Swarm Engineering for TSP

D. Lee, S. Kazadi*, R. Goodman†

Jisan Research Institute

Abstract

Swarm engineering is the development of a swarm of cooperating agents commensurate to some criterion designed to produce a global outcome. Once the swarm has satisfied this criterion, it will accomplish the desired task. We present a theoretical formalism for understanding how swarm-based TSP optimization algorithms work. We generate a swarm criterion and show that two of the algorithms of Marco Dorigo satisfy these conditions, but to differing degrees. Finally, we present a new method of tackling the problem based on the tunneling of agents through the grid which satisfies the swarm criterion. Results on three standard TSP problems found in the TSPLIB standard library are presented.

1 Swarm Engineering

Recently, an increasing amount of work has been done on the design of multiple agent systems which carry out a variety of different tasks in a decentralized way. Among these are puck collecting systems [$Holland96$] [$Holland97$], routing systems, and various systems that solve difficult combinatorial problems with varied success.

The main shortcoming in generating these interesting results is their ability to be generalized to less intuitive tasks with less obvious methods of solution. Without a general description of what has already been done, and what properties of each solution make it successful, it is difficult to understand how to tackle new problems, or even to state clearly which problems may be tackled using this new paradigm.

Several researchers have begun making progress in this area, attempting to generate an understanding of how to attack problems using swarms, using techniques that are less anecdotal and more general than those used previously. William Spears and colleagues [$Spears99$] [$Gordon99$] have begun advocating the use of techniques based on *artificial physics*. This is a term borrowed from artificial life and refers to a brand of physics based entirely on the imagination, implemented in artificial environments that dominate "physical" artificial life work. In this paradigm, a set of behaviors are set up, which may be implemented by artificial agents using available actuators, and which must be obeyed by each element of the swarm. The approach seeks to generate a set of behaviors which may be implemented in both simulation and real robots, based on a set of rules which may be easily simulated, and later uploaded to a group of real mobile robots.

Swarm engineering takes a somewhat complementary approach. The goal of swarm engineering is identical to that of the artificial physics approach, but the method is somewhat more general. Rather than generating a behavior and observing a simulated version of the behavior unfold, swarm engineering begins with a minimal condition under which a swarm of multiple robots should be able to complete a task. The first step of swarm engineering is the generation and justification of this criterion. No general method for accomplishing this task has yet been developed; this represents an open field of research. The second part of swarm engineering centers around the generation of a behavior that provably satisfies this criterion. In general this step is more directly accomplished than the first step.

In this paper, we present the application of swarm engineering techniques to the solution of the travelling salesman problem. In Section 2, we present our swam criterion. Section 3 discusses the design of a system which satisfies this criterion. Section 4 proposes an alteration to the swarm design, and gives the results of simulations using this swarm. Finally, Section 5 offers some concluding remarks.

*Also of the California Institute of Technology.

†Of the California Institute of Technology.

2 Swarm Criterion

In general, the goal of swarm engineering is to provide a scaffolding upon which to build a set of behaviors each agent will undertake. This scaffolding should provide enough structure so that any behavior which fits naturally in this design will accomplish the task.

2.1 Initial Definitions:

One way of stating the goal of the travelling salesman problem is that one wishes to find a path through a set of locations such that the distance travelled is the smallest possible given the set of locations. The main weakness with this statement of the problem is that it does not yield to a solution using a swarm of agents. Rather, it is very much centered around the generation of a single path, a single piece of a vast puzzle.

Swarms, by definition, have an effect as a group. Thus, we must look for a second statement of the problem which is amenable to solution by swarms. We choose to restate the problem this way.

Given a set of N nodes $\{n_i\}_{i=1}^{N}$, there are $\frac{(N-1)!}{2}$ different paths possible which completely travel through each node, ending up at the beginning node again, without passing through any one node twice. There is an inherant order from shortest to longest of these paths. Thus, we seek a method of removing paths from the longest end of the list to the shortest end until the only path left is the shortest.

Thus, the group of path lengths is the interesting quantity here, and we wish to create a system under which this set of paths will evolve, eventually generating a single path which is the smallest path among those generated by the set of locations.

Given the set of nodes, we have a set of connections $\{c_i\}_{i=1}^{N(N-1)}$ between nodes. These represent one part of a possible path. Each member of the swarm might be able to travel on these connections while travelling from node to node. We may model such travel as a stochastic process in which a particular node is chosen by an agent and travelled to with some probability p_{ij}, where i represents the agent's current node and j represents the next node. The object of the swarm's interaction with the network of nodes is to modify these probabilities in such a way as to effectively (or deliberately) remove all links except those contributing to the minimum path.

We restate this formally in the next section, and it becomes our swarm condition.

2.2 Swarm evolutionary conditions and actions

Each path has a total probability of being traversed equal to

$$p_\lambda = \prod_{i=1}^{N-1} p_{i(i+1)} \tag{1}$$

where $\lambda = \left\{n_{\lambda(1)}, n_{\lambda(2)}, \ldots, n_{\lambda(N)}\right\}$. We can therefore state the following set of swarm conditions.

Swarm Criteria

1. (Initialization) $p_\lambda < p_{\lambda'}$ for all paths λ and λ' with the distance of λ larger than that of λ'.
2. (Increasing dispersion) The swarm should affect the probabilities in such a way that as $t \nearrow$, $|p_\lambda - p_{\lambda'}| \nearrow$
3. (Absolute best probability) The swarm should also affect the probability of passing over the best path. As $t \nearrow$, $p_{\lambda_{min}} \nearrow$ where $p_{\lambda_{min}}$ is the probability of the minimum path being travelled.
4. (Robust against error.) The system must be robust to sampling error.

The swarm is capable of doing two things in its travels over a network of cities and paths.

1. A swarm is capable of carrying out the travel and evaluating the path it took.
2. A swarm is capable of modifying the probabilities of the links it passes over.

The way in which modification of the probability of travelling over a given link is done may be quite general. Many researchers have done this with using a link-specific marker, many times called a pheromone. We now illustrate formally the effect of the use of a paradigm that uses a marker or other probability measure.

When travelling over a trail, a real ant will leave pheromone scattered on the trail, in response to cues both internal and external. Simulated TSP algorithms also employ a pheromone-based update method. In these methods, pheromone is typically left on the trail as a function of the distance the agent has travelled.

$$\eta = f(d) \tag{2}$$

where d is the total distance travelled by the agent. Thus, the amount of pheromone left on a particular link is given by

$$ph = N \sum_c f(d_c)\, p_c \tag{3}$$

where p_c represents the probability of a given agent travelling over a circuit c of distance d_c and N is the number of agents passing through the system. The sum is assumed to run over all circuits c containing the given link.

Let

$$Z = \sum_c p_c \tag{4}$$

We define the average inverse distance travelled over the entire set of complete paths which use the link in question

$$\left\langle \frac{1}{d} \right\rangle = \frac{1}{Z} \sum_c \frac{p_c}{d_c} \ . \tag{5}$$

Then the amount of pheromone makes

$$ph = NZ \left\langle \frac{1}{d} \right\rangle \ . \tag{6}$$

where $f(d_c)$ is $\frac{1}{d_c}$.

In actuality, pheromone merely causes a change in probability of an agent passing through a given link. Thus,

$$\frac{dp_l}{dt} \simeq g\left(NZ \left\langle \frac{1}{d} \right\rangle \right) \tag{7}$$

where again d represents the average distance travelled through the given link and g represents the function of pheromone which indicates how a given amount of pheromone translates to a given probability.

Example: For instance, this may be given by

$$\frac{dp_l}{dt} = \alpha \left(\left\langle \frac{1}{d} \right\rangle - \left\langle \frac{1}{d_0} \right\rangle \right) \tag{8}$$

where α is some proportionality constant and $\left\langle \frac{1}{d_0} \right\rangle$ is the average of the inverse distances travelled by agents through the whole system, as opposed to those travelling through the link.

Now, suppose that we have a four city TSP problem as given in Figure 1.

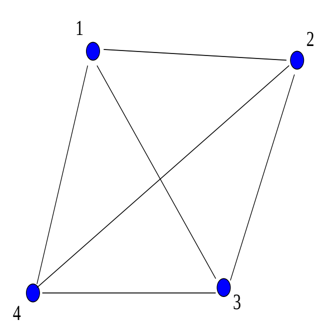

Figure 1: This figure gives an extremely small TSP problem.

We start this in the state that the probability of taking any one path between cities is equal. If this is the case, then we have the following equation for the amount of pheromone deposited on the link $1-2$

$$ph_{12} = Np_{12}\left(\sum_{jk}\sigma_{jk} f\left(d_{12jk}\right) p_{2j} p_{jk} p_{k1}\right) + C \tag{9}$$

where

$$\sigma_{jk} = \begin{cases} 1 & \text{if } j \neq k \text{ and } j,k \in \{3,4\} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

This 2-d *tensor* describes the paths which may be considered. This gives us, if the level of pheromone is proportional to the rate of change of probability

$$\frac{dp_{12}}{dt} = Np_{12}\left(\sum_{jk}\sigma_{jk} f\left(d_{12jk}\right) p_{2j} p_{jk} p_{k1}\right) + C \tag{11}$$

giving stationary solutions or solutions to this in the approximation that ps are not changing much

$$p_{12} = e^{\alpha t} \tag{12}$$

where

$$\alpha = N\left(\sum_{jk}\sigma_{jk} f\left(d_{12jk}\right) p_{2j} p_{jk} p_{k1}\right) + C \tag{13}$$

with C possibly being some evaporation term or normalization term.

The most compelling part of this study is that such a system will generally increase or decrease depending on how the terms in equation (13) work out. If f is a nonnegative function, then the probabilities will never decrease, and will saturate (become 1) quickly. The resulting system is not selective. On the other hand, if the function f is negative, the probabilities will decrease, making the set of paths impassible by an agent.

We note also that the evolution of any given probability does not depend on the probability p_{12} itself, but merely on the ensemble of paths going through it.

We choose to model our f as

$$f = \frac{1}{d} - \frac{1}{d_{ave}} \tag{14}$$

where d_{ave} is the average distance of the complete paths taken by agents moving through the grid. Thus,

$$\alpha = N\left(\sum_{jk}\sigma_{jk}\left(\frac{p_{2j}p_{jk}p_{k1}}{d_{12jk}} - \frac{p_{2j}p_{jk}p_{k1}}{d_{ave}}\right)\right) . \tag{15}$$

As long as this is negative, the link will decrease in probability. If it is positive, it will increase in probability. If there is a single path, the probabilities will remain stable.

2.3 Sampling Constraints

In general, it is very difficult, if not impossible to generate d_{ave} or d_{12}. The calculation of these quantities requires that one obtains the result for a large number of paths, comparable to the number of possible paths in the search space. Practical approximations are nonetheless approximations, and are limited by definition. Thus, any system attempting to approximate the previous design will be limited in its ability to faithfully reproduce the equations presented.

We return to this point later in this work, illustrating how our results are affected.

3 Swarm Engineering Case Studies

Dorigo et. al's seminal work on swarm based optimization initially centered around the use of computational agents he called 'ants' to solve the travelling salesman problem. This work provided both inspiration and a basis for later work on the use of ant systems in the solution of combinatorial problems. In this section, we apply our previous theory to the solutions initially obtained in an effort to begin to understand why these algorithms performed as they did.

3.1 Initial ant-based optimization.

Dorigo's initial ant-based optimization system makes use of discrete agents which travel over a graph made up of nodes and symmetric links between nodes. Each node represents a location, and each link represents the distance between nodes. Nodes are infused with a quantity of a "substance" called pheromone which influences the ant's likelihood of travelling along a given link. Ants are assumed to continually travel, keeping a record of the path they take. Ants are not allowed to retrace their path, and so cannot take a link that leads to a node they have already visited.

At each node, an ant must decide which path to take. This is done according to the amount of pheromone on each of the available links. The probability of an ant taking a particular node is given by

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_l [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} \tag{16}$$

where τ_{ij} represents the amount of pheromone on a link between node i and j, η represents a ***visibility*** parameter, the sum l runs over all links from node i, and α and β are adjustable parameters. Note that the sum of these is set to one, meaning that the ant will always take ***some path***, a condition we relax in Section 4.

Each complete path of all ants constitutes a complete iteration. At the completion of each iteration, each ant updates the pheromone levels on each link as indicated by equation (17).

$$\Delta\tau_{ij} = \begin{cases} Q/L^{k}(t) & \text{if } (i,j) \in T^{k}(t) \\ 0 & \text{if } (i,j) \notin T^{k}(t) \end{cases} \tag{17}$$

$T^{k}(t)$ is the tour undertaken by the ant at iteration t, and $L^{k}(t)$ is its length. Thus,

$$p_{ij}(t+\Delta t) = \frac{\left[\tau_{ij}(t) + \sum_{c_{ij}} \frac{Q(\Delta t)}{L_{c_{ij}}}\right]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_l \left(\left[\tau_{il}(t) + \sum_{c_{il}} \frac{Q(\Delta t)}{L_{c_{il}}}\right]^{\alpha} \cdot [\eta_{il}]^{\beta}\right)} \tag{18}$$

where c runs over all tours passing through this link. We assume that Q may be modelled as a function of time and that Q is continuous and $Q(0) = 0$. Thus, the difference between the two is

$$p_{ij}(t+\Delta t) - p_{ij}(t) = \frac{\left[\tau_{ij}(t) + \sum_{c_{ij}} \frac{Q(\Delta t)}{L_{c_{ij}}}\right]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_l \left(\left[\tau_{il}(t) + \sum_{c_{il}} \frac{Q(\Delta t)}{L_{c_{il}}}\right]^{\alpha} \cdot [\eta_{il}]^{\beta}\right)} - \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_l \left([\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}\right)} \tag{19}$$

Now, if $Q \ll \tau$ then

$$p_{ij}(t+\Delta t) - p_{ij}(t) \cong \frac{\alpha \sum_l \sum_{c_{ij}} Q(\Delta t) [\eta_{il}]^{\beta} [\eta_{ij}]^{\beta} \left(\frac{[\tau_{il}(t)]^{\alpha}}{L_{c_{ij}}} - \frac{[\tau_{ij}(t)]^{\alpha}}{L_{c_{il}}}\right)}{\sum_l \left(\left[\tau_{il}(t) + \sum_{c_{il}} \frac{Q(\Delta t)}{L_{c_{il}}}\right]^{\alpha} \cdot [\eta_{il}]^{\beta}\right) \sum_l \left([\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}\right)} \tag{20}$$

This leads us to

$$p_{ij}(t+\Delta t) - p_{ij}(t) \approx \alpha \sum_l \sum_{c_{ij}} Q(\Delta t) [\eta_{il}]^{\beta} [\eta_{ij}]^{\beta} \left(\frac{\tau_{ij}(t) [\tau_{il}(t)]^{\alpha}}{L_{c_{ij}}} - \frac{\tau_{il}(t) [\tau_{ij}(t)]^{\alpha}}{L_{c_{il}}}\right)$$

which, in the limit gives us that

$$\frac{\partial p_{ij}}{\partial t} = \alpha \frac{\partial Q}{\partial t} [\eta_{ij}]^{\beta} \sum_{l} \sum_{c_{ij}} [\eta_{il}]^{\beta} \left(\frac{\tau_{ij}(t) [\tau_{il}(t)]^{\alpha}}{L_{c_{ij}}} - \frac{\tau_{il}(t) [\tau_{ij}(t)]^{\alpha}}{L_{c_{il}}} \right) \tag{21}$$

Focussing for a moment on

$$\sum_{l} \sum_{c_{ij}} [\eta_{il}]^{\beta} \left(\frac{\tau_{ij}(t) [\tau_{il}(t)]^{\alpha}}{L_{c_{ij}}} - \frac{\tau_{il}(t) [\tau_{ij}(t)]^{\alpha}}{L_{c_{il}}} \right)$$

we can rewrite this as

$$\left(\sum_{l} \left\{ [\eta_{il}]^{\beta} [\tau_{il}(t)]^{\alpha} \right\} \sum_{c_{ij}} \left(\frac{\tau_{ij}(t)}{L_{c_{ij}}} \right) - \sum_{c_{ij}} [\tau_{ij}(t)]^{\alpha} \sum_{l} \left(\frac{\tau_{il}(t) [\eta_{il}]^{\beta}}{L_{c_{il}}} \right) \right) .$$

For simplicity, we may assign $\eta_{ij} = 1$ yielding

$$\left(\sum_{l} [\tau_{il}(t)]^{\alpha} \sum_{c_{ij}} \left(\frac{\tau_{ij}(t)}{L_{c_{ij}}} \right) - \sum_{c_{ij}} [\tau_{ij}(t)]^{\alpha} \sum_{l} \left(\frac{\tau_{il}(t)}{L_{c_{il}}} \right) \right) .$$

In principle, if the paths going through a link were long compared to other paths, the first term would tend to be smaller than the second, while if they were short, the second would tend to be smaller. This means that this system, would tend to satisfy our swarm criterion, and should lead to a system that creates a shortest path.

However in real systems, one cannot know the values of these sums accurately, and there will always be an inherant error. Thus, the system will evolve under conditions that are not quite accurately predicted by these equations. The amount of change of pheromone, then under an error will determine the stability of the system. In a real ant-based system, the amount of change will be determined by

$$\left(\sum_{l} [\tau_{il}(t)]^{\alpha} \sum_{a_{ij}} \left(\frac{\tau_{ij}(t)}{L_{a_{ij}}} \right) - \sum_{c_{ij}} [\tau_{ij}(t)]^{\alpha} \sum_{l} \left(\frac{\tau_{il}(t)}{L_{a_{il}}} \right) \right) .$$

where a represents the agent in the system over which the sums are actually taken. Now, if a long path has an initially large estimate of the first term, τ will increase along that link. This increase will in turn bias the ants from the second term to the first term, and if this bias is large enough, will tend to cause a link that should not increase to increase in probability, locking in a poor path. Indeed, this is precisely what is reported. Thus, although the system obeys our first three swarm engineering criteria, it fails to satisfy the fourth, which in practice significantly reduces the capability of the system.

3.2 Evaporation

Dorigo et. al. propose a modification to the system which includes the use of pheromone evaporation. This is an attempt to provide a restorative force which will satisfy criterion 4. In this case, we have

$$p_{ij}(t + \Delta t) = \frac{\left[\tau_{ij}(t)(1-\mu) + \sum_{c} \frac{Q(\Delta t)}{L_c} \right]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l} \left[\tau_{il}(t)(1-\mu) + \sum_{c} \frac{Q(\Delta t)}{L_c} \right]^{\alpha} \cdot [\eta_{il}]^{\beta}} \tag{22}$$

Here, again, we may assume that μ is a function of the change in time Δt which goes to zero when Δt goes to zero. Then

$$p_{ij}(t + \Delta t) - p_{ij}(t) = \frac{\left[\tau_{ij}(t)(1-\mu) + \sum_{c_{ij}} \frac{Q(\Delta t)}{L_{c_{ij}}} \right]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l} \left(\left[\tau_{il}(t)(1-\mu) + \sum_{c_{il}} \frac{Q(\Delta t)}{L_{c_{il}}} \right]^{\alpha} \cdot [\eta_{il}]^{\beta} \right)} - \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l} \left([\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta} \right)} \tag{23}$$

if again $Q \ll \tau$ then

$$p_{ij}(t+\Delta t) - p_{ij}(t) \cong \frac{\alpha \sum_l \sum_{c_{ij}} Q(\Delta t) [\eta_{il}]^\beta [\eta_{ij}]^\beta \left(\frac{\tau_{ij}(1-\mu)[\tau_{il}(t)]^\alpha}{L_{c_{ij}}} - \frac{\tau_{il}(1-\mu)[\tau_{ij}(t)]^\alpha}{L_{c_{il}}} \right)}{\sum_l \left(\left[\tau_{il}(t)(1-\mu) + \sum_{c_{il}} \frac{Q(\Delta t)}{L_{c_{il}}} \right]^\alpha \cdot [\eta_{il}]^\beta \right) \sum_l \left([\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta \right)} . \tag{24}$$

In this case, the damping term depresses the change in probability, controlling the random fluctuations. This allows the system to be tuned with an appropriatly chosen μ to a rate of change in probability that is small enough to manage and to relieve random fluctuations. Dorigo et al. report that this is sufficient to limit the effects of random fluctutation. However, there is a tradeoff between the strength of the evaporation and the effectiveness of the algorithm. If the evaporation term is too large, the denominator in equation (3.9) will also become small, making the size of the fluctuation increase. If this is the case, the system will never be able to lock into the desired minimum distance, but rather will continue to endlessly fluctuate.

4 Probabalistic Agents

The main weakness of ant-based systems is that they are vulnerable to random effects. This has led to a variety of different algorithms designed to minimize these effects. The weakness stems from the fact that the agent must make a decision about which path to take at every iteration. This is in contrast to first making a decision about whether or not to take a link in the path at all.

One might imagine a *geddanken* in which electrons are fired through an absorptive medium from each of the cities toward each of the other cities. If one were riding along with these electrons, one would note that the number of companion electrons travelling with the electron providing the ride would decrease as an inverse exponential of the distance. Thus, we would have

$$N = N_0 e^{-\lambda d} \tag{25}$$

where λ is some characteristic distance, and d is some distance. Were one to send the electrons along to the next city, one would find a similar phenomenon occurring. This would happen again and again, until the final electrons arrived at their destination, with numbers reflecting the distance travelled. Thus,

$$N = N_0 e^{-\lambda d_t} \tag{26}$$

with d_t representing the total distance travelled.

We choose as model agents for our swarm agents as improved 'electrons'. Each agent has a probability of travelling from one node to another given by equation (25). At each node, each agent duplicates itself so as to provide enough total agents to travel to each node not travelled to on the path it has already taken. Each new agent will then pick up where the first agent left off, travelling to the new nodes in the same way as the previous agent did in the last iteration. Note that if one agent is initialized at each node, and the probability travelling is 1 (λ is 0), each possible path will be travelled. The behavior of an agent and its duplicates is depicted in Figure 2.

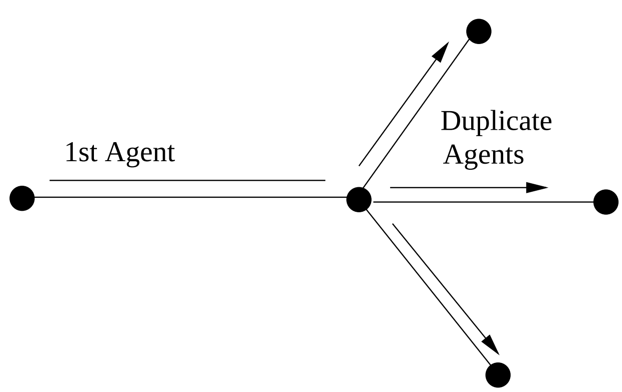

Figure 2: This depicts two parts of one iteration of the agent's motion on a map.

Note that in the first part of the iteration, one agent exists. In the second part, three agents exist. The number of agents equals the number of possible paths one might take while going through the first link.

It is straightforward to verify that the paths travelled by the agents in this system occur with an exponentially decreasing probability commensurate with equation (26). We illustrate this in Figure 3,

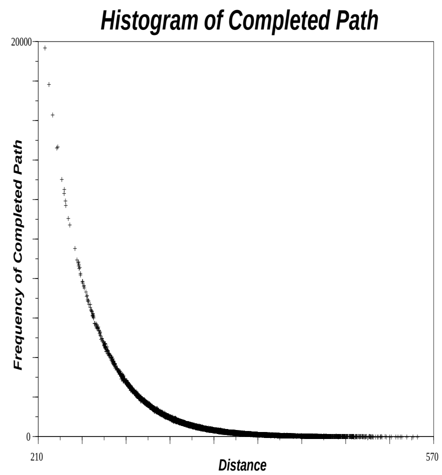

Figure 3: This illustrates the distribution of path lengths obtained from agents which completed a tour of ten randomly placed nodes.

in which 10^5 iterations of a sample graph have been run.

This method has two main flaws. Any attempt to handle large graphs immediately causes one of two outcomes:

1. A virtual explosion in the number of agents in the intermediate stages of the search.
2. No agents completing tours, as a result of the extremely low probability of any agent completing a tour given the low pairwise probabilities required to limit the explosion of agents.

Moreover, the method is not a swarm at all. Indeed, it uses the predefined system to find the smallest path, using a large number of trials only to verify that the correct path has indeed been found.

A swarm has been defined [$Kazadi00$] as well-defined affector of a given system in which the agents have the ability to modify the system they are in, other agents' behaviors, or both. We employ a swarm-based method which derives from these probabilistic agents. We initialize the system with one agent. This agent has a predefined initial path $\{a_1, ..., a_N, a_1\}$ with some distance d_a, which becomes the first estimated minimum distance.

All probabilities are initialized according to

$$p_{ij} = e^{-\lambda d_{ij}} \ . \tag{27}$$

This biases the algorithm towards a small number of completing paths, with a small average distance. At each iteration, this probability is reduced according to

$$p_{ij} \mapsto \upsilon p_{ij} \tag{28}$$

where υ is some positive number smaller than 1.

In order to generate new tours for sampling, the current tour is shifted to the left by M elements, i.e.

$$\{a_1, \ldots, a_N, a_1\} \mapsto \{a_{M+1}, \ldots, a_N, *, *, \ldots, *\} \tag{29}$$

where $*$ indicates an unassigned node in the tour. The last M elements are chosen according to the guidelines given for probabalistic agents above, yielding many new paths in the process. Thus, the probability istribution of paths resulting from this method is

$$\{p_d | p_d = p_1 \cdots p_M\} \tag{30}$$

where the probabilities are the given according to choices for last elments. At every step, the new paths become truncated and duplicated, yielding a small number of new final paths.

If we define d_0 to be the current estimate of the minimum distance, we denote the update of transition probabilities by

$$\frac{dp_{ij}}{dt} = \alpha\left(\frac{1}{d_{e_{ij}}} - \frac{1}{d_0}\right) \tag{31}$$

where $d_{e_{ij}}$ is given by

$$d_{e_{ij}} = \left(\sum_c \left[\frac{1}{d_c}\right]\right)^{-1} \tag{32}$$

and the sum is over all tours c which travel through a given link. Let us examine this in terms of our swarm criterion.

- The initial state is exactly as desired in the swarm engineering criterion.
- As more iterations pass, two types of updates occur. First, the probabilities are updated by multiplying a factor (for instance 0.99) to all link probabilities. This lowers the probability of the group of paths. Next, when an agent completes the path, the links along that path are updated by adding the multiplying by quantity $\exp\left(\frac{1}{d}\right)$ to each of the link probabilities along the path. This increases the probability of going on the link, thereby increasing the probability of going along any small distance path going through this link. This satisfies criterion 2.
- The probability of visiting a nearby link is limited by a small number of links, indicating that the previous problems of both explosion and low probability have been solved. Moreover, as shown above, the smaller paths will be visited more often than the larger paths. Thus, those paths with smaller portions contributing will be selected more often than those with large portions selected.
- The longer distance paths have a decreasing probability which decreases at a larger rate than the shorter distance paths. The links contributing to these paths will have a smaller number of signals completing, and a smaller positive controbution per signal. Thus criterion 3 is satisfied.
- As the best path is always of distance smaller than or equal to the current estimate d_0, the probability of this and all other paths decreases steadily. However, the system is reinitialized with the new estimate of the minimum distance, increasing the rate of decrease of long paths, but holding steady that of the minimum distance path.
- Finally, errors in sampling are of a different kind than previous errors. It is unlikely to lock in a path that is suboptimal, unless the rate of probability alteration is extremely high, as new paths may then be found easily which are superior. These will lead to a reduction in the relative probability of travelling over the suboptimal path. Since

  $$p\left(t+\Delta t\right) = p\left(t\right) + \Delta\alpha\left(\frac{1}{d_{comp}} - \frac{1}{d_{0_{est}}}\right) \tag{33}$$

  in practice, it is impossible have an estimated minimum distance $d_{0_{est}}$ smaller than the actual minimum. Thus, the probability of all paths will initially decrease, with the rate of decrease changing with each new better distance. The relative increases in rate of probability decrease is different for links of differing average distance, with those links which contribute to a small total path having a smaller acceleration than those links contributing to large paths.

We test these on three paths given in the TSPLIB, a standard set of travelling salesman test problems. In each case, the initial probabilities are given by $p_{ij} = e^{0.001 d_{ij}}$ and the initial path is $\{1, 2, \ldots, N, 1\}$ where the path has N nodes. In each case, the minimum is found with sufficient time. As a comprehensive analysis of the performance is beyond the scope of this paper, we present only several minimum distances found on small paths. The results are given in Table 1.

TSP Problem	Minimum Distance	Distance Obtained	Iterations (Average)
Burma (14)	3323.0	3323.0	47.7
Ulysses (16)	6859.0	6859.0	1000.2
Ulysses (22)	7013.0	7013.0	2194.2

Table 1: This table gives the performance of the swarm based on probabilistic agents on a set of three standard TSP test problems[1].

In each case, the minimum distance was obtained, indicating that the swarm criterion illustrated above does indeed produce the desired outcome.

Thus, the use of this type of swarm-based computation would seem to be promising, as reliable algorithms may be constructed simply by conforming to a given criterion.

5 Conclusions

In this paper, we have examined the design of a swarm-based optimization system using a rational technique to initially design a swarm criterion. Swarms are then engineered based on this criterion. Those that satisfy this criterion are expected to produce a desired final outcome for the system, while those that do not are not expected to be able to complete the task in a satisfactory and robust manner.

Investigating some of the previous work in this light has illustrated that, although most of the swarm criterion imposed for the TSP are satisfied, they are not robust under the effect of noise. Addition of evaporation served to limit the amount of noise available for changes in probability. However, an evaporation rate that is too high would seem to produce paths with great variability, as is observed in practice.

This work represents a first attempt at a general procedure for producing swarms of a desired behavior. The correct development and application of such techniques would seem to be a necessary step for creating a process whereby swarms can be applied to practical problems and used to produce useful results. Further work will address the developement of a swarm-based way of guaging efficiency, with the hope for such work lying in the generation of swarm actions of a particular efficiency.

References

[Dorigo96] Dorigo M., Maniezzo V., and Colorni A. (1996). *The ant system: optimization by a colony of cooperating agents.* **IEEE Transactions on Systems, Man, and Cybernetics** 26: 29-41.

[Dorigo97] Dorigo M. and Gambardella L. (1997). *Ant Colonies for the traveling salesman problem.* **Biosystems**. 43: 73-81.

[Gordon99] Gordon D., Spears W., Sokolsky O., and Lee I.(1999). *Distributed Spatial Control, Global Monitoring and Steering of Mobile Physical Agents.* **Proceedings of IEEE International Conference on Information, Intelligence, and Systems**. November, 1999.

[Holland96] Holland, O. *The use of social insect based control methods with multiple robot systems.* **UKACC International Conference on Control '96**. 1996.

[Holland97] Holland O. and Melhuish C. 1997. *An interactive method for controlling group size in multiple mobile robot systems.* **International Conference on Advanced Robotics**, Monterey, CA; July 7-9, Hyatt Regency Hotel.

[Kazadi00] Kazadi, S. (2000) **Swarm Engineering**. Phd thesis, in preparation.

[Spears99] Spears W., and Gordon D. (1999). *Using Artificial Physics to Control Agents.* **Proceedings of IEEE International Conference on Information, Intelligence**, and Systems. November, 1999.

[1]The average number of iterations for the Ulysses TSP algorithm is somewhat artificially low, as the runs were terminated at 10000 iterations, and these averages reflect the completing runs.