Inflight Rerouting for an Unmanned Aerial Vehicle

Amit Agarwal[1], Meng-Hiot Lim[2], Maung Ye Win Kyaw[2], and Meng Joo Er[1]

[1] Intelligent Systems Center, Techno Plaza, Nanyang Technological University
Singapore 639798
{pg0212208t, emjer}@ntu.edu.sg
[2] School of Electrical & Electronic Engineering, Nanyang Technological University
Singapore 639798
{emhlim, p193494}@ntu.edu.sg

Abstract. The availability of new information can often necessitate rerouting of a vehicle. In [1], the set of nonoverlapping useful vicinities of the reconnaissance sites, S, and the 3D space over them was represented using a discrete graph-model. Here, we consider two cases in which an unmanned reconnaissance aerial vehicle must be rerouted. The first scenario involves not visiting useful vicinities, $S_t \subset S$, due to newly reported unacceptable threat-levels in their neighborhood. In the second scenario, visitations to useful vicinities, $S_r \subset S$, is made redundant due to decreased importance of the sites to which they offer a vantage observation-point. Our earlier graph-model [1] is general enough to provide a basic framework for consideration of rerouting in both cases. We give an algorithm to obtain a graph-model, from the earlier one, that allows efficient gene encoding and efficient application of GA operators to obtain new alternative rerouting plans.

1 Introduction and Background Work

Unmanned reconnaissance aerial vehicles (URAVs) are often used for obtaining high definition photographs or thermographs of sites that are an object of battlefield reconnaissance or those hit by natural disasters such as forest fires or an earthquake [2]. Often, the URAVs use maps of threat, terrain and interesting-site information that are obtained *a priori* using satellite imagery, by high altitude aircrafts or other sources. These maps can however suggest false alarms regarding the presence of threats or potential sites of further interest or, can hide the threats or sites of interest in their resolution. Also, the objects that define these interesting features on the map need not be static in time even in the duration of operation of the URAV. In this work, we address a couple of rerouting problems that potentially arise due to one or more of these factors. Our model is more suited for URAVs that operate in hostile zones in which $|T| = \Theta(S)$ (see the following page for the definitions of the symbols).

Terrains masking permits a URAV to fly largely undetected and obtain very high resolution pictures. However, on the downside, several sites of interest about which information could have been obtained from only a few high altitude observation points now need to be covered from several tens of physically separated useful vicinities. Assuming the user's interest in obtaining information on as many sites in a

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 859–868, 2004.

single sortie as possible, terrain masking imposes a natural increase in the number of geographically separated useful vicinities and thereby, the combinatorial problem of planning the visitation-order to these sites becomes interesting. This problem often involves minimization of multiple parameters such as fuel consumption, time-in-the-threat-zone, distance of travel, total turn-angle or threats that are a nonincreasing function of the Euclidean distance to the vehicle. In this work, our rerouting model accounts for the constraints imposed by the last of these parameters.

Our model for rerouting is obtained from a minimal risk density target visitation (MRDTV) graph. In [1] we gave an algorithm to obtain a minimal risk density multigraph from information on the location of the airbase, *A*, location of useful vicinities, *S*, and the location of known threats, *T*, in the theatre of operation. An MRDTV graph can be derived from the multigraph using certain additional information related to operational requirements. An MRDTV graph can better account for threats that occur in route as opposed to those that are almost collocated with the useful vicinities. Moreover, we take care of risks due to revisitations to any useful vicinity in the gene encoding (Section 3.1) and objective function design (Section 3.3). Fig 1 shows an illustration of a typical input. Fig 2a and Fig 2b show the resulting minimal risk density multigraph and an MRDTV graph respectively. The interested reader may look up the algorithmic details in [1].

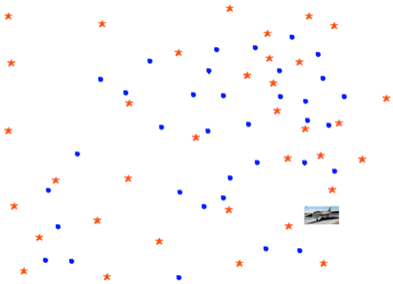


Fig. 1. Input to the algorithm for generating MRDTV graphs. The airbase is represented by the picture (copyright: NFTC, Aeronautical Development Agency). The useful vicinities of sites are the blue dots. The relevant threat sites are the red stars

The input (see Fig 1) information comprises the latitude and longitude coordinates of *A*, *S* and *T* and the strength of each threat $t_i \in T$. Without loss of generality, for the purpose of this illustration, all threats are assumed equipotent. As a brief note on the time complexity, an MRDTV multigraph can be generated in $\mathrm{O}\,S(\log S)$ time. Certain classes of MRDTV graph can be derived from an MRDTV multigraph in $\mathrm{O}(S)$ time.

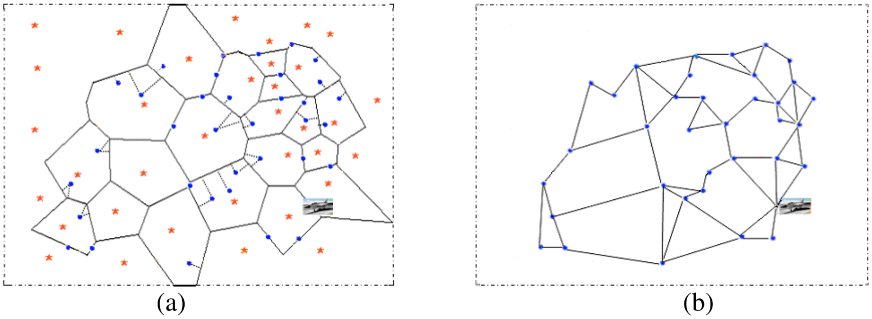

(a) (b)

Fig. 2. The resulting minimal risk density multigraph, $G_M((S \cup A), E)$ and the MRDTV graph $G((S \cup A), E)$

An MRDTV graph is a labeled (not shown in Fig 2b) graph. Each label uniquely identifies the corresponding useful vicinity $s_i \in S$. Thus the problem of finding suitable plans in a continuous domain, such as the one shown in Fig 1, is translated to the problem of finding them in a discrete space, such as the one shown in Fig 2b.

2 Inflight Rerouting

Inflight rerouting is necessitated by the reception of new information at the ground station or changes in the vehicle's perception of its own environment during flight. Two basic scenarios arise – *a*. visitations to certain vicinities $S_r \subset S$ are 'now' found to be unnecessary for the mission. *b*. visitations to certain vicinities $S_t \subset S$ should be suspended due to new knowledge of unacceptable threat-levels present in their neighborhood. The former case differs from the latter in that that the URAV can fly over the no-go useful vicinities in case needed without exposing itself to unacceptable threat levels. In our discussion, for simplicity we assume that $S_t \cap S_r = \Phi$. However, the case, $S_t \cap S_r \neq \Phi$, can be handled with trivial increase in programming effort.

2.1 Algorithm for Generating MRDTV_R Graphs

As mentioned in Section 1, a minimal risk density target visitation graph for rerouting (MRDTV_R graph) can be directly obtained from an MRDTV graph. In this section we give an algorithm to do this and illustrate it via an example. We leave the consideration of special cases out.

The inputs to the algorithm are sets S, S_t and S_r and, the set of useful vicinities that have already been visited.

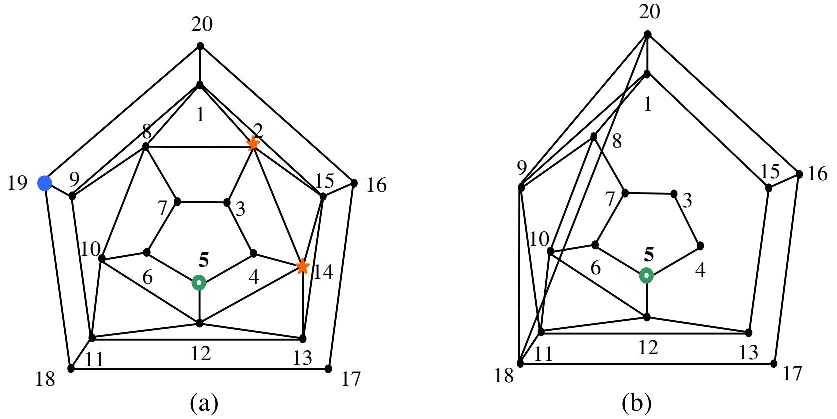
By definition, sites in S_t must be completely avoided. This idea is consistent with removing all vertices corresponding to the elements of S_t and, the edges they induce from the MRDTV graph.

Vertices corresponding to useful vicinities that have been visited before the consideration of rerouting may also be treated as above.

Handling S_r is a little less straightforward – in the gene encoding scheme used here, the mapping from the vertices of a graph to the genes in a chromosome is bijective. Since the objective is to find maximally long sequences of 'connected' (see Section 3.3) useful vicinities so that all sites that should be visited can be visited without requiring revisitations, the HGA will have a strong tendency to include elements of S_r in 'good' flight plans. If however we treat the elements in S_r in the same way as suggested for elements in S_t then, it is certain that no plan will reflect the choice of flying over one or more sites in S_r. This is excessively restrictive for the URAV given the relatively relaxed definition of sites in S_r. Thus, the vertices in the MRDTV graph corresponding to the elements of S_r are handled as follows.

Augment the edge-set of the MRDTV graph with edges that belong to the cliques formed by each set of adjacent vertices of each vertex corresponding to the useful vicinities in S_r. Remove all vertices from the MRDTV graph corresponding to the useful vicinities in S_r.

The application of these techniques yields an MRDTV$_R$ graph. Discovering rerouting plans will now be equivalent to exploring the topology of this graph. A graphical illustration is shown in Fig 3. Vertex *5* is the airbase.



(a) (b)

Fig. 3. An MRDTV graph (a) and the resulting MRDTV$_R$ graph (b) for $S_t = \{2, 14\}$ and $S_r = \{19\}$

The vertices of MRDTV$_R$ graph are a subset of the vertices of the MRDTV graph. Also, its edges are a subset of the union of edges of the MRDTV graph. Thus, an MRDTV$_R$ graph retains all the minimal risk density properties of the corresponding MRDTV graph.

The MRDTV$_R$ graph has $|S + A - S_r - S_t|$ vertices. Upon the application of the algorithm outlined above, its vertices are not labeled by integers from a continuous set. As $|S_r + S_t|$ grows large in proportion to $|S|$, the application of all GA operators whose method includes a random number generator will become inefficient. In order to avoid this, before the GA is applied to discover flight plans, its vertices are relabeled (not shown in Fig 3b) with a continuous set of integers 1 to $|S + A - S_r - S_t|$. The vertex relabeling information is stored in an array $A[|S + A - S_r - S_t|]$ as follows –

```
initialize $j = 1$
for $i = 1$ to $|S+ A|$,
        if $s_i \in (S + A - S_r - S_t)$
                $A[j] = s_i$
                $i$++, $j$++
        else $i$++
```

The GA is applied to the MRDTV$_R$ graph whose vertices are relabeled as above. Once alternative site visitation sequences are generated, the original labeling can be obtained from $A[|S + A - S_r - S_t|]$.

3 The Hybrid Genetic Algorithm

The availability of several competitive site visitation sequences can aid a URAV operator in better decision making. Alternatively, since multi-objective optimization (see Section 1 for its relevance to our problem) is far more difficult, a staged-solution can be helpful. Once several solutions meeting a criterion are obtained, another procedure that better satisfies the constraints on the other parameters can help select between those plans. The implicit parallelism of genetic algorithms [3, 4, 5] in yielding a fairly large set of solutions is thus useful for our application. Our HGA attempts to uncover as many visitation plans, having at least the minimal quality, specified by the user. Critical portions of the hybrid genetic algorithm we used for solving the URAV rerouting problem are outlined in the succeeding subsections.

3.1 Gene Encoding

We use a k-ary gene encoding scheme. Upon the application of the algorithm for generating the MRDTV$_R$ graph as discussed in Section 2, the vertices are labeled with a continuous set of integers from 1 to $|S + A - S_r - S_t|$. The mapping from each vertex of the MRDTV graph to the genes in a chromosome is bijective. Thus, for example, a gene with allelic value *2* represents the vertex labeled *2*, and vice versa. A restriction is placed on this encoding that the vertices corresponding to the useful vicinity that the URAV is heading towards at the time rerouting is necessitated and the airbase are always the first and last gene respectively in any chromosome. Simulations with these restrictions are however deferred.

3.2 Standard Genetic Operators

We use multiple site mutation, and elitism in our HGA. The crossover is implemented with the PMX operator [6] since the standard, single site crossover violates the feasibility of the solutions.

3.3 Fitness Function

Within the consideration of risk minimal rerouting itself (see Section 1 and Section 2 for details), several, not necessarily mutually compatible, measures of a good plan may be defined. A basic could be to discover a set of new alternative routes that begin at the current useful vicinity, or, from the next useful vicinity to which the URAV is in route at the moment when the need for rerouting is communicated or decided onboard, visit all useful vicinities belonging to the set ($S - S_r - S_t$) and possibly (only if absolutely necessary) fly over the useful vicinities in S_r and get back to the airbase without revisiting any useful vicinity in ($S - S_t$). This however would be impossible in some cases, for example, in which the next useful vicinity is a critical node in the underlying MRDTV graph and the airbase and at least one useful vicinity in ($S - S_t$) which has not yet been visited lie in the two unconnected graphs resulting from the removal of the critical node. Other alternative criteria may permit revisitations if made necessary by the underlying graph structure and penalize the number of revisitations. Our goal in this work is to introduce a GA-based approach and additional necessary algorithms for the problem of inflight rerouting for a URAV. In simulations presented here, we have kept the fitness function simple.

We call two genes in a chromosome *connected* if the vertices corresponding to them in the MRDTV graph are adjacent. Let S_c be the useful vicinity that the URAV is about to visit before deciding the new plan for revisitations. Then the fitness of any chromosome is defined as the path length in the MRDTV graph corresponding to the longest, uninterrupted sequence of *connected* genes in that chromosome beginning at S_c.

3.4 Single Gene Migration

In our simulations, the genetic algorithm with only the standard operators performed extremely slowly and, poorly on our test graphs with number of vertices, |V|, greater than 20. Using different combinations of GA parameter rates did not help much in improving the results. The GA would converge to a local optimum. The problems with the exclusive application of the standard operators are two folds. First, the standard operators do an almost blind search – good solutions for our problem cannot be constructively, efficiently, built without exploiting the underlying topology since the MRDTV_R graphs are sparse. The second problem with the exclusive application of standard operators is that that the solution-space is vanishing small in comparison with the problem-space. In fact, part of the answer we are trying to seek is not an optimization problem but, rather an NP-complete decision problem. This, in itself, makes the problem difficult for GAs.

Single gene migration is used to mitigate the aforementioned problems. It is a technique for extensive local search and may be applied in each generation. It takes $O(|X|^2)$ time per chromosome where $|X|$ is the length of a chromosome. It works as follows.

In a chromosome we locate the set of genes that define the chromosome's fitness (Section 3.3). Imagining the set of all genes in the chromosome to constitute a universal set, the negation of this set is the set of all genes that are a candidate for migration. Each gene from the candidate set is taken one at a time and inserted between successive consecutive pairs of genes that define the chromosome's fitness. Each time an insertion is made, the chromosome is checked locally if its fitness has decreased; if not then, the migration for the particular gene is preserved. Else the candidate gene remains in its place.

4 Results

Our algorithm is designed to find as many site visitation plans of the specified quality as requested by the user. The running time varies slightly with the number of such plans requested by the user. We generated up to 10 alterative site visitation plans in all simulations discussed here.

Our dataset comprises of fourteen simulated MRDTV graphs. Randomly, a few vertices in each of these graphs are assigned to the sets S_r and S_t and, a vertex is chosen as S_c. The data is available at [7].

Ten simulations were conducted for each graph on a Pentium IV, 2.4 GHz PC. The RAM of the machine is not a particularly important factor since neither our dataset nor the GA chromosomes require excessive memory.

GA Parameters. We tested our HGA using various combinations of crossover rates, mutation rates and elitism figures. Our simulations showed that for graphs with less than 20 vertices, a population size of about 25 proved good. Also, for graphs with vertices less than 25, sufficient number of good site visitation plans could be obtained in about 20 generations. In graphs with less than 50 vertices, we could obtain sufficient good quality results in about 35 generations. However, for graphs with 100 vertices, we ran the simulation for a maximum of 100 generations. The other GA parameters for simulations reported here are as follows –

Mutation probability = 0.05
Number of mutations per chromosome = 20% of the length of the chromosome
Crossover rate = 1.0

Figures 4 through 7 plot a typical solution in the HGA run versus the generation number. A preprocessing algorithm called *degree-2* graph compression [1] is run on each graph before the HGA. The plots exhibit an interesting feature during the initial generations – they are always extremely poor but rise quickly to near-optimal levels. The single gene migration operator strongly contributes to the steep upward slope of the graph exhibited during the initial generations.

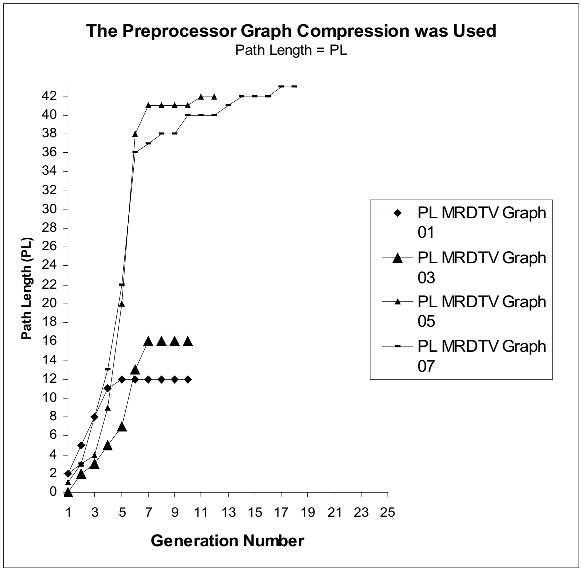


Fig. 4. Typical plots of path lengths obtained using the HGA in graph 01, graph 03, graph 05 and graph 07 v/s the generation number

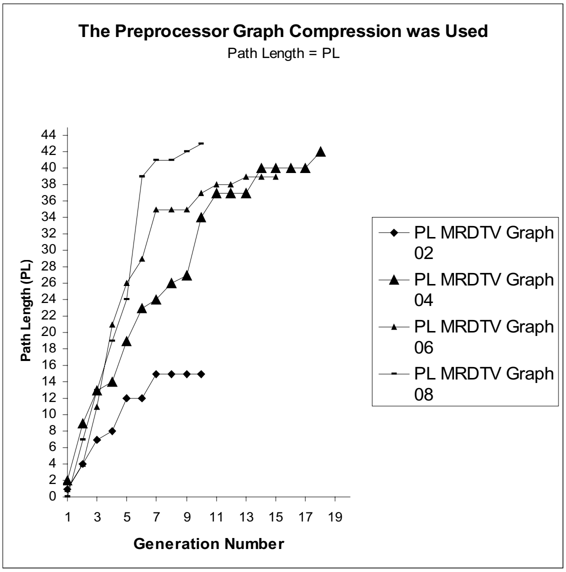


Fig. 5. Typical plots of path lengths obtained using the HGA in graph 02, graph 04, graph 06 and graph 08 v/s the generation number

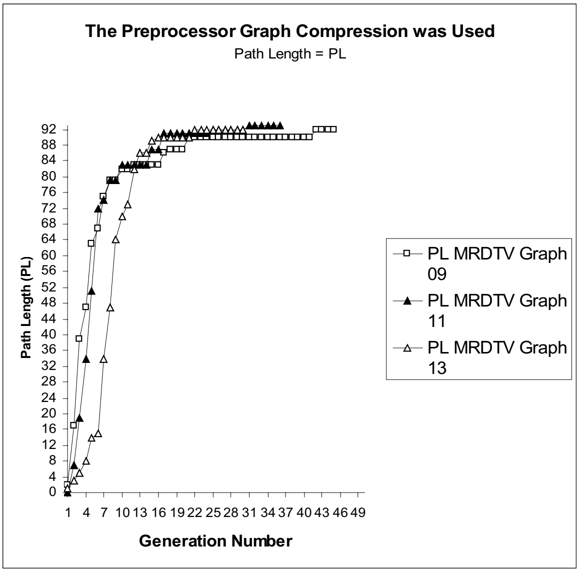


Fig. 6. Typical plots of path lengths obtained using the HGA in graph 09, graph 11 and graph 13 v/s the generation number

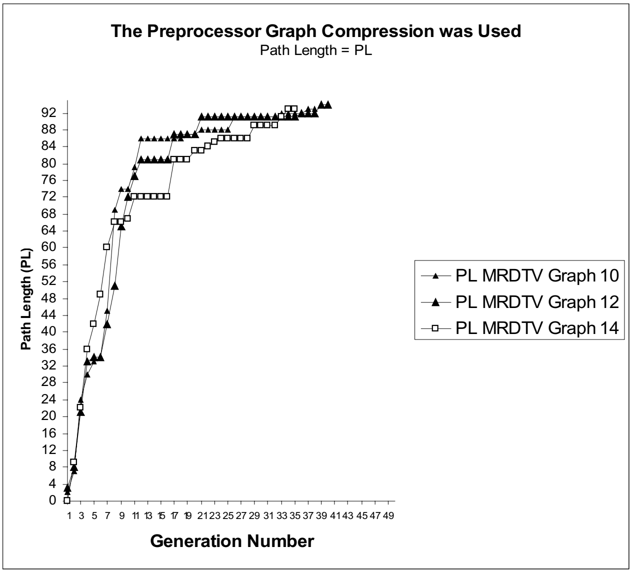


Fig. 7. Typical plots of path lengths obtained using the HGA in graph 10, graph 12 and graph 14 v/s the generation number

5 Conclusion

In this paper, we discussed tow scenarios that can necessitate rerouting for an unmanned reconnaissance aerial vehicle. We used hybrid genetic algorithms to give competitive sets of solutions comprising alternative useful vicinity visitation plans that enable the vehicle to avoid in route risks and conserve time and fuel by not flying over sites with mitigated importance. The input to the HGA is an $\mathrm{MRDTV_R}$ graph. We provided an algorithm to generate it from a *minimal risk density target visitation* graph. A simple algorithm was given to translate the labels of an $\mathrm{MRDTV_R}$ graph to another set of labels. This translation helps greatly in making the implementation of all GA functions that call a random number generator more efficient; especially if $|S_r + S_t| = O(S)$.

References

1. Agarwal A., Lim, M.H., Xu Y.: Graph Compression via Compaction of degree-2 nodes for URAV Visitation Sequencing. UVS Tech 2003. Brussels, Belgium, December 2003
2. Unmanned Systems 2003 Proceedings. Assoc. of Unmanned Vehicle Sys. Maryland, USA, July 2003
3. Erick C.P., James, A.F., Deb, K., Lawrence, D., Roy, R. (eds.): Proc. Genetic and Evolutionary Computation – GECCO 2003. Springer Verlag Pub. Chicago, USA, July 2003.
4. Mitchell, M.: An Introduction to Genetic Algorithms, Reprint Edn. MIT Press, February 1998
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Pub. Co. January 1989
6. Goldberg D.E., Lingle, R.: Alleles, Loci, and the TSP. Proc. of the 1st Intl. Conf. on Genetic Algorithms, New Jersey, USA, 1985
7. http://www.ntu.edu.sg/home/emhlim/dataset/ALY_IRUAV_GECCO04/
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. MIT Press, 2001
9. Unmanned Aerial Vehicles Roadmap 2002 – 2027. US DoD, December 2002
10. Pereira, B.F., Tavares, J., Machado, P., Costa, E.: GVR: a New Genetic Representation for the Vehicle Routing Problem. 13th Irish Conference on Artificial Intelligence and Cognitive Science. Limerick, Ireland, September 2002