Monograph:

Chen, Mingwu and Zalzala, A.M.S. (1995) A Genetic Approach to the Motion Planning of Redundant Mobile Manipulator Systems Considering Safety and Configuration. Research Report. ACSE Research Report 559 . Department of Automatic Control and Systems Engineering

RERVM COGNOSCERE CAVSAS

Department of AUTOMATIC CONTROL and SYSTEMS ENGINEERING
UNIVERSITY of SHEFFIELD

A GENETIC APPROACH TO THE MOTION PLANNING OF REDUNDANT MOBILE MANIPULATOR SYSTEMS CONSIDERING SAFETY AND CONFIGURATION

Mingwu CHEN and A.M.S. ZALZALA

Robotics Research Group
Department of Automatic Control and Systems Engineering
The University of Sheffield
P.O. Box 600, Mappin Street, Sheffield S1 4DU, United Kingdom

Research Report #559
17 January 1995

A Genetic Approach to the Motion Planning of Redundant Mobile Manipulator Systems Considering Safety and Configuration

Mingwu Chen and A.M.S. Zalzala

Robotics Research Group
Dept. of Automatic Control and Systems Engineering
University of Sheffield
Sheffield S1 4DU, UK
email: rrg@sheffield.ac.uk

Abstract

This paper presents a genetic algorithm approach to multi-criteria motion planning of a mobile manipulator system considering position and configuration optimisation. Travelling distance and path safety are considered in planning the motion of the mobile system. A wave front expansion algorithm is used to build the numerical potential fields for both the goal and obstacles by representing the workspace as a grid. The unsafeness of a grid point is defined as the numerical potential produced by obstacles. For multi-criteria position and configuration optimisation, obstacle avoidance, least torque norm, manipulability and torque distribution are considered. The emphasis is put on using genetic algorithms to search for global optimum and solve the minimax problem for torque distribution.

Various simulation results from two examples show that the proposed genetic algorithm approach performs better than conventional methods.

1. Introduction

A mobile manipulator system is a manipulator mounted on a mobile vehicle. Mobile manipulator systems are increasingly considered for applications in hazardous or hostile environments where human access is limited. They are also attracting significant interest in the manufacturing, military, and public service communities.

A typical characteristics of a mobile manipulator system is its high degree of kinematic redundancy created by the addition of the vehicle's degrees of freedom to the manipulator's. The redundancy, quite desirable for dexterous manipulation and transport functions in cluttered environments, allows the system to be optimally positioned and configurated for maximum performance when stringent task requirements, and/or system, environment and task constraints are given.

Moreover, a mobile manipulator is capable of performing a number of tasks in widely separated locations. Thus, in addition to using the redundancy to achieve optimal manipulator configuration for each task, one can also use it to solve for the global optimisation problem of finding optimal base trajectories for performing a sequence of tasks.

In planning the motion of mobile robots, travelling distance has been the primary object to be optimised because the shortest distance path may reduce the robot's travelling time and consequently the computational complexity of path planning [1-4]. However, another factor which should not be ignored during robot path planning is robot safety during path execution. Robot safety becomes important, when there are non negligible uncertainties in both the robot dynamics during path execution and the environmental information such as obstacles. The safety of a mobile manipulator becomes particularly important because when the arm of a mobile manipulator has to reach out while its vehicle is moving in a cluttered environment, the vehicle needs a larger clearance from obstacles than a mobile robot alone.

The safety of a robot path can be quantified by the clearance between the path and obstacles. If robot safety is the only concern, one would choose a path providing the maximum clearance from obstacles. However, such a path could be considerably longer than the shortest one. Therefore, it is not desirable to consider the safety criterion only when the robot's travelling distance is also important in path planning.

The safety of a path has not been considered explicitly in most known path planning approaches. In many papers [1,2], path safety was obtained by enlarging each obstacle by a specified amount. Though the method of growing obstacles is simple and attractive in many cases, a potential problem with this method is that some good paths could have been eliminated as a result of growing obstacles. Moreover, it may be very difficult to determine the degree of enlargement of obstacles during path planning because of its independence on the utilisation of the workspace as well as the uncertainties in the robot dynamics during path execution.

Potential field methods can generate safe paths. However, because no global search is undertaken, the solution path may be neither the shortest nor the safest. It is simply a negotiable path from the start configuration to the goal configuration, and no robust mechanisms have been developed for handling local minima. Thus, using potential field methods to elegantly solve the findpath problem is difficult.

Barraquand and Latombe [6] construct a numeric potential field using a grid representation for path planning. Their result is a numeric Voronoi diagram of the environment. Also, because it maximise clearance from obstacles, it suffers from a "too far" problem. Another drawback is that Barraquand and Latombe do not specifically consider clearance information to generate their numeric potential and therefore their method can guide the robot through narrow free space channels that are close to the goal, thereby endangering the robot. Suh and Shin [5] presented a variational dynamic programming approach to robot path planning with a distance-safety criterion. The method represented free workspace as channels and the safety cost of a path is defined as the deviation of the path from the centre-line path.

There often exist a large, even infinite, number of paths between the initial position and final position and path planning is not necessarily to determine the best solution but to obtain a good one according to certain requirements. Various search methods have been developed (e.g. calculus based methods, enumerative schemes, random search algorithms, etc.) for path planning. Calculus based methods are local in scope and easy to get stuck in local minima. Enumerative schemes are not effective when the search space is too large to explore all the parts. Random search algorithms are probabilistically complete, but may take a long time to find a solution. Genetic algorithms are robust search and optimisation methods. They search for the optimum globally and therefore they can avoid being trapped in local minimum. Moreover, it is easy to combine new requirements into GAs' cost functions. Many results [9-14] have shown that genetic-based algorithms performed better than traditional optimisation methods. Cleghorn et al [9] proposed a genetic algorithm to solve the shortest travelling distance problem while Leung and Zalzala [14] presented a genetic solution for the motion of wheeled robotic systems in dynamic environments. These two papers shown their genetic solutions to be a less computation intensive approach to robot path planning, but they only considered travelling distance and did not take path safety into consideration.

The particular kinematic redundancy of the mobile base to those of the manipulator of a mobile manipulator system are often utilised to optimise a wide variety of criteria and/or to meet various constraints. Carriker et al [20] used a simulated annealing method to optimise the docking positions of a mobile manipulator when it is travelling among a number of stations. Zhao, Ansari and Hou [21] solved a similar problem by using a genetic algorithm. But these two papers didn't take optimal configuration, obstacle avoidance and least torque norm into consideration. Pin and Culioli [18,19] discussed the multi-criteria position and configuration optimisation problem of a mobile manipulator during task commutation when it is required to perform a sequence of tasks. Optimisation criteria include obstacle avoidance, least torque norm, manipulability and joint actuator torque distribution. Because of the competition among various criteria, the multi-criteria optimisation problem typically exhibits many local minima. Moreover, there exist a minimax problem when using the redundancy solution of a mobile manipulator to optimise the torque distribution. Pin and Culioli [18] used a Newton algorithm to solve the multi-criteria optimisation problem, but the algorithm needed additional methods to handle poor local minima. They [19] used a projected subgradient algorithm to solve the minimax problem for joint torque distribution, but with a long run-time (reaching 15 minutes on the Macintosh II), and the result obtained was a local minimum.

In this paper we represent a cluttered environment as a grid by cell decomposition. Two numerical potential fields are built for obstacles and the goal point by using a wave front expansion algorithm. Each grid point has an unsafeness value from the obstacles and a minimum distance value from the goal point. The unsafeness value of a node is defined as the numerical potential from the obstacles. The cost of a path is defined as the sum of the travelling distance and the average unsafeness of all the points in the path. A genetic algorithm approach to multi-criteria motion planning and position and configuration optimisation of a mobile manipulator system is developed. Both travelling distance and path safety are considered in planning the motion of the mobile base. For multi-criteria position and configuration optimisation, obstacle avoidance, least torque norm, manipulability and torque distribution optimisation are considered. The emphasis is put on using genetic algorithms to search for global optimum and solve the minimax problem for torque distribution.

The remainder of the paper is organised as follows. In section 2 we begin by describing the numerical potential fields, then we propose the cost function for motion planning of mobile robots with a distance-safety criterion and the genetic algorithm approach. In Section 3 we introduce the criteria for position and configuration optimisation of a mobile manipulator and present the genetic algorithm approach to this problem. Then, in Section 4 we present various numerical simulation results from two examples. The results of this work are summarised in Section 5.

2. A genetic-based approach to robot motion planning with a distance-safety criterion

2.1 Two Numerical Potential Fields

Although it is difficult to construct an analytical potential field over a free space of arbitrary geometry, the computation of a numerical potential field over a work space in the form of a grid turns out to be much easier. Here, we develop the wave front expansion algorithm originally proposed by Barraquand and Latombe [6] for computing numerical potential fields. The algorithm is efficient when the dimension of the work space is small, i.e. m=2 and 3, and their time complexity is independent of the geometry of the free space.

First, a fine grid is thrown in the work space (see Fig. 2.), where "*" nodes represent regions of obstacles. Paths are constructed as moves between adjacent "+" nodes either laterally or diagonally. It is not permissible for the robot to pass between the corners of obstacle regions.

Given a node q in a m-dimensional grid, its p-neighbours ($1 \leq p \leq m$) are defined as all the nodes in the grid having at most p co-ordinates differing from those of q, the amount of the difference being exactly one increment in absolute value. There are $2m$ 1-neighbours, $2m^2$ 2-neighbours, and so on. Here we consider that two nodes in a grid are neighbours if and only if they are p-neighbours for a predefined $p \in [1, m]$.

The numerical potential field for the goal is constructed as follows: First, the potential value U is set to 0 at q_{goal}. Next, it is set to d_1>0 at every 1-neighbour node of q_{goal} : to d_2>0 at every 1-neighbour of these new nodes (if it has not been computed yet); etc. The algorithm terminates when all the nodes in the free space accessible from q_{goal} have been fully explored. Here $d_1 < d_2 < d_3 \cdots$ can be any positive real numbers. Choosing a set of d_i's properly can help improving the genetic-based search process. Fig. 1a. shows a numerical potential field for the goal point.

The numerical potential field for obstacles (including the natural boundaries) over the free space of a grid is built similarly. First, the boundary is constructed, then, the potential values of U in these boundary nodes are set to o_0. Next, the values are set to o_1>0 at every 1-neighbour node of these boundary nodes; to o_2>0 at every 1-neighbour of these new nodes (if it has not been computed yet); etc. The algorithm terminates when all the nodes in the free space have been fully explored. Here, $o_0 \geq o_1 \geq o_2 \cdots$ can also be chosen as any real numbers. Properly choosing them can help searching for safe paths. Fig. 1b. shows a numerical potential field for obstacles.

29	27	25	23	21	19	17	15	13	11	9	7	5	3	5	7
27	25	23	21	19	17	15	13	11	9	7	5	3	1	3	5
25	23	21	19	17	15	13	11	9	7	5	3	1	0	1	3
27	25	23	*	*	17	15	13	11	9	7	5	3	1	3	5
29	27	*	*	*	*	17	15	*	*	*	*	*	3	5	7
31	29	*	*	*	*	19	17	19	*	*	*	7	5	7	9
33	31	33	*	*	23	21	19	21	19	*	11	9	7	9	11
35	33	31	29	27	25	*	*	19	17	15	13	11	9	11	13
37	35	33	31	29	27	*	*	21	19	17	15	13	11	13	15
39	37	*	*	*	29	*	*	23	21	*	*	*	13	15	17
41	39	*	*	*	31	29	27	25	*	*	*	17	15	17	19
43	41	43	*	*	33	31	29	*	*	*	21	19	17	19	21
45	43	45	*	*	35	33	31	29	27	25	23	21	19	21	23
47	45	43	41	39	37	35	33	31	29	27	25	23	21	23	25
49	47	45	43	41	39	37	35	33	31	29	27	25	23	25	27
51	49	47	45	43	41	39	37	35	33	31	29	27	25	27	29

(a) $d_1 = 1,\ d_2 = 3,\ d_3 = 5$

9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	7	7	7	7	7	7	7	7	7	7	7	7	7	7	9
9	7	7	9	9	7	5	5	7	7	7	7	7	5	7	9
9	7	9	10	10	9	7	7	9	9	9	9	9	7	7	9
9	9	10	*	*	10	9	9	10	10	10	10	10	9	7	9
9	9	10	*	*	10	9	7	9	10	*	10	9	7	7	9
9	7	9	10	10	9	9	9	7	9	10	9	7	5	7	9
9	7	7	9	9	9	10	10	9	7	9	7	7	5	7	9
9	7	9	9	9	9	10	10	9	7	9	9	9	7	7	9
9	9	10	10	10	9	10	10	9	9	10	10	10	9	7	9
9	9	10	*	10	9	9	9	9	10	*	10	9	7	7	9
9	7	9	10	10	9	7	9	10	10	10	9	7	5	7	9
9	7	9	10	10	9	7	7	9	9	9	7	5	5	7	9
9	7	7	9	9	7	5	5	7	7	7	5	5	5	7	9
9	7	7	7	7	7	7	7	7	7	7	7	7	7	7	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

(b) $o_0 = 10,\ o_1 = 9,\ o_2 = 7$

Fig. 1. Two numerical potential fields

Now every node in the free space is assigned two numerical potentials. One is the numerical potential from the goal which represents the minimum distance value from the goal, and the other is the numerical potential from the boundary of the closest distance node, which is defined as the unsafeness value of the node. The farther away a node is from the boundary of the closest obstacle node, the lower the unsafeness value of a node is and the safer it is for a robot to move through it.

2.2 The Cost Function

Let a path P be described by a set of neighbouring nodes connecting the starting node and the goal node, denoted by $\{ n_i, i = 1,2,...N \}$.

Considering both distance and safety, the cost function is represented by

$$C(P) = \varepsilon\ D(P) + \lambda\ S(P) \qquad \varepsilon, \lambda \geq 0 \tag{1}$$

where $D(P)$ and $S(P)$ represent costs associated with length and unsafeness of path P, respectively, and ε and λ are the relative weightings between the two.

The distance cost of a path is defined as the sum of all the segments connecting the starting point and the final point.

$$D(P) = \sum_{i=2}^{N} \| n_i - n_{i-1} \| \tag{2}$$

where $\|\cdot\|$ represents the Euclidean norm.

The unsafeness of a path is defined as the average unsafeness of all the nodes in the path.

$$S(P) = \frac{1}{N} \sum_{i=1}^{N} S_i \quad , \qquad S_i \leq S_m \tag{3}$$

where S_i is the unsafeness of the node n_i and S_m is the given upper limit of the unsafeness, which is introduced to prevent a path from getting too close to obstacles. By adjusting the relative weightings ε and λ different paths can be achieved.

2.3 A Genetic Algorithm Approach

Genetic Algorithms(GAs) are improvement search algorithms based on many processes of natural selection in biological systems. They utilise a survival-of-the-fittest concept among string structures, through reproduction, crossover and mutation operations. GAs have following advantages over traditional optimisation and search methods [15]:

-- They are well adapted to search for solutions in highly dimensional search space.

-- They are very tolerant to the form of the function to be optimised, for instance, these functions do not need to be either differentiable or continuous.

-- They can be easily implemented on a massively parallel machine and they can achieve super-linear speed up with the number of processors [16,17].

To solve the mobile robot path planning problem by a genetic algorithm, we need a coding scheme to encode the parameters of the problem into genetic strings. Here, robot path is coded as a string of N nodes represented by their Cartesian co-ordinates as

$$\{ (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \}$$

with all the values stored in a decimal form. This code method yields variable-length paths and a proper genetic structure is required to deal with it, in particular while performing crossover.

A set of valid random paths are generated as the initial generation. In order to prevent the robot wandering endlessly inside the work space, a weighted vector of motion direction is employed according to the minimum distances from the goal of the 8 neighbouring nodes. A neighbouring node which has a lower distance value to the goal, has more chance to be selected as the next node in the path. A fitness value is assigned to each string according to its travelling distance and safety. In our algorithm, the fitness function is defined by

$$f = C_m - C(P) \tag{4}$$

where C_m is a properly selected positive real number not less than the maximum cost of $C(P)$.

A reproduction approach is applied to select strings for the next generation. Genetic algorithms use the fitness value of each string of the current generation to decide if and how many copies of the string should be passed to the next generation. The larger the fitness value of one string, i.e. the lower the cost of the path, the higher probability of the string being chosen for the next generation.

When in early generations there is a tendency for a few superstrings to dominate the selection process. Later on when the population is largely converged, competition among population members is less strong and the simulation tends to wander. In these two cases, fitness values must be scaled to prevent take-over of the population by a few superstrings in the early generations and to accentuate differences between population members to continue to reward the best performers. In this paper we use linear scaling to calculate the scaled fitness f' from the raw fitness f using a linear equation of the form

$$f' = af + b \tag{5}$$

In this equation, the coefficients a and b are chosen to do two things: enforce equality of the raw and scaled average fitness values and cause the maximum scaled fitness to be two times of the average fitness. These two conditions ensure that the average strings receive one offspring copy on average and the best receive two on average. When the scaled fitness value of a string becomes negative, we simply set it to zero. To reduce the stochastic error associated with the selection, we implement the stochastic remainder sampling without replacement.

Performing crossover is not straight forward because of the variable-length coding and, more important, since a random crossover would produce a discontinuous path. Thus the selected path pair is checked for nodes with a certain proximity (coincident, one or two nodes apart). If such a pair is found and there is not coincident node for both paths, a random segment is generated to connect both nodes, and exchange the remainders of the both paths. If the path pair have coincident points, then select one randomly as the crossover site and exchange the remainder of the two paths. To perform mutation, select two nodes randomly along a path, destroy the old path between them and generate a new path to connect them.

3. Using genetic algorithms for multi-criteria position and configuration optimisation for redundant mobile manipulator systems

3.1 Problem Formulation

The kinematic relations for a mobile manipulator system are given by

$$X_t = X_p + X_{m/p}(\Phi) \qquad (6)$$

where X_t is the task vector representing the position and orientation of the end-effector in the absolute reference frame, X_p is the vehicle vector representing the position and orientation of the vehicle in the absolute reference frame. $X_{m/p}(\Phi)$ represents the vector of the end-effector position and orientation with respect to the vehicle reference frame and Φ is the vector of joint positions of the manipulator. The components X_{p_i}, i =1, 2, .. , n and θ_j, j = 1, 2, ... , m, of the vector X_p and Φ, thus represent the system variables generating a space, S, of dimension typically greater than 6. It can be assumed in general that the components of the joint positions of the manipulator are constrained by independent upper and lower bounds specified by the vectors Φ_l, Φ_u . Thus, the constraints can be described by

$$\Phi_l \leq \Phi \leq \Phi_u \qquad (7)$$

where the vector inequalities are applied component wise.

Assuming that a point load, corresponding to a force F, is applied at the end-effector, the contributed manipulator actuator torque on the static system can be calculated as:

$$\tau = J(\Phi)^T F \qquad (8)$$

where is the manipulator Jacobian matrix. There are independent bounds on the components of the torque vector given by

$$\tau_l \leq \tau \leq \tau_u \qquad (9)$$

where τ_l and τ_u are the lower and upper bounds of the torque τ , respectively.

The position and configuration optimisation can be formulated as a local optimisation in S of a given criterion E subject to a set of constraints.

Various optimisation schemes corresponding to different modes of motion have been proposed by Pin and Culioli [18].

Minimum load-induced actuator torque minimises

$$E_1 = \|\tau\|^2 \qquad (10)$$

Minimisation of E_1 in cases where load forces are very high can lead to very uneven distributions of the joint actuator torques, sometimes exceeding the limit of an actuator while the others remain relatively low. A minimax criteria is thus introduced as

$$E_2 = Max_i|\alpha_i \tau_i|, \quad i = 1, 2, \ldots, m \tag{11}$$

where the coefficients α_i weigh the individual torque by the inverse of their relative limit value. Utilisation of this criterion leads to much more evenly distributed optimal actuator loads, but it needs a much more complex mathematical treatment because the corresponding Lagrangian is not differentiable and the optimality conditions need to be replaced by variational inequalities.

For obstacle avoidance schemes, the following criterion is utilised:

$$E_{obs_{ij}} = \frac{1}{\|X_i - X_{oj}\|^2} \tag{12}$$

where X_i and X_{oj} represent the Cartesian position vector of any given point on the mobile manipulator and on an obstacle, respectively.

To provide commutation configurations avoiding as much as possible the singular configurations of the manipulator, a manipulability criterion is utilised:

$$E_{man} = \frac{1}{Det(J(\Phi)J(\Phi)^T)} \tag{13}$$

To forecast the optimum commutation configurations of the system when several requirements constraints exist on the upcoming task, a general criterion which is to be minimised is introduced as follows:

$$E = \sum_{ij} \gamma_{ij} E_{obs_{ij}} + \varepsilon E_1 + \lambda E_2 + \delta E_{man} \tag{14}$$

where the coefficients represent the relative importance given to the corresponding requirements. Here, the criterion is subject to the following constraints:

$$X_t = X \;\; + X_{m/p}(\Phi)$$

$$\tau = J(\Phi)^T F$$

and

$$\Phi \;\; \le \Phi \le \Phi_u$$

$$\tau \le \tau_i \le \tau_u$$

3.2 A Genetic Algorithm Approach

In order to optimise the mobile manipulator's position and configuration by a genetic algorithm, we need to choose a coding scheme to encode the parameters of the system into genetic strings. Here the manipulator configuration Φ is chosen to be encoded. Each element θ_i ($i = 1, 2, \ldots, j\text{-}1, j\text{+}1, \ldots, m$) of the manipulator configuration Φ, except one element θ_j, is coded into a binary string. θ_j can be expressed in terms of θ_i, for $i \neq j$, by the constraint of Eq.(6.2.1). The genetic string is formed by concatenating the codes of the element θ_i ($i = 1, 2, \ldots, j\text{-}1, j\text{+}1, \ldots, m$).

The fitness function is defined by

$$f = E_m - E \tag{15}$$

where E_m is any positive real number not less than the maximum value of E given by Eq.(14).

In order to improve the selection process, here we use linear scaling to calculate the scaled fitness f' from the raw fitness f using a linear equation of the form

$$f' = a f + b \tag{16}$$

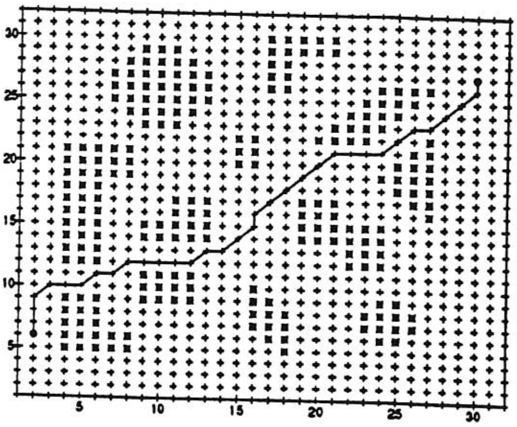
To reduce the stochastic error associated with the selection, we implement the stochastic remainder sampling without replacement.

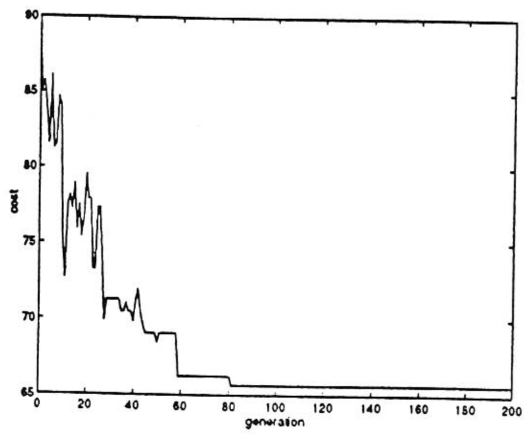
4. Computer Simulation Results

4.1 Motion planning of a point mobile robot in a cluttered workspace

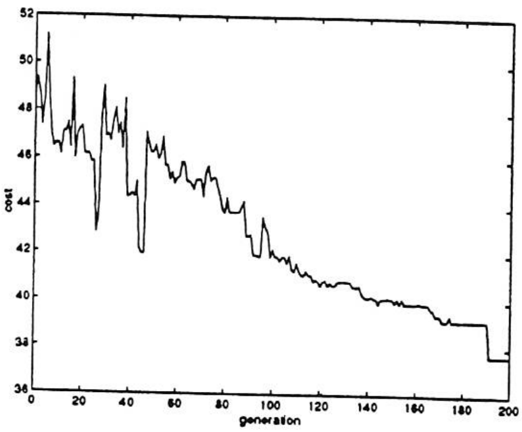
For this case study, a workspace with cluttered obstacles is constructed as shown in Fig. 2. The workspace is represented by a grid with 32×32 nodes. The starting point is (2,6), and the goal point is (30,27). Three genetic parameters n (population size), p_c (crossover probability), and p_m (mutation probability) are chosen as 100, 0.8 and 0.025 respectively. Simulations were conducted on a Sun Sparc station.

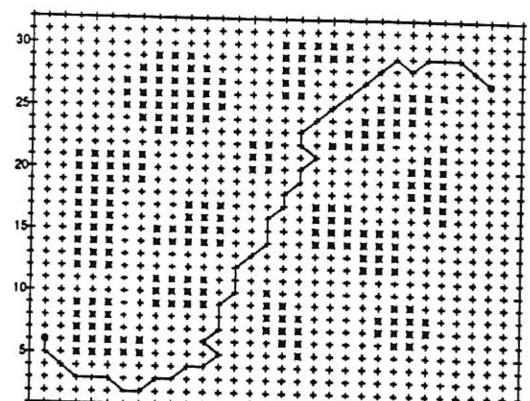
Fig. 2. shows path costs versus generation and the corresponding path obtained by the GA with a distance-safety criterion. When $\varepsilon=1$, $\lambda=0$, i.e. not considering safety, a nearly shortest path is obtained. When $\varepsilon=0.1$, $\lambda=10$, there is a compromise between distance and safety, a relative safe and short path is obtained. When λ is further increased to 20, i.e. safety is supposed to be a major concern, the algorithm generates a very safe path but it is much longer than the shortest one. In the figures, all the path costs decrease steadily and converge with generation.



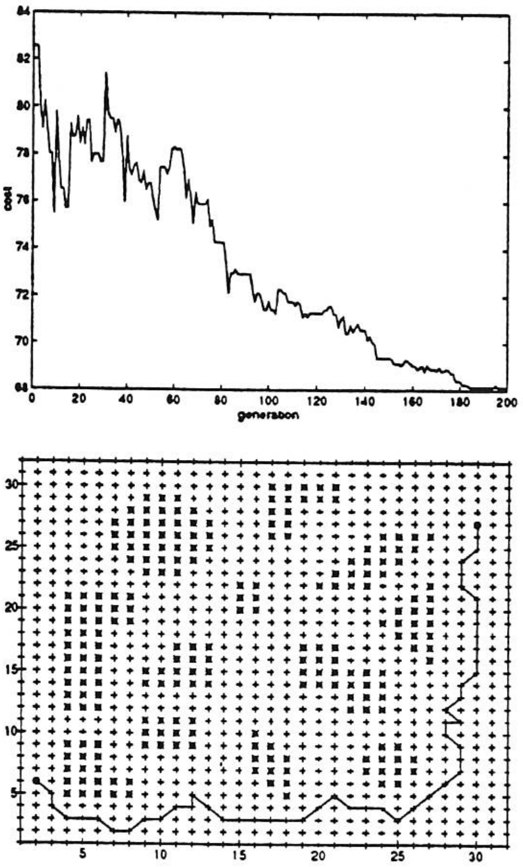


(a) $\varepsilon=1$ $\lambda=0$





(b) $\varepsilon=0.1$ $\lambda=10$



(c) ε =0.12 λ =20

Fig. 2. Path costs versus generation and the corresponding path obtained by the GA

4.2 Multi-criteria position and configuration optimisation for a system including a three-link manipulator mounted on a mobile platform

In this case study, we consider the system given by Pin and Culioli [11]. The system includes three links of length $l_1 = l_2 = 1$, $l_3 = .5$, with revolute joints, and a platform moving along the horizontal axis as shown in Fig. 3. In the figure, the platform is represented by a rectangle, and the three joint angles are θ_1, θ_2, and θ_3 .

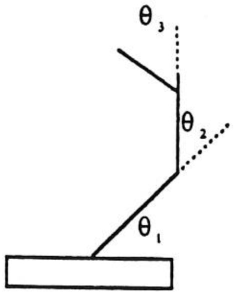


Fig. 3. A mobile manipulator system

The forward kinematic equations for the system are given by:

$$X_e = X_p + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (17)$$

$$Y_e = l_1 \sin(\theta)_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

(18)

where X_e, Y_e are the X, Y positions of the end-effector in the absolute reference frame and is the X position of the mobile platform in the absolute reference frame. The manipulator Jacobian matrix is thus:

$$J=\begin{bmatrix} -l_1S - l_2S_{12} - l_3S_{123} & -l_2S_{12} - l_3S_{123} & -l_3S_{123} \\ l_1C_1 + l_2C_{12} + l_3C_{123} & l_2C_{12} + l_3C_{123} & l_3C_{123} \end{bmatrix} \quad (19)$$

where $S_1 = \sin(\theta_1)$, $C_1 = \cos(\theta_1)$, $S_{12} = \sin(\theta_1 + \theta_2)$, $C_{12} = \cos(\theta_1 + \theta_2)$, $S_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$, $C_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$.

In each of the following examples, two degree of redundancy are provided to the system by imposing constraints on the end-effector position only as $(X_e, Y_e) = (5, 1.4)$. A force $F = (0, -5)$ is imposed at the end-effector.

A genetic string of 32 bits is used to encode the two parameters (θ_1 and θ_2) of the problem, each parameter being represented by 16 bits. Note also that having encoded θ_1 and θ_2, θ_3 and X_p can be obtained by the constraints defined by Eqs.(17) and (18). Three genetic parameters n (population size), p_c (crossover probability), and p_m (mutation probability) are chosen as 50, 0.6 and 0.02 respectively. The following are the simulation results for torque distribution optimisation, obstacle avoidance and torque optimsation, and manipulability and torque optimsation. All the simulations were conducted on a Sun Sparc station.

A. A minimax approach for optimisation of torque

Fig. 4. shows the least torque values versus generation and the corresponding commutation configuration obtained by the GA with a least torque norm requirement. The manipulator torque values are τ = {-0.61, 1.13, -0.3} with a torque norm value of $\|\tau\|^2$ = 1.73. The maximum actuator torque is at the second joint and ratio of maximum to minimum actuator torque is 3.73. Assuming the same limit for all actuators, we can see that if the load is increased, the load-induced torques would increase proportionally, eventually resulting in the limit of the second joint actuator being reached while the third joint would be providing only a third of its strength. For such cases, it is possible to include torque limits in the optimisation scheme using E_1, however, uneven utilisation of the actuator strength typically still results[19], and the system may fail in initiating a task for which a feasible solution exists.

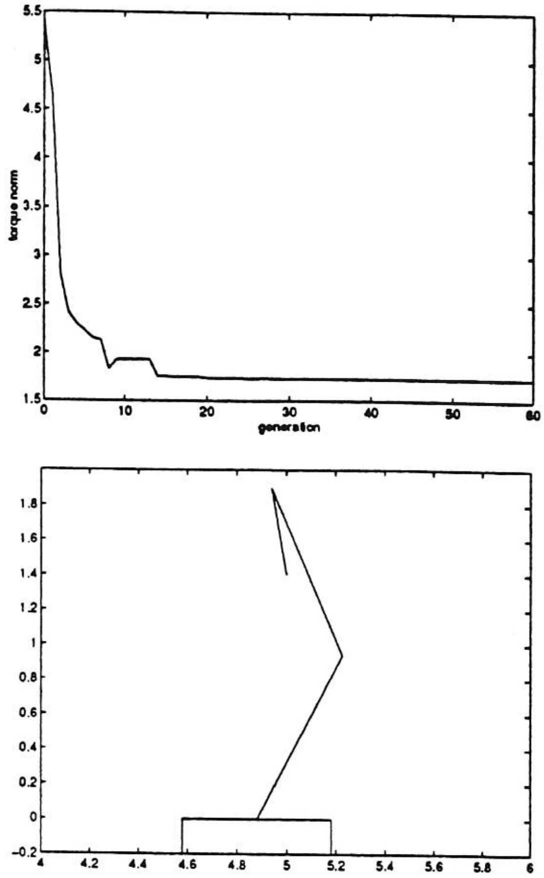


Fig. 4. The least torque values versus generation and the corresponding commutation configuration obtained by the GA with a least torque norm requirement

An alternative approach is to use another optimisation criterion $E_2 = Max_i|\alpha_i\tau_i|$, $i = 1, 2, \ldots, m$, where the coefficients α_i weigh the individual torques by the inverse of their relative limit value. For implementation purposes, minimisation of E_2 essentially consists in solving a minimax problem[19] that involves a much more complex mathematical treatment than for E_1. The Lagrangian in this type of problems is not differentiable and the solution of the optimisation problem calls specific numerical approaches. Pin and Culioli[19] utilised a projected subgradient algorithm, but with a long run-time (reaching 15 minutes on the Macintosh

II). Here, we use the genetic algorithm approach to solve the minimax problem. Fig. 5. shows fitness values versus generation and the corresponding commutation configuration obtained by the GA for the minimax problem. The resulting torque values are $\tau = \{-0.90, 0.90, -0.87\}$ with the joint angles $\theta_1 = 68.96$ and $\theta_2 = 41.74$ near the global optimum. Fig. 9a. shows the fitness values in the configuration space of joint angles θ_1 and θ_2 for this minimax problem. The run-time by the genetic algorithm approach is 2 seconds only which is much shorter than that by the projected subgradient algorithm.

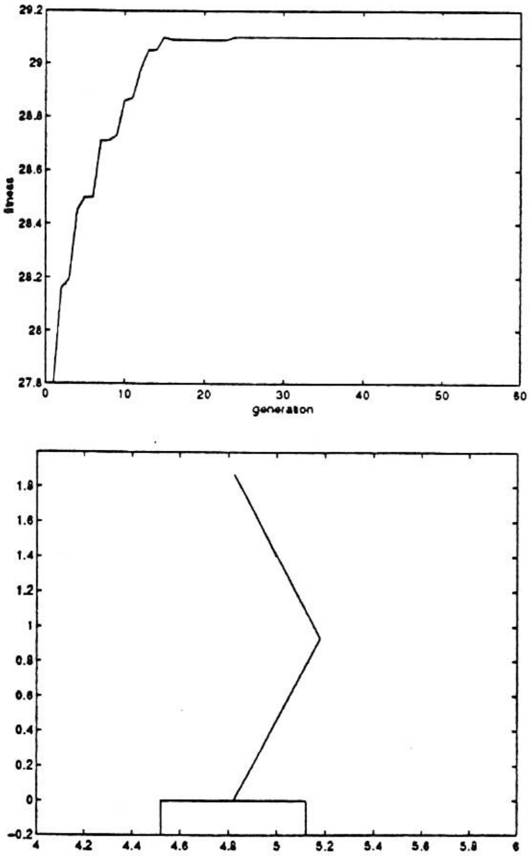


Fig. 5. Fitness values versus generation and the corresponding commutation configuration obtained by the GA for the minimax problem

B. Obstacle avoidance and torque minimisation

Fig. 6. shows fitness values versus generation and the corresponding commutation configuration obtained by the genetic algorithm when the global optimisation criterion involves a least torque norm requirement and avoidance by the midpoint of the second link of an obstacle located at $(X_o, Y_o) = (5.2, 1.0)$. Here, equal weigh is placed on the two requirements ($\gamma = \varepsilon = 1$ in Eq.(12)) and the resulting joint torque value is $\|\tau\|^2 = 2.08$.

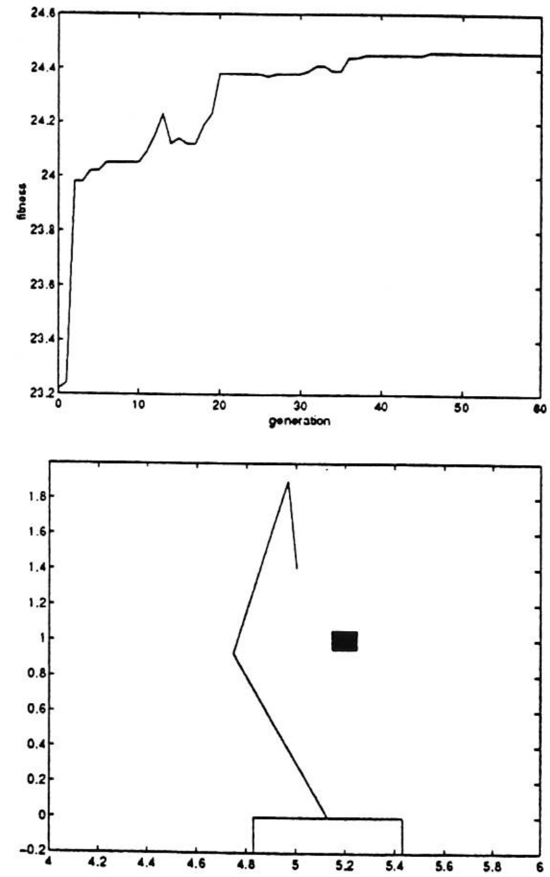


Fig. 6. Fitness values versus generation and the corresponding commutation configuration obtained by the GA with obstacle avoidance and least torque norm requirements

Fig. 7. presents another case with requirement similar to those of Fig. 6., but with the obstacle located on the

horizontal axis obstructing the position of the platform. The resulting joint torque value is $\|\tau\|^2 = 3.01$.

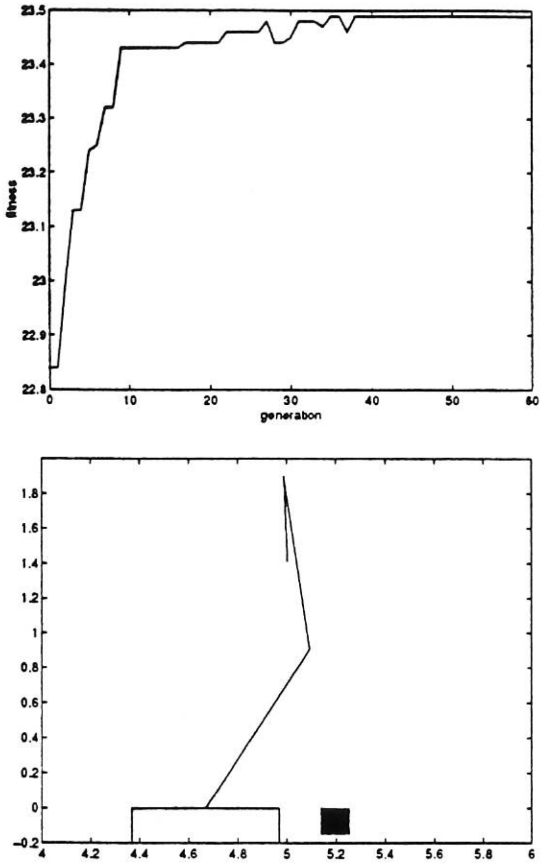


Fig. 7. Fitness values versus generation and the corresponding commutation configuration obtained by the GA with least torque norm and avoidance of an obstacle obstructing the platform

C. Manipulability and torque minimisation

Fig. 8. shows the results of the simulation when the global optimisation criterion involves equally weighted least torque norm and manipulability requirements. In this case, there is no obstacle in the workspace. This problem exhibits many local maxima due to competition between the minimum torque and maximum manipulability(see Fig. 9b.). The genetic algorithm converges to the global optimum very quickly (θ_1 = 114.24 and θ_2 = -36.93, in 2 seconds), while paper[19] got stuck by one of local minima.

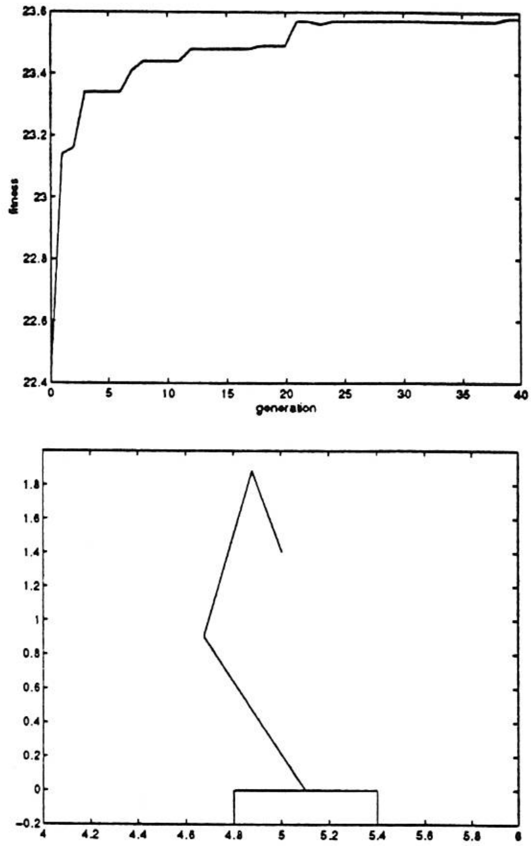
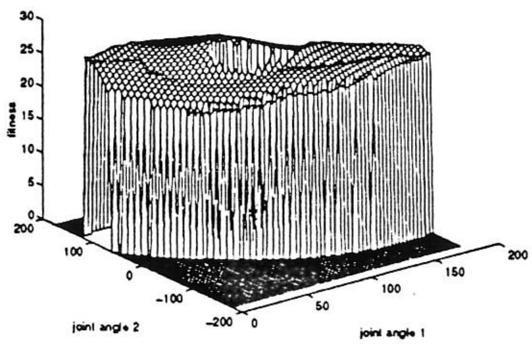
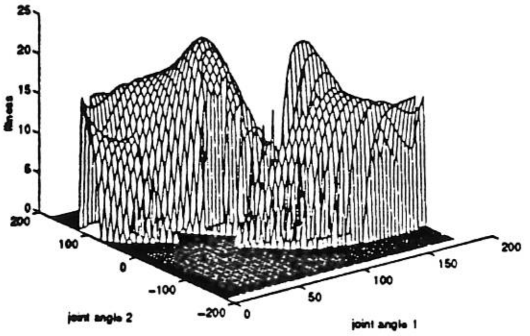


Fig. 8. Fitness values versus generation and the corresponding commutation configuration obtained by the GA with least torque norm and manipulability requirements



(a) with a minimax criterion for torque distribution



(b) with least torque norm and manipulability criteria

Fig. 9. Fitness values in the configuration space of joint angles θ_1 and θ_2

5. Conclusion

The integrated combination of mobility and manipulation is a promising area for robotics research, where the possibility of both a large workspace and high redundancy makes the combination highly attractive. The incorporation of the safety issue into motion planning of mobile manipulator systems becomes particularly important because mobile manipulator systems need larger clearance from obstacles and they are more likely to fall down. The multi-criteria motion planning and position and configuration optimisation typically exhibits many local minima. Traditional optimisation methods often cause difficulty in tackling this problem. Genetic algorithms are robust search and optimisation methods. They search for the optimum globally, and therefore they can avoid being trapped in local minimum. Various simulation results demonstrated the effectiveness and efficiency of the proposed genetic algorithm approach to reach near optimal solutions to different weighting multi-criteria optimisation problems. Although the algorithm is developed mainly for two dimensional problems, it can be extended to a class of three dimensional problems.

References

1. T. Lozano-Perez and M.A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles", Commun. ACM, Vol.22, No.10, pp.560-570, 1979.

2. K.G. Shin and N.D. Mckay, "Selection of near-minimum time geometric paths for robotic manipulators", IEEE Journal of Automation and Control, Vol.AC-31, No.6, pp.501-511, 1986.

3. E. G. Gilbert and D. W. Johnson, "Distance functions and their applications to robot path planning in the presence of obstacles", IEEE Journal of Robotics and Automation, RA-1, 1, 1985.

4. G. Wilfong, " Shortest path for autonomous vehicles", In the Proceedings of the IEEE International Conference on Robotics and Automation(Scottsdale, Ariz., May 14-19), IEEE, New York, pp.15-20, 1989.

5. Suk-Hwan Suh and K.G. Shin, " A variational dynamic programming approach to robot-path planning with a distance-safety criterion", IEEE Journal of Robotics and Automation, Vol.4, No.3, pp.334-349, 1988.

6. J. Barraquand and J. C. Latombe, "Robot motion planning: A distributed representation approach", The International Journal of Robotics Research, Vol.10, No.6, pp.628-649, 1991.

7. Y. K. Hwang and N. Ahuja, "Gross Motion Planning -- A Survey", ACM Computing Surveys, Vol.24, No.3, September 1992.

8. J. Barraquand and J. C. Latombe, "A Monte-Carlo algorithm for path planning with many degrees of freedom", In Proceedings of the IEEE International Conference on Robotics and Automation(Cincinnati, May 13-18), IEEE, New York, pp.1712-1717.

9. T. F. Cleghorn, P. T. Baffes and L. Wang, "Robot path planning using a genetic algorithm", Proc. 2nd Annual Workshop on Space Operations, Automation and Robotics, pp.383-390, 1988.

10. Y. Davidor, Genetic Algorithms and Robotics: A Heuristic Strategy for Optimisation, World Scientific, 1990.

11. A. R. Khoogar and J. K. Parker, "Obstacle avoidance of redundant manipulators using genetic algorithms", Proceedings of the IEEE International Conference on Robotics and Automation, pp.317-320, 1991.

12 J. M. Ahuactzin, E. G. Talbi, P. Bessiere, and E. Mazer, "Using genetic algorithms for robot motion planning", ECAI 92, The Proceedings of the 10th European Conference on Artificial Intelligence, pp.671-675, 1992.

13. K.K. Chan and A.M.S. Zalzala, "Genetic-based minimum-time trajectory planning of articulated manipulators with torque constraints", IEE Colloquium on Genetic Algorithms for Control and Systems Engineering, May 1993.

14. C.H. Leung and A.M.S. Zalzala, "A genetic solution for the control of wheeled robotic systems in dynamic environments", Proc. IEE International Conference on Control'94, 1994.

15. D.E. Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning, Addison Welsey, 1989.

16. T. C. Fogarty and R. Huang, "Implementing the genetic algorithm on transputer based parallel processing systems", Parallel Problem Solving from Nature, 1st Workshop, PPSN 1 Proceedings, pp.145-149, Vol.496.

17. P. Spiessens and B. Manderick, "A genetic algorithm for massively parallel computers", Parallel Processing in Neural systems and Computers, Elsevier Science Publishers B. V.(North-Holland), 1990.

18. F. G. Pin and J.C. Culioli, "Multi-Criteria position and configuration optimisation for redundant platform/manipulator systems", IEEE International Workshop on Intelligent Robots and Systems IROS'90, pp.103-107, 1990.

19. F. G. Pin and J.C. Culioli, "Optimal positioning of combined mobile platform-manipulator systems for material handling tasks", Journal of Intelligent and Robotic Systems: Theory and Applications, Vol.6, Part 2, pp.165-182, 1992.

20. W.F. Carriker, P.K. Khosla, and B.H. Krogh, "Path planning for mobile manipulators for multiple task execution", IEEE Transactions on Robotics and Automation, Vol.7, No.3, pp.403-408, June 1991.

21. M. Zhao, N.Ansari, and E.S.H. Hou, "Mobile manipulator path planning by a genetic algorithm", Journal of Robotic Systems, Vol.11(3), 143-153, 1994.