Automation and Robotics in Construction XVI © 1999 by UC3M

Path Planning of Construction Manipulators Using Genetic Algorithms

PL. Sivakumar
Research Asst., Dept of Civil Eng., I.I.T. Madras, India

Koshy Varghese
Asst. Prof., Dept. of Civil Eng., I.I.T. Madras, India – koshy@civil.iitm.ernet.in

N. Ramesh Babu
Asso. Prof., Dept. of Mech. Eng., I.I.T. Madras, India

Abstract

This paper presents the work done to investigate the potential of applying Genetic Algorithms for path planning of construction manipulators. Construction manipulators can take the shape of specialized equipment such as the pipe manipulator or common equipment such as cranes. The need for automating the path of construction manipulators has been identified in a number of past studies. Path planning is a well-established area in robotics research, work in this area focuses on identifying new algorithms and techniques to improve the accuracy and efficiency of the automated path planning process. The application of Genetic Algorithms for robotic path planning is a relatively new development. This paper reviews different construction situations in which path planning is required and identifies the factors influencing path selection in construction situations. Genetic Algorithm models to find satisficing paths are proposed. The GA model and initial results for a simple test problem using a 2 DOF planar manipulator are also presented.

1.0 Introduction

This paper presents the work done to investigate the potential of applying Genetic Algorithms for path planning of construction manipulators. Construction manipulators can take the shape of specialized equipment such as the pipe manipulator or common equipment such as cranes. In all cases, manual path planning is a cumbersome process. Tasks requiring the representation of space, modeling the kinematics of the equipment and checking for interference are not suited for manual analysis. As a result, the productivity of detailed operations planning becomes very low. The automation of the path planning task would reduce the chances of human error, increase planning productivity and allow the planners to evaluate alternate plans and thereby arrive at a better plan.

Path Planning is a well-established area in Robotics. Numerous techniques have been formulated and tested. These techniques are classified based on criteria such as dimension of space, mobility of manipulator and obstacles, representation of space and nature of information gathering. A description of the basic issues involved in path-planning and concepts of conventional techniques are presented effectively by Latombe [Latombe 1991] and Schwartz [Schwartz et al. 1988]. The application of Genetic Algorithms for path planning is relatively new. Genetic Algorithms are optimization techniques based on the mechanics of natural selection and genetics. In the path planning domain, they have been used as a solution tool to solve the inverse kinematics problem as well as a technique to determine optimal paths through real space and configuration space. As Genetic Algorithm implementations can be paralleled easily, it offers a potential to determine near-optimal paths through complex spaces in reasonable time. A fundamental coverage of genetic algorithms have been presented by Goldberg [Goldberg 1995].

Applications of conventional path-planning techniques to construction manipulators have met with limited success. Assumptions on the geometry of obstacles and motion of manipulators have to be made in order to obtain reasonable performance. These assumptions tend to negate the practical utility of the system. Further, there are differences in the criteria used to assess the suitability of a path in conventional robotics and construction situations. Although the current scope of this work is limited to off-line planning, the results from the system are expected in reasonable time. Recent work utilizing the configuration space representation with a heuristic search algorithm required several hours of execution time to determine a path for a 3 DOF crane within a realistic model of lift area [Reddy 1998]. Thus when situations involving multiple manipulators or more complex workspaces are encountered, the performance of the system will be unacceptable. Based on this experience, it was decided to investigate alternate approaches such as Genetic Algorithms for path planning of construction manipulators.

This paper consists of six sections. A brief review of the past work in the areas of path planning in construction and path planning using genetic algorithms are presented in the next section. The third section compares the factors influencing path optimization of construction manipulators with conventional robotics. Genetic Algorithm models incorporating these factors are proposed and discussed in the fourth section. The fifth section presents results obtained by implementing a genetic algorithm based planner for a test case using a two DOF manipulator arm in two dimensional space. Finally the conclusions of the study are presented.

2. 0 Past Work

This section briefly reviews some of the work done in the area of path planning of construction manipulators and application of genetic algorithms for path planning.

2.1 Construction Manipulators

The need and applicability of large scale manipulators in the construction environment are discussed by Hsieh. [Hsieh et al.1993]. The activities identified in this study include piping erection, elevated concrete placement, painting, structural steel erection, scaffolding erection and demolition. The availability of a reliable path-planning tool will assist the planner to plan these operations accurately and efficiently.

A path-planner based on dynamic programming concepts was developed for a six degree of freedom pipe manipulator [Alciatore 1989]. The implementation and testing of this algorithm was carried out in the simulation/visualization environment Walkthru on a Silicon Graphics platform and it's performance was reported to be acceptable in an environment where only the pipe racks were considered as obstructions.

A system, called Path-finder was developed for a generic construction manipulator [Morad et al 1992]. It utilizes a heuristic search method to determine a cost-effective path through state-space representing various legal positions of the manipulator. This system was also implemented and tested in the simulation/visualization environment Walkthru.

A system, to plan the lift paths of crane operations was developed using configuration space concepts in conjunction with heuristic search [Reddy 1998]. The system was implemented in a CAD environment on a Pentium platform. The tests carried out showed that the accuracy and performance of the system is acceptable for simple test cases. However, the performance requires significant improvement to be of practical value to the planner. Based on these findings, it was decided to investigate the application of more recent techniques such as Genetic Algorithms.

2.2 *Path Planning Using Genetic Algorithms*

Simple Genetic Algorithms (SGA) yield reasonably good results for wide range of practical problems. The basic concept of a GA is to define the search space in the form of a genetic model through a coding scheme. This scheme transforms the decision variables into a binary string. The initial population is generated using a random function. The fitness of each string is then calculated and a mating pool, consisting of the fittest strings is created using the reproduction operator. The crossover operator is applied to this mating pool to create a new generation of strings. A mutation operator which randomly change the characters in a population is also applied to create diversity in the population. The fitness of the offspring is calculated and the same process repeated until the fitness level converges.

GA's have been applied to solve the path planning problem by: (i) searching for a suitable path in real space and then solving the inverse kinematics problem to transform the path to an optimal trajectory for the manipulator. (ii) searching for a suitable path in configuration space. As the complexities of the path planning can be high, SGA cannot give good results. In such cases, special operators are used to improve the performance of GA's.

There are a variety of special operators in GA which have been investigated in relation to path planning. Yuval Davidor used variable string length GA for trajectory generation and optimization of order dependent production processes. A new crossover operator called Analogous Crossover was used in which matching parameter for cross over will be according to the phenotypic function rather than genotypic position. [Davidor 1991]

A path planning algorithm was developed based on Minimal Representational Size Cluster GA (MRSC GA). It was used to plan multi-paths with Speciation in GA [Sanderson 1996].

Reverse transcription operator (that plays the role of cross over and selection simultaneously) has been used for trajectory generation of redundant manipulators [Kuboto et. al. 1997].

Coevolutionary algorithms also have been used for path planning in a 2-D grid with static obstacles [Jan Paredis 1997].

Floating point vector strings and customized operators to perform path distribution and relaxation were used to enhance SGA by Zalzala [Zalzala 1998] for collision free motion planning of robotic manipulators.

Zomaya and Gill proposed GA solutions to inverse kinematics problem and conducted a number of experiments with parallel genetic algorithms to solve collision avoidance problems [Zomaya et al.1998].

Numerous studies are currently investigating the performance of Genetic Algorithm based path planning. A key objective of these investigations is to find specialized GA operators and techniques that can efficiently solve specific path planning problems. This is comparable to the specialized heuristics used in conventional search techniques once domain knowledge about a particular search space is available.

3.0 Comparison of Construction and Robotic Path Planning

In this section, the criteria governing the optimality of paths for construction manipulators – specifically cranes, with conventional robotics are presented.

For construction operations such as a crane lift, the primary objective in selecting the path is safety. Safety is ensured by selecting a path which will not overload the equipment, have adequate clearance from normal obstacles, totally avoid obstacles such as power lines and avoid lifting over hazardous material. After the safety is ensured, the planner attempts to address other points such as minimizing changes in operation, minimizing the distance of the lift and reducing riskier movements such as luffing. This situation is different from the standard robotic path planning problem where the primary objective is usually to minimize distance while avoiding obstacles.

Further, there are differences in the kinematic constraints of the manipulators in both situations. For example:
(i) Cranes can have telescoping capability which is not common in robots. (ii) Heavy loads are freely suspended from the end effector of the cranes by steel wire ropes where as relatively less weight objects are gripped by the end effector of robot manipulators.
(iii) Hoisting is an unique operation in case of crane manipulators. (iv)Lifting of loads using multiple cranes is a common occurrence whereas the use of multiple cooperative robots is not so common.
(v) For cranes which are capable of traveling while loaded, the degrees of freedom are based both on the manipulating capability and mobility. In most robotic situations, DOF is based either on the manipulating capability or mobility.

It can be seen that there are numerous considerations that make path planning for construction manipulators unique. Thus, the models used should reflect the objectives and constraints of the construction environment. Further investigations are required to determine specific genetic operators and techniques which can generate solutions efficiently for construction environment. The following section proposes a conceptual approach to model these objectives and constraints using Genetic Algorithms.

4.0 Conceptual Model

Between the two alternate spaces to search for a suitable path - configuration space (C-Space) and real space, the C-Space approach is appropriate for manipulators or mechanisms having redundant degrees of freedom. If there are no redundancies are present searching the real space is more appropriate.

Configuration space is an imaginary space having the same number of dimensions as degrees of freedom of the manipulator. [Lozano-Perez 1981]. Each point in C-Space represents a unique configuration of the manipulator.
In the case of construction manipulators, each configuration of the manipulator can be assigned a desirability level based on the feasibility and level of overall safety. Configurations of the loaded manipulator which are infeasible due to obstruction or overloading, are indicated as obstacle space.
The generation of such a configuration space can not be done directly with the basic techniques such as obstacle growth. The use of interference detection and spatial analysis routines for the generation of such a space was found to be feasible [Reddy 1998].

Once the C-Space is generated with the appropriate weighs for each configuration, GA can be applied to search through the space. The points in configuration space can be converted to binary form and represented as a string of appropriate length to represent the points along the path. Ideally, the string length should be variable to accommodate variations in path length. The fitness function of the genetic algorithm should capture the objectives and constraints governing the situation. The function would be based on distance traveled along each DOF, configuration costs at each point and changes in direction.

An alternate string representation could model the change along each dimension of the C-Space. In such a case, the string would be binary in its natural form. However, there is no guarantee that the end of the string will be at the place location. Thus the proximity of the end to the place location also has to be factored into the fitness function to ensure that the solutions converge toward a workable one.

Modified versions of the conventional genetic operators such as reproduction, crossover and mutation will have to be developed. In addition, customized operators and techniques which can improve the performance, have to be investigated.
To illustrate the basic steps, the following section describes an implementation of a SGA based path planner for a standard path-planning problem.

5.0 Test Implementation

A simple implementation of SGA based path planning was done as an initial step towards the investigations. This involved determining the path of a 2 DOF manipulator in 2D space as shown in Figure-1. The configuration space generated for this problem and the pick and place locations are shown in Figure-2. In Figure-1 (real space), obstacle is represented by the shaded area.

The SGA was implemented to search through this configuration space for a suitable path between the pick and place configurations. The SGA model developed for this problem, the parameters used and results obtained are presented in this section.

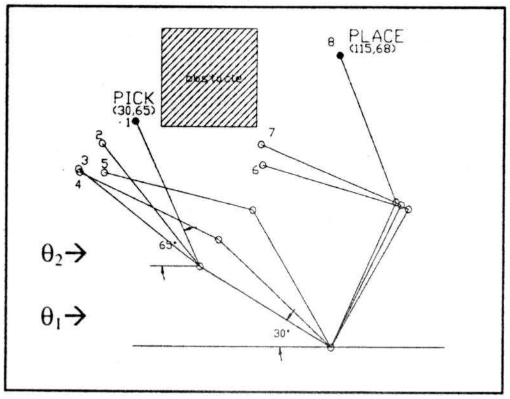

Figure-1: Two D.O.F. Planar Manipulator in Real Space

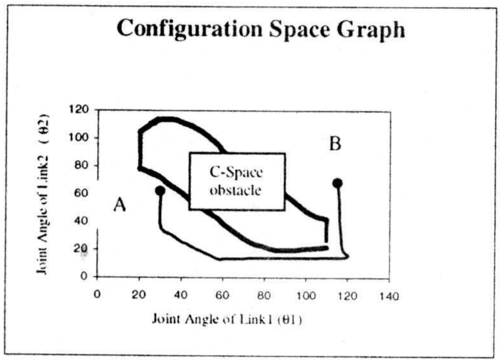

Figure-2: Configuration Space Representation
Pick Location: A (30,65)
Place Location: B (115,68)
Line joining A and B represents the path.

5.1 Path Representation as a String

Each string in SGA refers to an unique path. In this model, each path is expressed as a connection of eight points and the length of the string is kept constant in all populations. Thus, excluding the pick and place locations, there are six intermediate points in each string such as:

[PICK, $(\theta_1, \theta_2)_1$, $(\theta_1, \theta_2)_2$ .... $(\theta_1, \theta_2)_5$,$(\theta_1, \theta_2)_6$, PLACE]

Since each point represents two values of the variables, there are totally 12 variables in each string. The number of bits for each variable are fixed and is based on the limits of the search space. In this problem, the search space is limited to $0 \leq \theta_1 \leq 130$ and $0 \leq \theta_2 \leq 130$ (in degrees). Thus, a minimum of seven bits are required for each variable. However, for extendibility eight bits are allocated. Hence, the total number of bits in a string are 96 as shown in Figure-3.

5.2 Fitness Function

The fitness function for this example is dependant on the distance traveled and the obstacles encountered. It can be expressed as follows:

$f(x) = d(x)\,[1+C]$ where d(x) is distance function and C is obstacle violation coefficient. The above function is inversely proportional to the fitness value of a path. To make the algorithm maximize the fitness value in such cases (usually in case of minimization problems), it should be subtracted from a large constant so that all fitness values are non negative and individuals get fitness values according to their merit [Goldberg 1993]. The large constant is calculated as the sum of maximum and minimum fitness in a particular population. The final fitness function takes the following form.

$$F_i = [\,f(x)_{min} + f(x)_{max}] - f_i(x)$$

Distance Function

The distance function value [d(x)]is proportional the distance traveled by the manipulator from pick location to place location. The decoded values of the path variables in a string sample are given below to illustrate the calculation of fitness function.

(30,65), (36,52), (39,51), (48,43),
(100,67), (101,71), (110,70), (115,68)

Distance between the successive points in the path is determined by calculating the Euclidean distance and the final value of the function is equal to the sum of the distances. The value of distance (in relative units) is found to be 10.54.

Violation Coefficient

The violation coefficient is modeled based on number of points on the path that lie in the obstacle region. A path is said to be infeasible in any one of the following cases.

1. Key points (8 in this case)that represent the path , lie in the obstacle region.
2. A portion of the line that connects the successive key points crosses C-Space obstacle.

Fixed number of via points (4 in this case) are generated at regular intervals between successive key points to identify the infeasible path due to case-2. Number of via points to be generated depends on the degree of accuracy with which the crossing should be eliminated. Both key points as well as via points are taken into account to calculate the violation coefficient. The algorithm for

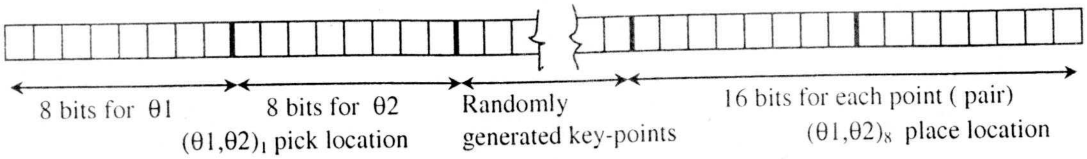
8 bits for θ1
8 bits for θ2
$(\theta1,\theta2)_1$ pick location
Randomly
generated key-points
16 bits for each point (pair)
$(\theta1,\theta2)_8$ place location

Figure-3: String description and bits allocation for path variables in each Chromosome

calculating violation coefficient for each string is as follows.

```
start
count = 0;
for  i = 1 to n  //n is total number key points and via points (36 in this case)
    check if $(\theta_1 , \theta_2)_i$ lies in obstacle region.  if true count = count +1;
continue
C = count*m / n; // m is a constant to magnify the effect of violation.
stop
```

In the given string sample, none of the key points lie in the obstacle region. But the line joining the points (48,63) and (100,67) crosses C-Space obstacle. Number of via points that lie in the obstacle region are 3. The violation coefficient [C] is calculated as follows:

count=3 and C= 3*15 / 36 = 1.25.

The value violation coefficient will vary from 0(if no point fall in the obstacle region) to 1 (if all points fall in the obstacle region), provided "m" is unity.

5.3 Genetic Operators

Reproduction operator

This operator is used to check the eligibility of a string (path) to be in the mating pool. A solution with high fitness will get more number of copies in the mating pool; where as a solution of low fitness may or may not have copies in mating pool. Since the number of individuals in the population and the mating pool are same, weak solutions will not be permitted to enter into mating pool. The fitness level required to enter the mating pool is calculated based on the ration of string fitness to average fitness of the population.

Crossover operator

The standard two-point crossover is implemented for the test case. In this type, two cross-sites are randomly selected and the sub string in between the two cross-sites will be interchanged to create new offspring. Probability of cross over represents the percentage of matches that will undergo cross over operations. Generally, a value in between 0.6 and 1.0 is adopted.

Mutation operator

Mutation refers to alteration of character values (0 into 1 and 1 into 0) in individual string with a given probability. The prime objective of mutation is to keep diversity in population. However very high values such as 0.01 to 0.1 probability of mutation make the search more random and cause high disturbance. Tests were run with different values mutation probability.

Convergence Criteria

In an ideal situation, all the individuals in the population converge to the same level of fitness. But in practice, it is very difficult to achieve this. Normally the iteration is stopped after specified number of generations or attaining same maximum fitness for certain percentage of population.

Usually long runs are required for attaining specified percentage of row convergence. For this problem, both the number of generations as well as the column convergence criteria have been adopted. In column convergence criteria, each column is checked whether all cells in that contain either 1 or 0 and the total number of such columns is noted. If the percentage of this number to the total number of columns in the population matrix is more than the column convergence parameter, iteration is stopped. For example, column convergence parameter of 80% refers the iteration will be stopped after 77 (i.e. 0.80* 96 (i.e. string length in bits)) columns individually contain either one or zero in all cells.

5.4 Results

The SGA was run with different values of population sizes. Figure-4 shows a plot between the best values of distance and number of generations. Type of crossover and column convergence parameter are kept same for all runs. It can be seen that the distance decreases with increasing generations. One of the best path is given below.

(30,65), (29,49), (30,37), (42,25),
(59,13), (120,16), (117,22), (115,68)

Figure-1 shows the step by step change in the configuration as the manipulator moves from pick to place location for the above path. The same path is shown in Figure-2 also(C-Space).The distance of the above path in relative units is 17.9.

5.5 Observations

In conducting this test, the following observations were made:

1. High values of probability of mutation (for example 0.01 or 0.05) causes more diversity in the population and restricts convergence. A value less than or equal to 0.001 is found to be suitable for this problem.
2. One point cross over is found to be not effective for this problem due to the lack of efficient genetic information transfer. Good results have been obtained by two point crossover.
3. This simple model could not find good solutions to more complex path planning situations for the given test case. Thus there is a need to adopt more sophisticated representations and operators in such investigations.
4. The results from the genetic search should match the resolution of the configuration space. If the path specified is of higher resolution than the space, the manipulator will not be able to adhere to it.

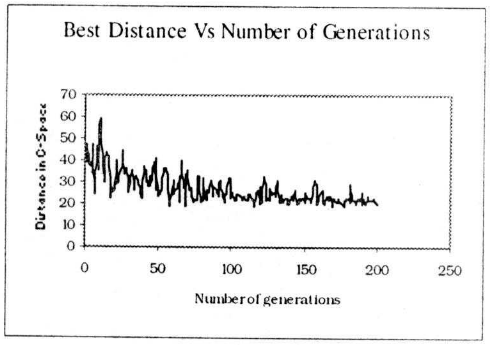

Figure-4: Generation History Model

Details of Run:

p(mut): 0.001 p(cross): 0.9

pop size: 110 col converg: 0.7

Cross over type: Two point

6.0 Conclusions

Based on the work done to date the following conclusions have been made:

1. The techniques developed for robotic path planning are applicable to construction manipulators. However, there are specific requirements and constraints not common in robotics, which have to be modeled.
2. Some of the requirements and constraints in the construction domain make the path planning problem too complex for solution by conventional techniques.
3. The potential of new techniques, such as Genetic Algorithms needs to be investigated.
4. The SGA has to be modified with appropriate representation schemes and operators to be effective for path planning problems. The performance of these schemes and operators are domain dependent. Thus investigations need to be conducted to develop the schemes and operators which are efficient for the construction domain.

Future work in this area will focus on (i) understanding the mathematical background for evolving new schemes and operators; (ii) developing schemes and operators for construction manipulators (iii) investigating the performance of these techniques.

Acknowledgements

This work was supported by the Department of Science and Technology, vide Grant No. III. 5(110)/97-ET.

References

1. Alciatore, D. (1989) *Path-Planning Algorithms for Pipe Manipulator* Dissertation, University of Texas at Austin.
2. Davidor, Y. (1991) *Genetic Algorithms and Robotics: A Heuristic study for optimization*. World Scientific Publishing Company.
3. Goldberg, D.E. (1989) *Genetic Algorithms in search, optimization and Machine Learning*, Addison-Wesley Publishing company.
4. Hsieh, Ting-Ya and Hass Carl, T. (1993) Applications of large scale manipulators in the construction environment, *Proceedings of the tenth International Symposium on Automation and Robotics in Construction (ISARC)*, Houston, Texas, USA, 55-62.
5. Latombe, J.C. (1991) *Robot Motion Planning*, Kluwer Academic Publishers, Boston, London.
6. Mark AC Gill and Albert Zomaya (1998) *Obstacle avoidance in multirobot systems: Experiments in parallel Genetic Algorithms*, World Scientific Publishing Company.
7. Morad , A. and Beliveau , J. (1992) *Path finder AI based path planning system*, ASCE Journal of Computing in Civil Engineering,6,114-128.
8. Paredis, J. Westra R. (1997) *Co-evolutionary computation for path planning*, Proceedings EUFIT, 1, 394-398.
9. Raghunatha reddy, H. (1997) *Automated path planning for crane lifts*, M.S. Thesis, Department of Civil Engineering, Indian Institute of Technology Madras.
10. Rana, A.S. and Zalzala, A.M.S. (1996) *Near time optimal collision free motion planning of robotic manipulators using Evolutionary Algorithms*, Robotica, 14, 621-632.
11. Schwartz, J.T. and Sharir, M (1988) *A survey of motion planning and related geometric algorithms*, Artificial Intelligence, 37, 157-169.