

Hybrid Co-evolutionary Motion Planning via Visibility-Based Repair

Gerry Dozier Shaun McCullough Edward Brown Jr. Abdollah Homaifar Mar-wan Bikdash

NASA Center for Autonomous Control Engineering
North Carolina A&T State University
Greensboro, NC 27411, USA
gvdozier@ncat.edu

Abstract

This paper introduces a hybrid co-evolutionary system for global motion planning within unstructured environments. This system combines the concept of co-evolutionary search along with a concept that we refer to as the visibility-based repair to form a hybrid which quickly transforms infeasible motions into feasible ones. Also, this system makes use of a novel representation scheme for the obstacles within an environment. Our hybrid evolutionary system differs from other evolutionary motion planners in that (1) more emphasis is placed on repairing infeasible motions to develop feasible motions rather than using simulated evolution exclusively as a means of discovering feasible motions, (2) a continuous map of the environment is used rather than a discretized map, and (3) it develops global motion plans for multiple mobile destinations by co-evolving populations of sub-global motion plans. In this paper, we demonstrate the effectiveness of this system by using it to solve two challenging motion planning problems where multiple targets try to move away from a point robot.

KEYWORDS: Evolutionary Algorithms, Co-evolution, Motion Planning, Visibility Graph, Visibility-Based Repair, Radcliffe's Crossover, Seed Crossover, Seed Mutation, Global Motion Planning, Sub-Global Motion Planning

1 Introduction

Evolutionary Algorithms (EAs) are search methods that evolve a population of candidate solutions through the use of natural selection. EAs typically solve difficult problems for which traditional search paradigms yield unsatisfactory results. EAs have been successfully applied to a variety of areas such as design optimization, machine learning, constraint satisfaction, and constrained optimization [9]. Recently, there has been a growing number of successful applications of EAs to the area of motion planning [2, 3, 6, 9, 11]. The Motion Planning Problem [5, 7] can be stated as follows. Given an environment $E(R, X, T, O)$ where R represents some robot, X represents the starting point (or point of origin), T represents the goal or destination point and O represents a set of obstacles, find a collision free (feasible) path from X to T (path planning phase) that R can traverse (navigation phase). Many of the evolutionary motion planning systems rely on simulated evolution almost exclusively as a means of discovering a feasible motion for R .

This paper introduces a hybrid co-evolutionary system for global motion planning within unstructured environments. This system combines the concept of co-evolutionary search along with a concept that we refer to as the visibility-based repair to form a hybrid which quickly transforms infeasible motions into feasible ones. Also, this system makes use of a novel representation scheme for the obstacles within an environment. Our hybrid evolutionary system differs from other evolutionary motion planners in that (1) more emphasis is placed on repairing infeasible motions to develop feasible ones rather than using simulated evolution exclusively as a means of discovering feasible paths, (2) a continuous map¹ of the environment is used rather than a discretized map, and (3) it develops global motion plans for multiple mobile destinations by co-evolving populations of sub-global motion plans. In this paper, we demonstrate

¹[6, 9] also use continuous maps

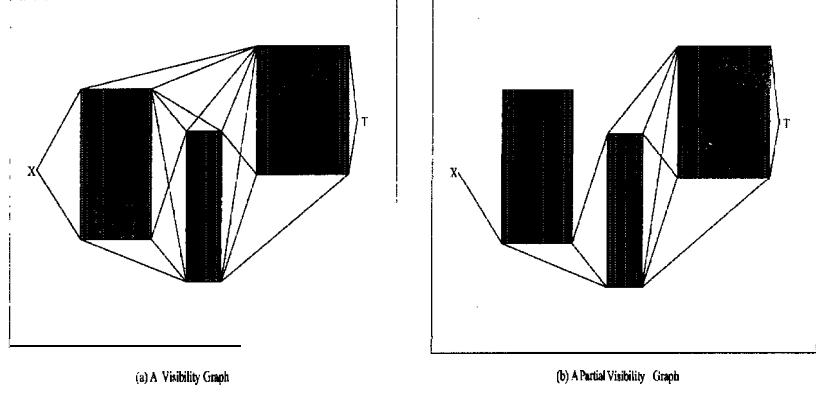


Figure 1: Full and Partial Visibility Graphs

the effectiveness of this new hybrid system by using it to solve two challenging motion planning problems where multiple targets try to move away from a point robot.

The remainder of the paper is organized as follows. Section 2 provides a brief introduction to the concept of visibility-based search. In Section 3, we present our hybrid co-evolutionary system called GEPOA-II (a Global Evolutionary Planning and Obstacle Avoidance system) in detail and, in Section 4, reintroduce our test suite. In Section 5, we present our results and conclusions and, in Section 6, we discuss some directions for future research.

2 Visibility-Based Search

A visibility-based search algorithm can be regarded as any search procedure that uses a visibility graph (VG)[3, 5] to aid in the discovery of feasible paths. A VG is a graph, (V, E) , where V is the set of all vertices of the obstacles within an environment including the coordinates of the robot (or starting position) and the destination, and E is the set of all edges connecting any two vertices in V that do not pass through any obstacles within the environment.

Figure 1a shows an example of a visibility graph. Notice that X is connected only to the vertices that are reachable by way of a straight-line segment that does not cut or pass through (or violate) any obstacles. Notice also that the vertices which are visible to X are connected, in similar fashion! to other vertices that are visible to them (again by way of straight-line segments). Once a VG has been constructed for a given environment, usually an A^* search algorithm is used to find the shortest path between starting and destination points.

It is not always necessary to construct a complete VG for an environment. Some researchers have experienced a great deal of success with using partial VGs(PVGs) [3]. Figure 1b shows an example of a PVG. An advantage of using a PVG rather than a VG is that a PVG requires less computation. One disadvantage of using a PVG is that it may not contain an optimal path. Both of these methods perform poorly on path planning problems with dynamic environments.

3 GEPOA-II

GEPOA-II is a co-evolutionary version of a successful hybrid evolutionary planner named GEPOA-I[4]. We developed GEPOA-II in an effort to take advantage of the decompositional nature of global motion planning. In GEPOA-II, global motion planning problems are decomposed into smaller, sub-global problems. A number of EAs are then used to quickly develop sub-global motion plans (one EA for each sub-problem) which are combined to represent a global motion plan. In this section, we discuss seven salient attributes of GEPOA-II. These attributes are as follows: the representation of environments, the concept of visibility-based repair, the representation of candidate paths (CPs), the visibility-based repair algorithm, the evaluation function, the selection algorithm, and the evolutionary operators. We conclude this section by providing the parameter settings for three GEPOA-II hybrids that will be tested with a test suite of two path planning problems.

3.1 Representation of an Environment

An obstacle within an environment is represented as a set of $\lceil \frac{k_i}{2} \rceil$ intersecting line segments, where k_i represents the number of vertices of obstacle i . Each line segment, i, j , connects two distinct vertices $v(i, j)$

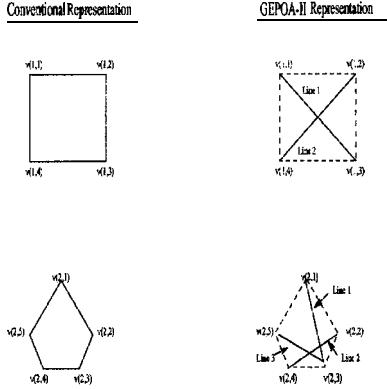


Figure 2: obstacle Representation in GEPOA-II

and $v(i, j + \lfloor \frac{k_i}{2} \rfloor)$ (where $j \leq \lfloor \frac{k_i}{2} \rfloor$) of obstacle i . Associated with each vertex within the environment, is a value which ‘represents the number of obstacles that the vertex is contained by. This value is referred to as the ‘containment value’ (CV) of a vertex. If a vertex lies along the boundary of an environment its CV is assigned a value of ∞ .

Figure 2 provides an example of how obstacles are represented in GEPOA-II. Notice, in Figure 2, that the four sided obstacle (Obstacle 1) is represented by only two lines in GEPOA-II. Line 1 of Obstacle 1 connects vertices $v(1, 1)$ and $v(1, 3)$ while Line 2 connects vertices $v(1, 2)$ and $v(1, 4)$. Obstacle 2 has five sides and is represented in GEPOA-II using three lines. Line 1 connects $v(2, 1)$ and $v(2, 3)$, Line 2 connects $v(2, 2)$ and $v(2, 4)$, and Line 3 connects $v(2, 3)$ and $v(2, 5)$. Since each of the vertices are contained by only one obstacle, the CV for each vertex is 1.

3.2 Visibility-Based Repair

Visibility-based repair (VBR) is performed as follows. When an obstacle, o_i , lies along a straight-line segment between two nodes P and Q , each line of o_i is checked to see if it is intersected by PQ . If a line of o_i is intersected by PQ then a repair node is created using the following set of rules:

- Rule 1: if the CVs of a line’s vertices are both equal to one, then the repair node is selected to be a point along an extent² of the vertex which is closer to the point of intersection;
- Rule 2: if the CVs of a line’s vertices are different, then the repair node is selected to be a point just outside of the vertex which has the lower CV;
- Rule 3: if the CVs of a line’s vertices are greater than one and equal, then the repair node is selected to be a point just outside of the vertex which is farther from the point of intersection.

Figure 3 shows an example of how VBR can be used to transform an infeasible path into one that is feasible. In Figure 3a, an infeasible path XPT is shown. The path XPT is infeasible because the line segment XP passes through Obstacle 1 and the line segment PT passes through Obstacle 3. Before proceeding further, notice that each vertex in the environment shown in Figure 3a has a CV of one.

Using VBR, the line segment XP can be repaired to XAP . Since XP intersects Line 1 of Obstacle 1, a repair node corresponding to a point just outside of either $v(1,1)$ or $v(1,3)$ must be selected. By applying Rule 1, Node A , which corresponds to a point just outside vertex $v(1, 1)$, is selected as the repair node.

Similarly, the line segment PT can be repaired to $PBCT$. Again Rule 1 must be applied to Line 1 and Line 2 of Obstacle 3. The repair node that results from the intersection of PT and Line 1 is Node B . The repair node that results from the intersection of PT and Line 2 is Node C . Figure 3b shows the result of using VBR on XPT . The repaired, feasible version of XPT is $XAPBCT$.

3.3 Representation of Candidate Paths

An individual representing a candidate path (CP) contains two fields. The first field is a chromosome which contains a gene corresponding to the cartesian coordinates of each node of the path (where each node of a path is connect by a straight-line segment). The second field is called the seed. The seed of an individual is the gene that will be crossed or mutated to created an offspring. Initially, an individual will have only three genes: the start gene, the seed gene and the destination gene. Repair genes are inserted

²The distance outside of an obstacle at which a repair node is placed is a user specified parameter.

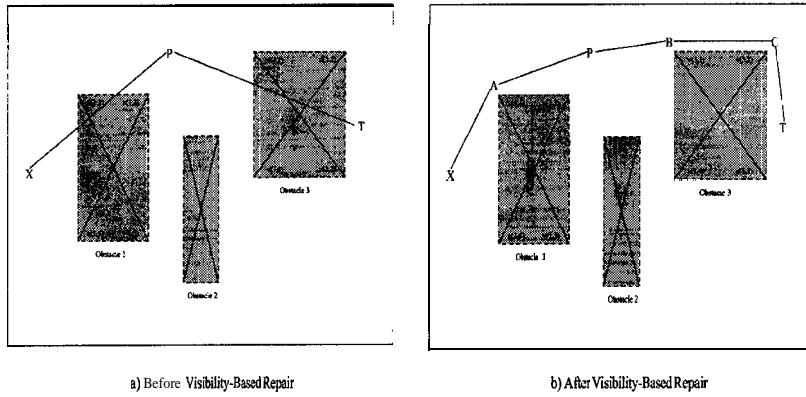


Figure 3: Visibility-Based Repair

into the chromosome by the visibility-based repair algorithm each time a straight-line segment of an individual is found to pass through an obstacle. The third field is a value referred to as the violation distance. The ‘violation distance’ represents the euclidean distance of the CP which cuts through one or more obstacles. The fourth field records the euclidean distance of the path from the start to destination.

3.4 The VBR Algorithm Used by GEPOA-II

Given a CP, the VBR algorithm used by GEPOA-II works as follows. Each obstacle within the environment is checked with each straight line segment from the start gene to the destination gene of the CP until a segment is found that passes through the obstacle. The infeasible segment is repaired via VBR and the process is repeated using the next obstacle.

For an example of how this repair algorithm works, notice once again Figure 3. When given the path XPT the algorithm works as follows. Obstacle 1 is checked to see if it is violated by segment XP . Since it is, a repair gene (Node A) is generated and Obstacle 2 is then considered. Obstacle 2 is checked to see if it is ‘cut’ by segment XA . Since it is not ‘cut’ by segment XA , Obstacle 2 is checked with segment AP then segment PT . Since there are no more segments to inspect, Obstacle 3 is considered. Obstacle 3 is checked to see if it is ‘cut’ by segments XA , and AP . Finally, Obstacle 3 is checked to see if it is ‘cut’ by PT . Since it is, two repair genes are generated (Nodes B and C) and the algorithm terminates.

3.5 Evaluation and Selection

The evaluation function computes the euclidean distance of each straight line segment of the path that an individual represents as well as the euclidean distance of each segment of the path that passes through one or more obstacles, called the violation distance. GEPOA-II uses a modified version of tournament selection, with a tournament size of 2, to select individuals to become parents. The selection process is as follows. Two individuals are randomly selected from the current population. If the violation distances of the two are different then the individual with the smaller violation distance is selected to be a parent. If the violation distances are the same then the individual with the smaller ‘overall’ distance is selected.

3.6 The Evolutionary Operators

GEPOA-II uses three operators to create and/or refine individuals. The first operator, the VBR algorithm, is applied to parents representing infeasible CPs 25% of the time as well as all newly created offspring. The other two operators, (1) a version of Radcliffe’s Crossover [10] that we refer to as seed crossover and (2) a version of uniform mutation we refer to as uniform seed mutation, are applied only to feasible CPs.

Seed crossover is as follows. Given two seed genes $s_1 = (x_1, y_1)$ and $s_2 = (x_2, y_2)$, a seed gene for an offspring, $s_{off} = (rnd(x_1, x_2) + N(0, 4.0), rnd(y_1, y_2) + N(0, 4.0))$, is created where rnd is a uniform random number generator and $N(0, 4.0)$ is a gaussian random number with zero mean and a standard deviation of 4.0. The resulting offspring, (X, s_{off}, T) , has a chromosome containing three genes: a gene corresponding to the start node, the seed node, and the goal node. The offspring then undergoes VBR and may have additional repair genes added by the VBR algorithm.

In uniform seed mutation, either the x or y coordinate of a parent is mutated using uniform mutation to create a seed gene for an offspring. A resultant offspring created by seed mutation is similar to one created by seed crossover in that it also has a chromosome containing three genes. Once again the offspring undergoes VBR and may have additional repair genes added by the VBR algorithm.

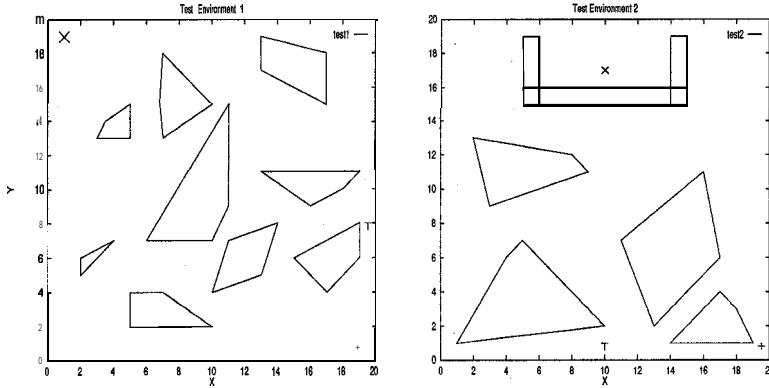


Figure 4: The Two Test Environments

3.7 Attribute Settings for Three Co-evolutionary Hybrids

Three GEPOA-II co-evolutionary hybrids with population sizes of 5, 10, and 20 were tested with a test suite of two motion planning problems. These three hybrids differ only in the size of the two populations that they co-evolve. The first of the two populations co-evolved by these hybrids contains motions from the point robot to the first target. The second population contains motions from the first target to the second target. The hybrids randomly generate their initial populations, use a seed crossover rate of 0.5 and a uniform seed mutation rate of 0.5. For each generation, only one offspring is created. This offspring replaces the worst fit individual in the population.

Every ten generations R is advanced a maximum of one unit along the shortest path to T_1 developed by the system. The amount of advancement is based on the change of direction of R . If no change of direction is needed, R is advanced one unit. If the change of direction is 90° or greater, R is not advanced. After R has been advanced, T_1 is allowed to move a half unit in a direction (north, south, east, or west) furthest away from the current position of R , and T_2 is allowed to move a half unit in a direction furthest away from T_1 ³. This process is repeated until R reaches T_1 . At this point, the second population is used exclusively to navigate R to T_2 .

4 The Test Suite

Figure 4 shows our test suite of two motion planning problems that will be solved by the three hybrids described above. In each of the test environments, X represents the starting point, T represents the first mobile destination, and "+" represents the second mobile destination. In Test Environment 1, T is located at (8.0, 19.5) and "+" is located at (1.0, 19.0). In Test Environment 2, T is located at (1.0, 10.0) while "+" is located at (1.0, 19.5).

5 Results and Conclusions

Each hybrid was run 50 times on each of the test environments. Each run was allotted a maximum of 200 moves. A run was considered successful if the robot reached both targets. On each run, the hybrids never failed to find a feasible solution within the initial population. Also, the motion plans evolved by the hybrids allowed the robot to reach the target 100% of the time. This is an indication of how effective visibility-based repair can be.

The results of the performances of the hybrids on both of the test environments are presented in Figures 5 and 6. They are organized into a matrix where the columns (from left to right) correspond to:

- the population size of the hybrid (P),
- the average length of the first feasible solution found ($\overline{Ln.1^{st}}$) from R to T to +,
- the standard deviation of the length of the first feasible solution found during each successful run ($\sigma(Ln.1^{st})$),
- the average number of moves needed to reach the first target ($Moves\bar{T}$), and
- the standard deviation of the number of "moves need to reach the first target" ($\sigma(Moves1)$),
- the average of the total number of moves needed to reach the first and second target ($Total$), and
- the standard deviation of the total number of moves needed to reach the first and second targets ($\sigma(Total)$).

In Figure 5, one can see that as the larger the population size the better the performance. This is especially the case when viewing the $Total$ column. However larger population sizes require more

³ Each time R , T_1 , or T_2 is moved each individual in the population is re-evaluated.

P	$Ln.1^{st}$	$\sigma(Ln.1^{st})$	$Moves1$	$\sigma(Moves1)$	$Total$	$\sigma>Total$
5	34.125	3.250	34.960	7.665	73.260	14.574
10	32.718	1.844	33.980	7.928	68.980	13.568
20	31..746	1.293	31.600	1.918	62.780	9.876

Figure 5: performances on Test 1

P	$Ln.1^{st}$	$\sigma(Ln.1^{st})$	$Moves1$	$\sigma(Moves1)$	$Total$	$\sigma>Total$
5	38.313	1.432	50.920	19.573	84.060	21.018
10	37.077	0.823	49.000	21.832	81.940	23.298
20	36.593	0.664	44.460	14.147	75.680	13.945

Figure 6: Performances on Test 2

computational effort. This is due to the fact that each time R , T , or “+” moves every individual in both populations must be updated. This means that $P = 20$ performs four times the computational effort than $P = 5$. In Figure 6, one can see once again that $P = 20$ has the best performance. One interesting observation is that the differences in the performances of the three hybrids on Test Environments 1 and 2 with respect to $Total$ have remained virtually the same. It will be interesting, in future efforts, to see if this constant remains over a larger test suite of problems.

6 Future Work

At present, we are particularly interested in (1) improving the search efficiency of small population evolutionary motion planners and (2) reducing the computational effort of larger population evolutionary motion planners. We also experimenting with a number of specially designed order-based and relational crossover operators [1, 8] that make better use of knowledge gleaned from the environment. Because of the speed at which these GEPOA-II hybrids develop feasible paths, our future work will be devoted to the development of hybrid motion planning systems that make use of VBR as well as other traditional motion planning concepts. This will allow for the development of evolutionary hybrids that incorporate the best of both traditional and evolutionary motion planning concepts.

Acknowledgment

This research is partially funded by grants from the NASA Autonomous Control Engineering Center under grant number ACE-48146. The authors wish to thank them for their financial support.

References

- [1] Brown, Edward, Jr. (1996). Relational Crossover with Seed Displacement, Technical Report ACETR-MP-96-03, NASA ACE Center, North Carolina A & T State University, Greensboro.
- [2] Ashiru, I. and Czarnecki, C. (1995). Optimal Motion Planning for Mobile Robots Using Genetic Algorithms, *Proceedings of the 1995 IEEE/IAS International Conference on Industrial Automation and Control*.
- [3] Chang, T.-Y., Kuo, S-W., and Hsu, J. Y.-J. (1994). A Two-Phase Navigation System for Mobile Robots in Dynamic Environments, *Proceedings of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pp. 297-300.
- [4] Dozier, G., Esterline, A., Homaifar, A., and Bikdash, M. (1996). Hybrid Evolutionary Motion Planning via Visibility-Based Repair, submitted to: *The 1997 IEEE International Conference on Evolutionary Computation*.
- [5] Fujimura, K. (1991). *Motion Planning in Dynamic Environments*, Springer-Verlag.
- [6] Hocaoglu, C. and Sanderson, A. (1996). Planning Multi-Paths Using Speciation in Genetic Algorithms, *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 378-383.
- [7] Latombe, J.C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers.
- [8] McCullough, Shaun. (1996). An Insertion Based PMX Operator for a Genetic Tracking System with Multiple Targets, Technical Report A CETR-MP-96-04, NASA ACE Center, North Carolina A & T State University, Greensboro.
- [9] Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag.
- [10] Radcliffe, N.J. (1991). Formal Analysis and Random Respectful Recombination, *Proceedings of the Fourth International Conference on Genetic Algorithms*.
- [11] Shibata, T., and Fukuda, T. (1993). Robot Motion Planning by Genetic Algorithm with Fuzzy Critic, *The 8th IEEE International Symposium on Intelligent Control*, pp. 565-570.