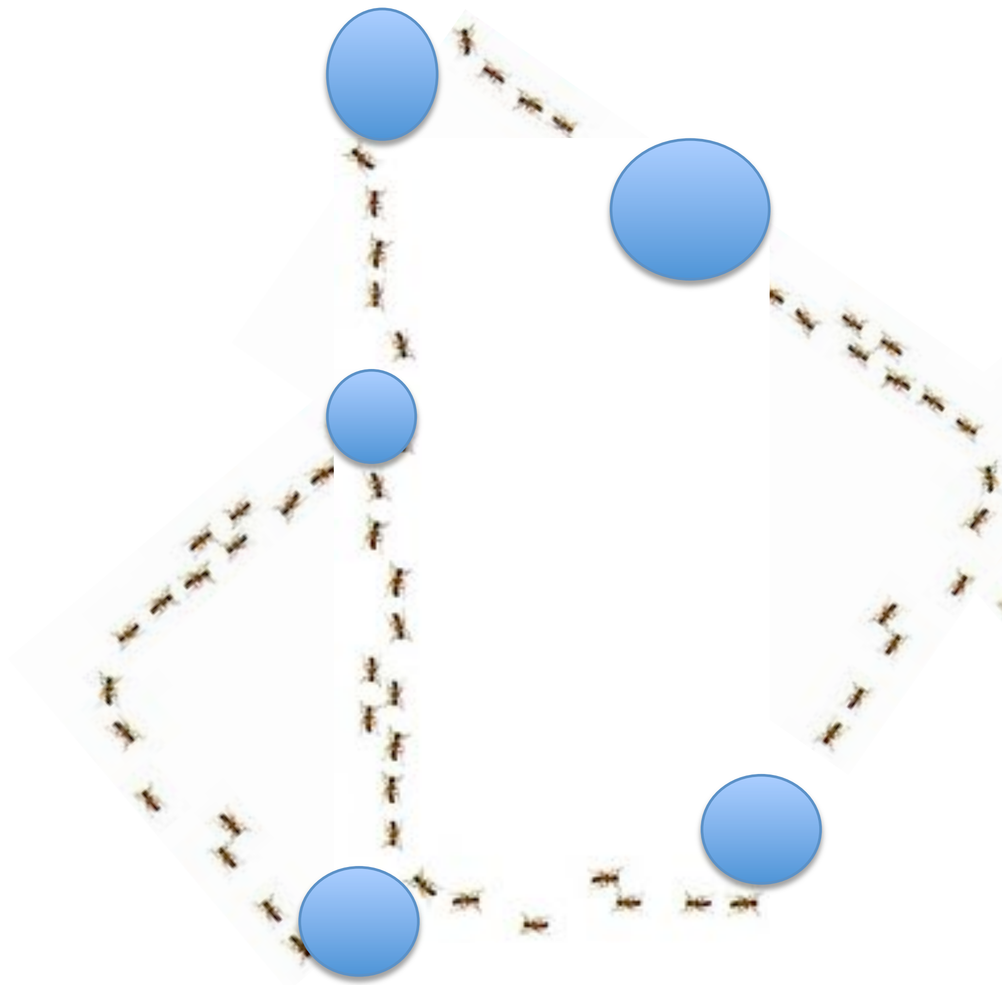
http://www.ecolibrary.org/page/DP162

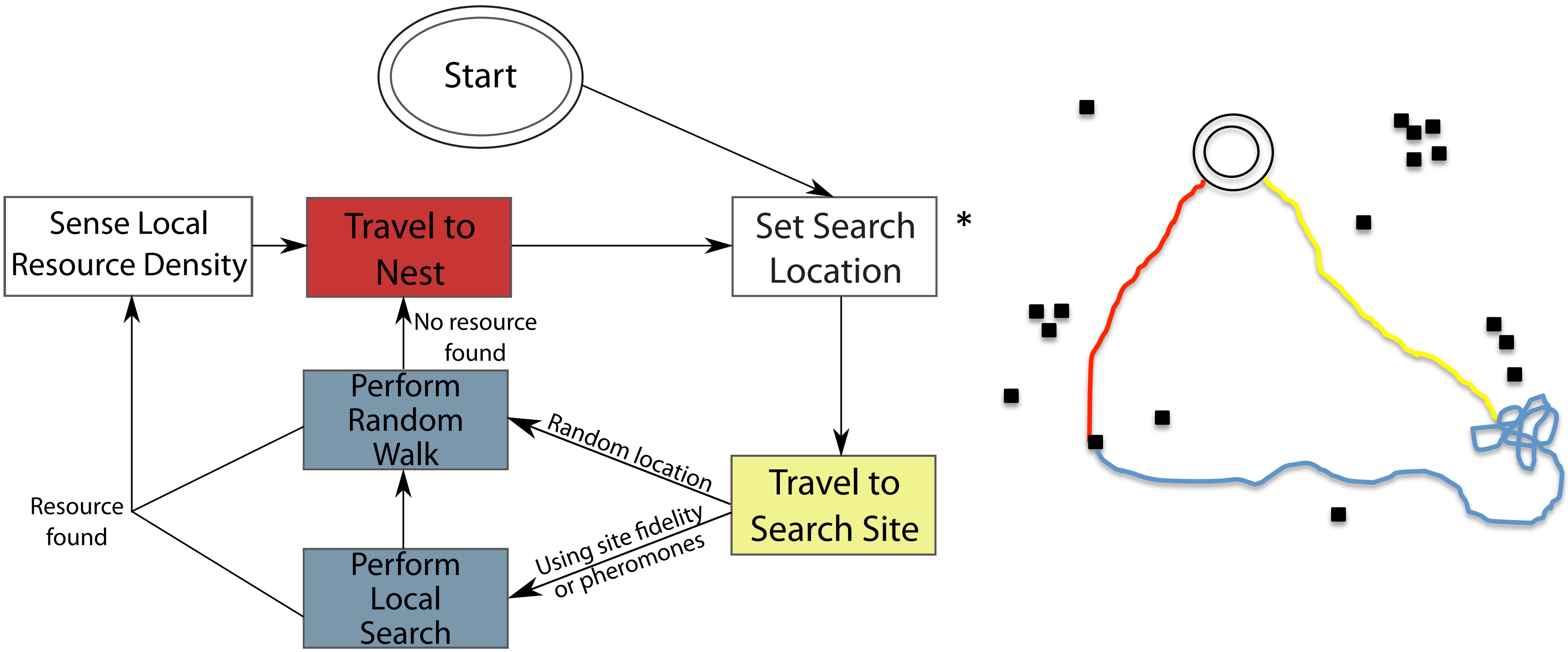


http://io9.com/5880539/10-frightening-facts-you-probably-didnt-know-about-ants



Multiple cooperative nests in invasive Argentine ants





*Search Location is specified by a direction and probability of stopping
In ants. In robots it's an x,y location (the nest is 0,0)

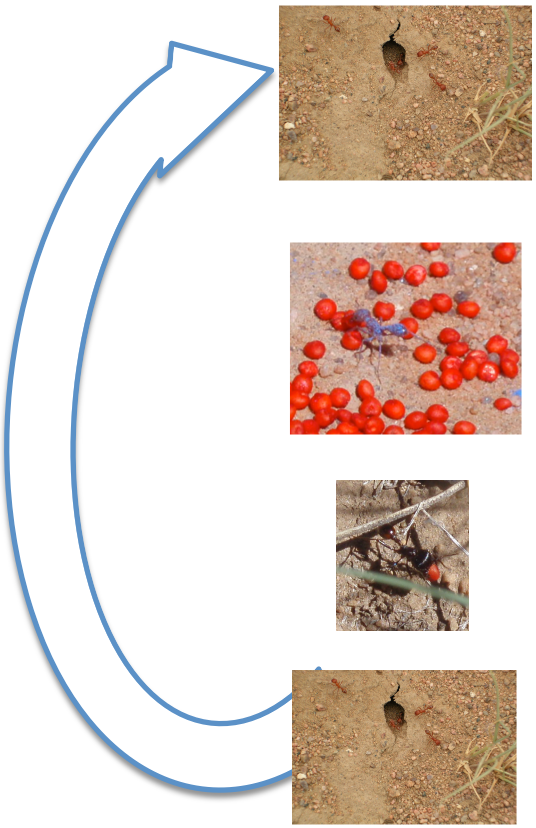
The decision to lay pheromone, return via site fidelity or search in a new location on the next trip depends on Local Resource Density

Memory vs. Communication

private vs. public information

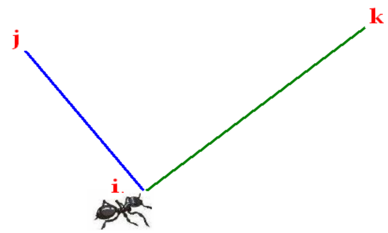
Site Fidelity

There and back again

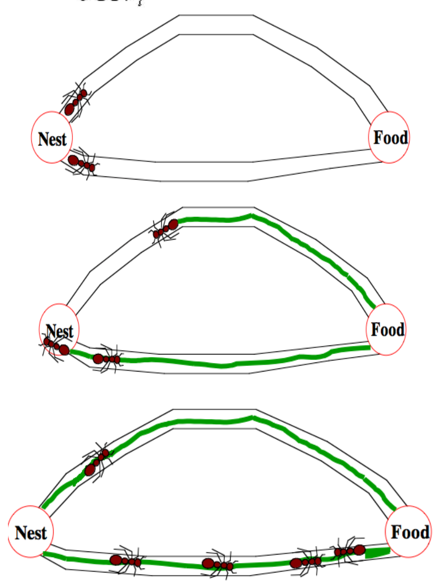


Pheromone Communication

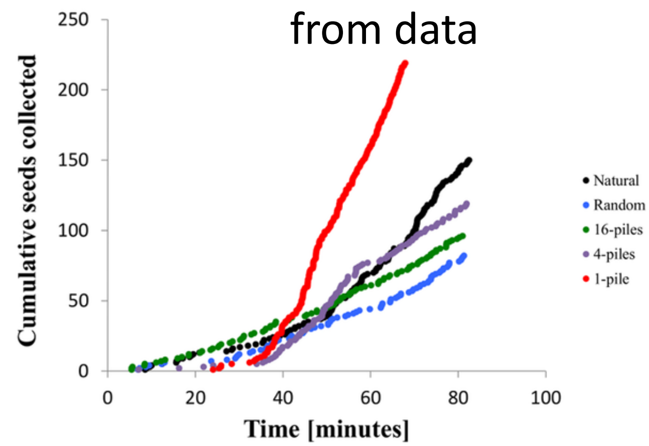
Recruit nestmates



$$p_{ij}^{k}(t)=\frac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{l\in N_i^k}[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}} \quad if\ j\in N_i^k$$



Both processes are indistinguishable from data



Key Model Parameters

- Ants leave nest, walk in a random direction and begin to search

  with probability *α*

- Searching ants move in a correlated random walk, turns draw from a normal distribution

mean $\theta_t = \theta_{t-1}$

$$SD = \omega + \gamma / t_s^{\delta}$$

***ω, γ, δ:* control degree of turning**

- Ants decide to recruit on the return trip depending on local resource concentration, *C*.

pheromone laid with $p_r = \lambda_r + C / \mu_r$

site fidelity with $p_s = \lambda_s + C / \mu_s$

pheromone followed with $p_t = \lambda_t - C / \mu_t$

truncated [0,1]

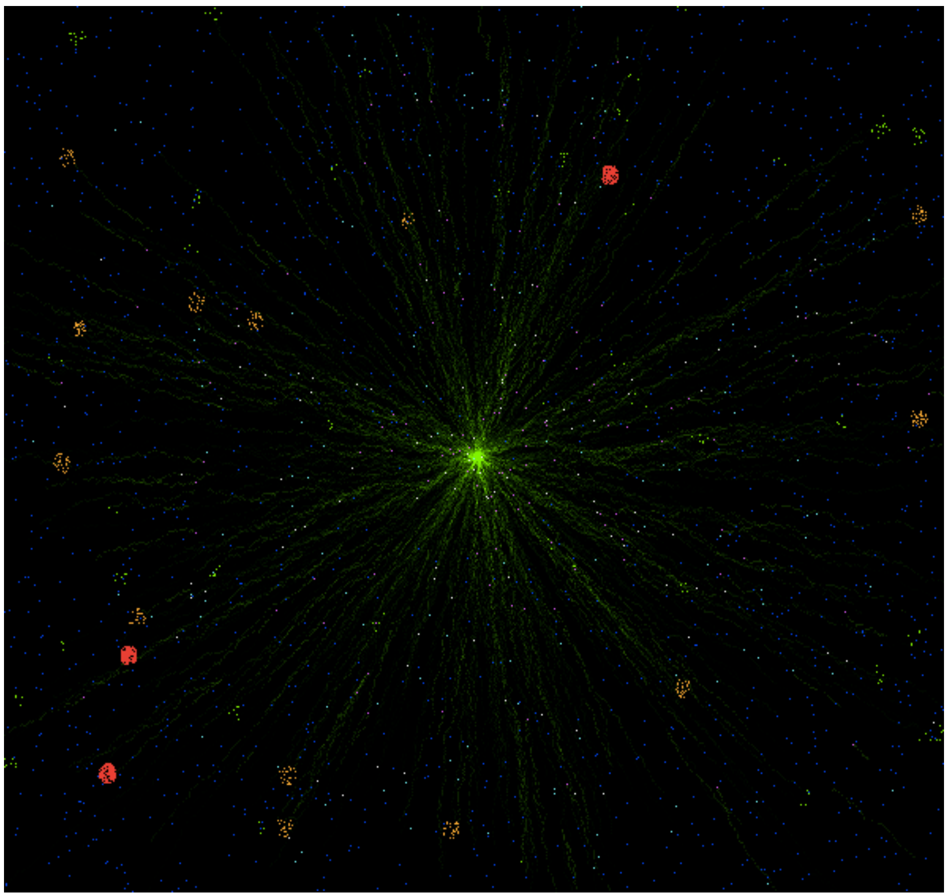
Pheromone evaporation

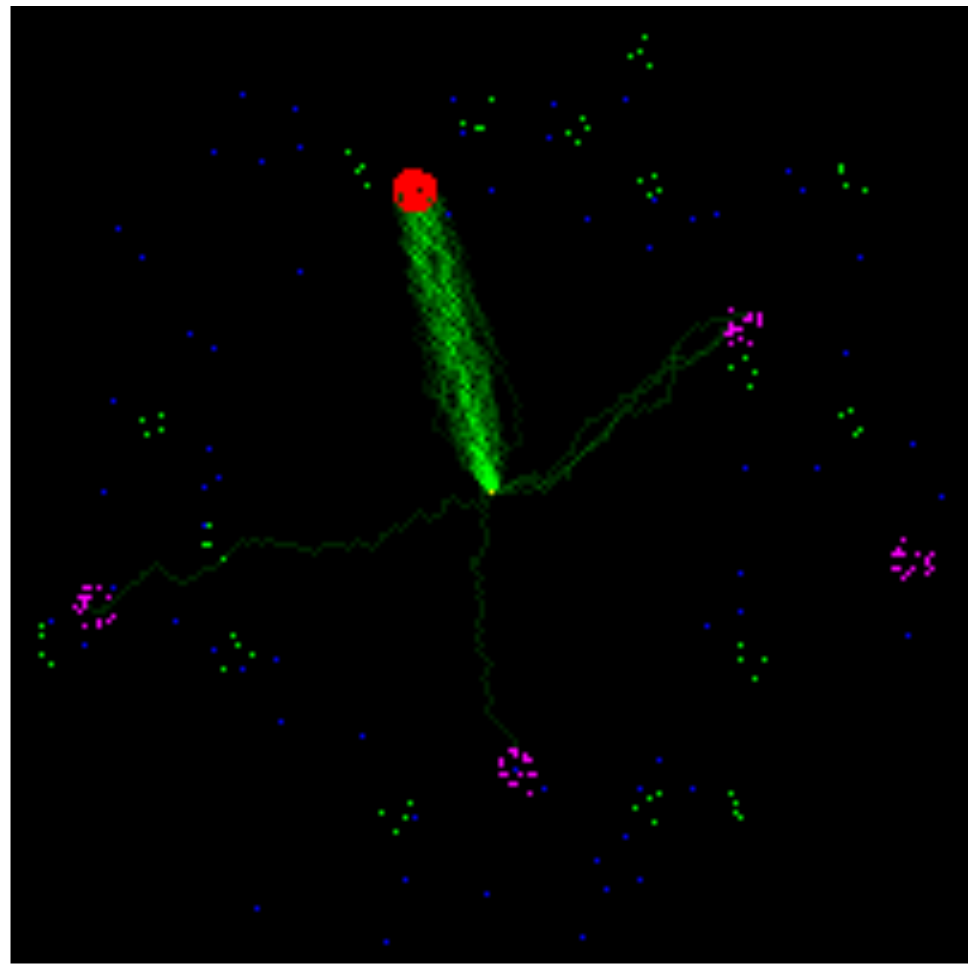
$$\Pi_{x,y,t} = \Pi_{x,y,t-1} * (1 - \eta)$$

Parameter		**Function**	
walk_drop_rate	α	Probability each time step that an ant walking from the nest will stop walking and begin to search.	
search_giveup_rate	-	probability of giving up search and returning to nest during the random walk.	
dir_dev_const	ω	For searching ants moving in a correlated random walk, ω determines the baseline degree of deviation in the direction an ant will move from one time step to the next. See equation 1.	$\mathrm{SD} = \omega + \gamma / t_s^{\delta}$ (1)
dir_dev_coeff2	γ	Determines the additional degree of deviation in turning early on in an ant's search. See equation 1.	
dir_time_pow2	δ	Exponent determines how quickly turning behavior approaches the baseline turning behavior as time spent searching (t_s) increases. See equation 1.	
trail_drop_rate	ε	For ants following a pheromone trail, determines the probability each time step that an ant will abandon the trail and begin searching before reaching its end.	
dense_thresh	λ_r	Determines ants' constant probability of recruiting to a site when picking up food. See equation 2.	$p_r = \lambda_r + C / \mu_r$ (2)
dense_const	μ_r	Determines how ants' probability of recruiting to a site responds to the count C of additional food in neighboring cells. See equation 2.	
dense_thresh_patch	λ_s	Determines ants constant probability of returning to a site when picking up food. See equation 3.	$p_s = \lambda_s + C / \mu_s$ (3)
dense_const_patch	μ_s	Determines how ants' probability of returning to a site responds to the count C if additional food in neighboring cells. See equation 3.	
dense_thresh_influence	μ_t	Determines ants constant probability of following trails when departing the nest. See equation 4.	$p_t = \lambda_t - C / \mu_t$ (4)
dense_const_influence	λ_t	Determines how ants' probability of following trails when departing the nest responds to additional food in neighboring cells at the last location it picked up food. See equation 4.	
decay_rate	η	Determines the rate at which pheromones evaporate. See equation 6.	$\Pi_{x,y,t} = \Pi_{x,y,t-1} * (1 - \eta)$ (6)
dir_dev_coeff1		unused	
dir_time_pow1		unused	
dense_sens		unused	
		The following parameters have no effect in the posted code, but can be used to adjust the proportion of ants that forage	
prop_active		Proportion of ants that forage at the start of the simulation–set to 1	
activate_sensitivity		Likelihood an ant leaves the nest based on # of encounters with incoming individuals	
decay_rate_return		Determines the length of time an ant remembers contacted incoming individuals	

3 behaviors determine collective foraging rate

Search via travel + random walk, increasingly biased over time
Balance site fidelity & pheromones
Decision dependent on local seed density





Genetic Algorithms select parameters to maximize seeds collected in fixed time Group Selection Experiments *in silico*

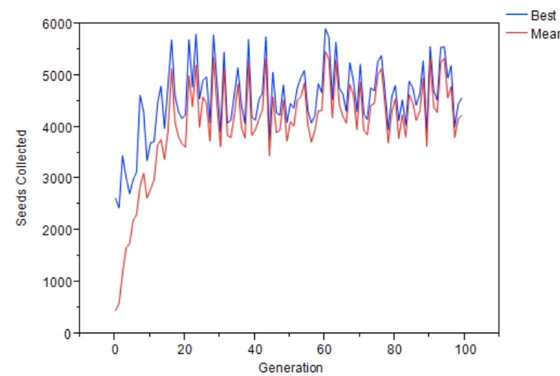
Each model run requires a set of input parameters [*α, ω, γ, δ, λ, μ, η, ε, . . .*]

Each individual in a colony is identical

"Simulated Evolution" (group selection)

G0: [*α,ω, γ, δ, λ, μ, η, ε*] x [*α,ω, γ, δ, λ, μ, η, ε*]

G1: [*α',ω, γ, δ, λ, μ, η, ε*]



100 runs with different parameter sets (individuals) for 100 Generations

Each colony, each generation evaluated on 8 grids for 20,000 time steps

Colonies with highest 'fitness' (seeds collected) replicate into next generation

Crossover & Mutation rates = 10%

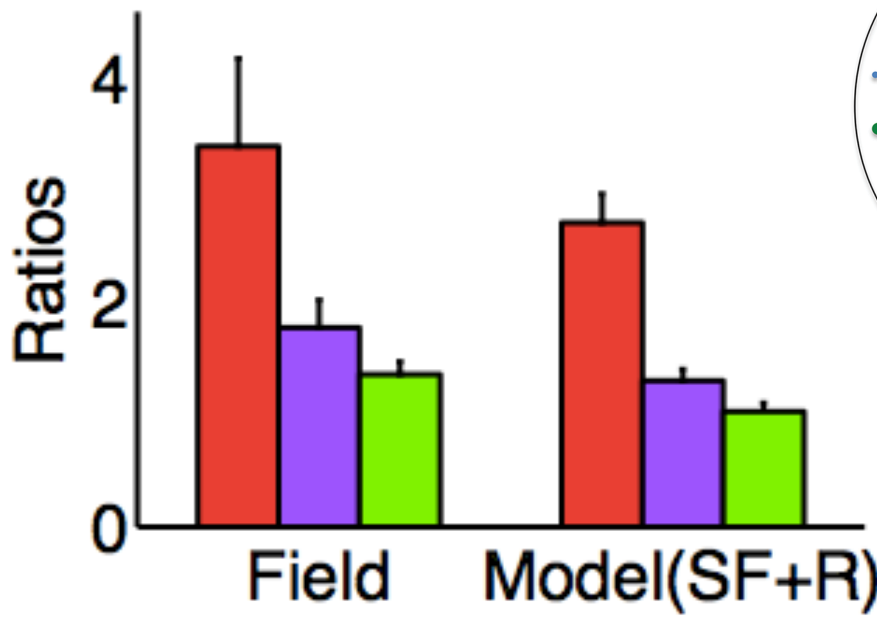
Run for colony sizes 10, 100, 1000, 10,000 foragers

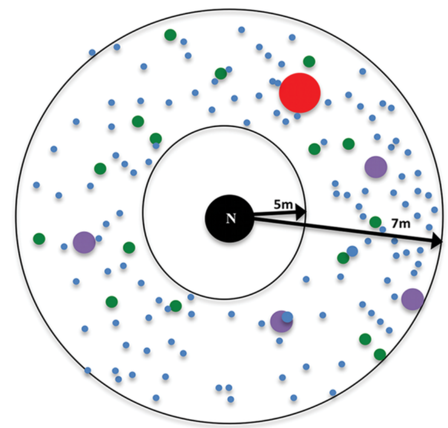
RESULT: A simulated colony 'evolved' to maximize foraging rate

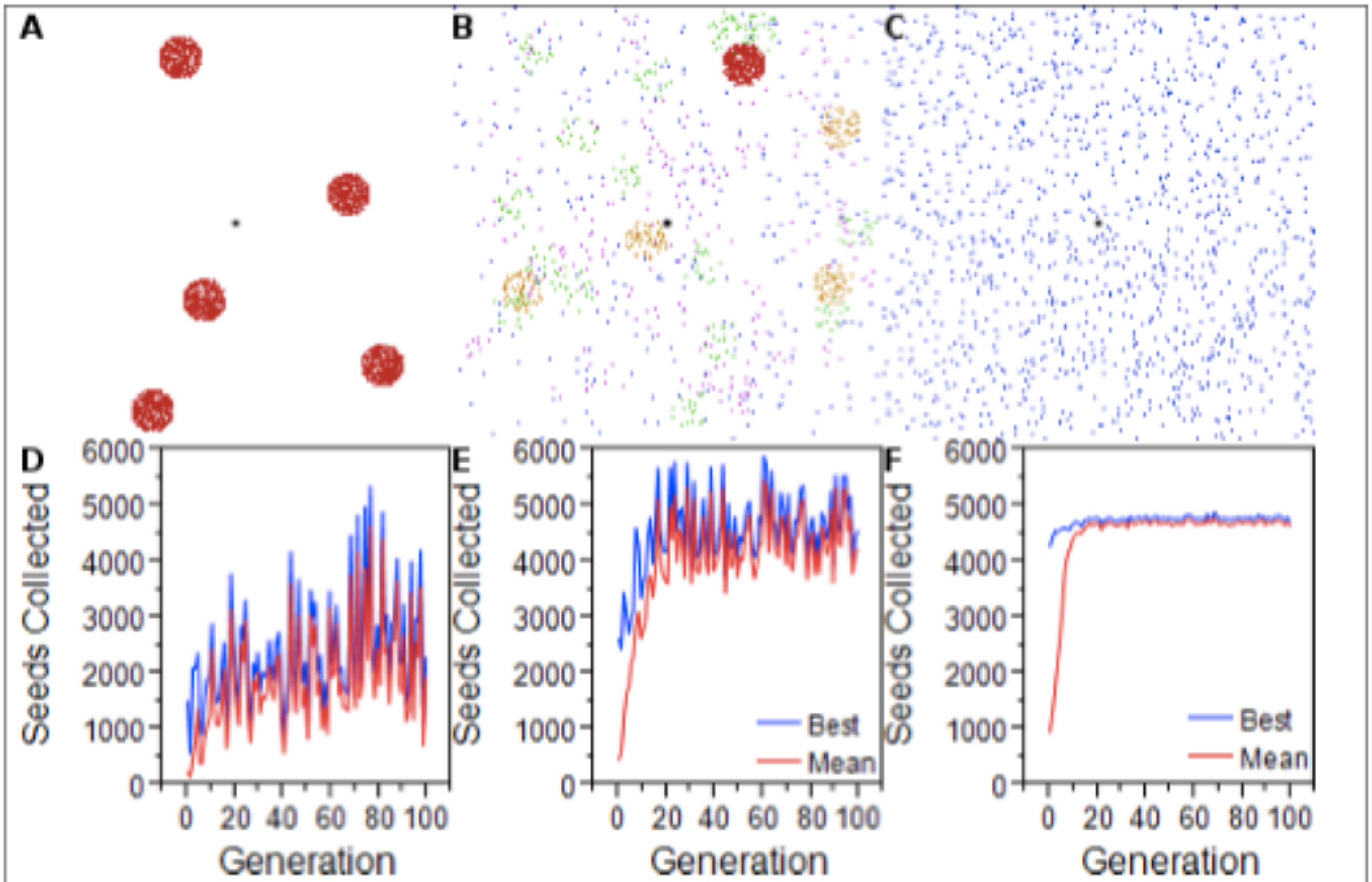
Model maximizes seed collection by balancing site fidelity & pheromone use

Foraging rates vs pile size, indistinguishable from field data

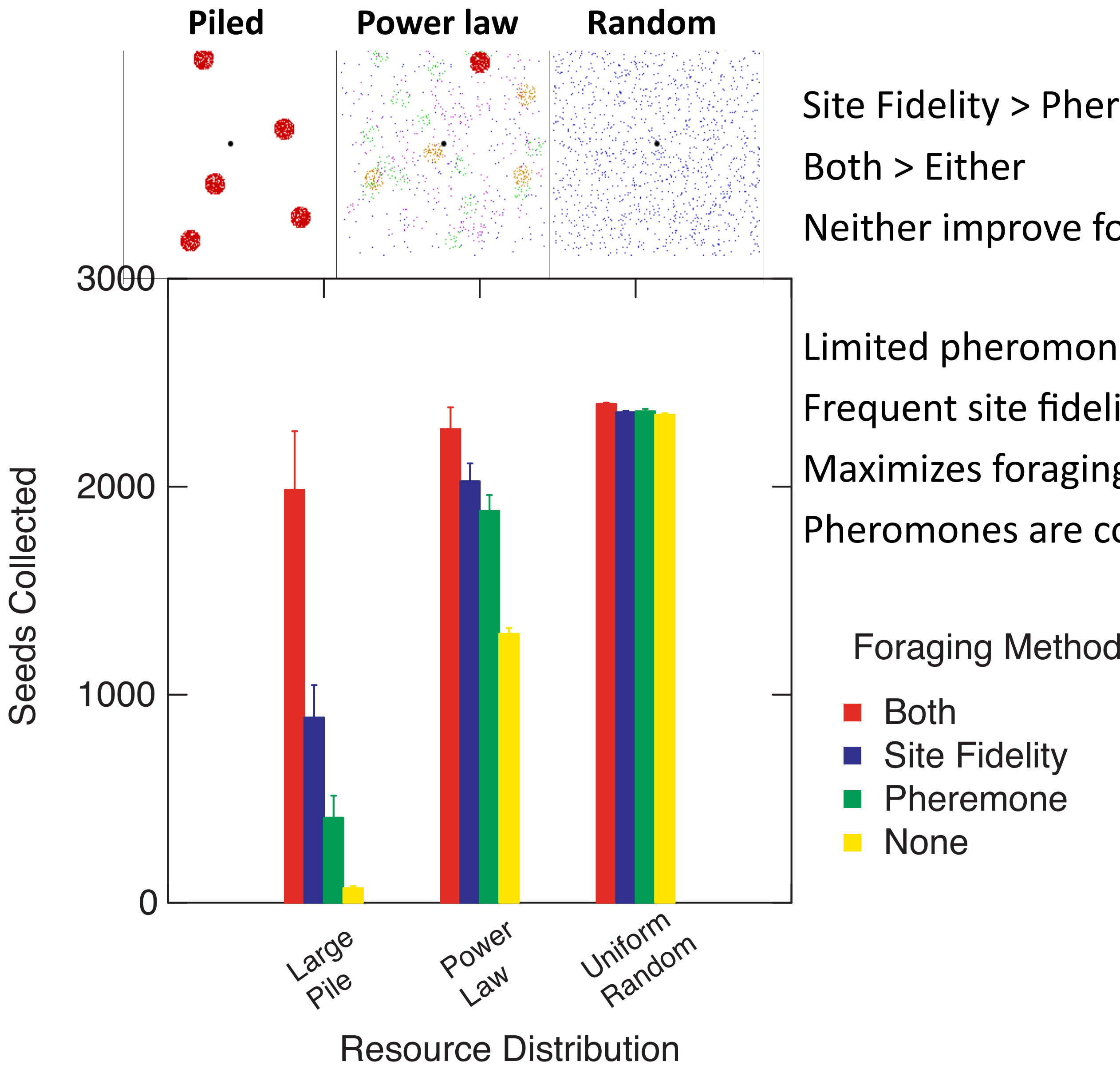
Territory area, indistinguishable from field data
Seeds collected per ant declines size with a -¼ power






A
B
C
D
E
F
Seeds Collected
0
1000
2000
3000
4000
5000
6000
0 20 40 60 80 100
Generation
Best
Mean

Value of memory and communication changes with food distribution



Site Fidelity > Pheromones

Both > Either

Neither improve foraging for random seeds

Limited pheromone use (< 5%) +

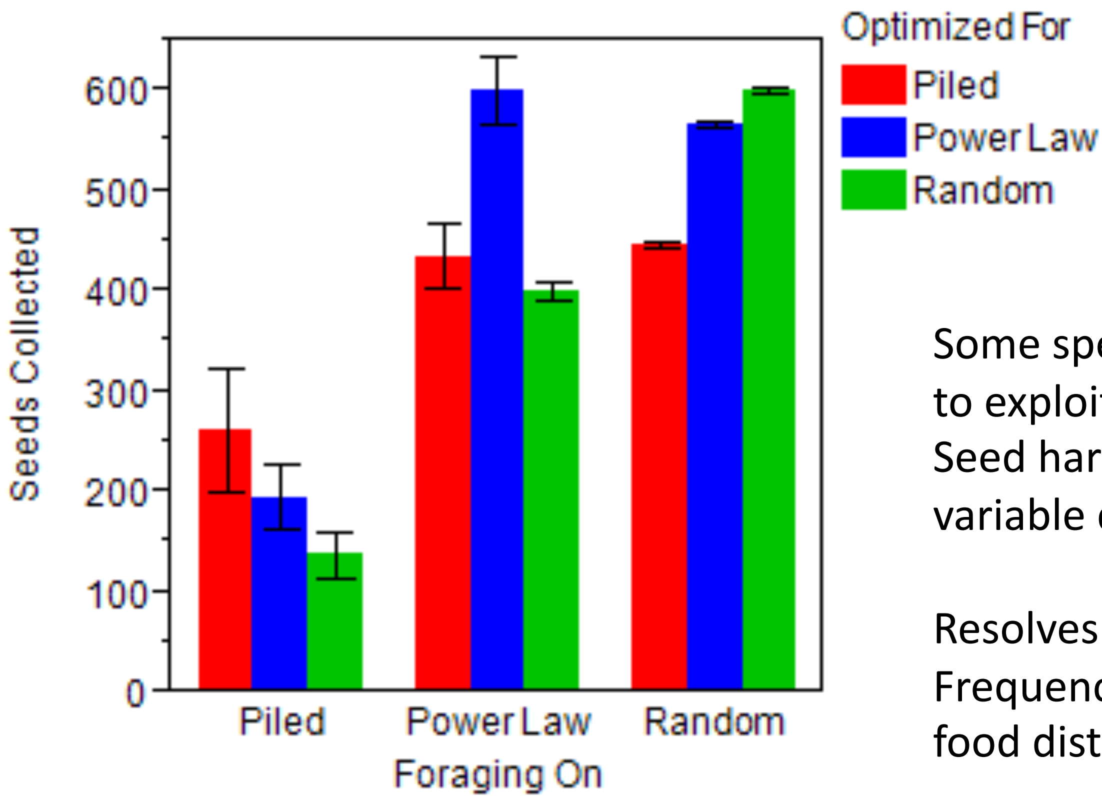
Frequent site fidelity (> 95%)

Maximizes foraging rate

Pheromones are contribute to scalability

Letendre & Moses in review

Power law evolved colonies are most versatile



Some species may have evolved strategies to exploit a particular food distribution
Seed harvesters are likely adapted to variable distributions

Resolves existing biological debate:
Frequency of pheromone use depends on food distribution, longevity, competition…

Rare pheromones substantially improve foraging in some environments

Letendre & Moses in review

Project 2

- Normally, randomization seeds should be stored, but because replicability is difficult given the way code is threaded, it's not needed for this assignment
- Table 2, Part 1: Wide standard deviations are OK

- Try runs with small # of generations & interactions to initially test parameters
- Cooperative Fitness is sum of seeds collected by both colonies
- Part 2.2: you decide how to best demonstrate how and why your changes to xover, mut & selection were or were not effective
- Flexibility in 2.3—These are examples of changes you can try
  - Alter fitness to equal time to collect all seeds
  - Evolve colonies to cooperate or compete with themselves (2 colonies have identical genomes) or to compute against a a fixed strategy
  - Evolve mutation & xover rates over time
  - Evolve where colonies are placed on the field
  - …. Be creative, but focus on evolving strategies to improve cooperation & competition
  - You decide how to best display your results

- Check your results by running parameters through visrun (see visrun 1.1 posted)
- Review Assignment Information on webpage for turnin & readme
- You will turn in code that
  - demonstrates your evolutionary runs over only a few generations
  - Calls visrun to demonstrate your best parameter sets