Adaptive UAV Surveillance Using a Prize-Collecting Vertex Routing Model

Michael D. Moskal II, Rajan Batta

342 Bell Hall, Department of Industrial and Systems Engineering, University at Buffalo, Amherst, NY 14260

Abstract

In this work we analyze a special case of the prize-collecting vertex routing problem which a variation of the orienteering vehicle routing having profitable vertex collections, costs for each collection, and a maximum travel cost restriction.

The problem is defined on an undirected graph $G(V, E)$ with unique vertices α and Ω preset as the source and sink vertices, respectively. Each vertex $v_j \in G(V, E)$ has a corresponding profit value of I_j and collection cost r_j, where $I_j, r_j \in [0, 1] \; \forall \; j \in V$. Each edge $t_{i,j} \in E$ is a travel time from vertices $v_i \rightarrow v_j$ and a collection time d_j representing the time taken to collect profit on v_j. The sum of all travel times between vertices and profitable collection times must not exceed the predefined parameter T_{max}. Each vertex may be profitably visited multiple times, however, the vehicle-specific parameter S represents the vehicle's effectiveness at servicing a vertex and collecting profit which is represented as a collection percentage. This definition of profit-effectiveness is used in the context of unmanned aerial vehicle (UAV) surveillance specifically in a sensor's ability to collect information. It is in this context of UAV routing for Intelligence, Surveillance, and Reconnaissance (ISR) missions that this work is framed.

The problem is first put into context with a UAV, its equipment, area of operation, and routing (mission) parameters. The Prize-Collecting Vertex Routing (PCVR) model is introduced as an exact but computationally burdensome mixed integer program to generate the prize-collecting route. This

Email addresses: mmoskal@buffalo.edu (Michael D. Moskal II), batta@buffalo.edu (Rajan Batta)

is accompanied by a simulated annealing heuristic as a means to solve larger problem instances efficiently without the restrictions of optimization models. These methods are benchmarked alongside each other and provide further insight into solution properties. A hardware-in-the-loop demo is provided, and we finish with remarks and comments regarding the performance of these approaches and relevance of this work in the field.

Keywords: Optimization, modeling, mixed integer programming

---

1. Introduction

Prize-collecting models are specialized vehicle routing problems similar to the orienteering problem with the exception that the prize-collecting models include a cost term in the objective which is either based on travel location or travel time. These models are useful in trucking and freight transfer, where directed arcs in the graph represent a transfer job with a certain travel/fuel cost, and the vertex arrival is completion of a job with a corresponding revenue. In this example, the objective is to maximize total profit by completing jobs with the best payouts less expenses within a given time frame. Having a problem with multiple collections per vertex creates a graph with self-incident vertices with diminishing returns by revisiting the same vertex which is different from traditional orienteering, prize-collecting, and traveling salesman problem definitions.

In the less-conventional UAV example, vertices on the network represent areas of interest with a corresponding value to intelligence gathering operations effectively representing a utility value for a mission or series of concurrent missions. In determining whether to use edges or arcs, a directed graph is appropriate for fixed-wing or low maneuverability aircraft whereas an undirected network better serves more agile aircraft such as helicopters and other rotor-based drones. Each aircraft is equipped with sensors to collect information and its effectiveness is linked to factors such as weather, terrain, resolution, and mode. This furthers the motivation for the inclusion of the profit/sensor effectiveness parameter resulting in a decreasing profit upon revisiting an area of interest.

1.1. Relevant Literature

The orienteering problem initializes with predefined start and end vertices, a maximum allowed path length, and a set of candidate vertices with

an associated value [1]. The objective in the orienteering problem is to create a path along the network, limited in length, to maximize the values collected between the start and end vertices. This is also known as the generalized [2] or selective [3] traveling salesman problem and more intuitively, the maximum collection problem [4]. Another popular variation is the team orienteering problem [5] requiring multiple orienteering paths, with each path having a different combination of start and vertices and/or path length restrictions. An extensive literature review from Vansteenwegen et al. [6] covers all relevant work related to orienteering prior to 2010 including exact algorithms, stochastic heuristics, and deterministic heuristics. Golden et al. show that the orienteering problem is NP-Hard [7], as it relates closely with the traveling salesman and knapsack problems, while the team orienteering problem is NP-Complete [8].

More recently, research has shifted to the time-dependent orienteering problem allowing objective-improving visits to a vertex only if the visit falls within a time window [9]. The time-dependent team orienteering problem requires multiple orienteering paths to maximize the total collections across the entire graph, with one schedule for the time-dependent collections across all paths [10]. Given the added complexity to time-dependent orienteering problem, simulated annealing heuristics have been shown to be an effective, flexible, and efficient means for delivering approximate solutions [11] [12].

Another related NP-Hard problem is the privatized rural postman problem (PRPP) which considers a version of the traveling salesman problem with profits along the edges [13]. The objective is to find a cycle throughout the network passing through the source vertex while maximizing profits on each edge, subtracting travel costs incurred for each traversal but without limiting the total cycle length. The undirected capacitated arc routing problem with profits (UCARPP) [14] includes a fleet of capacitated vehicles, a set of profitable edges and demand nodes along the graph, and a maximum duration constraint for each vehicle. Instead of costs imposed on the objective from travel to demand vertices, vehicles are bound by their capacities and route length. The prize-collecting PRPP [15] is a generalization of the PRPP, focusing only on profitable collections minus displacement costs and without the need to traverse each edge.

The clustered prize-collecting arc routing problem (CPARP) [16] is a special case of the former problems which include predefined arc clusters which either each cluster must have all or none arcs traversed. The time-dependent prize-collecting arc routing problem (TD-PARP) [17] with costs

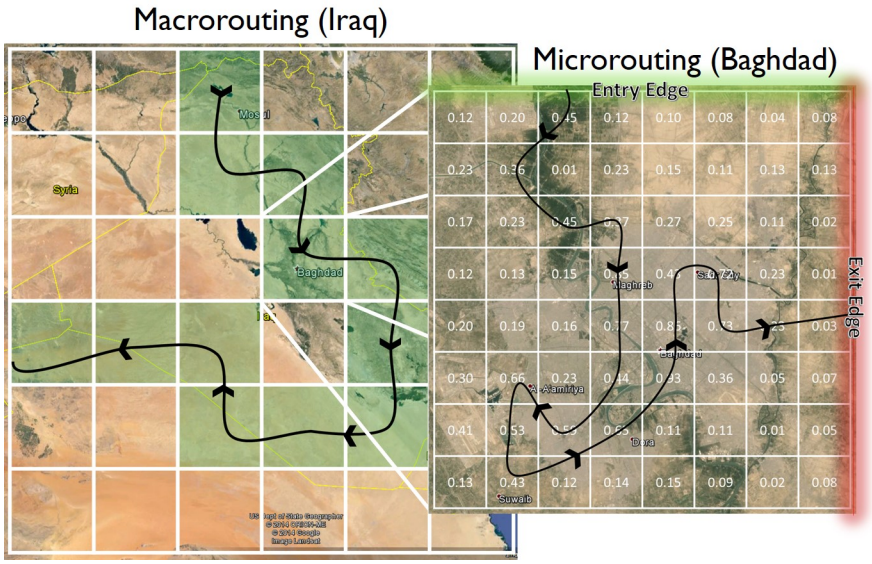
varying over time, possibly representing varying travel costs throughout the day due to congestion. Black et al. also include a concise table categorizing the work in arc routing problems, some of which are covered above. The problems in literature are differentiated by the number of vehicles considered, number of collections permitted per arc, the presence of a route duration constraint, capacity constraints on the vehicles, clustering, and whether or not the graph used is directed.

An emerging topic in the military domain is the routing and logistics of unmanned aerial vehicles, extending traditional optimization methods to consider practical implementations. Examples of optimization problems framed in the UAV context include theater distribution routing and scheduling [18], target coverage maximization [19], coordination and control of multiple UAVs [20] [21], priority-based target assignments [22], dispatching and loitering policies [23], and platform sensor selection [24]. The popularity and extensibility of these problems has led to the creation of universal routing framework specifically designed for UAV route optimization [25]. One common consideration across these pieces of literature is delivering accurate solutions in an often limited time frame, bridging optimization methods to application. An example of this is the maximum coverage stochastic orienteering problem with time windows (MCS-OPTW) [26] utilized in dynamic reconnaissance missions. Evers et al. considers an online approach, continually optimizing throughout the route, balancing the expected percentage of time-sensitive targets successfully reached versus the expected value of the foreseen targets.

1.2. Problem Definition

The initial formulation considers an AO discretized into a series of uniformly shaped microgrids representing areas of potential interest to intelligence operations illustrated in Figure 1. The environment is translated to an undirected graph $G(V, E)$. Each vertex in V represents the centroid of each microgrid with a corresponding prize, I_j where $I_j \in [0, 1] \; \forall \; v_j \in V$, which in the context of UAV routing may be interpreted as the normalized expected value or importance to the mission. The edges of the graph $t_{i,j} \in E$ captures the cost to travel from vertices $i \rightarrow j$ in V.

In the actual deployment of this work, prize values, or information gain values, are provided as an input to the model. These information gain values have different interpretations depending on the mission directives. A platform on a surveillance mission tasked with intelligence gathering surveys



Imagery ©Landsat, Map data ©2015 Google

Figure 1: Discretization of the problem

areas of high interest, numerically represented as utility values. Information gain values for support missions are interpreted as the relative demand for air intelligence of blue force operations within the UAV's AO. Information gain values captures the relative priority of different cells within the operating area. To generalize, "information gain" values represent the magnitude of vertex's profit relative to other vertices within the transportation network. In this interpretation, I_j is contained within $[0, 1]$ but defining $I_j \geq 0$ is effectively the same.

A platform's sensory equipment is imperfect, resulting in only partial information collection at each vertex. The sensor effectiveness, denoted by S, indicates the percentage of information collected at each visit. For example, a platform with sensor effectiveness of $S = .5$ visiting a vertex x with $I_x = 0.7$ would only collect $S * I_x = 0.50 * 0.7 = 0.35$ units worth of intelligence at this visit, and the next arrival at x would result in 0.175 units collected. This further complicates the formulation, allowing multiple collections of a single vertex's prize.

This problem considers aggressive routes as opposed to exhaustive routes, where aggressive routing limits the amount of time spent in the AO so information must be intelligently gathered. Exhaustive routing considers explor-

ing a region with excessive search time requiring a route focused on efficiency requiring a separate solution methodology. The entry and exit points for each AO are a known input to the model, treated as source and sink vertices in the problem, respectively.

2. Material and methods

The primary PCVR model is first introduced to solve the static prize variation of the problem formulated as a mixed binary program. The resulting solution provides an optimal sequence of vertices on a route that maximizes information gain for the parameters provided. The solution is then used in a Stage 2 PCVR (S2-PCVR) model which minimizes the total cost of the route since the PCVR model does not guarantee a shortest path between the waypoints on the route.

2.1. Prize-collecting Vertex Routing Model

The graph $G(V, E)$ contains the vertex set of all grid centroids with prize values I_j and edge costs $t_{i,j}$ indicating the time to travel between two vertices. The maximum time allotted to survey the operational area is set by T_{max}. The vertex set V also contains the source node α and sink node Ω, requiring the route to originate from α and end with Ω before T_{max} is reached.

Each vertex in the graph may be revisited multiple times to collect intelligence, which is controlled by parameter C. This limits the number of profitable collections that may occur at the same vertex, forcing the model to explore the entire space. Since the sensor effectiveness S results in partial collections at each grid, it leads to a new parameter definition to allow multiple collections defined as $I_{j,c}$ which captures the c^{th} prize collected at vertex j, where $I_{j,c} = S * (I_j - \sum_{l=1}^{i} I_{j,l}) \ \ \forall \ \ j \in V, c \in C$. This manipulation of the prize values is useful in the PCVR model and enables the solution to adequately capture the correct objective values.

Another important parameter set is T, indicating the maximum number of profitable collections a UAV may make which imposes a hard limit on the number of waypoints in the route. Two sets of binary decision variables are used in the formulation to build a feasible route and keep track of profitable collections in the objective. The decision variable $X_{t,i,j}$ identifies a traversal along edge (i, j) with discrete time-step index $t \in T$. The decision variable

$Y_{j,c}$ maintains the number of profitable collections at each vertex. The value of $\sum_{c \in C} Y_{j,c}$ indicates the number of collections at each vertex j.

$$\textbf{(PCVR)} \qquad MAXIMIZE \ \sum_{j \in N} \sum_{c \in C} I_{j,c} * Y_{j,c} - (1 - Y_{j,c}) r_{j,c} \tag{1}$$

$$\sum_{i \in N} X_{t,i,k} = \sum_{j \in N} X_{t+1,k,j} \quad \forall \quad k \in N, \quad \{t \in T | t \neq |T|\} \tag{2}$$

$$\sum_{i \in N} \sum_{j \in N} X_{t,i,j} \leq 1 \quad \forall \quad t \in T \tag{3}$$

$$\sum_{i \in N} \sum_{j \in N} \sum_{t \in T} X_{t,i,j} t_{i,j} + \sum_{j \in N} \sum_{c \in C} Y_{j,c} d_j \leq T_{max} \tag{4}$$

$$\sum_{j \in N} X_{1,\alpha,j} = 1 \tag{5}$$

$$\sum_{i \in N} X_{T,i,\Omega} = 1 \tag{6}$$

$$\sum_{c \in C} Y_{j,c} \leq \sum_{i \in N} \sum_{t \in T} X_{t,i,j} \quad \forall \quad j \in N \tag{7}$$

$$X_{t,i,j} \ \in \{0,1\}, \ Y_{j,c} \geq 0 \quad \forall \quad t,i,j,c \tag{8}$$

The objective value in (1) aims to maximize information gain relative to the prize values associated with each vertex and the number of times each vertex was profitably collected. To differentiate this formulation from an orienteering model, a penalty $r_{j,c}$ may also be imposed on the objective, resulting in an additional implied cost for each prize collection. Constraint (2) is a traditional flow balance constraint, requiring any vertex with an inbound flow at time-step index t to have an outbound flow at time-index $t+1$. Constraint (3) only allows one traversal at each time-index, and with the flow-balance constraints they ensures the route generated is continuous and also eliminates the need for sub-tour elimination constraints.

Constraint (4) relates an edge traversal from $i \rightarrow j$ to its associated travel cost $t_{i,j}$. These edge costs are regarded as time which may not exceed T_{max} for the solution route. Additional costs such as fuel consumption may be added with an additional constraint to ensure excessive fuel is not burned, but these constraints are omitted since it is assumed the UAVs will be operating

within their endurance limits and time is the only binding route duration constraint. Constraint (5) is the source constraint which states that the route must originate from the source location α at the first time index. The sink constraint is conceptually similar, requiring the UAV to reach the sink destination Ω at the end of the route. There are ultimately two ways to reach the sink vertex, by either visiting Ω by another vertex ($X_{T,i,\Omega}$) or by loitering at the sink vertex ($X_{T,\Omega,\Omega}$). The loitering cost at the sink vertex, $t_{\Omega,\Omega}$, is set to zero to ensure feasibility of the model when determining the most profitable set of $X_{t,i,j}$ variables to set. The source and sink vertices are unique since there are generally no prizes associated with them and the cost to loiter at those vertices is zero to prevent infeasibilities.

Constraint (7) relates $X_{t,i,j}$ to $Y_{j,c}$, stating that the maximum number of profitable collections on a vertex is limited to the number of times the vertex has been visited. The binary requirement on the decision variables is shown in (8).

The difficulty with modeling this problem is capturing the effect of sensor effectiveness. Figure 2 shows the network decomposition for a four by four area of operation. By setting $C = 3$ in the PCVR model, collections are limited to three per vertex which results in the omission of optimal PCVR routes with greater than three collections on a vertex. The decision space has $Tn^2 + Cn$ variables and by adding an extra collection only adds n decision variables with no additional constraints needed.

The decision variable $Y_{j,c}$ allows for an accurate representation of information collection in the objective since the $I_{j,c}$ coefficient is non-increasing as you increase the c^{th} index (i.e. $I_{j,1} \geq I_{j,2} \geq \ldots \geq I_{j,C}$), setting the $I_{j,c}$ with the largest collection first.

2.2. Solution Improvements

Optimality for the PCVR formulation requires a route that maximizes the value of the route that meets the time constraints which means it may not provide the shortest path between all waypoints. To avoid adding more complexity to the initial formulation, a secondary formulation takes the solution to the PCVR and solves to find the shortest path for the most profitable route. This Stage Two PCVR (S2-PCVR) model is formulated very similar to the PCVR formulation, but instead the objective minimizes the total cost of the route which includes travel and search costs or more simply: a

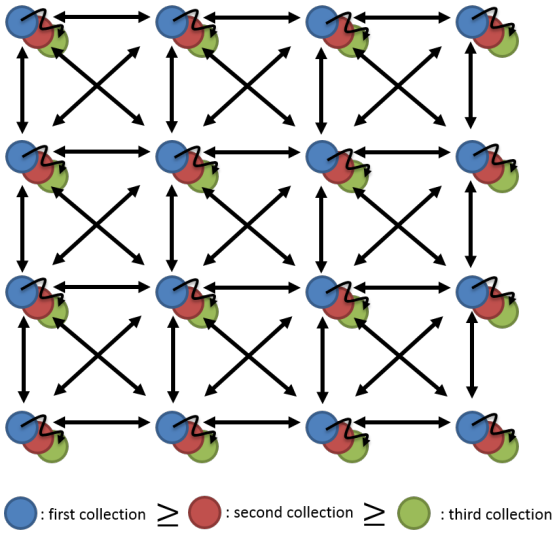


Figure 2: An example of a 16 vertex grid of a simplified UAV network, with $C = 3$, illustrating the increased network size by including the sensor effectiveness term

traveling salesman binary program customized for this problem.

$$\textbf{(S2-PCVR)} \qquad MINIMIZE \quad \sum_{t\in T}\sum_{i\in V}\sum_{j\in V} X'_{t,i,j}(t_{i,j}+d_j) \tag{9}$$

subject to

$$\sum_{i\in V} X'_{t,i,k} = \sum_{j\in N} X'_{t+1,k,j} \qquad \forall \quad k \in V \quad \{t \in T | t \neq |T|\} \tag{10}$$

$$\sum_{t\in T}\sum_{i\in V} X'_{t,i,j} \geq \sum_{c\in C} Y_{j,c} \qquad \forall \quad j \in V \tag{11}$$

$$\sum_{i\in N}\sum_{j\in N} X'_{t,i,j} \leq 1 \qquad \forall \quad t \in T \tag{12}$$

$$\sum_{j\in V} X'_{1,\alpha,j} = 1 \tag{13}$$

$$\sum_{i\in V} X'_{T,i,\Omega} = 1 \tag{14}$$

$$X'_{t,i,j} \in \{0,1\} \qquad \forall \quad t,i,j \tag{15}$$

The new graph $G'(V, E)$ contains only the vertices in the initial PCVR solution and their corresponding edges, where $G'(V, E) \subset G(V, E)$. The new decision variable $X_{t,i,j}$ serves as the new binary indicator for sequencing the waypoints. The decision variable $Y_{j,c}$ from the PCVR formulation now becomes the main input into the S2-PCVR formulation in constraint (11), forcing the shortest path formulation to visit each waypoint from the PCVR solution. The flow balance (10), single-traversal at each step (12), source (13), and sink (14) constraints all remain the same from its parent formulation.

Although the S2-PCVR is a smaller and generally easier problem to solve than a non-trivial parent PCVR problem, the S2-PCVR is still solving an NP-Hard problem. Practical, real-world problem instances are too large to solve using integer optimization leading to the creation of a traveling salesman heuristic technique followed by a simulated annealing approach in Subsection 2.3 to both complement and replace these models.

2.3. Traveling Salesman Problem - Heuristic / Simulated Annealing Heuristic

The Traveling Salesman Problem Heuristic / Simulated Annealing (TSP-H/SA) is designed as both a solution improvement method to the PCVR and a means for providing efficient and profitable routes on large problem instances, outlined in Figure 3. The heuristic is broken into three independent components: the Traveling Salesman Heuristic (TSP-H), Simulated Annealing (SA) objective improvement, and SA cost improvement. The TSP-H is a lightweight, greedy heuristic which quickly delivers an initialization route. The SA components focus on solution improvements given an initial route, and together with the TSP-H forms a standalone route generation algorithm.

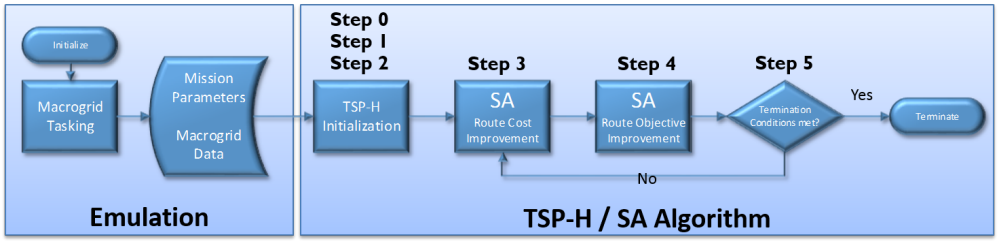


Figure 3: Flow chart of the emulation environment feeding data into the TSP-H / SA Algorithm

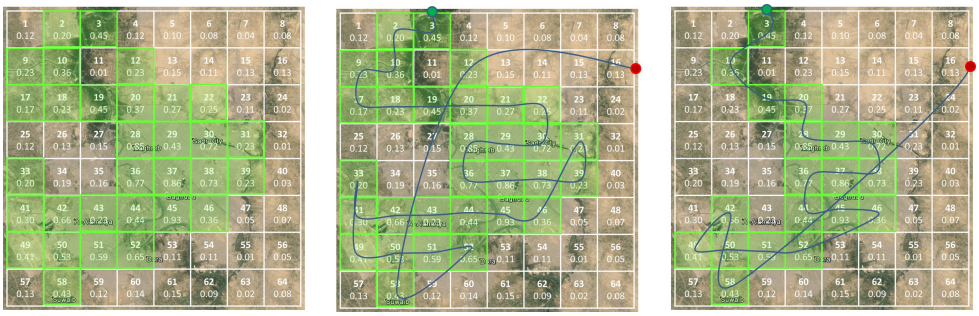
The algorithm is constructed in this manner to allow the implementation, interchanging, and a parameter tuning of individual components with

minimal overhead. This algorithm is also multi-threaded, allowing full utilization of the computer's processor. The algorithm implementation in this paper utilizes the main thread for mission emulation and algorithm initialization. Once a routing mission is created, the main thread runs the TSP-H, creates a number of worker threads, each as an instance of SA, and executes the algorithm until termination conditions are met. Each independent SA instance iterates between the route cost component and the route objective improvement, sharing only the best known solution between threads.

2.3.1. Traveling Salesman Problem Heuristic

The TSP-H is an implementation of a greedy solution generation heuristic, commonly referred to as the nearest-neighbor approach. The vertices within the macrogrid are ordered from highest to lowest, and a certain percentage of the most profitable vertices are selected as hotspots considered along the route. The heuristic starts at the source vertex, and constructs a route by traveling to the lowest cost waypoint with ties broken arbitrarily. Once all hotspots are on the route, the heuristic terminates by traveling directly to the egress point of the macrogrid. Once a complete route is generated, a feasibility and slack test is conducted to determine if the size of the hotspot subset should be adjusted. If the route cost is infeasible, the least-profitable vertices are removed, otherwise vertices are added to reduce all slack. Once the maximally-sized hotspot vertex set is determined according to this greedy algorithm, the route is then passed to the simulated annealing component.

Figure 4 illustrates an example TSP-H solution generation procedure, assuming an 8 unit^2 macrogrid with a total alotted mission time of 31.5 units, complete sensor effectiveness ($S = 1$,) and route cost is computed using Euclidean distances between grids. Initialization occurs in the left image with 50%, or 32 most profitable microgrids, selected as candidate hotspots. Given the source and sink nodes on the top and right edges, respectively, the nearest-neighbor generates a suboptimal TSP solution given the corresponding hotspots. This solution has an objective value of 13.79 units of information collection, but the route cost is 43.796 units which exceeds total alotted mission time. After several iterations of modifying the hotspot set size, the image on the right is an example of a maximally-sized hotspot set with a feasible nearest-neighbor solution. The objective for this example solution is 10.99 units of information collection and a route cost of 31.005 which is feasible.



Map data ©2013 Google

Figure 4: TSP-H hotspot initialization (left,) feasibility test on candidate hotspots (center,) and an example TSP-H solution (right)

Johnson et al. [27] show that the nearest-neighbor heuristic solution is on average 25% longer than optimal, and the example solution provided above is 22.3% longer than the shortest path between the selected hotspots. In addition, the candidate hotspots chosen to be in the final route are not typically representative of an optimal subset selection, since lower-valued vertices are ignored even if they do not add significant cost to the route. This solution is practically useful knowing that the highest-valued cells are visited with some expectation on the route cost using minimal computational resources.

2.3.2. Simulated Annealing: Overview

The SA component is broken up into two subclasses: the route cost (time) improvement algorithm and the route objective (information collection) components. Although these heuristics run independent of each other, they are instances of simulated annealing superclass where their core functionality is derived. There are options to reset the candidate solution back to the best known solution once either a set number of iterations without improvement have been reached, or once the candidate objective becomes a certain percentage worse than the best known solution.

The simulated annealing superclass also handles the logic of running both the route cost and objective improvement instances together. Once the TSP-H solution is complete, minimization of the route cost occurs and once complete, maximization of the objective occurs. This process continues until the terminating conditions are met. In the multi-threaded implementation explored, each instance of the simulated annealing heuristics may have their

own cooling schedules and restart parameters allowing the possibility of conservative and aggressive exploration of the solution space concurrently.

The cost improvement simulated annealing is an alternative to the S2-PCVR formulation, eliminating the need for optimization software. The main SA implementation has built in cooling schedules, parameter optimization, and restart conditions. Having two different subclasses allows another model or heuristic to call an SA instance improve its solution by improving only the efficiency of the route, improving only the objective, or improving both and agitating solutions between iterations. The cost improvement algorithm keeps all points on the route but swaps them in order to find a quicker route. The route objective improvement algorithm adds and deletes waypoints, but unlike the cost improvement algorithm, it must maintain route feasibility. In the case where both route cost and the objective are optimized, the route recursively passes through each instance of the algorithm until the maximum solve time is reached or the heuristic adequately converges to a solution. In the multi-threaded implementation of the TSP-H/SA, each instance of the simulated annealing runs independent of each other only sharing the best known solution between threads. The best known solution is only accessed during a restart event or in the beginning of a new cooling cycle.

The operations conducted at each cooling step do not necessarily involve one vertex at a time. A set of probabilities $O = \{o_1, o_2, \ldots, o_Z\}$ determines how many how many waypoints are considered for each step where typically $Z = 10$, representing the maximum possible number of waypoints to consider. The probabilities are non-increasing (i.e. $o_1 \geq o_2 \geq \ldots \geq o_Z$) to favor small operations. This set is created by generating a string of Z non-increasing numbers and normalizing their sum to 1. Once the number of waypoints are chosen, the route cost improvement component cuts a series of consecutive waypoints from the route and places them randomly. The objective improvement component is slightly more complicated since it uses the set two different probabilistic sets to determine how many waypoints to add and remove separately. Since the objective improvement component only accepts time-feasible solutions, waypoints must often be deleted in saturated routes to make room for feasible candidate solutions. The only bias introduced to the solution procedure is when a waypoint is added when attempting to improve the objective, placing it along the route at point with the lowest impact to total route cost in an attempt to reduce cost infeasibilities.

Traditional simulated annealing methods involve moving to a higher energy state (worse solution) with probability $P = exp(-\Delta E/kT)$. In this

implementation ΔE is the percent decrease in the objective value or the increase in route cost of a candidate solution relative to the best known solution multiplied by a factor of 100. Routes with improved solutions are automatically accepted and checked against the best known solution across all threads. Parameter $k = 1$ and T is the current temperature of the system according to the cooling schedule. The energy state ΔE is the a function of percent change rather than comparing objective values and route costs directly to avoid scaling the parameter k for specific graph types or maximum allowed route lengths.

3. Results

In order to evaluate the performance of the PCVR model and the TSP-H/SA heuristic we must run a series of computational experiments. The data emulator and TSP-H/SA heuristic are developed in Java and the PCVR is modeled using Concert Technology for Java utilizing the CPLEX 12.6 library. Default optimization and solution parameters are used within CPLEX. All experiments utilize 6 cores of an Intel Xeon E5645 Westmere-EP 2.4 GHz processor on a Dell production server running RedHat Enterprise Linux 6.1 on the 2.6.32 Kernel. Each instance had 24GB of memory allocated, with a maximum heap space of 23GB allocated to the JVM to ensure adequate headspace. Every problem instance is permitted 30 minutes (1800 seconds) of CPU time to generate a solution which does not include data emulation or statistical computations.

Several key assumptions must be made in order to limit the scope of the study and provide meaningful and comparable results. In this study, macrogrid dimensions are fixed to 10 units by 10 units and the travel cost incurred through routing are the Euclidean distances between microgrid centroids. To avoid non-square microgrid sizes and ensure uniform microgrid placement throughout the macrogrid, only vertex sizes of perfect squares are considered. Adding more microgrids to the candidate problem results in a greater density of smaller microgrids in the fixed 10x10 unit macrogrid space, requiring scaling the loitering and search costs inversely with the number of microgrids. Search cost is restricted to 30 units ($T_{max} = 30$), all penalties from visiting a microgrid are set to zero, sensor effectiveness is at 50% ($S = 0.50$), and all prize values at each microgrid are generated uniformily on [0,1]. All routes start at a source vertex located at (0,0) and must terminate at the sink vertex located on (10,10) located in the opposite corner.

The TSP-H/SA heuristic uses a linear cooling schedule for both the distance and objective improvement components. All restart conditions are removed from the annealing process and the only terminating conditions for each iteration are the ending temperature or maximum solve time for the entire heuristic. The starting temperature is 25 degrees with a cooling rate of 0.000125 degrees/iteration and the cooling terminates at 0.0001 degrees across all threads. This high starting temperature permits a better exploration of the solution space over 200,000 annealing iterations per each step of the TSP-H/SA heuristic. This in effect acts as the restart conditions as a new simulated annealing instance is created according to the procedure outlined in Figure 3 above. Five of the six available threads execute the SA component in parallel, while the main thread first initializes the TSP-H solution and then manages all active threads and their solutions.

3.1. Small Problem Instances: PCVR vs. TSP-H/SA

The purpose of this section is two-fold: (1) to identify the computational limitations of the PCVR model and (2) to compare the performance of the PCVR model against the TSP-H/SA heuristic. The study consisted of testing 16, 25, 36, and 49 profitable vertices where instances of 64 vertices and up ran into consistent memory issues using CPLEX. The size of the PCVR model and solution quality is in part a function of the maximum number of allowed collections per microgrid, so the parameter C is tested at 1, 5, and 10 for each macrogrid. A maximum of 50 waypoints are permitted on a PCVR route ($T = 50$) but if $N * C < 50$, then $T = N * C$ to reduce presolve and model reduction time within the CPLEX engine.

A total of 10 replications are made for each size of N considered. Each replication utilizes the same macrogrid, and the TSP-H/SA heuristic and three PCVR model instances with differing values of C are conducted independent of each other resulting in 120 different routing instances. Although 1800 seconds are allocated for run time, CPLEX may converge to optimality beforehand and these solutions and solution times are compared to those of the TSP-H/SA heuristic's best known solution within the solve time window. Table 1 below captures the standalone performance of the PCVR model while Table 2 the average relative percent difference of the TSP-H/SA heuristic solution on the same problems.

The columns within Table 1 have the following meanings: N is the number of vertices or microgrids considered, C is PCVR model parameter for the maximum number of collections per vertex, $|V|$ is the number of vertices

represented in the PCVR where $|V| = N * c$, Dvars are the total number of decision variables, and Constraints are the total number of constraints in the model. The term ST_{PCVR} is the average solve time across all replications for each unique combination of N and C in seconds, IG_{PCVR} is the average integrality gap provided by CPLEX, and WP_{PCVR} is the average number of waypoints on the route.

Table 1: Computational results of the PCVR model on small problem instances varying parameter C

Computational Results of the PCVR Model											
N	**C**	**|V|**	**|E|**	**Dvars**	**Constraints**	**ST_{PCVR}**	**HW**	**IG_{PCVR}**	**HW**	**WP_{PCVR}**	**HW**
16	**1**	16	240	4112	819	11.10	6.40	0.00%	0.00%	12.0	0.00
	5	80	6320	320400	4053	167.57	89.34	0.00%	0.00%	13.3	0.68
	10	160	25440	1281600	8053	945.47	434.32	1.06%	1.75%	13.1	0.53
25	**1**	25	600	15650	1303	119.35	36.01	0.01%	0.00%	14.5	0.61
	5	125	15500	781875	6303	1704.45	216.69	15.22%	4.47%	16.1	0.63
	10	250	62250	3127500	12553	1800.48	0.13	29.61%	5.83%	16.0	1.35
36	**1**	36	1260	46692	1839	1800.26	0.04	14.98%	3.33%	16.4	0.50
	5	180	32220	1620900	9053	1800.43	0.07	49.60%	9.15%	16.7	1.75
	10	360	129240	6483600	18053	1800.48	0.05	57.27%	9.50%	16.7	1.91
49	**1**	49	2352	117698	2502	1800.36	0.14	39.57%	5.45%	19.4	1.02
	5	245	59780	3002475	12303	1800.77	0.14	79.57%	16.54%	17.1	3.10
	10	490	239610	12009900	24553	1801.08	0.55	80.75%	17.48%	14.1	2.51

Table 2 compares the solution values of the TSP-H/SA model relative to the PCVR model averaged across each replication for the problem size. For example $\mathbf{O_{TSPH/SA} - O_{PCVR}}$ represents the average percent change of objective values of the TSP-H from the PCVR solution, with the $O_{TSPH/SA}$ in the numerator and O_{PCVR} in the denominator. This trend continues for the average percent differences in the route length, $\mathbf{RL_{TSPH/SA} - RL_{PCVR}}$, and the solve times $\mathbf{ST_{TSPH/SA} - ST_{PCVR}}$. The value for $ST_{TSPH/SA}$ represent the time recorded when the best known solution was found since the only terminating conditions for the TSP-H/SA heuristic was set only to solve time in this study. The value of $\mathbf{R}$ counts the number of replications where the PCVR model outperformed the TSP-H/SA heuristic with a total of 40 TSP-H/SA runs, 10 for each value of N, are compared to 120 PCVR runs resulting from each value of N and C. All half widths denoted **HW** are generated with $\alpha = 0.05$ using a Student's T distribution with standard deviations based on the sample.

Once the problem sizes grow from $N = 36$ and beyond, the PCVR model no longer converges to optimality within the prescribed solve time with a substantial theoretical gap in these cases. Although the mixed integer program is computationally burdensome, there are 35 out of 120 cases where the model provided a better solution than the heuristic right up until the threshold where the problem sizes became too large for the PCVR model. The dynamic search procedure in a few special cases can quickly find a strong set of cutting planes that dramatically reduce the branch and cut tree structure

Table 2: The computational results of the TSP-H/SA heuristic relative to the PCVR model solutions above on identical problem instances

Computational Results of the TSP-H/SA Relative to the PCVR Model								
N	**C**	$\mathbf{O_{TSPH/SA} - O_{PCVR}}$	**HW**	$\mathbf{RL_{TSPH/SA} - RL_{PCVR}}$	**HW**	$\mathbf{ST_{TSPH/SA} - ST_{PCVR}}$	**HW**	R
16	**1**	5.10%	2.40%	9.59%	0.56%	8952.98%	5919.96%	1
	5	-0.97%	3.13%	13.21%	1.30%	696.33%	616.34%	5
	10	-0.81%	3.14%	13.44%	0.77%	68.47%	139.46%	5
25	**1**	1.94%	4.21%	10.03%	0.90%	202.39%	298.65%	2
	5	-2.22%	3.42%	12.21%	1.04%	-80.76%	17.05%	7
	10	4.17%	4.30%	12.34%	1.39%	-81.92%	17.15%	3
36	**1**	-0.98%	3.86%	10.10%	0.80%	-82.70%	15.63%	6
	5	12.30%	9.67%	11.09%	1.14%	-82.70%	15.63%	2
	10	16.69%	8.18%	11.46%	1.00%	-82.70%	15.63%	0
49	**1**	4.30%	6.21%	11.70%	1.09%	-79.44%	22.73%	2
	5	27.94%	14.16%	16.70%	6.01%	-79.44%	22.72%	1
	10	29.06%	13.90%	14.25%	6.89%	-79.45%	22.72%	1

and provide quality candidate solutions. This is in part responsible for better PCVR solutions over the TSP-H/SA heuristic, and also allows CPLEX to inconsistently handle some problems where $N = \{64, 81, 100, 121\}$.

The TSP-H/SA is does not intelligently search the solution space as efficiently as our mixed integer program, but it has no restrictions on the maximum number of collections per vertex is less affected by longer route lengths. Although the TSP-H/SA heuristic outperformed the PCVR model in 85/120 possible comparisons, the model converged quicker than the heuristic on the smallest instances. Although $T_{max} = 30$ for all instances, every TSP-H/SA solution was longer than every PCVR solution since the SA objective improvement component accepts all objective improving waypoints into a route as long as it is time feasible. Any significant improvements made by the distance improvement component is often quickly taken advantage of my the next iteration of the objective improvement component, even if it is a trivial or inefficient improvement in the objective.

As the problem sizes increase, the heuristic finds favorable solutions quicker than the model, although not all of these solutions are better. One way to make problems more tractable in the PCVR model is to reduce both parameters T and C to the tightest possible value but then there is a substantial trade-off of tractability versus true optimality based on the problem definition. In this study, when $C = 1$, $T = N$ which places a strong restriction on the solution but for N equal to 36 and 49 with $C = 1$, the solution gaps average 15.0% and 39.6%, respectively.

3.2. Large Problem Instances: TSP-H/SA

This section focuses on the TSP-H/SA on large problem instances with $N = \{100, 400, 900, 1600, 2500, 3600, 4900, 6400, 8100, 10000\}$. Since there are no benchmark cases available for this problem, we restrict our attention to the relative improvement of the heuristic based on the TSP-H initial

solution, the SA solution after 5 minutes of run time, and the best known solution after 30 minutes. A total of 25 replications are conducted for each value of N each requiring a new macrogrid, resulting in a study of 250 total replications across this study. The resulting statistics from these replications are found below in Table 3. Table 4 shows the relative improvement of the TSP-H/SA heuristic after five minutes of run time to the initial TSP-H solution along with a comparison of the final best known solution relative to the solution after five minutes.

The columns in 3 follow a similar convention to 2. The term $\mathbf{O_{TSP-H}}$ is the average objective of the initial TSP-H route, $\mathbf{WP_{TSP-H}}$ is the average number of waypoints on the TSP-H route, $\mathbf{O_{BKS}}$ is the average objective value of the best known solution and the best known solution's average number of waypoints and average time to find the best known solution are $\mathbf{WP_{BKS}}$ and $\mathbf{ST_{BKS}}$, respectively.

Table 3: Computational results using the TSP-H/SA algorithm averaged across 25 replications for each value of N

Computational Results of the TSP-H/SA Heuristic on Large Problem Instances										
N	$\mathbf{O_{TSP-H}}$	**HW**	$\mathbf{WP_{TSP-H}}$	**HW**	$\mathbf{O_{BKS}}$	**HW**	$\mathbf{WP_{BKS}}$	**HW**	$\mathbf{ST_{BKS}}$	**HW**
100	6.02	0.4164	16	1.09	8.38	0.28	31.72	0.72	413.47	176.08
400	8.57	0.8896	20.04	1.95	15.01	0.53	59.92	1.70	662.26	187.63
900	7.64	1.2164	17.6	2.53	20.25	0.59	90.24	1.99	1494.78	156.42
1600	6.96	1.3664	16.08	2.78	24.62	0.68	112.52	2.56	1655.96	79.37
2500	5.62	1.7727	13.32	3.59	28.12	0.80	123.32	3.65	1647.46	65.22
3600	4.49	0.0014	11	0.00	29.42	0.76	130.64	2.90	1704.44	38.37
4900	5.98	0.0015	14	0.00	30.56	0.71	134.80	4.65	1655.04	57.56
6400	7.98	0.0019	18	0.00	31.17	0.87	135.72	5.02	1686.81	56.24
8100	9.98	0.0022	22	0.00	31.21	1.24	130.96	5.60	1697.18	36.64
10000	12.47	0.0021	27	0.00	30.95	1.25	124.92	6.92	1689.15	47.33

The values for $\mathbf{O_{SA5} - O_{TSP-H}}$ in Table 4 represent the average percent objective improvement of the simulated annealing iterations after five minutes compared to the initial TSP-H solution. The best known solution's objective is compared to the simulated annealing's objective after five minutes through $\mathbf{O_{BKS} - O_{SA5}}$, capturing the average percent improvement of the simulated annealing with an additional 25 minutes of run time. The half widths are similary denoted **HW** with $\alpha = 0.05$ using the same procedure.

Table 4: The average improvement of the initial solution to TSP-H/SA solution after five minutes, and the TSP-H/SA solution after five minutes relative to the best known solution of the TSP-H/SA after 30 minutes

Relative Solution Improvements of the TSP-H/SA Heuristic				
N	$\mathbf{O_{SA5} - O_{TSP-H}}$	**HW**	$\mathbf{O_{BKS} - O_{SA5}}$	**HW**
100	**39.62%**	6.79%	**41.62%**	7.01%
400	**80.46%**	17.81%	**85.84%**	20.20%
900	**181.86%**	41.54%	**203.72%**	44.90%
1600	**350.11%**	144.66%	**425.63%**	174.99%
2500	**486.58%**	108.43%	**638.24%**	135.48%
3600	**403.54%**	22.53%	**555.54%**	17.06%
4900	**265.96%**	19.91%	**410.69%**	11.88%
6400	**133.56%**	17.37%	**290.59%**	10.93%
8100	**72.57%**	11.44%	**212.79%**	12.41%
10000	**33.26%**	7.72%	**148.26%**	9.99%

The most interesting information of this study is the relative improvement of the SA best known solution and the SA best known solution after 5 minutes of run time versus the initial TSP-H solution. The average percent improvement of the SA improvement iterations are increasing from 100 to 2,500 vertices. Beyond 2,500 vertices, the relative improvement from the TSP-H solution is still suitable, but the magnitude of the average relative improvement is decreasing up to the 10,000 vertex problems. It is important to point out that the TSP-H initial solutions on problems for $N \geq 3600$ fails to find the maximum subset of high-ranking vertices in the predefined iteration limit, so a default number of waypoints is chosen for a problem of that size resulting in no variance among TSP-H solution waypoint counts. This results in a less-profitable TSP-H objective which should allow a greater chance for improvement in these problems. Although the cooling schedules, number of parallel instances running SA, number of waypoints considered for each SA temperature step, solve time, and initial solution quality play a role in the quality of the best known solution, ultimately the problem size becomes the major limiting factor. There is considerable memory overhead storing all of the information contained within a macrogrid and the candidate routes, although no memory issues were reported even with the largest problem sizes.

The inefficiencies of the SA components is apparent once route sizes approach 120 waypoints. The "move" operation in the distance improvement component and the "add/remove" operations in the objective improvement component require rebuilding a list that comprises the route. These operations occur at each temperature step, and every new list must be iterated over to calculate the new route length and/or objective value. Operations on larger routes take longer, meaning it also takes longer to complete a cooling schedule within an SA component. Longer operations on large problem sizes result in less exploration of the solution space which is shown in this study for $N \geq 3600$. Allocating solve time as a function of the problem size is one means to achieve comparable results across most values of N. Introducing metaheuristics to the SA components such as tabu or local neighborhood search may create provide better solution exploration, but they also have computational overhead and may bias solutions towards local optima if not carefully implemented.

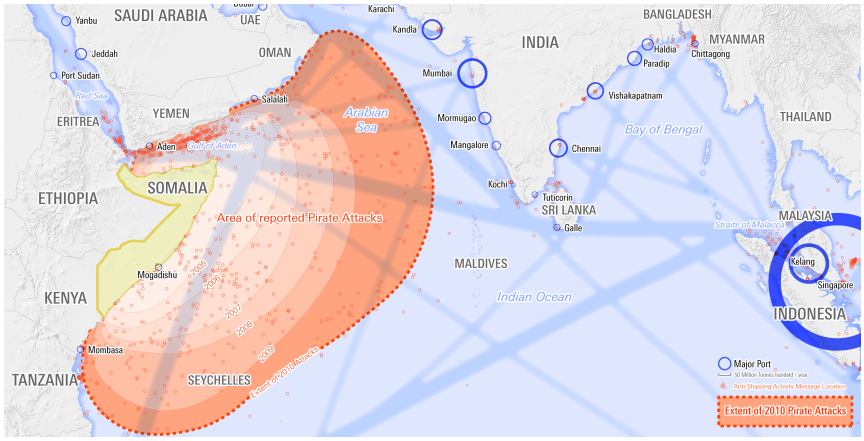
4. Hypothetical Scenario Usage

An emphasis of this research is focused heavily on the applicability and relevance in the military domain. This section aims to motivate a hypothetical scenario where these models are used on actual hardware in the context of a real-world situation. The issue of piracy off the coast of Somalia is chosen as a less-controversial topic which may benefit from automated routing of UAVs. A strong naval presence has been present off the coast of Somalia which may benefit from adaptive surveillance routing of UAVs. The hypothetical scenario is built around a suspected hijacking in the Gulf of Aden requiring an immediate response by patrolling forces. Operational details of the scenario are ported to the PCVR model which is executed in real-time. The resulting solution is transmitted to the flight hardware, the prescribed route is flown, and information is collected and processed along the route.

4.1. Motivating Background

The Gulf of Aden located between Yemen and Somalia in the Arabian Sea is an integral part of the Suez Canal with an estimated 21,000 ships traveling on the gulf each year [28]. Somali piracy attempts began to dramatically increase in 2005 in part due to the instability in the region due to the ongoing Somali Civil War and economic hardship faced by fisherman due to polluted waters [29] [30]. The pirates began to target merchant vessels to deter commercial trading from the area, loot the cargo, or hold the vessel, passengers, and cargo for ransom. Estimated losses due to piracy was estimated upwards of $6.9 billion dollars (USD) in 2011 for global trade [28]. The proliferation of piracy is shown in Figure 5, illustrating the distribution of attacks relative to trade routes and major ports in the area.

Combined Task Force 150 (CTF-150) is a US-led 25-nation coalition of combined naval resources based in Bahrain initially tasked with maritime security and counter terrorism in the area comprising 2.5 million square miles around the Horn of Africa [31]. From 2002-2005, the task force was primarily focused on threats regarding the Global War on Terrorism as part of Maritime Security Operations under Operation Enduring Freedom - Horn of Africa (OEF-HOA) [32]. Their directives included visit, board, search, and seizure (VPSS) operations on suspect vessels to deter threatening activities such as black market drug smuggling and arms trafficking. By 2006, CTF-150 was primarily engaged in anti-piracy efforts off the coast of Somalia and within the Gulf of Aden. The Maritime Security Patrol Area (MSPA) was established



Adapted from- Planemad [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

Figure 5: A map covering Somali piracy from 2005-2010

in 2008 by United States Central Command (USCENTCOM) [33] within the Gulf of Aden to promote stability and increase security of the region. The MSPA outlines a suggested corridor for merchant vessels actively patrolled by Coalition Navy warships and aircraft led by the commander of CTF-150.

Combined Task Force 151 (CTF-151) was established on January 12, 2009 as a direct response to Somali piracy [34]. This task force was created under a series of mandates under the United Nations Security Council resolutions with the sole purpose of combating piracy without geographic constraints, operating under a different charter than CTF-150 [31]. Operation Atalanta is a counter-piracy operation initiated by the European Naval Force launched in December of 2008 to work cooperatively with CTF-151 in the Gulf of Aden, Arabian Sea, and Indian Ocean [35]. The North Atlantic Treaty Organization also established Operation Ocean Shield in August of 2009 [36] as part of OEF-HOA to protect the relief supply transports provided by the World Food Programme in the area which includes anti-piracy measures.

The most recent ongoing maritime successes in the fight against piracy are most easily attributed to cooperative anti/counter-piracy operations of CTF-151, Operation Atalanta, and Operation Ocean Shield which is shown in Table 5 below. An alarming number of boardings, hijacks, and attacks/attempted boardings occurred in 2011. The following years illustrate a dramatic reduction of piracy activity in the Horn of Africa, with piracy seemingly ineffective

as of 2013. The Office of Naval Intelligence routinely publishes weekly Piracy Analysis and Warning reports detailing any piracy incidents and potential threats in the area. Although it appears piracy has been practically eliminated in the Horn of Africa, continued operations ensure it does not return and destabilize the region.

Table 5: A table outlining the decline in piracy for the Horn of Africa from January 2011 through December 2nd, 2015 [37]

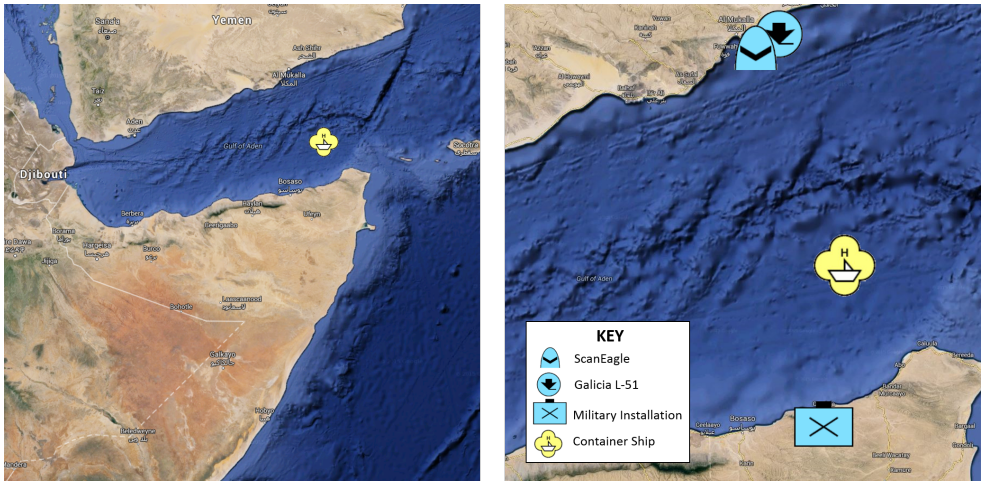
Summary of Horn of Africa Piracy Events					
Vessels	**2015**	**2014**	**2013**	**2012**	**2011**
Hijacked	0	0	0	7	27
Boarding	0	0	0	1	17
Fired Upon/Attemped Boarding	0	2	9	24	122
Total	0	2	9	32	166

4.2. Hypothetical Scenario

Consider the scenario where a small container ship in the Gulf of Aden has manually signaled distress using an Emergency Position-Indicating Radio Beacon (EPIRB). On-board GPS tracking was disabled shortly after the initial EPIRB broadcast. The last position update was approximately 30 minutes ago with a 2km precision from the last point of broadcast. The emergency contacts on file from EPIRB registration are unresponsive and the status of the the container ship is unknown. The features of this event are characteristic of a Somali piracy hijack suggesting the need for military intervention.

Coalition warships are presently out of range for immediate Search and Rescue (SAR) and VPSS operations increasing the urgency for response. As part of the European Naval Force (EU-NAVFOR) in Somalia under Operation Atalanta, the Spanish ship Galacia L-51 has been tasked as the first wave of response to the possible hijacking. The Galicia L-51 amphibious assault ship is outfitted with small, long-endurance ScanEagle UAV for autonomous surveillance within the area of operation. EU-NAVFOR-ATALANTA command has advised the Galicia L-51 to deploy its ScanEagle in the Gulf of Aden to fulfill information deficits, support ongoing missions, and ultimately provide more information to interdiction forces regarding the possible location of the distressed container ship. The last known location of the container

ship in the Gulf of Aden is shown in the left image of Figure 6, while operational assets are shown in the right image relative to the distressed ship.



Imagery ©2015 Data SIO, NOAA, U.S. Navy, NGA, GEBCO, Landsat, Map data ©2015 Google

Figure 6: The Somali coast with indicator symbol of a hijacked vessel of unknown status (left) and an enhanced image in the area of operation of the hypothetical SAR operation with military resources (right)

A area of $1200km^2$ is considered the immediate search area based on the container ship's last known location, the precision of EPIRB GPS location, the time to dispatch resources, and the current weather conditions. The operational area is expanded beyond this distance to account for lead times in deploying resources, the unpredictability of pirates, and to establish better situational awareness of the area outside of the MSPA corridor.

4.3. Scenario Execution

The scenario translates to an approximately $130000km^2$ area of operation which extends from the mid-gulf to the Arabian Sea, touching both coasts of Yemen and Somalia. This effectively becomes the macrogrid for platform routing containing 16 microgrids available for platform tasking which is presented in Figure 7. Mission parameters outline a maximum search time of 13.5 hours which allows the ScanEagle to travel upwards of $2025km$ at maximum operating speed. Search time within a microgrid is considered negligible considering operating speed of the ScanEagle allows for reasonable

sensor coverage of an area. The ScanEagle must originate from the Galacia L-51 and proceed to a local military base at the end of its mission.

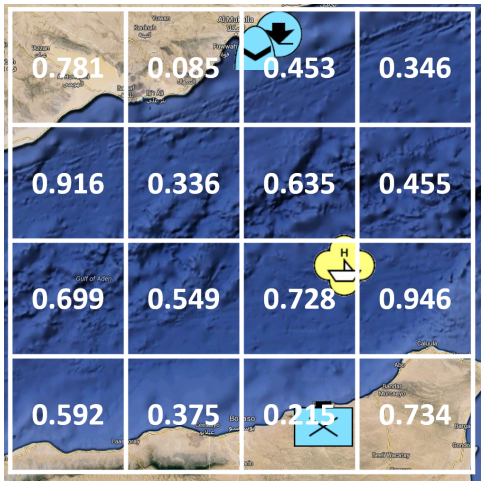


Figure 7: An information gain overlay of the area of operation for the scenario capturing the relative importance of information in each microgrid

The PCVR model is initialized with $T_{max} = 13.5$, $T = 48$, $C = 3$, the source vertex at the location of the Galacia L-51, and a sink vertex at the military installation on the coast of Somalia. The operating conditions estimate sensor effectiveness to 40%, setting $S = 0.40$. The associated travel times t_i, j are based off estimates, using the Euclidean distances between microgrids divided by the maximum cruising speed of the ScanEagle. Although the ScanEagle is a fixed-wing aircraft, its agility in a large operating environment such as this suggest loiter times are cheaper on route cost rather than visiting a nearby microgrid and returning.

The results of the PCVR model conducted on this graph using the outlined parameters are shown in the left image of Figure 8. The total information collection along this route results an objective of 3.3342 with an estimated route duration of 13.4 hours. The S2-PCVR route shown in the right has an identical objective value but an estimated route duration of 12.7 hours. The S2-PCVR model was able to save 42 minutes off the original route by resequencing the waypoints at the end of the route and allow the ScanEagle to loiter at the third last microgrid for an additional collection,

since it has the highest value across the entire map.

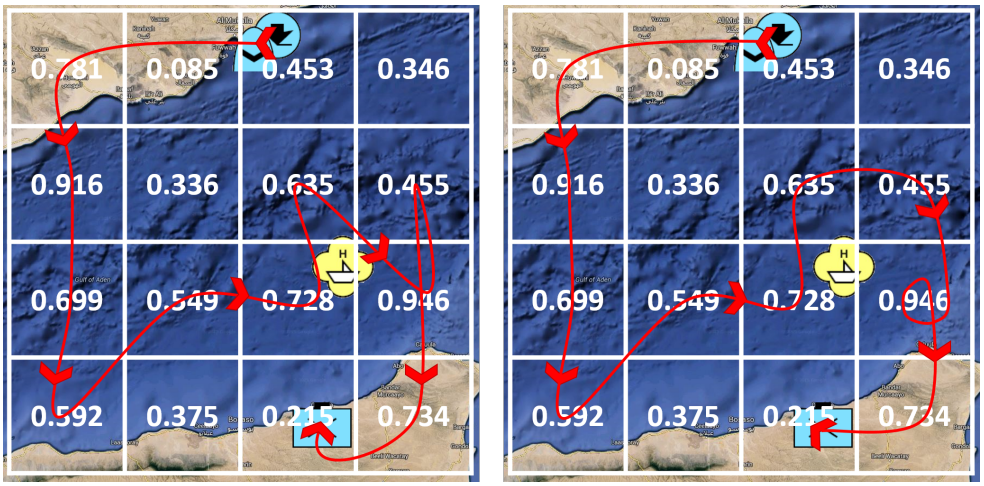


Figure 8: The prescribed PCVR route taking off from the Galacia L-51 and landing at a base on the Somali coast (left) and efficient S2-PCVR solution (right)

A hardware demonstration of this scenario using the PCVR model was conducted at the Lab for Autonomous and Intelligent Robotic Systems in the Mechanical & Aerospace Engineering Department of the University at Buffalo. A video demonstration including an overview of the laboratory may be found in the working paper section of `http://www.acsu.buffalo.edu/~batta`. This hardware demonstration establishes a proof of concept for this research in the field using real-world assumptions.

5. Conclusions

This paper had introduced a new interpretation of the prize-collecting and orienteering problems by incorporating partial and recurring collections along the route. The PCVR model is formulated as a mixed integer program to find exact solutions on small problem instances. Inexact solutions may be generated through early termination of the program or reducing model parameters to reduce the size of the model. The S2-PCVR model is formulated in similar notation, and takes the output of the PCVR model to generate a

shortest path between those routes. The convergence to optimality of problems approaching 50 vertices slows dramatically in part due to the size of the model as well as the complexity of the problem.

The TSP-H/SA heuristic was introduced to handle larger problem instances and provide instantaneous results using a parallelized architecture. The TSP-H component provides an initial route, finding a time-feasible subset highly-valued of vertices within the graph and sequencing the waypoints based on shortest distance. The TSP-H solution is then passed to a series of worker threads which execute the simulated annealing to improve both the route cost and objective value. The simulated annealing continually iterates between independent cost improvement and objective improvement components, serving as a means to agitate the solution, reset the candidate route to the best known solution, and to allow the heuristic to complete multiple cooling cycles. This procedure offers and aggressive solution procedure to generate acceptable routes in a practical setting where solutions may be needed at any time.

The PCVR and TSP-H/SA model were benchmarked against each other on a set of small problem instances to understand limitations of the approaches and establish performance baselines. The TSP-H/SA outperformed the PCVR on 85 of the 120 possible comparisons suggesting that the heuristic is preferable on smaller problem instances. However, the randomness of the simulated annealing procedure has no guarantees on convergence while the PCVR model may appear slower, it is far more predictable as a mixed integer program. The TSP-H/SA heuristic decisively surpasses the PCVR model since the heuristic scales to larger problem instances. Although the TSP-H/SA heuristic can load and solve graphs with over 10000 vertices, there is a decreasing marginal return on solution quality for graphs larger than 2500 vertices when considering the same solution parameters and solve times.

A major contributing factor in this research is its relevance in field applications, reducing the number of limiting assumptions made by these approaches. The PCVR model and TSP-H/SA model are two methods to generate routes in a time-sensitive environment. A scenario regarding Somali piracy was chosen to be the background for a hardware demonstration and test case in which these solution procedures may be useful. Given the relative importance of a series of grids within an area of operation, the PCVR model was able to generate an information-maximizing route given the mission parameters. The performance of the model and heuristic in the computational

study along with their relevance to autonomous UAV operation allows us to conclude that our results are satisfactory.

6. Acknowledgement

The authors are grateful to the Lab for Autonomous and Intelligent Robotic Systems in the Mechanical & Aerospace Engineering Department of the University at Buffalo for providing a platform to conduct hardware demonstrations. We also acknowledge the University at Buffalo's Center for Computational Research for allowing us access to their high powered computing resources. This research has been partially supported by the Office of Naval Research through CUBRC, Inc.

[1] I.-M. Chao, B. L. Golden, E. A. Wasil, A fast and effective heuristic for the orienteering problem, European Journal of Operational Research 88 (3) (1996) 475 – 489. doi:http://dx.doi.org/10.1016/0377-2217(95)00035-6.
URL http://www.sciencedirect.com/science/article/pii/0377221795000356

[2] T. Tsiligirides, Heuristic methods applied to orienteering, The Journal of the Operational Research Society 35 (9) (1984) pp. 797–809.
URL http://www.jstor.org/stable/2582629

[3] G. Laporte, S. Martello, The selective travelling salesman problem, Discrete Applied Mathematics 26 (2) (1990) 193 – 207. doi:http://dx.doi.org/10.1016/0166-218X(90)90100-Q.
URL http://www.sciencedirect.com/science/article/pii/0166218X9090100Q

[4] S. E. Butt, T. M. Cavalier, A heuristic for the multiple tour maximum collection problem, Comput. Oper. Res. 21 (1) (1994) 101–111. doi:10.1016/0305-0548(94)90065-5.
URL http://dx.doi.org/10.1016/0305-0548(94)90065-5

[5] I.-M. Chao, B. L. Golden, E. A. Wasil, The team orienteering problem, European Journal of Operational Research 88 (3) (1996) 464 – 474. doi:http://dx.doi.org/10.1016/0377-2217(94)00289-4.
URL http://www.sciencedirect.com/science/article/pii/0377221794002894

[6] P. Vansteenwegen, W. Souffriau, D. V. Oudheusden, The orienteering problem: A survey, European Journal of Operational Research 209 (1) (2011) 1 – 10. doi:http://dx.doi.org/10.1016/j.ejor.2010.03.045.
URL http://www.sciencedirect.com/science/article/pii/S0377221710002973

[7] B. L. Golden, L. Levy, R. Vohra, The orienteering problem, Naval Research Logistics (NRL) 34 (3) (1987) 307–318. doi:10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D.
URL http://dx.doi.org/10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

[8] W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Berghe, D. V. Oudheusden, A personalized tourist trip design algorithm for mobile tourist guides, Appl. Artif. Intell. 22 (10) (2008) 964–985. doi:10.1080/08839510802379626.
URL http://dx.doi.org/10.1080/08839510802379626

[9] Q. Hu, A. Lim, An iterative three-component heuristic for the team orienteering problem with time windows, European Journal of Operational Research 232 (2) (2014) 276 – 286. doi:http://dx.doi.org/10.1016/j.ejor.2013.06.011.
URL http://www.sciencedirect.com/science/article/pii/S0377221713004803

[10] N. Labadie, R. Mansini, J. Melechovsk, R. W. Calvo, The team orienteering problem with time windows: An lp-based granular variable neighborhood search, European Journal of Operational Research 220 (1) (2012) 15 – 27. doi:http://dx.doi.org/10.1016/j.ejor.2012.01.030.
URL http://www.sciencedirect.com/science/article/pii/S0377221712000653

[11] S.-W. Lin, V. F. Yu, A simulated annealing heuristic for the team orienteering problem with time windows, European Journal of Operational Research 217 (1) (2012) 94 – 107. doi:http://dx.doi.org/10.1016/j.ejor.2011.08.024.
URL http://www.sciencedirect.com/science/article/pii/S037722171100765X

[12] S.-W. Lin, Solving the team orienteering problem using effective multi-start simulated annealing, Applied Soft Computing 13 (2) (2013) 1064 – 1073. doi:http://dx.doi.org/10.1016/j.asoc.2012.09.022.
URL http://www.sciencedirect.com/science/article/pii/S1568494612004425

[13] J. Aràoz, E. Fernàndez, C. Zoltan, Privatized rural postman problems, Computers & Operations Research 33 (12) (2006) 3432 – 3449, part Special Issue: Recent Algorithmic Advances for Arc Routing Problems. doi:http://dx.doi.org/10.1016/j.cor.2005.02.013.
URL http://www.sciencedirect.com/science/article/pii/S0305054805000687

[14] C. Archetti, D. Feillet, A. Hertz, M. G. Speranza, The undirected capacitated arc routing problem with profits, Computers & Operations Research 37 (11) (2010) 1860 – 1869, metaheuristics for Logistics and Vehicle Routing. doi:http://dx.doi.org/10.1016/j.cor.2009.05.005.
URL http://www.sciencedirect.com/science/article/pii/S0305054809001452

[15] J. Aràoz, E. Fernàndez, O. Meza, Solving the prize-collecting rural postman problem, European Journal of Operational Research 196 (3) (2009) 886 – 896. doi:http://dx.doi.org/10.1016/j.ejor.2008.04.037.
URL http://www.sciencedirect.com/science/article/pii/S0377221708003974

[16] J. Aràoz, E. Fernàndez, C. Franquesa, The clustered prize-collecting arc routing problem, Transportation Science 43 (3) (2009) 287–300. arXiv:http://dx.doi.org/10.1287/trsc.1090.0270, doi:10.1287/trsc.1090.0270.
URL http://dx.doi.org/10.1287/trsc.1090.0270

[17] D. Black, R. Eglese, S. Wohlk, The time-dependent prize-collecting arc routing problem, Computers & Operations Research 40 (2) (2013) 526 – 535. doi:http://dx.doi.org/10.1016/j.cor.2012.08.001.
URL http://www.sciencedirect.com/science/article/pii/S0305054812001669

[18] J. Crino, J. Moore, J. Barnes, W. Nanry, Solving the theater distribution vehicle routing and scheduling problem us-

ing group theoretic tabu search, Mathematical and Computer Modelling 39 (68) (2004) 599 – 616, defense transportation: Algorithms, models, and applications for the 21st century. doi:http://dx.doi.org/10.1016/S0895-7177(04)90543-2.
URL http://www.sciencedirect.com/science/article/pii/S0895717704905432

[19] J. Ryan, T. Bailey, J. Moore, W. Carlton, Reactive tabu search in unmanned aerial reconnaissance simulations, in: Simulation Conference Proceedings, 1998. Winter, Vol. 1, 1998, pp. 873–879 vol.1. doi:10.1109/WSC.1998.745084.

[20] M. Alighanbari, Y. Kuwata, J. How, Coordination and control of multiple uavs with timing constraints and loitering, in: American Control Conference, 2003. Proceedings of the 2003, Vol. 6, 2003, pp. 5311–5316 vol.6. doi:10.1109/ACC.2003.1242572.

[21] T. Shima, S. J. Rasmussen, A. G. Sparks, K. M. Passino, Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms, Computers & Operations Research 33 (11) (2006) 3252 – 3269, part Special Issue: Operations Research and Data Mining. doi:http://dx.doi.org/10.1016/j.cor.2005.02.039.
URL http://www.sciencedirect.com/science/article/pii/S030505480500095X

[22] V. K. Shetty, M. Sudit, R. Nagi, Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles, Computers & Operations Research 35 (6) (2008) 1813 – 1828, part Special Issue: {OR} Applications in the Military and in Counter-Terrorism. doi:http://dx.doi.org/10.1016/j.cor.2006.09.013.
URL http://www.sciencedirect.com/science/article/pii/S0305054806002255

[23] N. Bednowitz, R. Batta, R. Nagi, Dispatching and loitering policies for unmanned aerial vehicles, J Simulation 8 (1) (2014) 9–24.
URL http://dx.doi.org/10.1057/jos.2011.22

[24] F. Mufalli, R. Batta, R. Nagi, Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans, Computers & Operations Research 39 (11) (2012) 2787 – 2799.

doi:http://dx.doi.org/10.1016/j.cor.2012.02.010.
URL http://www.sciencedirect.com/science/article/pii/S0305054812000366

[25] R. Harder, R. Hill, J. Moore, A java universal vehicle router for routing unmanned aerial vehicles, International Transactions in Operational Research 11 (3) (2004) 259–275. doi:10.1111/j.1475-3995.2004.00457.x.
URL http://dx.doi.org/10.1111/j.1475-3995.2004.00457.x

[26] L. Evers, A. I. Barros, H. Monsuur, A. Wagelmans, Online stochastic {UAV} mission planning with time windows and time-sensitive targets, European Journal of Operational Research 238 (1) (2014) 348 – 362. doi:http://dx.doi.org/10.1016/j.ejor.2014.03.014.
URL http://www.sciencedirect.com/science/article/pii/S0377221714002288

[27] D. Johnson, L. McGeoch, The Traveling Salesman Problem: A Case Study in Local Optimization, John Wiley and Sons, Ltd., 1997, Ch. 2, pp. 215–310.

[28] A. Bowden, The economic cost of somali piracy 2011, oceans Beyond Piracy (2011).
URL http://oceansbeyondpiracy.org/sites/default/files/attachments/View%20Full%20Report_3.pdf

[29] T. Dagne, Somalia: Conditions and prospects for lasting peace (2009).
URL https://www.fas.org/sgp/crs/row/RL33911.pdf

[30] D. Axe, Somalia redux: a more hands off approach (2009).
URL http://object.cato.org/sites/cato.org/files/pubs/pdf/pa649.pdf

[31] J. Garamone, Pakistani admiral takes command of regional maritime task force (April 2006).
URL http://web.archive.org/web/20070109205251/http://www.defenselink.mil/news/Apr2006/20060424_4907.html

[32] S. Tucker, Encyclopedia of Middle East Wars, The: The United States in the Persian Gulf, Afghanistan, and Iraq Conflicts: The United States in the Persian Gulf, Afghanistan, and Iraq Conflicts, ABC-CLIO, 2010.

[33] U. N. F. C. C. Commander, Combined task force 150 thwarts criminal activities (September 2008).
URL http://www.cusnc.navy.mil/articles/2008/115.html

[34] A. Clark, Combined maritime forces (cmf) operations: Counter piracy operations, challenges, shortfalls and lessons learned (June 2009).
URL http://www.nato.int/structur/AC/141/pdf/PS-M/Combined%20Maritime%20Forces%20Ops.pdf

[35] E. U. C. Security, D. Policy, Eu naval operation against piracy (eunavfor somalia - operation atalanta) (February 2010).
URL http://www.consilium.europa.eu/uedocs/cmsUpload/100201%20Factsheet%20EU%20NAVFOR%20Somalia%20-%20version%2014_EN.pdf

[36] NATO, Counter-piracy operations (March 2015).
URL http://www.nato.int/cps/en/natohq/topics_48815.htm

[37] O. of Naval Intelligence, Horn of africa/gulf of guinea/southeast asia: Piracy analysis and warning weekly (paww) report for 25 november - 2 december 2015 (December 2015).
URL http://www.oni.navy.mil/Intelligence_Community/piracy/pdf/20151203_PAWW.pdf