MAPPING TRAFFIC FLOW FOR TELEMETRY SYSTEM PLANNING

Grant Rivera (Student) and Kurt Kosbar (Advisor)
Telemetry Learning Center
Department of Electrical and Computer Engineering
Missouri University of Science and Technology

ABSTRACT

Telemetry receivers must typically be located so that obstacles do not block the signal path. This can be challenging in geometrically complex indoor environments, such as factories, health care facilities, or offices. An accurate method for estimating the paths followed by typical telemetry transmitters in these environments can assist in system planning. It may be acceptable to provide marginal coverage to areas which are rarely visited, or areas which transmitters quickly transit. This paper discusses the use of the ant colony optimization and its application to the telemetry system planning problem.

Keywords: System and Mission Planning, Medical Telemetry Applications, Path Planning, Ant Colony Optimization.

INTRODUCTION

Many telemetry systems use multiple receivers to collect data from one or more mobile transmitters. In environments with reasonably simple geometry or few obstructions, selecting a location for the receivers can be rather straightforward. It becomes a bit more challenging when transmitters and receivers are both near ground level, where terrain other obstacles may block, attenuate, reflect, diffract or scatter the signal. Geographic information systems can help evaluate the system design in these cases.

However, there are a number of indoor telemetry applications, where the environment is geometrically complex. Typical examples include health care facilities, offices and factory floors. These areas may require a large number of telemetry receivers to obtain 100% coverage. In spite of that, complete coverage in many cases may not be required or may not be economical. There may be areas where transmitters are rarely located or through which they transit reasonably quickly. Depending on the application, it may be acceptable to either tolerate an interruption in the data transmission, or install buffers and interweavers in the transmitters so the data can be recovered after one of these interruptions, or deep fades.

To perform system level planning in these applications, it will be necessary to have a geometric description of the environment, an understanding of the transmitter and receiver performance

characteristics, a reasonably accurate channel model, and a description of the typical paths followed by the telemetry transmitters. This last item is the focus of this paper.

In many indoor applications, a telemetry transmitter may spend an extended time at one location. In health care settings, one might expect a transmitter to remain in a patient room or examination room for an indefinite period. In a factory setting, an area used for storage may have transmitters which remain stationary for an extended length of time. The system may need to insure that these areas are well covered by telemetry receivers.

However, there may be areas where one expects transmitters to quickly pass. A hallway or moving assembly line may be locations where a telemetry transmitter rarely remains for more than a few seconds. And within these areas, there may be locations a transmitter almost never occupies. For example, if a moving vehicle needs to negotiate a curve or corner – the speed of the vehicle will limit how close it can come to a wall or other obstacle.

The goal of this project is to develop a software, or mathematical, model which can accurately predict the movement of telemetry transmitters in an indoor environment. There will be a behavioral model, which indicates that transmitters typically remain in some areas for an extended period. Then once the transmitter begins to move, it will do so because of desire to reach a particular destination. The goal of this project is to predict the paths which the transmitter will most likely follow, in addition to modeling the number of transmitters in these areas.

Once the behavior of the transmitters is understood, it will be easier to determine the most economical locations for the telemetry receivers. In addition, the system designer, or the system design tools, can estimate the length of time the telemetry length will typically be interrupted.

A variety of algorithms may be suitable for this application. The remainder of this paper looks at the usefulness of Ant Colony Optimization (ACO) and some of its relatives such as the Elitist Ant System (EAS), Rank-Based Ant System (AS_{rank}), and Max-Min Ant System (MMAS) for this problem. Other algorithms which could be used, but which are not described here are rapidly-exploring random trees (RRTs)[2-4], memetic computing (MC)[5], and other versions of ACO such as fast ant system(FANT)[1], approximated non-deterministic tree search (ANTS)[1] , and ACO-GA[6-8], and ACO with a potential field[9].

ANT COLONY OPTIMIZATION

One way to perform path planning is by using the Ant Colony Optimization (ACO) algorithm. As the name implies, this approach simulates the behavior of a colony of ants. These insects are able to accomplish very complex tasks that a single ant could not, because of their social interactions. Some of these behaviors include division of labor, cemetery organization, brood care, and construction of nests [1]. However, the primary interest for this work is the colony's foraging behavior.

Foraging ants are able to find the shortest route to a food source from their nest. These ants randomly search until a food source is found and then deposit pheromone from the food back to the nest. This pheromone trail attracts other wandering ants to the food source. Gradually, the searches become more organized, and eventually, all the ants will converge on the shortest path.

Dorigo [1] developed an algorithm to simulate this foraging behavior, the Ant System (AS), in 1992. The first step in the AS algorithm is to construct a search graph. This graph consists of a series of interconnected nodes. The ants may traverse the graph from node to node, only by passing over the interconnecting paths. One of the nodes is designated as the nest, where ants will return food, and a limited number of other nodes are designated to contain food.

Next, an initial pheromone is placed on all interconnects. The pheromone level must be chosen carefully. If the initial value is too low, then once the first ant finds a food source, all the other ants will quickly converge upon the path that particular ant used. This path will most likely not be the shortest available. In contrast, since ants only deposit a small amount of pheromone, if the initial value is too high, then the deposited pheromone trails to the food would have close to the same amount of pheromone that an unexplored area has. Thus, an extensive search will continue until the initial pheromone levels decrease with time, or evaporate, enough to allow the ants to recognize the path to the food source.

To optimize the rate of convergence, the initial pheromone intensity should be slightly higher than the estimated amount of pheromone deposited by the ants in one iteration. This can be determined by the following equation:

$$\tau_0 = m / f^*$$

Where:

- τ_0 is the pheromone intensity
- m is the number of ants
- f^* is an estimation of the optimal value, or distance, of the shortest path for the function. (For some complicated problems, it might be too difficult to estimate a value. Therefore, trial and error with logical guesses a little on the larger side (better to be too large and waste computational time exploring than be too small and receive bad values) can be implemented for a few test runs. This value should then be replaced with the shortest distance recorded from those tests).

The ants start from their nest exploring the search space for an optimum solution. The ants use a probabilistic selection when deciding which direction, or node, to visit next. They take into consideration the quantity of pheromone and some heuristic information to choose their path. In the traveling salesman problem (TSP), the heuristic information is the reciprocal of the distance between nodes. Since all nodes are equally space for the telemetry system planning problem, the heuristic information is the reciprocal of the distance from the current node to the food location, not to the next node as in the TSP.

Pheromone is placed on every path each ant takes. When an ant chooses a certain path, the quantity of pheromone of that trail is increased and more ants will be attracted to that route. Over time, the pheromone level decreases, or evaporates, which causes the attraction to that path

to be weakened. Eventually, if ants don't follow a particular path to replenish its pheromone, the path will disappear. This allows the colony to essentially discard unproductive paths. After all ants have completed their tour, a fitness parameter is calculated for each ant. There are two ways to evaluate this fitness: to allow only feasible solutions or to allow infeasible solutions but with a penalty.

Problems typically have some sort of criteria or constraint. Generally, the infeasible solutions with penalties are used in complicated problems in which it is difficult to stay within the constraints. Instead of deleting an entire infeasible path and wasting all of that computational time, the path is kept, but the fitness value receives a penalty. The penalty ensures that the ants will not consider this to be the best path found, but it also allows the ants to better explore that path to determine if there is something else that can be done to not be penalized.

Infeasible solutions are also used for problems in which solutions found outside of the constraints would produce a poor fitness value even without a penalty. In this kind of problem, the ants generally won't step out of the desired bounds, and even if they do, it will be reflected in their fitness. The ants will therefore learn to stay within the constraints so there is no reason to waste time in trying to prevent this from happening. For mostly every other kind of problem, only feasible solutions are allowed.

Once the fitness of each ant has been calculated, the pheromone is evaporated, and then, an additional amount of pheromone is deposited only on the interconnects that the ants traveled. The amount of pheromone added to these interconnects depends on the fitness values as explained in *Implementing ACO*. The algorithm is then terminated when an acceptable predefined solution has been found, all of the ants have converged, or the predefined maximum number of iterations has been reached.

An Elitist Ant System (EAS), Rank-Based Ant System (AS_{rank}), and Max-Min Ant System (MMAS) can also be used. The EAS is the same as the AS algorithm but provides additional pheromone to the interconnects that belong to the global-best ant's tour. This will typically lead to a better solution in a lower number of iterations than AS. In AS_{rank} algorithm, ants deposit an amount of pheromone that decreases with rank in which the global-best ant always deposits the most. For MMAS, the only ants that can deposit pheromone are the global-best ant or the iteration-best ant.

Using only the global-best ant can lead to premature convergence in MMAS. The iteration-best ant allows for more exploration. Therefore, a combination of the two can be used to allow for exploration and exploitation by having the global-best update periodically in which the frequency of the updates gradually increases over a number of iterations. These methods of updating can cause all the ants to converge to one trail quickly which will usually lead to a sub-optimal solution. Therefore, a min/max limit of how much pheromone a trail can have is predetermined. The initial pheromone trails are set near the upper pheromone limit with a small evaporation so that there is a large exploration of the search space at the beginning. The pheromone trails are reinitialized when there have not been any improved tours over a certain number of iterations or when the pheromone trails exceed the min/max limit. [1].

IMPLEMENTING ACO

The search graph for the ACO algorithm is based on a physical environment. This environment will typically have obstacles which the ants must avoid, or at least would prefer to avoid. There are a variety of ways to describe obstacles, but for simplicity this work will use a regularly spaced grid, where each node is assigned a binary value. If a node has a value of zero, then it will be considered an obstacle which no ant can traverse. A node with a value of one will be a path that ants are allowed to travel. The number of nodes in the grid must be chosen carefully, since a small grid may not be sufficient to model a particular environment, while a large grid will result in very long simulation execution times.

Once the grid size is selected, and depending on if there are more free spaces or obstacles, the matrix should be initialized with either all ones or zeros respectively. The other will then be entered in specific locations to create the obstacles or free spaces of the search space. The outer perimeter of the terrain matrix should be set as an obstacle.

The next step is to initialize the parameters: population size, initial pheromone intensity, pheromone decay rate, pheromone weight, heuristic weight, and max number of iterations. These parameter values are all chosen based on the goals and the problem. If these values aren't correctly optimized, then the ants might not find the optimal solution, and the ants might not all converge to the global-best ant's path. Generally, trial and error or an optimizing algorithm is used to optimize these values. The current global-best ant's fitness should also be initially set if EAS, AS_{rank}, or MMAS algorithm is used. This value should be initialized at infinity since no exploration has taken place, and therefore, no fitness has been calculated.

The pheromone is initially placed on all possible interconnects. A matrix is used for this representation. The rows and columns of the pheromone matrix correspond with the rows and columns of the search graph matrix. This is done by multiplying the terrain grid matrix by the initial pheromone intensity. For this very reason, the obstacles were represented with zeros so that there would be a pheromone value of zero at their locations and all the clear spaces would have a pheromone value equal to the initial pheromone intensity value.

For the tour construction, an ant completes one full tour before the next ant travels. It's important to set another pheromone variable equal to the current pheromone matrix before each ant starts its tour because this new pheromone variable, τ, will be changed after each step an ant takes. The ant chooses its next step based on the probability of the surrounding matrix locations.

$$p_{ij} = \frac{\left(\tau_{ij}^{\alpha}\right)\left(\eta_{ij}^{\beta}\right)}{\sum\left(\tau_{iu}^{\alpha}\right)\left(\eta_{iu}^{\beta}\right)}$$

Where:

- p_{ij} is the probability of traveling from i to j
- i is the current node
- j is the next node

- τ is the quantity of pheromone
- η is the heuristic information, $\eta_{ij} = \frac{1}{d_{if}}$
- f is the food location
- α is the pheromone weight
- β is the weight of the pheromone heuristic information.
- u is the surrounding nodes

If α is too low or set to zero, then little to no pheromone information is used [1]. The probability will be based only on the heuristic information. This sort of greedy search can also happen if β is too large of a number. Likewise, if β is too small or zero, only the pheromone information is used and the attraction to the food location will be weakened or negated. The ants will randomly explore the search graph until the food location is found. This usually takes more computational time and leads to sub-optimal solutions.

Now that a probability value is calculated, there needs to be a way to choose the next node based on that information. One way to do this is to place the probabilities of each node in an array. The last node value in a new array should equal the sum of all the values in the original array which will always equal one. The next node under consideration is the second to the last node. This node should be the sum of all the probability values except for the last one in the original array unless the probability of this node is zero. If zero, the probability for this node will equal zero. This will continue until all nodes have been assigned a probability value in the new array.

For example, given the 9 probability values of [0.05, 0.1, 0.05, 0.2, 0, 0.3, 0.1, 0.1, 0.1]; the new array would be [0.05, 0.15, 0.2, 0.4, 0, 0.7, 0.8, 0.9, 1]. In the simulation, a random number generator is used to produce a number between 0 and 1, and then a search is performed to find the first node of the new array that has a value above that level. In the previous example, if a random number of 0.4789 is generated, then the ant should move to the node that corresponded to the output value of 0.7.

Once the ant moves to its next location, the τ value of that location should be set to zero, to prevent the ant from choosing to stay at its current position when using the probability equation, and to prevent revisiting that node at a later time. However, this can cause the ant to sometimes trap itself with a probability of zero for all surrounding nodes. If this happens, τ should be re-initialized to the current pheromone matrix. If a position is revisited, then all steps from the first instance to the last repeated position needs to be removed since those steps didn't help the ant get any closer to its goal. Once all ants have completed their tour, the fitness of each ant is computed using the Euclidean distance formula:

$$L = \sum_{k=1}^{n} \sqrt{\left(r_f(k) - r_i(k)\right)^2 + \left(c_f(k) - c_i(k)\right)^2}$$

Where:

- L is the total length of the tour
- n is the total number of steps
- $r_f(k)$ is the final row value for step k

- $r_i(k)$ is the initial row value for step k
- $c_f(k)$ is the final row value for step k
- $c_i(k)$ is the initial row value for step k

The ant which traveled the least overall distance of a complete tour (has the smallest fitness value) will become the iteration-best ant. Next, all pheromone will be decayed by the pheromone decay rate.

$$pheromone = (1 - decay\_rate) * pheromone$$

Pheromone will then be deposited only to the interconnects the ants traveled to by adding the reciprocal of the fitness to those pheromone positions.

$$pheromone(ant's\_path) = pheromone(ant's\_path) + \frac{1}{fitness(ant's\_path)}$$

The AS algorithm terminates once an acceptable solution has been found; when all the ants have converged; or if a predefined maximum number of iterations have been reached. The best path is then plotted for the user's inspection. However, for the EAS, AS_{rank}, and MMAS, the iteration-best ant's fitness is compared to the global-best ant's fitness at the end of each iteration. If the iteration-best ant has a smaller fitness value than the global-best ant, then that ant becomes the new global-best ant and will be plotted when the algorithm terminates.

SIMULATION RESULTS

The results from some sample searches are illustrated in Figures 1-3. The EAS algorithm was used for all the figures. Only feasible solutions were allowed when determining the fitness. As seen from Figure 1, if the actual pheromone is used instead of using τ which prevents revisits, and if all of the repeated positions of the path are remembered, then all of the steps, regardless of how useful they are, are deposited with pheromone. The revisited positions are also given extra pheromone every time the ant traveled to it even if it was a "bad" move or even if it only needed to be traveled to once. Figure 2 shows what the solution is with the use of τ. It is also important to note that not only was the path "bunched up" in certain spots but also, because of the extra pheromone deposits in Figure 1, it found a worse path compared to Figure 2.

Figure 3 demonstrates what happens when the EAS algorithm needs to be optimized better. This path was caused by the heuristic weight being too large. As mentioned earlier, if β is too large then the heuristic information guides the ants in a greedy search. In this case, the ants can't search far enough to the right because the heuristic information is giving it too small of a probability each time. Even after an extended simulation run time, the first ant could not move past this general spot while searching for the goal. Since this ant never found a solution, the other ants were never able to explore the terrain.

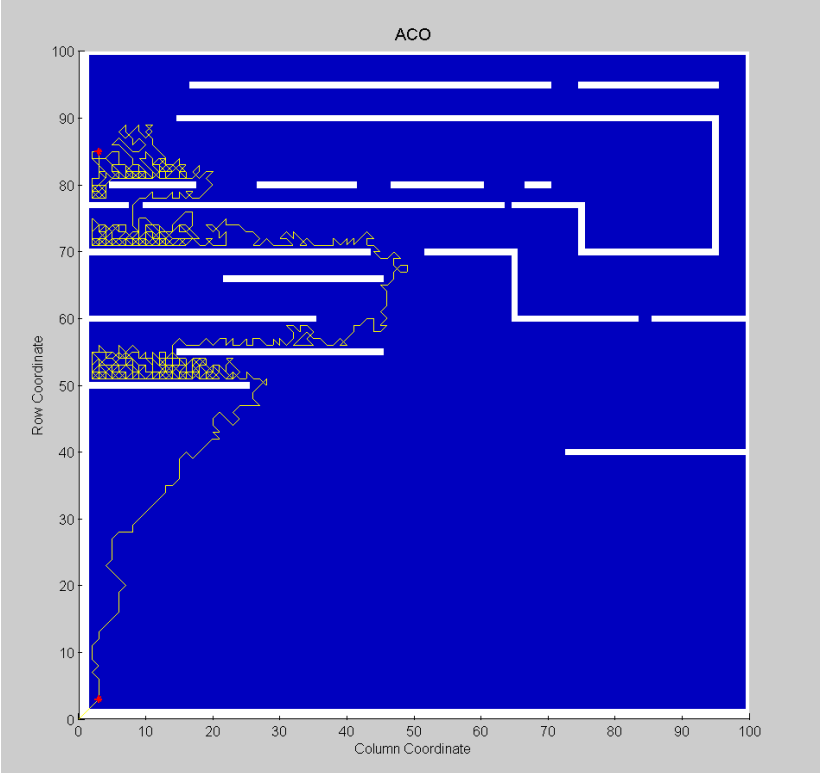

Figure 1. Inefficient Path from Low τ Value

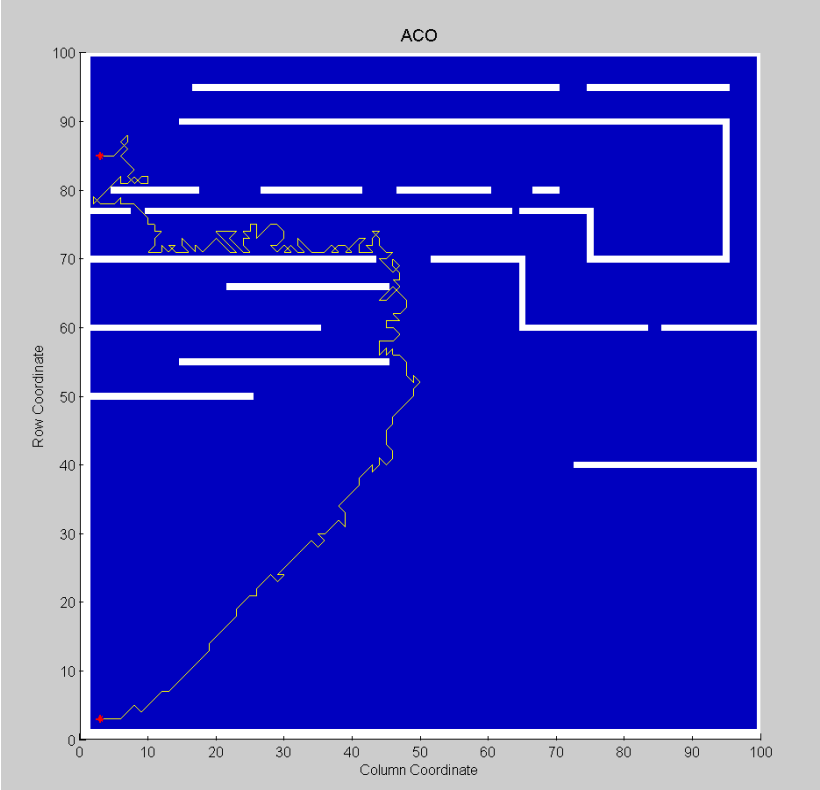

Figure 2. More Efficient Path by Using Appropriate τ Value

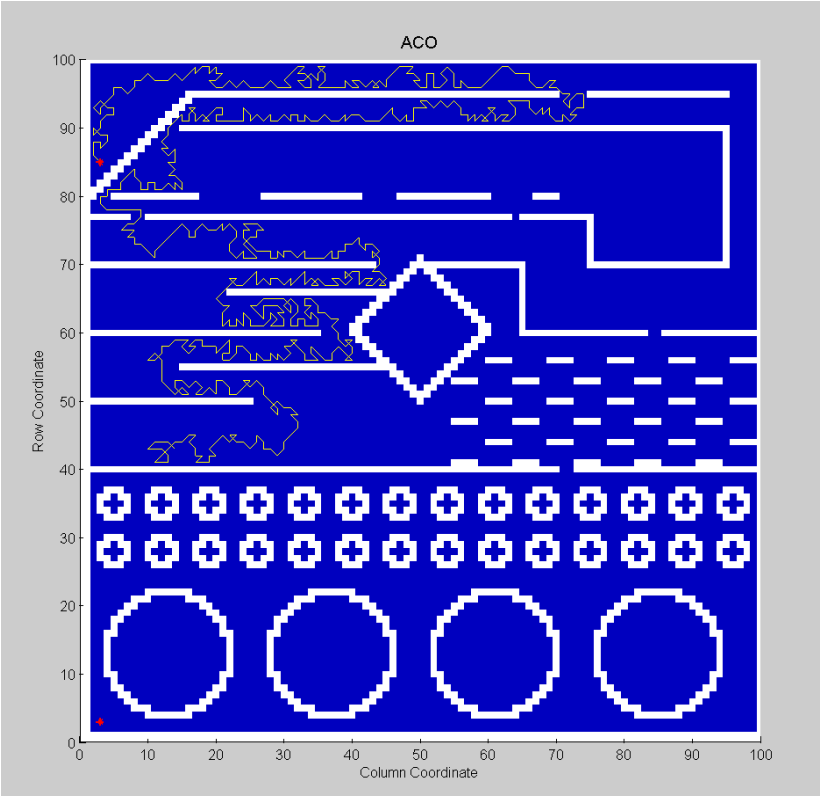

Figure 3. Difficult Search Problem for large β

Conclusion

Biological ants converge on the shortest path. This happens because as ants follow a shorter path, they are able to deposit pheromone quicker than by taking a longer path and over time this extra pheromone causes more ants to follow that path. To represent this artificially, the ACO ants deposit pheromone based on the length of their tour. By using this form of optimization, typical telemetry transmitters' paths can be predicted. These predictions can then be used to economically position telemetry receivers.

REFERENCES

[1] Engelbrecht, A.P.; "Ant Algorithms," *In Computational Intelligence: An Introduction*. 2nd ed., 359-411. West Sussex, England: John Wiley & Sons Ltd, 2007.

[2] Melchior, N.A.; Simmons, R.; "Particle RRT for Path Planning with Uncertainty," *IEEE Robotics and Automation*, April 2007.

[3] Jaillet, L.; Yershova, A.; La Valle, S.M.; Simeon, T.; "Adaptive Tuning of the Sampling Domain for Dynamic-Domain RRTs," *IEEE Intelligent Robots and Systems*, Aug. 2005.

[4] Yu Tian; Lei Yan; Gun-Young Park; Seung-Han Yang; Young-Suk Kim; Sang-Ryong Lee; Choon-Young Lee; "Application of RRT-based Local Path Planning Algorithm in Unknown Environment," *IEEE Computational Intelligence in Robotics and Automation*, June 2007.

[5] Meuth, R.J.; Wunsch II, D.C.; "Memetic Mission Management," *IEEE Computational Intelligence Magazine*, May 2010.

[6] Garro, B.A.; Sossa, H.; Vazquez, R.A.; "Path Planning Optimzation Using Bio-Inspirited Algorithms," *IEEE Artificial Intelligence*, Nov. 2006.

[7] Garro,B.A.; Sossa, H.; Vazquez, R.A.; "Evolving Ant Colony System for Optimizing Path Planning in Mobile Robots," *IEEE Electronics, Robotics and Automotive Mechanics*, Sept. 2007.

[8] Joon-Woo Lee; Jeong-Jung Kim; Ju-Jang Lee; "Improved Ant Colony Optimization Algorithm by Path Crossover for Optimal Path Planning," *IEEE Industrial Electronics*, July 2009.

[9] Joon-Woo Lee; Jeong-Jung Kim; Byoung-suk Choi; Ju-Jang Lee; "Improved Ant Colony Optimization Algorithm by Potential Field Concept for Optimal Path Planning," *IEEE Humanoid Robots*, Dec. 2008.