Swarm Intelligence

presented by:

MSD86911
1. **Definition:**

   “Swarm Intelligence is a property of systems of non-intelligent robots exhibiting collectively intelligent behavior.”

2. **Characteristics of a swarm:**

   ✓ Distributed, no central control or data source.
   ✓ No (explicit) model of the environment.
   ✓ Perception of environment, i.e. sensing.
   ✓ Ability to change environment.

3. **Swarm systems are examples of behavior-based systems exhibiting:**

   ✓ Multiple lower level competences.
   ✓ Situated in environment.
   ✓ Limited time to act.
   ✓ Autonomous with no explicit control provided.
   ✓ Problem solving is emergent behavior.
   ✓ Strong emphasis on reaction and adaptation.

### Motivations

1. **Robust nature of animal problem-solving:**

   ✓ Simple creatures exhibit complex behavior.
   ✓ Behavior modified by dynamic environment.

2. **Emergent behavior observed in:**
Emergent Problem Solving

1. For Lasius Niger ants, [Franks, 89] observed:
   ✓ Regulation of 1 degree celcius range.
   ✓ Forming bridges.
   ✓ Raiding specific areas for food.
   ✓ Building and protecting nest.
   ✓ Sorting brood and food items.
   ✓ Cooperating in carrying large items.
   ✓ Emigration of a colony.
   ✓ Finding shortest route from nest to food source.
   ✓ Preferentially exploiting the richest food source available.

Stigmergy

1. Indirect communication via interaction with environment [Gassé, 59]
   ✓ Sematonic [Wilson, 75] stigmergy
     Action of agent directly related to problem solving and affects behavior of other agents.
   ✓ Sign-based stigmergy
     Action of agent affects environment not directly related to problem solving activity.

Ant Colony

1. Ants are behaviorally unsophisticated; collectively perform complex tasks.
2. Ants have highly developed sophisticated sign-based stigmergy
   ✓ Communicate using pheromones;
   ✓ Trails are laid that can be followed by other ants.
Pheromone Trails

1. Species lay pheromone trails travelling from nest, to nest or possibly in both directions.
2. Pheromones evaporate.
3. Pheromones accumulate with multiple ants using path.

Distances are 1 or 0.5 in graph.
Consider discretized intervals \( t=0, 1, 2 \).
Suppose that 30 new ants come from A to B every time unit and 30 from E to D. Each ant walks at a speed of 1 per time unit, and that an ant lays
down at time \( t \) a pheromone of intensity 1 while walking and that this evaporates completely and instantaneously in the middle of time intervals \((t+1, t+2)\).

At \( t=0 \), there is no trail yet but 30 ants are in B and 30 in D. Their choice of direction is random. Therefore, on average 15 will go to H and 15 to C.

At \( t=1 \), the 30 new ants come to B from A and find a trail of intensity 15 on the path that leads to H, laid by the 15 ants that went that way from B, and a trail of intensity 30 on the path to C obtained as the sum of the trail laid by the 15 ants that went that way from B and by the 15 ants that reached B coming from D via C. The probability of choosing a path is therefore biased; so that the expected number of ants going towards C will be double that of those going towards H. The same is true for the new 30 ants in D which came from E.

Process continues until all ants choose the shortest path. The idea is that if at a given point an ant has to choose among different paths, those which were heavily chosen by preceding ants are chosen with higher probability.

This is an example of an autocatalytic process.

The Ant System

The Ant System: TSP

The Ant System: TSP routing

Ants are agents that:

Choose next town to go to with probability that is a function of distance of town and amount of pheromone on edge.

Legal tours are “forced” by use of a tabu list; an ant can only visit a town once. Each ant has its own “tour memory”.

When the tour is complete, a pheromone is laid down on the trail.

Iteration is defined to be \( m \) moves -- one by each ant.

Tour complete in \( n \) moves i.e. \( n \) iterations.

\[ T_{ij} = \text{pheromone intensity on edge} \]

\[ b_i(t) = \# \text{ants at } i_{th} \text{ town at } t \]
(1 - \(e\)) = evaporation rate

When tour complete:

\[
T_{ij}(t+n) = e \ T_{ij}(t) + \delta \ T_{ij}
\]

\[
\delta T_{ij} = \sum \delta T_{ij}^k
\]

\[
\delta T_{ij}^k = \frac{Q}{L_k} \text{ (on route, else 0)}
\]

[Dorigo ‘92] and later...

Transition probability:

\[
p_{ij}^k(t) = \frac{[T_{ij}(t)]^\alpha [1/dij]^\beta}{N_k}
\]

\[
N_k = \sum_{k \in (S-Tabu(k))} [T_{ij}(t)]^\alpha [1/dij]^\beta
\]

\(\alpha, \beta\) are control parameters that determine the sensitivity of the algorithm to distance and pheromone.

"A pheromone is a chemical signal that triggers a natural response in another member of the same species. There are alarm pheromones, food trail pheromones, sex pheromones, and many others that affect behavior or physiology. Their use among insects has been particularly well documented. In addition, some vertebrates and plants communicate by using pheromones."

**Ant System: TSP algorithm**

1. Initialize
   
   set \(t:=0\)
   
   set \(NC:=0\) \{number of cycles\}
   
   For every edge \((i,j)\) set an initial value \(T_{ij}(t)\) for trail intensity and \(\delta T_{ij} = 0\). Place \(m\) ants on the \(n\) nodes.

2. Set \(s:=1\) \{tabu list index\}
   
   for \(k:=1\) to \(m\) do
   
   Place starting town of the \(k\)th ant in \(tabu_k(s)\).

3. Repeat until tabu list full
   
   Set \(s := s + 1\)
   
   for \(k:=1\) to \(m\) do
   
   Choose the town \(j\) to move to with probability \(p_{ij}^k(t)\)
   
   Move the \(k\)th ant to the town \(j\).
   
   Insert town \(j\) in \(tabu_k(s)\).

4. For \(k:=1\) to \(m\) do
Compute the length $L_k$ of the tour described by $\text{tabu}_k(s)$. Update the shortest tour found.

For every edge $(i,j)$

For $k:=1$ to $m$ do

$\delta T_{ij} = \delta T_{ij} + \delta T_{ij}^k$

5. For every edge $(i,j)$, compute

$T_{ij}(t+n) = e T_{ij}(t) + \bar{e} T_{ij}$

Set $t:=t+n$

Set $NC:=NC+1$

For all edges $(i,j)$, set $\bar{e} T_{ij}=0$

6. If $NC < Nc_{max}$

empty tabu lists, go to 2

else

print shortest tour; stop

**Ant System: TSP models**

**Ant density model**

✓ $\delta T_{ij}^k = Q$
✓ Increase in trail is independent of $d_{ij}$

**Ant quantity model**

✓ $d T_{ij}^k = Q / d_{ij}$
✓ Shorter edges made more desirable by making trail inversely proportional to $d_{ij}$

**Experimental studies**

30 city problem, NC = 5000 cycles

$Q$ found to be (relatively) unimportant

<table>
<thead>
<tr>
<th></th>
<th>Best parameter set</th>
<th>Average result</th>
<th>Best result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ant density</td>
<td>$\alpha=1,\beta=5,p=0.99$</td>
<td>426.740</td>
<td>424.635</td>
</tr>
<tr>
<td>Ant quantity</td>
<td>$\alpha=1,\beta=5,p=0.99$</td>
<td>427.315</td>
<td>426.635</td>
</tr>
<tr>
<td>Ant cycle</td>
<td>$\alpha=1,\beta=5,p=0.5$</td>
<td>424.250</td>
<td>423.741</td>
</tr>
</tbody>
</table>
GAs [Whitley, 89] found a tour of 424.635

**Note:**
Ant-density and ant-quantity have inferior results when compared to ant-cycle. This is due to the kind of feedback information which is used to direct the search process. Ant-cycle uses global information, that is, its ants lay an amount of pheromone which is proportional to the amount of trail whose ants whose tour was poor. On the other hand, ant-density and ant-quantity use only local information. The search is not directed by any measure of the final result achieved.

The optimal value of $r = 0.5$ can be explained by the fact that the algorithm, after using the greedy heuristic to guide search during the early stages of computation, starts exploiting the global information contained in the values delta $T_{ij}$ of trail. Ant-cycle clearly needs to be able to forget part of the experience gained in the past in order to better exploit new incoming global information.

**Ant System Parameter Sensitivity**

**Bad solutions and stagnation**
- For high values of $\alpha$ the algorithm enters stagnation behavior very quickly without finding very good solutions.

**Bad solutions and no stagnation**
- $\alpha$ too low, insufficient importance associated with trail.

**Good solutions**
- $\alpha, \beta$ in the central area (1,1), (1,2), (1,5), (0.5, 5)

**Note:**
Results consistent with understanding of algorithm:
$\alpha$ too low $\Rightarrow$ too little importance associated with the trail. High $\alpha$ means that that trail is *very* important and therefore ants tend to choose
edges chosen by other ants in the past (too little exploration of the search space).
Optimal values determined experimentally $a = 1, b = 5, r = 0.5, Q = 100$. Within range of parameter optimality, ant-cycle always finds good solutions for all tested problems.
The algorithm quickly finds good solutions (when compared to say, GAs) and does not exhibit stagnation behavior -- the ants continue to look for new and better solutions.
AS system sensitivity investigated w.r.t. problem dimensionality. Found little sensitivity with increasing problem size.
No theory currently to explain parameter settings. Parameters need hand crafting.

**Exploiting ant synergy**

**In original algorithm, all ants start from one town. Modify algorithm to distribute ants amongst nodes**
- Better than “one town” algorithm.
- Approximately $n = m$ proved optimal.
- Allow communication between ants i.e. pheromone sensing ($0 < \gamma < 1$)

**Initialization**
- Placing ants uniformly (rather than aggregated on individual nodes) resulted in superior performance.

**Employ ‘elitest’ (GAs) strategy**
- best-so-far trail is reinforced more than in the standard algorithm ($n_e - Q/L$);
- found optimal number of elitest ants.

**Note:**
(elitest ants) Below optimal value increasing it results in discovery of better tours and/or the best tour being discovered earlier. Above it, the elitest ants force exploration around suboptimal tours in the early phase of search with resulting inferior performance.
For Oliver30, $n_e = 8$ was optimal (in terms of time) although the range 2-16 could be tolerated.

**Ant System: Other heuristics**

Ant system compared with:
Tabu search (best=420, avg=420.6, sd=1.5)
SA (best=422, avg=459.8, sd=25.1)
AS (best=420, avg=420.4, sd=1.3)

NOTE: Integer distances used.
AS better than SA and more consistent than even TS.

Note:
Dorigo has also applied the ant system AS to the asymmetric TSP (distances are not symmetric \(d_{ij}\) not equal to \(d_{ji}\)). AS found to perform as well as on the basic TSP problem.
Also looked at Quadratic Assignment Problem (assigning n facilities to n locations for the purpose of flow optimization).

**Ant System: Routing Problem**

**Idea**

✓ Ants dropping different pheromones used to compute “shortest” path from source to destination(s);
✓ more flexible adaptation to failures and network congestion;
✓ use only local knowledge for routing and avoid costly communication of state to all network nodes.


**Ant System: Why Routing?**

Conventional routing often relies on:
✓ global state available at all nodes;
✓ centralized control;
✓ fixed “shortest path” (Dijkstra) algorithms;
✓ limited ability to deal with congestion or failure.

Ideally, would like to have network adapt routing patterns to take advantage of free resources and move existing traffic if possible.
[White et al, 96+] (Nortel internal only)
[Schoonderwoerd et al, 97] (*)

**Differences**
- Link cost metric constant in *
- Point to point traffic only in *
- AS Parameter settings constant in *

**Ant System: Agent types**

**Agent types:**
- explorer
  - used for route determination
- allocator
  - allocates resources in network when route emerged
- deallocator
  - deallocates resources in network at end of call

**Ant System: Point-2-Point**

For explorer agents:
- At each node, they choose path with probability proportional to $F(c_e, p_e)$.
- explorers visit nodes once only (achieved through use of tabu list).
- when destination reached, ants return along the path explored laying down pheromone trail.
- when explorers return a decision is made regarding path emergence.

**Ant System: Path Emergence**

At source node ("nest"):  
- store paths for previous m explorer agents;
- when p% follow same path allocator agent is sent to allocate bandwidth in network;
- explorer agents continue to look for new (possibly better) paths.

**Applies for one or many pt-2-pt connections:**
- Ants use different, non-reacting pheromones.

**Ant System: explorer agent algorithm**
1. Initialize
   set t:= 0
   For every edge (i,j) set an initial value $T_{ij}(t)$ for trail intensity. Place m ants on
   the source node. [Generate new explorers at freq. $e_r$ ]
2. Set $s:= 1$ { tabu list index)
   for k:= 1 to m do
   Place starting town of the $k_{th}$ ant in $tabu_k(s)$.
3. Repeat until dest’n reached:
   Set $s := s + 1$
   for k:=1 to m do
   Choose the node j to move to
   with probability $p_{ij}^k(t)$
   Move the $k_{th}$ ant to node j.
   Update explorer route cost:
   \[ r_k = r_k + c(i,j) \]
   if ($r_k > r_{\text{max}}$)
   kill explorer $r_k$
   Insert town j in $tabu_k(s)$.
   At destination go to 4.
4. While $s > 1$
   traverse edge (i,j)
   $T_{ij} = T_{ij} + p_e$
   $s := s - 1$
5. At source node do:
   if (path$_e$ = pathBuffer * d)
   create and send allocator
   if $t > T_{\text{max}}$
   create and send allocator
Evaporation occurs concurrently with exploration

**Ant System: Point-2-Multipoint**
*For j destinations, consider as j pt-2-pt connections with:

- Same pheromone, i.e. all explorers communicate;*
✓ j allocator agents only allocate bandwidth once;
✓ Allocator send decision made when % of all j explorer ants agree on spanning tree.

**Ant System: Routing Function**

*Transition probability:*

\[
p_{ij}^k(t) = \left[ T_{ij}(t) \right]^{\alpha}[C(i,j)]^{-\beta} / N_k
\]

\[
N_k = \sum_{k \in (S-Tabu(k))} \left[ T_{ij}(t) \right]^{\alpha}[C(i,j)]^{-\beta}
\]

\(\alpha, \beta\) are control parameters that determine the sensitivity of the algorithm to link cost and pheromone.

\(C(i,j)\) a function that depends upon the type of traffic, the length and utilization of the link

**Ant System: Allocator agents**

4. **Allocator agents can fail:**

✓ bandwidth already allocated by time allocator is sent;
✓ allocator agent backtracks to source rolling back resource allocation and decreases pheromone levels;
✓ decision to re-send allocator made at a later time (a backoff period is observed);
✓ explorer ants continue to search for routes.

**Ant System: Experimental Parameters (fixed)**

✓ Number of ants to create = 20
✓ Frequency of creation = every 10 cycles
✓ Amount of pheromone dropped = 10
✓ pheromone evaporation rate = 0.9
✓ Sample window size = 50
✓ Emergence criterion = 0.9

**Ant System: Results**
1. Shortest paths emerged quickly
3. Routing responds to changes in environment:
   ✓ node failure;
   ✓ link failure

4. Cost functions: \( kC(i,j) \)
   Four functions used principally:
   ✓ constant
   ✓ linear
   ✓ linear threshold
   ✓ quadratic

At high occupancy (> 50%) quadratic appeared to give the best results.
At low occupancy (<25%) non-linear was favoured.

**Ant System: Cost Fn Results**

\[ G = gsdev*Gsdev + gtime*Gtime + gcost*Gcost \]

**Ant System: Self Adaptation**

Pheromone and cost sensitivities should vary during search:
✓ avoid premature convergence;
✓ speed up search considerably.

Explorers encode sensitivity values:
✓ fitness of encoding is cost of route;
✓ new agents are created with and use genetically-manipulated values for route finding.

Each agent has its own cost and pheromone sensitivity. At the beginning, all the agents have random values. When an agent returns, its set of parameters is stored. Its parameters are linked to the cost of the path found. This cost will have the same role as the fitness function of a GA. When creating a new agent, the sets of all of the last returning agents is considered. An intermediate population of parameters is created: each set has a probability of being chosen proportional to its fitness. Random factors (mutations) are also added to the population.
Random fluctuations in range [-0.25, 3] for pheromone sensitivity; [-0.125, 1.5] for cost sensitivity. The negative values allow agents to flee the main trail and therefore explore new routes. If the routes prove to be short, the parameter settings will be retained in the returning population. Otherwise, they will die away. The adaptive system proved much more efficient than the static system. Routes which took up to 30 seconds to emerge, emerged in 2-3 seconds with adaptive settings. This approach is somewhat different to a conventional GA in that in this algorithm we are trying to avoid convergence of the population because it tends to lead to local optima. It is, perhaps, closer to the work on co-evolving populations because the environment of the agents is modified by their actions.

**Ant System: Continuous Spaces**

- [Bilchev and Parmee, 96] extended AS to continuous spaces.
- Idea to use AS as a meta-heuristic for search.
- Represent finite number of vectors from nest.
- Vectors evolve in time based upon ant fitness.