System Architecture Design Through Ant Colony Optimization Methods

CMPLXSYS 530
Colin Shields

Introduction

Naval vessel design requires the successful integration of multiple systems within a geometrically constrained space. Naval vessels must operate independently of external commodities thus there is a high level of functional interdependency between systems. Geometric constraints create spatial interdependency where systems must coexist in close proximity and performance constraints demand system performance in suboptimal conditions. Tightly constrained sets of interdependencies between heterogeneous systems make designing system routings a complex engineering problem. The difficulty of defining system connectivity or routing is usually avoided by only providing initial parametric estimates for system cost through regression. Once a design has developed further and predicated a small enough solution space for system routing, systems are then "filled in" based on best practices. Current methods disregard the negative impact of adding significant design information into a complex design space which frequently leads to design failure and cost overruns (United States Government Accountablity Office, 2009). This project will consider redefining naval system routing into a known problem archetype and propose a solution through an agent-based evolutionary algorithm.

Desired Contribution

This study presents a novel method for developing distributed system architecture(s) through an expanded ant colony optimization framework. A background on the generalized network design problem, complexity and survivability are presented first. Then, case study results demonstrate the application of the model and provide examples of model outputs. Next, a small study is presented to evaluate the effectiveness of pheromone dissipation rates on developing high-quality Pareto fronts of solutions. The paper concludes with a discussion and possible future work.

Background

This section will provide an overview of the general system design problem and ant colony optimization.

Multicommodity capacited network design

Spatial constraints on the paths available to route commodities means that multiple commodities may be required to share edges and nodes. Additionally, the nature of naval design implies that capacities cannot be changed once installed. The resulting problem is notoriously difficult and is known as multicommodity capacited network design (MCND) (Gendron, Crainic, & Frangioni, 1997). Furthermore, naval vessels must be survivable – able to provide adequate flow despite missing nodes and/or edges. Due to the difficulty of generating a survivable multicommodity capacited network design (SMCND) there is currently no solution or consensus best algorithm.

Solutions to MCND and SMCND have been generated through evolutionary algorithms, search techniques, and mixed-integer dynamic programming (Kleeman et al., 2007). Thus far, evolutionary algorithms have proven to be the most successful in terms of computation time and solution quality (Kleeman et al., 2007). However, the current formulations are intricate and computationally intensive making them ill-suited for large-scale naval design efforts. The proposed simplifications below to the SMCND formulation may reduce the difficult of the problem:

1. Many constraints found in SMCND will be captured in objective functions, increasing the number of feasible solutions, decreasing the effort of finding feasible solutions, and instead allowing the optimizer to determine sufficiency.
2. Capacities will be discrete, modeling commercial off-the-shelf (COTS) parts which has been shown to provide tractable solutions in similar problems (Maier & Simpson, 2003).
3. Commodities sharing a node or edge will be assumed to flow independently, modeling an applied system where pipes which belong to different system do not directly affect each other.
4. Routings will be evaluated by transforming topologies into single source, single sink systems and evaluated with max flow algorithms which reduces the details required for modeling while providing useful system information (Zhou, She, Xu, & Yokoyama, 2009).

Bi-objective ant colony optimization

Ant colony optimization (ACO) is a probalistic technique which models the ability of a swarm of ants to find high fitness paths through graphs (Dorigo & Di Caro, 1999). ACOs have also been successful applied to enumerative optimization problems and the basic algorithm has many extensions to improve its performance (Dorigo & Stützle, 2009). Expansions on the original ACO include multi-objective optimization which use multiple ant colonies and find solutions through inter-colony interactions. This project will use a bi-objective optimization approach with two colonies and an environment update approach which creates selection pressure to find solutions in unexplored areas of the Pareto frontier (Iredi, Merkle, & Middendorf, 2001). The two objectives, survivability and complexity are discussed below.

Survivability

Distributed systems with high survivability are required in naval design. Currently survivable systems are designed through best practice methods, fitting a topology into a routing template which has proven successful. This application requires an objective evaluation of survivability, making measurement of conformation to current templates a poor objective. To evaluate survivability, commodity flow will be evaluated during network percolation, accounting for cascading failures caused by inter-commodity dependencies. Cascading failures occur when there are interdependencies between systems causing an unaffected system to fail due to a disruption in another. Because the systems generated in this model share the same space, there is considerable opportunity of this type of failure to occur.

Evaluating survivability requires modeling network destruction on the flow network as well as considering how one commodity depends on another. For this paper, initial flow values of a system were found using network maximum flow. If that flow satisfied all sinks, the flow was sufficient and the evaluation returned a functionality score equal to the number of satisfied sinks. However, if the flow to a sink was not satisfied (above a specified threshold), that sink failed. When a sink fails it may cause a cascading failure through other connected systems. To evaluate this, when a sink failed that was linked to another system, the linked element was removed from the system and the flow algorithm was reapplied. If the flow values remained unchanged, the functionality score return, but if the linked failure caused additional change in the network flow or new failures, this step was repeated until a steady state was reached. After reaching an initial steady flow, percolation began and a random edge was removed. The functionality of this graph was then evaluated through the same process. This removal and evaluation cycle was repeated for a set number of removals. The overall survivability score of system was then the sum of functionality score over all removals, divided by the maximum functionality score (number of removals times number of sinks).

Complexity

Complexity has been identified as a significant cost indicator in engineering design and recent studies have shown strong correlation between design complexity and naval vessel cost (Arena, Blickstein, Younossi, & Grammich, 2006; Birkler et al., 2010; Dobson, 2014). To avoid cost model sensitivity, a network complexity metric was substituted for a cost objective. Complexity will be measured through a functional complexity measure. This complexity increases with the shared usage of network elements by multiple commodities. The complexity metric also accounts for the total capacity constraint by severely penalizing commodity routing combinations which exceed the maximum allowable capacity.

In this paper, complexity C [1] of a system represented by graph G is evaluated through a specified version of functional complexity as presented in Bar-Yam (2003). Functional complexity is defined as $C(\mathrm{f}) = \mathrm{C(a)}2^{C(e)}$ where $C(\mathrm{a})$ is the complexity of actions and $C(e)$ is the complexity of the environment (Bar-Yam, 2003). For the system model, the overall complexity is the sum of edge complexities C_e [2] and node complexities C_n [3] over the entire system,

$$C = \sum_{e \in G} C_e\,(e) + \sum_{n \in G} C_n\,(n) \qquad [1]$$

$$C_e(e) = \sum_{i \in S} c_i * 2^{\frac{\sum_{i \in S} c_i}{c_{max}} * \sum_{i \in S} \delta_i} \qquad [2]$$

$$C_n(n) = \left(\sum_{e \in G:t(e)} 1 + \sum_{e \in G:h(e)} 1 \right) * 2^{\sum_{i \in S} \delta_i} \qquad [3]$$

where c_i is the capacity of system i routed through an edge, c_{max} is the total available capacity of that edges and δ_i is an indicator variable for system i containing the element e or n.

Methods

This section will present the model framework, detail important method mechanisms and elaborate on the formulation of the objective functions.

Framework

ACO models the foraging behavior of a swarm of ants that travel at random while laying down pheromone trails (Dorigo & Di Caro, 1999). When a food source is found, the successful pheromone trail is reinforced, increasing the probability that other ants follow it. Over time, pheromone trails evaporate, reducing the attractiveness of paths which are not found to be optimum. Expansion on the original ACO to account for multicommodity routing and bi-objective optimization will be discussed below.

Agents

Simple ant agents are mostly unchanged in this formulation. They will still respond probabilistically to pheromones in the environment and attempt to follow a best path. To encourage multipath routing in the system, ants will have the ability to spawn another ant at each node based on a specific pheromone (Yang, Xu, Zhao, & Xu, 2010). The spawned ant will behave as a regular ant agent. At each agent step, ants will chose to spawn another ant, chose a new node to travel to, and chose a capacity path to that new node. No heuristics will be used for the determining probabilities of decisions, as non-heuristic variants have been shown to provide better solutions at negligible computation cost for bi-objective ACOs (Rada-Vilela, Chica, Cordón, & Damas, 2013); however heuristics have been added to sub-routines of the ACO. Probabilities will be calculated using the bi-objective method discussed in (Iredi et al., 2001). Commodities will be routed independently, with a single ant routing each commodity one after the other to create the proposed multicommodity routing.

Environment

Similar to the original ACO, the environment will contain pheromones which degrade with each environment step and are augmented by successful ants. However, the pheromone strategy must be updated to reflect the two extra decisions that ant agents must make. First, nodes will be given a branching pheromone which if high will encourage ants to branch out to multiple nodes. Second, for each node traversal a capacity decision must be made, the capacity decision will be modeled exactly as a classic node decision, with possible capacities for a single edge each having a pheromone level which determines the probability it is chosen. Additionally, to allow for algorithm termination, each node will carry a termination pheromone, which will remove an ant from the current random walk if the 'termination' path is chosen. All four pheromone types (branch, termination, direction, capacity) will exist for each commodity and each objective will have its own pheromone matrix. Operationally, the pheromones are stored in network data structure where nodes carry branching and termination pheromones and edges carry direction and capacity pheromones.

Institutions

Bi-objective optimization requires an institution structure to balance the search for both objectives and guide solution development. In this method, a multi-colony institution is used to encourage development of solutions along the Pareto front (Iredi et al., 2001). In single colony algorithms, ants which create non-dominated solutions are allowed to update the environment with their pheromones. However, if the same method is applied in multi-colony algorithms, it corresponds to merely increasing the number of ants in each generation, adding no benefit to the single colony approach. Introducing collaboration between the colonies through composite pheromones and update procedures provides better local and global selection pressure on the algorithm (Rada-Vilela et al., 2013).

The two pheromone matrices (one for each objective) described above, allow for composite pheromones to be created. Composite pheromones are composed by weighting the pheromone of each objective based on ant colony and ant number within the colony. To force the ants to search in different regions of the Pareto front, each of the m ants in a colony uses a unique weighting factor to create its composite pheromones. The weighting factor for ant k, $k \in [1, m]$ is $\lambda_k = \frac{k-1}{m-1}$. This weighting is used to create the probability of making decision d, p_d according to [4],

$$p_d = \frac{\tau_d^{\lambda_k} * \tau_d'^{(1-\lambda_k)}}{\sum_{d \in S} \tau_d^{\lambda_k} * \tau_d'^{(1-\lambda_k)}} \qquad [4]$$

where τ_d is the pheromone of decision d corresponding to the first objective function and τ_d' is the pheromone of the second objective function. Thus in extreme cases where the ant is either the first or last in the colony, the composite pheromone is composed of only the second or first objective function respectively.

When multiple colonies are employed using the same composite pheromone/probability structures, each colony will put forth a solution for a given weighting. This creates groupings along the Pareto front, improving local selection pressure along with the overall algorithms global selection pressure. Once the Pareto front solutions are selected, those corresponding ants are allowed to update the pheromone matrices. Figure 1 from Iredi et al. visualizes this selection process.

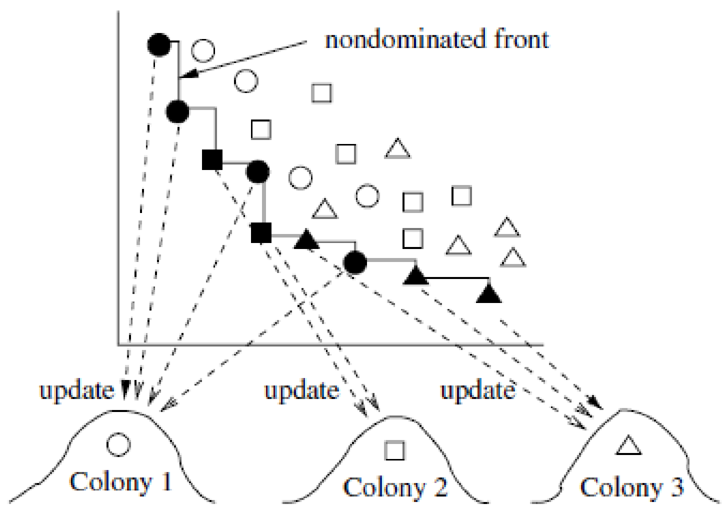

Figure 1: Pheromone update selection

The pheromone update procedure is completed by ants which create solutions on the global Pareto front when the solutions from all colonies are compared. Pheromone updates are completed as follows:

1. Environment pheromone dissipation - every pheromone is reduced by a dissipation rate ρ as seen in [5].

$$\tau_d = (1-\rho) * \tau_d \qquad [5]$$

2. Environment pheromone increment – Pareto front ants are allowed to update the pheromones by adding $\Delta\tau_d$ and $\Delta\tau'_d$ to the initial pheromone for the first and second objective functions respectively. In this study, $\Delta\tau_d := f(S)$, the value of the first objective function (survivability) which was formulated to be $0 \leq f(S) \leq 1$ and $\Delta\tau'_d := \frac{1}{\Gamma}$ where Γ is the total number of ants allowed to update. These update values where chosen because of they provided reasonably similar values without created redundant pheromone matrices; however, new increments may be explored in future work.

Model Mechanics

Individual agents produced solutions through the high-level process described above. During the random walk process, there is a set of sub-routines that each step of the walk follows. The list of active ants (one ant is initialized at each source) tracks the progress of the random walk. Each random walk step has all active ants preform this series:

1. Termination determination – In this sub-routine, which only occurs if all sinks are included in the current system, the ant decides if it will take another step or remove itself from the list of current ants. If the ant terminates, the random walk moves the next ant and starts at sub-routine 1.

2. Branch determination – In this sub-routine the ant choses how many branches they will create from the current nodes based on a pheromone. If the node has already been visited in this walk and the ants are not cycling, the ant chooses the number of branches that was previously chosen. However, if the ants are cycling – have not added a new node in 100 steps – the ant will choose one branch.
3. Direction determination – In this sub-routines the ant selects a destination node for each of its branches. Destinations are chosen from a pheromone list of all possible neighbors that have not been chosen as destination for the current branch. If all sinks are included in the current walk, the ant will prefer destinations already included in the walk. However, the ant will avoid its immediate predecessor, the node it was on during the previous step.
4. Capacity determination – In this sub-routines the ant selects a capacity for each of its branches. Capacities are chosen from a pheromone list of all capacities available for the given system with preference given to capacities greater than or equal the capacities of edges going into the node.

After this set of sub-routines was completed, the list of active ants is updated with the new nodes and the next step is taken. The random walk terminates when no active ants exist on the list or when the cycling criteria is met. Once a random walk is completed, the walk is pruned to remove non-sink, dead end nodes and 'cul-de-sac' nodes, where a small branch was created that returned immediately to its predecessor. The pruned graph iss then submitted for evaluation to the phase-one criteria.

To ensure solution quality a phase-one criteria has to be satisfied before an ant can return a solution to the solution set. The phase-one criteria – initial flow to every sink – ensured that all sinks were included in the generated systems, despite possible failures in the initial solution. This prevented infeasible solutions, which often had low complexity from being evaluated in the Pareto front and possibly allowed to update. However, the phase-one criteria caused many solutions to be rejected, especially during initial generations, significantly increasing computation time.

Case Study

This method has been applied at small case study to demonstrate its operation and provide a basis for an optimization behavior analysis. The case study considers a 16 node network, arranged in a grid, as shown below in Figure 2. Two systems will be routed through this network with the goal of maximizing survivability and minimizing complexity formulated as described in previous sections. The parameters for global space parameters and the system parameters are displayed in Table 1.

This case study represents designing two collaborative systems, sharing possible pathways that each must supply a set of unique sinks. The linkage parameter indicates that the sink at node $(3,3)$ of System 1, supplies the source at node $(3,3)$ of System 2. This problem instance was run for 5 trials of 25 generations, with the pheromone update methods described, allowing ants with solutions on the global Pareto front of all generations to update. Further, the pheromone dissipation rate ρ was swept across values $\rho \in [0,0.6]$ at increments of 0.1 to investigate the effect of pheromone dissipation on the development of optimal solutions and high quality Pareto fronts.

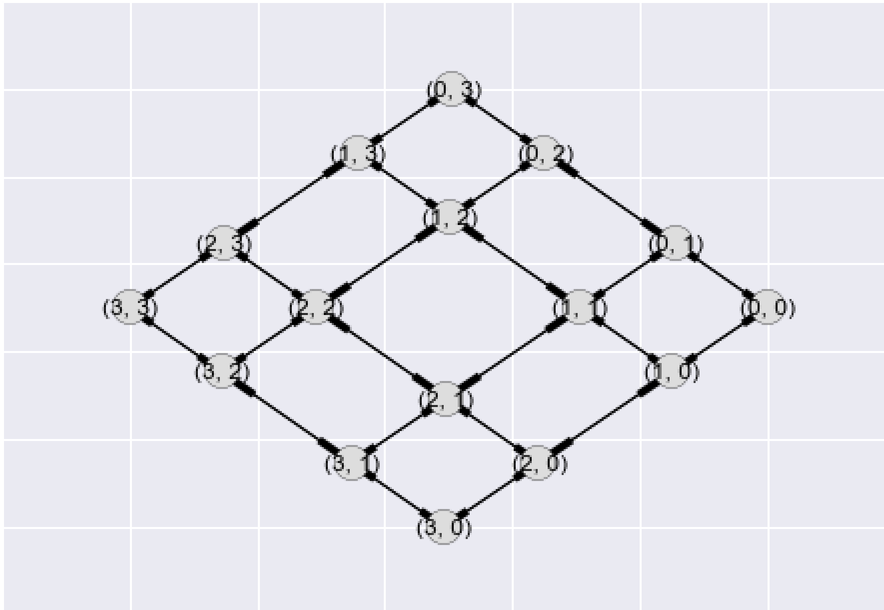

Figure 2: Case study network

Table 1: Case study parameters

	Space
Sink/source linkage	(3,3), 1->2
Edge capacity	20
Initial pheromone	1
Number of colonies	10
Ants per colonies	10

	System 1	System 2
List of sources: (location, magnitude)	(0,0), 20	(3,3), 10
List of sinks: (location, magnitude)	(3,3), 10 (1,3), 10	(0,0), 5 (2,0), 5
Sink operation thresholds	(3,3), 50% (1,3), 100%	(0,0), 100% (2,0), 100%
Routing capacities	5, 10	5

Results

Implementing the system architecture generation methodology on the case study has demonstrated that the expanded ACO formulation can generate and optimize high quality solutions with respect to nonlinear objective functions. Additionally, the case study illustrates how an ACO approach to network design successfully avoids common pitfalls of network generation such as disconnected graphs or graphs that do not include all critical nodes.

Example outputs

Based on the model instance defined in the case study, the outputs of a single trial will be shown in this section. The current model produces two outputs, system architectures and a history of the Pareto front through the generations. Outputs presented in this section use $\rho = 0.2$ and are only shown as an example. Figure 3 shows the Pareto front history of the solutions in this instance. The progression of the front from a single solution, to a spread of seven non-dominated solutions illustrates that the model formulation is capable of producing solutions and improving upon them. In this case, the front history also shows a tendency to 'jump', moving from a small set of local solutions to larger fronts and then converging again. The dynamics of this behavior and the mechanisms that cause this are still being considered.

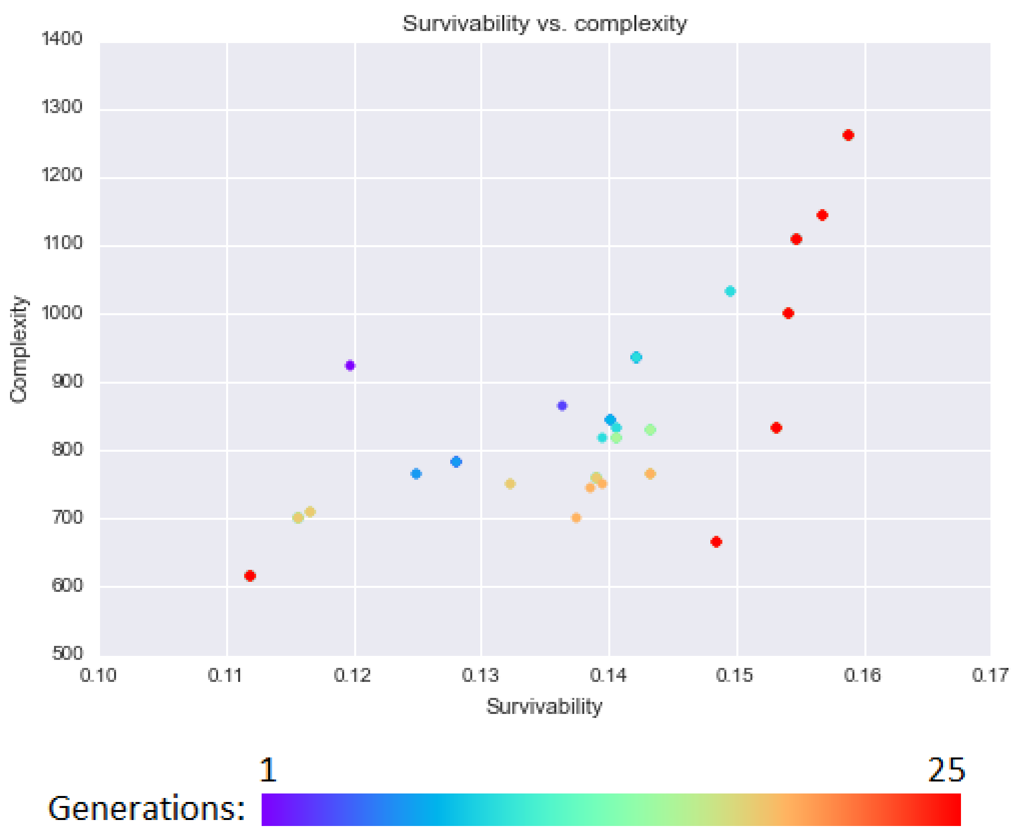

Figure 3: Example Pareto front history

Figure 4 shows system architectures at the two ends of the Pareto front in Figure 3. In these outputs, red nodes and edges represent the route of System 1, blue nodes and edges represent System 2 and orange nodes and edges are shared by both systems. As seen in the figure, both networks provide fully connected routing from each system's sources to respective sink. The system on the left, with the lowest complexity, has much fewer shared edges and nodes than its more complex counterpart, but suffers from decreased survivability due to a lack of redundant routing. On the other hand, the system on the right has considerably more shared network elements which results from individual systems enable rerouting of flow during failure.

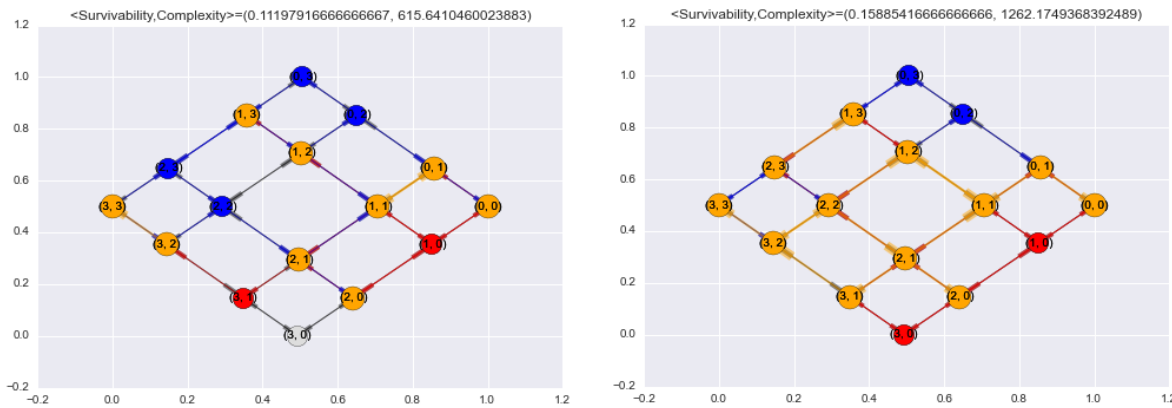

Figure 4: System topologies

The capability to generate the system architectures alone is an important outcome of the model. Demonstrating the ability to guide system creation in an optimization formulation shows that not only is the model applicable to small cases, but it can be expanded to more in-depth problem instances. The outputs of the model also provide a check on the model mechanics and behavior, showing that the process creates intuitive and reasonable results.

Dissipation rate testing

The case study was tested for 5 trials for pheromone dissipation rates $\rho \in [0,0.6]$ at increments of 0.1 to study how pheromone dissipation affected development of the Pareto front. It should be noted that there are many parameters that could have been tested (e.g. number of ants, avoidance), but due to the computation time necessary to generate model outputs only a small set of test could be conducted.

The Pareto front selection policy guarantees that the solution quality will not decrease through successive generations. Due to this, the solution set quality improved over the course of all generations; however, because the algorithm did not necessarily converge to a set of solutions over the 25 generations, it is difficult to determine how the Pareto front developed over the generations of ant. Wu and Azaram (2000) presented a number of metrics for evaluating the quality of a Pareto front. Two of those metrics, distance to utopia and Pareto spread are used for this analysis. Distance to utopia measures the minimum distance any point on the Pareto front is from the utopia point in Euclidian distance. In this case, the utopia point is $(1,0)$ indicating a completely survivable system with no complexity. The second metric, Pareto spread, is the area covered by the Pareto front, as illustrated in Figure 5 from (Wu & Azarm, 2001).

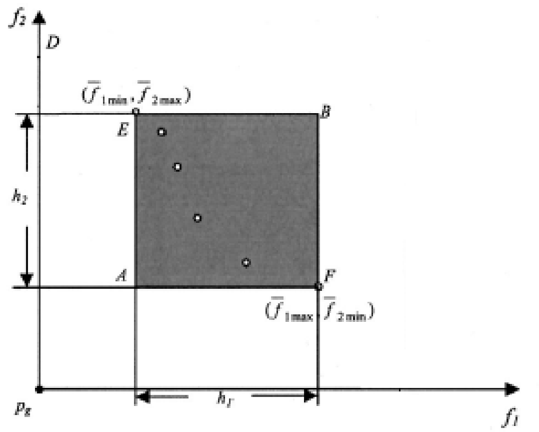

Figure 5: Pareto spread

The results of the parameter sweep did not show a strong correlation between pheromone dissipation rate and Pareto front development. Table 2 shows non-significant correlation between ρ and both distance to utopia and Pareto spread of the final generation of each trial. This can also be identified in Figure 6 and Figure 7 which graphically show the results of the trials for Pareto spread and distance to utopia respectively.

Table 2: Correlation of dissipation rate, ρ

	ρ	Pareto spread	Distance to u.
ρ	1.000	-0.288	0.246
Pareto spread	-0.288	1.000	-0.353
Distance to U.	0.246	-0.353	1.000

The non-significant correlation values for the metrics of Pareto front quality and pheromone dissipation rate imply that the persistence of pheromones is not critical to developing high-quality solution sets. This may be the consequence of early algorithm termination caused by the imposed 25 generation maximum. The low maximum generations may limit the ability of high quality solutions to imbed their pheromone trails due to noise from other solutions and randomness. The results may also be a ramification of the Pareto front selection process where only global solutions from all generations update the pheromones. This may create enough selection pressure by itself, that even with a high dissipation, the optimal solutions are able to guide the algorithm.

Ambiguity of the parameter sweep results warrants further study into the effectiveness of the pheromone update and Pareto selection procedure. A reasonable future study would compare the results presented here to random graph generation (without any pheromones) to see how much, if any, the update and selection process improved on the random case.

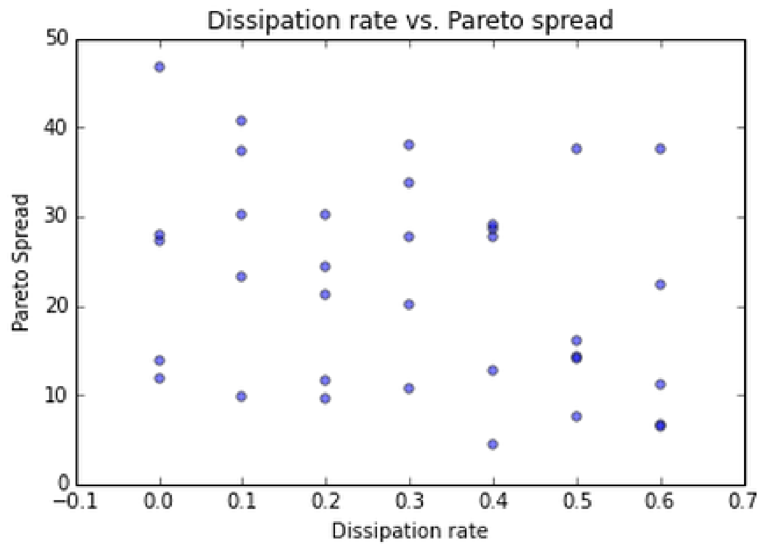

Figure 6: Dissipation rate vs. Pareto spread

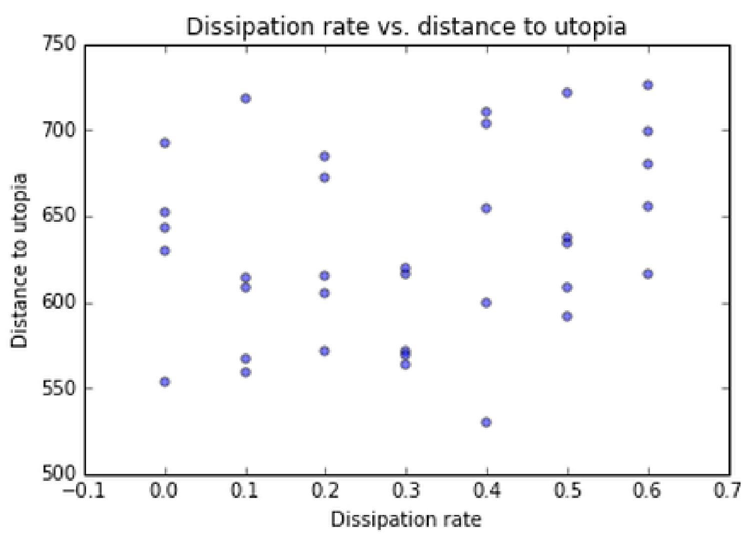

Figure 7: Dissipation rate vs. distance to utopia

Conclusion and discussion

This study presents a novel method for developing distributed system architectures through an expanded ant colony optimization framework. The case study results demonstrate the application of the model and provide examples of model outputs. Model outputs include the history of the Pareto front solutions over the generations and network representations of each solutions (including visualization). Additionally, a small study has been conducted to evaluate the effectiveness of pheromone dissipation rates on developing high-quality Pareto fronts of solutions. The results of this paper have demonstrated the system generation method is capable of creating architectures that are optimized to complex objective in an agent-based model. The quality model results and possible method applications warrant further exploration of the method.

There is a wide scope of applications for this method because it is generalized, requiring only system sources and sinks and the available pathways to produce solutions. As such, this formulation may be applied to any set of systems, collaborative or not. The objective functions used in this paper are indicative of complex non-linear objective, but are not the only objectives that could be used. Solutions could be evaluated on many alternative objective functions and the optimization would maintain is solving capability. Further, different flow evaluations could be employed to model other system dynamics. For instance, electrical flow or data transfer could be considered as opposed to a generic maximum flow as was done in this paper.

Future work may investigate more efficient implementations of the described method, improving termination criteria and improving pheromone update techniques. These avenues are left unexplored in this paper and could provide considerable benefit to further method application and development.

Bibliography

Arena, M. V., Blickstein, I., Younossi, O., & Grammich, C. a. (2006). *Why Has the Cost of Navy Ships Risen? Distribution*.

Bar-Yam, Y. (2003). Unifying principles in complex systems. *Converging Technology (NBIC) for Improving Human …*. Retrieved from http://necsi.edu/projects/yaneer/complexsystems.pdf

Birkler, J., Rushworth, D., Chiesa, J., Pung, H., Arena, M. V., & Schank, J. F. (2010). *Differences Between Military and Commercial Ship Building*. *Distribution*. http://doi.org/RAND

Dobson, A. T. (2014). *Cost Prediction Via Quantitative Analysis of Complexity in U.S. Navy Shipbuilding*.

Dorigo, M., & Di Caro, G. (1999). *The Ant Colony Optimization Meta-Heuristic*. *New Ideas in Optimization*. Retrieved from http://portal.acm.org/citation.cfm?id=329055.329062

Dorigo, M., & Stützle, T. (2009). Ant colony optimization. *Encyclopedia of Machine Learning*, *1*(May), 28–39. http://doi.org/10.1109/MCI.2006.329691

Gendron, B., Crainic, T. G., & Frangioni, A. (1997). MULTICOMMODITY CAPACITATED NETWORK DESIGN ebec Bernard Gendron Teodor Gabriel Crainic, (May 1997).

Iredi, S., Merkle, D., & Middendorf, M. (2001). Bi-criterion optimization with multi colony ant algorithms. *Evolutionary Multi-Criterion Optimization, Proceedings*, *1993*, 359–372. http://doi.org/10.1007/3-540-44719-9

Kleeman, M. P., Lamont, G. B., Hopkinson, K. M., & Graham, S. R. (2007). Solving Multicommodity Capacitated Network Design Problems using a Multiobjective Evolutionary Algorithm. *2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications*, *16*(4), 449–471. http://doi.org/10.1109/CISDA.2007.368132

Maier, H. R., & Simpson, a R. (2003). Ant Colony Optimization for Design of Water Distribution Systems. *Journal of Water Resources Planning and Management*, *129*(June), 10. http://doi.org/10.1061//asce/0733-9496/2003/129:3/200

Rada-Vilela, J., Chica, M., Cordón, Ó., & Damas, S. (2013). A comparative study of Multi-Objective Ant Colony Optimization algorithms for the Time and Space Assembly Line Balancing Problem. *Applied Soft Computing*, *13*(11), 4370–4382. http://doi.org/10.1016/j.asoc.2013.06.014

United States Government Accountablity Office. (2009). *High Levels of Knowledge at Key Points Differentiate Commercial Shipbuilding from Navy Shipbuilding*.

Wu, J., & Azarm, S. (2001). Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set. *Journal of Mechanical Design*, *123*(1), 18. http://doi.org/10.1115/1.1329875

Yang, J., Xu, M., Zhao, W., & Xu, B. (2010). A multipath routing protocol based on clustering and ant colony optimization for wireless sensor networks. *Sensors*, *10*, 4521–4540. http://doi.org/10.3390/s100504521

Zhou, Y. C., She, J. H., Xu, C. H., & Yokoyama, R. (2009). Evaluating robustness of power systems based on their topological structure using maximum-flow algorithm. *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, (1), 0–3. http://doi.org/10.1109/APPEEC.2009.4918341