Route planning in the space of complete plans

Brian Logan and Riccardo Poli

School of Computer Science
University of Birmingham, Birmingham B15 2TT
b.s.logan@cs.bham.ac.uk

Abstract

Computer Generated Forces (CGF) are software agents which simulate the behaviour of military units or equipment in a distributed interactive simulation environment. Route planning in 'realistic' terrain is a critical task for CGF agents, as many of the agent's higher-level goals can only be accomplished if the agent is in the right place at the right time. In this paper we present a new approach to route planning in complex terrains for CGF agents based on searching the space of complete plans. We describe an implementation of these ideas, the SALIX planner, and report some preliminary results obtained by running the planner on a number of test problems. Our work is also potentially relevant to other applications, e.g. route planning for autonomous vehicles.

1 Route planning in continuous terrains

Computer Generated Forces (CGF) are software agents which simulate the behaviour of military units or equipment in a distributed interactive simulation environment. Such systems are becoming increasingly important in areas as diverse as staff training and equipment procurement. For example, CGF agents offer the potential of dramatically reducing the cost and complexity of mounting training exercises for commanders, with many of the functions carried out by large teams of human controllers replaced by intelligent software agents. These agents must achieve their goals in complex, uncertain and changing environments.

Route planning in 'realistic' terrain is a critical task for CGF agents, as many of the agent's higher-level goals can only be accomplished if the agent is in the right place at the right time. The problem can be viewed as one of finding a minimum-cost (or low-cost) route between two locations in a digitised map which represents a complex terrain of variable altitude. The cost of a particular route is typically a complex function of factors such as the distance travelled, the time required to execute the plan and the visibility of the route. The problem is complicated by the non-linearity of the cost of going from one point to another which varies with the magnitude and the sign of the local gradient (e.g. moving downhill costs much less than moving uphill) as well as the distance travelled.

In this paper we present a new approach to route planning in complex terrains for CGF agents based on searching the space of complete plans. In the next section, we identify some of the characteristics that a route planner should posses, and highlight some of the problems of existing planners. In section 3 we outline the key idea of plan refinement by continuous deformation and describe an implementation of these ideas, the SALIX planner. In section 4, we report some preliminary results obtained by running the SALIX planner on a number of test problems and compare these with results from more conventional planners. In the final section we discuss some of the outstanding problems with the approach and the implementation, and identify a number of directions for future research.

2 Requirements for a planner

In addition to the minimum-cost requirement, we can identify a number of additional characteristics that a route planner should possess (in no particular order):

Continuous and discrete terrain: the planner should be able to cope with route planning over and round all the features to be found in the agent's environment. These include both continuous features such as hills and valleys, 'discrete' features such as rivers and bridges, and 'no-go' areas in which the continuous terrain exceeds some threshold value, e.g. the maximum safe gradient in the case of cliffs.

Anytime planning: the amount of time an agent can afford to spend on planning depends on the current situation: uncertainty about the terrain, the positions of opponents etc. may mean that it is not worth developing a detailed plan. It is therefore desirable if the planner can quickly return a partial plan, or a crude plan only the first segment of which has been developed in detail, as a basis for immediate action. Needless to say, the planner should return reasonably good (partial) plans reasonably quickly.

Plan repair: the environment in which the agent is embedded is constantly changing, and while it would always be possible to replan from a given point (for example to avoid an obstacle), it is often desirable if the agent can patch an existing plan.

However the conventional route planners described in the CGF literature (for example [1, 3, 4, 5, 6]) suffer from a number of problems. These systems typically work by incrementally extending an initial partial plan. Starting from one or more fixed points (often the start point or destination), the planner successively selects atomic plan steps on the basis of their cost and/or the estimated cost of completing the plan. In the case of discretised terrain models, the plan steps are often taken to be the straight line segments connecting the cell centres (though interpolation is also used, see for example [3]). This can work well if the resolution of the terrain model is adequate, but results in artifacts in the case of coarse grained models. Moreover these planners are typically incapable of repairing a plan following a change in the environment that invalidates the unexecuted portion of the current (partial) plan.

3 Searching the space of complete plans

In an attempt to overcome these problems, we have developed an alternative approach which is based on searching the space of complete plans. Rather than incrementally extend a partial plan, we start with a complete plan and refine it by deforming it. We use a novel representation for plans based on the idea that a complete path between any two points A and B can be considered as the result of applying a set of deformation functions to some initial path connecting A and B. If the deformation functions are orthogonal (or at least linearly independent) any plan can be obtained as a linear combination of such functions.

A plan is therefore a linear combination of orthogonal or independent deformations applied to an initial plan. New plans are generated by changing the deformational coefficients of the linear combination of deformations representing the parent plan. This allows a very compact plan representation, since we only need to remember a fixed length vector of coefficients for each plan (one for each deformation function). As a result, the actual route is only implicitly represented and in particular there is no list of points traversed by the plan. The current implementation of the SALIX planner uses a set of eleven independent triangular plan deformation functions shown in Figure 1, and twenty two search operators which increment or decrement by a fixed amount, δ the deformation coefficients of the plan being expanded.

Starting with a initial plan (e.g. the straight line segment connecting A and B) the search proceeds in a manner similar to best-first search. At each iteration the twenty two deformation operators are applied to the unexpanded plan with the lowest cost. Usually, this is a successor (deformation) of the current plan. However if the current plan is a local minimum, an exhaustive search of the list of unexpanded plans will result. (Since we are searching in the space of plans and hence have no clearly defined goal state, some blind search is inevitable.) The search terminates when a user-specified expansion limit, n, is reached.

Our approach has the advantage that the planner is 'anytime' in that the planner can always return the best plan found so far. Moreover, plan repair in response to changes in the agent's environment is straightforward, as the deformation operators can be applied directly to the current plan.

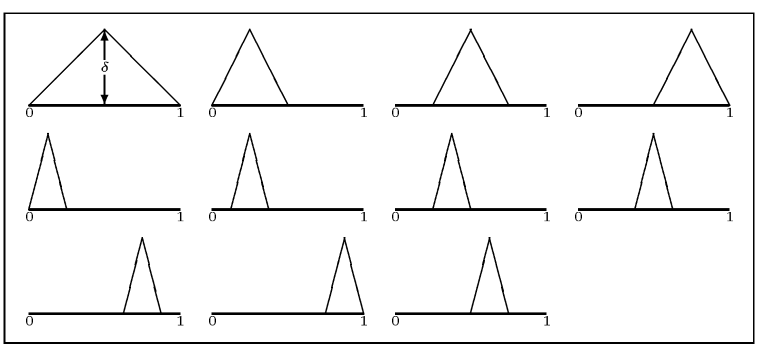

Figure 1: Eleven independent deformation functions.

4 Preliminary results

Despite this rather simpleminded implementation, initial results have been encouraging. In this section we briefly describe the performance of the SALIX planner in planning routes in two sample terrains: *t1*, a 256×256 grid of spot heights representing a 2km $\times$ 2km region of a synthetic terrain model; and *t2*, an 80×80 grid of spot heights representing a 10km $\times$ 10km region of Southern California.[1] For each model we used 50 randomly generated problems consisting of pairs of start and destination positions which are at least four cells apart. Figure 2 shows a typical plan between the points (17, 127) and (244, 30) in the *t1* terrain model generated by the SALIX planner after 25 expansions (lighter grey levels represent higher altitudes).[2]

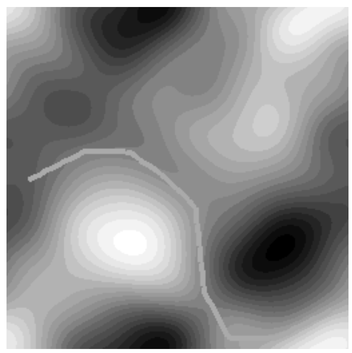

Figure 2: An example route produced by the SALIX planner.

For the purposes of comparison, we also solved the same problems using an A^* planner similar to those described in the CGF literature. The A^* algorithm [2] has a number of attractive properties for route planning problems. A^* search is both complete and optimal.[3] Moreover among optimal algorithms of this type—algorithms that search outwards from the start state(s)—A^* is optimally efficient for any given heuristic function, i.e. no other optimal algorithm is guaranteed to expand fewer nodes than A^*. A^* in its various forms has been used in a number of CGF systems as the basis of their planning component, for example for planning road routes [1], avoiding moving obstacles [4], avoiding static obstacles [5] and for planning concealed routes [6].

The memory requirements of A^* mean that it is impractical in its pure form, and for our tests we used the variant known as A^*_ϵ [8] which is guaranteed to find solutions that can be worse than optimal by at most ϵ. For comparison, Figure 3 shows the plan produced by A^*_ϵ with $\epsilon = 0.1$ on the same test problem as Figure 2.

[1] We are grateful to Richard Penney and Jeremy Baxter at DRA Malvern for providing the terrain models.

[2] Note that to aid the presentation, the z values shown in Figures 2 and 3 have been discretised into twenty steps. However all tests were conducted using the original 'continuous' models.

[3] Strictly, A^* is complete on locally finite graphs—graphs with a finite branching factor provided there is some positive constant c such that every operator costs at least c. (If any operator has negative cost, nothing but an exhaustive search of all nodes would find the optimal solution.)

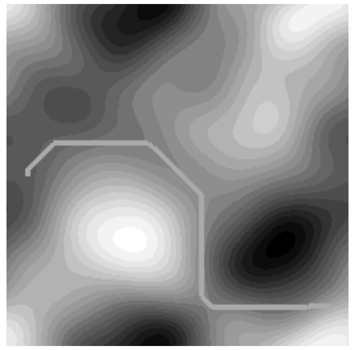

Figure 3: An example route produced by the A^*_ϵ planner.

Both planners used the same cost function which takes as its input a list of cell coordinates in the discretised terrain models. In the case of the SALIX planner, we sampled the plan to to produce the list of coordinates. In the case of A^*_ϵ the operators encode motion from cell to cell and and the resulting list of cells can be used directly as input to the cost function. The A^*_ϵ heuristic function, $h(n)$ (the cost of reaching the goal from the current state), was assumed to be the cost of the straight line plan from the current position to the destination. The resulting plans contained on average about 140 individual steps in the case of the first (*t1*) terrain model and about 40 steps for the second (*t2*) terrain model.

Table 1 shows the average cost of the plans produced by the SALIX planner for both terrain models and for various values of the expansion limit n. The average cost of the plans produced by A^*_ϵ with $\epsilon = 0.1$ were 2335.8 and 17505.7 for *t1* and *t2* respectively. There is a marked difference between the performance of the SALIX planner on the two terrain models. In one case it produces plans which are, on average, within 25% of the (inferred) optimum, while for the second model the planner produces plans which are only within a factor of three of the optimum. The reason for this difference is unclear and further work is required to isolate the characteristics of the terrain models that result in poor performance.

	Number of expansions				
Model	5	25	50	100	500
t1	2852.4	2675.1	2675.1	2675.1	2675.1
t2	44612.0	43717.3	43717.3	43717.3	43717.3

Table 1: average cost of plans produced by SALIX.

Although in some cases the planner can quickly find plans within 25% of the optimum (using less than 25 expansions), the results show the reduction in plan cost rapidly levels off, and it is clear that the SALIX planner is becoming trapped in local minima. The best plans are found after a relatively small number of expansions and increasing the number of expansions does not improve the quality of the plans produced.

In an attempt to overcome this problem, we replaced the best-first search strategy of the SALIX planner with the GA^* algorithm [11]. GA^* is a generalisation of A^* inspired by work in Evolutionary Algorithms in which the operators are not constrained to be unary (i.e. an operator can take more than one state as input) and selection of the next state to be expanded is probabilistic rather than deterministic. One of the major sources of power of Evolutionary Algorithms derives from their use of non-unary operators (crossover), and the use of such operators can significantly improve the power of the search algorithm (especially when $h(n)$ is not an underestimate).[4] Probabilistic selection can lead to important forms of optimality, like the optimum exploration/exploitation tradeoff, i.e. the optimum compromise between the need to sample the search space to collect information and the need to produce good solutions as soon as possible. The algorithm uses a selection probability, s, which controls whether the current best unexpanded plan should be selected for expansion or whether the next best should be considered. We used a simplified version of GA^* (a best-first version with $h(n) = 0$ and no crossover) in our experiments.

Table 2 shows the average cost of the plans produced by the SALIX planner with GA^* and $s = 0.1$ for both terrain models and various values of the expansion limit n.

	Number of expansions				
Model	5	25	50	100	500
t1	3112.9	2703.3	2614.1	2603.7	2603.7
t2	49349.0	41833.3	39888.8	39396.6	39253.4

Table 2: average cost of plans produced by SALIX with GA^*.

As expected, rank selection prevents GA^* from becoming trapped in local minima, and the completeness of the algorithm guarantees that even if it is temporarily trapped by a minimum it eventually explores other parts of the search space. However the initial plans produced by GA^* have a higher cost than those produced by best-first search. In the test examples, between 25 and 50 expansions are required before the average plan cost drops to below that of the plans produced by best-first search. This suggests that the choice of search control should be based on the anticipated time available before the agent has to commit to a plan. If the agent has only a limited amount of time, it may be better to use best-first search; if more time is available then GA^* will eventually produce better plans. (Of course in both cases there is always a plan available if it becomes necessary for the agent to act due to a change in the environment.)[5]

[4] The coefficients of the linear combinations of plan deformation functions map naturally into the finite-length chromosomes of GA^*, but the results reported below do not use crossover.

[5] Unfortunately simply reducing the selection probability with time doesn't result in the best of both worlds, as best-first search tends to fill the initial segment of the list of unexpanded plans with small perturbations of the current best plan, reducing the probability that radically different plans will be explored.

5 Conclusion

We have outlined a new approach to route planning in continuous terrains based on search in the space of complete plans, and briefly described out initial implementation of this approach, the SALIX planner. For agents in an uncertain and changing environment, planning in the space of complete plans has a number of advantages over the ϵ-admissible planners described in the CGF literature. Our approach is inherently anytime, in that the planner can always return the best plan found so far. In addition, plan repair in response to changes in the agent's environment can use the current best plan as the starting point directly without having to estimate how much of the existing plan can be reused.

However the current implementation has a number of limitations. The simpleminded search strategy means that the planner can get trapped in local minima. While using GA^* avoids this problem, it does so at the cost of poorer initial plans, and a more sophisticated search strategy is required which combines the advantages of both best-first and GA^*. The current set of deformation functions is not well suited to planning routes around discrete obstacles. Moreover plans can never extend beyond the normals to the line connecting the start and end points of the plan at the endpoints.[6] It is not clear if this limitation is responsible for the generation of sub-optimal plans in the test problems. However, there is no reason why the set of basis functions could not be expanded to include step or other functions, splines, polylines, or even orthogonal deformation procedures which would behave differently according to the terrain features. Such functions might be enabled and disabled by some more abstract examination of the terrain model and/or the progress of the planner. In addition, the current implementation of the planner is rather slow, as each operator application typically requires costing an entire plan, rather than computing the incremental change in cost resulting from the operator application as with A^*_ϵ. We are currently investigating plan representations that allow us to cost operator applications rather than the resulting complete plans. This problem is currently unsolved.

We are hopeful that these limitations can be overcome and believe that this is an area worth exploring. We are currently investigating a number of extensions to the basic framework described above, including alternative search strategies, e.g. varying plan deformation factor δ with the depth of the search, using different perturbation functions at different times, and using hierarchical search, i.e. sampling the plan produced after a small number of expansions and using the planner to plan routes between the endpoints of the resulting plan segments. Another area which we hope to investigate is plan reuse or case-based planning, to exploit the ability of the perturbation operators to adapt existing, similar plans (for example plans with similar start and end positions, and similar cost functions) to solve new problems.

Acknowledgements

The authors wish to thank Aaron Sloman and all the members of the Cognition and Affect and EEBIC (Evolutionary and Emergent Behaviour Intelligence and

[6] More precisely, the distance along the straight line segment from the start point to the end point is monotonic in d, the distance along the plan.

Computation) groups at the School of Computer Science, University of Birmingham for useful discussions and comments. This research is partially supported by a grant from the Defence Research Agency (DRA Malvern) and a grant under the British Council-MURST/CRUI agreement.

References

[1] C. Campbell, R. Hull, E. Root and L. Jackson. *Route planning in CCTT*, in Proceedings of the Fifth Conference on Computer Generated Forces and Behavioural Representation, Technical Report, Institute for Simulation and Training, pp. 233–244, 1995.

[2] P. E. Hart, N. J. Nilsson and B. Raphael. *A Formal basis for the heuristic determination of minimum cost paths*, IEEE Trans. Syst. Sci. Cybern. SSC-4(2), pp. 100–107, 1968.

[3] B. Hoff, M. F. Howard and D. Y. Tseng. *Path planning with terrain utilisation in ModAF*, in Proceedings of the Fifth Conference on Computer Generated Forces and Behavioural Representation, Technical Report, Institute for Simulation and Training, pp. 255–263, 1995.

[4] C. Karr, M. Craft and J. Cisneros. *Dynamic obstacle avoidance for Computer Generated Forces*, in Proceedings of the Fifth Conference on Computer Generated Forces and Behavioural Representation, Technical Report, Institute for Simulation and Training, pp. 245–254, 1995.

[5] C. Karr and S. Rajput. *Unit route planning*, in Proceedings of the Fifth Conference on Computer Generated Forces and Behavioural Representation, Technical Report, Institute for Simulation and Training, pp. 295–304, 1995.

[6] M. Longtin and D. Megherbi. *Concealed routes in ModSAF*, in Proceedings of the Fifth Conference on Computer Generated Forces and Behavioural Representation, Technical Report, Institute for Simulation and Training, pp. 305–314, 1995.

[7] P. Norvig. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*, Morgan Kaufman, 1992.

[8] J. Pearl. A^*_ϵ *– An algorithm using search effort estimates*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 4, No. 4, pp. 392–399, 1892.

[9] J. Pearl. *Heuristics: intelligent search strategies for computer problem solving*, Addison-Wesley, 1984.

[10] I. Pohl. *The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving*, Proc. IJCAI-73, pp. 20–23, 1973.

[11] R. Poli and B. Logan. *On the relations between search and evolutionary algorithms*, Cognitive Science Research Paper CSRP-96-7, School of Computer Science, University of Birmingham, 1996.