Simultaneously Determining Ingress/Egress Points and Time Allocations for UAV Grid Routing

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 propose a method for simultaneously determining the best ingress and egress points and the time allocations for a series of linked Unmanned Aerial Vehicle (UAV) routing instances to maximize global information gain across all routes within an Area of Operation (AO). The area of operation is decomposed into a network of grids based on geographic location and then partitioned into a series of zones called macrogrids which each contain a set of grids representing numerically-valued areas of surveillance interest, or microgrids, which effectively become potential waypoints for the UAV. Given a sequence of macrogrids for an Intelligence, Surveillance, and Reconnaissance (ISR) mission, a series of heuristics are used to generate and score potential ingress and egress points which serve as in input to a mathematical allocation model to determine a set of parameters that improves global information collection.

Keywords: Optimization, modeling, binary programming, partitioning, selection

1. INTRODUCTION

The increased use of UAVs for commercial, civilian, and military applications spurred an increase of research in efficient means for deployment of these platforms. One limiting factor faced by many vehicle routing methods on a transportation network is the increased complexity of the routing problem as a function of the problem size. In the case of tactical surveillance, a UAV is often tasked with coverage over an expansive AO in a continuous routing environment and must methodically visit high-valued areas in a cost-effective manner.

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

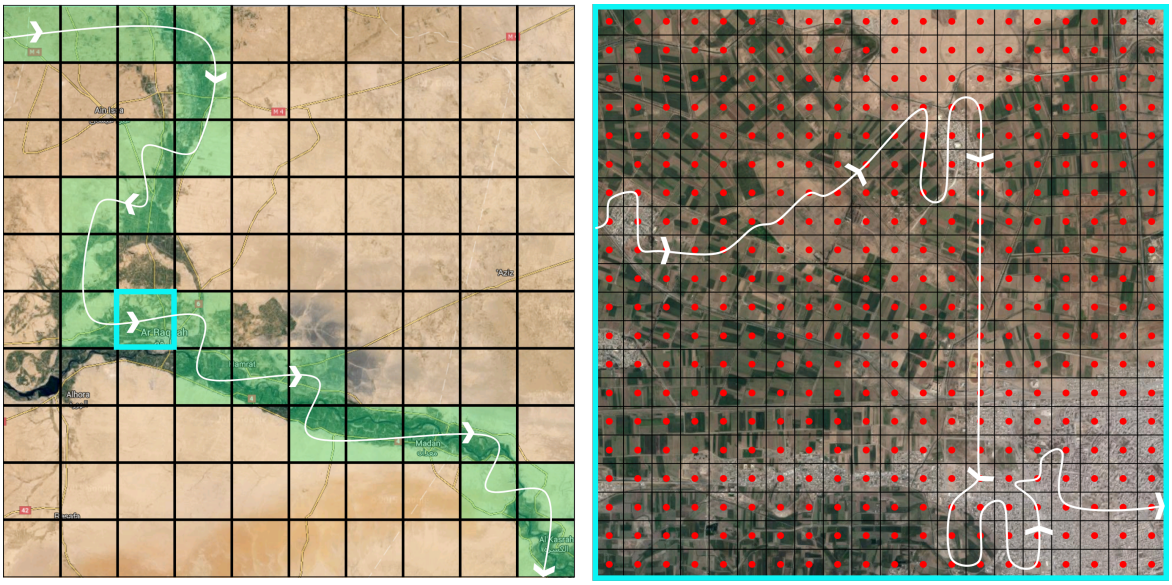
Partitioning the AO into a series of macrogrids allows the identification of high-potential areas for information collection and further microgrid discretization provides candidate waypoints with information values for route generation. With macrogrids sized appropriately to contain a manageable amount of microgrids allows the use of traditional vehicle routing solution methods to generate local routes within a macrogrid.

Generating an efficient or optimal route on a transportation network is a relatively well-studied problem in the literature. This application provides a method for partitioning the space into solvable problem instances, allocating cost resources across each subproblem, and linking the solutions to generate one continuous route. This approach allows the creation of high quality solutions for each partitioned graph. With the proper ingress and egress points and time allocations for each subproblem, a single path across all partitions is created with relatively high solution quality on a problem that would otherwise be considered computationally intractable.

The area of operation is represented as an undirected graph $G(V, E)$ broken up into a series of identically sized regions which are referred to as macrogrids. Each macrogrid contains a set of smaller, more detailed, and searchable regions represented by an independent subset of vertices called microgrids, with the vertex geographically placed at the microgrid's centroid with an assigned non-negative information value of I_j for vertex v_j. An example of an area of operation partitioned into a series of macrogrids is illustrated in the left image of Figure 1. The area of operation is overlayed with a set of 10 by 10 macrogrids covering over 5600 square miles over the Syrian city of Al-Raqqah, where each individual macrogrid is approximately 7.5 miles both tall and wide. The identified sequence of macrogrids for routing is highlighted in green with a white line annotated with arrows to indicate the progression of the UAV's flight through the sequence.

The image on the right within Figure 1 is a zoomed in satellite image of an individual macrogrid highlighted in teal on the map of the AO. This macrogrid, along with all other macrogrids in the area, is further overlayed with 20 by 20 microgrids and their centroids are shown by evenly spaced red dots on the map. It is at this level which routes are generated, using each microgrid centroid as a potential waypoint along the route to collect its corresponding information value. These centroids are effectively the vertices used in the translation of

the AO to the graph $G(V, E)$, and each vertex is assumed to have full-connectivity to all other vertices within the network. Given the macrogrid sequence, the route must originate from the left edge and terminate along the right edge. The ingress chosen for this grid is also chosen as the egress point to the previous grid on the macroroute, similarly, the egress point for this grid must also be the ingress point to the successive macrogrid in order to generate one continuous route across all subproblems.



Imagery ©2015 DigitalGlobe, Map data ©2015 Google

Figure 1: The area of operation (left) with a highlighted macrogrid sequence in green and an individual macrogrid (right) with 400 microgrids as available surveillance waypoints

Information gain values in this problem definition are defined as a non-negative utility values representing the magnitude of a vertex's importance of collection relative to all other vertices within the AO. Depending on the application, these information gain values may represent the need for aerial coverage for ongoing or future missions, deficits of information that impact situational awareness, or just simply, surveillance demand. These information gain values are again assumed provided along with the sequence of macrogrids for routing.

To further explain the concept of information gain, Figure 2 shows a comparison of raw satellite imagery of a macrogrid region overlayed with microgrid boundaries compared next to an information gain value heatmap of the region. The information gain values are represented numerically, however, colors are associated to the heat map for simpler visualization. Green

values in the heatmap are low-valued, relatively unimportant microgrids, orange grids are moderately valued, and red microgrids have the highest relative significance. In comparing the satellite imagery to the heatmap in this, areas with the largest surveillance benefits include roads, dense urban environments, and industrial regions. The areas consisting of desert and the rural outskirts of the city are considered less valuable. An optimal route across this region does not guarantee all or any high-priority microgrids are visited since their is a route cost constraint and the objective is to maximize the overall value of the route.

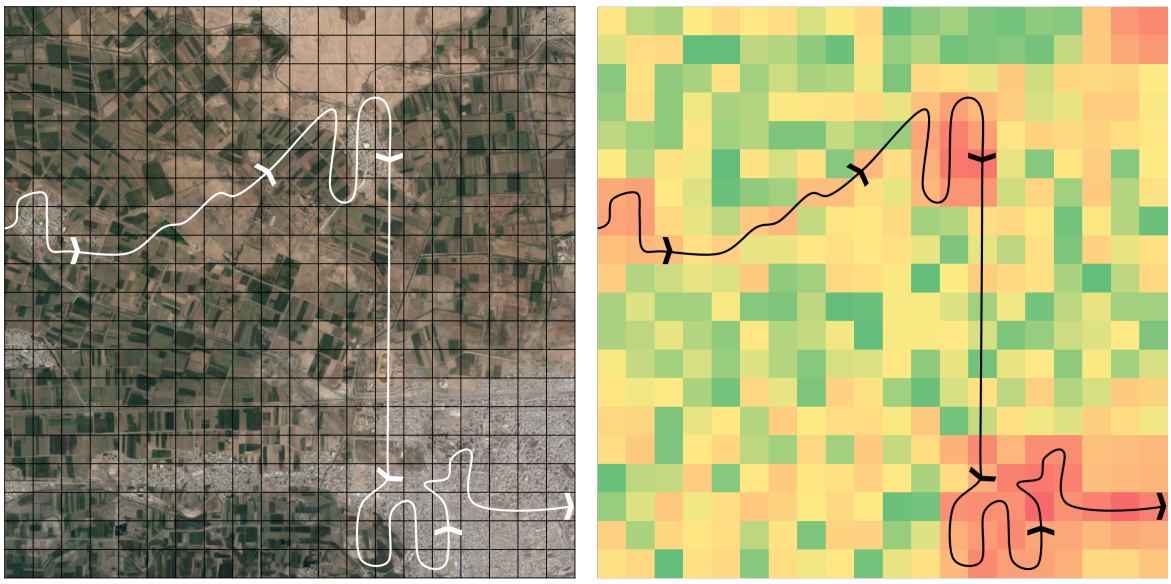


Figure 2: Satellite imagery of the selected macrogrid partioned into 400 microgrids (left) and its corresponding heatmap (right) visualizing information gain, both overlayed with an example route

1.1. ***Relevant Literature***

Assignment-based methods are often the core of many algorithms related to clustering and partitioning (Kearns et al., 2013). Every clustering method requires some level of initialization and allocation procedures (Lance and Williams, 1967) for both the computation and the desired operations within a cluster. Assignment methods frequently occur in the formulation of common optimization problems involving clustering and classification, most notably demonstrated by the Quadratic Assignment problem (Floudas and Pardalos, 2009).

Clustering methods are prevalent in the creation of ad-hoc data networks requiring robust efficient data routing across a large network infrastructure (Iwata et al., 1999). The creation of clustered communication networks is common in UAV applications (Gu et al., 2000b) (Gu et al., 2000a) to reduce bandwidth limitations and provide a lower-latency communication link between entities. These multi-hop networks transmit data across a sequence of communication clusters from the UAV to its ground control station and inversely as well. These communication nodes must first be clustered then an efficient transmission sequence must be established, both between and within the clusters. These clusters are often hierarchically generated through some multi-level partitioning algorithm (Hendrickson and Leland, 1995), although many other ad-hoc partitioning schemes exist each with their own advantages.

Set partitioning has been used to address a wide variety of optimization-based routing models (Cullen et al., 1981), improving solution quality through subproblem generation and routing. Nagy and Salhi (Nagy and Salhi, 2007) provide a comprehensive literature covering the issues, models, and methods pertaining to location routing. A section is dedicated to clustering-based routing literature used in a wide breadth of approaches to location routing problems. Partitioning and clustering is found useful in the General Pickup and Delivery problem and generic Vehicle Routing Problem (Savelsbergh and Sol, 1995) employing a master problem to iteratively improve cluster solution quality while also simultaneously improving the master solution. The use of set partitioning in vehicle routing problems may be regarded as an inexact approach (Toth and Vigo, 2001) when considering the loss of vital problem data through either improper partitioning or the inability for lossless set generation.

The Military Grid Reference System (MGRS) is a geocoordinate system defined by NATO standards to organize and identify grids and points on the earth (Lampinen, 2001). The MGRS is based off of the Universal Transverse Mercator (Langley, 1998) (Stott, 1977) and Universal Polar Sterographic (Hager et al., 1992) coordinate systems, both employing a cartesian-based coordinate projection. Sixty different grid-zones are designated across the surface of the earth represented as a series of latitude bands and each grid-zone may further partitioned using this standard with varying precision down to a 1 square meter grid. These grids are roughly square shaped, however, the projection of a 2-dimensional to the spherical surface of the earth results in grids that may only strictly be considered polygonal (Dana,

1999). Many other geocoordinate systems exist but the MGRS is a widely adopted system for both commercial and military applications with different degrees of grid-size locations.

In the context of this paper, a partition is considered a cluster of information bearing vertices in the area of operation that are converted to independent subgraphs for vehicle routing. The clustering designates groups of similarly geolocated vertices while the partitioning divides the area of operation with some loss of data since the clusters are no longer linked. The MGRS is one potential method for grid partitioning, using a combination of NATO and civilian standards to establish a grid structure with variable degress of precision. The use of vehicle routing methods to solve a set of partitioned problems require some master problem to determine appropriate allocation parameters to optimize across all partitions.

1.2. ***Problem Definition***

The area of operation is defined as an undirected graph $G(V, E)$ with each vertex representing a surveillance target or microgrid and the edges represent the travel costs between the targets. Each vertex $v_j \in V$ has an associated profit value I_j, where $I_j \geq 0 \; \forall \; v_j \in V$ captures the relative importance of visiting that vertex relative to other vertices within the network. The travel costs between vertices v_i and v_j are predefined as $t_{i,j} \in E$ for a specific vehicle. The parameter T_{MAX} defines the maximum allowed traveled cost along a path $P = (v_1, v_2, \ldots, v_n) \in V$ within the graph. The overarching objective in ISR-based routing problems is to find a path R along $G(V, E)$ that maximizes total profit collection but does not exceed a specified cost T_{MAX} which is usually linked to vehicle endurance or the time-sensitivity of improving situational awareness in the AO. The overall problem is framed as an orienteering or prize-collecting path generating problem, but the scope of this work focuses on creating manageable subproblems through graph partitioning.

The problem also allows for multiple profitable traversals to the same vertex in the instances of imperfect information collection with each successive visit to that vertex resulting in diminishing information potential. A UAV's sensor effectiveness S_a captures the percentage of information collected at each vertex visit in macrogrid a which allows infinite vertex visits, however, in practice with time-constrained routing a high-valued vertex is only visited several times before its information value is depleted. This sensor effectiveness term does not

affect the allocation model but it is a necessary in regards to the scoring methods and results of the model.

It is assumed that the graph $G(V, E)$ is too large for traditional routing methods. The graph $G(V, E)$ is broken into a series of subgraphs (or partitions) $H_a(V_a, E_a) \in O$ where O represents a set of uniformily-sized neighboring subgraphs such that $H_a \cap H_b = \emptyset \; \forall \; \{H_a, H_b \in O | H_a \neq H_b\}$ and $|V_1, V_2, \ldots, V_{|O|}| = |V|$ and $|E_1, E_2, \ldots, E_{|O|}| < |E|$. In other words, the area of operation is discretized into a series of macrogrids and all vertices within grid a are added to V_a. All edges between vertices in macrogrid a are kept but all connectivity to vertices outside of the macrogrid boundaries are removed, leaving set E_a. These elements create subgraph $H_a(V_a, E_a)$ which is independent from all other subgraphs and all vertices contained in O span graph $G(V, E)$ with a reduced edge set.

The problem is stated as follows: given a total mission time T_{MAX} and sequence of neighboring macrogrids M, where $M \subset O$, simultaneously determine a set of entry and exit points linking each macrogrid $H_g(V_g, E_g) \in M$ while allocating maximum local route costs for each macrogrid, t_g, such that $\sum_{g \in M} t_g + \epsilon_g \leq T_{MAX}$. Once a path leaves a macrogrid, return to that macrogrid is not permitted. We wish to use this partitioning method provided to create tractable subproblem instances while linking solutions to each subgraph from the ordered set M to identify ingress/egress points and time allocation parameters for each macrogrid to increase information collection across the entire macroroute.

2. MATERIAL AND METHODS

The first model introduced is the Macrogrid Allocation Model formulated as a Nonlinear Program (MGAM-NLP) covered in Subsection 2.1.1. The MGAM-NLP formulation considers information collection sampled over continuous time. The difficulty in solving nonlinear mathematical programs particularly in time-sensitive optimization environments motivated the creation of the Macrogrid Allocation Model formulated as a Binary Program (MGAM-BP) which is covered in Subsection 2.1.2. The MGAM-BP formulation is a parameter selection program which is the core component to this research. The formulation is presented as a replacement to the MGAM-NL since it reduces the computational resources to determine adequate allocation parameters but also requiring intelligent sampling methods.

The scoring methods outlined in Subsection 2.2 generate input data for the MGAM-BP model. The Quick Scoring Method is one of the most lightweight models possible for generating input score data by leveraging the problem definition to determine allocation parameters with high potential for information collection. The Greedy Ratio Traveling Salesman Problem Heuristic is provided as an alternative scoring method which generates actual routes across the macroroute and utilizes the information collection objectives of each route as an input datum point. The performance of these models in terms of solve time and solution quality are characterized in Section 3.

2.1. ***Macrogrid Allocation Models***

The macrogrid allocation models require a score capturing the potential information gain for a macroroute $H_g(V_g, E_g) \in M$ given as a function of ingress/egress points combined with either discrete time samples or a continuous time allocation function for each macrogrid on the route. An optimal solution is linked to these allocation samples and an optimal solution from the allocation models result in the best combination of ingress and egress points for each grid along with time allocations to sequence a single route with optimal information gain given the input sets.

Both models discretize neighboring edges along the macroroute to determine a single vertex to bridge graphs $H_g(V_g, E_g)$ and $H_{g+1}(V_{g+1}, E_{g+1})$ in M. The set of ingress and egress points for macrogrid g is denoted P_g^α and P_g^Ω, respectively. In order to keep a continuous route across all macrogrids, $P_g^\Omega = P_{g+1}^\alpha$ since $|M| - 1$ edges are shared between macrogrids. The complement of the ingress and egress set intersection across the macroroute, $(\{P_g^\alpha \cap P_g^\Omega : g \in H_g(V_g, E_g)\})^c = \{P_1^\alpha,\ P_{|M|}^\Omega\}$, allows this notation to uniquely identify the ingress and egress edge sets, specifically at the beginning and end of the macroroute. The edge set elements are better shown in Figure 3. This notation is critical for the creation of intuitive macrogrid allocation models.

2.1.1. Nonlinear Allocation Program

The MGAM-NL formulation is relevant in the application of a UAV continuously collecting information on its mission. This model assumes the sensor is always on and collecting information of a region even between waypoints. Net information collection is non-decreasing

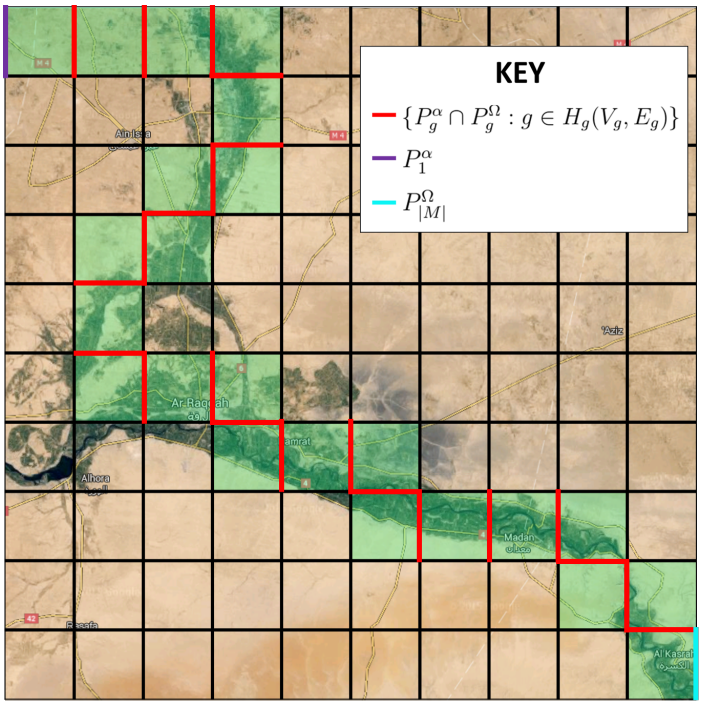


Figure 3: A visual representation of the edges sampled for the macrogrid allocation models and their set notations

as more time is allocated to a region since $I_j \geq 0 \; \forall \; v_j \in V$ for the entire area of operation. Focusing on an individual macrogrid $H_g(V_g, E_g)$, the net information collection as a function of the time allocated generally follows an S-shaped curve for optimal information collection route between points $p \in P_g^\alpha$ and $q \in P_g^\Omega$ which is shown in Figure 4.

The net continuous information curve has three distinguishable ranges annotated in the figure: (1) a warm up period, (2) a period of strong increasing returns on information gain, and (3) a period of diminishing returns. The warm up period ranges from value t_{min}^g, the minimum designated search time for the grid, to the the halfway point to the inflection point, often with a slow but increasing return on information collection as more time is allocated. The y-intercept of the graph typically represents the information collected when the UAV immediately travels to the egress point upon entering the grid. In many cases, a route immediately exiting a macrogrid does not include the most profitable regions and more time is required to reach profitable targets within the region resulting in some warm up time to significantly increase route value. The most profitable range for adjusting time allocation occurs in the period centered around the inflection point resulting in the greatest increases

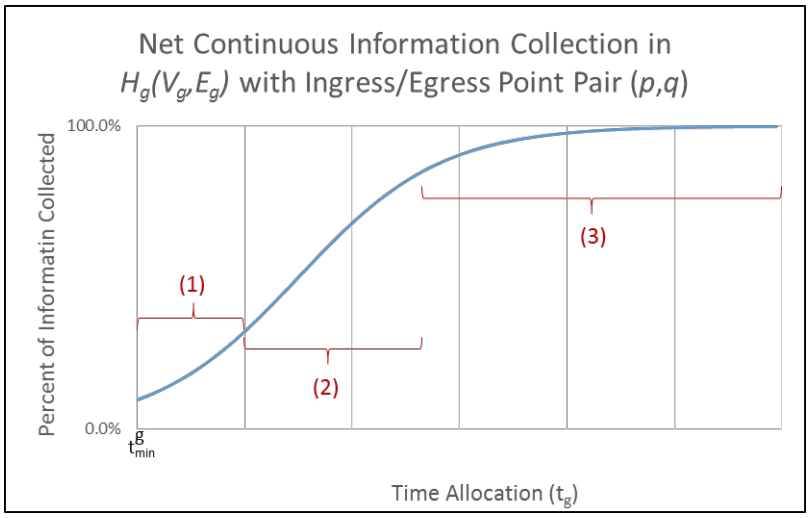


Figure 4: The percentage of continuous information gain as a function of time allocation t_g for macrogrid $H_g(V_g, E_g)$, ingress point $p \in P_g^\alpha$, and egress point $q \in P_g^\Omega$

of returns. The time allocation t_g in this range allows the UAV to reach high-valued regions and any additional time allotted results in a relatively significant increase in information collection for the macrogrid. When sensor effectiveness is included in the routing models where $S \neq 1$, there comes a point where the return on net information gain by increasing t_g is diminishing. With larger time allocations, high-valued vertices are revisited with reduced information potential, low-valued vertices become more significant on the route, and the problem is more focused on exhaustive routing rather than efficient.

The information contained in net continuous information curves contains all of the data necessary for the MGAM-NLP model. This data must be generated for combination of macrogrids, time allocations, ingress, and egress points. These curves must also be a function of continuously increasing the time allocation which is simply not a reasonable approach in conventional routing optimization problems without appropriate regression techniques. The model itself is somewhat easy to understand, could theoretically result in optimal solutions, and provides a foundation for the MGAM-BP model.

The nonlinear objective for the MGAM-NLP model is given in Objective Equation 1 which aims to maximize global information gain across all macrogrids on the route through the optimal selection of discrete ingress/egress points and continuous time allocations. The binary decision variable $r^g_{p,q}$ is set to one if ingress point p and egress point q are chosen

for grid g, zero otherwise. Decision variable t_g is the time allocated for that grid, where $t_g \geq 0 \; \forall \; g \in M$. The function $f_{r^g_{p,q}}(t_g)$ is the net continuous information gain curve shown in Figure 4, which is a nonlinear function of the decision variable t_g. The value of $f_{r^g_{p,q}}(t_g)$ impacts the objective only if $r^g_{p,q} = 1$.

Constraint 2 is the macroroute cost constraint stating that the sum of all time allocations for each macrogrid must not exceed the global route cost upper bound T_{MAX}. The parameter ϵ_g is a route cost error term added for each grid to avoid exceeding T_{MAX} in real routing environments. This term may account for uncertainty in the planned route cost or minor retasking of the UAV mid-mission. Constraint 3 is a flow balance equation linking elements in sets P^α_g and P^Ω_g. It states that for any given egress point q, it must be both an egress point on macrogrid g and an ingress point on macrogrid $g+1$. Similarly, any point p must flow through point q on the route and exit to some point l on the macrogrid sequence. This flow balance constraint is responsible for linking solutions on each subgraph $H_g(V_g, E_g) \in M$. Constraint 4 states that only one ingress/egress pair must be selected for each macrogrid. The equality is necessary, forcing all macrogrid assignments of an ingress and egress point.

$$\text{(MGAM-NLP)} \quad MAXIMIZE \sum_{g \in M} \sum_{p \in P^\alpha_g} \sum_{q \in P^\Omega_g} r^g_{p,q} f_{r^g_{p,q}}(t_g) \tag{1}$$

subject to

$$\sum_{g \in M} t_g + \epsilon_g \leq T_{MAX} \tag{2}$$

$$\sum_{p \in P^\alpha_g} r^g_{p,q} = \sum_{l \in P^\Omega_{g+2}} r^{g+1}_{q,l} \qquad \forall \quad \{g \in M | i \neq |M|\} \quad q \in P^\Omega_i \tag{3}$$

$$\sum_{p \in P^\alpha_g} \sum_{q \in P^\Omega_g} r^g_{p,q} = 1 \qquad \forall \quad g \in M \tag{4}$$

$$r^g_{p,q} \in \{0, 1\}, \quad t_g \geq 0 \; \forall \; g \in M \tag{5}$$

Two strong limitations of this model are the proper generation of score values $f_{r^g_{p,q}}(t_g)$ and the nonlinear objective. If the function of $f_{r^g_{p,q}}(t_g)$ is truly considered a sigmoid function, the curve may be estimated by sampling four points across time for each macrogrid, ingress point, and egress point combination. Even then, $f_{r^g_{p,q}}(t_g)$ is still a function of decision variable

t_g which is multiplied by binary selection variable $r^g_{p,q}$. The use of disjunctive programming would linearize the objective at the cost of creating nonlinear constraints which is not necessarily easier to solve. The MGAM-BP is a discretized version of the MGAM-NLP and is more appropriate for solving the allocation problem.

2.1.2. Binary Allocation Program

The MGAM-BP formulation is conceptually identical to that of the MGAM-NLP except it considers discrete time samples and vertex collections. Figure 5 shows the net information for optimal information gain routes for discrete vertex collections with varying time allocations $t_a \in T_g$. The function still follows a similar S-shaped trend except with net information gain sampled randomly across a macrogrid. The estimated information gain for a grid g, ingress point p, egress point q for any $t_a \geq t^g_{min}$ can utilize either a linear piecewise interpolation or a piecewise constant step function between these points. The step function is the more conservative approach since it never overestimates information gain potential and is the recommended approach for input into the MGAM-BP model.

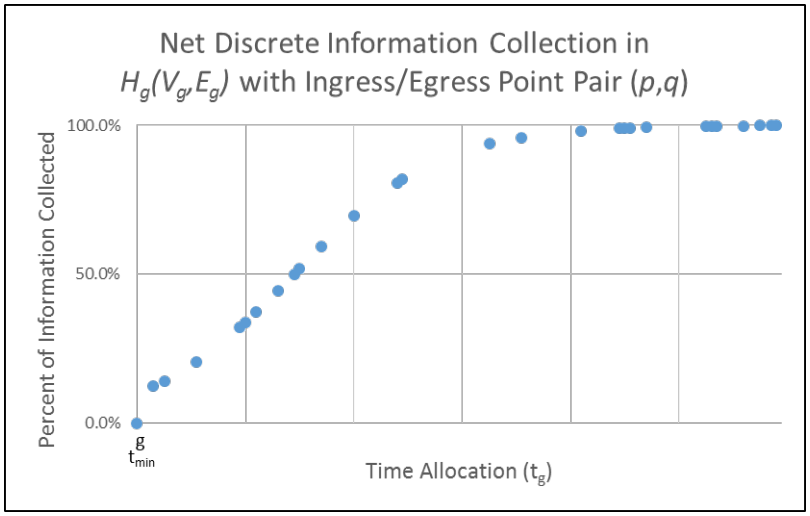


Figure 5: The percentage of information gain discretely sampled as a function of time allocation t_a for macrogrid $H_g(V_g, E_g)$, ingress point $p \in P^\alpha_g$, and egress point $q \in P^\Omega_g$

The objective of the MGAM-BP in Equation 6 maximizes global information gain using data from the sampled points from all data points generated for each macrogrid. Binary selection variable $r^g_{a,p,q}$ has the same interpretation of $r^g_{p,q}$ from the MGAM-NLP except it is

now indexed by time samples $a \in T_g$ for each macrogrid. The time allocation t_a is no longer a decision variable and is instead an element of the set T_g which includes a set of times sampled for routing within macrogrid $g \in M$. The value of $f^g_{a,p,q}$ is the discrete score for discrete time samples t_a.

Constraint 7 now links the binary selection variables $r^g_{a,p,q}$ to a time allocations t_a which in aggregate may not exceed T_{MAX} across the entire macroroute. The flow balance in Constraint 8 now includes summations for the time index without any change in interpretation. Constraint 9 now includes that only one time may be allocated to a grid, along with only allowing one ingress and egress point pair.

$$\text{(MGAM-BP)} \quad MAXIMIZE \sum_{g\in M}\sum_{a\in T_g}\sum_{p\in P^\alpha_g}\sum_{q\in P^\Omega_g} r^g_{a,p,q} f^g_{a,p,q} \tag{6}$$

subject to

$$\sum_{g\in M}\sum_{a\in T_g}\sum_{p\in P^\alpha_g}\sum_{q\in P^\Omega_g} r^g_{a,p,q} t_a + \epsilon_g \leq T_{MAX} \tag{7}$$

$$\sum_{a\in T_g}\sum_{p\in P^\alpha_g} r^g_{a,p,q} = \sum_{t\in T_{g+1}}\sum_{l\in P^\Omega_{g+2}} r^{g+1}_{a,q,l} \quad \forall \quad \{g \in M | i \neq |M|\} \quad q \in P^\Omega_i \tag{8}$$

$$\sum_{a\in T_g}\sum_{p\in P^\alpha_g}\sum_{q\in P^\Omega_g} r^g_{a,p,q} = 1 \quad \forall \quad g \in M \tag{9}$$

$$r^g_{a,p,q} \in \{0,1\} \; \forall \;\; g \in M, \; a \in T_g \; p \in P^\alpha_g, \; q \in P^\Omega_g \tag{10}$$

This slight change in approach results in a formulation that is far more computationally tractable than its nonlinear counterpart. The MGAM-BP is modeled as purely a binary program, removing time allocation as a decision variable and increasing the dimension of the binary selector to include time sampling. The solution quality of the MGAM-BP is very closely tied to the quality and coverage of the discrete samples $f^g_{a,p,q}$ outlined in Section 2.2.

2.2. ***Scoring Methods***

The scoring methods contained in this section outline a procedure for generating $f^g_{a,p,q}$ for the MGAM-BP formulation. A set for both the entry and exit points on grid g, P^α_g, q and P^Ω_g respectively, are generated for all $H_g(V_g, E_g) \in M$ along the macroroute. Each value

of $f^g_{a,p,q}$ is linked to the time allocation t_a for that macrogrid. The Quick Score Method (QSM) is a lightweight and relatively straightforward method for generating input into the MGAM-BP model. The Greedy Ratio Traveling Salesman Heuristic (GRTSP-H) employs the methodology contained in the QSM but the scores provided by the GRTSP-H are the results of actual routes which may be useful in constructing an initial macroroute.

2.2.1. Quick Score Method

The QSM initializes by first establishing a set of time priorities for each macrogrid. Each time priority p_g captures the potential information collection for each macrogrid g. The time priority is defined Equation 11 which calculates the percentage of information contained in a macrogrid relative to all other macrogrids contained on the macroroute.

$$p_g = \frac{\sum\limits_{v_j \in V_g} I_j}{\sum\limits_{z \in M} \sum\limits_{v_j \in V_z} I_k} \quad \forall \;\; H_g(V_g, E_g) \in M \tag{11}$$

All information gain values I_k are summed across the entire routing region contained in O. The priority p_g is established by summing all information values $I_j \in V_g$ and dividing it by the total possible information collection in the routing region. This serves as an effective time allocation method in problems with information gain values uniformly distributed across the region. These time priorities are less effective in regions with regions with locally clustered regions within a macrogrid of either high or low information potential since these areas can be easily collected sequentially or avoided all together. This percentage is converted from a percentage to an actual time allocation relative to parameter T_{MAX} in Equation 12.

$$t_g^{priority} = t_g^{min} + p_g * (T_{MAX} - \sum_{H_z \in M} t_g^{min}) \quad \forall \;\; H_g(V_g, E_g) \in M \tag{12}$$

The time values of t_g^{min} define the minimum search time required within a macrogrid, either defined as minimum time to enter and immediately exit a macrogrid or as a parameter defined by the mission planner. The calculation of $t_g^{priority}$ takes any surplus time defined as $(T_{MAX} - \sum_{H_z \in M} t_g^{min})$, scales that time according to its time priority and adds it to the minimum search time. The set T_g governs how many time samples are considered for an individual, where $|T_g| \geq 2$. For $|T_g| \geq 2$, this set contains both t_g^{min} and $t_g^{priority}$. The

parameter τ_g defines the maximum desired cardinality of T_g. For cardinality exceeding two for the defined number of time samples in a macrogrid, additional time values $a_{i,g}$ are created according to Equation 13. The paramater s_g defines the maximum percentage of the surplus time allowed for the grid, where $s_g \in (0, 1]$.

$$a_{i,g} = t_g^{min} + i * s_g * \frac{(T_{MAX} - \sum_{H_z \in M} t_g^{min})}{\tau_g} \quad \forall \; i = 1, .., (\tau_g - 2), \;\; H_g(V_g, E_g) \in M \tag{13}$$

The values of $a_{i,g}$ are members of set T_g where the value τ_g defines how intensively the surplus time is explored. As $\tau_g \to \infty$, the solution space is explored in continuous time similar to Figure 4 in Subsection 2.1.1. This comes at both a cost to the allocation model and scoring procedures, requiring some care in balancing the solution quality and solve times associated with the time sampling.

Determining the set of candidate edge points shares some similarity to the creation of $a_{i,g}$ values. Given the set M as a sequence partnered with the bounding points for each macrogrid allows the identification of the ingress/egress edges. The parameter γ_g defines the desired cardinality of sets P_{g+1}^{α} and P_g^{Ω}, and γ_0 is the cardinality of P_1^{α}. Notation for individual waypoints on an edge are annotated as ingress point $\alpha_p \in P_g^{\alpha}$ and egress point $\Omega_q \in P_g^{\Omega}$. If $\gamma_g = 1$, the entry point is placed on the midpoint of the ingress/egress edge and if $\gamma_g = 2$, two points are spaced one third the edge length from each edge end leaving one third the edge length distance between the points. For $\gamma_g \geq 3$, an edge point is added for each corner along the ingress/egress edge and all other points are evenly spaced out along the edge. Determining favorable points for P_g^{α} and P_g^{Ω} is not trivial, thus sets of points are chosen to explore as much of the edge is possible without introducing a bias that could potentially narrow the optimization procedure to a suboptimal region.

The last step of the QSM involves generating scores values $f_{a,p,q}^{g}$. The method to generating these scores are shown in Equation 14 below. For a given vertex in a macrogrid, its corresponding information gain value is divided by its distance to both the candidate ingress and egress point. The maximum value between these two ratios are summed across all other vertices within the macrogrid to effectively generate a closeness score of high-valued vertices and their proximity to either the ingress or egress point. This closeness score is then

scaled by the allocated search time t_a divided by the grids time priority $t_g^{priority}$. This term is a multiplier which increases the score for time allocations exceeding the calculated time priority and discounting time allocations that are shorter.

$$f_{a,p,q}^{g} = \frac{t_a}{t_g^{priority}} \sum_{v_j \in V_g} max(\frac{I_j}{cost(\alpha_p, v_j)}, \frac{I_j}{cost(\Omega_q, v_j)}) \quad \forall \ g \in M, \ \ a \in T_g, \ \ p \in P_g^{\alpha} \ \ q \in P_g^{\Omega} \tag{14}$$

In short, this scoring method identifies the ingress and egress points that are the close to highly-profitable vertices and the time multiplier allows the allocation of more time to a macrogrid's candidate ingress/egress pair if it is likely to collect those points on the route. This method is best used with information hotpots clustered closely to a potential ingress or egress point or when the information gain is distributed uniformily across a region. The method is limited in its ability considering there is no formal link between the actual collection of information and the allocation parameters. The QSM has less-intuitive values of $f_{t,p,q}^{g}$, but in return the scoring procedure is computationally efficient, the approach is straightforward, and the scores retain some meaning in identifying a strong allocation parameter set.

2.2.2. Greedy Ratio Traveling Salesman Heuristic

The GRTSP-H leverages the backend of the QSM, utilizing the same procedures to generate allocation sets T_g, P_g^{α}, and P_g^{Ω}. The GRTSP-H is designed to be a route generation heuristic with reasonably fast solve times and ultimately deliver a route for each macrogrid, time allocation, and ingress and egress point pair combination with a corresponding objective value in units of information collected.

For each macrogrid allocation combination, the GRTSP-H initializes with time allocation $t_a \in T_g$, ingress point $\alpha_p \in P_g^{\alpha}$, and egress point $\Omega_q \in P_g^{\Omega}$ on grid $H_g(V_g, E_g) \in M$. Waypoint α_p is designated as the start of the GRTSP-H route R_g and Ω_q is the last waypoint with a feasible path generated using only these two points. The next waypoint considered for route R_g is determined using the method in Equation 15.

$$max_{\{j \in V_g | v_j \notin R_g\}}(\frac{I_j}{min(cost(v_j, \{v_r \in R_g | v_r \neq v_j\}))}) \tag{15}$$

The initial route sequence R_g contains only elements $\{\alpha_p, \Omega_q\}$. For each vertex $v_j \in V_g$ for a specific macrogrid, there vertex's corresponding information gain value I_j is divided by the minimum travel cost to any element within the route sequence R_g. The vertex with the highest profit-to-cost ratio is then chosen as the next possible waypoint added to R_g. The candidate waypoint is added to the route if and only if a feasible route is generated with the inclusion of the new waypoint. Waypoints are sequentially added until the new vertex candidates create cost-infeasible solutions.

Routes are generated using a greedy traveling salesman heuristic which is often referred to as the nearest-neighbor approach (Rosenkrantz et al., 1977). Given all elements in R_g along with the candidate waypoint, a new route R_g^{new} is initialized starting with source vertex α_p. The waypoint with the lowest travel cost to the last waypoint added on the route is next vertex visited with ties broken randomly. The last vertex added to the route is the designated egress point, Ω_q. Once this new route is created, a feasibility and slack test is performed. If the new route cost does not exceed the time allocation t_a, the candidate is accepted, $R_g = R_g^{new}$, and additional candidates are explored using the same procedure. However, if the cost of R_g^{new} exceeds t_a, no new route is accepted and the current sequenced set of R_g is chosen as the final GRTSP-H solution.

The objective value for each finalized GRTSP-H route becomes MGAM-BP input $f_{a,p,q}^g$. Each macrogrid creates $|T_g|*|P_g^\alpha|*|P_g^\Omega|$ subproblems for the heuristic, each requiring a feasible route. The subproblems require its own copy of the macrogrid, and the iterative process of evaluating candidate waypoints and updating the route comes with some computational overhead. This procedure is still relatively lightweight compared to more complex vehicle routing models, but one iteration of the candidate waypoint evaluation in the GRTSP-H can require more CPU cycles than an entire set of QSM scores for all combinations of parameters in a macrogrid. One advantage of the GRTSP-H scoring method over the QSM is that it adds waypoints sequentially to a route and new candidates are selected based on the elements of this dynamic set wheras in the QSM, scores are based exclusively to their proximity to α_p and Ω_q. The GRTSP-H ultimately provides an actual route with an objective in terms of net information collection and an associated cost making it a preferred method over the QSM if the total allowed score time permits it.

3. RESULTS

A full-factorial study is conducted on the MGAM-BP to benchmark its performance with varying numbers of ingress/egress points and discrete time samples. The QSM and GRTSP-H scoring methods are used alongside each other and their scores are input into the MGAM-BP, and the allocation output from the MGAM-BP solutions are then reevaluated to see which method provides better results. The data emulator for the AO and scoring methods are developed in Java and the MGAM-BP is modeled using Concert Technology for Java utilizing the CPLEX 12.6 library. No optimization optimization or solution parameters are modified from the default CPLEX parameters. 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 provide the operating system with 1GB of memory.

Instead of generating an entire AO, only a sequence of macrogrids M is created. Each macrogrid $H_g(V_g, E_g) \in M$ is assumed to have the same 10 unit by 10 unit dimension. The number of microgrids, $|V_g|$, remains constant within a study and must be a perfect square to ensure the microgrids are squarely shaped and the density of microgrids within a macrogrid is uniform. Vertices for all microgrids are placed at their centroid and must be visited in order for a collection to occur. The Euclidean distance is used between travel between vertices and a search time d_g is imposed for any information I_j collected on vertex v_j where $d_g = 1.25 * \sqrt{|V_g|}$. The sensor effectiveness is set to 50% ($S = 0.50$) although this only affects the scale of any objective values in terms of information gain since a vertex may only be profitably collected once in these studies. All information gain values I_j are generated randomly on $[0, 1]$ using a uniform distribution for all microgrids.

The scoring methods only use one of the six available threads while any CPLEX instances uses all six threads. The data emulator first generates the series of macrogrids, the scoring methods generate the inputs for the MGAM-BP model, and the MGAM-BP is executed. Each stage is executed sequentially to avoid any performance interference and allows the monitoring CPU run time for each step. These studies aim to (1) better understand how the input set sizes to the MGAM-BP affects the model and (2) characterize the advantages and

disadvantages of the the QSP vs. GRTSP-H.

3.1. ***Full Factorial Study***

The full factorial study on the MGAM-BP varies the number of ingress/egress points and discrete time samples across 18 randomly generated macrosequences. Each macrogrid $H_g(V_g, E_g) \in M$ contains 100 microgrids ($|V_g = 100|$) and $|M| = 75$. The number of ingress points and egress points generated for each grid $\gamma_g = \{5, 10, 15\}$ and the discrete time samples $\tau_g = \{5, 10, 15\}$ also use three levels with the same values. The number of decision variables in the MGAM-BP equals $|M| * \tau_g * \gamma_g * \gamma_g$, so before conducting the study it is likely that the number of ingress/egress points considered have the largest impact on solve time. There are no restrictions placed on solve time or solution quality for the MGAM-BP and the model is expected to solve to optimality. The GRTSP-H generates the values of $f^g_{a,p,q}$ and the sets of ingress/egress points and time allocations as outlined previously in Subsection 2.2. The total allotted mission time T_{MAX} is set to 1875 units, averaging 25 units per each macrogrid. Each macrogrid may not be allotted any more than 10% of the surplus time created from minimally routing throughout the space.

The initial results of the full factorial study is gathered in Table 1 below. The two primary responses of the study is the solution value and the solve time of the optimal MGAM-BP model. In the table, the runs are grouped for each combination time allocations and ingress/egress points considered for the study. The means are averaged across 18 runs for each row and the half widths are generated with $\alpha = 0.05$ using the Student's T distribution and standard deviations for samples.

As somewhat expected, having only 5 time samples and 5 ingress/egress points had did worse in terms of the objective and also solved the quickest. Having 15 time samples and 15 ingress/egress points had the best objective and, on average, took the longest to solve. The average improvement over the objective by simply increasing the number of points considered results in a 6.8% increase in information collection. All runs converged to optimality in less than 40 seconds with the larger models having more variance on the convergence times. The residual plots for both the solution values and the solve times are also included in Figure 6.

Looking at the residuals with respect to the observation order of the runs, it appears there were at least 5 blocks of replications that resulted in below-average solutions. Three

Table 1: The mean results of the full factorial study of the MGAM-BP grouped by parameters τ_g and γ_g across 18 replications with their respective half widths ($\alpha = 0.05$)

MGAM-BP Full Factorial Mean Responses					
τ_g	γ_g	**Solution Value**	$HW_{SolutionValue}$	**Solve Time (s)**	$HW_{SolveTime}$
5	**5**	199.207	2.098	1.911	0.462
5	**10**	202.085	1.952	4.922	1.022
5	**15**	203.733	1.931	3.500	0.833
10	**5**	206.480	2.153	3.402	1.253
10	**10**	209.156	2.030	6.971	2.136
10	**15**	210.815	2.186	10.699	2.584
15	**5**	208.666	2.032	4.751	1.873
15	**10**	210.963	1.868	14.583	3.293
15	**15**	212.757	1.912	18.239	4.674

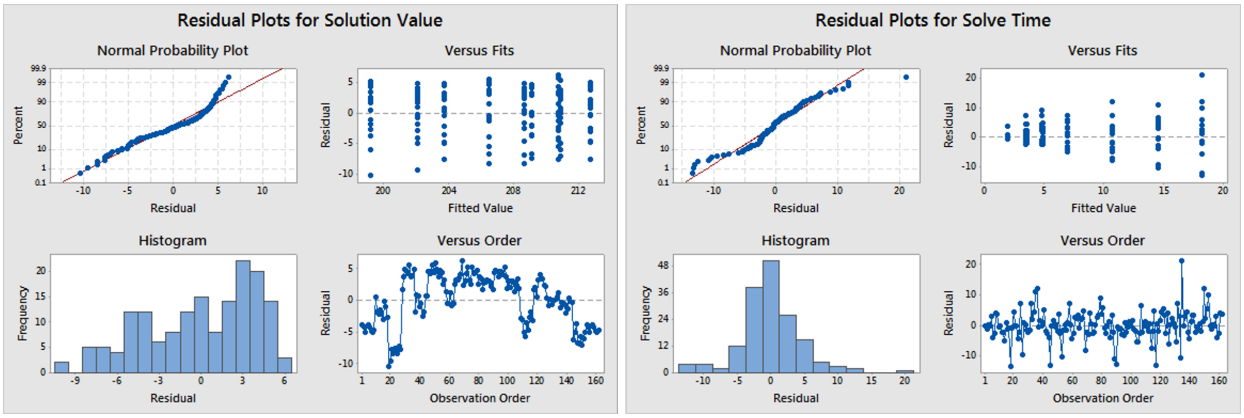


Figure 6: The residual plots for the factorial study with respect to the solution value and model solve times

possible explanations exist for this trend, either: the randomness of the macrogrid emulator created low information gain values, allocation sets were suboptimally chosen, or the scoring methods scored lower for the entire block. The most likely cause is that simply the GRTSP-H resulted in paths that when sequenced together resulted in a slightly lower objective than average ($< 5\%$) in part due the randomness of the study. The residuals for the solve time appear normal, with many MGAM-BP instances solving very quickly and a few of the larger models taking noticeably longer to solve compared to the mean which is expected. The main

effects plots shown in Figure 7 give a slightly better understanding on how the different set sizes affect the model.

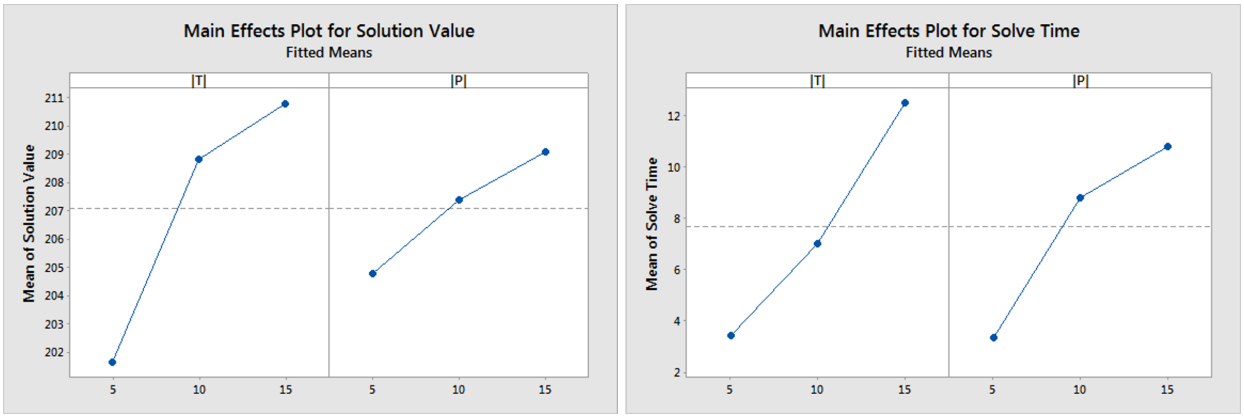


Figure 7: The main effects plots for the factorial study with three levels for both the number of time samples $|T_g|$ and ingress/egress points $|P_g|$

The main effects for the solution value suggest that increasing the number of time values sampled for allocation assignment has the greatest effect on solution quality. The lowest number of discrete time samples selected ($\tau_g = 5$) result in the worst solutions observed and the most number of time samples ($\tau_g = 15$) resulted in the best solutions. Varying the number of ingress/egress points also affected solution quality but with a less dramatic effect. The main effects plot for solve time also reveal that the number of time points considered seems to also have a greater effect on solve time versus the number of ingress/egress points. These findings are futher reinforced by the interaction plots in Figure 8.

The most substantial increases of solution quality as observed in the interaction plot is the result of increasing the number of time points sampled. The interaction plot of the mean solve time has a plotted value of $\gamma_g = 15$ lower than $\gamma_g = 10$ when $\tau_g = 5$ which may seem concerning since a larger model is expected to take longer to solve. The solve times of the MGAM-BP on these sized problem instances are relatively small with significant variance. The variability in the dynamic search procedure used by CPLEX has no guarantees on solve time, especially when dealing with models that solve within seconds. Statistically, the means fall within each other's confidence intervals suggesting their difference is insignificant and adding more replications may result in a better measure of the means. This does not affect the overall interpretation of the interaction between parameters.

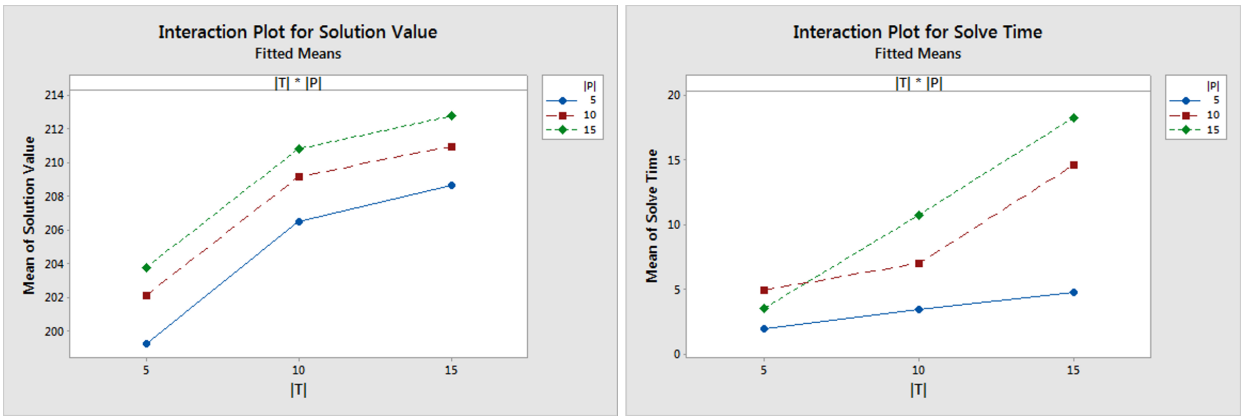


Figure 8: The interaction plots for the factorial study with three levels for both the number of time samples $|T_g|$ and ingress/egress points $|P_g|$

The full factorial study on the MGAM-BP emphasizes the importance of determining proper set sizes for the candidate time allocations and ingress/egress points for the macro-route. A particular emphasis is on the candidate time allocation set sizes and values since it appears to have the greatest impact on solution quality and solve time. This does not discount the importance of generating a proper set of ingress/egress points for the macrogrids, but in the ingress/egress points seem to be more predictable in their effects to the responses in this study.

3.2. ***Scoring Method Comparison***

The QSM and GRTSP-H outlined in Subsection 2.2 are critical components to the MGAM-BP, generating sets of time allocations and ingress/egress points with a corresponding score value $f^g_{a,p,q}$. The two most important points of comparison are the solution quality and score solve times for the methods. The score values of the QSM are unitless while the scores of GRTSP-H represent the information gain from an actual route, making the two difficult to compare. A prize-collecting vertex routing (PCVR) model is used to take the output from the MGAM-BP solutions utilizing each scoring method, generating a new series routes, and standardizing the output.

The PCVR model (Moskal II and Batta, 2015) is an orienteering-framed mixed binary program which generates routes with the objective to maximize prize collections which is synonymous with information gain in the definition of this paper. Details of the PCVR model

is omitted for the sake of brevity, but in this implementation it is a standard orienteering math model adapted to the context of UAV routing. The constraints allow for a source and sink vertex to start and end the route which is the same as routing between an ingress and egress point on the macrogrid. The PCVR model provides more consistent solutions than the GRPTSP-H, but it is far more computationally burdensome so it may only be used to route on macrogrids containing small to medium quantities of microgrids.

The scoring method comparison considers 55 unique macrosequences where $|M| = 25$ and $|V_g| = 49$. The total mission cost $T_{MAX} = 750$, averaging 30 units of travel cost for each macrogrid. Any unused slack during for a route relative to its maximum time allocation is not used in any other macrogrids. This is unconventional in application but to truly gauge the scoring models' performances, routes are purely constrained by the time designated by the MGAM-BP. The MGAM-BP is again unrestricted in solve time, however, the PCVR is restricted to a 30 minute solve time for each macrogrid resulting in 25 PCVR subproblems used for each MGAM-BP solution. The computational burden of models like the PCVR is in part motivation for the MGAM-BP model and scoring methods, reducing the dependence on these routing models and partitioning the routing environment into manageable subproblems. A total of 10 ingress/egress points and 10 different time samples are considered for each macrogrid. The main responses in this study are the standardized objective after using each scoring method, running the MGAM-BP model to determine the best ingress/egress points using the scores, and then generating a route using the same routing model. Table 2 summarizes the results of the comparison study consisting of over 1100 hours of CPU time with each individual run shown in Table A.3 in the Appendix.

The standardized objective of the MGAM-BP using the GRTSP-H results in an average of 6.45 ± 0.75 percent improvement over the model using the QSM for score generation. Out of the 55 runs, there was only one instance of the QSM outperforming the GRTSP-H resulting in a 1.86% difference in the standardized objective. A probability plot is generated in Figure 9 which identifies the under-performing GRTSP-H solution as the worst GRTSP-H solution across all runs. Using the Grubbs' Test for outliers, the null hypothesis states that all standardized GRTSP-H solutions come from the same normal population and the alternative hypothesis states that the smallest or largest data value is an outlier. At a 5%

Table 2: The mean results and respective half widths ($\alpha = 0.05$) for the responses comparing the GRTSP-H and QSM scores over 55 runs

GRTSP-H vs. QSM Aggregated Statistics						
Stat	**Standardized Objective**		**Score Time (s)**		**MGAM-BP Solve Time (s)**	
	GRTSP-H	**QSM**	**GRTSP-H**	**QSM**	**GRTSP-H**	**QSM**
Mean	87.26563	82.01107	37.62469	0.102236	5.0888	34.29024
HW	0.450613	0.53842	0.181551	0.000725	0.530209	5.682245

level of significance, we are unable to reject the null hypothesis and we conclude that we cannot statistically consider this point as an outlier.

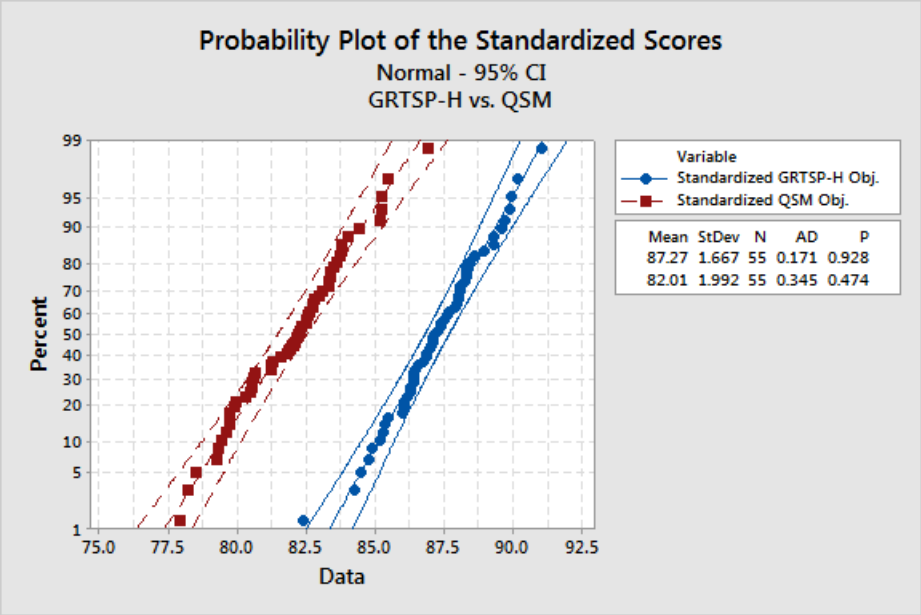


Figure 9: A probability plot of the standardized objectives of the GRTSP-H and QSM approaches comparing the solution quality across all runs

One advantage of the QSM is the incredibly quick solve time, taking consistently just over a tenth of a second to score all combinations. The GRTSP-H took averaged 37.6 seconds for the entire macrosequence which equates to roughly 0.0015 seconds for each of the required 25,000 scores needed as MGAM-BP input in this portion of the study. Although the GRTSP-H consistently outperforms in scoring to the QSM, some care must be taken to in defining

the number of ingress/egress and time allocations considered to avoid serious degradation of the efficiency in this scoring method. Interestingly, the MGAM-BP solved noticeably quicker with inputs from the GRTSP-H as opposed to the QSM. The difference in score values of $f^g_{a,p,q}$ generated by the QSM are less dramatically different numerically than those of the GRTSP-H. As a result, CPLEX cannot cut the solution tree as efficiently resulting in longer solve times.

We believe that the results of this study conclusively show that the GRTSP-H is superior in generating input for the MGAM-BP for allocation assignment. The longer score generation time is offset by its ability to create inputs that reduce MGAM-BP solve times compared to its QSM counterpart. The QSM should not be discarded as it has the ability to generate solutions predictably lower than the GRTSP-H but in a fraction of the time. The QSM serves as an adequate lightweight backup if the problem parameters define values of $|M|$, τ_g, and γ_g that are computationally burdensome for the GRTSP-H to handle.

4. CONCLUSIONS

This paper presents a method to efficiently deploy vehicle routing methods across a partitioned area of operation. In a large area of operation, a prescribed macroroute is broken down into a series of subproblems. The MGAM-BP model is used to link the subproblems by identifying the best sequence of ingress and egress points between subgraphs and also determine the time allocation for the subgraph to maximize global information collection. The MGAM-BP considers a series of discrete points scored as a function of an all ingress/egress point pairs and time samples for all macrogrids. This formulation is derived from the MGAM-NLP which considers discrete ingress/egress points but allocates over continuous time.

The QSM and GRTSP-H methods characterize the AO and provide an alternative method to generating input into the MGAM-BP rather than using traditional route generation algorithms. The QSM is specifically designed to be computationally efficient and favors ingress/egress point pairs that are close to areas of high information gain. The GRTSP-H generates its scores by generating a route between in the ingress and egress points that is feasible for given a time allocation. Instead of favoring only the ingress and egress points, nearby points with high values of information gain are sequentially added to the waypoint

set until all slack on the route cost is filled. These methods do not guarantee optimal selection scores; instead they provide quantitative insight into a the partition sequencing and allocations on solvable problem instances.

This research is strongly motivated by the need for efficient routing of UAVs in a near real-time scenarios. The full factorial study of the MGAM-BP reveals reasonable solve times for problem instances that could be considered large in field applications. A particular emphasis is placed on both the number of time samples considered along with their quality when trying to maximize information gain across the macroroute and reduce the solve time of the model. The GRTSP-H dominates the QSP in terms of solution quality after standardizing the objectives of the scoring methods through post-processing of their respective MGAM-BP solutions. Given a macroroute, the MGAM-BP combined with the GRTSP-H provides an all-inclusive partitioning and sequencing methodology to provide a continuous route across all macrogrids. This route serves as an initialization solution to further optimization algorithms or functions as a stand-alone routing procedure capable of delivering routes with small computational overhead.

5. ACKNOWLEDGMENT

We wish to acknowledge the University at Buffalo's Center for Computational Research for allowing us access to their high powered computing resources while benchmarking our work.

6. AUTHOR STATEMENT

This research has been partially supported by the Office of Naval Research through CUBRC, Inc. The views in this paper are those of the authors and do not reflect the official policy or position of the Office of Naval Research, CUBRC, Inc., Department of Defense, of the U.S. Government.

7. References

Cullen, F. H., Jarvis, J. J., Ratliff, H. D., 1981. Set partitioning based heuristics for interactive routing. Networks 11 (2), 125–143.
URL http://dx.doi.org/10.1002/net.3230110206

Dana, P. H., 1999. Coordinate systems overview. University of Colorado Boulder 15.

Floudas, C., Pardalos, P. (Eds.), 2009. Encyclopedia of Optimization, 2nd Edition. Springer.

Gu, D., Pei, G., Ly, H., Gerla, M., Hong, X., 2000a. Hierarchical routing for multi-layer ad-hoc wireless networks with UAVs. In: MILCOM 2000. 21st Century Military Communications Conference Proceedings. Vol. 1. pp. 310–314 vol.1.

Gu, D., Pei, G., Ly, H., Gerla, M., Zhang, B., Hong, X., 2000b. UAV aided intelligent routing for ad-hoc wireless network in single-area theater. In: Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE. Vol. 3. pp. 1220–1225 vol.3.

Hager, J. W., Fry, L. L., Jacks, S. S., Hill, D. R., 1992. Datums, ellipsoids, grids, and grid reference systems. Tech. rep., DTIC Document.

Hendrickson, B., Leland, R., 1995. A multi-level algorithm for partitioning graphs. SC Conference 0, 28.

Iwata, A., Chiang, C.-C., Pei, G., Gerla, M., Chen, T.-W., Aug 1999. Scalable routing strategies for ad hoc wireless networks. Selected Areas in Communications, IEEE Journal on 17 (8), 1369–1379.

Kearns, M., Mansour, Y., Ng, A. Y., 2013. An information-theoretic analysis of hard and soft assignment methods for clustering. CoRR abs/1302.1552.
URL http://arxiv.org/abs/1302.1552

Lampinen, R., 2001. Universal transverse mercator (UTM) and military grid reference system (mgrs).

Lance, G. N., Williams, W. T., 1967. A general theory of classificatory sorting strategies: 1. hierarchical systems. The Computer Journal 9 (4), 373–380.
URL http://comjnl.oxfordjournals.org/content/9/4/373.abstract

Langley, R. B., 1998. The UTM grid system. GPS world 9 (2), 46–50.

Moskal II, M., Batta, R., 2015. Adaptive UAV surveillance using a prize-collecting vertex routing model. Computers & Operations Research, *Working Paper*.

Nagy, G., Salhi, S., 2007. Location-routing: Issues, models and methods. European Journal of Operational Research 177 (2), 649 – 672.
URL http://www.sciencedirect.com/science/article/pii/S0377221706002670

Rosenkrantz, D. J., Stearns, R. E., Lewis, II, P. M., 1977. An analysis of several heuristics for the traveling salesman problem. SIAM journal on computing 6 (3), 563–581.

Savelsbergh, M. W. P., Sol, M., 1995. The general pickup and delivery problem. Transportation Science 29 (1), 17–29.
URL http://dx.doi.org/10.1287/trsc.29.1.17

Stott, P. H., 1977. The UTM grid reference system. IA. The Journal of the Society for Industrial Archeology, 1–14.

Toth, P., Vigo, D. (Eds.), 2001. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Appendix A. Computational Results

Table A.3: The results of 55 runs comparing the GRTSP-H vs. the QSM using the PCVR model for standard route generation outlined in Section 3

GRTSP-H vs. QSM Run Statistics						
Run	**Standardized Objective**		**Score Time (s)**		**MGAM-BP Solve Time (s)**	
	GRTSP-H	**QSM**	**GRTSP-H**	**QSM**	**GRTSP-H**	**QSM**
1	87.919	84.445	37.388	0.101	5.337	35.692
2	87.196	80.603	37.236	0.103	4.792	53.064
3	89.314	85.491	36.802	0.1	4.074	53.238
4	84.278	82.591	37.69	0.102	4.323	54.646
5	88.361	83.538	37.466	0.102	7.125	56.942
6	88.119	80.485	37.679	0.102	5.658	10.397
7	86.317	79.626	37.349	0.102	3.828	34.693
8	88.352	85.245	37.701	0.1	2.229	4.899
9	86.455	82.183	38.355	0.104	3.818	49.057
10	90.211	82.385	38.043	0.103	5.601	51.852
11	87.626	80.553	37.48	0.118	6.055	37.393
12	87.110	79.746	37.48	0.101	3.647	56.854
13	89.948	83.639	36.219	0.108	10.767	8.114
14	85.397	81.819	38.364	0.102	3.02	6.568
15	86.406	79.310	37.767	0.102	9.004	10.991
16	88.484	82.807	38.109	0.102	4.388	9.093
17	89.757	82.217	37.271	0.102	6.532	41.65
18	88.040	85.292	37.263	0.101	2.97	9.815
19	86.896	82.149	37.001	0.101	2.01	30.928
20	84.932	82.530	37.699	0.102	7.28	41.61
21	86.872	83.121	37.07	0.102	2.784	6.315
22	87.396	80.328	38.466	0.102	6.621	32.748
23	87.714	81.284	37.047	0.102	4.619	48.033
24	85.514	79.281	38.332	0.104	4.113	60.526
25	85.198	80.572	38.137	0.102	4.685	9.47

Run	Standardized Objective		Score Time (s)		MGAM-BP Solve Time (s)	
	GRTSP-H	QSM	GRTSP-H	QSM	GRTSP-H	QSM
26	87.550	77.919	37.701	0.102	1.789	28.342
27	88.967	83.404	37.6	0.102	5.121	5.675
28	85.304	86.923	36.874	0.104	4.723	63.825
29	82.407	79.731	38.23	0.104	6.52	34.575
30	84.520	81.980	37.767	0.104	6.808	36.693
31	88.343	84.031	38.66	0.101	3.703	57.682
32	89.643	83.793	38.365	0.103	3.88	4.091
33	87.323	83.340	37.896	0.1	4.441	4.166
34	86.335	82.096	38.785	0.103	5.922	57.797
35	86.772	79.718	38.071	0.103	1.753	48.424
36	86.440	79.899	37.595	0.102	5.532	4.758
37	87.388	85.199	37.729	0.102	10.618	4.288
38	86.025	79.991	38.28	0.098	5.337	50.586
39	84.813	83.021	38.562	0.104	6.244	58.635
40	86.401	82.625	37.498	0.103	5.43	6.225
41	86.213	83.394	38.008	0.103	6.106	51.164
42	88.056	83.320	37.493	0.099	7.105	53.645
43	89.357	81.246	37.53	0.098	5.485	32.367
44	87.070	82.746	37.205	0.1	1.445	70.978
45	88.288	83.832	35.655	0.103	4.318	9.52
46	91.078	83.761	37.199	0.101	6.653	10.899
47	86.605	79.448	37.844	0.102	5.874	37.446
48	87.032	78.247	35.264	0.101	5.144	42.378
49	87.149	82.758	37.607	0.102	7.59	50.222
50	88.643	81.267	37.761	0.101	6.839	35.647
51	86.073	78.526	38.391	0.101	3.931	58.698
52	88.090	80.701	37.914	0.102	5.108	57.897
53	86.052	82.314	36.839	0.102	2.518	5.773
54	89.886	81.599	38.063	0.101	4.112	53.541
55	87.976	82.539	37.558	0.102	4.555	35.438