CULTURAL ALGORITHMS: KNOWLEDGE-DRIVEN PROBLEM SOLVING IN SOCIAL SYSTEMS

Robert G. Reynolds, Ph.D.

Department of Computer Science, Wayne State University

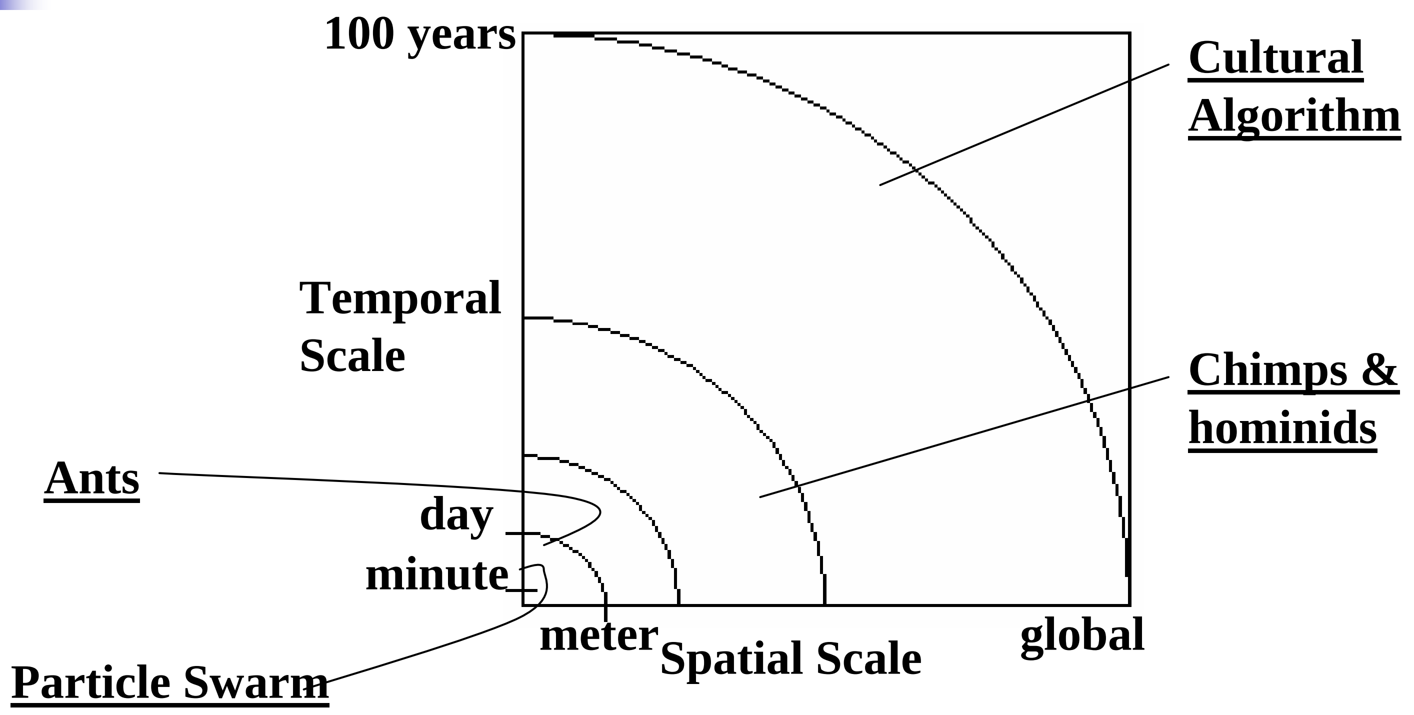
The Emergence of Social Intelligence and Socially Motivated Problem Solving

- Traditionally Artificial Intelligence has focused on the nature of “individual intelligence”. Goal was to create programs that compete with the best “individual”. E.G. Deep Blue in chess.
- With the advent of high performance distributed networks of computers and grid computation there is much incentive to investigate socially motivated problem solving.
- Example frameworks have come from naturally occurring systems.

The Emergence of Social Intelligence and Socially Motivated Problem Solving Continued.

- Particle swarms (colloids) from physical systems. Ant Colony Optimization from Biological Systems. Cultural Algorithms from Cultural Systems.
- Allows us to begin to understand the nature of social intelligence in terms of group problem solving.

Scale of Social Interaction



Socially Motivated Approaches

- Particle Swarm Optimization (Kennedy and Eberhart 1995)
  - Based on social behaviors of birds and fish
  - Particles (potential solutions) move by following its own best and best of its neighbor
- Ant Colony Optimization (Dorigo et. al. 1996)
  - Based on ant colony observation
  - the pheromone and the density of the pheromone along the trail is the knowledge that the ACO shares among its individual ants

Cultural Algorithms as a Vehicle for Socially Motivated Problem Solving

- Cultural Algorithms consist of a social population and a belief space [Reynolds, 1979]
- Derived from models of Cultural Evolution in Anthropology [Flannery,2000, 1986]
- Provides a framework to accumulate and communicate knowledge so as to allow self-adaptation in an evolving model.
- Encompasses the scales of the other approaches so should be able to solve their types of problems as well.

Cultural Algorithms

- Have been applied to modeling many problems in the evolution of complex social organizations. E.G. Simulation of the Ancient Anazazi (Scientific American, July 2005), Organization of Design Knowledge (Rychtycjky and Reynolds,2005)
- Here we are interested in stripping down the cultural algorithm to its computational minimum in order to understand how cultures solve problems.

Example: El Farol Problem

- Individuals decide whether to visit the El Farol bar. Fun when there are not too many there, but less fun when too crowded.
- Each individual selects a knowledge model to use based upon the experience of those who used the model last time.

Optimization Problem Solving

- A population of individuals move through a landscape characterized by a distribution of resource cones looking for the highly productive regions.
- Movement of an individual is controlled by a knowledge model.
- Roles emerge when an individual uses the same model repeatedly.

Role Emergence

- In the static case (the cones did not move) two major roles emerged.
- “Explorers” used a combination of knowledge sources over time, and “Exploiters”.
- It turns out that biologists have observed two emergent roles in foraging species, “Producers” and “Scroungers” which correspond well with our emergent roles. (Barnard and Sybley, 1981)
- These roles were produced by the interaction of the knowledge sources and allows us to explain the roles emergence in terms of their shared experience or knowledge.

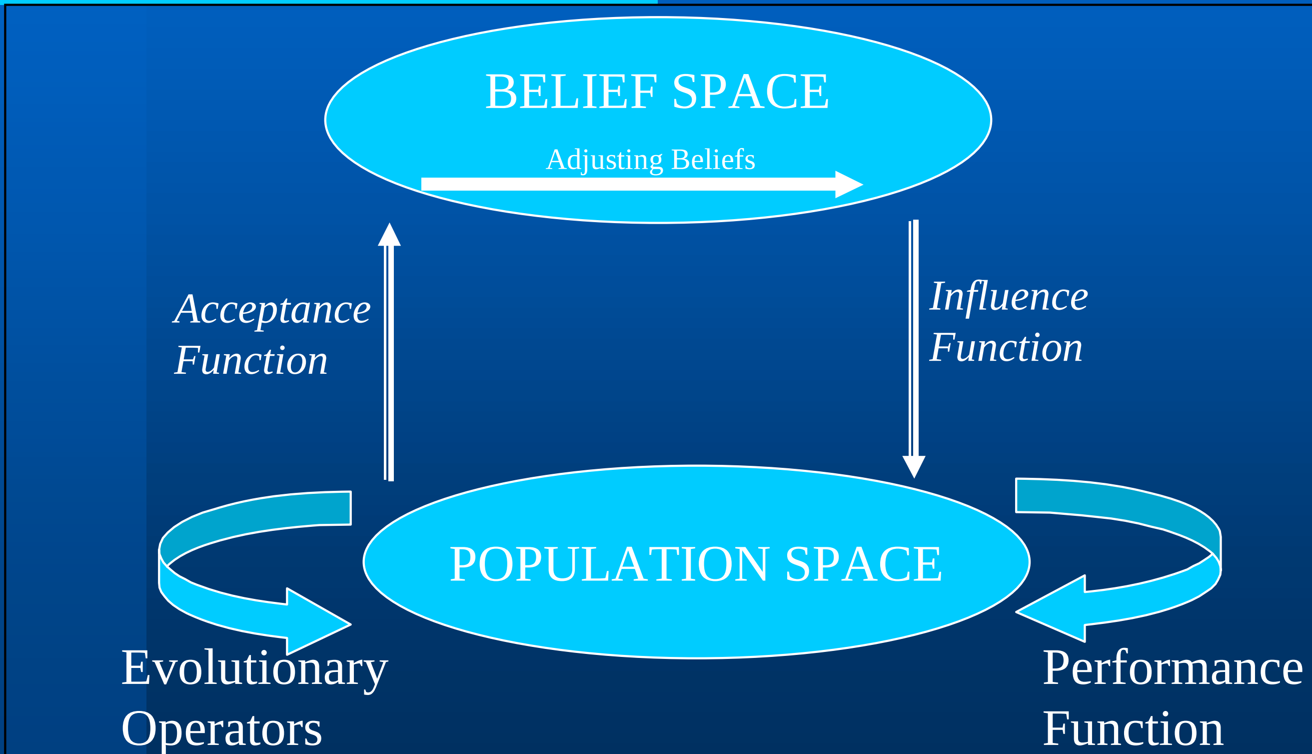
Role Emergence in Dynamic Environments

- Here we allow the environment to be dynamic, but predictable.
- The question is to see how this affects the nature of role emergence here?
- Produces a hierarchy of roles, since the dynamic environment provides more information that can be differentially exploited by the knowledge sources.

Cultural Algorithm Overview

- Population Space
- Belief Space
- Protocol
  - Describes how the Population Space and the Belief Space communicate and interact
  - Acceptance Function
  - Influence Function

Cultural Algorithm Framework



Population Space

- Provides a fundamental base in which individuals reside and interact
- Could accommodate any population-based evolutionary computation model
- Evolutionary Programming (EP) is used in this paper

Framework of EP

Begin

initialize population P;

evaluate(P);

repeat

$P' = P$;

for each individual $P_i \in P$

$P' = P' \cup \text{variate}(P_i)$

end

evaluate(P');

compete(P');

$P = \text{top}(P')$;

until (time exhausted or acceptable solution discovered)

End

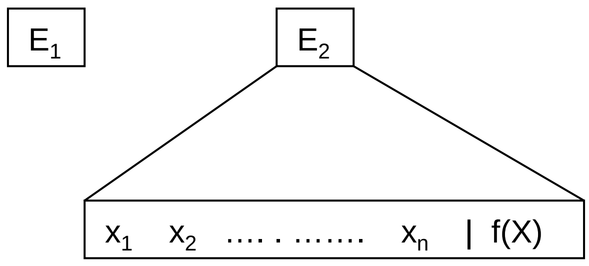
Basic knowledge categories

- Classification due to Saleem & Reynolds
  - Normative
  - Situational
  - Domain
  - History
  - Topographical
- Classification is complete for a given domain

Cognitive Basis for Each Knowledge Type in Pre-Hominids

- Procedural(hard coded)
- Declarative (Bechtel, 1999)
  - Episodic (remembered sequence)
  - Semantic (known facts)
- Concerning social, natural, and technical knowledge for a given domain.
- Natavistic predisposition for acquiring spatial, numeric, and communicational types of knowledge. (Landau, 1999)

Situational Knowledge



- Comprise a set of exemplars $\{E_m\}$ from the population
- Exemplar added from population if better than current best
- Situational knowledge reinitialized if environment changes

Domain Knowledge - Definitions

- Direction relative to an individual $X < x_1, x_2, \ldots, x_n >$
  - direction $\Delta X = < d_1, d_2, \ldots, d_n >$, where d_i = -1, decreasing x_i

    0, being the same

    1, increasing x_i

Domain Knowledge – Definitions

- Gradient $\nabla_x(\Delta X)$

$$\nabla_X(\Delta X) = \frac{obj(X + \Delta X) - obj(X)}{\Delta X}$$

the rate at which the fitness changes relative to change in coordinate values

Domain Knowledge

The best-performance Individual

The steepest-slope Individual

The max∇ (ΔX)
Direction $\Delta X < d_1, d_2, \ldots, d_n >$

The max∇ (ΔX)
Direction $\Delta X < d_1, d_2, \ldots, d_n >$

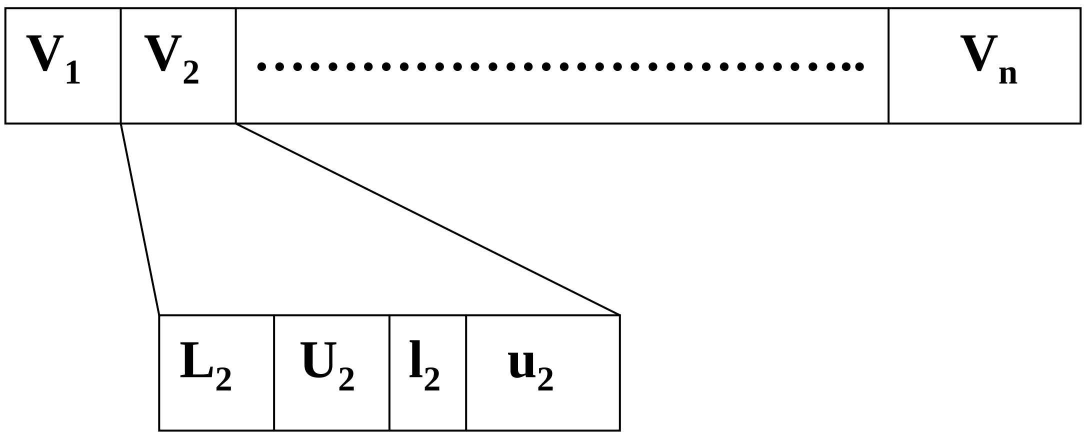
- The direction with steepest gradient max∇, that is, the direction leads to the fastest improvement in fitness

Domain Knowledge

- Use change in fitness values to generate diversity and mutation values
- Updated from population if best individual outperforms current best
- Not reinitialized when environment changes

Normative Knowledge

$$\{(l_1,\ u_1),\ (l_2,\ u_2),\ \ldots,\ (l_n,\ u_n)\}$$



- A set of promising ranges for each parameter.

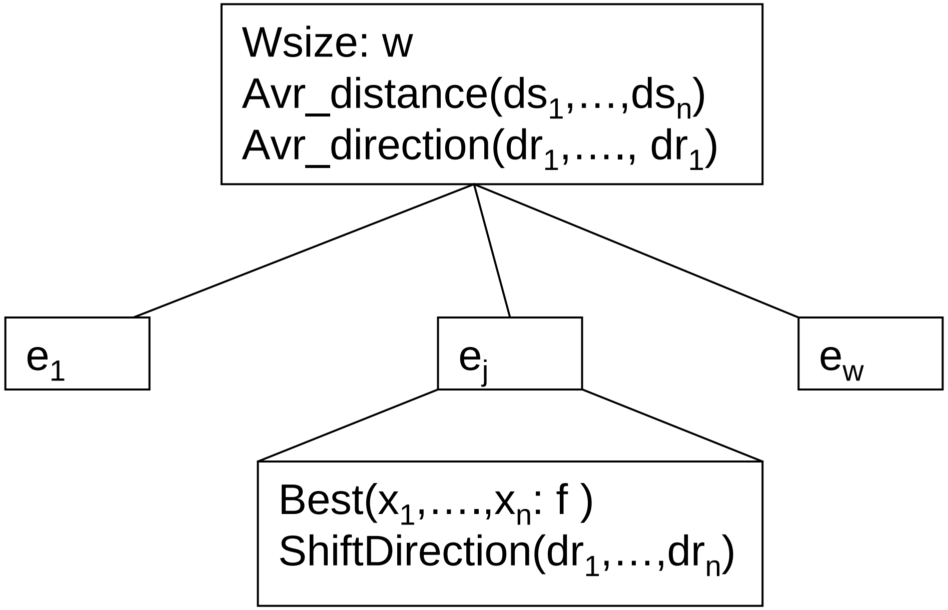
Normative Knowledge

- Represents best estimate of values that produce a good solution
- Interval characterized by upper and lower bounds and the best performance at those limits
- Update bound when better performance is found

History Knowledge

- A list of up to *w* temporal events / points on the search path $\{P_1, P_2, \ldots, P_w\}$
  - *w* is the window size of the history list
  - Each P_j represents a significant point on the search path

Historical Knowledge



- Comprises list of most recent changes in optimum solution, plus average shift direction and distance
- Shift direction is summation over all parameters, adjusted for number of events in memory

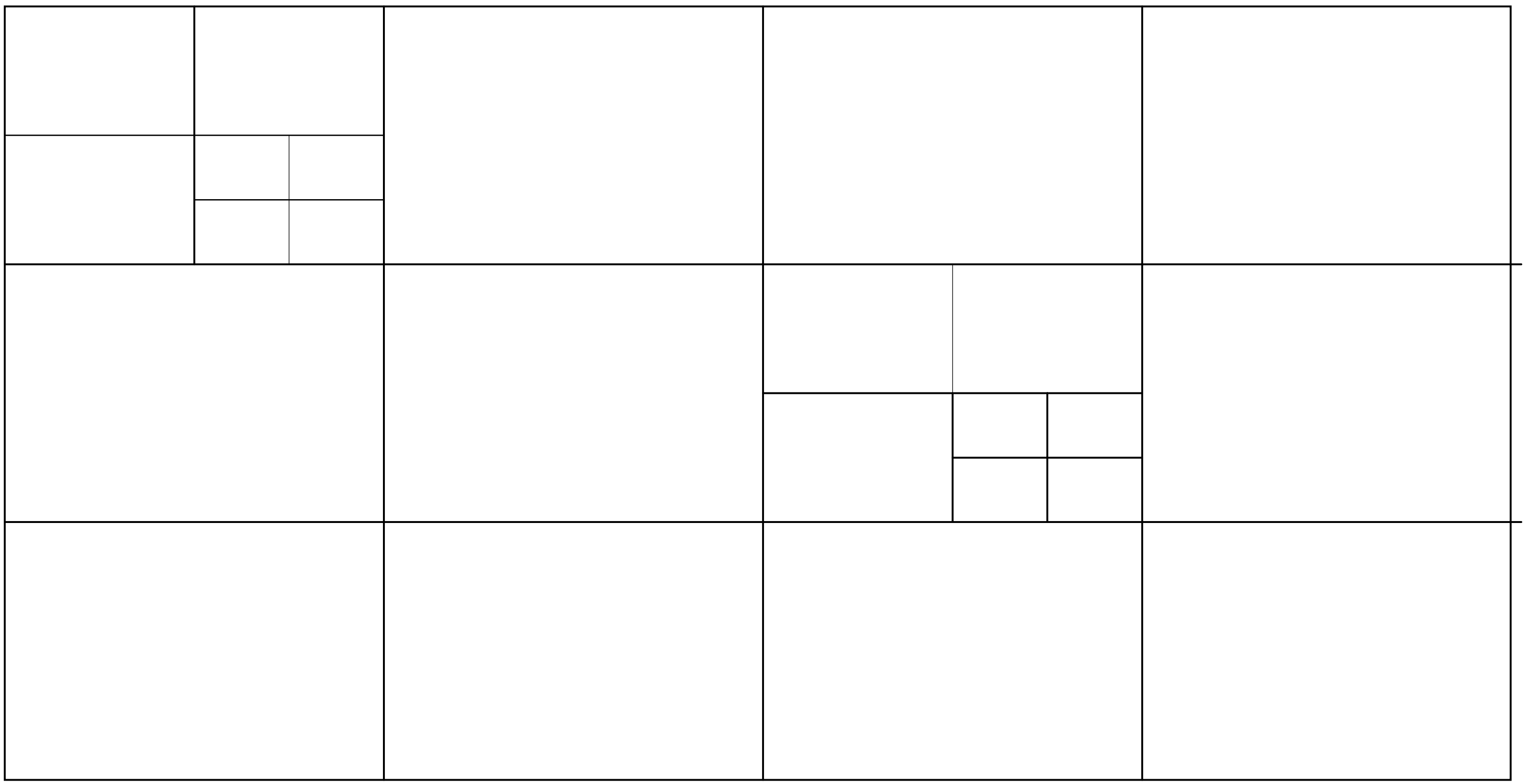
Historical Knowledge

- Memory of environmental changes
  - Local list
    - Records within one landscape
  - Global list
    - For dynamic environment

Topographical Knowledge

- Representation of spatial location of best solutions
- Variable resolution
  - Focus on most productive areas
  - Increased resolution in productive areas
- Hybrid structure
  - Array of cells
  - Linked list of subdivided cells

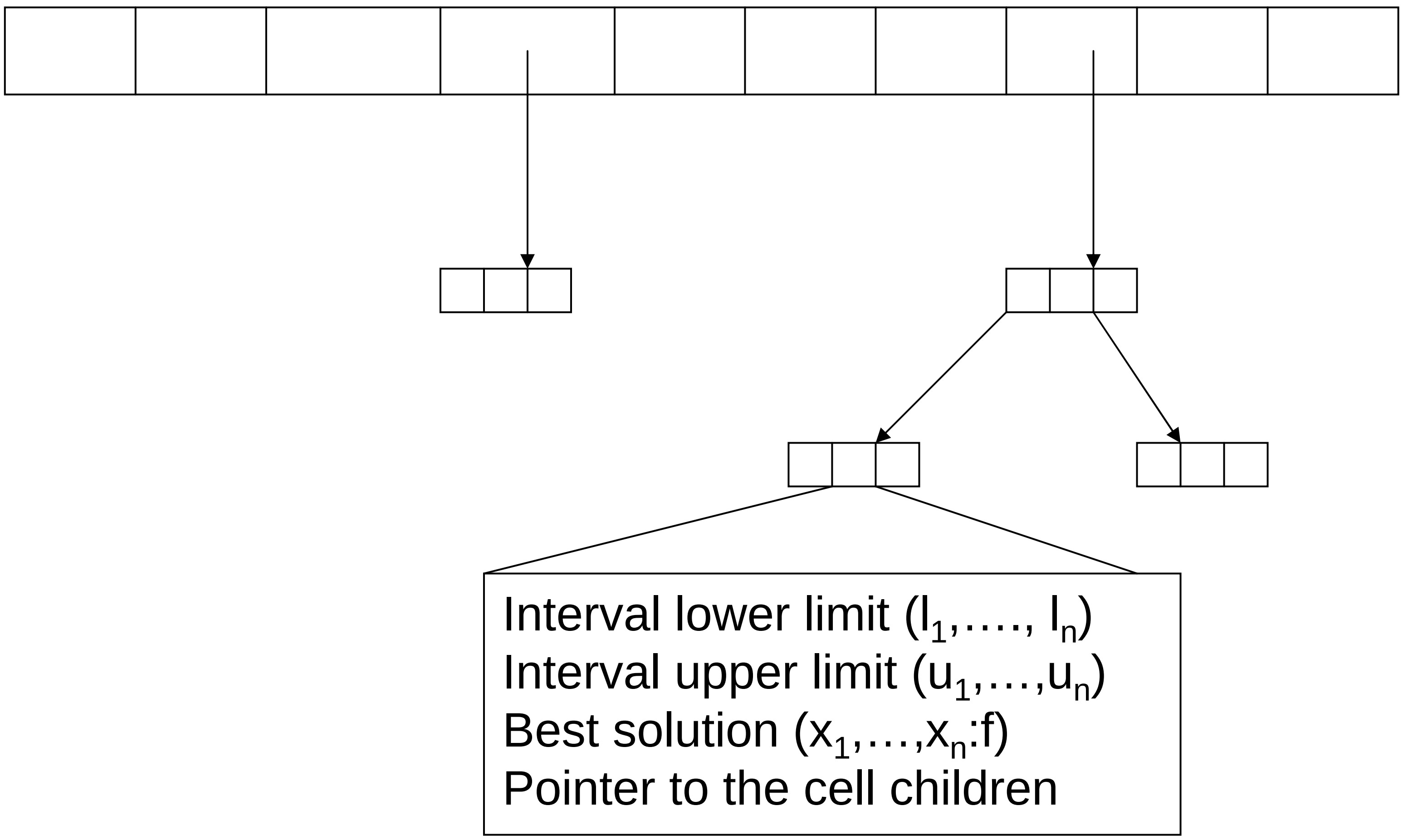
Topographical Knowledge



Topographical Knowledge

- The whole search space is initially divided into t cells {*C*1, *C*2, …, *C*t}
- Each cell could be further divided into smaller cells during the search
- Cells are organized into a hierarchical tree structure
- The initial t cells form the top / root level.
- Cells without sub-cells become the leaf cell.
- All leaf cells form the whole search space.
- Each cell stores the cell-best individual *cb*

Structure of Topographical Knowledge



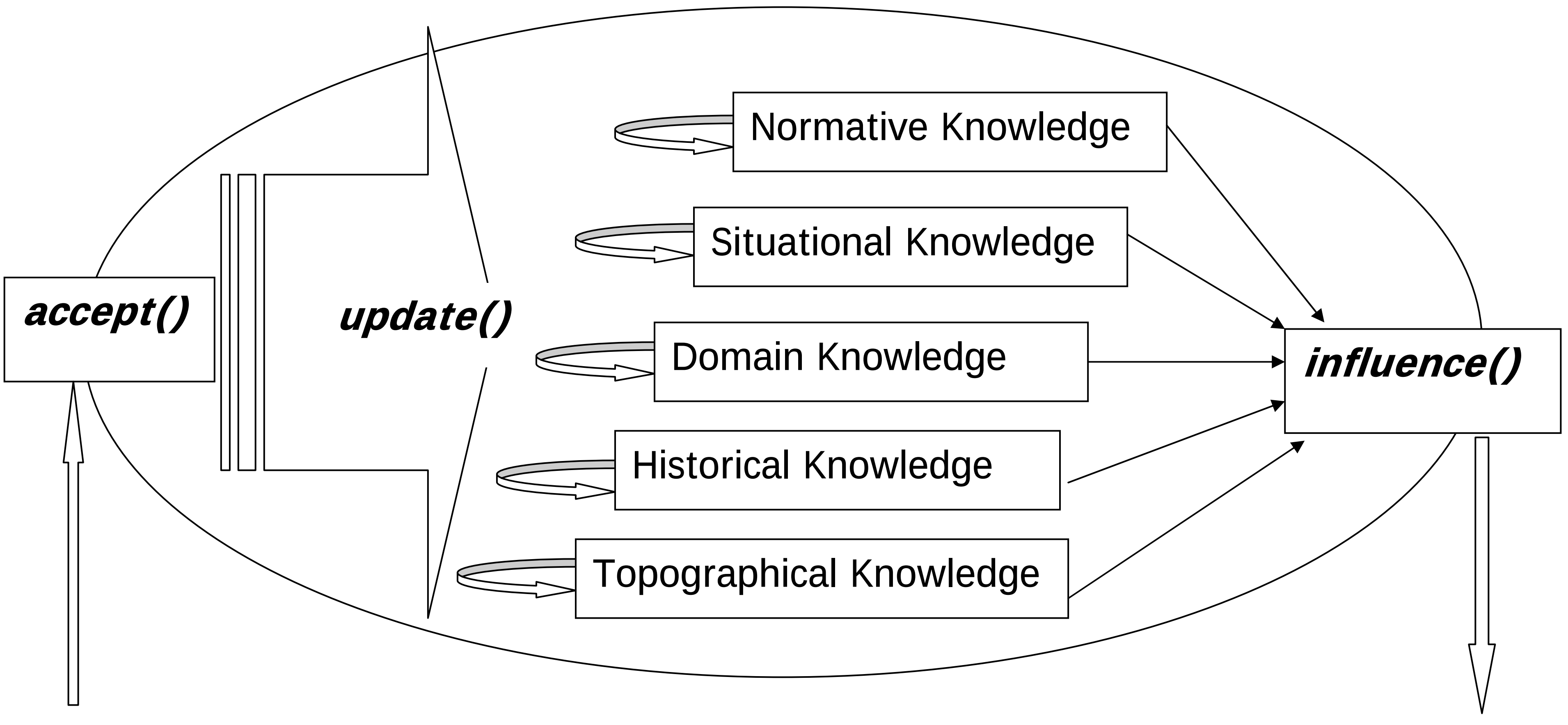
Acceptance Function

- Population Space -> Belief Space
- A subset of the population is chosen to impact the Belief Space. In an optimization problem it is generally a percentage of the top performers.

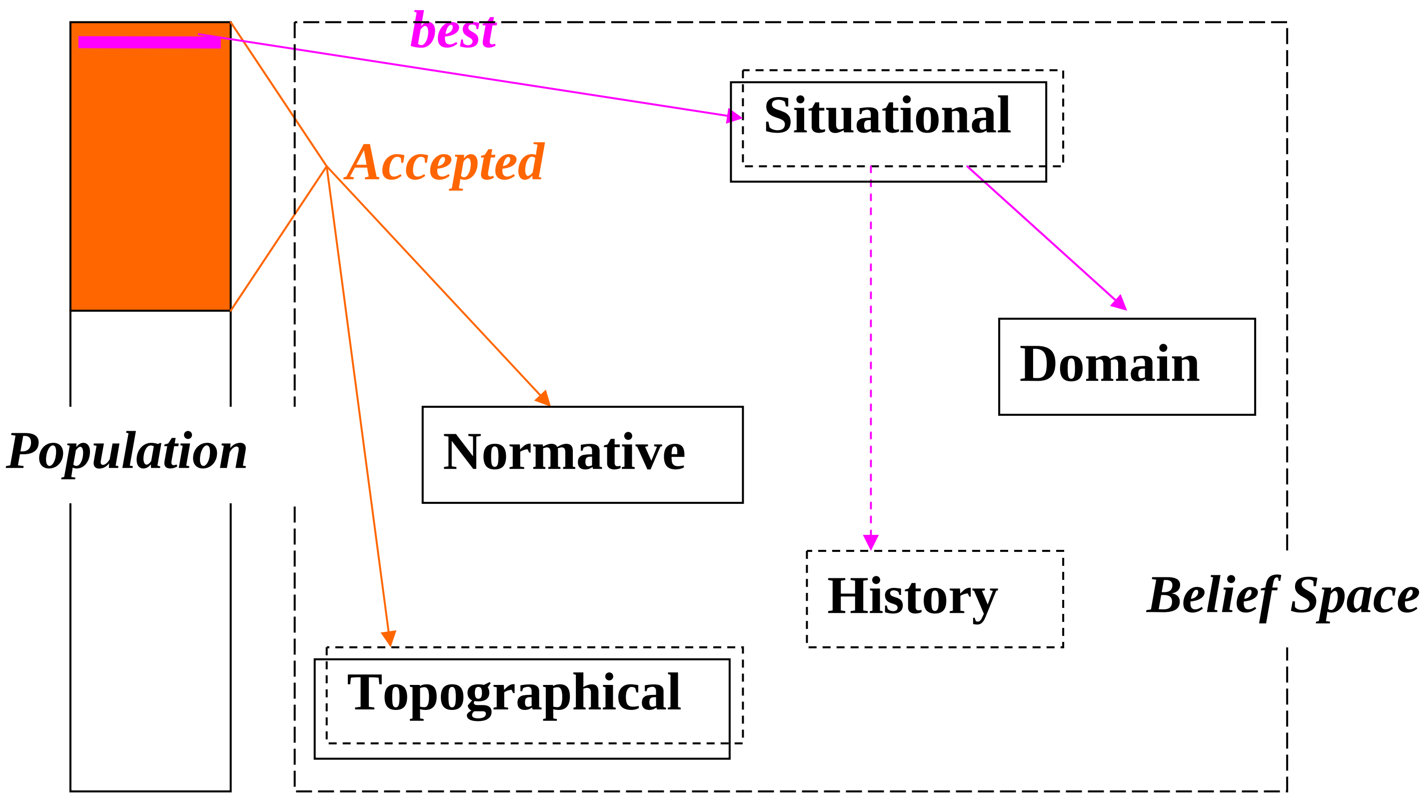
Influence Function

- Belief Space guides the changes made to individuals in the Population Space.
- Early Cultural Algorithms had only one knowledge source which was always applied to individuals.
- When additional sources were used in the Belief Space they were just randomly applied.
- Here we want to investigate the use of a systematic approach to the application of these multiple knowledge sources in the solution of optimization problems

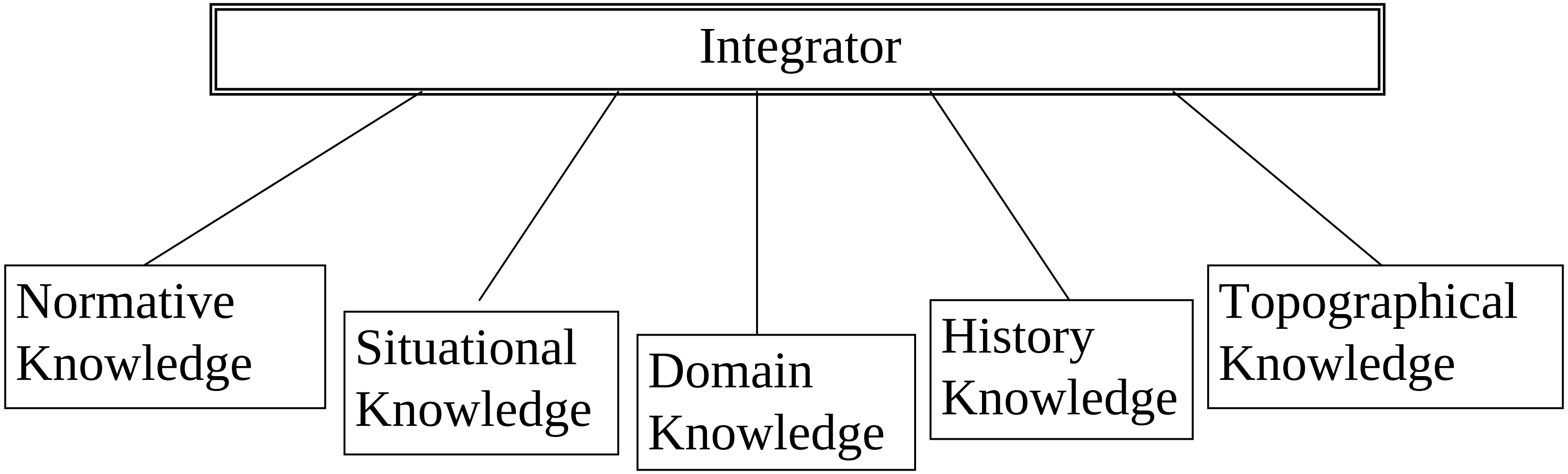
Integrating the Knowledge Sources in the Influence Function



Interaction Between Knowledge Sources



Hierarchical Integration of the Knowledge Sources



An Approach Based on a Biological Analogy

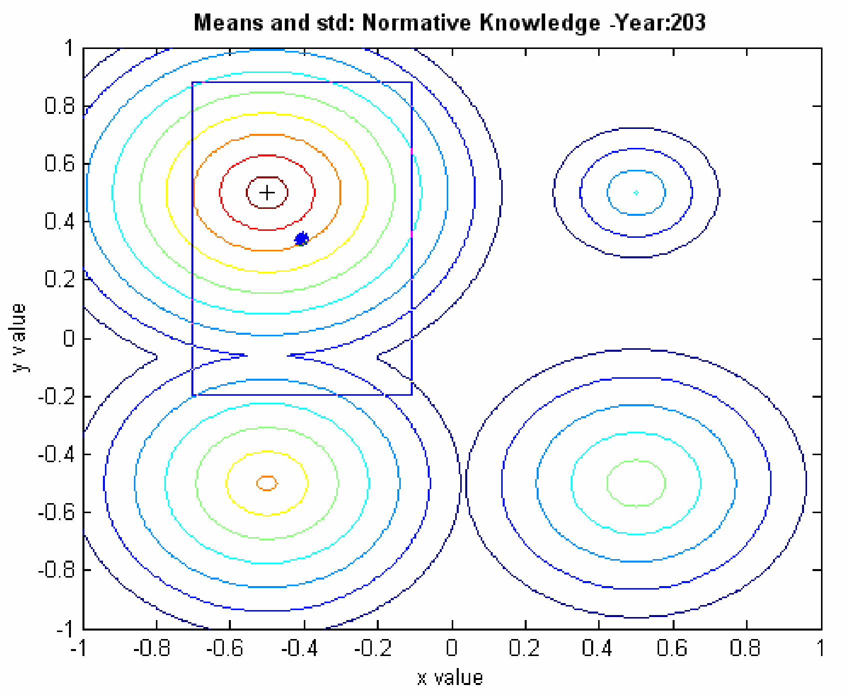
- View each knowledge source as a predator that exploits a patch of the performance function landscape.
- The KS directs individuals in the population to take parameter values that allows them to occupy a location in its current patch. The performance function achieved by the individual accrues to the knowledge source as its energy intake.

A Real-Valued optimization Environment

- For example, assume a two dimensional problem landscape.
- The performance function is continuous and its shape is shown by contour lines.
- The patch for a knowledge source is the sub-region of the landscape that it is the most likely to place individuals into.

Knowledge Source as a Solution Generator

- Generates new solutions stochastically based on its current state where its preferred area is described as a bounding box centered around the mean and bounded by the standard deviation (Wolpert, NCSA,2004).
- Shifts its patch when new individuals perform below the mean.

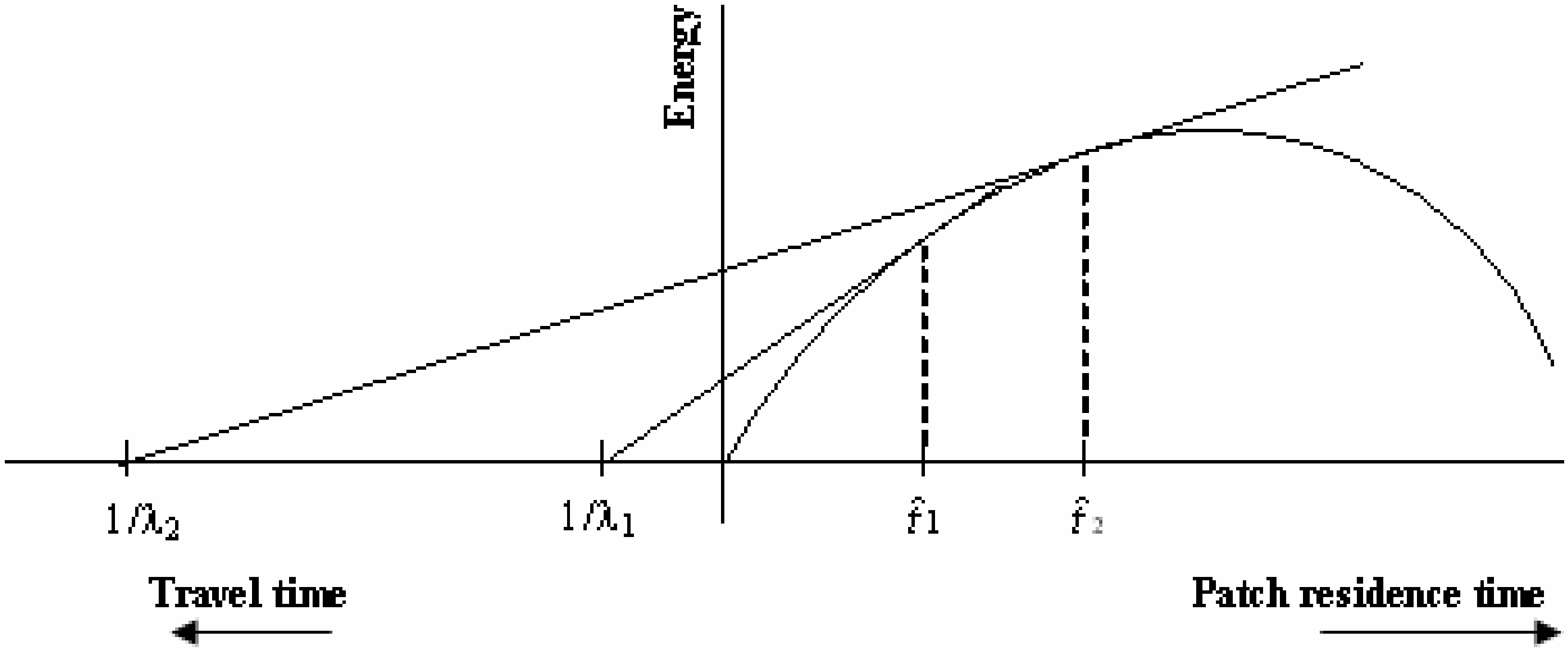


Means and std: Normative Knowledge -Year:203
y value
x value
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

The Marginal Value Approach (Charnov's Patch Model, 1976)

- How long a forager will stay in a patch
  - Residence time in a patch by a forager affects the expected energy gain
  - The goal is to maximize the long-term energy intake by staying in a patch
  - "When the intake in any patch drops to the average rate for the habitat, the animal should move to another patch".

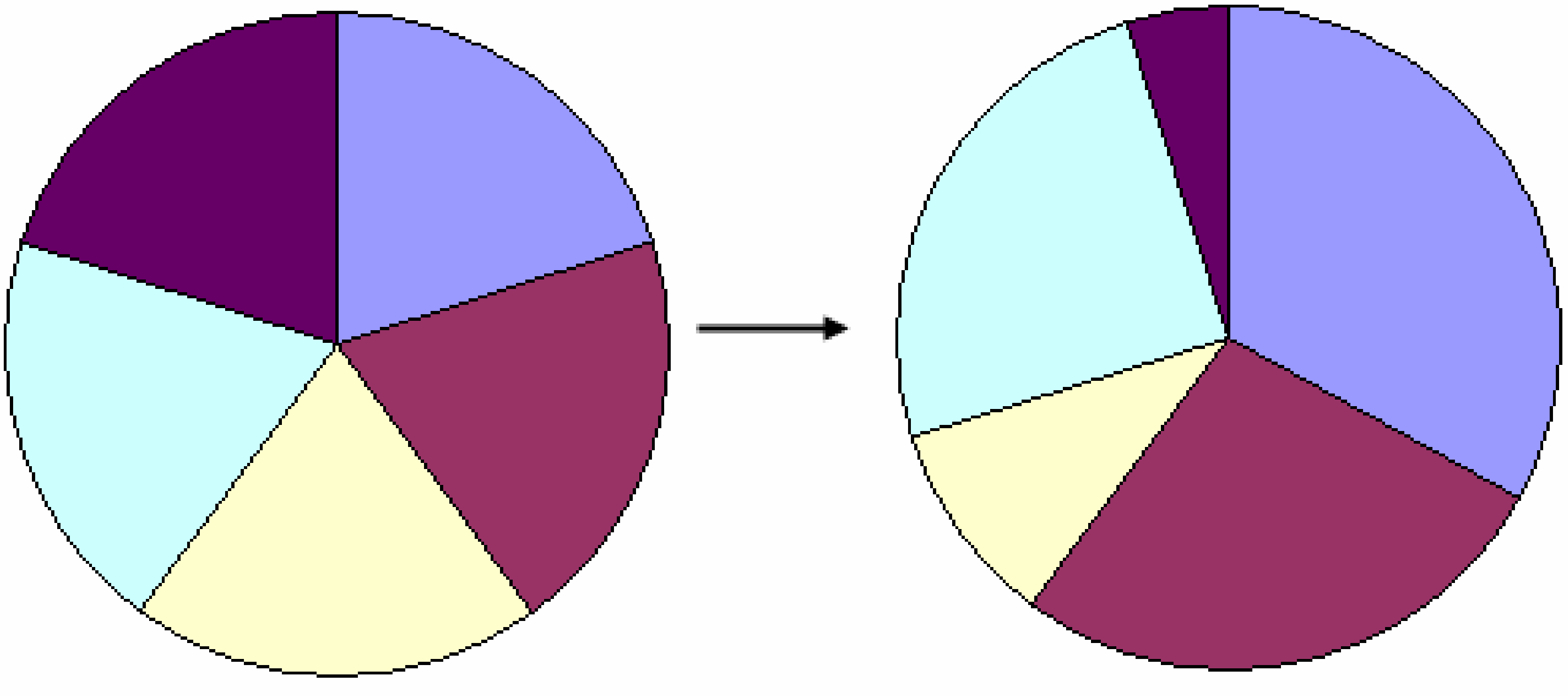
Assumes a Negatively Accelerated Gain Function



Implementation

- At each time step we select a knowledge source to “control” the positioning of an individual in the space.
- Similar to “spinning a wheel of fortune”. The proportion of the wheel belonging to a knowledge source reflects their current ability to allocate individuals to areas that have yields above expectations.
- As that ability decreases, their share of the wheel is reduces and their parameters shifted to define a new patch.

Roulette Wheel Function



Emergent Social Structures

- The goal is to investigate the structures that **emerge** at the knowledge, population, and individual levels as a result of using this marginal value approach.
- Since the knowledge sources used are not unique to humans, using the predator/prey approach to their integration may tell us something about the social structures that emerge from the use of knowledge in general.

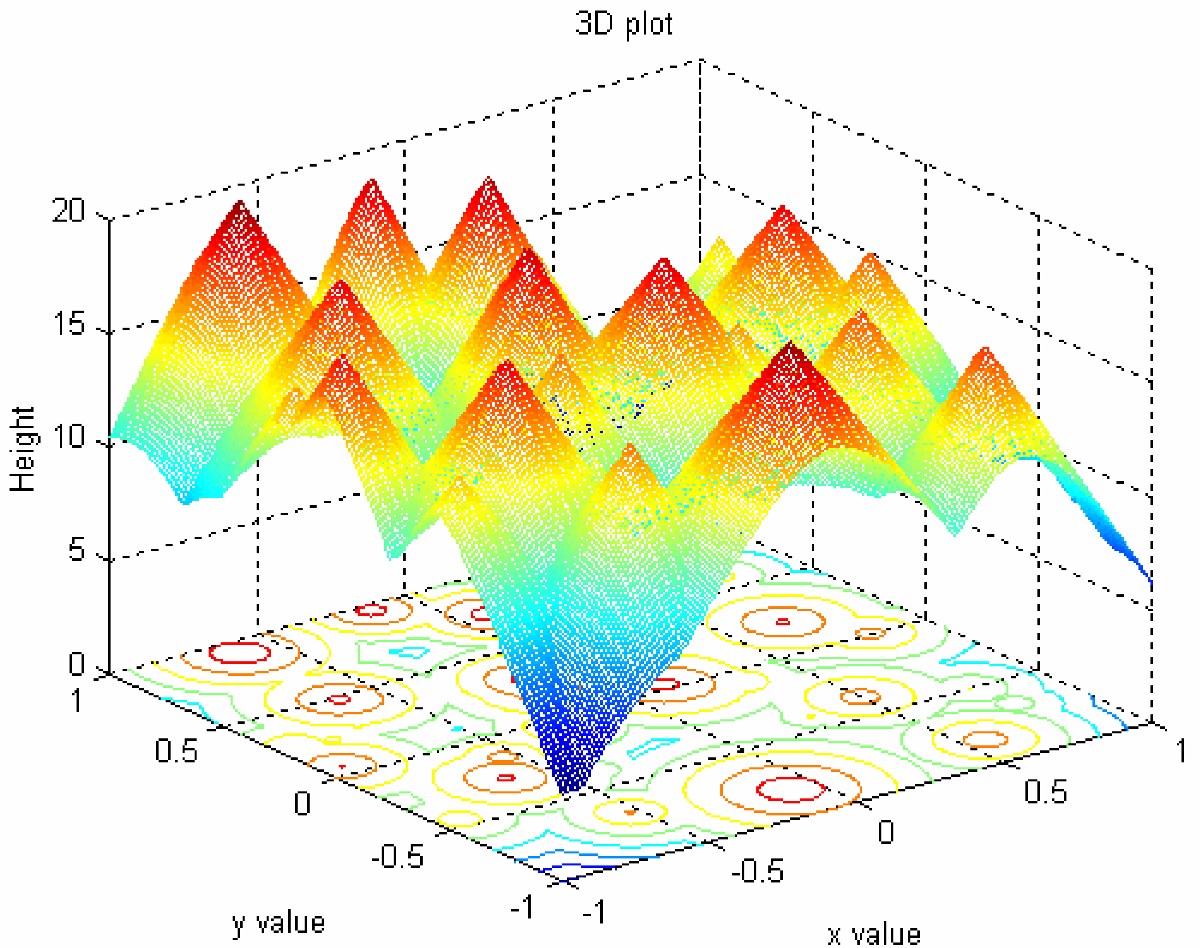
Problem Applications

- Apply the approach to problems that have been used with other swarm intelligence approaches.
- Cones world where resource cones are distributed in Sugarscape fashion (Axtell) and moved around (ACO, Dorigo, Bonabeau).
- Engineering Design Problem Used with Particle Swarm.

Problem Domain

- An extension of the Dynamic Problem Generator (Morrison and DeJong 1999)
- Distribute resource cones throughout a multi-dimensional landscape.
- Objective of the system is to locate highest peak
- Allows the static placement of cones as well as the relocation of one or more cones during run for dynamic environment

Sample Cones World Landscape



Cones World Parameters

- Number of cones (4 – 100)
- Cone height (5 – 20)
- Slope (20 – 30)
- Reset frequency (1 per 300 generations)
  - Allows chaotic environments

CAEP Java System

- Cultural Algorithm with Evolutionary Programming
- The system runs the search, plots the results in real-time, and saves data for later analysis as well
- On-line plots include Fitness-Time plot and generation-best plots (overall best and best from each knowledge type)

The Interface of CAEP

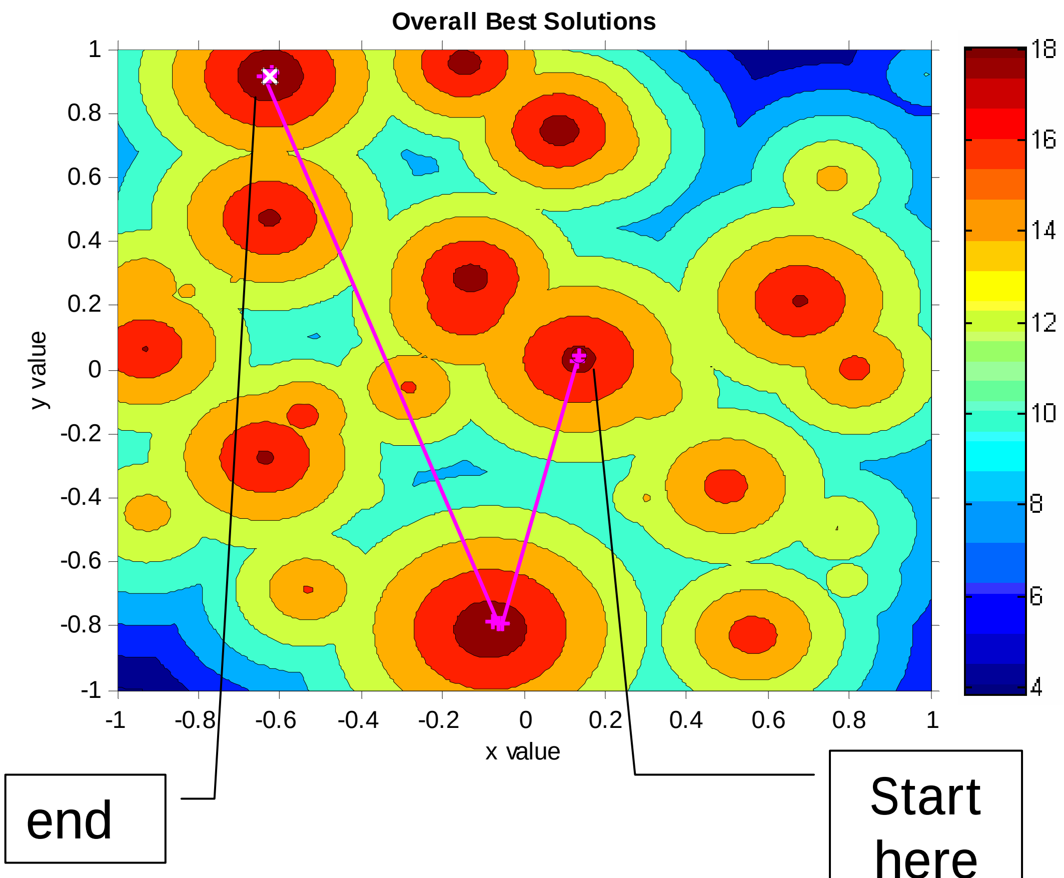
- Parameter & Command Area (1)
- Result Display Areas
  - Text-format (2): Real Optimum, Overall Best, Best from N, S, D, H, and T
  - Graph-format
    - Fitness Trace (3)
    - Location Traces (4): Overall Best, Bests from N, S, D, H, and T

Three types of views

- Three types of views
  - Bests traces
  - Population Swarm Plot
  - Meta-level Knowledge Plots
- In all viewers, a same color scheme is used to discriminate knowledge sources

Knowledge Types	**Color**
Normative	blue
Situational	black
Domain	green
History	yellow
Topographical	cyan

A Sample Run

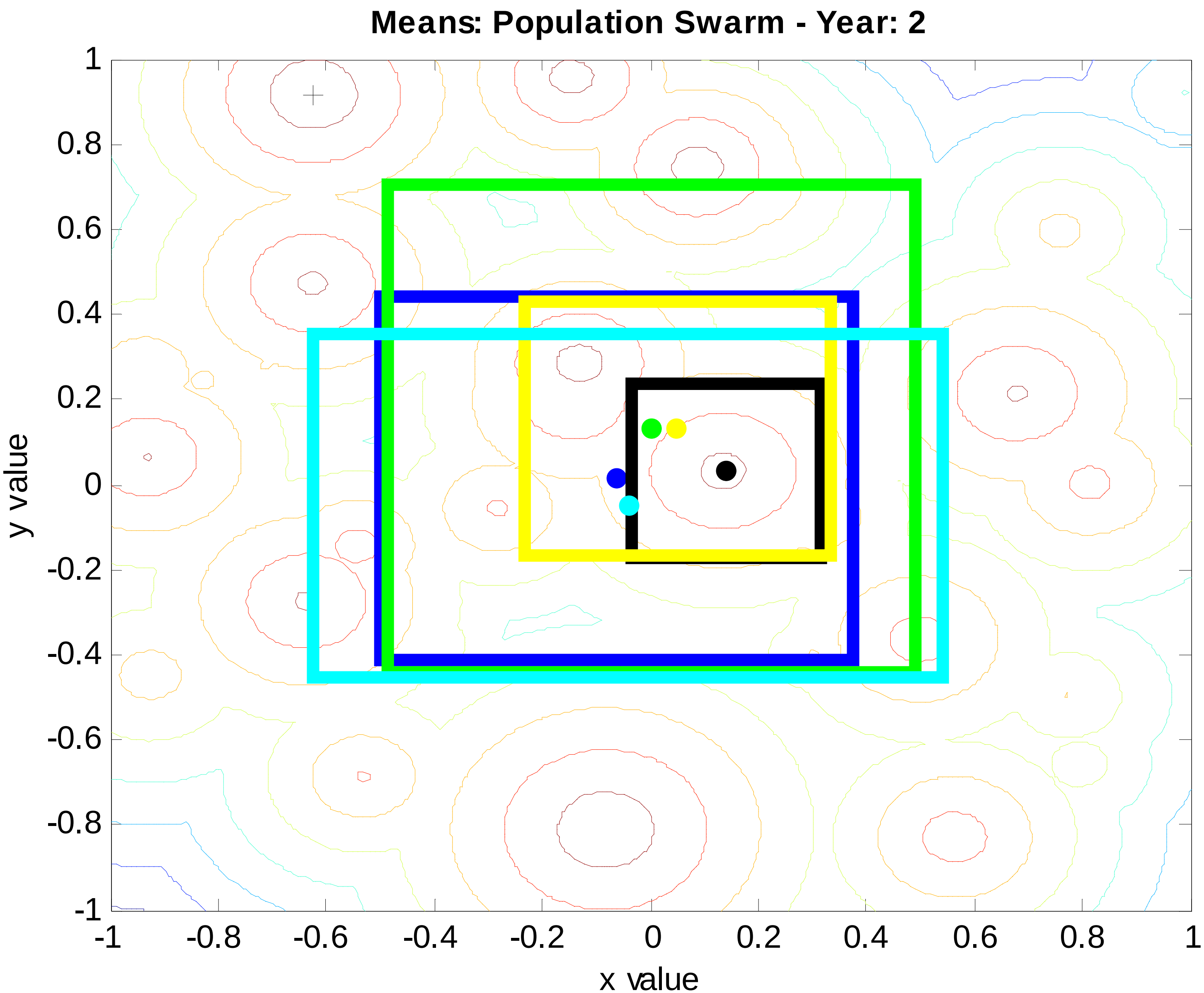


- Fitness contour
- This example run goes through two local maxima before reaching the real optimum

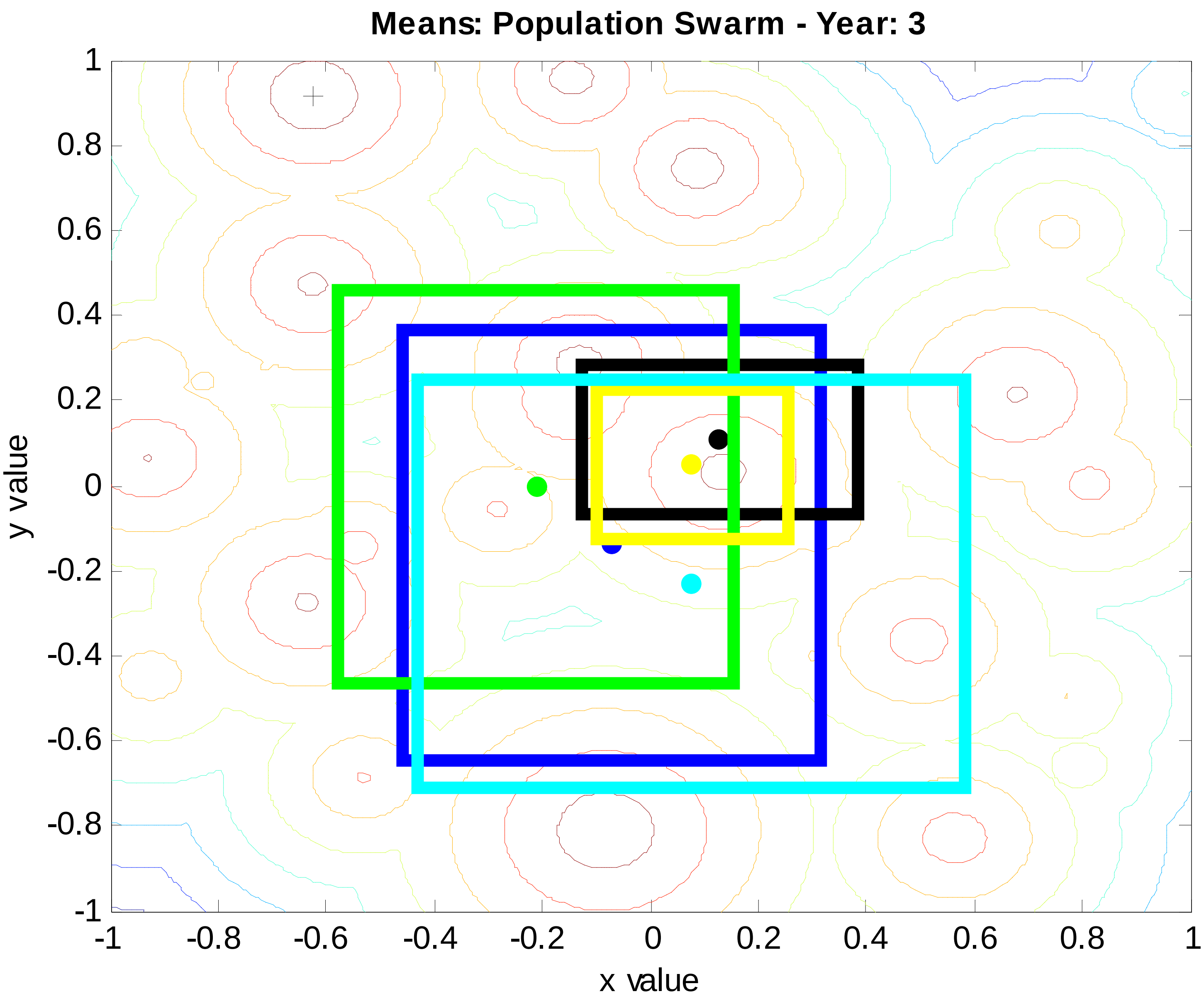
Emergence of “Knowledge Swarms”

- We observe “knowledge swarms” appearing in the Belief Space.
- Coordinated distribution of patches.
- Certain KS serve to “explore” the space, and the results can attract other knowledge sources that “exploit” it.
- Induces two basic phases of problem solving, coarse grained, and fine grained.

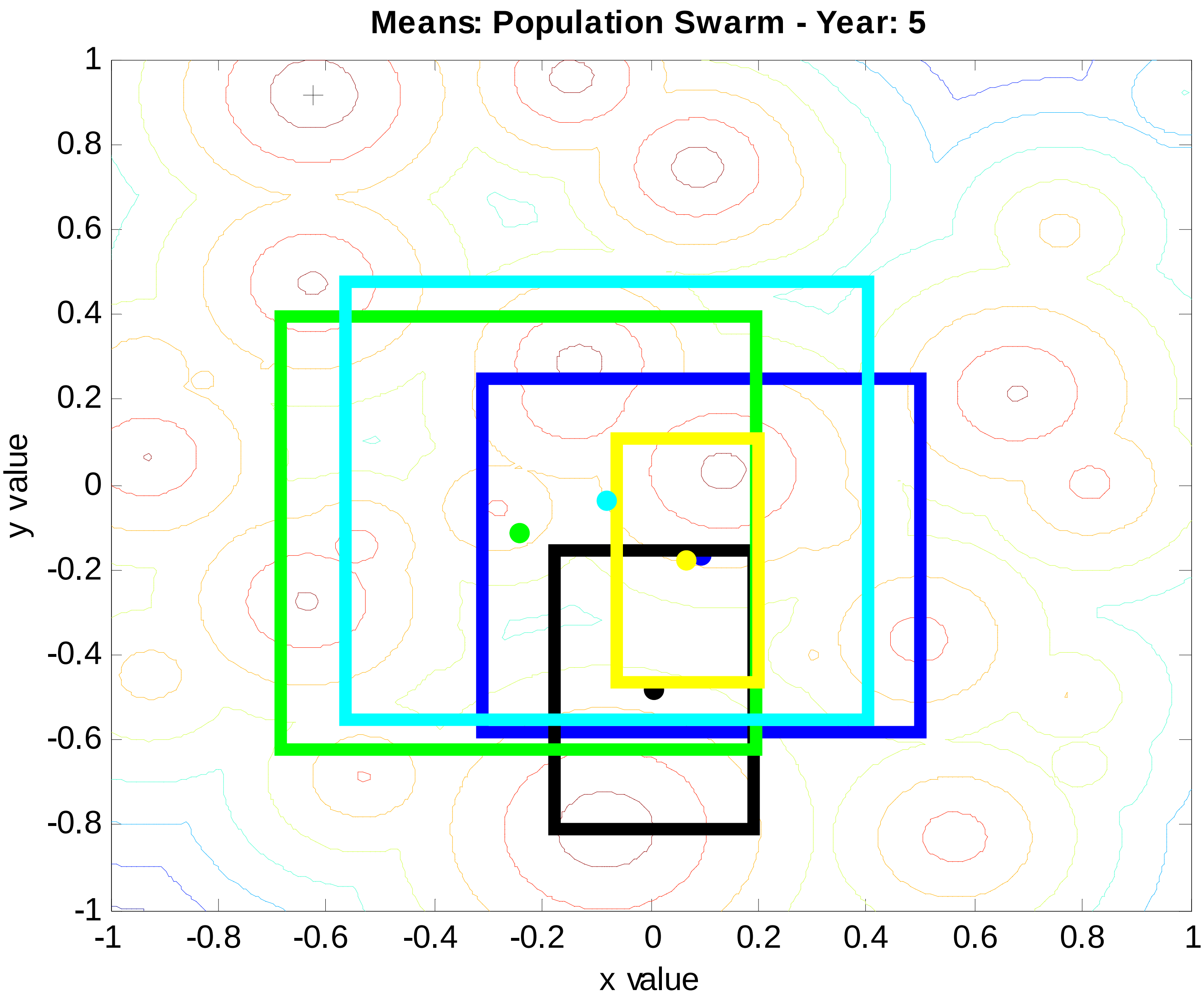
Meta-level, Generation 2, Find First Peak



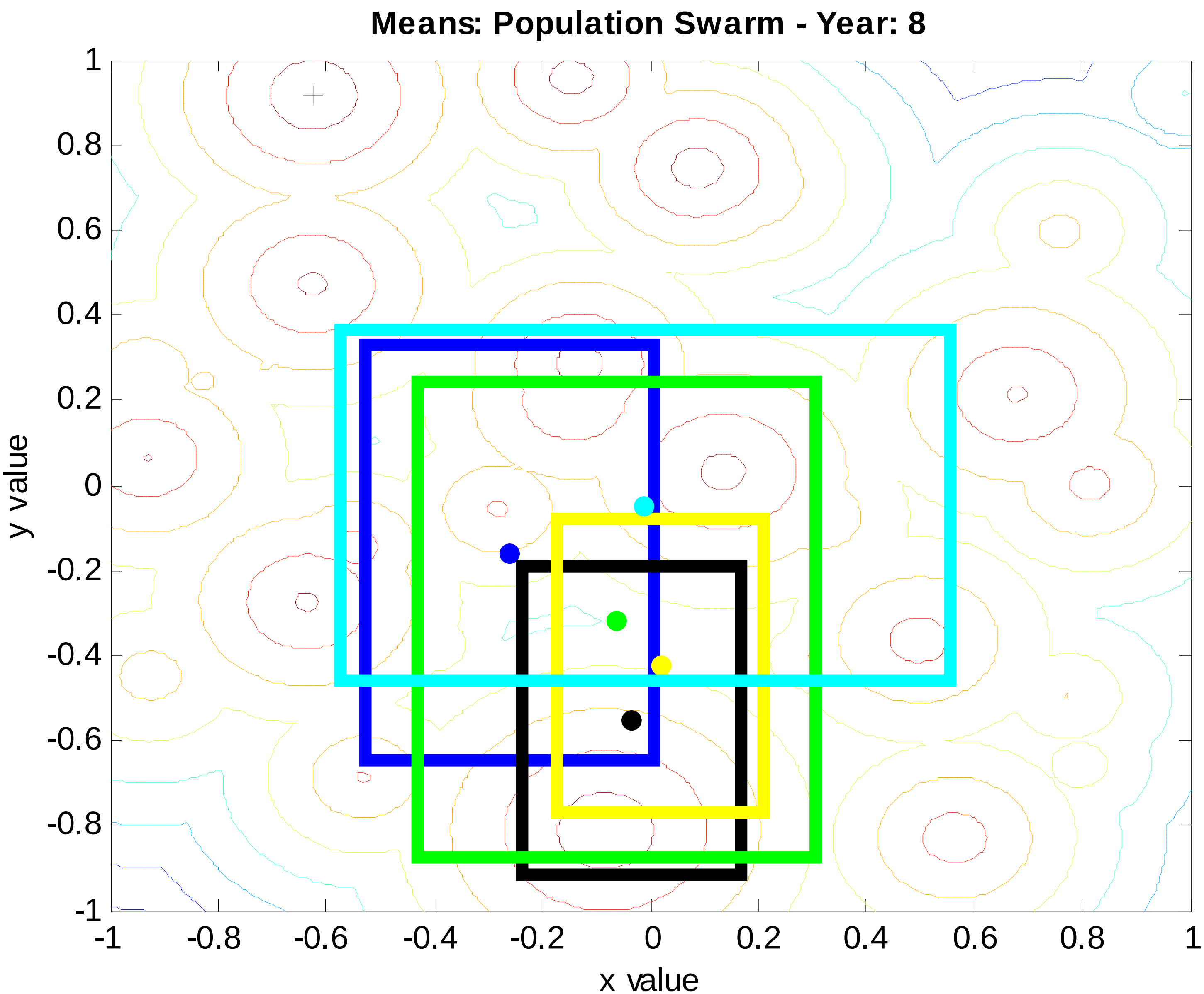
Meta-level, Generation 3, Second Peak Spotted



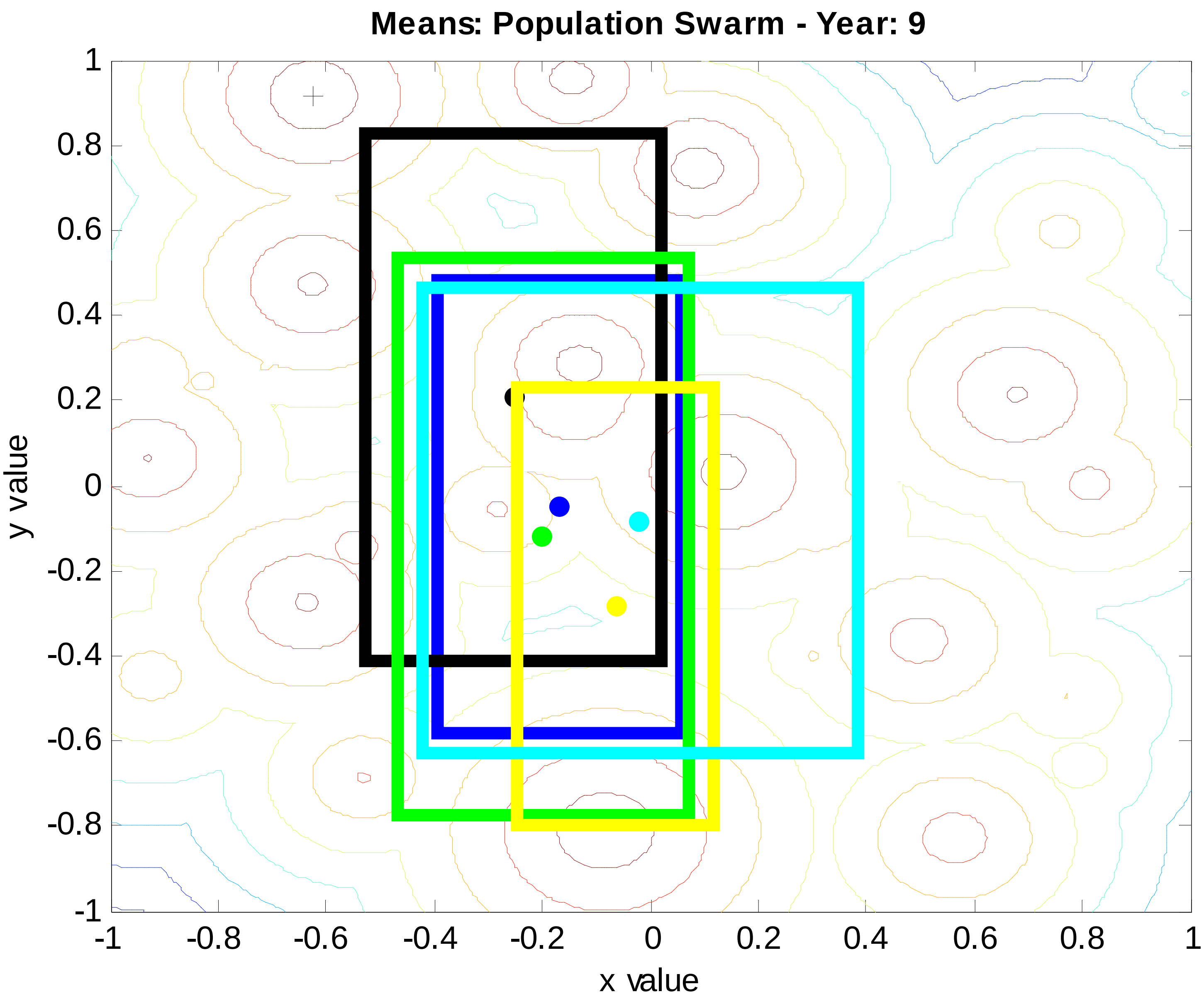
Meta-level, Generation 5, First Migration



Generation 8, In the Middle of Moving, Found a Third Peak



Meta-level, Generation 9, Move Again



Generation 20, Global Maximum Found, Scouts Still Active

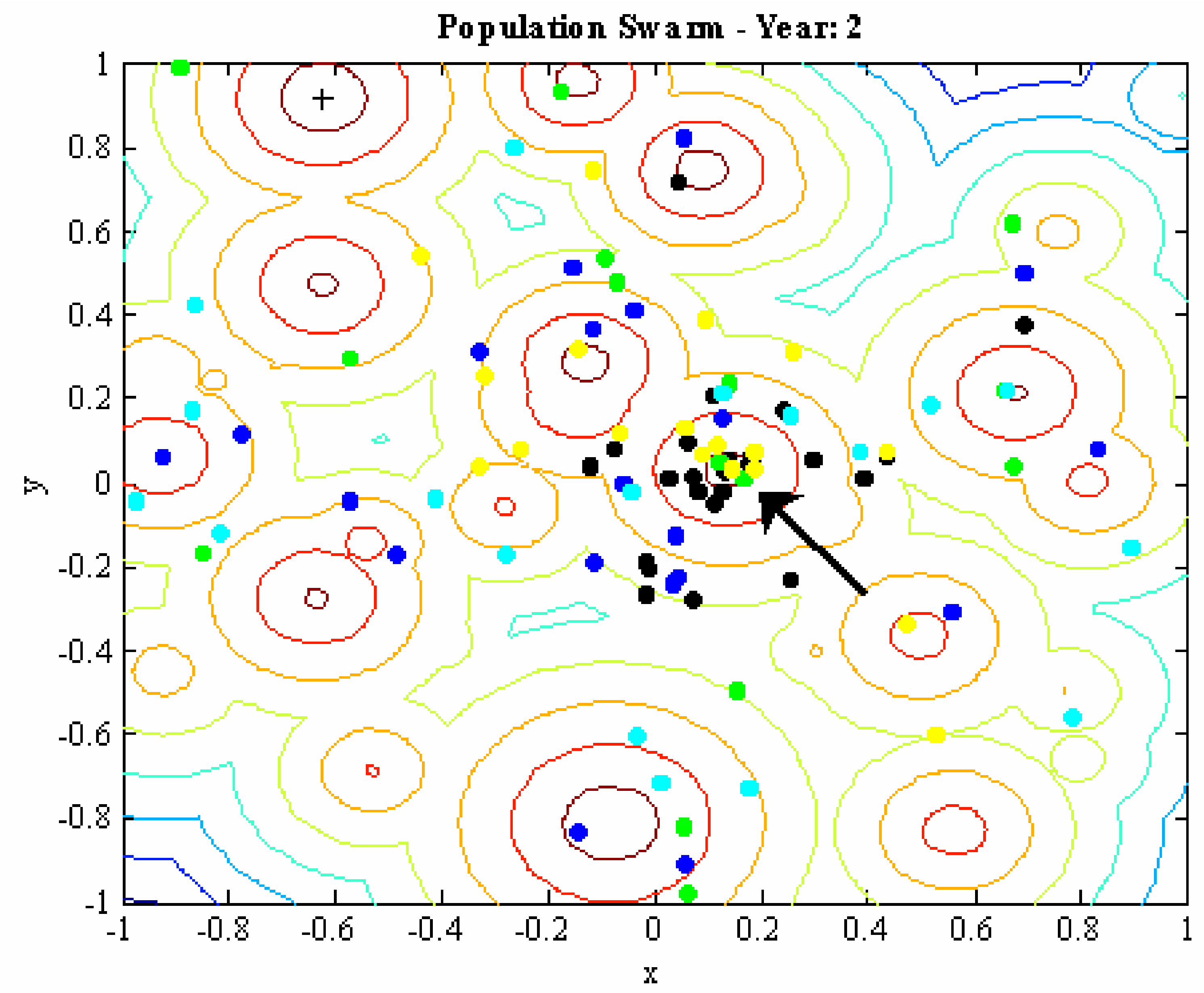
Knowledge Swarms

- The success of a generalist or exploratory knowledge sources in a region attracted specialist knowledge sources with smaller more concentrated patches
- By moving into the patch the specialists reduced the relative performance of the generalists who then moved their patches to other less exploited areas
- Coordination of exploration and exploitation in terms of knowledge source control

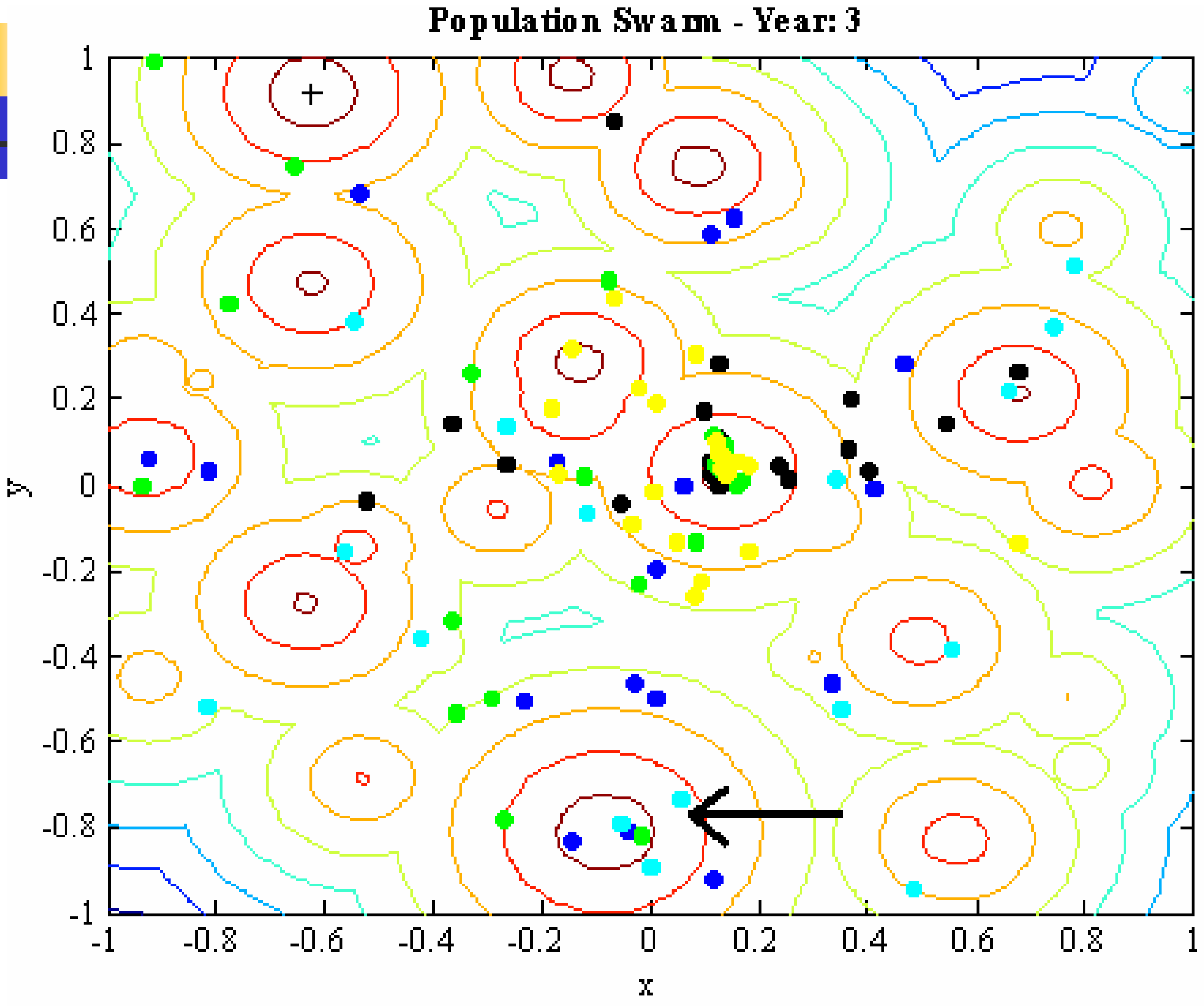
Emergence of Population Swarms

- “Population swarms” emerge under the control of the different knowledge sources.
- Each individual is displayed as a colored dot and the color tells which knowledge source is in control.
- Certain individuals under the control of normative and topographic knowledge lead the swarms with individuals influenced by situational, domain, and history knowledge following.

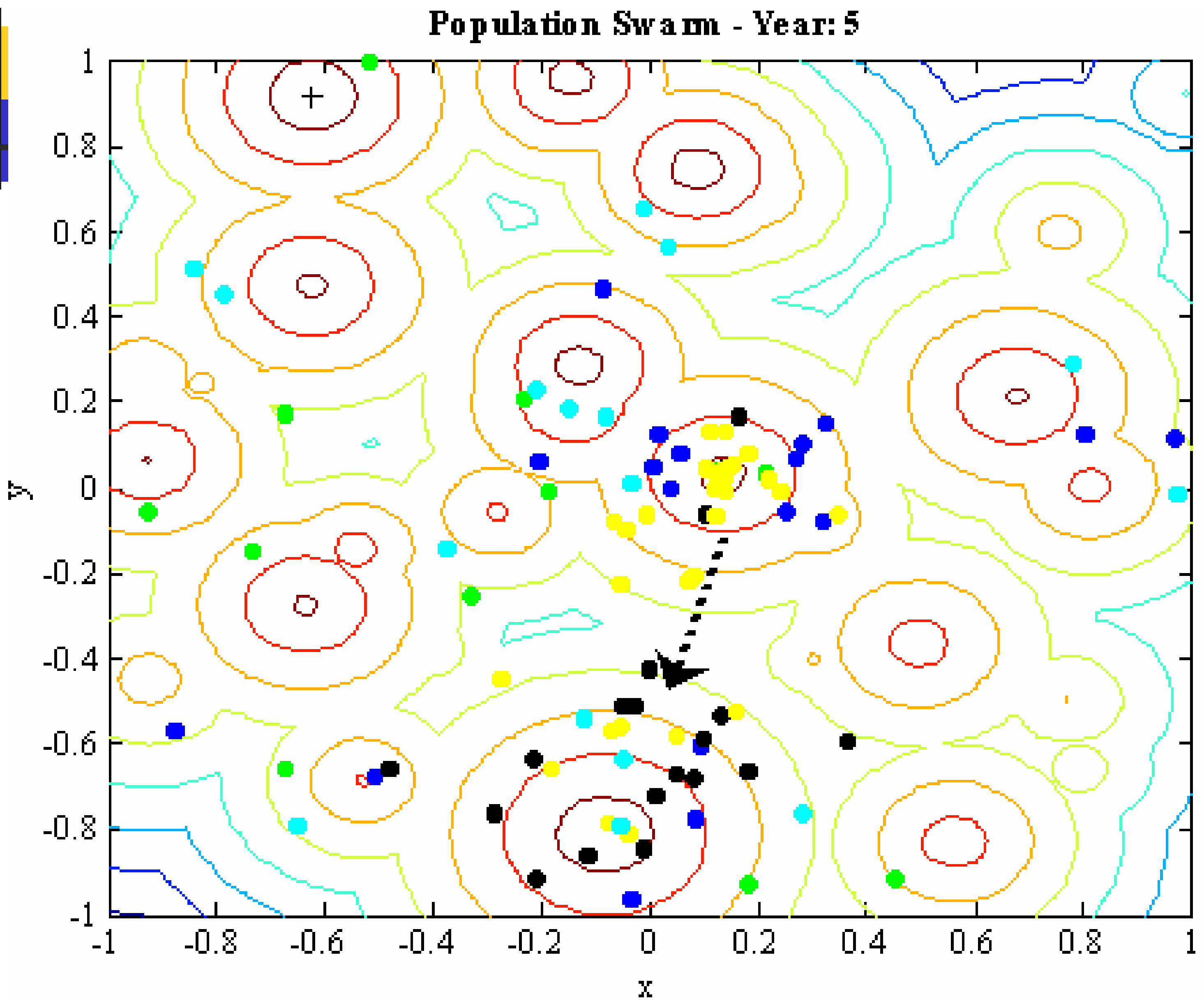
Population Plot, Generation 2, Find First Peak



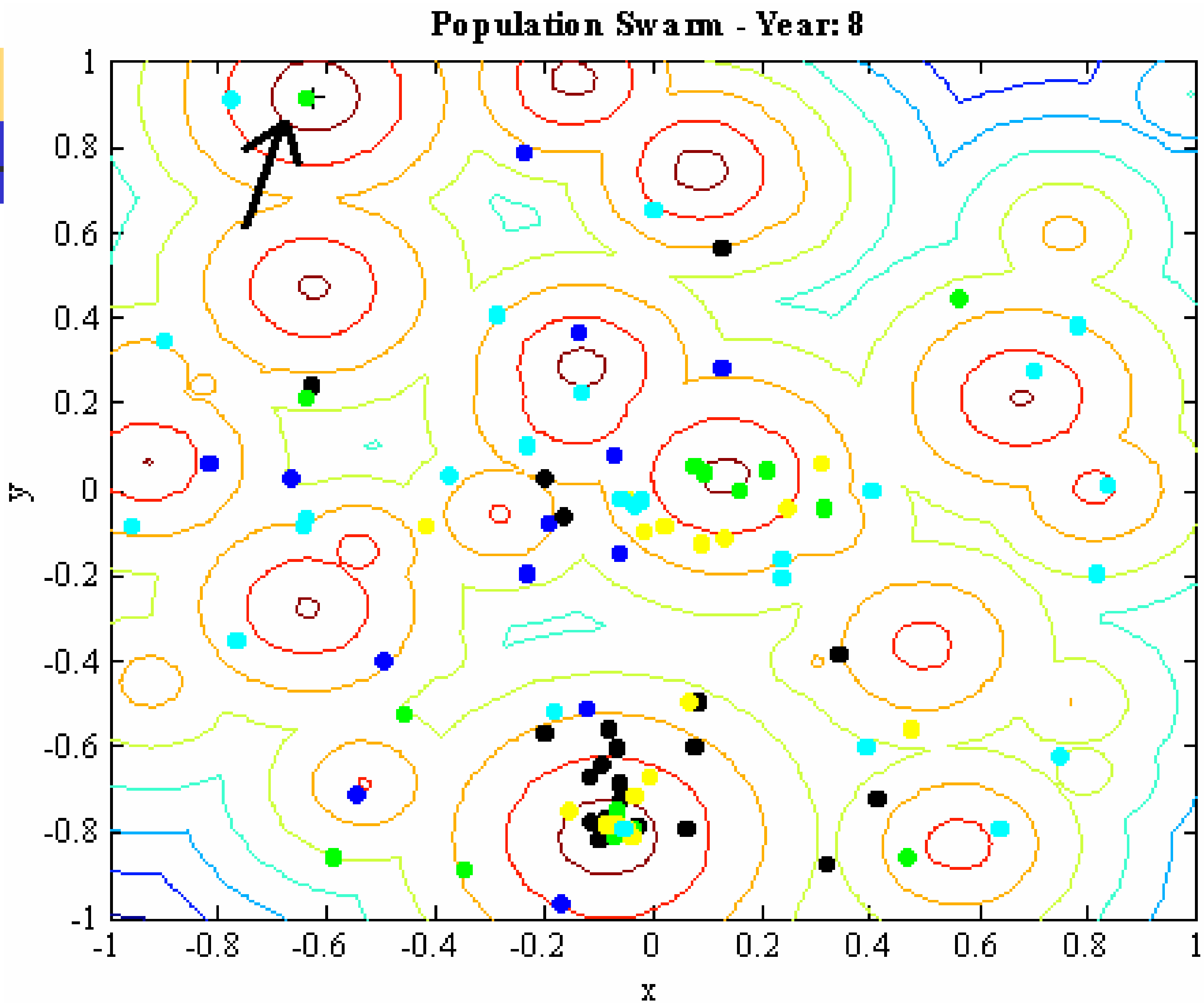
Generation 3, Second Peak Spotted



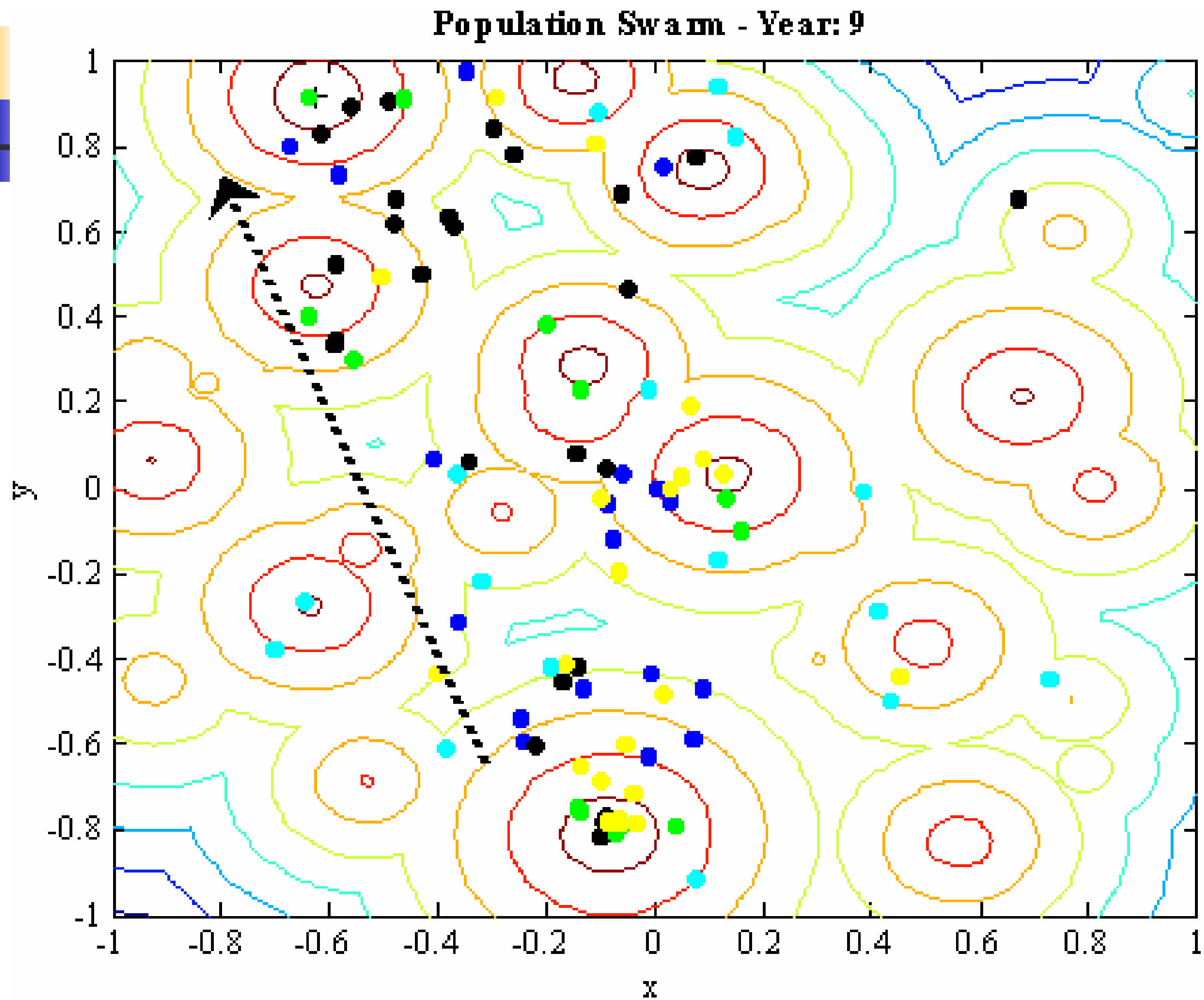
Generation 5, First Migration



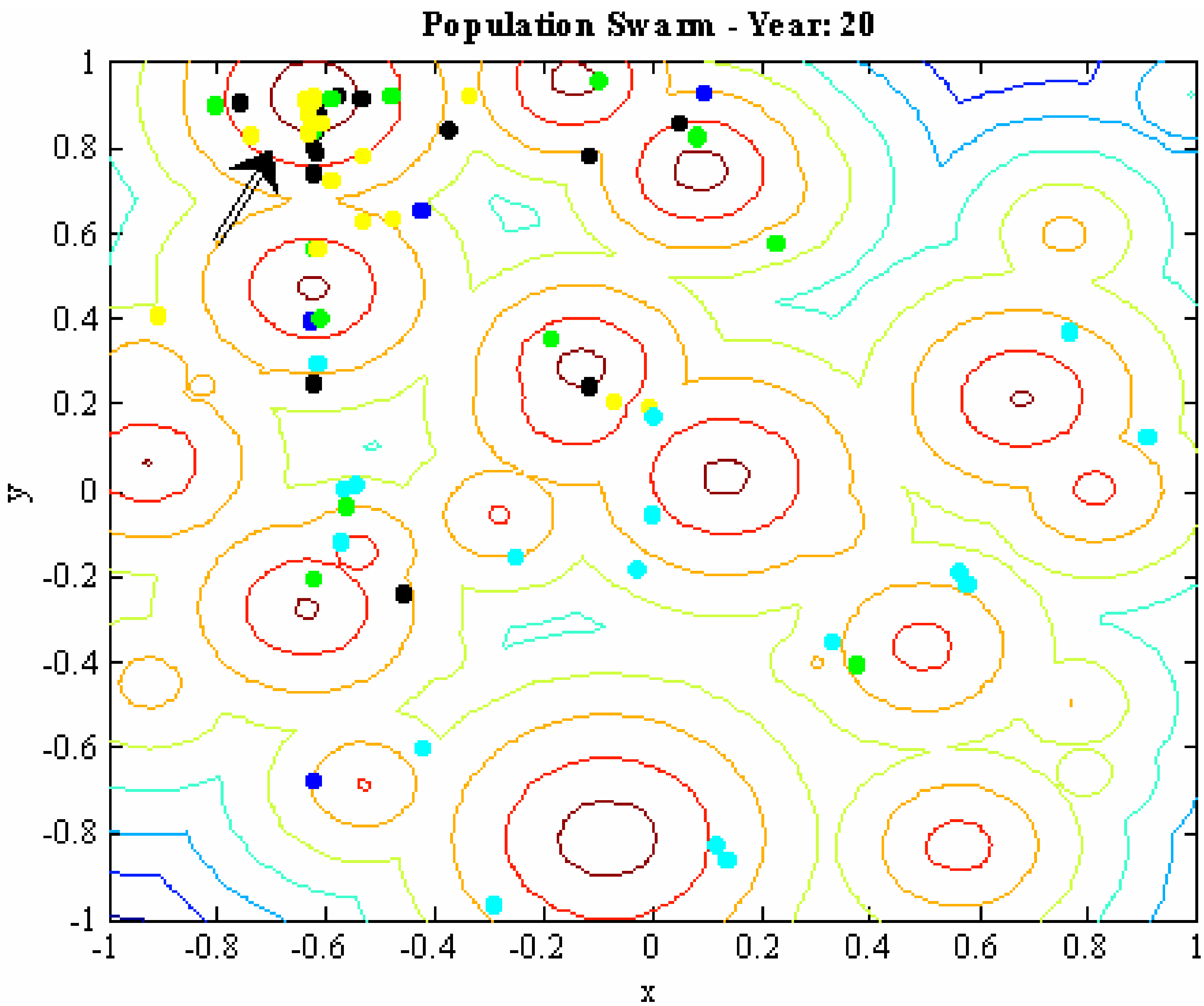
Generation 8, In the Middle of Moving, Found a Third Peak



Generation 9, Move Again



Generation 20, Global Maximum Found, Scouts Still Active



Emergence of Roles

- Here a “role” relates to an individual repeatedly using the same knowledge source even though the likelihood is probabilistic.
- Notice that the number of “exploiters” is more subject to change over time based on the occurrence of new discoveries.
- If a species is able to make adjustments that serve to “fit “ individuals into commonly occurring roles they will have a competitive advantage.

Role Emergence

- **Certain knowledge sources are better at taking "big steps" and used by <u>leaders</u> of the group, other taken "smaller steps" and used by <u>followers</u>, while others "record" the steps for subsequent use.**
- **This results in the emergence of problem solving phases. These phases are called "<u>coarse grained</u>" or "<u>fine grained</u>" depending on the relative amount of exploration and exploitation in each.**
- **Same knowledge swarms and related social roles emerge in different problems during a successful problem solving endeavor.**
- **Engineering Design Problem**

Engineering Problem Description

- Minimize the weight of a tension spring
- Subject to constraints on
  - minimum deflection,
  - shear stress,
  - surge frequency,
  - limits on outside diameter and on design variables

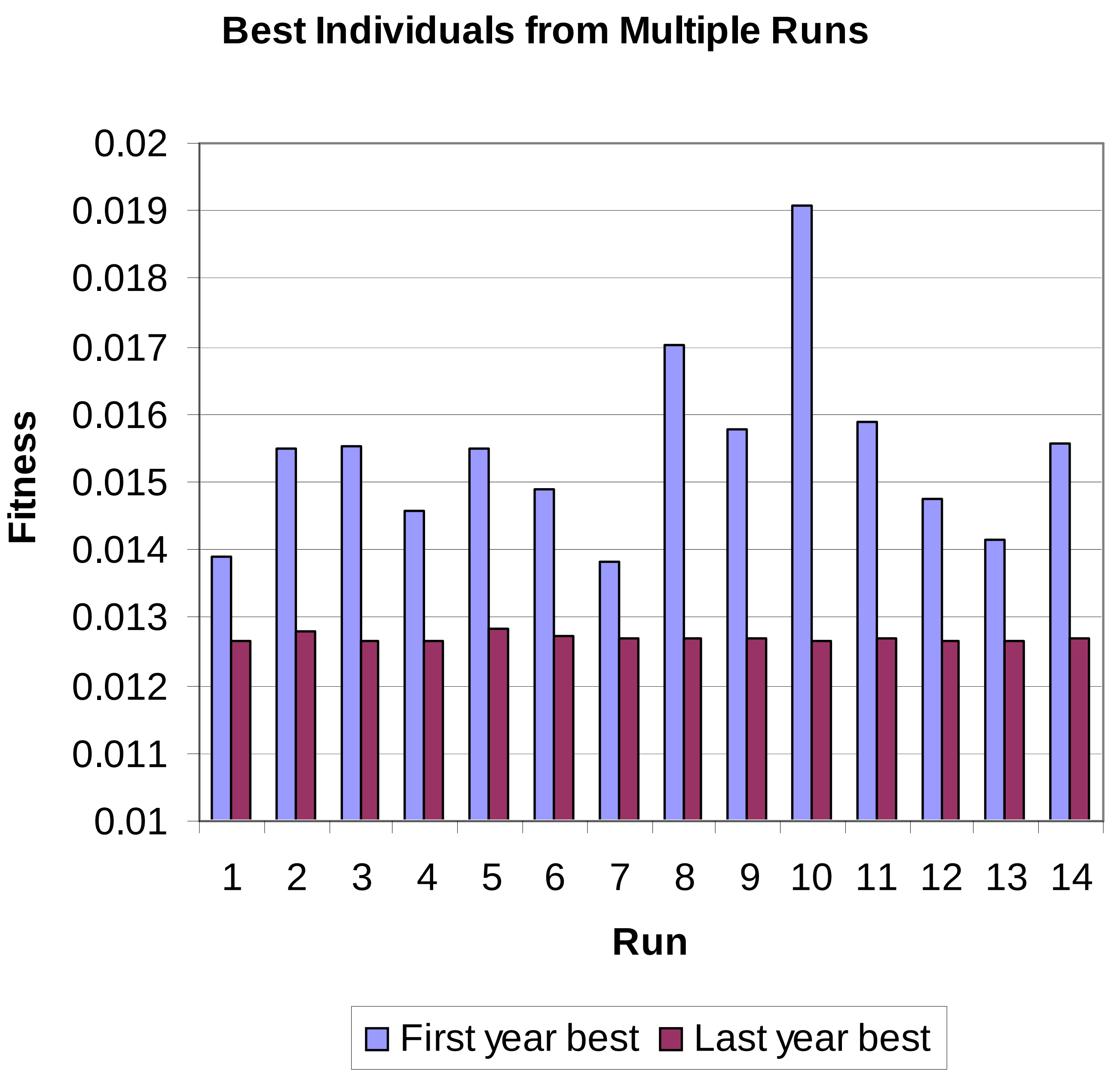
Problems Description – Cont.

- Design variables
  - Mean coil diameter D (0.25 <= D <= 1.3)
  - Wire diameter d (0.05 <= d <= 2.0)
  - Number of active coils N (2.0 <= N <= 15.0)

Constraint Handling

- Penalty Function

```
if(individual satisfies all constrains)
  fitness = obj(individual)
else
  fitness = IN-FEASIBLE-VALUE
endif
```


Best Individuals from Multiple Runs
0.02
0.019
0.018
0.017
0.016
0.015
0.014
0.013
0.012
0.011
0.01
Fitness
1 2 3 4 5 6 7 8 9 10 11 12 13 14
Run
First year best
Last year best

The Results from 14 Repeated Runs

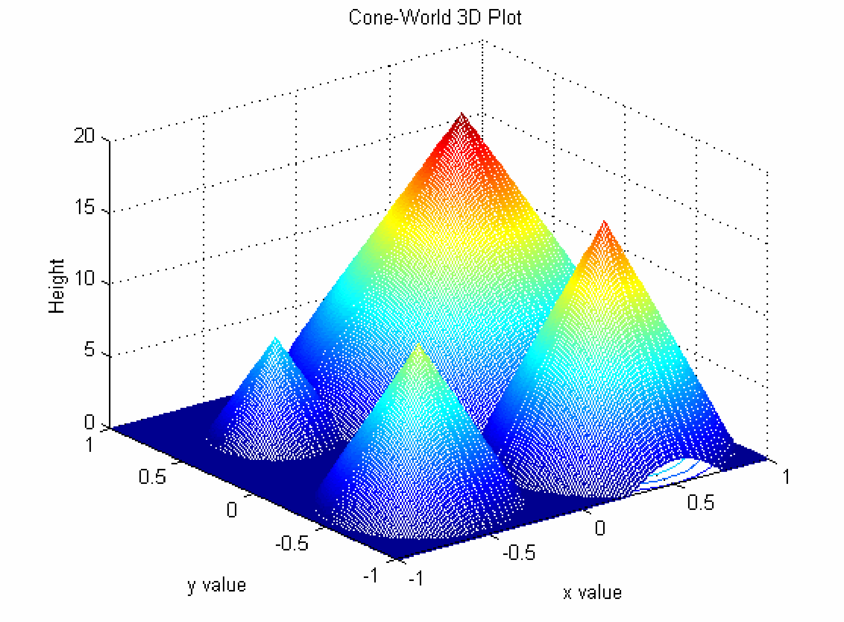
Run #	Best f(X) in First Year	Best f(X) in Last Year
1	0.013914342	0.012665483
2	0.015499381	0.01278739
3	0.015515272	0.012668523
4	0.01458971	0.012667248
5	0.015504649	0.012829209
6	0.014890658	0.012716912
7	0.013812628	0.012681938
8	0.017036807	0.01268631
9	0.015783767	0.012697047
10	0.019077229	0.012666253
11	0.015890667	0.012684916
12	0.014747043	0.012677166
13	0.014138278	0.01266972
14	0.015556953	0.012686394
Mean		**0.012698894**
Standard Deviation		**4.91214E-05**

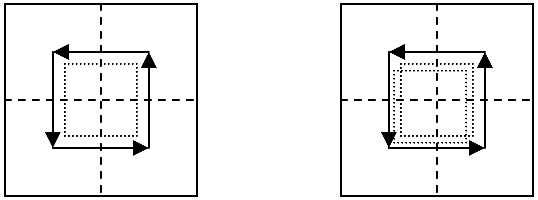
Engineering Problem - Result Comparison

	In this Work	Particle Swarm Paper
Best f(X)	0.012665483	0.0126661409
Mean	0.012698894	0.012718975
Standard Deviation	4.91214E-05	6.446 E-05

Structural Emergence in Dynamic Scenarios

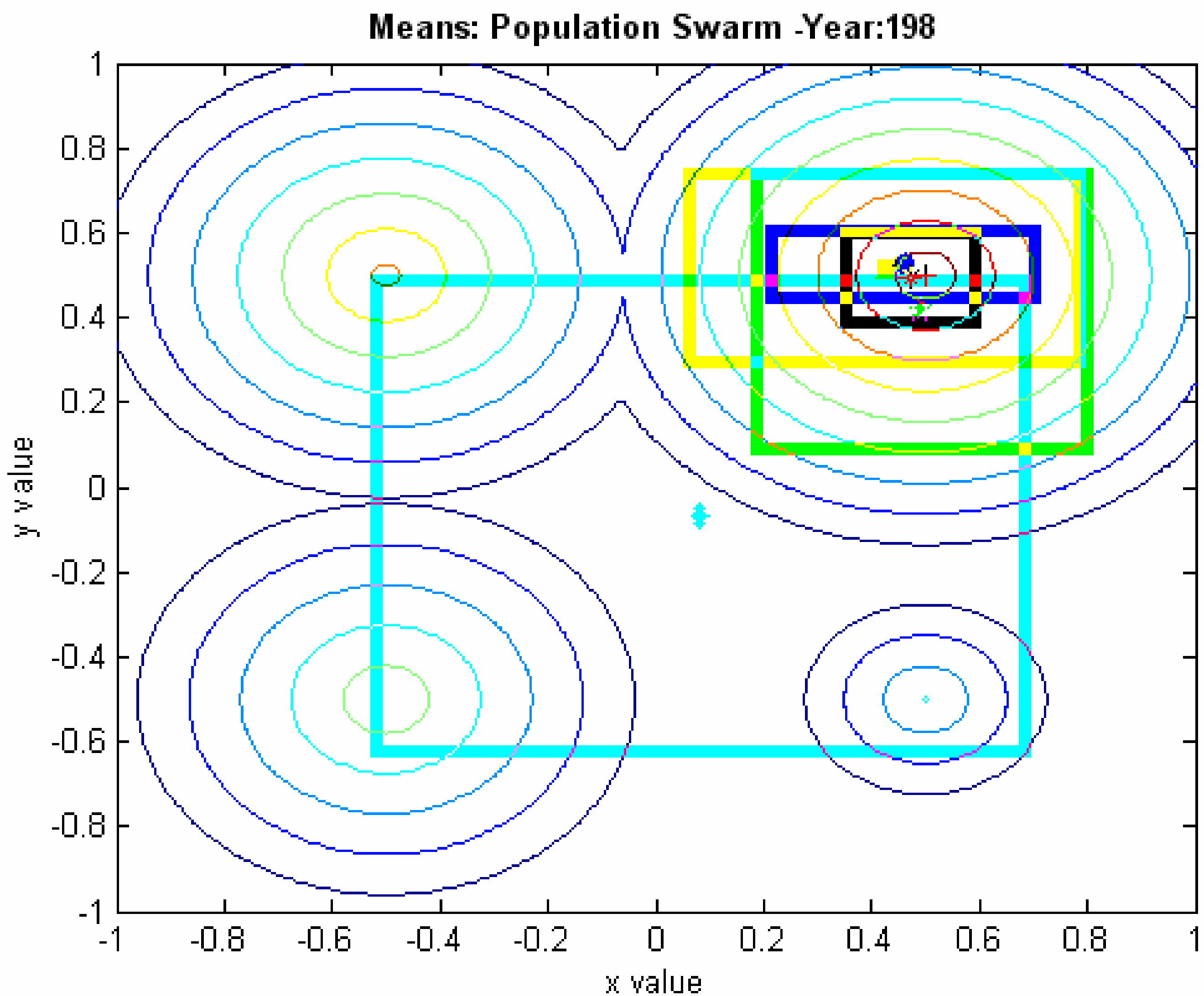
Dynamic Scenarios

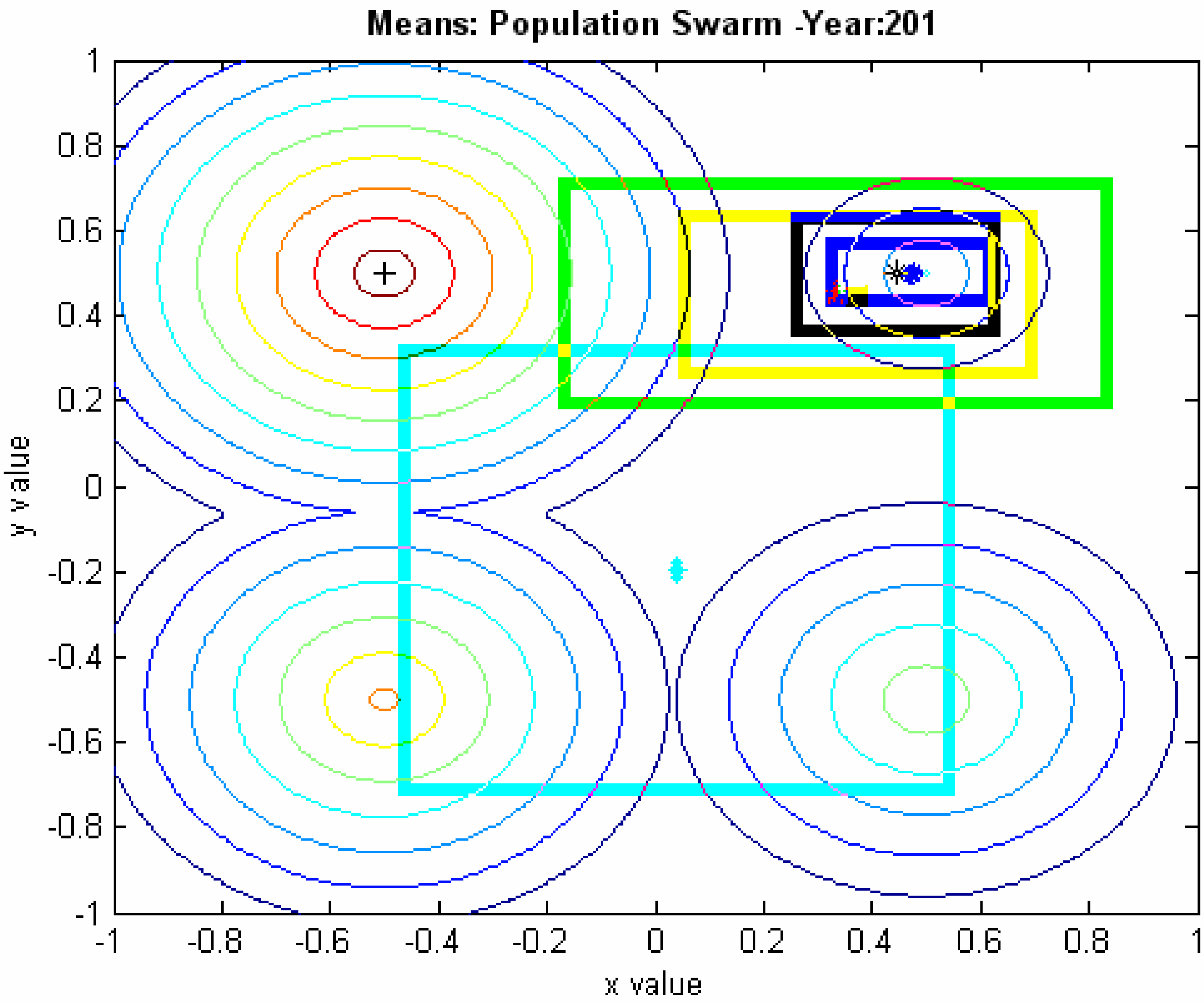


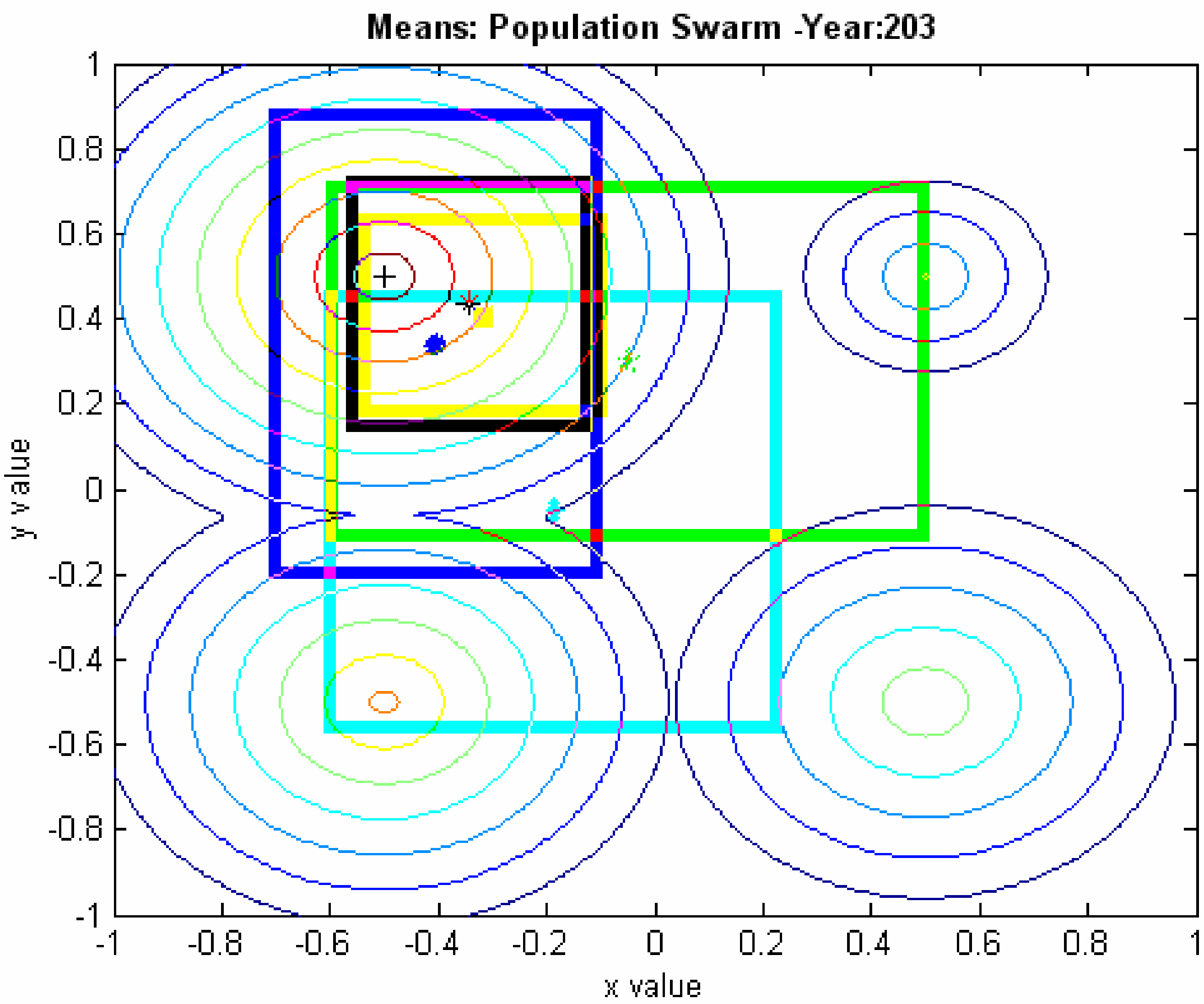


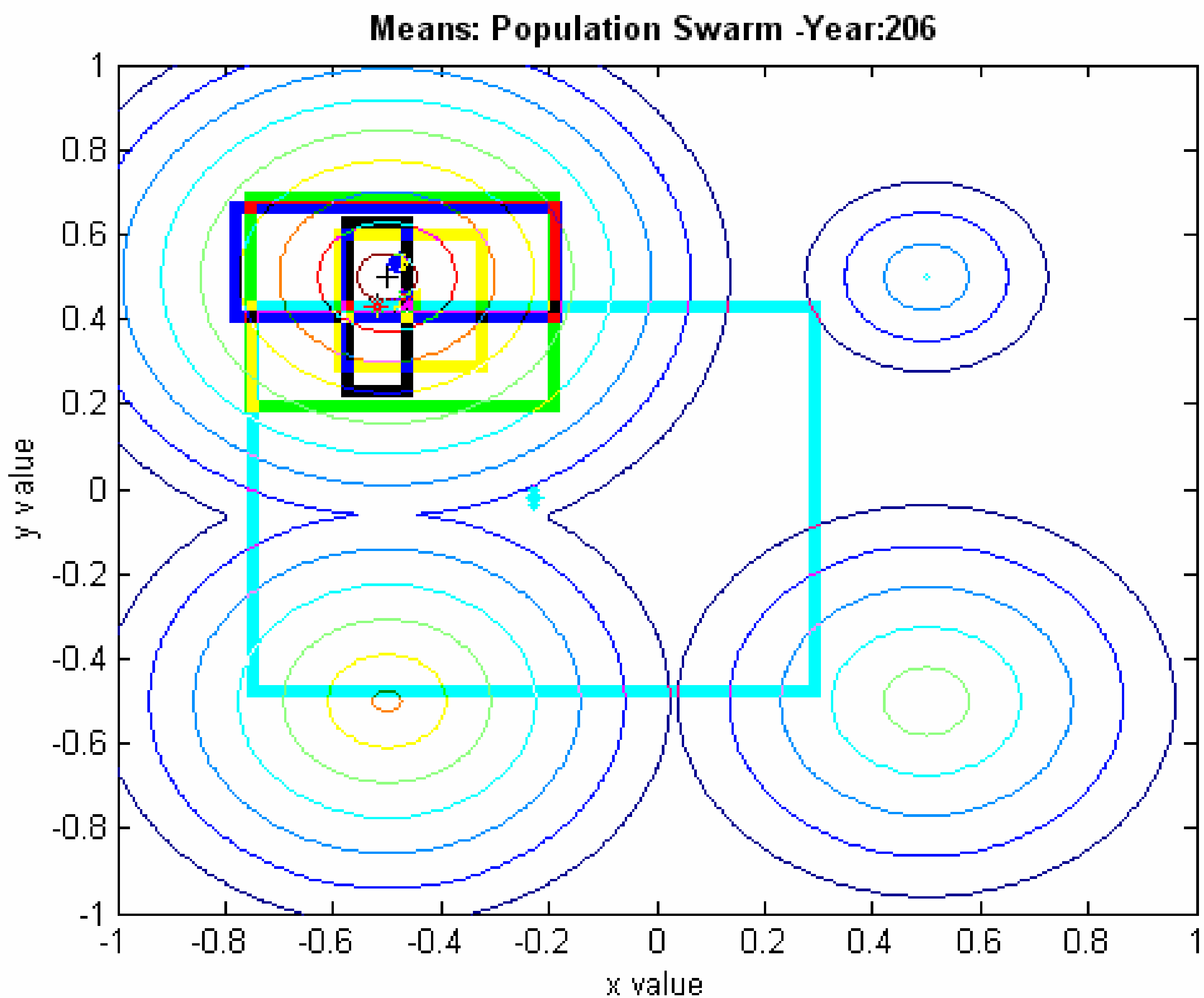
A Dynamic Scenario

- A baseline landscape with four cones randomly distributed, one into each of the four quadrants of the Cartesian plane
- The four cones are shifted counterclock wise periodically (every 200 iterations)
- A total of 16 shifts or 4 times around the track.


Means: Population Swarm -Year:198
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y value
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x value


Means: Population Swarm -Year:201
y value
x value
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

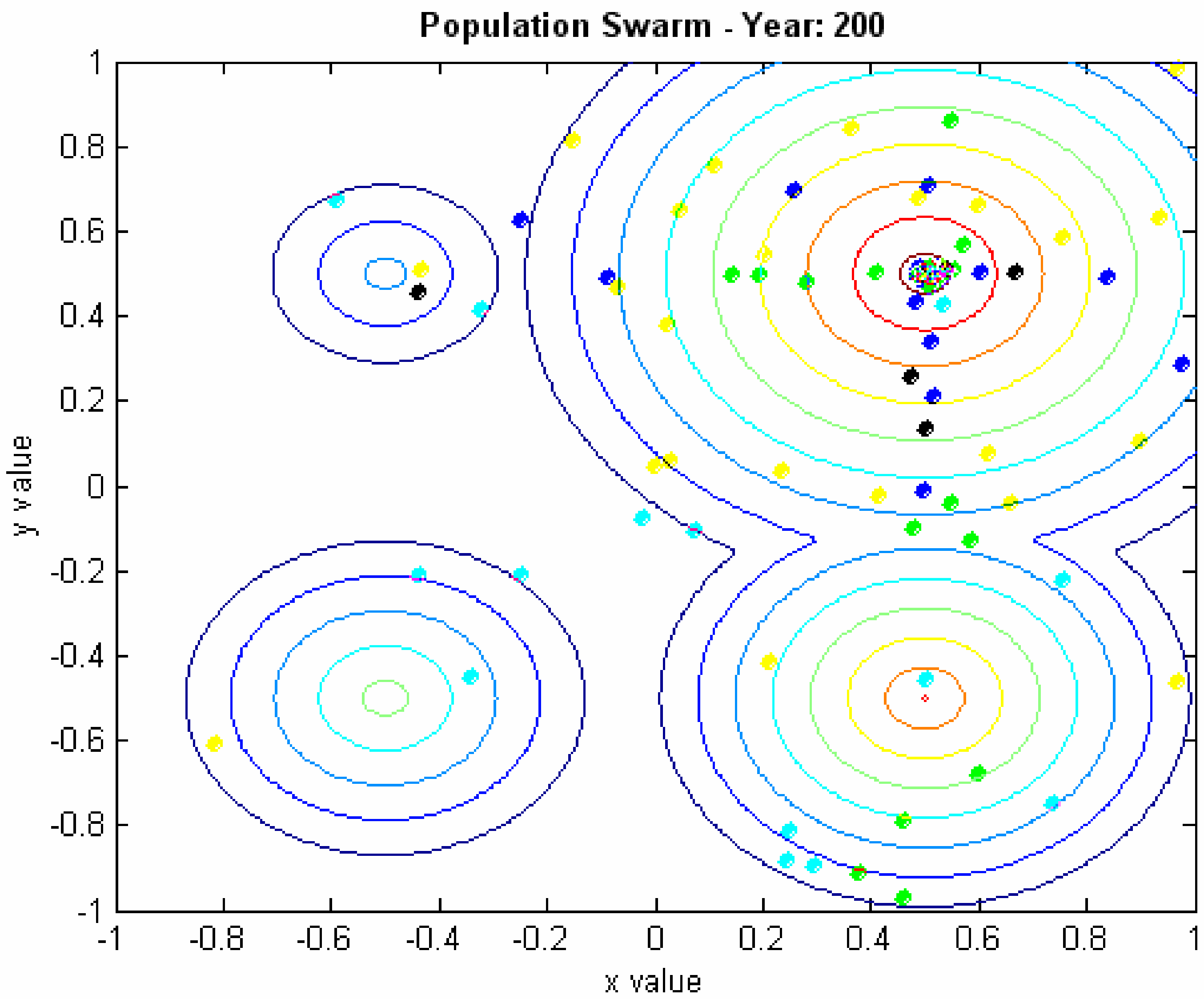

Means: Population Swarm -Year:203
y value
x value
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



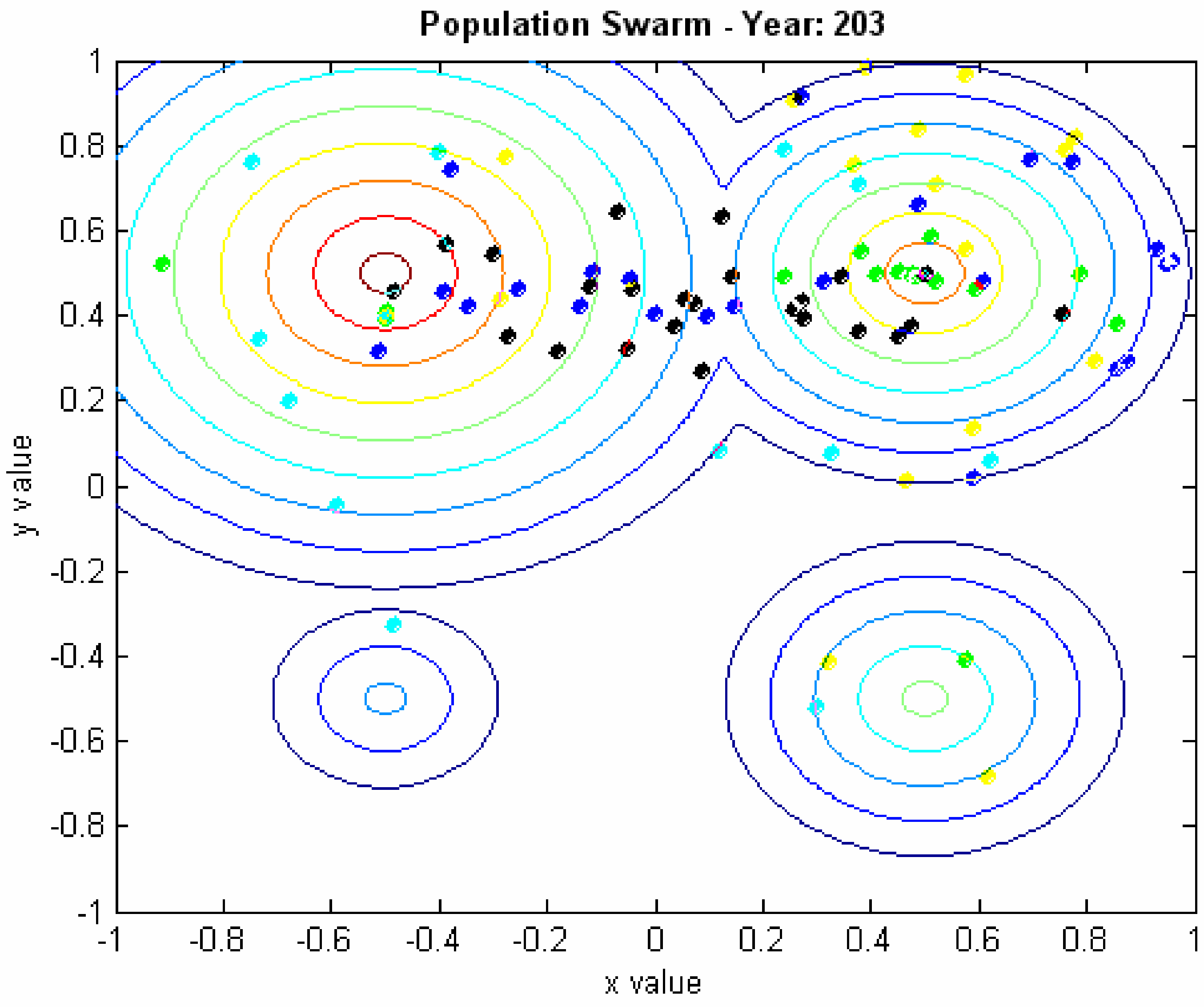
Means: Population Swarm -Year:206
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y value
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x value

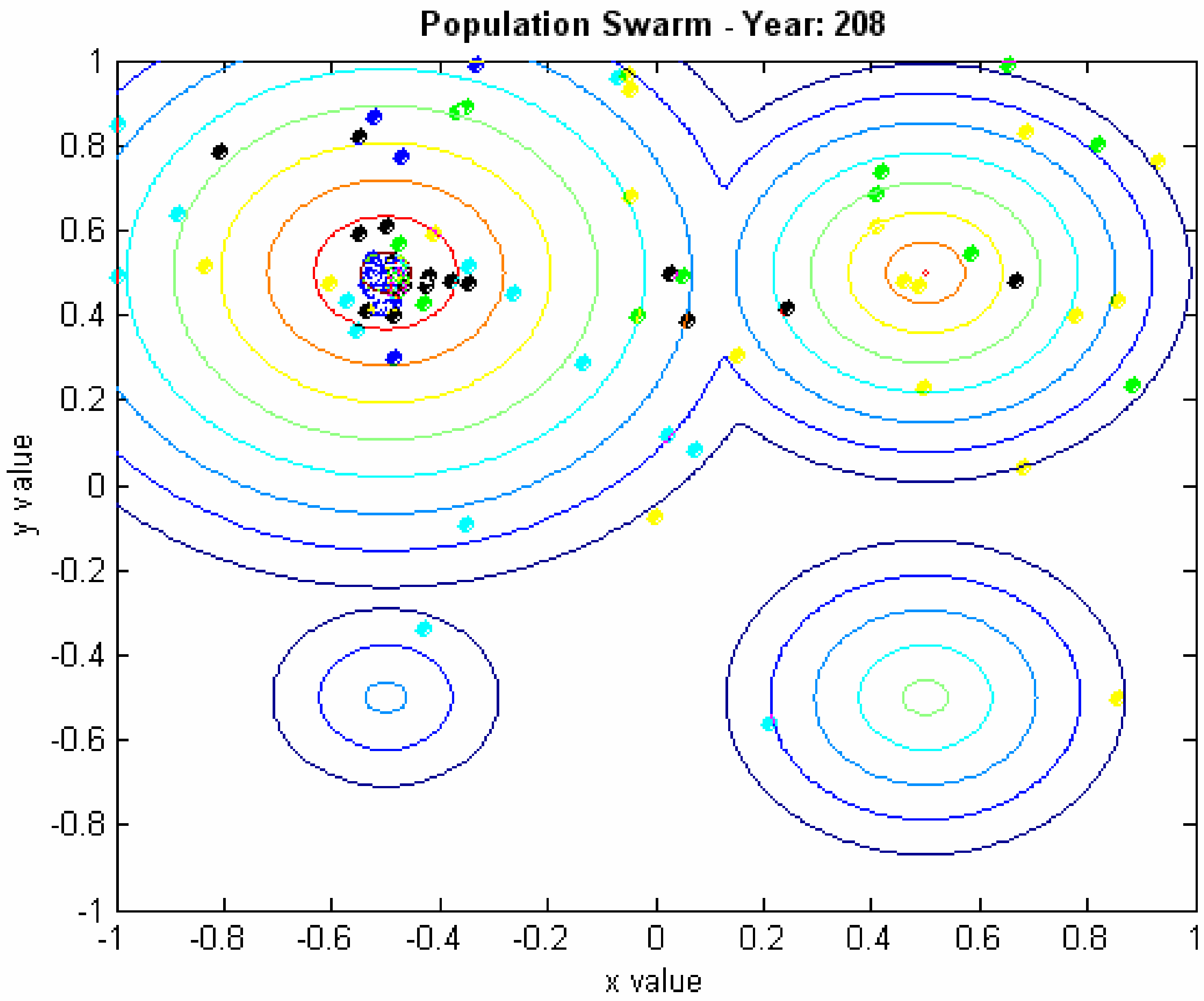
Population Swarm Emergence

- Initially the largest swarm located at current best resource peak.
- Note that a history individual resides in NW corner near previous best peak in last cycle.
- When new shift occurs this immediately draws individuals over to the new peak.
- Thus, exploration and exploitation still take place, but exploration can take a “shortcut” via backtracking information provided by history knowledge.



Population Swarm - Year: 200
y value
x value
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Population Swarm - Year: 203
y value
x value
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

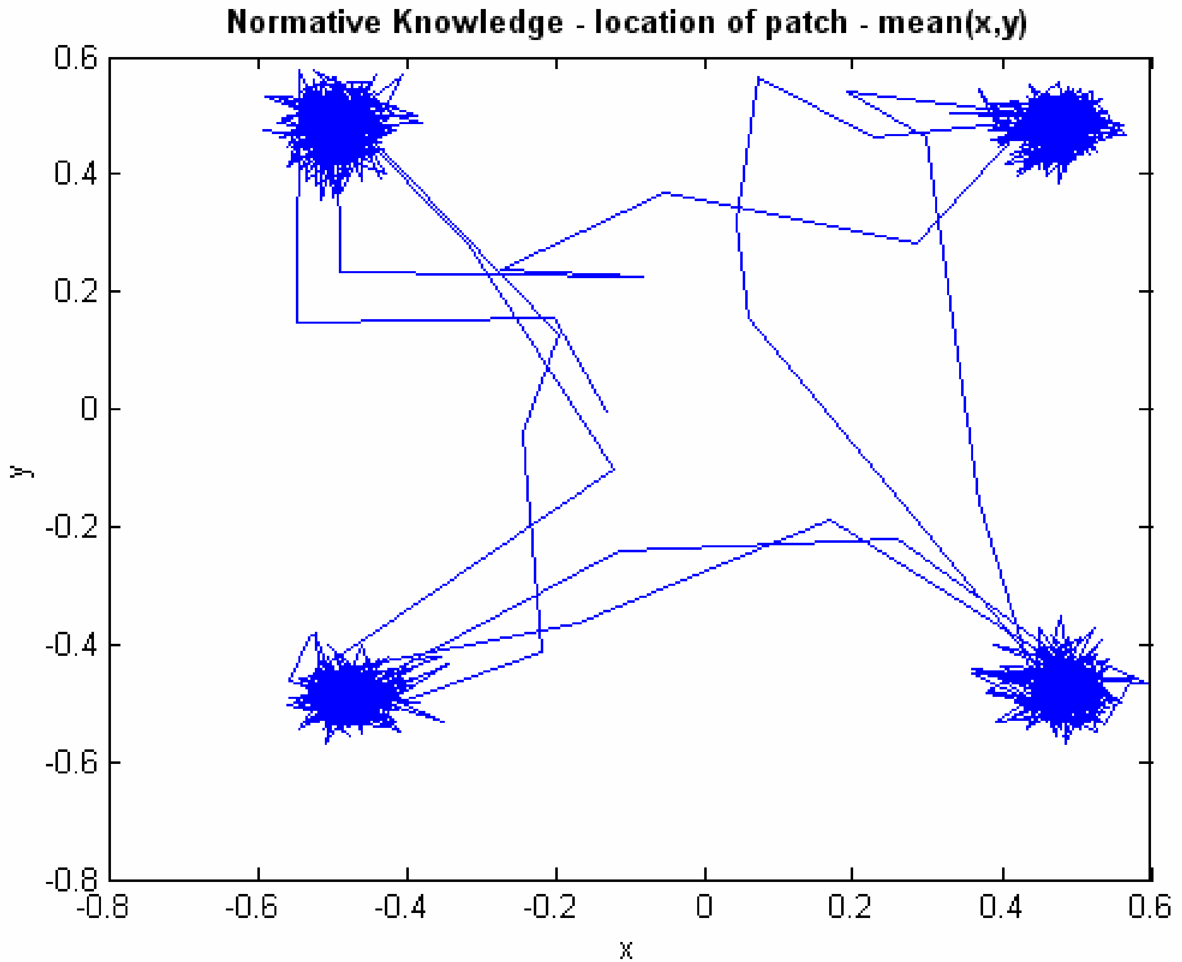


Population Swarm - Year: 208
y value
x value
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

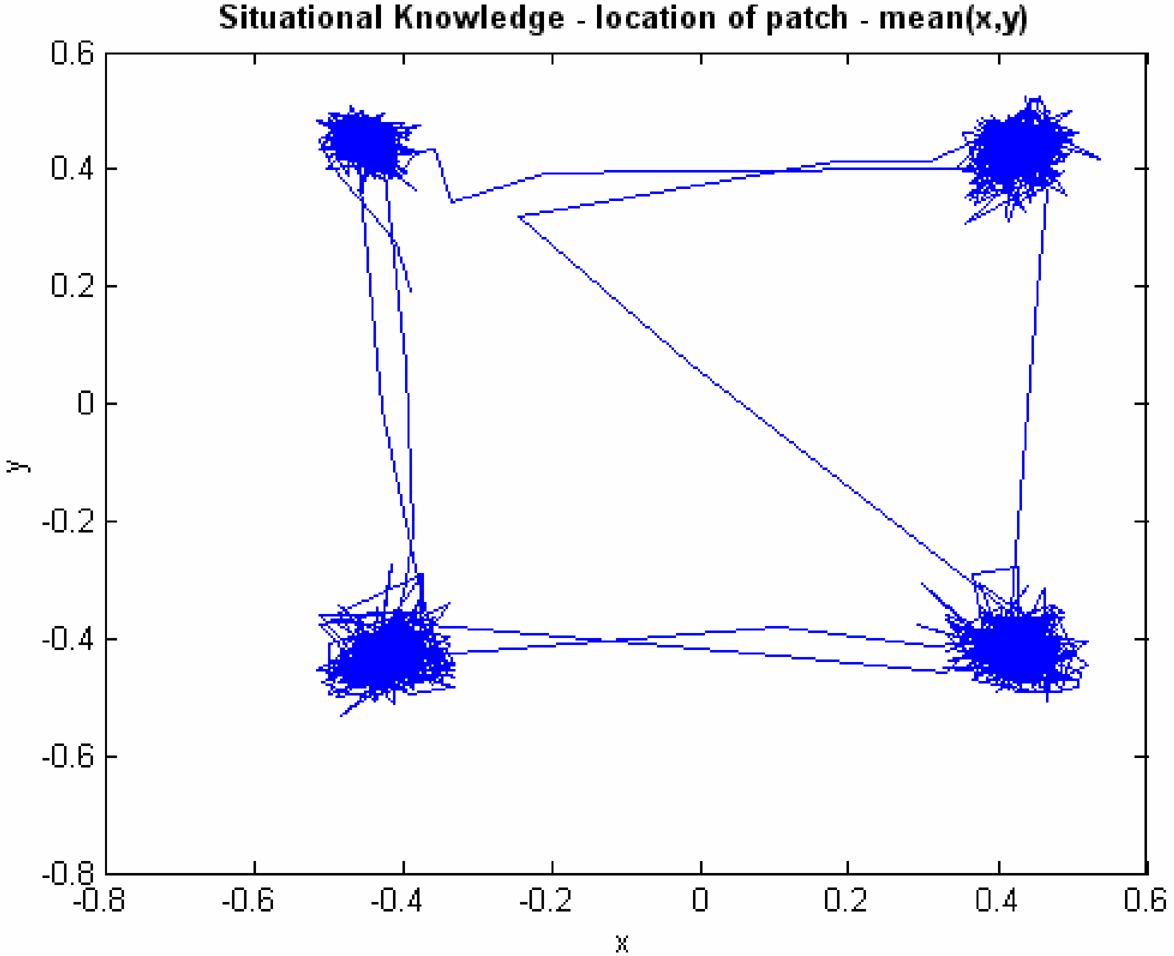
Different roles in dynamic envir.

- **Knowledge Resources**
  - **explorers**
    - **global**
    - **regional**
  - **exploiters**
    - **anticipator (history & domain)**
      - **anticipate next state**
      - **anticipate future state**
    - **reactor (situational)**

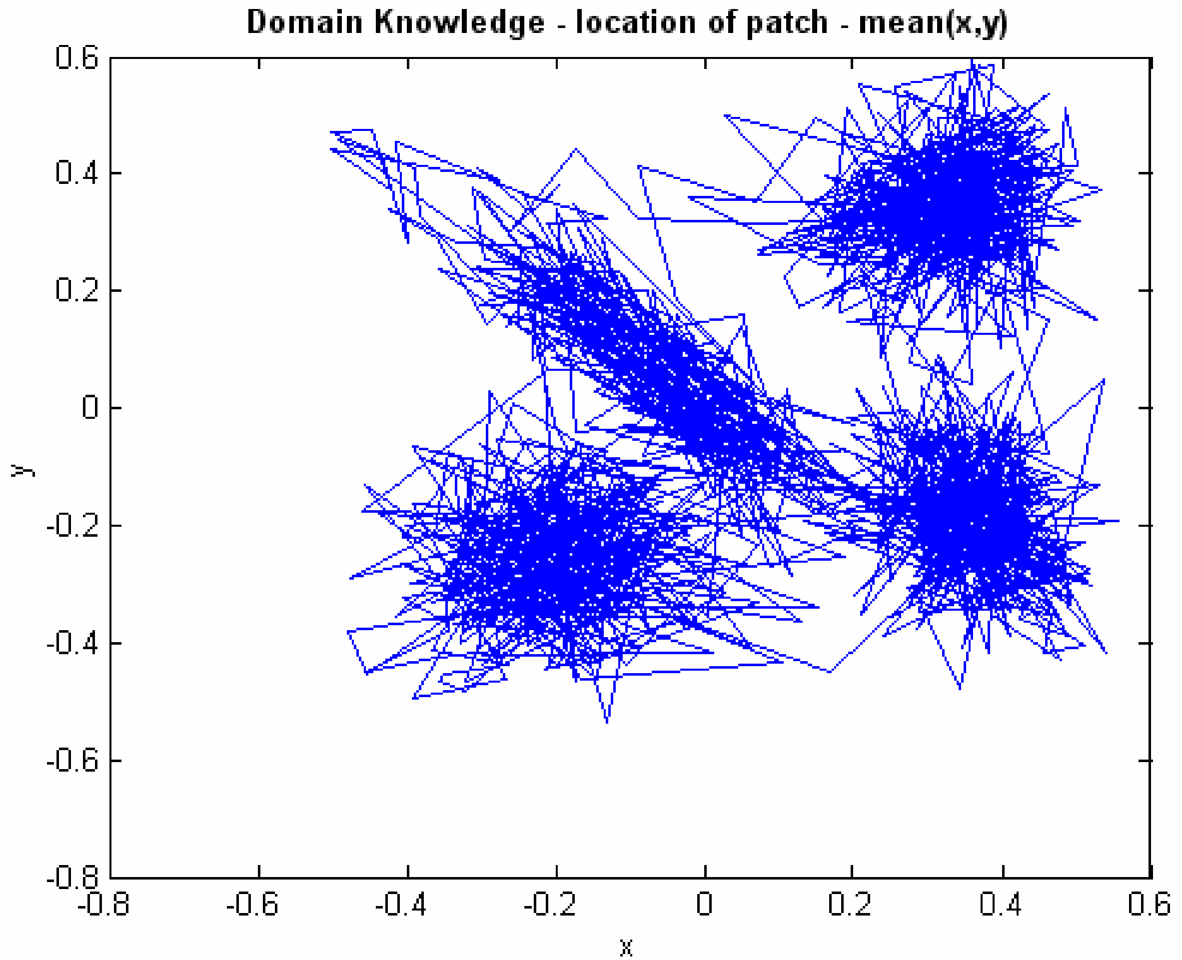
Normative Knowledge Trace



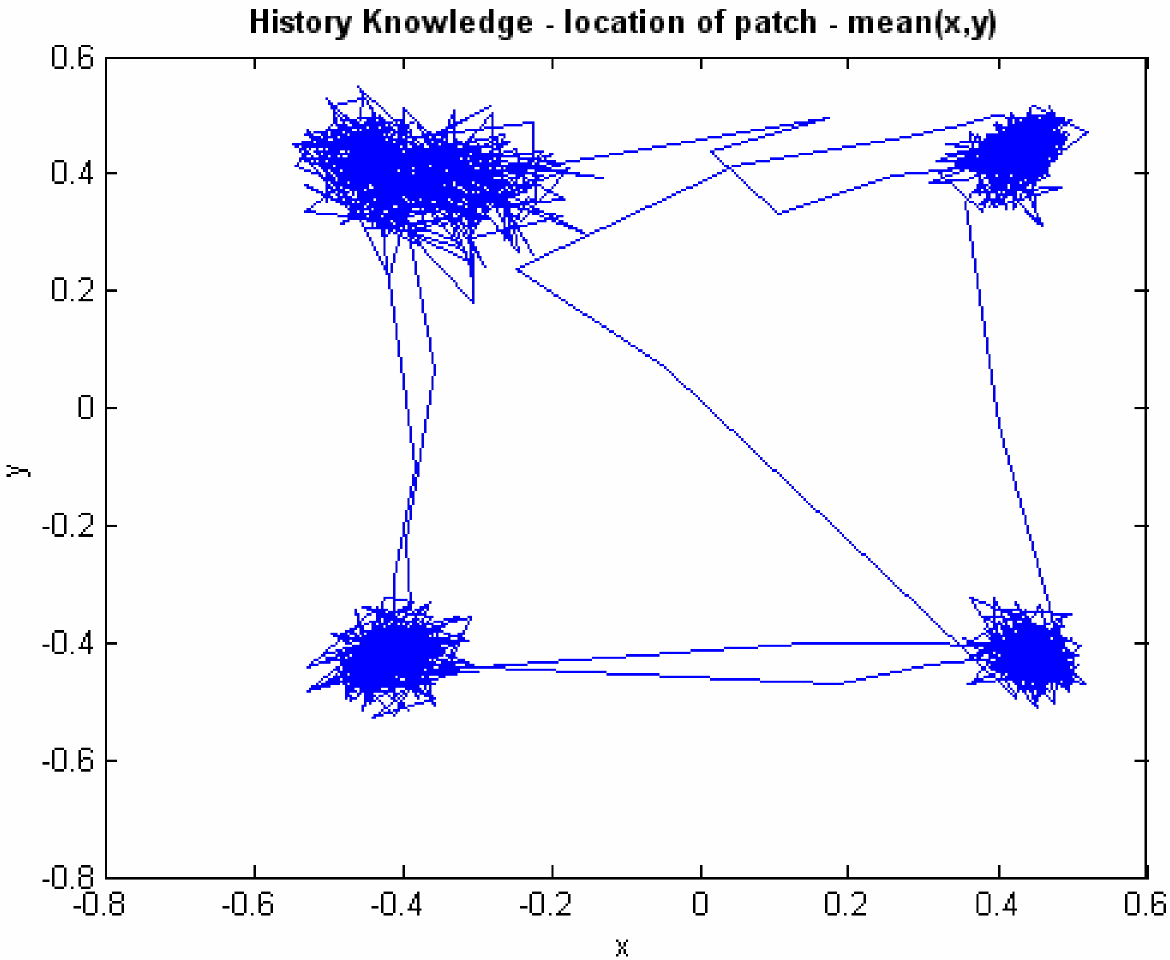
Situational Knowledge Trace



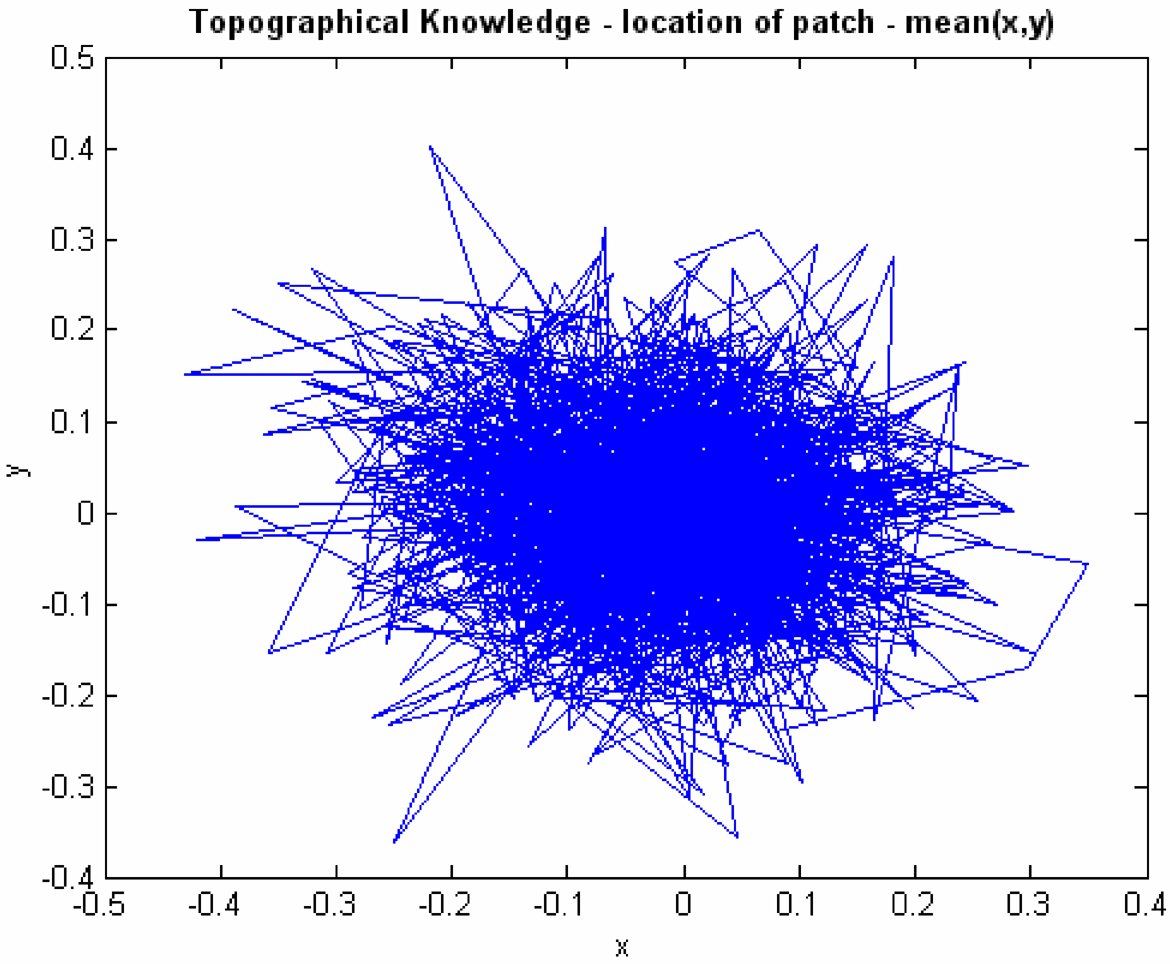
Domain Knowledge Trace



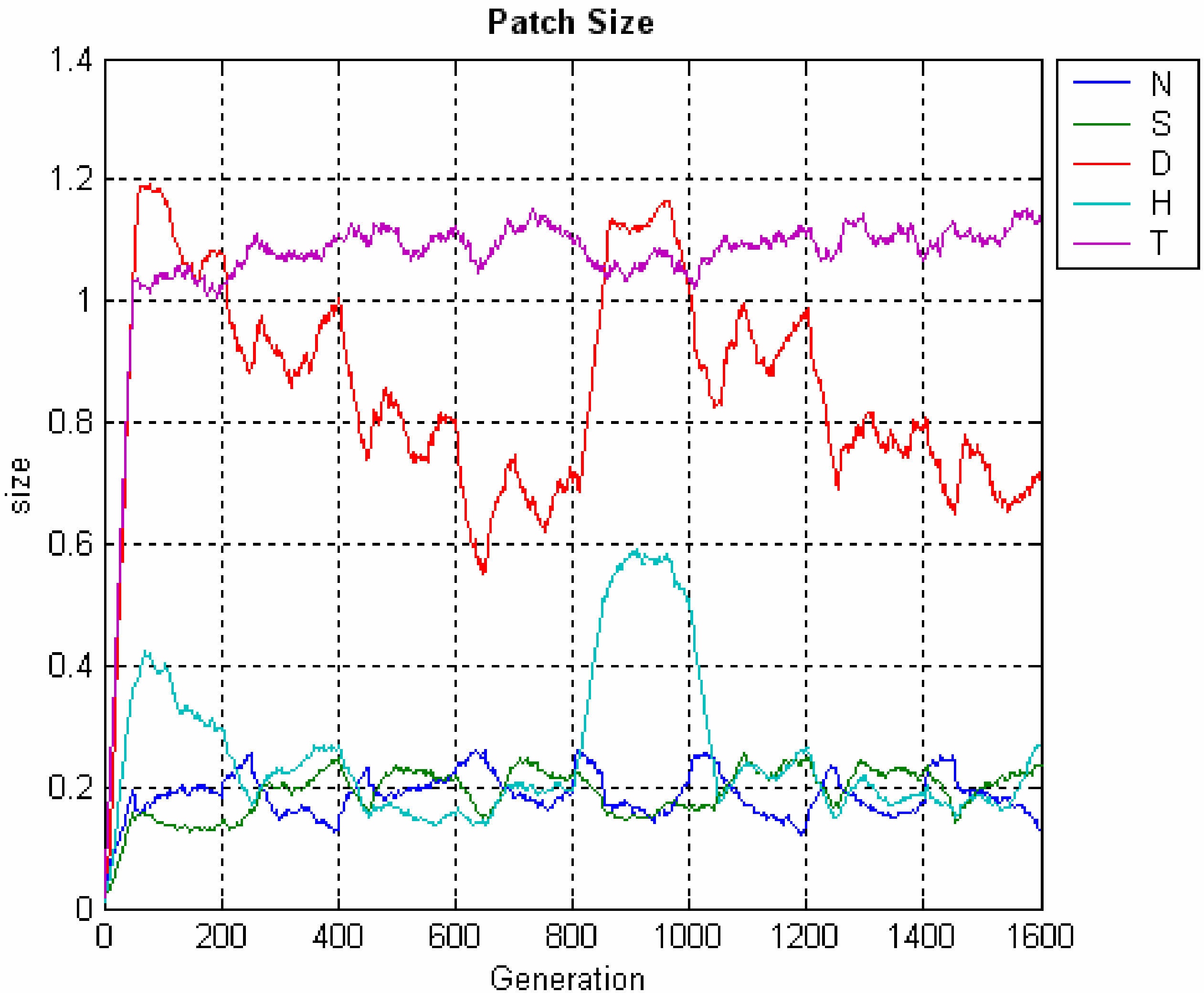
History Knowledge Trace



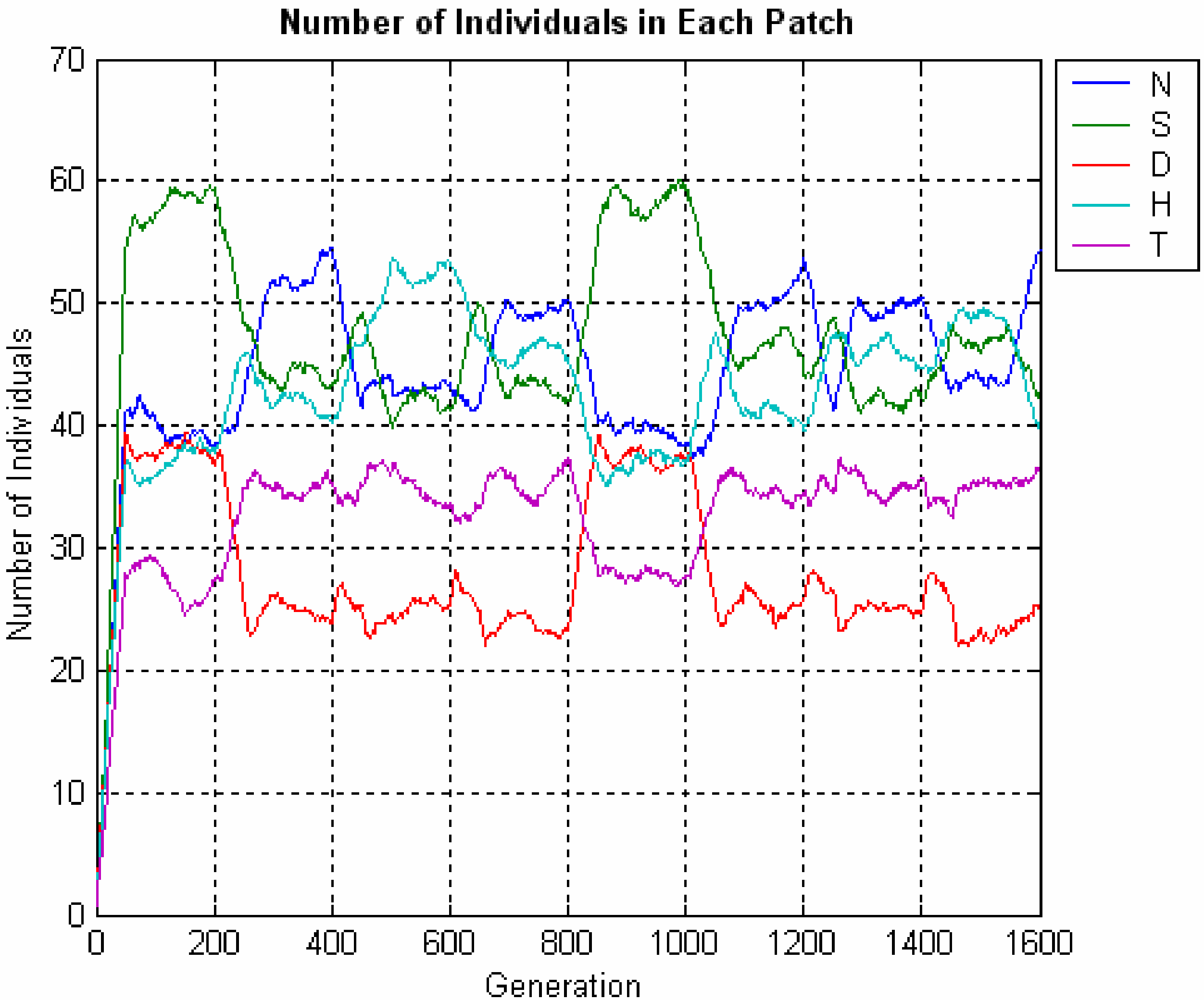
Topographic Knowledge Trace



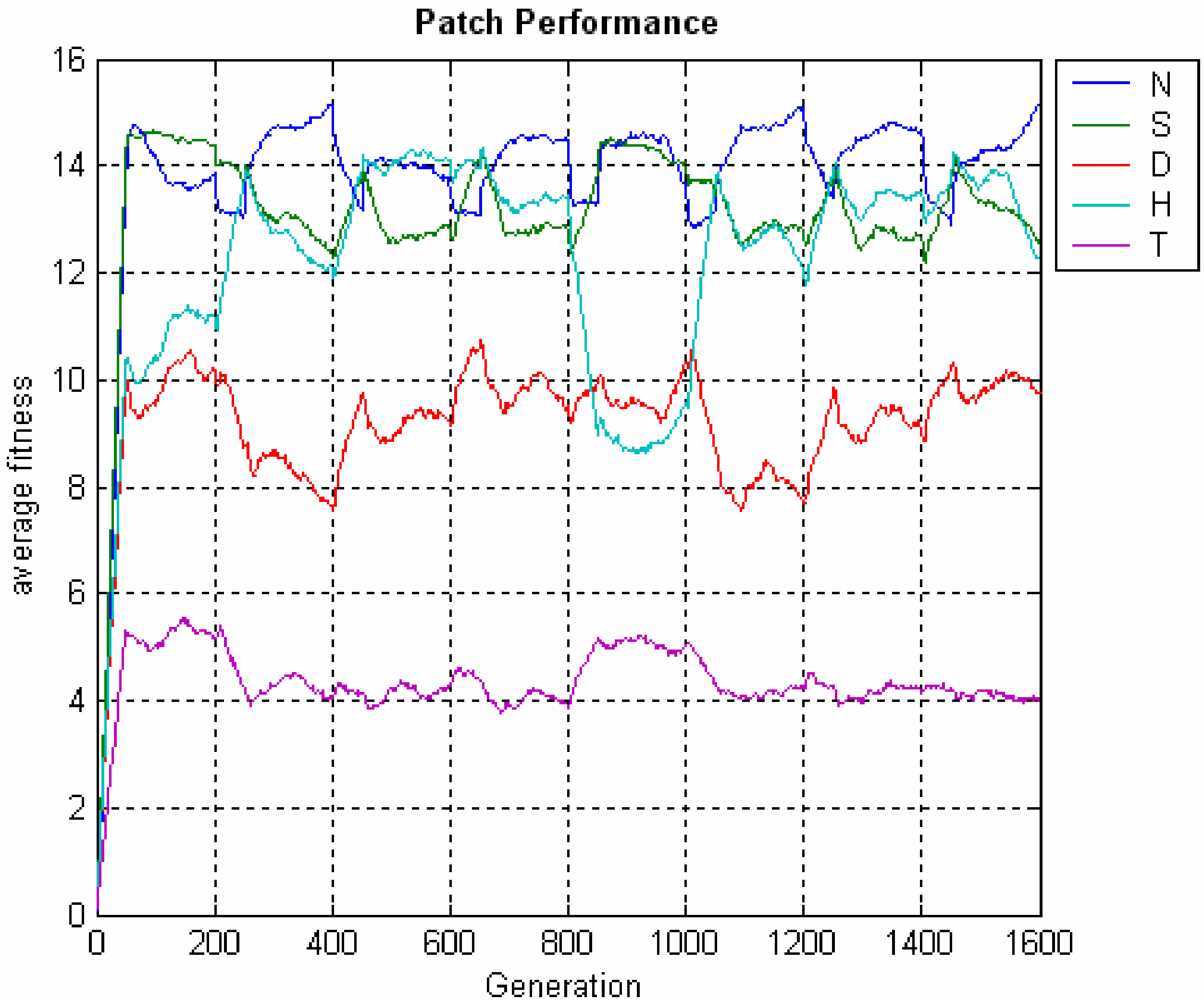
200 generations / per shift, 4 shifts back to original place



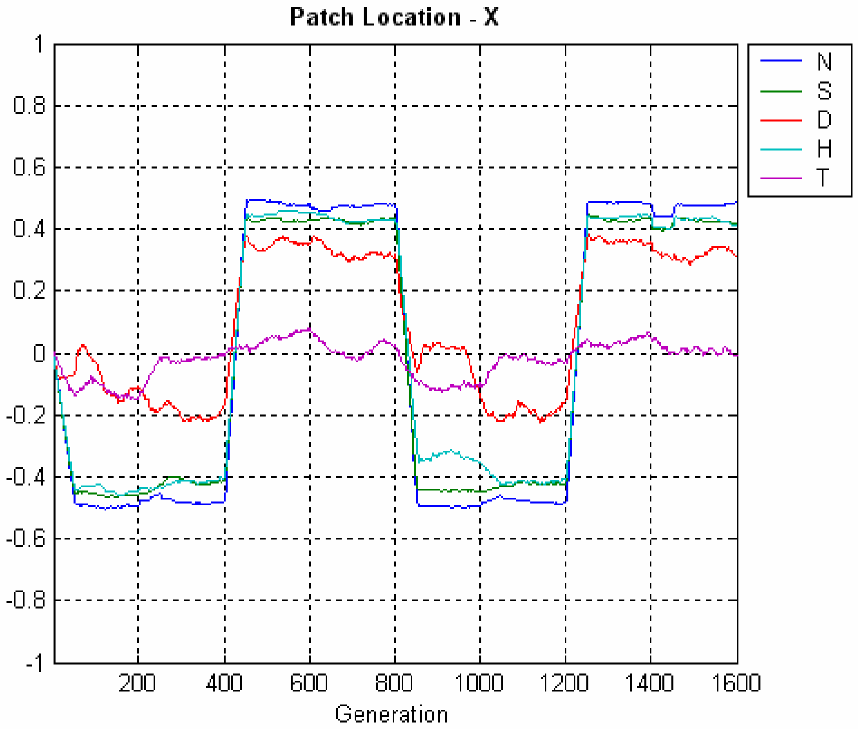
Number of Individuals in the patches

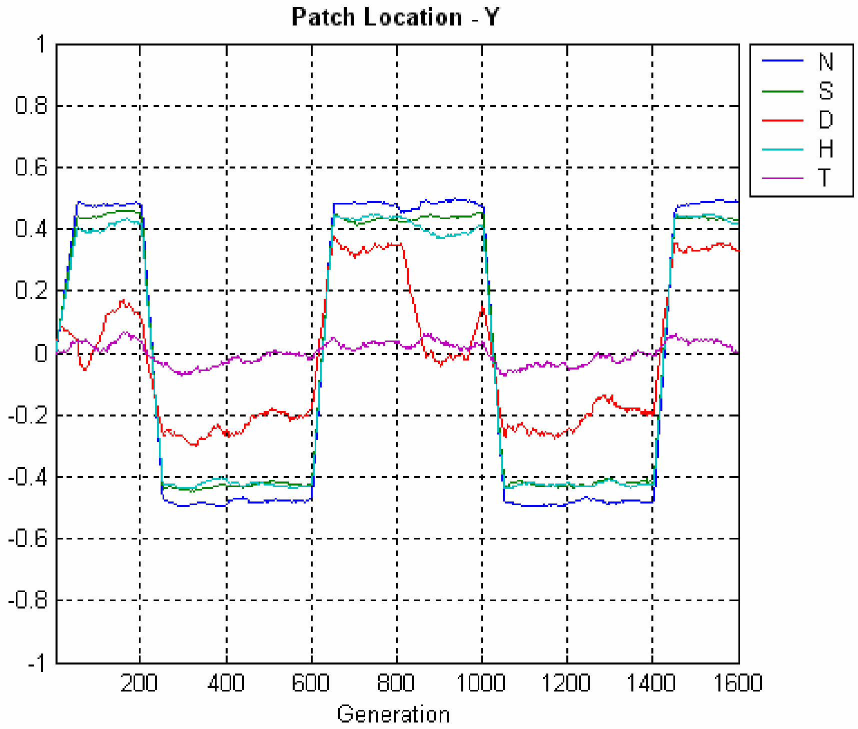


Averages of Fitness for Individuals in the patches

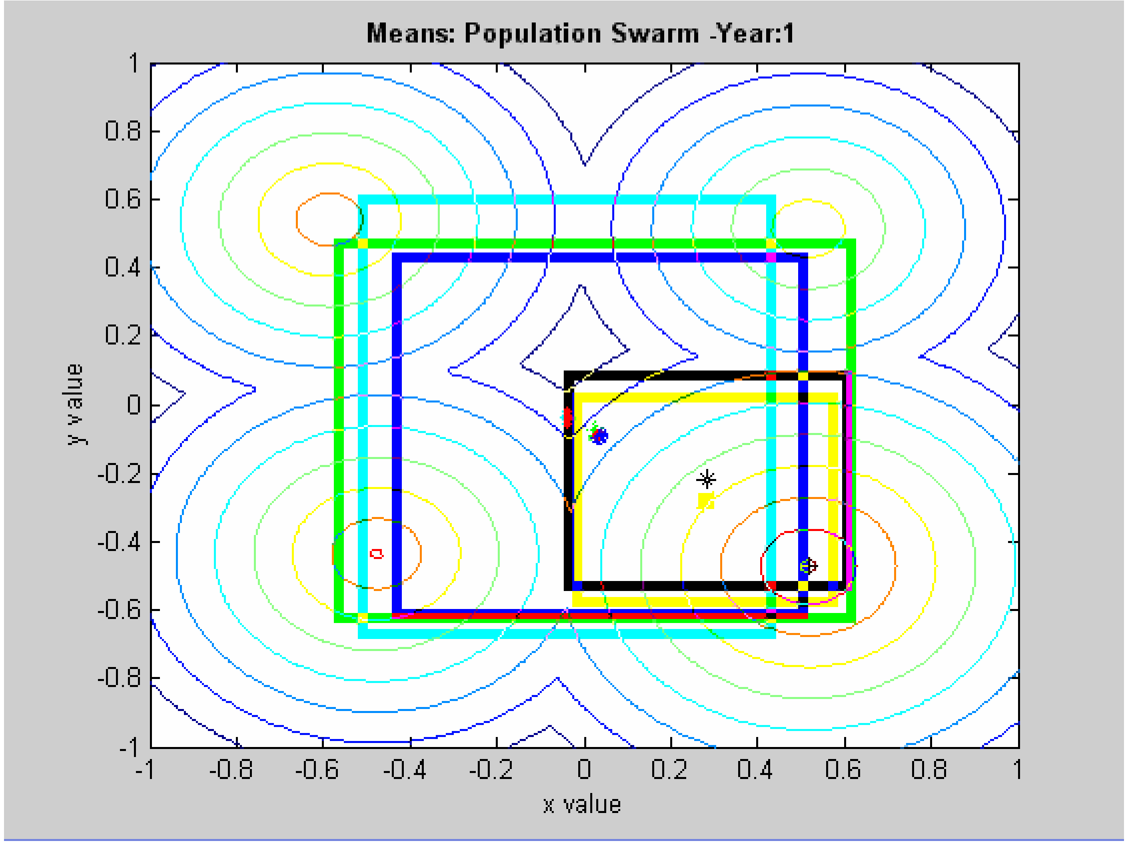


Location of patch center

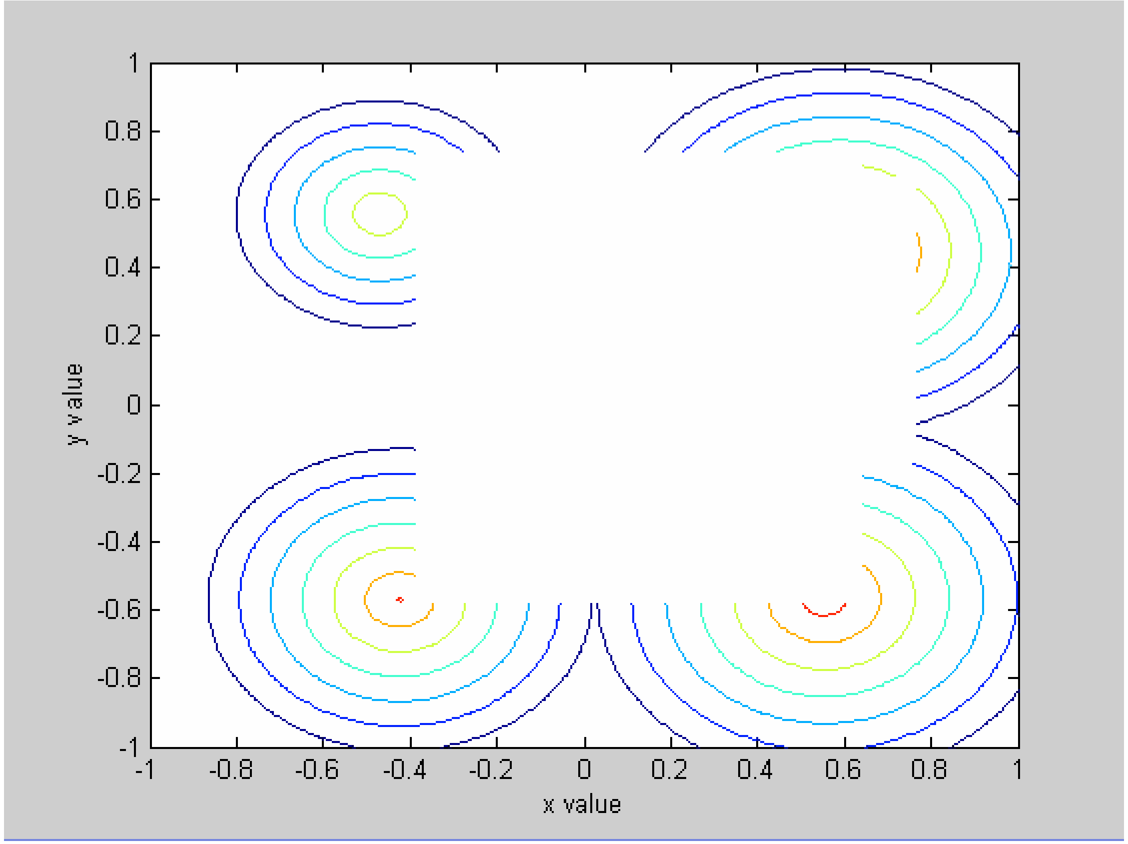




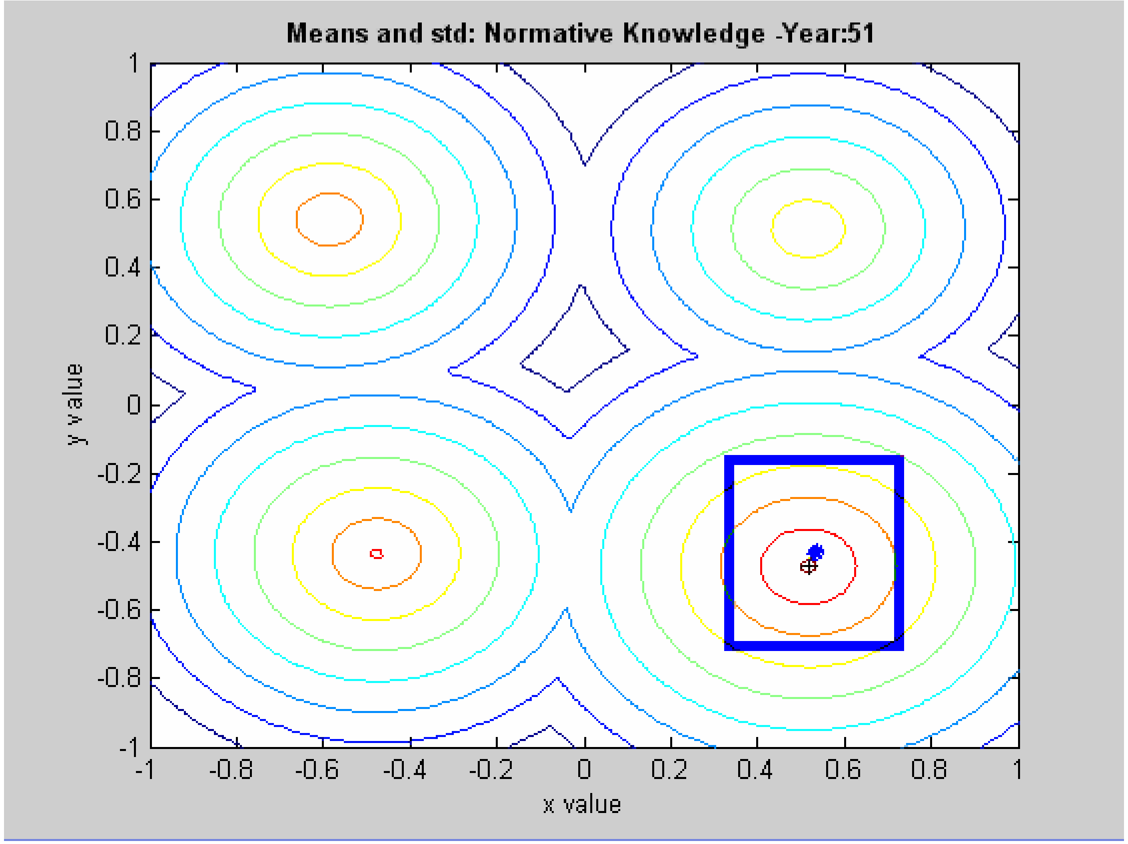
200 years/landscape – All five



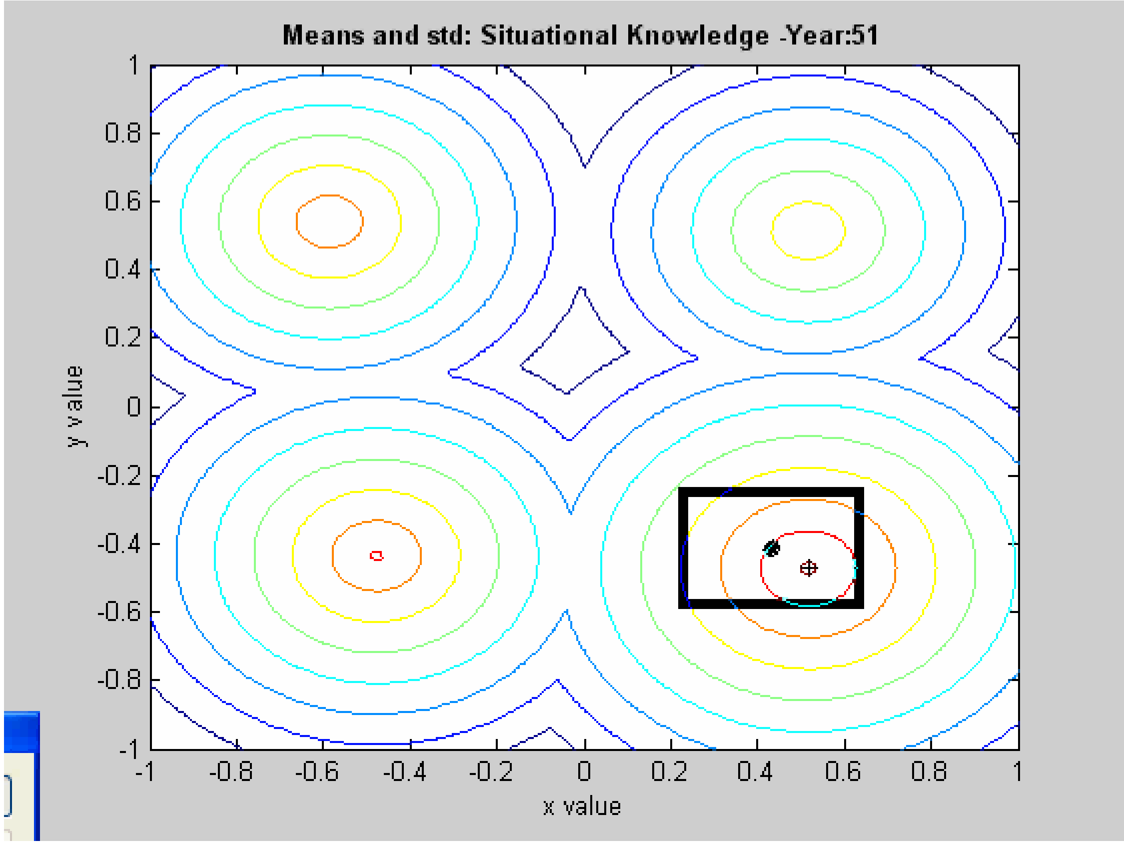
50 years/landscape – All Five



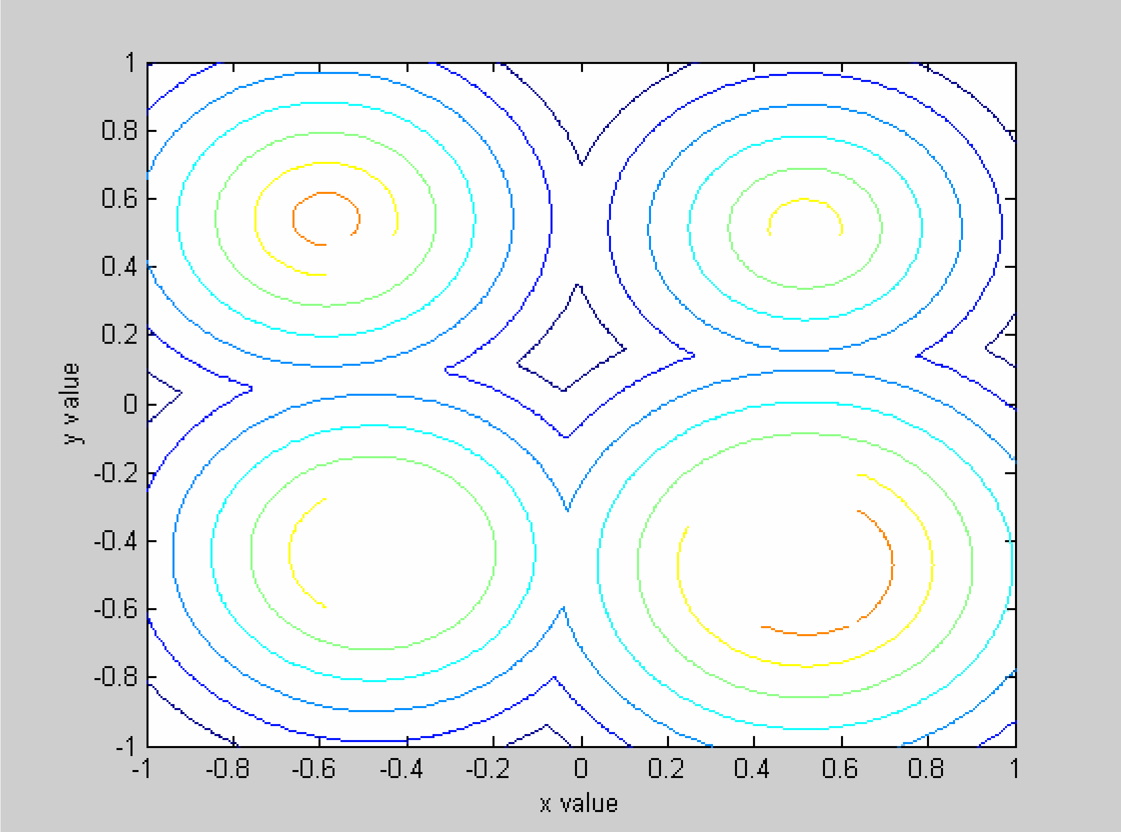
Movie: 200 years/landscape - Normative



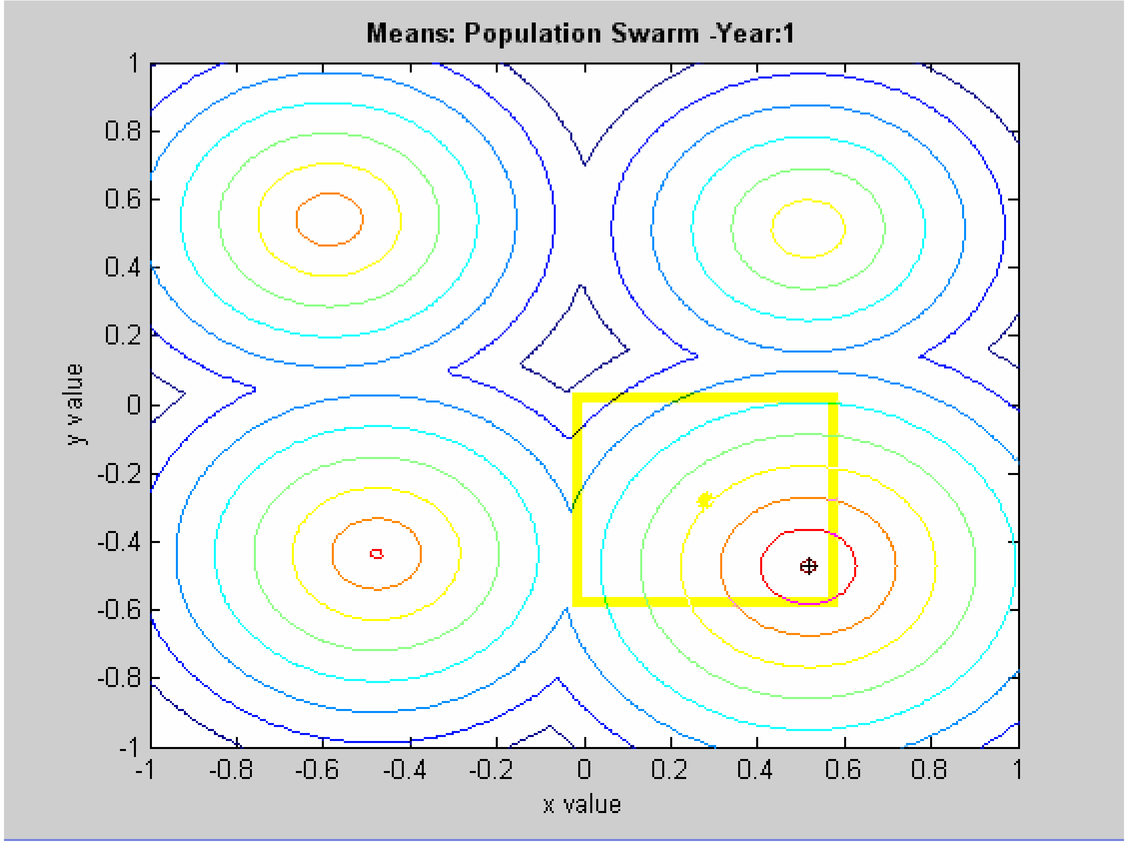
200 years/landscape - Situational



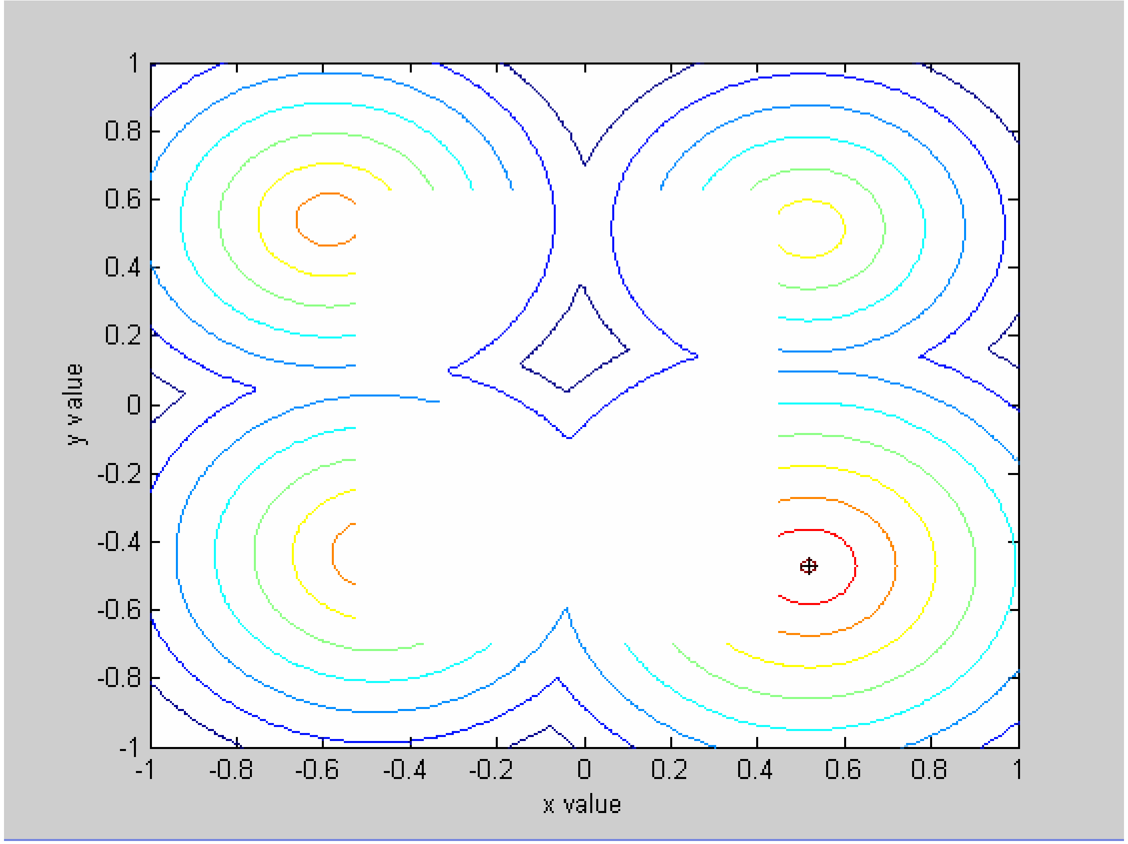
200 years/landscape - Domain



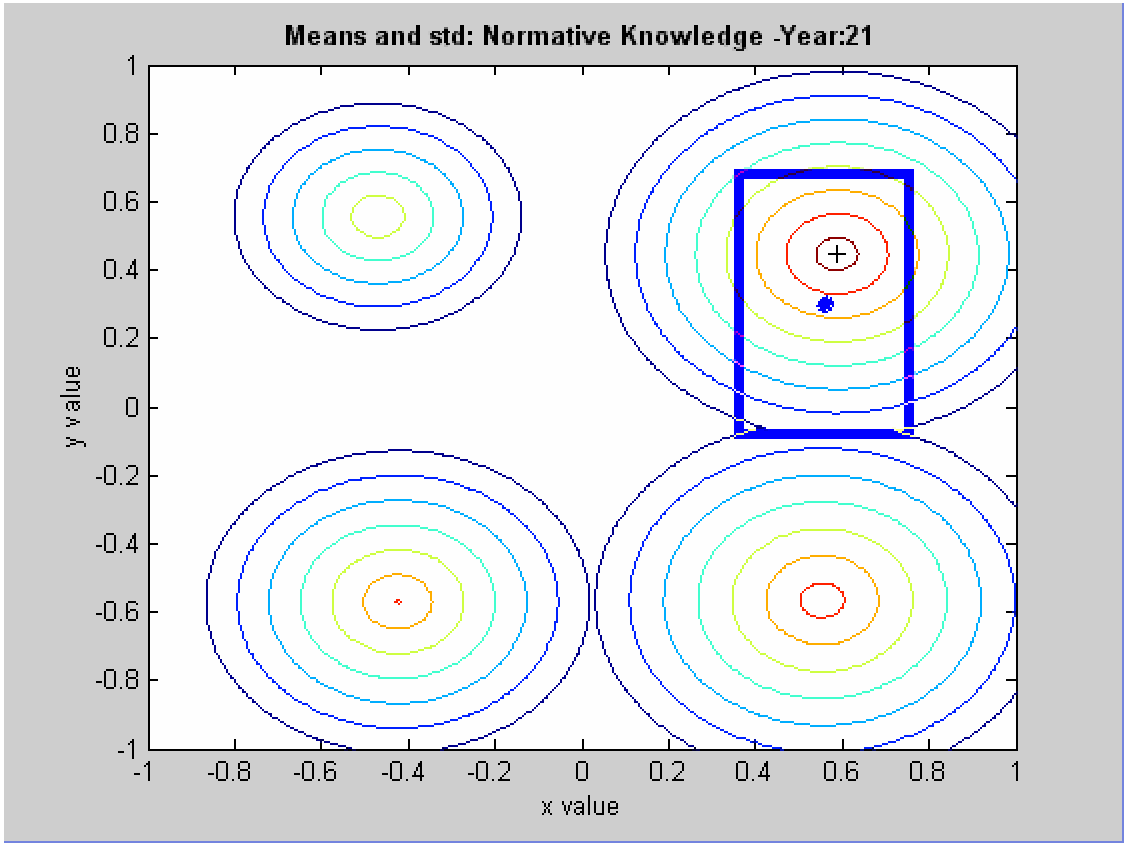
200 years/landscape - History



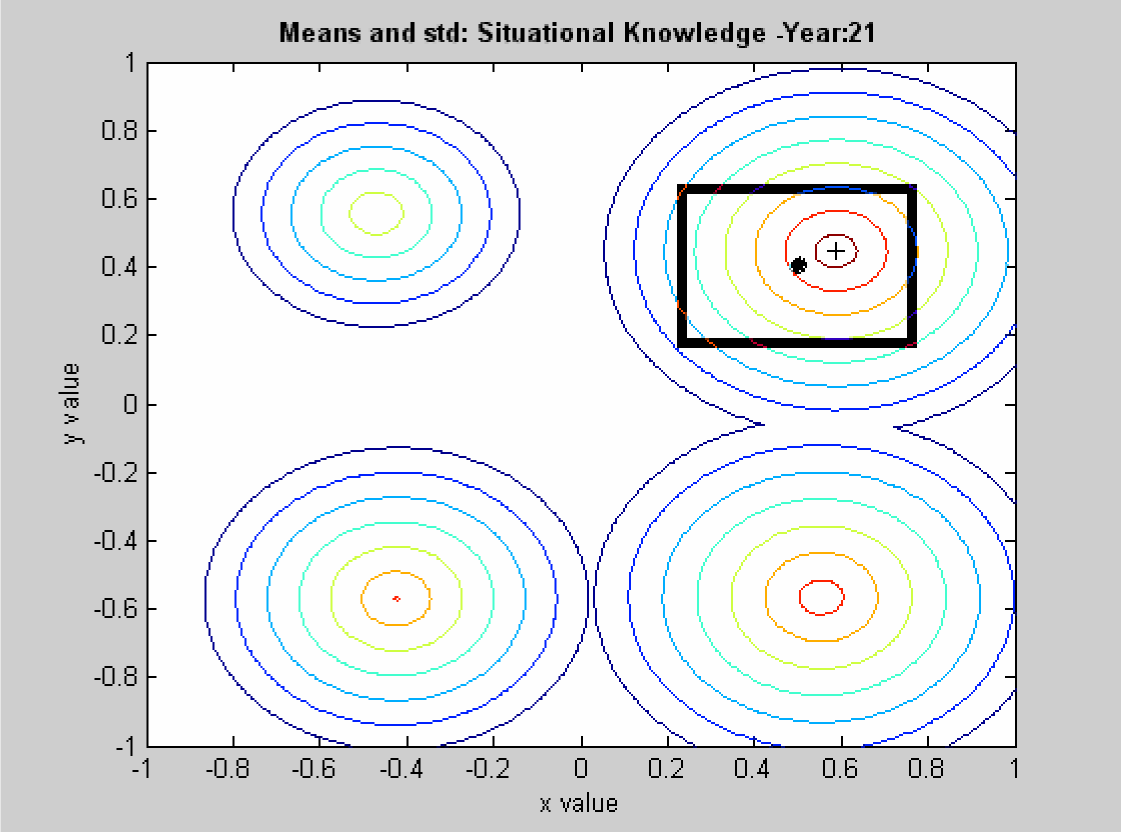
200 years/landscape - Topographical



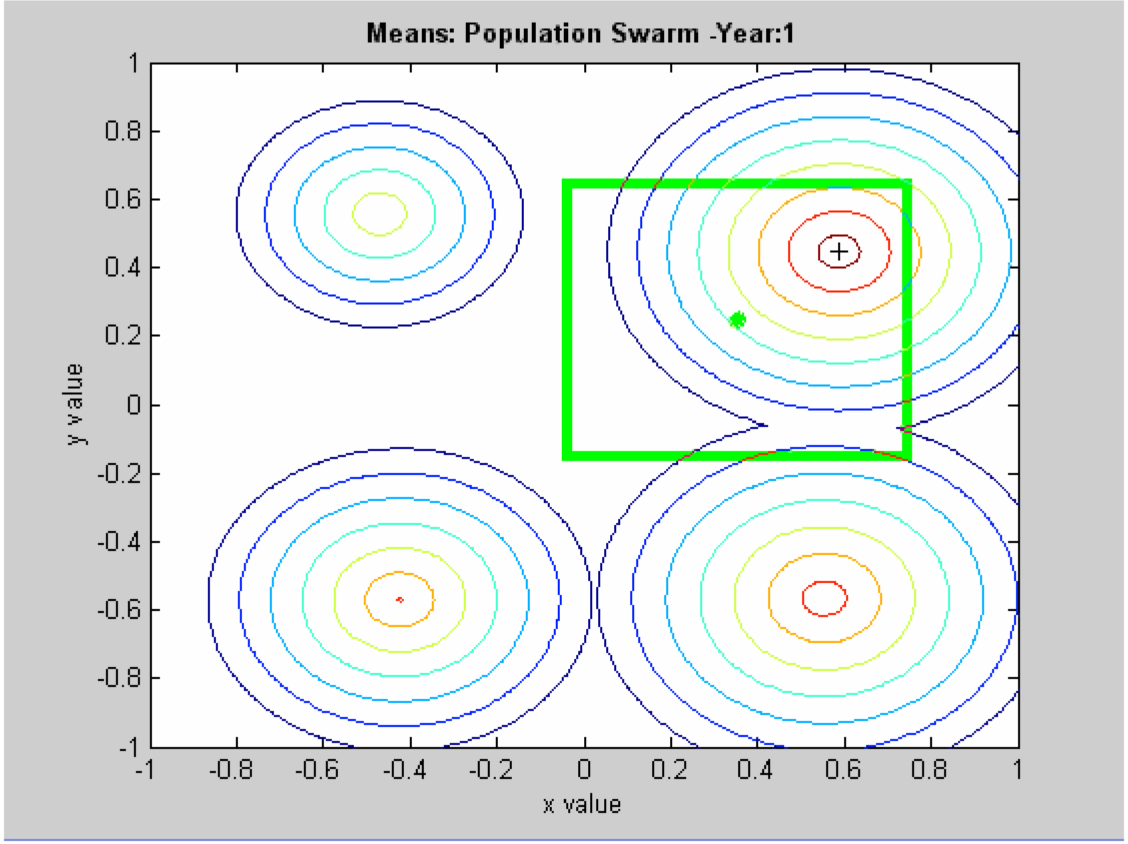
50 years/landscape - Normative



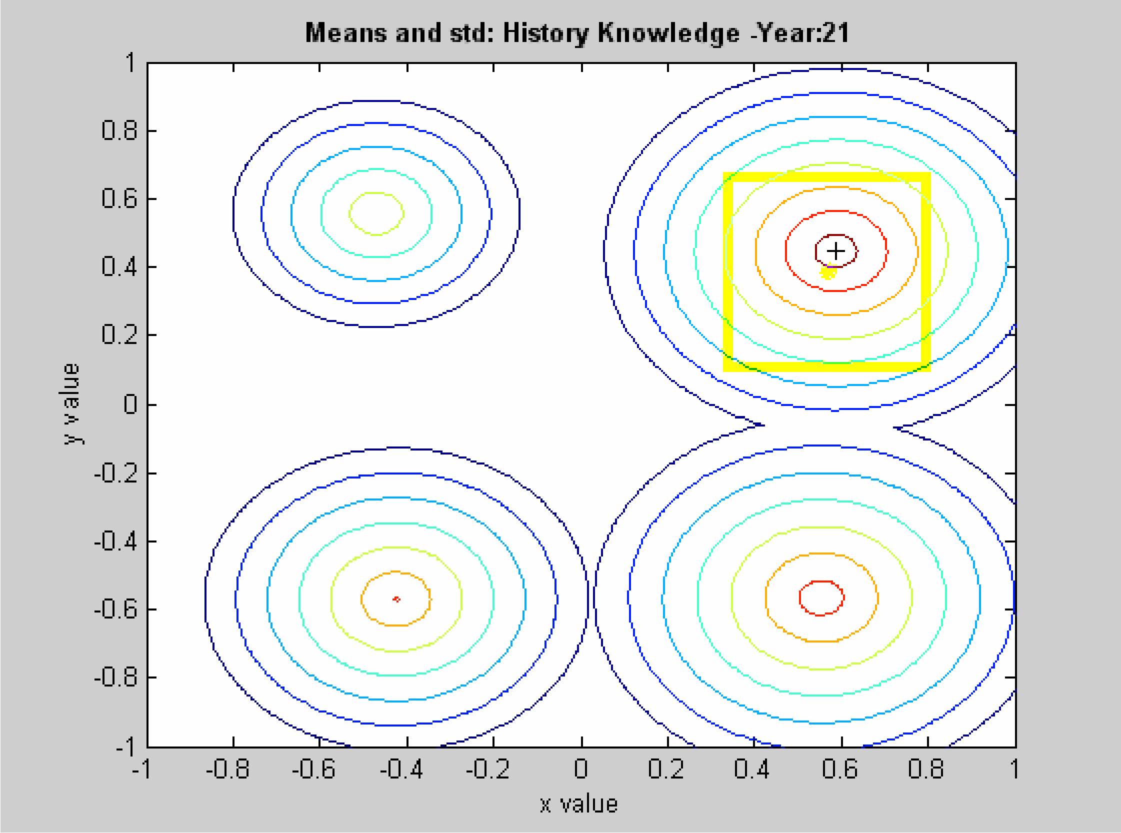
50 years/landscape - Situational



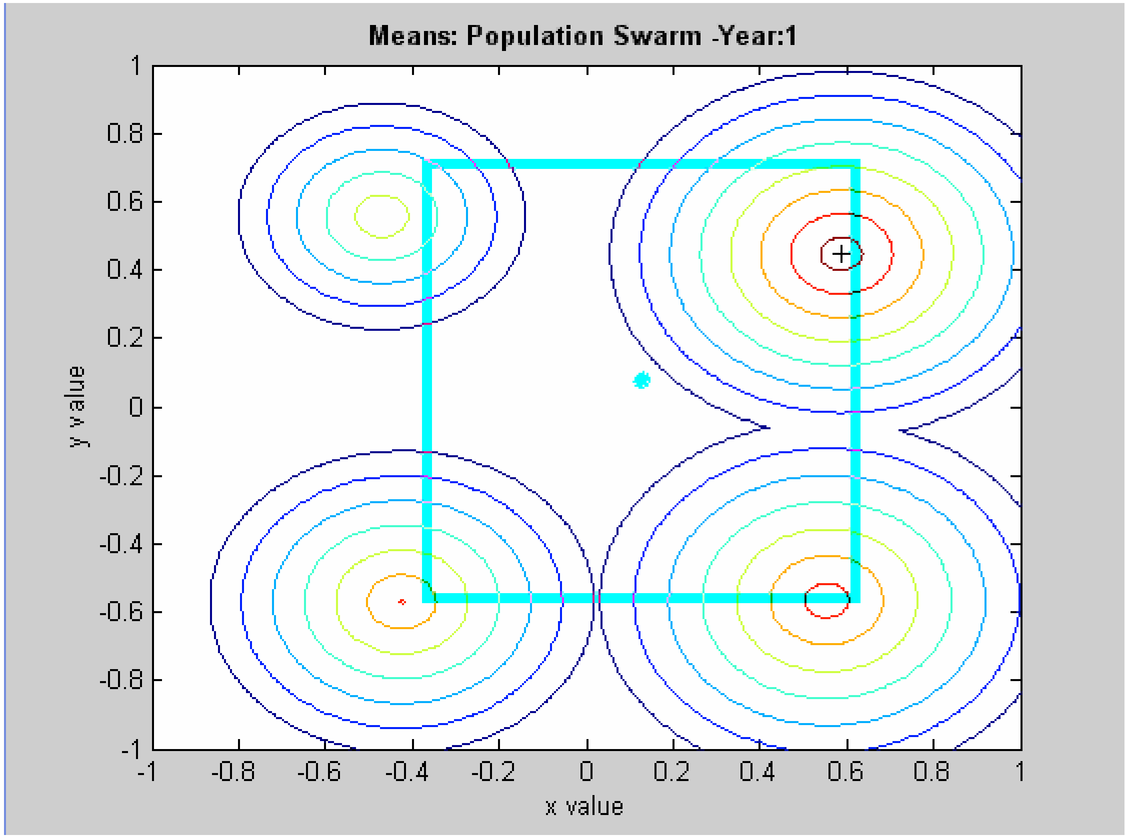
50 years/landscape - Domain

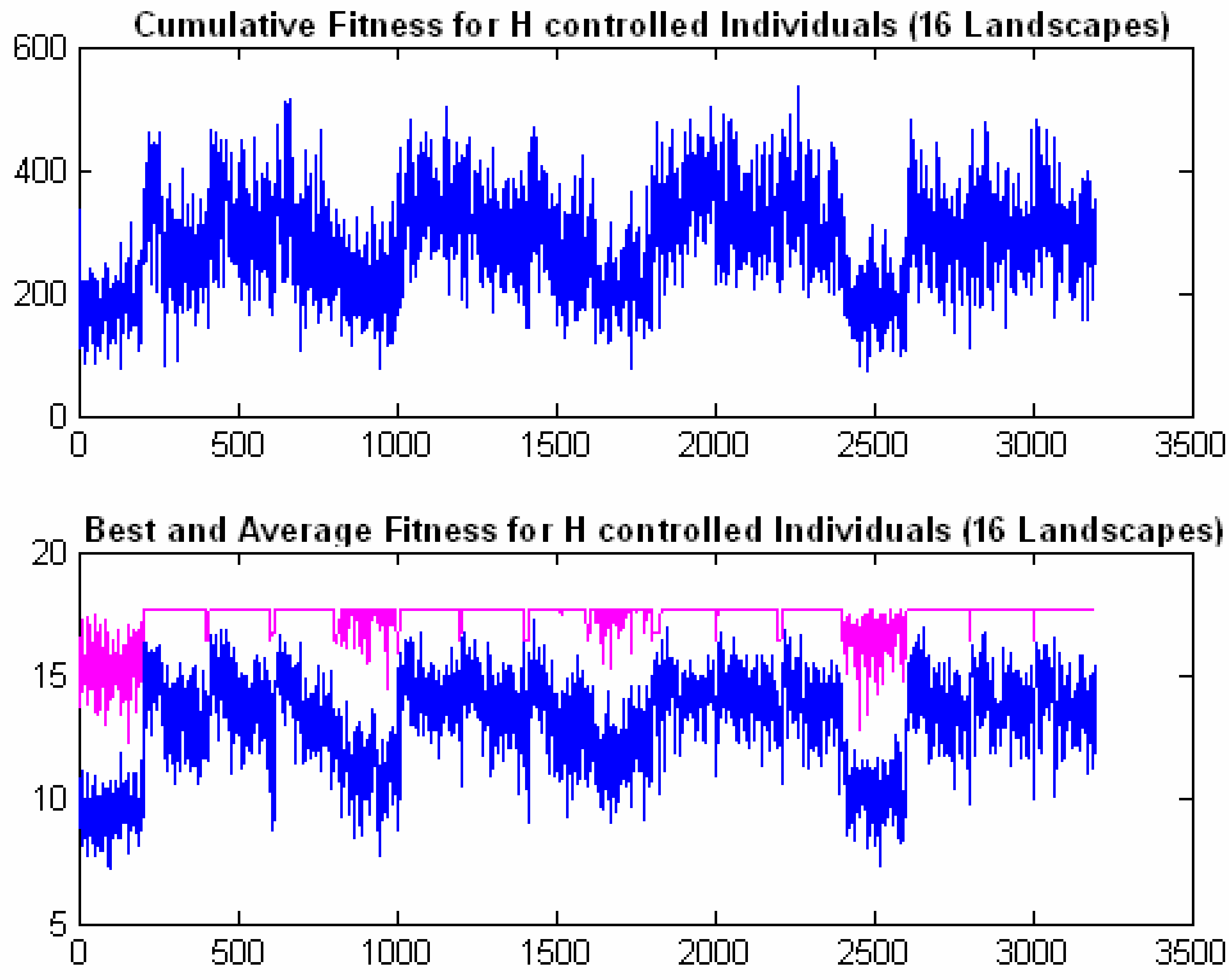


50 years/landscape - History



50 years/landscape - Topographical



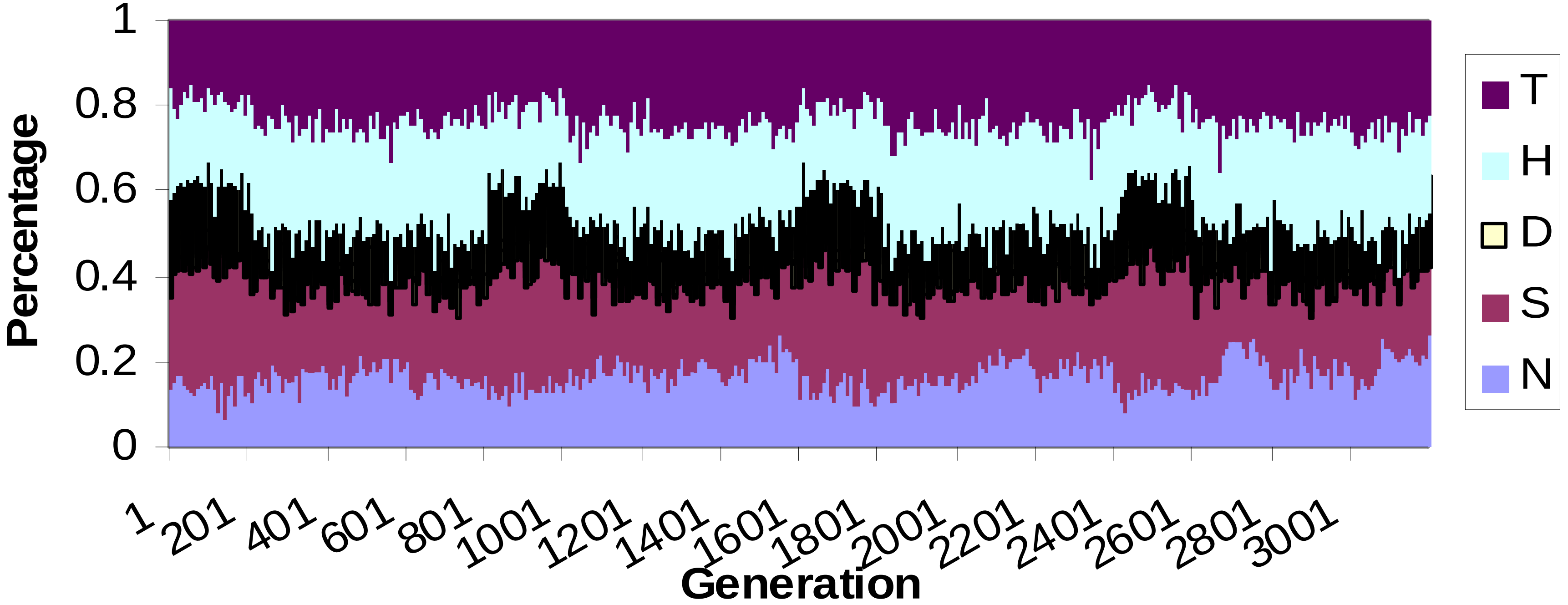


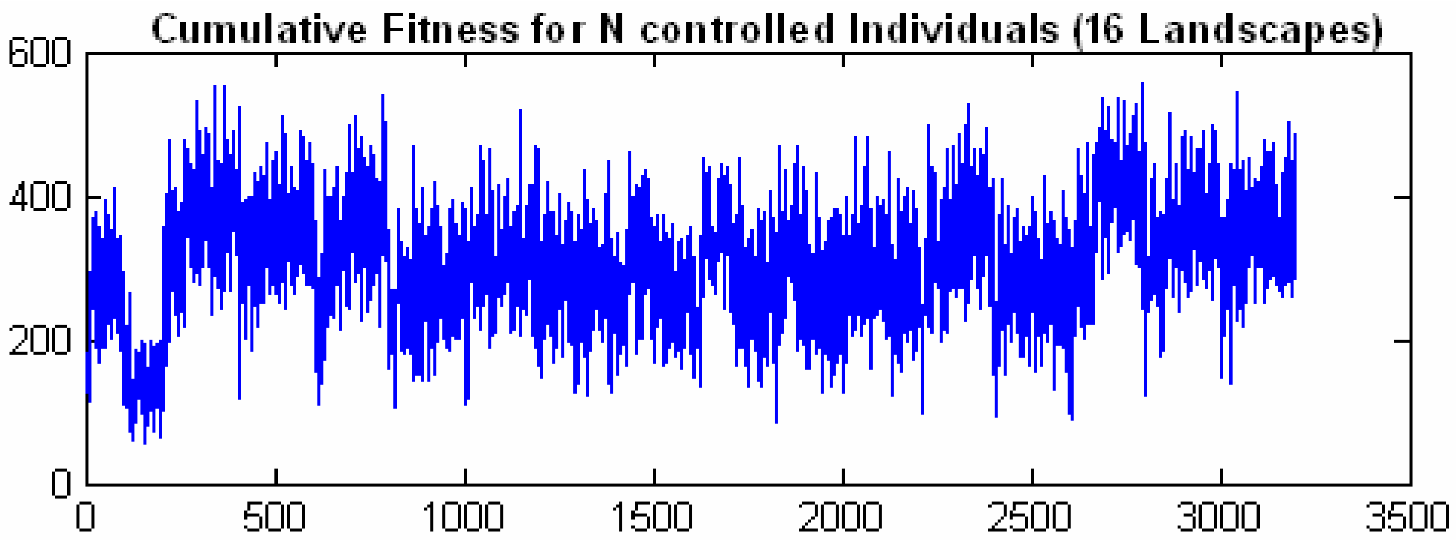
Cumulative Fitness for H controlled Individuals (16 Landscapes)
600
400
200
0
0
500
1000
1500
2000
2500
3000
3500
Best and Average Fitness for H controlled Individuals (16 Landscapes)
20
15
10
5
0
500
1000
1500
2000
2500
3000
3500

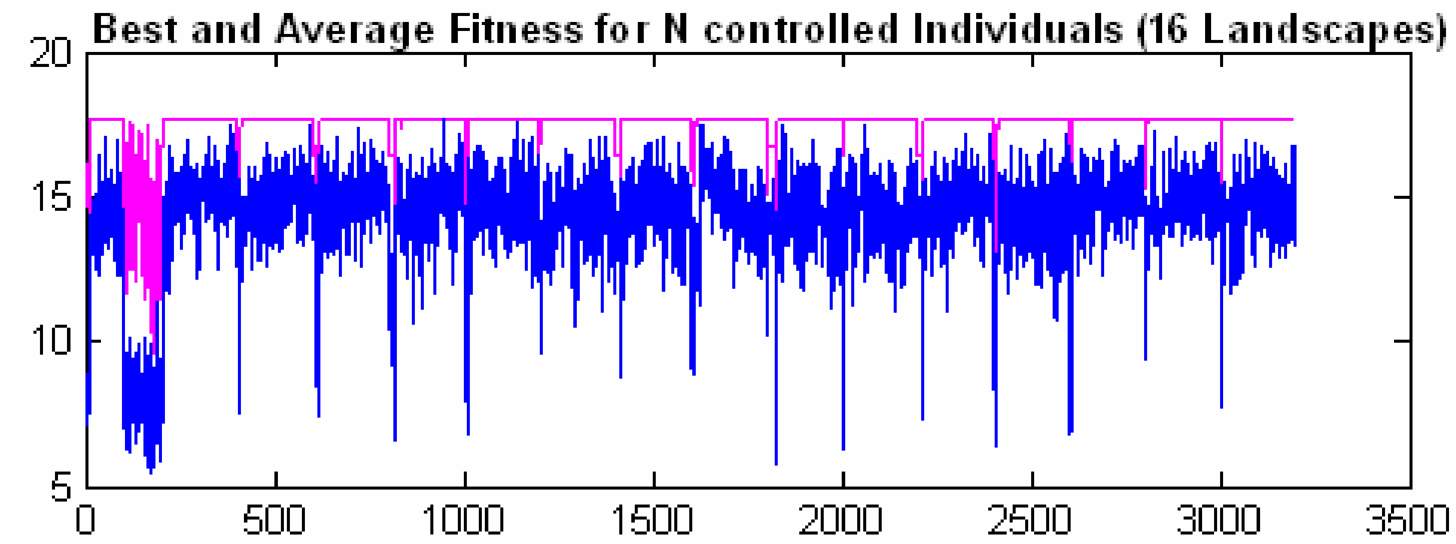
Energy Intake in Dynamic Environments

- What we saw for normative knowledge in one static environment is repeated over and over as the environment is shifted.
- History knowledge cycles differently as previous information is exploited anew.

Population Influenced by Each Knowledge Type (per Generation)

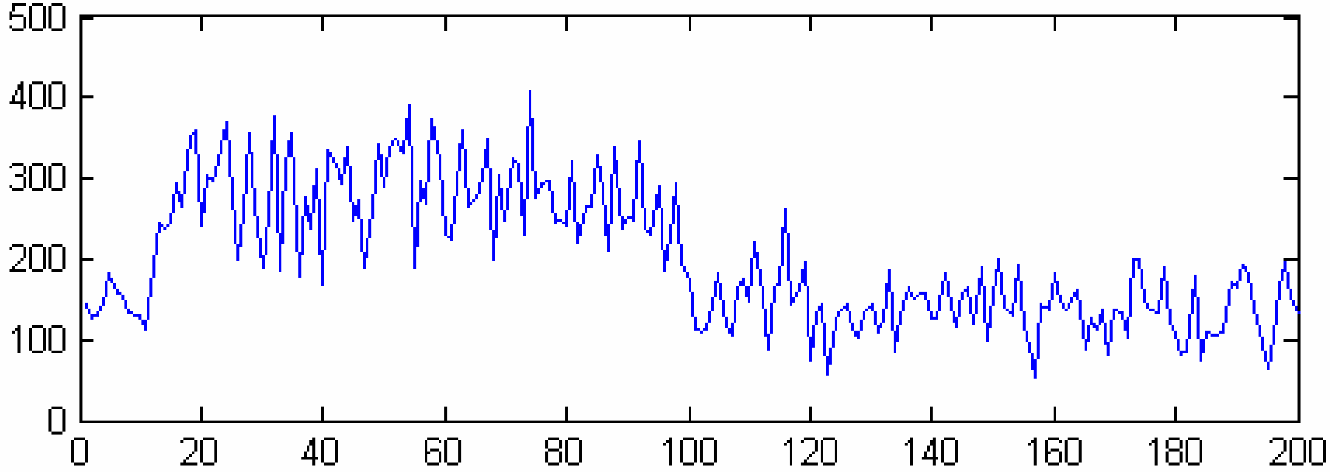


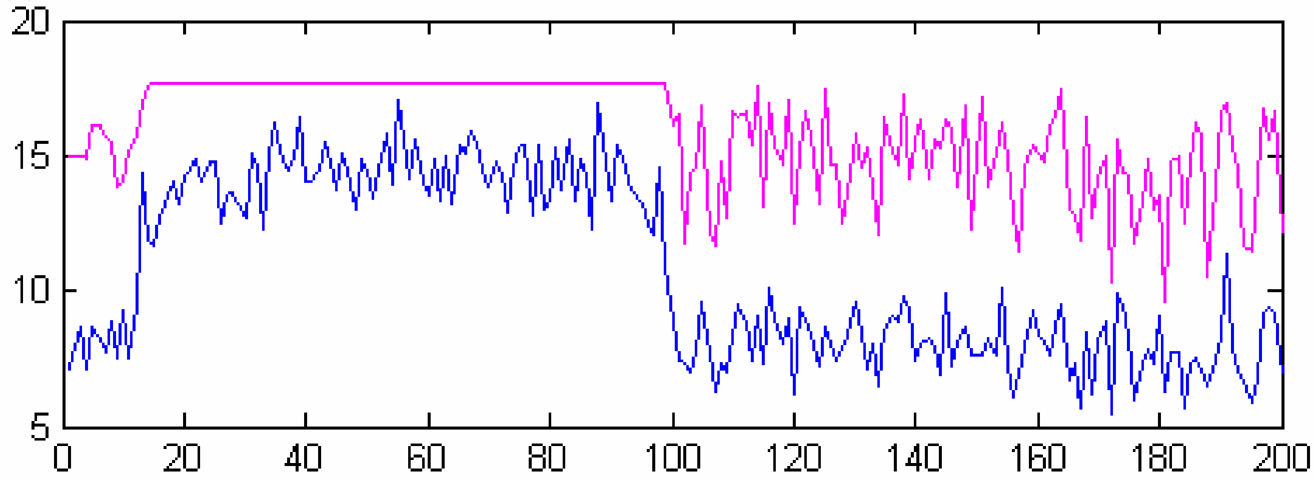

Cumulative Fitness for N controlled Individuals (16 Landscapes)
600
400
200
0
0
500
1000
1500
2000
2500
3000
3500


Best and Average Fitness for N controlled Individuals (16 Landscapes)
20
15
10
5
0
500
1000
1500
2000
2500
3000
3500

What does the Energy Intake Curve Look Like?

- Show accumulated performance and average per year for Normative Knowledge. This is one of the “explorer” patches which induces discovery by changing the “norms of a population.
- Given for a static configuration.


500
400
300
200
100
0
0
20
40
60
80
100
120
140
160
180
200


20
15
10
5
0
20
40
60
80
100
120
140
160
180
200

Conclusions

- **Flocking is viewed here not as the result of pure individual actions but the fact that individuals make similar decisions relative to shared knowledge. <u>Knowledge-Driven</u>.**
- **The use of the <u>Marginal Value Theorem</u> to drive the application of knowledge produces several emergent structures at the knowledge, population, and individual level in both static and dynamic environments.**
- **Swarming of individuals in population space (<u>population swarms</u>) is a result of swarming of knowledge in the belief space (<u>knowledge swarms</u>).**

Conclusions

- Repeated phases of problem solving lead to the emergence of **roles**, which here means that an individually is repeatedly controlled by the same knowledge sources over several periods of time.
- In dynamic environments, additional roles emerged as a response to additional environmental information.
- Suggests that if animal groups are monitored over a longer dynamic period more than just the two roles will emerge.