Robotics: Science and Systems 2009
Seattle, WA, USA, June 28 - July 1, 2009

Positioning Unmanned Aerial Vehicles as Communication Relays for Surveillance Tasks

Oleg Burdakov[a], Patrick Doherty[b], Kaj Holmberg[a], Jonas Kvarnström[b], Per-Magnus Olsson[b,*]
[a] Dept. of Mathematics. E-mail: {olbur, kahol}@mai.liu.se
[b] Dept. of Computer and Information Science. E-mail: {pdy, jonkv, perol}@ida.liu.se
Linköping University, SE-581 83 Linköping, Sweden
[*] Corresponding author

***Abstract*— When unmanned aerial vehicles (UAVs) are used to survey distant targets, it is important to transmit sensor information back to a base station. As this communication often requires high uninterrupted bandwidth, the surveying UAV often needs a free line-of-sight to the base station, which can be problematic in urban or mountainous areas. Communication ranges may also be limited, especially for smaller UAVs. Though both problems can be solved through the use of relay chains consisting of one or more intermediate relay UAVs, this leads to a new problem: Where should relays be placed for optimum performance? We present two new algorithms capable of generating such relay chains, one being a dual ascent algorithm and the other a modification of the Bellman-Ford algorithm. As the priorities between the number of hops in the relay chain and the cost of the chain may vary, we calculate chains of different lengths and costs and let the ground operator choose between them. Several different formulations for edge costs are presented. In our test cases, both algorithms are substantially faster than an optimized version of the original Bellman-Ford algorithm, which is used for comparison.**

I. Introduction

A wide variety of applications for unmanned aerial vehicles (UAVs) include the need for surveillance of distant targets, including search and rescue operations, traffic surveillance and forest fire monitoring as well as military applications. In most cases the information gathered must be transmitted continuously from a *survey UAV* to a base station where the current operation is being coordinated. As this information may include urgent high-volume sensor data such as live video, one often requires considerable uninterrupted communications bandwidth. To minimize quality degradation, UAV applications therefore tend to require line-of-sight (LOS) communications, which is problematic in urban or mountainous areas. The maximum communication range is typically also limited, especially when smaller and lower-cost UAVs are used as relays.

Though the problem of achieving line-of-sight can be ameliorated by increasing altitude, this would require greater communication ranges and is not always permitted by aviation regulations. UAVs may also be unable to fly at sufficient altitude to avoid detection in the case of military or police surveillance, preferring instead to hide behind obstacles. Instead, both intervening obstacles and limited range can be handled using a chain of one or more intermediate *relay UAVs* [15] passing on information to the base station (Figure 1).

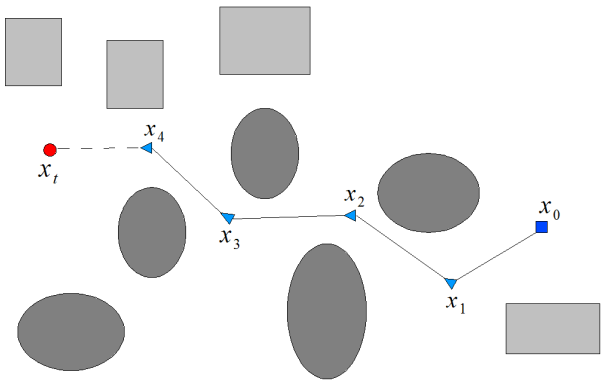

Fig. 1. UAVs at x_1, x_2 and x_3 are acting as relays, connecting the base station at x_0 with the survey UAV at x_4, which is surveying the target at x_t. This is a relay chain of length 4.

Rather than limiting ourselves to the use of a single relay UAV, we are interested in the general relay positioning problem for an arbitrary number of relays. A solution to this problem should be sufficiently scalable to enable the use of a large number of comparatively inexpensive miniature UAVs with highly limited communication range. While the number of relays used is an obvious quality measure of a relay chain, one can also benefit greatly from taking other scenario-dependent measures into consideration, even at the cost of using additional UAVs. For example, position flexibility and safety margins may be important in case of wind drift or moving targets, and in some cases one may want to minimize the risk of detection by others. As the desired trade-off may vary from case to case and can be difficult to formalize, we prefer to generate multiple relay chains using different numbers of UAVs and leave the final choice of solution to a ground operator.

In this paper we formally define the relay positioning problem in a continuous setting. As this problem is intractable, we continue by generating a corresponding discrete graph, where nodes correspond to physical locations where relays may be placed, edges connect nodes between which communication is

possible, and edge costs are used to model quality measures related to UAV placement and communication paths. The desired set of relay chains can then be generated by solving the *all hops optimal path* (AHOP) graph search problem [7]. We present two distinct algorithms for solving the AHOP problem considerably more efficiently than previous algorithms, the first being a dual ascent algorithm and the second being a modification of the Bellman-Ford [6] algorithm. Though examples in this article focus on the use of UAVs, both algorithms can equally well be used for relay placement for unmanned ground vehicles (UGVs).

II. Previous Work

Control behavior for teams of unmanned ground vehicles involving line-of-sight was investigated in Sweeney et al. [17]. In an indoor setting, a lead UGV advances from the base station towards the goal position and incrementally determines where to place relay UGVs along the way in order to maintain communication with the base station. Various strategies are evaluated in terms of time and energy usage. A small survey of positioning algorithms for UGVs is available in Nguyen et al. [13]. The algorithms presented in these articles have several commonalities. No quality or cost measure is used and it is not certain that the goal position will be reached, as no a priori calculation or evaluation of paths is performed.

Arkin and Diaz [2] used a behavior-based architecture to allow teams of ground robots with line-of-sight communication to explore buildings and to find stationary objects, using only limited knowledge about the area in which those objects may be placed. This problem differs significantly from ours, where the goal position is known in advance.

An algorithm for maintaining LOS between groups of planetary rovers exploring an area is presented by Anderson et al. [1]. The algorithm is based on several heuristics and although testing indicates that the algorithm performs well, it does not guarantee a solution even if one exists. Similarly, no quality measure exists.

The concept of using a UAV as a communication relay, including intended platforms and communications equipment, is discussed in Pinkney et al. [15], but no algorithms are presented. In limited cases, where a single relay UAV is sufficient, the survey UAV could plan and fly a trajectory to the goal while the relay UAV continuously attempts to maintain line-of-sight to both the survey UAV and the base station [16]. The benefit of using a single relay UAV in an urban environment has also been simulated [5]. Here the UAV works as a relay between two entities on the ground. The focus of the work is to determine the percentage of an urban area with acceptable coverage for UAVs positioned at different heights. Only a single relay is used and no algorithm for determining the positioning of the UAV is provided.

Problems that are superficially very similar to the multiple relay positioning problem are encountered in ad-hoc networks, where messages are to be delivered in a network where there is no control of the network topology. Routing algorithms for such networks must be able to handle addition and removal of nodes at runtime [10, 12]. Using a swarm of UAVs to improve the range and reliability of an ad-hoc network has also been investigated [14]. Good results are achieved, mainly through significantly increasing the range using the same transmission power, compared to using a direct ground link. However, significant differences exist between the problem investigated here and ad-hoc networks in that we have full control over node (UAV) positioning and in that we are only interested in transferring information between the survey UAV and the base station.

III. Problem Formulation

We formally define the *relay positioning problem* as follows.

Let $L \geq 1$ UAVs with identical communication capabilities be available, one of which is the survey UAV.

Assume as given the free space $X \subseteq R^3$ where relay and survey UAVs may safely be placed, together with the position $x_0 \in R^3 \setminus X$ of a base station and the position $x_t \in R^3 \setminus X$ of a survey target.

Assume also as given a boolean *communication reachability function* $g(x, x')$ that holds iff communication is possible between points $x, x' \in R^3$ and a *communication cost function* $f(x, x')$ denoting the non-negative cost of establishing such a communication link. Finally, assume as given a *survey reachability function* $s(x, x')$ that holds iff a survey UAV at $x \in X$ is able to survey a target at $x' \in R^3 \setminus X$, and a *survey cost function* $t(x, x')$ denoting the non-negative cost of such a survey.

A *relay chain* of length l between x_0 and x_t is a tuple of free UAV positions $\langle x_1, \ldots, x_l \rangle$, where $\{x_1, \ldots, x_l\} \subseteq X$, such that $\forall i,\ 0 \leq i < l \rightarrow g(x_i, x_{i+1})$ and $s(x_l, x_t)$. The *cost* of a relay chain $\langle x_1, \ldots, x_l \rangle$, denoted by $cost(\langle x_1, \ldots, x_l \rangle)$, is defined as $(\sum_{i=0}^{l-1} f(x_i, x_{i+1})) + t(x_l, x_t)$.

A relay chain c of length l between x_0 and x_t is *relevant* iff there exists no relay chain c' of length $l' < l$ between x_0 and x_t such that $cost(c') \leq cost(c)$. This reflects the fact that one would not use c if there is an alternative chain that uses fewer relays and is not more expensive.

The relay positioning problem consists of finding, for each $1 \leq l \leq L$, a relevant minimum-cost relay chain of length l between x_0 and x_t, or determining that no such chain exists. Note that any problem instance may admit multiple (optimal) solutions, since there may be multiple minimum-cost relay chains of any given length.

A. Reachability Functions

The communication reachability function g determines whether communication should be considered feasible between two points, regardless of the quality of the resulting link. For our purposes, g is usually defined by a limited radius r and a requirement of free line-of-sight between positions. More formally, let $O \subseteq R^3$ be the set of coordinates that are free from obstacles. Note that this may differ from the positions in X, where it should also be possible to place UAVs safely and minimum distances to obstacles may have to be taken into account. Then, $g(x_i, x_j)$ holds iff $\|x_j - x_i\| \leq r$ and

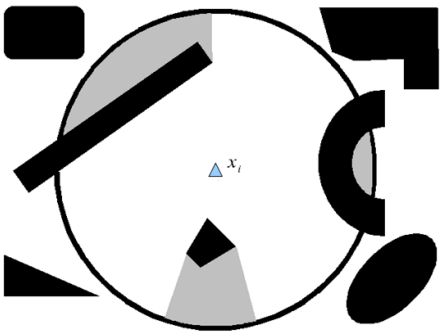

Fig. 2. Example of a node with UAV communication radius r. The obstructed volume is the sum of the black obstacles and the shaded area representing the non-visible volume inside the sphere.

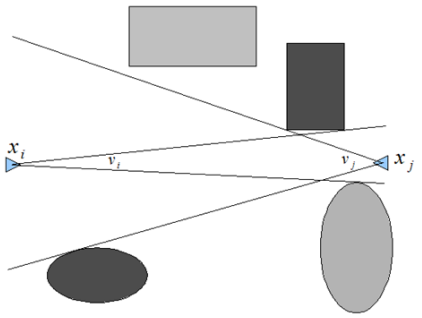

Fig. 3. Minimum free angle used as edge cost.

$[x_i, x_j] = \{x \in R^3 : \alpha x_i + (1-\alpha)x_j, \alpha \in [0,1]\} \subseteq O$. However, the algorithms to be defined below work equally well with other definitions of g, with or without free LOS.

The survey reachability function s is usually defined in a similar manner, though often with a different maximum distance between the survey UAV and its target.

Note that reachability functions may be asymmetric. For example, a UAV may only be able to survey targets below it.

B. Cost Functions

In the definition of communication and survey cost functions, the term "cost" is used in a very general sense, where both functions can be used to model arbitrary position-related quality measures determining the overall quality of a relay chain. For example, a high survey cost $t(x, x')$ may be used to indicate that surveying a target at x' from a UAV positioned at x would yield information of comparatively low quality. Similarly, given a suitable model of the environment and the communications hardware being used, distance- or position-dependent transmission power requirements as well as many forms of communication performance degradation can be modeled as communication costs and taken into account by positioning algorithms. Finally, communication costs can be used to model arbitrary penalties for specific relay positions.

We will now provide two specific examples of communication cost functions $f(x, x')$ related to the existence of obstacles that may hinder line-of-sight communications. Note that one can easily use a weighted sum of such measures together with costs related to continuous communication degradation such as path loss. Survey cost functions can be defined analogously.

Obstructed volume. Given a maximum communication range r, a UAV positioned at $x \in X$ can communicate with other UAVs in a sphere of volume $4\pi r^3/3$. Given line of sight requirements, parts of this volume may be obstructed by obstacles such as buildings or hills. Though we normally operate in three dimensions, a 2D example is shown in Figure 2 for simplicity. The greater the obstructed volume is, the less flexible the given position is in terms of future UAV movements. Thus, one can directly define $f(x, x')$ as the obstructed volume within a sphere of radius r centered at x'.

Minimum free angle. While the obstructed volume measure is useful if one anticipates attempting to maintain communication during future movements to entirely new UAV positions, other measures are more relevant for maintaining flexibility around current relay positions. This can be important for several reasons, such as the need to minimize susceptibility to disturbances and be robust against temporary wind drift. One such measure of flexibility is the minimum free angle between two successive positions x_i and x_j in a relay chain. Figure 3 shows a two-dimensional example, where the UAV at x_i must not move outside the angle originating in x_j and vice versa. The cost associated with this edge might then be inversely correlated to the minimum of these angles. One might also use the inverse of the square root or logarithm of the angle, which increases the relative penalty of including small-angle edges in a path.

For both of these cost functions, terrains with higher and more variable obstacle densities will be more likely to give rise to a spectrum of alternative relevant relay chains. In this situation, the shortest chains (in terms of the number of edges) are close to obstacles and consequently quite expensive, while the cheapest chains take long detours in order to maintain a greater distance from the obstacles. Conversely, terrains with lower and less diverse obstacle densities will be more likely to give rise to only a few relevant relay chains, as longer chains are also likely to be more expensive.

IV. Solving the Relay Positioning Problem

The relay positioning problem is a multi-extremal problem where the feasible set is typically disjoint, and in many cases the number of disjoint feasible subsets is large, which makes the continuous problem intractable. We therefore suggest to discretize the environment, in order to decrease the computational burden. The solution is thus divided into two distinct steps: First, generate a discrete visibility graph, and second, use this graph to solve the all hops optimal path problem.

A. Discretization and Visibility Graph Generation

The first step of discretizing an instance of the continuous relay positioning problem consists of selecting a finite set of positions $X' \subseteq X$ among the free coordinates. These are the

positions that will be considered for relay and survey UAV placement in the discretized version of the problem. Once this selection has been made, a discrete visibility graph can be created as follows.

Associate each position $x \in X' \cup \{x_0, x_t\}$ with a unique node, where n_0 denotes the base station node associated with x_0 and n_t denotes the survey target node associated with x_t, and let N be the set of all nodes.

For each $x \in X'$ corresponding to $n \in N$ and satisfying $g(x_0, x)$, create an edge $e = (n_0, n)$ of cost $f(x_0, x)$ representing the possibility of communication between the base station and position x. For each $x, x' \in X'$ corresponding to $n, n' \in N$ and satisfying $g(x, x')$, create an edge $e = (n, n')$ of cost $f(x, x')$ representing the possibility of communication between positions x and x'. Finally, for each $x \in X'$ corresponding to $n \in N$ and satisfying $s(x, x_t)$, create an edge $e = (n, n_t)$ of cost $t(x, x_t)$ representing the fact that a survey UAV at x would be able to survey the target at x_t. Let E be the set of all edges.

Then, $G(N, E)$ is a directed graph corresponding to the original continuous problem instance. Note in particular that the survey target node has no outgoing edges and that all its incoming edges satisfy the survey reachability function, ensuring that its predecessor in any path from the base station to the survey target must be suitable for a survey UAV. Note also that most parts of this graph only depend on the environment and not on the position of the base station or survey target, and can be precalculated.

A number of different alternatives are possible when determining which free positions to include in the graph. One approach consists of placing a regular three-dimensional grid over the terrain and constructing a graph from the unobstructed grid cells, where the size of the grid cells would depend on the maximum obstacle density. Regular grids are quite suitable for the type of urban and mountainous terrain we are interested in, where we are unlikely to find pathological obstacle distributions that only permit communication if relays are positioned exactly in the right place. Alternative approaches may distribute and place nodes depending on local obstacle density, for example by using structures with variable cell size or by using techniques such as bridge sampling [9] to increase the density of nodes in narrow passages.

B. The All Hops Optimal Path Problem

Given a discrete graph representation, a number of associated problems can be solved using well-known algorithms. For example, Dijkstra's algorithm or A* can be used to find the cheapest relay chain (path), the relay chain with the smallest number of hops, or the cheapest relay chain from those having the smallest number of hops[1] [6, 8].

However, these algorithms only generate a single relay chain, whereas we are interested in generating a spectrum of relevant relay chains ranging from the shortest in terms of hops to the cheapest, allowing the ground operator to choose whichever chain best fits the current resource constraints and the requirements on the current mission. This corresponds directly to the *all hops optimal path* (AHOP) problem, which has previously been applied to network routing [7].

Given the use of additive path costs, the best known algorithm for this problem is currently the AHOP version of the standard Bellman-Ford algorithm, which at its kth iteration finds all cheapest paths using k hops from a given source [7]. However, although this algorithm is capable of solving the problem, it uses a considerable amount of time and is not practically useful for larger problem instances involving large maps, fine-grained grids, or long communication ranges. This is especially true in mixed-initiative settings where a ground operator initiating a survey mission should be given a prompt response.

We have therefore developed two distinct algorithms for solving the all hops optimal path problem more efficiently than the standard Bellman-Ford algorithm. Both algorithms are complete and generate optimal solutions under the discretization provided in Section IV-A.

The following terminology and variables are common to both algorithms: n_0 is the start node corresponding to the base station, n_t is the target node, and given a node n, n_- is its set of predecessor nodes and n_+ is its set of successor nodes. The cost associated with an edge between nodes n and n' is denoted by $c_{n,n'}$.

We use the term *cheapest path tree* for a tree rooted in n_0, storing the least-cost path to each node in N given compound path costs of the form $\langle c, l \rangle$, where c is the cost of the path, l is its length, and $\langle c_1, l_1 \rangle < \langle c_2, l_2 \rangle$ iff ($c_1 < c_2$) or ($c_1 = c_2$ and $l_1 < l_2$). This is in a sense the inverse of the previously discussed compound path cost, in that it prefers the shortest path among those that have the lowest cost. The cheapest path tree can be calculated by any shortest path tree algorithm, for example, Dijkstra's algorithm.

C. Dual Ascent Algorithm

Our first algorithm (Figure 4) is a dual ascent algorithm that repeatedly calculates the cheapest path p from n_0 to n_t using modified edge costs $c'_{n,n'} = c_{n,n'} + \alpha$, where α is increased in every iteration. This is also assumed to yield, for each node n, its depth q_n (the number of hops along the path from the root to n) and its optimal path cost y_n given the current value of α.

Let l be initialized to the maximum number of acceptable hops, that is, one more than the number of available UAVs, where the last hop is between the survey UAV and the target. Give α the initial value α_0. In most cases we use $\alpha_0 = 0$, causing the algorithm to begin by generating the cheapest path possible using original edge costs. Higher values of α_0 can be used to discourage longer paths, with the effect that the algorithm no longer generates the longest relevant relay chains.

If the length of the generated path is at most l, it is feasible and uses fewer relays than any path found so far. In this case, the path corresponds directly to a relevant relay chain and can

[1] The latter can be calculated using compound path costs of the form $\langle l, c \rangle$, where l is the length of the path, c is its cost, and $\langle l_1, c_1 \rangle < \langle l_2, c_2 \rangle$ iff ($l_1 < l_2$) or ($l_1 = l_2$ and $c_1 < c_2$).

1 $l \leftarrow L + 1$
2 $\alpha \leftarrow \alpha_0$
3 Calculate cheapest path tree from n_0
using costs $c'_{n,n'} = c_{n,n'} + \alpha$.
From the tree, obtain p and y_n, q_n for all n
4 **if** length$(p) \leq l$ **then**
5 Yield p *// Feasible and shorter than previous solutions*
6 $l \leftarrow$ length$(p) - 1$
7 $F \leftarrow \{(n, n') \in E : q_{n'} \geq q_n + 2\}$
8 **if** $F = \emptyset$ **then**
9 **return** *// Path can not be shortened*
10 Calculate $\epsilon_{n,n'} \leftarrow \frac{c'_{n,n'} + y_n - y_{n'}}{q_{n'} - q_n - 1}$ $\forall (n, n') \in F$
11 $\epsilon \leftarrow \min \epsilon_{n,n'}$
12 $\alpha \leftarrow \alpha + \epsilon$
13 **Goto** 3

Fig. 4. Dual Ascent Algorithm

be yielded as partial output of the algorithm. Also, l can be updated to reflect that we are now only interested in paths strictly shorter than the one we just found.

It is clear that increasing α by ϵ will increase the cost of a path containing k edges by $k\epsilon$, thus penalizing paths in proportion to hop counts. The algorithm therefore calculates such an ϵ in a way that guarantees that some paths are shortened but no relevant paths between n_0 and n_t are missed. This calculation considers all edges (n, n') in the original graph where reaching n' from n_0 in the current cheapest path tree requires at least two hops more than reaching n. The current path to n' could then be shortened by replacing it with the current path to n and the edge between n and n'. Since this has not already been done, we know that we currently have $y_n + c'_{n,n'} > y'_n$, that is, going through n is currently more expensive. Making the path through n *equally* expensive[2] entails increasing the cost of the longer path by $y_n + c'_{n,n'} - y_{n'}$. (To be more exact, the cost of the longer path must be increased that much *more* than the cost of the shorter path.) This additional cost has to be split by $q'_n - (q_n + 1) = q'_n - q_n - 1$ nodes (the length of the old path to n', minus the length of the potential new path through n), yielding the expression for $\epsilon_{n,n'}$ found in the algorithm above. Finally, to ensure no solutions are missed, we increase α by the minimum of all such $\epsilon_{n,n'}$ and iterate. The formal proof that increasing α in this manner yields exactly the intended answer is somewhat lengthy and for details we refer the reader to Burdakov et al. [4].

When no path can be shortened, the algorithm terminates.

It should be noted that this algorithm can be used in an anytime manner, initially generating the cheapest path and then generating incrementally shorter but more expensive paths.

[2]Recall that the cost function for the cheapest path tree will prefer the shorter of two equally expensive paths. Thus, making the path through n strictly less expensive is not necessary.

k (path length)	g_k (cost)	p_k (predecessor)
4	18	n_{20}
6	12	n_{10}
7	10	n_{11}

TABLE I
EXAMPLE OF INFORMATION STORED IN EACH NODE AFTER EXECUTION OF THE MODIFIED BELLMAN-FORD ALGORITHM.

D. Modified Bellman-Ford Algorithm

Our second algorithm (Figure 5) is a modification of the original Bellman-Ford algorithm, or to be precise, a modification of its AHOP version [11]. The algorithm incrementally generates and updates a set of *reachability records* for each node. This set can also be represented as one table for each node as shown in Table I. Each reachability record $\langle k, g_k, p_k \rangle$ associated with a node indicates that the node can be reached from the base station n_0 in k hops at a cost of g_k using the predecessor p_k. A complete path from n_0 to n can always be reconstructed (in reverse order) by considering the reachability record of the predecessor p_k for $k - 1$ hops and continuing recursively until n_0 is reached.

After termination, each reachability record is guaranteed to correspond to a relevant path. Conversely, if there is a relevant path of length l between n_0 and n, then n is guaranteed to have a reachability record for $k = l$. Thus, the fact that a target node n_t is associated with a reachability record $\langle k, g_k, p_k \rangle$ corresponds exactly to the existence of a relevant relay chain between n_0 and n_t of length $k - 1$ and cost g_k. In other words, n_t can be reached from the base station using $k - 1$ intermediate UAVs, one of which is the survey UAV. For example, Figure 1 shows a path of length $k = 5$ corresponding to a relay chain of length $k - 1 = 4$, which can be extracted using the reachability record for $k = 5$ for the target node.

The algorithm requires a preprocessing step consisting of calculating the cheapest path tree with n_0 as root. We then retrieve the height k^*_{max} of this tree. Clearly, no relay chain consisting of more than k^*_{max} nodes can be relevant, since all longer paths must also be at least as expensive. This limits the number of relay positions ever required for this graph as well as the depth to which the graph needs to be searched.

N is partitioned into sets N^*_k of nodes occurring at depth k in the cheapest path tree, where $0 \leq k \leq k^*_{max}$. Clearly, any relay chain using more than k hops to reach a node $n \in N^*_k$ cannot be relevant, as it would be possible to shorten the chain (by reaching n in exactly k hops) without increasing its cost (since only k hops were required in the *cheapest* path). In addition to using this fact in the main algorithm as described below, we also use it to create an initial reachability record corresponding to the cheapest relevant relay chain for each node. In Table I, this corresponds to the row where $k = 7$.

In the main algorithm, $g(n)$ denotes the cost of the cheapest path from n_0 to node n found so far, calculated using standard

```
1  for each n ∈ N do g(n) ← +∞
2  g(n_0) ← 0
3  for each n ∈ n_0− do            // Incoming edges...
4     E ← E \ {(n, n_0)}           // ... are removed
5  V_1 ← {n_0}
6  for k = 1, ..., min(L + 1, k*_max − 1) do
7     for each n' ∈ N*_k do
8        for each n ∈ n'_− do       // Incoming edges...
9           E ← E \ {(n, n')}      // ... are removed
10    V_{k+1} ← N*_k
11    for each n ∈ V_k do
12       for each n' ∈ n_+ do
13          c ← g_{k−1}(n) + c_{n,n'}   // To n' through n in k hops
14          if c < g(n') then
15             g(n') ← c           // Lowest cost so far
16             g_k(n') ← c         // Lowest cost in k hops
17             p_k(n') ← n         // Predecessor for k hops
18             V_{k+1} ← V_{k+1} ∪ {n'}
```

Fig. 5. Modified Bellman-Ford Algorithm

additive edge costs. Initially, $g(n) = \infty$ for all nodes. The algorithm will construct and use a sequence of sets V_k, each of which is characterized by the fact that any relevant relay chain of length k must consist of a path to a node $n \in V_k$ in $k-1$ hops followed by a single outgoing edge from n.

Lines 2–5 take care of the initial node n_0. No reachability record needs to be created for this node, but $g(n_0)$ must be updated to indicate that it can be reached with cost 0 (line 2). Clearly no relevant relay chain ending in n_0 can pass *through* n_0, so all incoming edges to n_0 can be removed (lines 3–4). Finally, line 5 prepares for the first iteration by setting $V_1 = \{n_0\}$, indicating that any relevant relay chain of length 1 must consist of a path to n_0 in 0 hops (the empty path) followed by a single outgoing edge. This handles all paths of length 0.

In lines 6–18, each iteration considers paths of length $k \geq 1$, up to a maximum of $\min(L+1, k^*_{\max} - 1)$. This reflects the fact that (i) we allow at most L UAVs, which yields a total path length of up to $L+1$ when edges to the base station and target are included, (ii) no paths of length greater than k^*_{max} can be relevant, since k^*_{max} reflects the depth of the *cheapest* path tree, and (iii) any relevant path of length exactly k^*_{max} was already generated during preprocessing. Setting $L = \infty$ ensures that the algorithm finds relevant relay chains of all possible lengths.

Recall that any node $n' \in N^*_k$ occurs at depth k in the cheapest path tree. Reachability records for strictly shorter paths to n' were already created in earlier iterations, and a record for a path of length k was created during preprocessing. Records for paths of length strictly greater than k cannot be created, as this would correspond to finding a path which is longer but strictly cheaper than the one of length k, which was by definition a cheapest path. Thus, no new paths ending in any such n' can be relevant, and we can remove all incoming edges to n' without affecting correctness or optimality (lines 7–9). However, we do need to consider longer paths going *through* these nodes (line 10, explained further below).

In lines 11–18, we consider all potentially relevant relay chains of length k. Recall that any such chain must consist of a path to a node $n \in V_k$ followed by a single outgoing edge from n. We consider each such path in turn, determining its destination n' and calculating its cost c (lines 11–13). If the path is cheaper than the cheapest path found previously (with a cost of $g(n')$) we create a new reachability record with $g_k(n')$ set to the new path cost and $p_k(n')$ set to the new predecessor n (lines 14–17). Note that though reachability records are sparse, we know that any node in V_k does have a reachability record for $k-1$ due to the construction of V_k.

What remains is to prepare for the next iteration by constructing V_{k+1} according to its definition. That is, we should construct V_{k+1} in such a way that any relevant given relay chain of length $k+1$ necessarily consists of a path of length k to a node $n \in V_{k+1}$ followed by a single outgoing edge from n. It is clear that no relevant relay chain would use k hops to reach a node n if it could be reached in $k-1$ hops without incurring additional costs. Thus, it is only necessary to consider nodes whose costs were decreased by allowing paths of length k or that could not be reached at all with paths of length $k-1$. This is achieved in line 10, for nodes in N^*_k where we found a cheapest path of length k during preprocessing, and in line 18, for nodes where we found a cheaper path of length k in this iteration.

This algorithm calculates the path with the fewest hops, as well as the cheapest path regardless of the number of hops and all relevant paths in between. Like the Dual Ascent algorithm, this algorithm can be used in an anytime manner, though in the reverse direction, where the main algorithm generates chains in order of increasing length and decreasing cost.

When the algorithm terminates, a table similar to the one in Table I exists for each node. This table contains all the information necessary to trace a path back to the start node. The first row corresponds to using the minimal number of UAVs and the following rows correspond to using an increasingly larger number of UAVs, giving a path with progressively lower cost, until the least-cost path with the largest number of UAVs, which is found on the last row. For path lengths lower than the smallest k in the table, no path can be found. For example, no path of length $k = 3$ could be found for the node represented by Table I regardless of cost. Other "missing" values of k indicate that even if a path of length k could be found, it would be at least as expensive as some shorter path in the table. For example, the fact that Table I contains no record for $k = 5$ means that a path of length 5 to the given node would have a cost of at least 18, the cost of the closest record for a smaller value of k, and that using such a path would be pointless.

When the operator makes a decision about how many UAVs to allocate to the mission, he chooses whether to allocate the minimum number of UAVs required, or any other number greater than this. This requires more UAVs but yields a better solution in terms of whatever quality measure was provided

to the algorithm.

See Burdakov et al. [3] for additional details and proofs.

E. Multiple Survey Targets

The graph creation procedure discussed in Section IV-A can easily be altered to generate separate relay chains to an arbitrary number of survey targets. Instead of generating a single target node, one node t_i is created for every position x_i in a set $T \subseteq R^3$ of survey target positions. Similarly, instead of creating incoming edges to a single target node, incoming edges are created for every $x \in X', t \in T$ satisfying $s(x,t)$.

This provides support for a preprocessing phase where a single call to either of the new algorithms generates relay chains to hundreds or thousands of potential target positions. Relay chains can then quickly be extracted from the given graphs. Especially for the modified Bellman-Ford algorithm, this would take only marginally more time than generating relay chains to a single target position, given that the number of additional nodes and edges is small compared to the number of nodes and edges used for free positions where relays and survey UAVs may be located.

Though this is not a solution to the multiple target surveillance problem, since no attempt will be made to minimize the total number of relays used for a given set of targets, it can still be highly useful in the case where the position of the survey target is not known in advance.

V. Test Results

The AHOP version of the original Bellman-Ford algorithm has a time complexity of $O(|N||E|) \subseteq O(|N|^3)$. Applying the widely used optimization of terminating as soon as no node cost has been decreased during iteration k, this bound can be tightened to $O(k^*_{max}|E|) \subseteq O(|N||E|) \subseteq O(|N|^3)$.

Though the *worst* case complexity of the modified Bellman-Ford algorithm is identical, using Dijkstra's algorithm for preprocessing and partitioning ensures that one almost always considers far fewer than $|E|$ edges in each iteration. Using Dijkstra in this manner is possible due to the use of non-negative edge costs, and sensible because we want to generate multiple alternative paths, possibly to multiple targets, as opposed to a unique cheapest path to a single target, where Dijkstra alone would have sufficed.

Dual Ascent performs at most $|N|^2$ iterations, each time calling Dijkstra. Thus, its worst case complexity is in $O(|N|^2(|E| + |N|\log|N|)) \subseteq O(|N|^4)$. However, only extremely specific graph structures and cost functions require $|N|^2$ iterations, and the algorithm can still outperform Bellman-Ford in practice.

In our test cases, the obstructed volume was used for both communication cost and survey cost. We used line-of-sight requirements with varying maximum range for communication reachability and survey reachability. The environment graph was constructed using a three-dimensional grid and the grid cell size was varied between 10 and 40 meters, though it was constant within each problem instance. The test cases were taken from both rural and urban environments and the positions of base stations and targets were chosen with the intention to create challenging test cases. The lengths of relevant relay chains varied, using between 3 and 19 UAVs.

All test results were obtained from a C++ implementation on a PC running Windows Vista with an Intel Core 2 Duo 2.4 GHz CPU and 2 GB RAM. Only one core was used in testing. For comparison, we also implemented an AHOP version of the standard Bellman-Ford algorithm using the common optimization discussed above. Dijkstra's algorithm was used to calculate the cheapest path tree for both the dual ascent algorithm and the pre-processing phase of the modified Bellman-Ford algorithm. To ensure that all solutions were found, we used $L = \infty$ and $\alpha_0 = 0$.

Table II shows empirical test results for all three algorithms, averaged from 1000 executions.

The first three columns show the number of nodes and edges of each discretized graph together with the height k^*_{max} of the associated cheapest path tree rooted in the base station node.

Opt. BF shows the time requirements for our optimized implementation of the standard Bellman-Ford algorithm.

As the modified Bellman-Ford algorithm requires the calculation of the cheapest path tree as well as the determination of the N^*_k sets, the times for this algorithm are displayed in four separate columns. The column labelled ***CPT*** contains the times for calculating the cheapest path tree and the column labelled N^*_k shows the times for determining the N^*_k sets. The time used by the main algorithm are shown in the column marked *Mod. BF*. The time used by the complete algorithm including preprocessing is the sum of the three previous columns and is shown in the column marked *Total*. The times might not add up due to rounding.

The final column shows the complete time requirements for the Dual Ascent algorithm.

In the smaller cases tested here, the modified Bellman-Ford appears to offer the best performance, while the dual ascent algorithm is somewhat faster for the larger test cases. In all cases, both of the new algorithms are considerably faster than the standard Bellman-Ford algorithm, even when using the optimization mentioned above. Thus, the new algorithms enable us to solve considerably larger problem instances than before within reasonable time limits and permit the use of larger maps as well as finer-grained discretizations.

VI. Future Work

For simplicity, in this paper all UAVs had the same communication range. In the future we plan to extend the algorithms to handle UAVs with different communication ranges.

We also intend to use the algorithms and formulations used here as a basis for solving the relay problem for a moving target. One of the suggested formulations for edge cost was to use the obstructed volume within a certain radius. In general this will place the UAVs at positions with high visibility. If the target starts to move, it is beneficial to be located at such positions as it is more likely that the UAVs can stay in the same place and maintain communication to the other UAVs, as compared to being placed at positions with lower visibility.

			Timing in milliseconds					
				Modified Bellman-Ford				
$\|N\|$	$\|E\|$	k^*_{max}	Opt. BF	CPT	N^*_k	Mod. BF	Total	Dual Ascent
1395	14322	35	34.6	0.018	0.001	0.014	0.033	2.4
1654	33340	27	68.6	1.75	0.000	6.75	8.5	16.5
1993	10272	33	45.6	0.036	0.002	0.027	0.065	3.7
3411	122246	19	175	2.3	0.005	15.1	17.4	51.0
5294	1134634	14	1079	17.1	0.013	12.2	29.4	28.1
5824	624386	19	823	10.1	0.024	6.7	16.8	16.1
7117	1625726	15	1680	25.0	0.050	18.1	43.1	40.5
9226	5190976	10	3923	79.7	0.000	221.4	301.5	139.0
9226	23086152	5	12344	324.6	0.083	435.0	759.7	574.3
15105	6311322	14	6422	97.3	1.28	81.1	179.7	156.0
18801	13225274	13	12529	194.0	2.17	163.3	359.5	323.7
38542	28327214	17	36382	429.2	6.53	379.3	815.1	729.6

TABLE II

TEST RESULTS. ALL TIMES ARE IN MILLISECONDS AND ARE AVERAGES OF 1000 EXECUTIONS.

VII. CONCLUSION

The use of relay chains is essential to a large variety of UAV and UGV applications where communication range is limited, including but not limited to surveillance tasks. We have presented two algorithms for the relay positioning problem that are efficient both in terms of the number of relays used and in terms of the amount of time required to generate a solution. These algorithms build on a discretization of the continuous problem, but differ in the methods used for solving the all hops optimal path problem in the resulting directed graph: While the first algorithm uses a dual ascent technique, the second builds on the Bellman-Ford algorithm. An extension to efficiently generate independent paths to multiple survey targets was also presented. Both algorithms were shown to be significantly faster than the standard Bellman-Ford algorithm on a variety of problems of different sizes, generating solutions in a small fraction of the time previously required.

ACKNOWLEDGEMENTS

This work has been supported by LinkLab (`www.linklab.se`), the National Aeronautics Research Program NFFP04 S4203, the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII, the Center for Industrial Information Technology CENIIT (06.09) and the Linnaeus Center for Control, Autonomy, and Decision-making in Complex Systems (CADICS), funded by the Swedish Research Council (VR).

REFERENCES

[1] Stuart O. Anderson, Reid Simmons, and Dani Goldberg. Maintaining Line of Sight Communications Network between Planetary rovers. In *Proceedings of the 2003 IEEE/RSJ International Conference of Intelligent Robots and Systems*. IEEE, 2003.

[2] Ronald C. Arkin and Jonathan Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *7th International Workshop on Advanced Motion Control*, 2002.

[3] Oleg Burdakov, Patrick Doherty, Kaj Holmberg, and Per-Magnus Olsson. Optimal placement of communications relay nodes. Technical Report LiTH-MAT-R-2009-03, Linköping University, Department of Mathematics, 2009.

[4] Oleg Burdakov, Kaj Holmberg, and Per-Magnus Olsson. A dual-ascent method for the hop-constrained shortest path with application to positioning of unmanned aerial vehicles. Technical Report LiTH-MAT-R-2008-07, Linköping University, Department of Mathematics, 2008.

[5] Carmen Cerasoli. An analysis of unmanned airborne vehicle relay coverage in urban environments. In *Proceedings of MILCOM 2007*. IEEE, 2007.

[6] Thomas Cormen, Charles Leiserson, and Ronald Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[7] Roch Guérin and Ariel Orda. Computing shortest paths for any number of hops. *IEEE/ACM Transactions on Networking*, 10(5), 2002.

[8] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost graphs. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 1968.

[9] David Hsu, Tingting Jiang, John Reif, and Zheng Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.

[10] David B. Johnson and David A. Maltz. *Dynamic Source Routing In Ad Hoc Wireless Networks*. Kluwer Academic Publishers, 1996.

[11] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.

[12] Qun Li and Daniela Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000.

[13] Hoa G. Nguyen, Narek Pezeshkian, Michelle Raymond, A. Gupta, and Joseph M. Spector. Autonomous communication relays for tactical robots. In *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.

[14] Ramesh Palat, Annamalai Annamalai, and Jeffrey Reed. Cooperative relaying for ad-hoc ground networks using swarm UAVs. In *Proceedings of MILCOM 2006*. IEEE, 2006.

[15] Frank J. Pinkney, Dan Hampel, and Stef DiPierro. Unmanned aerial vehicle (UAV) communications relay. In *Proceedings of MILCOM 1996*. IEEE, 1996.

[16] Tom Schouwenaars. *Safe Trajectory Planning of Autonomous Vehicles*. PhD thesis, Massachusetts Institute of Technology, February 2006.

[17] John Sweeney, TJ Brunette, Yuandong Yang, and Roderic Grupen. Coordinated teams of reactive mobile platforms. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 2002.