



This is a repository copy of *Optimisation of Cubic Polynomial Robot Motion Using Flexible Polyhedron Search, Weighted-Sum-GA and Pareto-Based GA: A Comparative Study*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/81095/>

Monograph:

Zalzala, A.M.S. and Ang, M.C. (1997) *Optimisation of Cubic Polynomial Robot Motion Using Flexible Polyhedron Search, Weighted-Sum-GA and Pareto-Based GA: A Comparative Study*. Research Report. ACSE Research Report 670 . Department of Automatic Control and Systems Engineering

Reuse

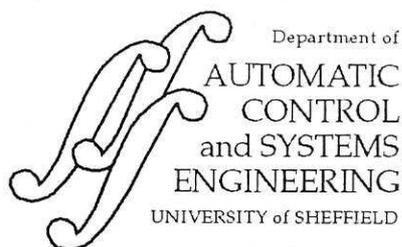
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>



**OPTIMISATION OF CUBIC POLYNOMIAL ROBOT MOTION
USING FLEXIBLE POLYHEDRON SEARCH, WEIGHTED-SUM GA
AND PARETO-BASED GA: A COMPARATIVE STUDY**

AMS Zalzala and MC Ang

*Robotics Research Group
Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield S1 3JD, United Kingdom*

Research Report #670
February 1997

 Tel : +44 (0)114 2225250
Fax : +44 (0)114 2731729
EMail : rrg@sheffield.ac.uk
Robotics Research Group

200391381



OPTIMISATION OF CUBIC POLYNOMIAL ROBOT MOTION USING FLEXIBLE POLYHEDRON SEARCH, WEIGHTED-SUM GA AND PARETO-BASED GA: A COMPARATIVE STUDY

AMS Zalzala and MC Ang

Robotics Research Group

Department of Automatic Control and Systems Engineering,

The University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom

Email: rrg@sheffield.ac.uk

1 SUMMARY

This project was concerned with the motion optimisation of a SCARA robot subject to kinematics constraints, where two methods are used, namely: genetic-based algorithms and the flexible polyhedron search. Both methods use cubic spline functions to generate motion profiles. In formulating the GAs, tailor-fit operators and procedures were used to seek an application-dependent structure. In addition, an initial evaluation is reported in the form of comparisons between a Pareto-based and weighted-sum (parametric) approaches to multi-criteria optimisation. Case study results are summarised for the RTX robot with six joints.

2 INTRODUCTION

The application of GAs in multi-objective optimisation is a major interest in GA research. Current multi-objective GA approaches include the classical aggregation of the different objectives into a single function, population-based non-Pareto approaches and the most recent work on the ranking schemes based on the definition of Pareto optimality¹ [1]. Pareto-based ranking was first proposed by Goldberg [2], as a mean to assign equal probability of reproduction to all Pareto optimum² individuals. However this is not the only technique required for the Pareto-based multiobjective GAs. There are additional niche formation method that must be included to prevent the population converge to one peak, a phenomenon know as genetic drift, as described below.

A genetic optimisation for the PUMA 560 manipulator for various different optimisation criteria has been reported [3], including minimum time, constraints on torque and end-point velocities is presented. In studies with two 3-DOF PUMA like multi-arm robots [4], an evolutionary algorithm has been proposed for three different objectives, namely, the minimum path length, minimum uneven distribution of via-points and collision avoidance. The planning for an RTX robot [4] used the weighted-sum approach when considering different objectives, e.g. minimum travel time, collision avoidance, and compliance with the displacement constraints. The study made use of customised operators to compensate the problem of large position jumps at the crossover sites.

The flexible polyhedron search³ is one of the unconstrained non-linear optimisation techniques [5]. It is used for the minimisation of a function of n variables and depends on the comparison of function values at the $n+1$ vertices, followed by the replacement of the vertex with the highest value

¹ The name is derived from the Italian economist Pareto who introduced it in 1896. It is an important concept in particular in the competitive equilibrium in economics and also optimal mixed strategies in statistical decision theory [7].

² Also called efficient point, non-dominated or admission solution.

³ Originally the simplex method by Nelder and Mead (1964) when they introduced the method, but the new name was later used by Himmelblau [5] as being more descriptive.

by another point. It has been chosen by researchers since it does not assume the function to be differentiable and continuous over the range of interest, and has the additional advantage that the user can be sure of the design variables taking only positive values. This method has been adopted [6] to optimise the best (minimum) combination of time intervals subject to constraints on joint velocities, accelerations and jerks for PUMA type robot with six joints.

3 MOTION PLANNING ESSENTIALS

3.1 Motion based on Cubic Splines

The objective here is to construct joint trajectories that fit a number of joint displacements at a sequence of time instants by using cubic polynomial functions. Consider a vector of via-points for the j th joint along some initial path as $[\theta_{j1}(t_1), \theta_{j2}(t_2), \dots, \theta_{jn}(t_n)]$, where $t_1 < t_2 < t_3 < t_4 < \dots < t_{n-2} < t_{n-1} < t_n$ is an ordered time sequence, indicating that the position of the j th joint at time $t = t_i$ is $\theta_{ji}(t_i)$. Let v_{ji} and w_{ji} denote the velocity and acceleration of joint j at knot i . At the initial time $t = t_1$ and the terminal time $t = t_n$, the joint displacement, joint velocity, and joint acceleration are $\theta_{j1}, v_{j1}, w_{j1}$ and $\theta_{jn}, v_{jn}, w_{jn}$ respectively. In addition, joint positions θ_{jk} at $t=t_k$ for $k = 3, 4, \dots, n-2$ are also specified. However, θ_{j2} and $\theta_{j,n-1}$ are the two extra knots required to provide the freedom for solving the cubic polynomials. For simplicity, the subscript j for j th joint is dropped since the result is the same for all the joints. From [6], the acceleration vector w of knots is obtained as follows:

$$A w = b \quad (1)$$

where: $w = [w_2, w_3, w_4, \dots, w_{n-3}, w_{n-2}, w_{n-1}]^T$,

$$A = \begin{bmatrix} 3h_1 + 2h_2 + \frac{h_1^2}{h_2} & h_2 & \dots & 0 & 0 \\ h_2 - \frac{h_1^2}{h_2} & 2(h_2 + h_3) & \dots & 0 & 0 \\ 0 & h_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2(h_{n-3} + h_{n-2}) & h_{n-2} - \frac{h_{n-1}^2}{h_{n-2}} \\ 0 & 0 & \dots & h_{n-2} & 3h_{n-1} - 2h_{n-2} + \frac{h_{n-1}^2}{h_{n-2}} \end{bmatrix}, \quad b = \begin{bmatrix} 6\left(\frac{\theta_3}{h_2} + \frac{\theta_1}{h_1}\right) - 6\left(\frac{1}{h_1} + \frac{1}{h_2}\right)\left(\theta_1 + h_1 v_1 + \frac{h_1^2}{3} w_1\right) - h_1 w_1 \\ \frac{6}{h_2}\left(\theta_1 + h_1 v_1 + \frac{h_1^2}{3} w_1\right) + \frac{6\theta_4}{h_3} - 6\left(\frac{1}{h_2} + \frac{1}{h_3}\right)\theta_3 \\ 6\left(\frac{\theta_5 - \theta_4}{h_4} - \frac{\theta_4 - \theta_3}{h_3}\right) \\ \dots \\ \frac{6}{h_{n-2}}\left(\theta_n - v_n h_{n-1} + \frac{h_{n-1}^2}{3} w_n\right) - 6\left(\frac{1}{h_{n-2}} + \frac{1}{h_{n-3}}\right)\theta_{n-2} + \frac{6}{h_{n-3}}\theta_{n-3} \\ -6\left(\frac{1}{h_{n-1}} + \frac{1}{h_{n-2}}\right)\left(\theta_n - v_n h_{n-1} + \frac{h_{n-1}^2}{3} w_n\right) + \frac{6\theta_n}{h_{n-1}} + \frac{6\theta_{n-2}}{h_{n-2}} - h_{n-1} w_n \end{bmatrix}$$

and

$$h_i = t_{i+1} - t_i \quad (i = 1, 2, \dots, n-1), \quad (2)$$

$$w_i(t) = \frac{t_{i+1} - t}{h_i} w_i + \frac{t - t_i}{h_i} w_{i+1}, \quad (3)$$

$$v_i(t) = -\frac{w_i}{2h_i}(t_{i+1} - t)^2 + \frac{w_{i+1}}{2h_i}(t - t_i)^2 + \left(\frac{\theta_{i+1}}{h_i} - \frac{h_i w_{i+1}}{6}\right) - \left(\frac{\theta_i}{h_i} - \frac{h_i w_i}{6}\right), \quad (4)$$

$$\theta_i(t) = \frac{w_i}{6h_i}(t_{i+1} - t)^3 + \frac{w_{i+1}}{6h_i}(t - t_i)^3 + \left(\frac{\theta_{i+1}}{h_i} - \frac{h_i w_{i+1}}{6}\right)(t - t_i) + \left(\frac{\theta_i}{h_i} - \frac{h_i w_i}{6}\right)(t_{i+1} - t), \quad (5)$$

where $i = 1, \dots, n-1$. The positions, velocities and accelerations can therefore be obtained provided that each time interval h_i is known.

The optimal time solution for the time interval vector X should be obtained with the corresponding knot velocities and accelerations computed from the above equations, and the jerks are the rate of

change for the corresponding accelerations. The resulting via-point velocities, accelerations and jerks may or may not violate their limits and should be compared with their own limits to obtain the time-optimal path satisfying the joint constraints.

3.2 Physical limits

Note that there are six joints that must be considered simultaneously and there are three constraints, i.e. velocity, acceleration, and jerk limits for each joint. For convenience, let

- VC_j = velocity constraint for joint j
- WC_j = acceleration constraint for joint j
- JC_j = jerk constraint for joint j
- $Q'_{ji}(t)$ = velocity for joint j between knot i and $i+1$.
- $Q''_{ji}(t)$ = acceleration for joint j between knot i and $i+1$.
- $Q'''_{ji}(t)$ = jerk for joint j between knot i and $i+1$.
- X = $(h_1, h_2, \dots, h_{n-1})$, the vector of time intervals.
- $Q_{ji}(t)$ = piece wise cubic polynomial trajectory for joint j between knot i and $i+1$, and
- w_{ji} = acceleration of joint j at knot i .

The acceleration in (3) can therefore be rewritten as

$$Q''_{ji}(t) = \frac{t_{i+1} - t}{h_i} w_{ji} + \frac{t - t_i}{h_i} w_{j,i+1} \quad (6)$$

and also the velocity function in (4) can be rewritten as

$$Q'_{ji}(t) = -\frac{w_{ji}}{2h_i} (t_{i+1} - t)^2 + \frac{w_{j,i+1}}{2h_i} (t - t_i)^2 + \left(\frac{\theta_{j,i+1}}{h_i} - \frac{h_i w_{j,i+1}}{6} \right) - \left(\frac{\theta_{ji}}{h_i} - \frac{h_i w_{ji}}{6} \right) \quad (7)$$

Following from [6], the objective thus is to minimise $\sum_{i=1}^{n-1} h_i$ subject to constraints

$$|Q'_{ji}(t)| \leq VC_j, \quad |Q''_{ji}(t)| \leq WC_j, \quad \text{and} \quad |Q'''_{ji}(t)| \leq JC_j, \quad (8)$$

for $j = 1, 2, \dots, N$ and $i = 1, 2, \dots, n-1$.

(a) Velocity constraints

The maximum absolute value of velocity exists at t_i , t_{i+1} or \bar{t}_i , where \bar{t}_i satisfies $Q''_{ji}(\bar{t}_i) = 0$ and is in the interval $[t_i, t_{i+1}]$. The velocity constraints then become

$$\max_{t \in [t_i, t_{i+1}]} |Q'_{ji}(t)| = \max \left\{ |Q'_{ji}(t_i)|, |Q'_{ji}(t_{i+1})|, |Q'_{ji}(\bar{t}_i)| \right\} \leq VC_j$$

for $i = 1, 2, \dots, n-1; j = 1, 2, \dots, N$, where

$$|Q'_{ji}(t_i)| = \left| -\frac{w_{ji}}{2} h_i + \frac{(\theta_{j,i+1} - \theta_{ji})}{h_i} + \frac{(w_{ji} - w_{j,i+1})}{6} h_i \right|$$

$$|Q'_{ji}(t_{i+1})| = \left| \frac{w_{j,i+1}}{2} h_i + \frac{(\theta_{j,i+1} - \theta_{ji})}{h_i} + \frac{(w_{ji} - w_{j,i+1})}{6} h_i \right|$$

$$|Q'_{ji}(\bar{t}_i)| = \begin{cases} \left| \frac{w_{ji} w_{j,i+1} h_i}{2(w_{ji} - w_{j,i+1})} + \frac{(\theta_{j,i+1} - \theta_{ji})}{h_i} + \frac{(w_{ji} - w_{j,i+1})}{6} h_i \right| & \text{if } w_{ji} \neq w_{j,i+1} \text{ and } \bar{t}_i \in [t_i, t_{i+1}] \\ 0 & \text{if } w_{ji} = w_{j,i+1} \text{ or } \bar{t}_i \notin [t_i, t_{i+1}] \end{cases}$$

(b) Acceleration constraints

The acceleration is a linear function of time between two adjacent knots. Hence, the maximum absolute value of acceleration exists at either t_i or t_{i+1} and therefore equals to

$$\max\{|w_{j1}|, |w_{j2}|, \dots, |w_{jn}|\} \leq WC_j, \text{ for } j = 1, \dots, N.$$

(c) Jerk constraints

The jerk is the rate of change of acceleration. The constraints are represented by

$$\left| \left(\frac{w_{j,i+1} - w_{ji}}{h_i} \right) \right| \leq JC_j, \text{ for } j = 1, 2, \dots, N \text{ and } i = 1, 2, \dots, n-1.$$

3.3 The feasible solution converter (time scaling)

If the correct time intervals are guessed, then w_{ji} can be uniquely determined from equation (1). However, if constraints on joint velocities, accelerations, and jerks are not satisfied, then time intervals $[h_1, h_2, \dots, h_{n-1}]$ should be expanded to bring the unsatisfied velocities, accelerations, and jerks to their constrained values.

For example, let $Q_i(t)$ be the original spline function of the joint displacement defined on the time interval $[t_i, t_{i+1}] = [t_i, t_i + h_i]$. If all the time intervals are expanded according to $\tau = \lambda t$ so that the new intervals are $h_i^* = \lambda h_i$, then the new acceleration w_i^* can be determined as w_i/λ^2 . Thus, the new spline function $Q_i^*(\tau)$ defined on the interval $[\lambda t_i, \lambda(t_i + h_i)] = [\tau_i, \tau_i + h_i^*]$ can be shown to have the same form as $Q_i(t)$. Therefore, the first, second, and third time derivatives of $Q_i^*(\tau)$ will be $(1/\lambda)Q_i'(\tau/\lambda)$, $(1/\lambda^2)Q_i''(\tau/\lambda)$, and $(1/\lambda^3)Q_i'''(\tau/\lambda)$, respectively.

Now let

$$\lambda_1 = \max_j \left[\max_{t \in [t_i, t_{i+1}]} |Q_{ji}'(t)| / VC_j \right] \tag{9}$$

$$\begin{aligned} \lambda_2 &= \max_j \left[\max_{t \in [t_i, t_{i+1}]} |Q_{ji}''(t)| / WC_j \right] \\ &= \max_j \left[\max_i |w_{ji}| / WC_j \right] \end{aligned} \tag{10}$$

$$\begin{aligned} \lambda_3 &= \max_j \left[\max_{t \in [t_i, t_{i+1}]} |Q_{ji}'''(t)| / JC_j \right] \\ &= \max_j \left[\max_i \left| \frac{w_{j,i+1} - w_{ji}}{h_i} \right| / JC_j \right] \end{aligned} \tag{11}$$

$$\text{with } \lambda = \max(1, \lambda_1, \sqrt[2]{\lambda_2}, \sqrt[3]{\lambda_3}). \tag{12}$$

If the time interval h_i is replaced by λh_i for $i=1, 2, \dots, n-1$, then the velocity, acceleration and jerk will be replaced by factors of $1/\lambda$, $1/\lambda^2$, $1/\lambda^3$, respectively. These changes assure the satisfaction of constraints on velocities, accelerations and jerks.

4 THE GENETIC FORMULATIONS

The formulation of the multi-objective GAs are described in some details in the following sections. The general procedure is depicted in Figure (1).

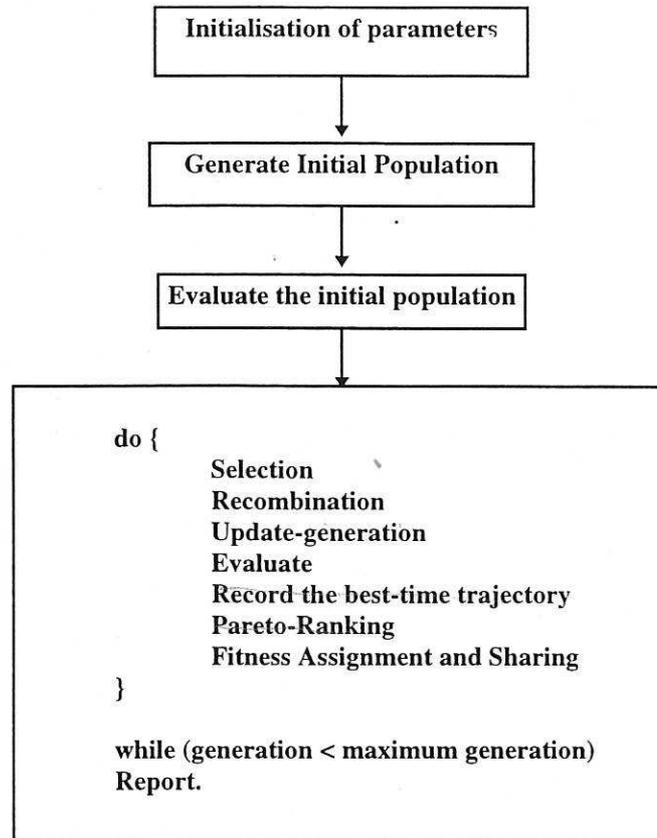


Figure (1): Outline of the Genetic Procedure

4.1 The objective functions

4.1.1 Pareto-Based GA

The objective vector is set as:

$$\text{Minimise } \left\{ \sum_{i=1}^{n-1} h_i, 1-\lambda_1, 1-\lambda_2 \right\}$$

subjects to constraint: $\lambda_3 \leq 1$.

where h_i is the i th time interval, $1-\lambda_1$ is the criticality to the velocity constraints, $1-\lambda_2$ is the criticality to acceleration constraints. λ_1 , λ_2 and λ_3 are computed by equations (9-11) respectively. The criticality is a measure of how close is a trajectory to the joints' velocity and acceleration limits.

4.1.2 Weighted-Sum GA

Two types of weighting methods is considered in this case study, namely a normalised objective function and a non-normalised function.

(a) The formulation of objective function in **normalised** form can be written as to

$$\text{Minimise } \{ w_1 f_1 + w_2 f_2 + w_3 f_3 \}$$

subject to jerk constraints, i.e. $\lambda_3 \leq 1$, where $f_2 = 1 - \lambda_1$ and $f_3 = 1 - \lambda_2$. The weight coefficients w_i operate on the i th objective function and can be interpreted as "the relative worth" of that objective when compared to the other objectives. The weights are normalised so that the total is equal to 1, and the two objective values for f_2 and f_3 are in the range of $[0,1]$.

The objective function f_1 is calculated as a relative value to the range of motion time $[MIN, MAX]$. Assume the upper bound of travel time MAX is the maximum trajectory motion time obtained from the initial population, and the lower bound of the travel time MIN is the maximum time among the six joints when they travel at their velocity limits between start and end points, i.e. θ_{j1} and θ_{jn} . Thus,

$$MIN = \max_j \left(\frac{\theta_{jn} - \theta_{j1}}{VC_j} \right)$$

where $j = 1, \dots, N$ represent the six joints and VC_j is the velocity constraint imposed on joint j . Note that the lower bound of the motion time calculation has not taken into account the zero velocity and acceleration values imposed at the start and end positions. Hence, the function f_1 is set as:

$$f_1 = \frac{\sum_{i=1}^{n-1} h_i - MIN}{MAX - MIN}$$

(b) Some researchers have not adopted the concept of normalising the objectives when using weighting method in their studies [3,4], and the objective functions are arbitrary weighted with different values. To compare with the normalised formulation, the **non-normalised** objective function is formulated as to

$$\text{Minimise } \left\{ \omega_1 \sum_{i=1}^{n-1} h_i + \omega_2 f_2 + \omega_3 f_3 \right\}$$

subject to jerk constraints, where ω_i is the weight of objective f_i .

4.2 Parameter initialisation

The definitions of the used genetic parameters are listed in Table (1), and these are referred to in the following sections.

Table (1): The GA Parameters

Parameter	Description
maxgen	maximum generation
maxinterval/(n-1)	maximum time interval
maxjoint	maximum joint number
maxknot/n	maximum knot number
maxpop	maximum population size
SEED	constant integer to initialise the random number generator
sp	selective pressure
pcross	crossover probability
pinject	injection probability
pmutate	BGA (Breeder Genetic Algorithm) mutation probability
pmutate_time	time mutation probability
reserve_num	number of best trajectory to be kept, it is set to 1,(when perform BGA with truncation scheme, it is also the number of selected parents)
cross_over_dis_meter	maximum distance to perform cross-over for joint 1 (Zed).
cross_over_dis_radian	maximum distance to perform cross-over for joints 2-6.
v1	velocity at the start position
a1	acceleration at the start position
vn	velocity at the end position
an	acceleration at the end position
sizeC	the number of constraints
inc	time step for generating motion profiles

4.3 Evaluating the population

The population performance is evaluated with each trajectory in the population receiving a fitness value prior to the selection process.

4.3.1 Ranking

All trajectories are ranked based on its total travelling time and the criticality to the joints' velocity and acceleration limits based upon Pareto ranking.

Using the Pareto optimality definition, a point x is said to be non-dominated if it is not dominated by any other point. Thus, an individual can be ranked by counting the number of individuals that dominate it [1]. Hence, the non-dominated ones that are the best performers and will be assigned with highest rank, i.e. zero. When all individuals in the population are ranked, the fitness values will be assigned to them according to their rank. This can be done by interpolating some linear or exponential function from the best individual (rank $\hat{=} 0$) to the worst individual (rank $< N$). Following that, same rank individuals will receive the same fitness values by averaging the total values assigned to them.

Therefore, fitness values are assigned according to an individual's rank in the population, thus ensuring that the population will strive for all the three objectives: smallest travel time and achieving the two limits allowable.

4.3.2 Fitness Assignment

The fitness of trajectories are obtained on the basis of their relative fitness in the population rather than their raw performances. The trajectories are first sorted into a descending order based on their Pareto rank. Then by interpolating between the best rank individual to the worst rank individual, each trajectory fitness value can be calculated as follows:

$$F(x_i) = 2 - sp + 2(sp - 1) \frac{x_i - 1}{Maxpop - 1},$$

where x_i is the position in the ordered population of trajectory i [8]. This rank-based fitness assignment provides only a small bias towards the most fit trajectories so that no trajectory will generate an excessive number of offspring and thus prevent premature convergence (Whitley, 1989). The 'sp', i.e. selective pressure, defines the maximum number of offspring that the best trajectory can reproduce.

4.3.3 Sharing Scheme

Fitness sharing uses a sharing parameter to control the extend of sharing, or in other words, it is a measure of the maximum distance between individuals that could form niches [1]. The trajectory fitness will be increased or decreased depending on the trajectory similarity with each other. After calculating the share count, the new total fitness values in the population will be altered and the value is usually different with the total fitness before share count.

4.4 Selection Scheme

Selection scheme is a process to determine the number of trials a particular individual is chosen for reproduction. The selection technique adopted in this project is based on stochastic universal sampling (SUS) [9]. This method uses a single spin and N equally spaced pointers, where N is the number of population size. The actual selection process begins with generating a random number, say p , from the range of $[0, sum/N]$, of which sum is the total fitness values of the population. The N trajectories are then chosen by generating the N pointers spaced by sum/N . Hence, N trajectories whose fitness span the positions of the pointers will be selected. The selected trajectories are then shuffled randomly before recombination. Note, however, that the number to be selected for this particular problem is not equal to N and a modification has been introduced.

4.5 Shuffling

After the selection stage, the selected individuals' indexes are sorted because of the previous fitness assignment procedure. It is therefore necessary to perform a shuffling procedure before the recombination process. This can be done by using a set of randomly generated numbers. By using the sorting procedure to sort the random number into an ascending order, the initially sorted indexes will be randomised and the selected individuals will be rearranged according to the randomised indexes.

4.6 Recombination Mechanisms

The selected trajectories will be paired up for crossover or recombination subject to their mating distance and cross-over probability. Self-mating is not allowed in the program. **Three** different genetic operators are implemented from the Breeder GA [10]. Considering two parents $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$, then, the offspring $\mathbf{z} = (z_1, \dots, z_n)$ may be composed in the following ways.

- (a) Discrete recombination: $z_i = \{x_i\}$ or $\{y_i\}$, where x_i or y_i are chosen with probability 0.5.
- (b) Extended intermediate recombination: $z_i = x_i + \alpha_i(y_i - x_i)$, $i = 1, \dots, n$, where α_i is chosen randomly in the interval $[-0.25, 1.25]$.
- (c) Extended line recombination: $z_i = x_i + \alpha(y_i - x_i)$, $i = 1, \dots, n$, where α is chosen randomly in the interval $[-0.25, 1.25]$.

In addition, a **fourth** path redistribution/relaxation operator is used [4]. A robot trajectory consists of joint angles that may produce large position jump in the offspring strings after conventional crossover. Therefore, this operator is proposed where cubic splines are fitted to the offspring's via-points with each time interval set to unity. The path length is then computed as the Euclidean distance between the start and end via-points along the splines as:

$$\sum_{t=0}^{t=n-\Delta t} \sqrt{\left(\sum_{j=1}^m (\theta_{j,t+\Delta t} - \theta_{j,t})^2 \right)}$$

where $j = 1, \dots, m$ and Δt is chosen to be a small number, n is the number of via-points and m is the number of robot joints. Each joint knots are then 'redistributed' evenly over these splines at equal intervals. The paths are then 'relaxed' by moving each via-point with a small step towards the point that will bisect the line between its neighbouring points as:

$$\theta_{ji} = \theta_{ji} + \delta \left(\frac{\theta_{j,i-1} + \theta_{j,i+1}}{2} - \theta_{ji} \right)$$

where $i = 1, \dots, n$ and δ is a positive random number less than one.

4.7 Modified feasible solution converter

When a travel time is produced following the process of population generation or recombination, the time value is evaluated using the feasible solution procedure, as described in section 3.3. Then, the total time intervals are scaled such that the motion time of an individual trajectory is optimal and does not violate the kinematics constraints. From section 3.3, equation (12) is rewritten as:

$$\lambda = \max(\lambda_1, \sqrt[2]{\lambda_2}, \sqrt[3]{\lambda_3}).$$

Thus, if $\lambda > 1$ then h_i should be increased to λh_i to satisfy the limits of the velocities, accelerations and jerks. Also, If $0 < \lambda < 1$ then h_i can be contracted to λh_i to obtain the time-optimal joint paths.

The corresponding velocity, acceleration and jerk values are scaled by the factors of $1/\lambda$, $1/\lambda^2$, $1/\lambda^3$, respectively.

4.8 Time intervals mutation

The offspring's initial time is given from the parent and it will be passed through the modified version of feasible solution converter for $n-1$ times. Each pass, a time interval will be selected randomly and increased or decreased depending on the criticality of that interval. If the interval is critical, i.e. very close to the limit, the time will be increased (to slow down), otherwise the time will be decreased (to speed up). The decrement and increment values are selected randomly between ranges of $[0.75, 1]$ and $[1, 1.25]$ respectively. Only the smallest time intervals will be used as the offspring travelling time. If $n-1$ trials are not successful, the offspring will accept the parent trajectory travel time which has been scaled appropriately with the factor obtained the converter.

5 SIMULATION RESULTS

The algorithms have been tested for the RTX robot with 6-joint motion planning in the configuration space. The case study is listed in Table (2), with the model parameters and constraints listed in Table (3) and (4).

Table (2): Initial and final configurations

	Column (m)	Shoulder r (rad)	Elbow (rad)	Yaw (rad)	Pitch (rad)	Roll (rad)
Initial configuration	0.4	$-\pi/6$	$-\pi/3$	$-\pi/2$	0	$-\pi/4$
Final configuration	0.8	$\pi/6$	$\pi/3$	$\pi/2$	$-\pi/6$	$\pi/4$

Table (3): Link parameters

Joint i	θ_i	α_i	a_i	d_i	Lower Bound	Upper Bound
1	0°	0°	0	0	-61 mm	+881 mm
2	0°	0°	a_2	$-d_2$	-90°	$+90^\circ$
3	0°	0°	a_3	$-d_3$	-180°	$+151^\circ$
4	0°	0°	0	0	-110°	$+110^\circ$
5	0°	90°	0	0	-8°	$+94^\circ$
6	90°	90°	0	d_6	-132°	$+181^\circ$
gripper	-	-	-	-	0°	$+90^\circ$

Table (4): Velocity, acceleration and jerk constraints

	Zed ⁴	Shoulder r	Elbow	Yaw	Pitch	Roll
Velocity	0.1116	0.1654	1.2092	1.9715	1.3780	1.2412
Acceleration	1.7755	6.2018	14.081	31.055	28.063	26.180
Jerk	297.59	894.67	3718.9	3377.6	3933.1	4172.7

Note: The zed velocity, acceleration, and jerk are in m/s , m/s^2 , and m/s^3 , respectively. The other joint angle velocities, accelerations and jerks are in rad/s , rad/s^2 , and rad/s^3 , respectively.

⁴ zed is also known as column.

5.1 Case 1: Pareto-based GA

(a) Breeder Genetic Algorithm (BGA) operators

The three BGA recombination operators namely, discrete, extended intermediate and extended line recombination operators are experimented to determine the best among them. Crossover probability is set to (pcross = 0.9). The performance for BGA genetic operators are shown in Table 5.1. The results are obtained by using a population size of 200 and 100 as the maximum number of generations.

Table 5.1: Results From BGA Genetic Operators

	Minimum Time(sec)
Discrete Recombination	4.3381
Intermediate Recombination	4.1497
Line Recombination	4.0854

(b) Path redistribution-relaxation operator

The path redistribution-relaxation operator is used with a different population size, and 0.9 crossover probability. The injection rate adopted [4] is 10% of the population size while in this simulation, the injection rate is 2.5%. The results for 100 generations with different population size are shown in Table 5.2.

Table 5.2: Results For Path Redistribution-Relaxation Operator

Population Size	Minimum Time(sec)
100	3.9530
200	3.9335
300	3.9049

(c) Comparisons

From the above case studies, one can conclude that the extended line recombination operator is the best among the three BGA genetic operators and the path redistribution-relaxation operator performs better than the BGA recombination operators. A further simulation is carried out for 300 population size and 500 generations with crossover probability of 0.9 and injection of 8 new trajectories in each generation. The results are shown in Table 5.3, indicating improved optimisation with larger population and generation for both BGA and redistribution-relaxation operators. However the results also show that the redistribution-relaxation operator still produces the best motion time (smallest) with bigger population and generation.

Table 5.3 Path Redistribution-relaxation operator vs. BGA Recombination Operators

Path Redistribution-Relaxation	Line Recombination	Discrete Recombination	Intermediate Recombination
3.8743 sec	3.8894 sec	4.0144 sec	4.0404 sec

(d) Truncation selection

To complete the study of the BGA operators, truncation selection has been tested with the consideration of sharing but without mating restriction. Hence, T % best rank individuals are selected and mated randomly in each generation to produce the new population. The smallest time trajectory will remain in the population. The results shown in Table 5.4 are obtained with a 200 population size and 100 maximum generation. T % is chosen in the range of 10-50% [10].

Table 5.4 Performance of the BGA Operators Without Mating Restriction

Truncation Threshold Percentage($T\%$)	Discrete Recombination (sec)	Intermediate Recombination (sec)	Line Recombination (sec)
10%	4.0404	4.0884	4.0373
20%	4.4436	4.1031	4.0817
30%	4.2443	4.2164	4.0648
40%	4.2344	4.2160	4.1575
50%	4.7487	4.1698	4.1319

The results above show that by using truncation selection, the motion time is better than the ones without it if the threshold percentage is small for recombination (note Table 5.1). However the time obtained by redistribution-relaxation operator is still better than the one by truncation selection. Table 5.4 also shows again that extended line recombination is indeed the best BGA operator for the minimum time motion planning problem for non-mating environment.

5.2 Case 2: Pareto-GA vs. Flexible Polyhedron Search

The optimal paths reported in Table 5.3 are fed into the FPS program, and the outcome is given in Table 5.5. While the GA yielded better optimisation of time, the procedure did require longer computation time than the FPS.

Table 5.5 Pareto-based GA vs. Flexible Polyhedron Search

Operator Responsible for the Optimal Path	Minimum Time(sec) Obtained from Section 5.1.2	Results from Flexible Polyhedron Search Method(sec)
Path Redistribution-Relaxation	3.8743	4.4870
Line Recombination	3.8894	4.1253
Discrete Recombination	4.0144	4.1325
Intermediate Recombination	4.0404	4.1141

5.3 Case 3: Weighted-Sum GA

(a) Normalised formulation

Several combinations of weights have been tried and the results for 200 population size for 100 maximum generation are listed in the following Table 5.6. Only path redistribution-relaxation operator has been applied in the tests for weighted-sum GA.

From the results shown in Table 5.6, when compared to redistribution-relaxation operator of Pareto-based GA (refer Table 5.2) under same population and generation size, the optimum motion time produced by Pareto-based GA has a slightly more optimal (3.9335 seconds). Weighted-sum GA using the weight of 0.1, 0.0 and 0.9 managed to obtain the best motion time of 3.9531 seconds. However, this result actually considers only two objective functions, and can not be used to compare with the results obtained using Pareto-based GA. In this case, the best motion time for normalised weight is 3.9570 seconds, obtained using the weight of 0.3, 0.3 and 0.4 for w_1 , w_2 and w_3 respectively.

Table 5.6 Results for Normalised Formulation

Weights(w_1, w_2, w_3)	Minimum Time(sec)
(0.1,0.1,0.8)	3.9804
(0.2,0.2,0.6)	4.0106
(0.2,0.6,0.2)	4.0044
(0.6,0.2,0.2)	4.0124
(0.05,0.15,0.8)	4.0121
(0.1,0.0,0.9)	3.9531
(0.1,0.8,0.1)	4.0109
(0.8,0.1,0.1)	3.9673
(0.3,0.3,0.4)	3.9570
(0.3,0.4,0.3)	3.9628
(0.4,0.3,0.3)	3.9875
(0.5,0.2,0.3)	3.9807

The effect of the different number of generations is indicated in Figure (2), while Table 5.7 shows the effect of changing the size of the genetic population.

Figure (2): Generations size vs. motion time

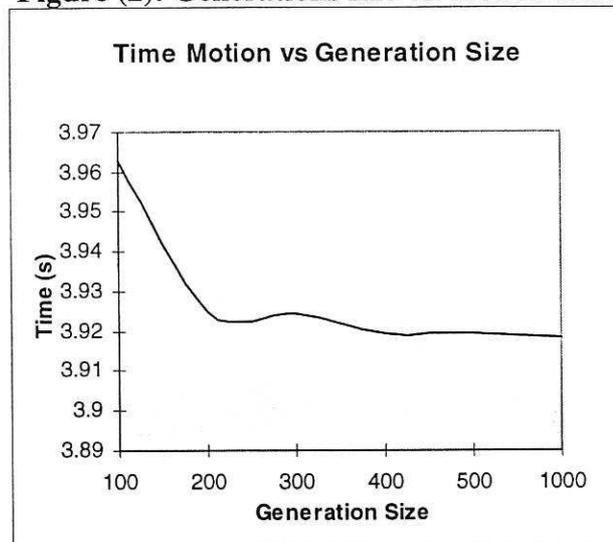


Table 5.7 Effects of Different Population Size*

Population Size	Minimum Time(sec)
100	4.0149
200	3.9804
300	3.9630

* Results obtained are using weight of (0.1,0.1,0.8) and 100 generation size.

(b) Non-normalised formulation

A set of non-normalised weights are also simulated. The results obtained when considering different combinations of weights are shown in the following Table 5.8, and show no motion time improvement in comparison with the normalised approach.

Table 5.8 Results for Non-Normalised Formulation*

Weights($\omega_1, \omega_2, \omega_3$)	Minimum Time(sec)
(400,200,400)	3.9984
(0.01,30,4000)	4.0057
(0.001,0.008,0.001)	4.0075
(5,4,1)	4.0260
(5,4,20)	3.9881
(5,1,20)	3.9441
(5,10,20)	4.0000
(5,0,20)	4.0066
(5,2,20)	3.9485
(4,1,20)	4.0209
(6,1,20)	3.9847
(8,1,20)	4.0034

* Results obtained are using 200 population size and 100 generations

(c) Choice of weights

Moving from one set of weights to another may result in skipping a non-dominated point. In other words, it is quite possible to miss using weights that would lead to an extreme point (optima). Consequently, the most that should be expected from the weighting method is an approximation of the non-dominated set. While this approach may yield meaningful results only when solved many times for different values of weights, the results reported in this Case show the difficulty of realising a solution. Normalised formulation has narrowed down the scope to search for the appropriate combination of weights, but the range of joint travel time is only an approximation.

6 CONCLUSIONS

In comparing between the Pareto GA, Breeder GA, Weighted-Sum GA and the Flexible Polyhedron Search, the following points were observed.

1. The customised path redistribution-relaxation operator has produced the best performance in the case studies amongst all tested operators, supporting the opinion to custom-build operators to the requirements of an application.
2. Breeder GA operators can perform better in non-BGA environment (e.g. truncation selection environment).
3. Results produced by weighted-sum GA depends on the weight given to the objective function, generation size and population size. Hence there are too many parameters to be tested to obtain an optimum solution. The difficulties with setting these initialisation parameters are high and time-consuming.
4. Both GA methods, weighted-sum and Pareto-based, require significant amount of processing time to obtain the optimum motion time, while FPS requires much less computations.
5. Pareto-based GA appear to have an edge in optimisation over weighted-sum GA, because it can produce a better optimum motion time (though small in comparison) using smaller generation and population sizes.
6. The optimality of the motion time is improving with population and generation size. However, a compromise is needed to ensure limiting processing time for complex cases.

Further investigations are in progress.

REFERENCES

1. Fonseca, C. M. and Fleming, P. J. (1995b). An Overview of evolutionary algorithms in multiobjective optimisation. *Evolutionary Computation.*, **3**, No. 1, pp. 1-16.
2. Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimisation & Machine Learning*. Reading: Addison-Wesley Publishing Company, Inc.
3. Wang, Q. and Zalzala, A.M.S., "Genetic algorithms for PUMA robot motion control: A practical implementation", *Int. J. Mechatronics*, Vol.6, No.3, pp 349-65, 1996.
4. Rana, A.S. and Zalzala, A.M.S., "Near time-optimal collision-free motion planning of robotic manipulators using an evolutionary algorithm", *Robotica*, Vol. 14, pp. 621-32, 1996.
5. Himmelblau, D. M. ((1972). *Applied Non-linear Programming*. New York: McGraw-Hill, Inc.
6. Lin, C.-S., Chang, P.-R. and Luh, J. Y. S. (1983). Formulation and optimisation of cubic polynomial trajectories for industrial robots. *IEEE Trans. Automatic Control*, **AC-28**, No. 12, pp. 1066-1073.
7. Ben-tal, A. (1980). Characterisation of Pareto and lexicographic optimal solutions. In Fandel, G. and Gal, T. (ed.), *Lecture Notes in Economics and Mathematical Systems: Multiple Criteria Decision Making Theory and Application*, **177**, pp. 1-11. Berlin: Springer-Verlag.
8. Whitley, D. (1989). The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. *Genetic algorithms: Proceedings of the Third International Conference on Genetic Algorithms*, pp. 116-121. Schaffer, J. D. ed., Morgan Kaufmann.
9. Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm, *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14-21 Grefenstette, J. J. ed., Lawrence Erlbaum Associates, Publishers.
10. Mühlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimisation. *Evolutionary Computation*, **1**, No. 1, pp. 25-49.

