Swarm Intelligence

Particle Swarm Optimization

Erick Luerken

13.Feb.2006

CS 790R, University of Nevada, Reno

Motivation

- Discuss assigned literature in terms of complexity leading to actual applications of PSO.
- 1. Camazine et. al. – General overview
- 2. Huth & Wissel – 2d fish schooling
- 3. Reynolds - Boids & Flocking
- 4. Kennedy & Eberhart; PSO website – Particle Swarm overview & applications

Camazine et. al.

- The fundamental question that this section seeks to answer is: “How are flocks/swarms/schools/herds able to move with such fluidity and coordination within the group?
- The answer?
  - Simple/Local interactions, not explicit organization.
- So what are the properties of the group?

Camazine (Cont'd)

- Group Properties:
  - Each Element is reacting locally
  - Preferred shapes of schools
  - Preferred distances, elevation, and bearing relative to neighbors inside the school.

Camazine (Cont'd)

- So why school?
- Prey:
  - Large groups can confuse the predator (i.e. Fountain Effect and Flash Expansion make it difficult for a predator to focus on a single target.
  - A dense clustering can prevent a predator from safely attacking (i.e. a Falcon may injure itself in the attack if it collides with anything but it's talons)

Camazine (Cont'd)

- Why School (Cont'd)?
- Predators:
  - Predators use group behavior to "corral" prey, such as:
    - Tuna herding a school between them and then surrounding the school.
    - Killer Whales herding dolphins into a tight group while single whale charge into the group.

Camazine (Cont'd)

- Other possible reasons for schooling
  - It seems that birds and lobsters flock/cluster for aero/hydrodynamic reasons
  - However, fish do not seem to achieve any hydrodynamic benefit by schooling.
- The remainder of the book deals with Huth & Wissel article

Huth & Wissel

- Goal: To roughly but generally model a group of fish forming a school using simple, locally defined rules.
- Requirements: Fish need to create realistic schooling behavior, specifically highly parallel orientation and being collision free.

Huth & Wissel (Cont'd)

The Model

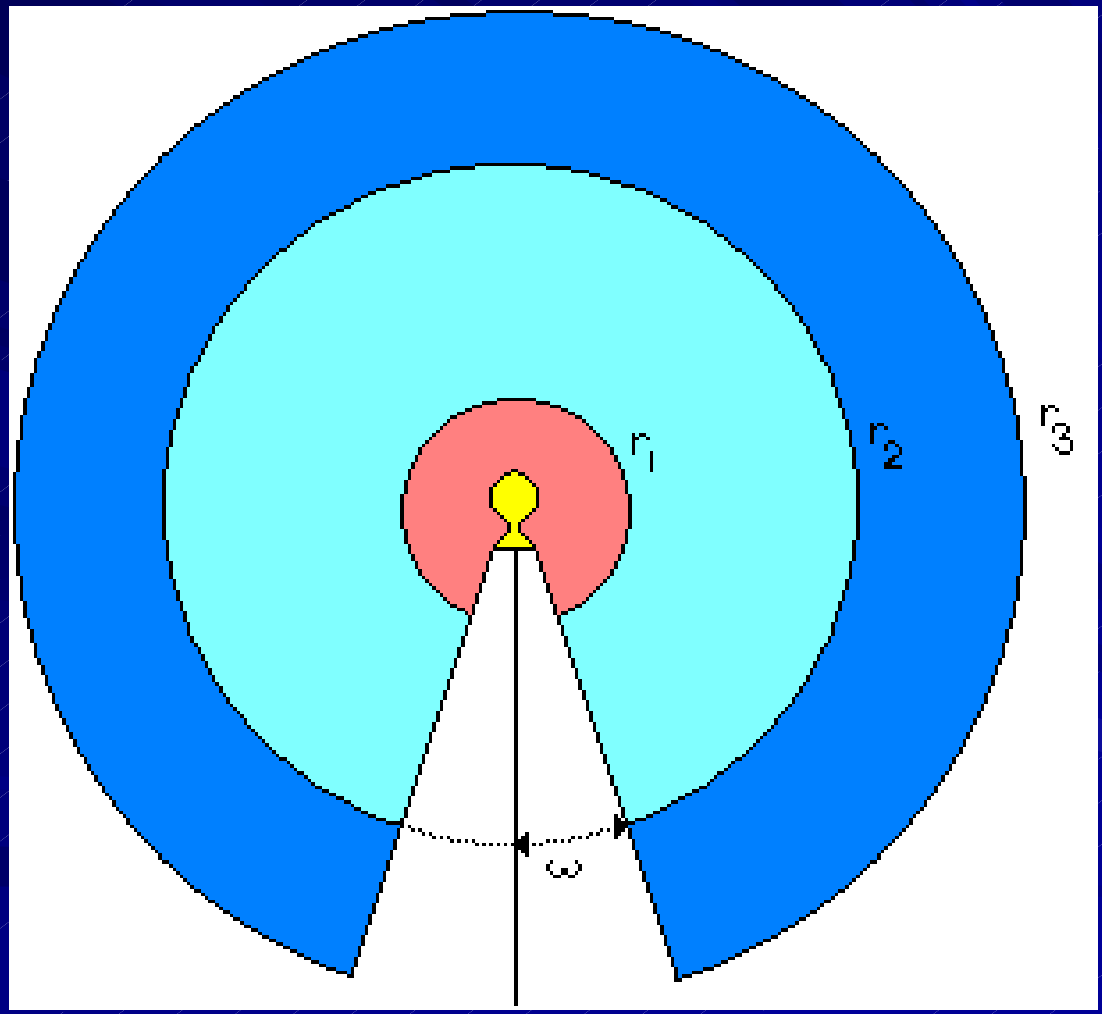
- Assumptions:
  - **All fish in the school have the same behavior rules.**
  - **No destination for the school, and external stimuli don't affect them**
    - **External Stimuli can actually "simplify the school organization because they appoint a swimming direction to the fish.**

Huth & Wissel (Cont'd)

The Model

- Assumptions:
  - Random influences are taken into account for the individual fish.
  - Motion influenced only by it's nearest neighbors
  - This is only supposed to be a simple model, and not meant to model every detail of fish behavior.

Huth & Wissel (Cont'd)



Huth & Wissel (Cont'd)

- Fish have 4 zones, where "r" is the range between fish in the center.
- Below is the notation for each of the four cases to be described below:

"i" refers to the fish in the center of the diagram, "j" refers to a different fish
"β_{ij}" is the turning angle that fish "i" will take in response to the proximity of fish "j"

$C1) = \beta_{ij} = \sphericalangle min\{\sphericalangle(v_i^o, v_j^o) \pm 90\}$ *where* $\sphericalangle(v_i^o, v_j^o) \pm 90\}$ *is difference in velocity vectors between fish "i" and "j"*

$C2) = \beta_{ij} = \sphericalangle(v_i^o, v_j^o)$

$C3) = \beta_{ij} = \sphericalangle(v_i^o, x_j - x_i)$ *where* $x_j - x_i$ *is the vector determined by the differences in the positions between the two fish.*

$C4) = \beta_{ij} =$ *chance([-180,180]) i.e. a random bearing*

Huth & Wissel (Cont'd)

- **Case 1** *(C1): $r < r_1$* is the *Repulsion Area*. If another fish enters the region, the fish in the center will take evasive action to avoid it.
- **Case 2** *(C2): $r_1 < r < r_2$* is the *Parallel Area*. If another fish is in this range, the fish in the center will attempt to match the bearings of the other fish.

Huth & Wissel (Cont'd)

- **Case 3** *(C3):* $r_2 < r < r_3$ is the *Attraction Area*. Another fish in this region will attract the fish in the center, and it will head towards the other fish.
- **Case 4** *(C4):* $r_3 < r$ or in the *Dead Area, ω* means that the fish can't "see" anyone and will begin actively searching randomly for other fish.

Huth & Wissel (Cont'd)

- In this "two-fish" model, a fish won't exactly match the bearing between itself and the other fish; rather, it will assume a random angle $\alpha_i \sim N(\beta_{ij}, \sigma)$, a normally distributed random variable with mean β_{ij} and a standard deviation of σ = 15 deg.

Normal Distribution with mean β_{ij} and variance σ^2 is $\frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{\alpha_i - \beta_{ij}}{\sigma}\right)^2}$

Huth & Wissel (Cont'd)

- Velocity is chosen *Independent!!* of other fish according to a Gamma Distribution ~ Γ(A,k) where A=3*3, k = 4. The average velocity is 1.2 Body Lengths (BL)/sec. Since k is an integer, the Gamma Distribution reduces to:

$$\left[\frac{A^k}{\Gamma(k)} e^{-Av} v^{k-1} = \frac{A^k}{(k-1)!} e^{-Av} v^{k-1}\right.$$

Huth & Wissel (Cont'd)

- Equations:

1) Position Equation
2) Velocity Equation
3) Initial Velocity Equation

$$
\begin{bmatrix}
(1)\ x_i(t+\Delta t) = x_i(t) + \Delta t\, v_i(t+\Delta t) \\
(2)\ v_i(t+\Delta t) = v_i^{o}(t+\Delta t)\, v_i(t+\Delta t) \\
(3)\ v_i^{o}(t+\Delta t) = \begin{bmatrix} \cos\alpha_i(t) & -\sin\alpha_i(t) \\ \sin\alpha_i(t) & \cos\alpha_i(t) \end{bmatrix} . v_i^{o}\, t
\end{bmatrix}
$$

Huth & Wissel (Cont'd)

- The aforementioned velocity and position vectors will work for any number of fish, but how will a fish determine it's angle of change if there are multiple fish within it's vision range?
- There are two basic ways to deal with this problem: Decision vs. Averaging.

Huth & Wissel (Cont'd)

- In the Decision Model, a fish will align itself based off of a particular neighbor using an aggregate normal distribution (i.e. summing normal distributions) where a weight function is applied giving the closest neighbors more influence. Since the total sum of the weight function is one, the result is a valid probability distribution.

Huth & Wissel (Cont'd)

- In the Averaging Model, a fish sums the collective bearings of all of it's neighbors within range, and creates a normal distribution with a mean that is the average of the collective bearings.

Huth & Wissel (Cont'd)

- The final results showed that the Decision Model created widespread confusion. The average distance between the fish in the school was never stable the average polarization was widely varied with no stability. On the other hand, the Average Model produced realistic schooling behavior with excellent polarization.

Huth & Wissel (Cont'd)

- Several variants were attempted, one removing the collision region, and another limiting the maximal turning radius that any fish could make. None of these modifications was able to make the Decision Model as accurate as the Averaging Model.

Reynolds

- Like Huth & Wissel, he wanted simulate realistic flocking behavior in computer simulations.
- Two choices for a model: Explicit Instruction or Self-Organization.
- Result – Explicit = Bad

  Self-Organized = Good

Reynolds (Cont'd)

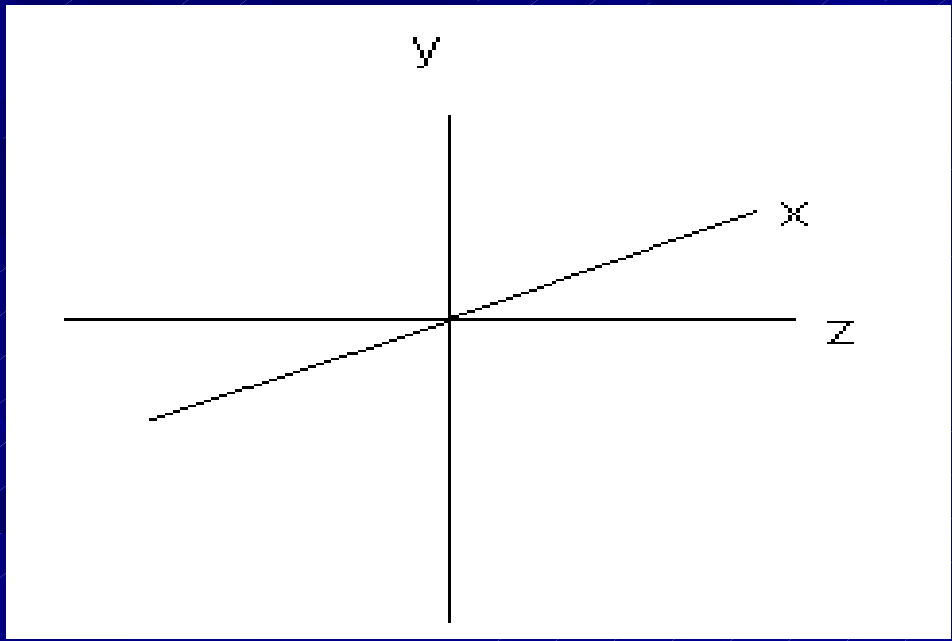
- The model that Reynolds chose to implement was similar to particle systems, except that:
  - Boids create more complex geometrical structures
  - The individuals have orientation
  - The boids interact with each other

Reynolds (Cont'd)

- Each of the elements was an *actor*, "the computational abstraction that combines process, procedure, and state."
- The elements were going to posses the geometric ability to fly. This is based on incremental translations along the object's "forward direction"

Reynolds (Cont'd)

- The Z-Axis is forwards or backwards, the Y-axis is up and down and the X-axis is right and left.



Reynolds (Cont'd)

- The model makes extensive use of the object's own coordinate system, with local space representing what each boid "sees".
- Geometric flight models conservation of momentum (i.e. objects in flight tend to stay in flight), and there is a maximal/minimal speed for each element.
- Note: Many physical forces are not supported, and gravity is only marginally applied to allow for "banking" (rotating about the Z-Axis).

Reynolds (Cont'd)

- Key traits that the model should be able to duplicate are:
  - No collisions with objects or other boids
  - No limit to the number of boids in a flock.
  - The amount of time a boid must use to "think" should be largely independent of the number of birds in the flock.

Reynolds (Cont'd)

- Therefore the three basic rules, in order of decreasing precedence are:
  1) Collision Avoidance
  2) Velocity Matching – (attempt to match velocity with nearby flockmates)
  3) Flock Centering – attempt to stay close to nearby flockmates.

Reynolds (Cont'd)

- The previous rules create a suggestion about how to "steer" the boid.
- The problem is that each bird has it's own "acceleration requests".
- The easiest way to combine these requests would be to average them (which is actually a weighted average)
  - However, this could produce "indecision" which could cause the boids to collide with each other or other objects.

Reynolds (Cont'd)

- The solution is prioritized accleration allocation.
  - The acceleration requests are considered in priority order and added into an accumulator.
  - The magnitude of each request is measured and added into another accumulator.
  - The process continues until the sum exceed a maximal acceleration magnitude. Then the last request is trimmed back to allow more pressing needs to be acted upon.
  - If all available acceleration is "used up," the less pressing behaviors might be temporarily unsatisfied.

Reynolds (Cont'd)

- A constraint of the model was that the boids should only have local perception, and not have access to the database.
  - This localized view caused the flocks converging on the centroid.
  - A result of this simulation was showing that flocking/swarming/herding *depends* upon a limited localized view of the world.
  - In this model, sensitivity to neighbors was determined as an inverse exponential of distance.

Reynolds (Cont'd)

- These rules created realistic flocking behavior.
- By including a "goal point" or "global target", the boids would actually perform to prearranged specifications (i.e. cross into the screen at this time, etc…,).

Particle Swarm Optimization (PSO)

- A form of swarm intelligence.
- If one particle “sees” a desirable path to some goal (i.e. food source, protection,etc…,), the rest of the swarm will be able to follow quickly even if they are on the opposite side of the swarm.
- Inspired by swarms of insects, schools of fish, flocks of birds, etc…,.

PSO (Cont'd)

- In particular, each particle traveling through *n*-dimensional space, has *position* and *velocity*.
- Particles "remember" the "best" position they have "seen".
- Particles also communicate this information to all other members in the swarm.
- A "Global Best" is then known to each particle (i.e. the Maximum "best" of all the particles).

PSO (Cont'd)

- This topic was developed by Kennedy and Eberhart in 1995.
- They were intrigued by the model created by Heppner.
  - Unlike earlier models, the simulated birds in Heppner's model were flocking around a "roost"—a position that attracted all the birds until they landed there.

PSO (Cont'd)

- However, an unrealistic assumption that the model made was that all the birds "knew" where to roost.
- In nature though, birds are able to quickly locate a food source or nesting site without any previous knowledge what-so-ever.
- How can this be accomplished?

PSO (Cont'd)

- The notion behind their model was that elements of the swarm would be randomly distributed about the plane. At each step, the agents would take the x and y coordinates of their position and plug them into an evaluation function:

$$Eval = \sqrt{(presentx - 100)^2} + \sqrt{(presenty - 100)^2}$$

PSO (Cont'd)

- Each agent will remember their "best" value of *Eval*, and the x and y coordinates that produced this value. This becomes the value *pbest* or "personal best"
- A key feature of this model is that the agents communicate with each other. That is, after each step, agents will compare their *pbests*, with the best value becoming the "global best", or *gbest.*

PSO (Cont'd)

- In the next step, each agent will then determine it's heading based on the value of *pbest* and *gbest*.
- Velocity is adjusted based on their location to *pbest* and *gbest*, in such a way that the agents will hopefully slow down as they approach the global maximum (depending on the value of rand()*g_increment).

PSO (Cont'd)

- In order to improve the algorithm, several other features were removed, including velocity matching, so in essence the agents were participating less like a flock and more like a swarm.
- The next step was modifying the swarm to find a target in n-dimensional space.

PSO (Cont'd)

- Later, the authors and others realized that "finding" a roost/target in n-dimensional space is the same as solving an n-dimensional equation.
- For example, the four dimensional equation below can be solved by the particle swarm. At each step, their (w,x,y,z) coordinates will be plugged in to the equation, and absolute value represents their distance from the target, which is an answer of zero.

$$5x^2 + 2y^3 - \frac{z^2}{w^2} + 4 = 0$$

PSO (Cont'd)

- The previous problem is somewhat tricky to solve with pencil and paper, but computer algebra systems exist that can solve this using mathematical techniques fairly quickly.
- But if one extends the previous problem to 25, 50, or 2,542 dimensions, then it is unlikely that the aforementioned algebra systems can solve these.

PSO (Cont'd)

- However, particle systems are only computing simple mathematical operations at each step, and so they remain theoretically capable of solving such high dimensional problems.
- Other applications for particle swarms have been in training Neural Networks at least as well, if not better (sometimes) than the traditional methods

PSO (Cont'd)

- Finally, they have also been used in optimizing genetic algorithms.
- While these are truly just approximations, they have proven themselves to be very effective at finding the global maximimum in situations where it was previously known.

Conclusion

- Once again, we have seen that simple rules defined only locally can simulate complex biological behavior.
- What is different about this topic is that there is finally an application where simple behavior can actually be programmed to achieve some goal.
- More importantly, it can do this very accurately with difficult problems.