Closed-Loop Time-Optimal Path Planning Using a Multi-Objective Diversity Control Oriented Genetic Algorithm

Nantawatana Weerayuth* and Nachol Chaiyaratana**

Abstract

This paper presents the use of a multi-objective diversity control oriented genetic algorithm (MODCGA) for solving a closed-loop time-optimal path planning problem. The MODCGA is a result of the integration between two types of genetic algorithm: a multi-objective genetic algorithm (MOGA) and a diversity control oriented genetic algorithm (DCGA). The MODCGA is benchmarked against the MOGA and a random search in the path planning problem which is treated as a multi-objective optimisation problem. In this case, the planning problem is represented by a position control task which is given to a 3-dof revolute joint robot. From the optimisation viewpoint, the decision variables consist of the magnitude of torque limits for each joint and the initial and final positions of a fixed length path at which the robot end-effector has to track.The corresponding search objectives are thus expressed in terms of the position tracking error and trajectory time. Two chromosome coding schemes are explored in this investigation: Gray and integer-based coding schemes. The simulation results suggest that the integer-based coding scheme is more suitable at representing the decision variables. In addition, the use of diversity control in conjunction with the integer-based coding scheme can further improve the search results.

Keywords : Genetic algorithm, path planning, robotics, time-optimal control

1. Introduction

Time-optimal control has been one of the major research interests in robotics during the past decade. Time-optimality can lead to an overall improvement in the level of productivity from a manufacturing viewpoint and an increase in the effectiveness of a task execution from an operational viewpoint. One particular aspect of research is the theory and application of time-optimal control of a robot arm along a pre-defined path. An algorithm that can lead to time-optimality of this kind was firstly developed by Bobrow et al. [1]. Over the years, this algorithm has undergone a number of refinements and one of the latest modifications has been described in Shiller and Lu [2]. In summary, a time-optimal motion of a robot arm along a pre-defined path is achieved when the motion is executed with either the maximum possible acceleration or deceleration along the path. This can be done when one of the actuators on the robot arm is always saturated and the other actuators adjust their torque values so that their torque limits are not violated [3].

Although this time-optimal control algorithm has been proven to be useful in a number of tasks, the use of an additional path planning algorithm is usually required. This is because one necessary input for the time-optimal control algorithm is the pre-defined path of end-effector in the Cartesian space. Since the time-optimal control algorithm is

* Department of Mechanical Engineering, Ubonratchathani University.
** Research and Development Center for Intelligent Systems, King Mongkut's Institute of Technology North Bangkok.

developed by considering only the open-loop dynamics of the robotic system [1-2], early works in the area of time-optimal path planning are usually carried out in the open-loop mode [4-5]. However, closed-loop path planning has also received much attention since it can produce a more accurate result in terms of the difference between the desired path and the actual path obtained. This is because the closed-loop dynamics has been taken into the consideration during the planning process. Nevertheless, the use of the time-optimal control algorithm in conjunction with closed-loop path planning has one drawback; actuator dynamics and the delays caused by an on-line feedback controller would lead to a reduction in the efficiency of the algorithm [6]. Three possible methods have been used to solve this problem. The first method is based on a modification of the original time-optimal control problem into a time-energy optimal control problem which can be regarded as a lagrangian constraint optimisation problem and can only be solved numerically [7]. A drawback of this method is that the modification also leads to an increase in the resulting trajectory time. The second method is based on the use of a simplified friction model to compensate for the actuator dynamics and the implementation of a trajectory pre-shaping to account for the dynamics of the controller [6]. Finally, the third method covers the use of a neural network which is trained using either feedback error learning [8] or model-based reinforcement learning [9] as an additional controller in the control loop. The primary function of this neural network is to compensate for modelling errors and delays caused by the main controller in the system. It has also been demonstrated that the compensation performance of the neural network controller is higher than that of the trajectory pre-shaper.

The works initiated by Chaiyaratana and Zalzala [8-9] will be continued in this paper where the investigation will concentrate only on the treatment on the closed-loop time-optimal path planning process as a multi-objective optimisation problem. The interested planning problem is inspired by an observation that many tasks in manufacturing systems can be accomplished using lesser processing time provided that the trade-off in the product quality is acceptable. For instance, in tasks like welding and edge-deburring, the time that the robot end-effector required to track the pre-programmed path can be reduced if the allowable tracking error bound is increased. The optimisation problem interested involves the selection of torque limit combination and the initial and final positions of a fixed length path where the search objectives are expressed in terms of the position tracking error and trajectory time. An approach on multi-objective optimisation using a genetic algorithm, namely a multi-objective diversity control oriented genetic algorithm (MODCGA) will be used to solve the problem. The MODCGA is a result of the integration between a multi-objective genetic algorithm or MOGA [10] and a diversity control oriented genetic algorithm or DCGA [11]. Note the additional neural network controllers as described in Chaiyaratana and Zalzala [9] are used in closed-loop planning process in order to minimise the effect of closed-loop dynamics on the planning results.

This paper is organised as follows. The time-optimal control algorithm as described by Shiller and Lu [2] is briefly explained in section 2. In addition, the use of additional neural network controllers in the closed control loop is also explained in this section. In section 3, the overview of the closed-loop path planning problem is discussed. In section 4, the background on the MOGA and DCGA and the genetic algorithm integration will be discussed. The application of the MODCGA on the closed-loop time-optimal path planning problem will be explained in section 5. The simulation results obtained after applying the MODCGA to the problem are shown in section 6. Finally, discussions on the simulation results and conclusions are given in section 7.

2. Time-Optimal Control Algorithm and Neural Network Controllers

In summary, the time-optimal control algorithm as described by Shiller and Lu [2] can be used to generate the time-optimal profiles of the reference joint position and the open-loop control torque signal provided that the physical properties of the robot arm are known and a pre-defined path of the robot arm in the workspace is available. In particular, the torque limits on the actuators within the robot are the key factors which have a major influence on the trajectory time obtained from the algorithm. As stated earlier, the time-optimal motion is achieved when one of the actuators on the robot arm is always saturated and the torque values of other actuators are within the bounds of the corresponding limits. This means that with the large values of the torque limits, the obtained trajectory time will be short. On the other hand, with the smaller values of the torque limits, the obtained trajectory time will be relatively larger. A schematic diagram describing input and output of the time-optimal control algorithm is given in Figure 1. In Figure 1, the time-optimal control algorithm takes the robot physical properties and the information regarding the pre-defined robot's path as inputs. The outputs from the algorithm are the reference joint position and the open-loop torque profiles.

Nonetheless, the time-optimal control algorithm will produce a result based on the open-loop

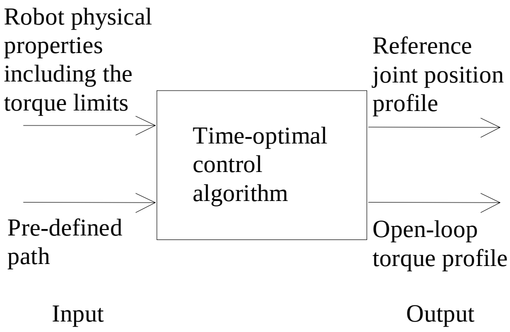

Figure 1 Schematic diagram of the time-optimal control algorithm.

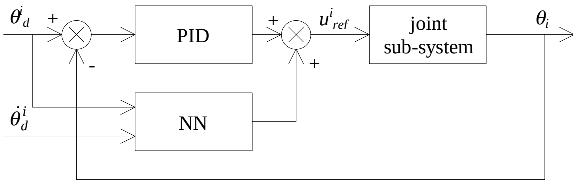

Figure 2 Neural network and PID controllers in each joint control loop.

dynamics of the system. This means that a certain number of problems will arise when using the reference joint position profile obtained from the algorithm as input to the closed-loop system [6-7]. In order to solve the problem, Chaiyaratana and Zalzala [8-9] propose the use of neural networks as additional controllers in the closed control loop where the neural networks have a role of compensating for the dynamics of the primary controllers and the possible modelling errors. This arrangement is illustrated in Figure 2.

In Figure 2, the joint sub-system block represents a linear second order system which is obtained after de-coupling the robot model using a non-linear de-coupled feedback control scheme [12]. With the use of additional neural network controllers, the de-coupling scheme can be executed with high efficacy even when there exists modelling errors in the system [8-9]. Note that in this paper the neural network controllers utilised during the planning process are trained using a model-based reinforcement learning strategy.

3. Close-loop Time-Optimal Part Planning

The interested path planning problem involves the use of a 3-dof robot in a position control task. The robot is required to track a one-metre straight-line path which is illustrated in Figure 3. Referring to Figure 3, point *A* (0.736, 0.226, 0.093) is the initial location of the robot end-effector and point *B* (0.0, 0.854, 0.354) is the final desired location of the robot end-effector on this path. The time-optimal control algorithm is then used to generate the trajectory

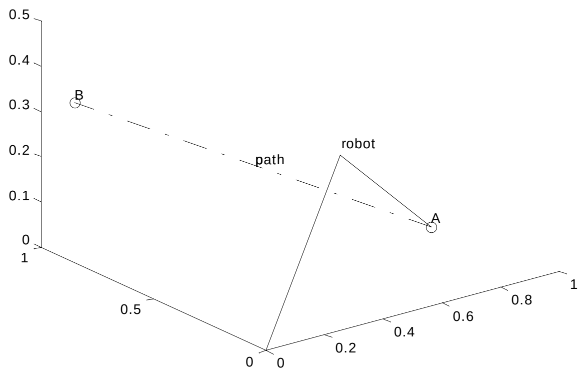

Figure 3 Robot and the straight-line path.

time history, which is subsequently used as the input to the position control loop.

In practice, the maximum torque limits, which are used in the time-optimal trajectory calculation process for a closed-loop control, are usually less than the actual torque limits on the actuators. This safety precaution is done in order to allow some margins of error for possible discrepancies introduced to the system by modelling errors and controller dynamics [6]. This implies that for a given set of the actual torque limits of the actuators, there is a set of admissible torque limit combinations that can lead to a certain level of time-optimality within an acceptable range of tracking error. In addition, in certain applications such as welding or edge-deburring it is possible to modify the end-effector trajectory in Cartesian space without effecting the task requirement provided that the position and orientation of the work piece at which the end-effector has to remain in contact with can be modified accordingly. The position control task discussed above is an example which reflects such applications. By modifying the initial and final locations of the straight-line path, the task description in the application viewpoint would remain the same while the angular trajectory at which the robot joint has to follow would be different. Such change in the angular trajectory would lead to a variation in the position tracking error. Combining with the issue on torque limits, this points to a design problem in robotic applications. The objective of such problem is to find a combination of torque limits from a set of admissible torque ranges and the initial and final position of the end-effector which will lead to a trajectory which meets the time-optimality and tracking error constraints. This is a multi-objective optimisation problem since it would be highly unlikely to obtain a single trajectory that can minimise both the trajectory time and tracking error simultaneously. A multi-objective diversity control oriented genetic algorithm (MODCGA) will be used to solve the problem associated with the torque limit and end-effector position selection in this study. The description of the MODCGA will be given in the next section.

4. Multi-Objective Diversity Control Oriented Genetic Algorithm

The multi-objective diversity control oriented genetic algorithm (MODCGA) is a result of the integration between a multi-objective genetic algorithm or MOGA [10] and a diversity control oriented genetic algorithm or DCGA [11]. A brief description of the algorithms follows.

4.1 Multi-Objective Genetic Algorithm

The multi-objective genetic algorithm (MOGA) was first introduced by Fonseca and Fleming [10]. The MOGA functions by seeking to optimise the components of a vector-valued objective function. Unlike single-objective optimisation, the solution to a multi-objective optimisation problem is a family of points known as the Pareto optimal set. Each point in the set is optimal in the sense that no improvement can be achieved in one component of the objective vector that does not lead to degradation in at least one of the remaining components. Given a set of possible solutions, a candidate solution is said to be Pareto optimal if there are no other solutions in the solution set that can dominate the candidate solution. In other words, the candidate solution would be a non-dominated solution. Assuming, without loss of generality, a minimisation problem, an n-dimensional cost vector **a** is said to be dominating another n-dimensional cost vector **b** if, and only if, **a** is partially less than **b** (**a** $p<$ **b**), i.e.

$$\mathbf{a}\ p<\mathbf{b} \leftrightarrow \forall i=1,...,n : a_i \leq b_i \wedge \exists i=1,...,n : a_i < b_i \qquad (1)$$

By identifying the number of solutions in the solution set that dominate the solution of interest, a rank value can be assigned to the solution. In other words, the rank of a candidate solution is given by the number of solutions in the solution set that dominate the candidate solution. After a rank has been assigned to each solution, a fitness value can then be interpolated onto the solution where a genetic algorithm can subsequently be applied in the optimisation procedure. Note that since the aim of a search by the MOGA is to locate Pareto optimal solutions, in essence the multi-objective optimisation problem has also been treated as a multi-modal problem. Hence, the use of additional genetic operators including the fitness sharing and mating restriction procedures is also required. However, in addition to the usual application of the fitness sharing and mating restriction procedures in the decision variable space [13], they can also be carried out in the objective value space [10]. A comprehensive description of the MOGA which covers other advanced topics including goal attainment and priority assignment strategies can be found in Fonseca and Fleming [14].

4.2 Diversity Control Oriented Genetic Algorithm

The diversity control oriented genetic algorithm (DCGA) was first introduced by Shimodaira [11]. Similar to other single-objective steady-state genetic algorithms, the parent population and the offspring population are merged together during the DCGA run where the appropriated individuals are extracted from the merged population. However, instead of selecting the highly fit individuals from the population straightaway, the extraction process in the DCGA starts with the elimination of duplicated individuals in the merged population. The remaining individuals are then sorted according to their fitness values in descending order. Following that the best individual from the remaining individuals is determined and kept for passing onto the next generation. Then either a cross-generational deterministic survival selection (CDSS) method or a cross-generational probabilistic survival selection (CPSS) method is applied in the top-down fashion to the remaining non-elite individuals in the sorted array. In the case of the CDSS, the remaining non-elite individuals will have an equal chance of being selected. In contrast, a survival probability value is assigned to each non-elite individual according to its similarity to the best individual in the case of the CPSS. If the genomic structure of the individual interested is very close to that of the best individual, the survival probability assigned to this individual will be close to zero. On the other hand, if the chromosome structure of this individual is quite different from that of the best individual, its survival probability will be close to one. Each individual will then be selected according to the assigned survival probability. If the total number of all selected individuals including the pre-selected elite individual does not reach the required population size after the survival selection loop, randomly generated individuals will be added to the individual array until the required number is met. A comprehensive description of the DCGA and its benchmarking performance in various test problems can be found in Shimodaira [15]. Note that in this investigation, the DCGA mechanism that is a part of the MODCGA is the one which utilises the CPSS method.

4.3 Genetic Algorithm Integration

By combining the MOGA and the DCGA together, the resulting algorithm can be referred to as a multi-objective diversity control oriented genetic algorithm or MODCGA. Similar to the MOGA, the rank of each individual will be obtained after comparing it with the remaining individuals. However, the comparison will be made among individuals in the merged population which is the result from combining parent and offspring populations together. Since the best individuals in the MOGA are the non-dominated individuals, there will be more than one survival probability value which can be assigned to each dominated individual. In this study, the lowest value in the probability value set is chosen for each dominated individual. After the survival selection routine is completed and the fitness values have been interpolated onto the individuals, the standard genetic operations can then be applied to the population in the usual way.

5. Application of the MODCGA on the Closed-Loop Time Optimal Path Planning Problem

The multi-objective diversity control oriented genetic algorithm (MODCGA) will be used to solve the closed-loop time-optimal path planning problem. The problem formulation and the genetic operators used are discussed as follows.

5.1 Decision Variables

A 3-dof robot with the task of tracking a straight-line path in Cartesian space presented earlier is used to demonstrate this multi-objective optimisation problem. The decision variables of the problem consist of the torque limit combination and the initial and final positions of the end-effector. Assuming that the magnitudes of the maximum and minimum torque limits are the same for each actuator, the torque limit part of the decision variables would consist of the magnitude of the torque limits of each joint. In this study, the range of the magnitudes of the torque limits on joints 1, 2 and 3 are set to 15-30, 25-40 and 5-20 Nm, respectively. The lower bounds of the limits (i.e. 15, 25, 5) are based on the maximum allowable trajectory time requirement of 0.3 seconds, while the upper bounds of the torque limits (i.e. 30, 40, 20) are set by the actual torque limits of the actuators.

Moving onto the part of decision variables which involves the positions of the end-effector. In order to create a fixed-length path in Cartesian space, two vectors are required: the position vector for the initial position of the end-effector and the direction vector pointing from the initial position toward the desired final position of the end-effector. This requirement can be achieved by setting up two search variables. The first variable will be the initial location of the end-effector while the second variable will be another point in the robot workspace at which a direction vector pointing from the initial position of the end-effector toward this point can be established. In this investigation the search range for the initial position of the end-effector is given by (0.721-0.751, 0.211-0.241, 0.078-0.108) in the *x*, *y* and *z* directions, respectively. In contrast, the search range for the location of the other point in the robot workspace is set to (-0.015-0.015, 0.839-0.869, 0.339-0.369) in the *x*, *y* and *z* directions, respectively. Note that the search ranges for these two points are in the vicinity of the initial and final positions of the straight-line path described earlier in section 3.

5.2 Objective Variables

There are two optimisation objective variables in this problem: the tracking error and the trajectory time objectives. The tracking error objective is expressed in terms of the sum of the mean absolute errors over three joints, calculated over the whole trajectory. The trajectory time objective is the optimal trajectory time obtained from the time-optimal control algorithm. Note that the sampling period used in the simulation of this 3-dof robotic closed-loop system is 0.01 seconds. Hence, the trajectory time will always be in the form of 0.01m where m is a positive integer.

5.3 Chromosome Coding

Nine decision variables - the magnitudes of the torque limits from all three joints and the co-ordinates along three axes of the two points for identifying the straight-line path - are concatenated together and coded to form a chromosome. Two chromosome coding schemes are explored here: Gray and integer-based coding schemes. The torque ranges for all three joints are discretised using a search step of 0.5 Nm. This leaves 31 search points for the magnitude of the torque limits of each joint. In a similar way, the search ranges of the co-ordinates of the two points for dictating the location of the straight-line path are discretised using a search step of 0.001 m. This also leaves 31 search points for the co-ordinate in each axis. With the use of a Gray coding scheme, a Gray code of length 5 can be used to represent a decision variable. The total length of the chromosome in this case would be equal to 45. Note that there are certain search points obtained after decoding the chromosome which lie outside the required search space. These points are mapped back into the feasible region by changing the most significant bit of the Gray code section representing the particular decision variable that violates the feasibility constraint into zero. In contrast to the case of the Gray coding scheme, with the use of an integer-based coding system a single gene can be used to represent a decision variable. Each gene can then take an allele value from a set which is composed of 31 integers ranging from 0 to 30. The chromosome length in this case would be equal to nine.

5.4 Fitness Assignment and Fitness Sharing

The ranking method as described in Fonseca and Fleming [10] is used to rank each individual in the population. Following that, a linear fitness interpolation is used to assign fitness to each individual. Fitness sharing, with the use of triangular sharing function, is then carried out in normalised objective space.

5.5 Selection, Crossover and Mutation Methods

Stochastic universal sampling selection [16] is used in the fitness selection.Then a standard one-point crossover technique is used in the recombination. Two individuals are allowed to perform crossover if, and only if, they are within the mating restriction distance from each other. For simplicity, the mating restriction radius is set to equal to the sharing radius and the consideration on the distance between the two individuals is also done in normalised objective space. For the case of chromosome coding using a Gray code, a standard bit-flipped operation is used for the mutation. In contrast, the value 1 will be added to or subtracted from the allele value of the mutated gene to achieve mutation in the integer-based coding system.

5.6 Diversity Control

After offspring individuals are created, they are combined with parent individuals where duplicate individuals in the merged population are eliminated. The remaining individuals are then sorted according to their ranks in descending order. Following that the non-dominated individuals from the remaining individuals are determined and kept for passing onto the next generation. Then a cross-generational probabilistic survival selection (CPSS) method is applied to the remaining dominated individuals where a probability value is assigned to each individual according to its similarity to the genomically closest non-dominated individual. The survival probability of an individual is given by

$$p_s = \{(1-c)h/L + c\}^{\alpha} \tag{2}$$

where p_s denotes the survival probability, h is the genomic difference between the interested individual and its closest non-dominated individual, L is the highest possible genomic difference between two individuals, c is the shape coefficient and α is the exponent coefficient. Each individual will then be selected according to the assigned survival probability. If the total number of all selected individuals including the pre-selected non-dominated individuals does not reach the required population size after the survival selection loop, randomly generated individuals will be added to the individual array until the required number is met.

For the purpose of comparison, the MOGA and the random search technique is also used to find the Pareto optimal solutions in this study. The parameter settings for the MOGA and MODCGA are summarised in Table 1. The description of the case studies explored and the simulation results will be given in the next section.

Table 1 Parameter Settings for the MOGA and MODCGA

Parameter	Value
Chromosome length	
Gray code	45
Integer-based code	9
Crossover probability	0.8
Mutation probability	
Gray code	0.2
Integer-based code	0.01
Sharing and mating restriction radii	0.03
Diversity control (MODCGA only)	
Shape coefficient, c	0.235
Exponent coefficient, α	0.51
Population size	30
Number of generations	30

6. Simulation Results

Two case studies are investigated in this paper. The aim of the first case study is to find a set of torque limit combinations and straight-line paths which lead to trajectories with the sum of the mean absolute tracking errors $\leq$ 0.15708 radians (3 degrees per joint) and the trajectory time $\leq$ 0.27 seconds. The aim of the second case study is to find a set of torque limit combinations and straight-line paths which lead to trajectories with the sum of the mean absolute tracking errors $\leq$ 0.07854 radians (1.5 degrees per joint) and the trajectory time $\leq$ 0.30 seconds. The purpose of the first case study is to find solutions that concentrate more on optimising the trajectory time while the second case study emphasises on the tracking error optimisation. The simulation results for these two cases are summarised in Tables 2-5. Note that the displayed results are the combination of Pareto optimal solutions obtained from five different simulation runs. In addition, the initial populations used in both approaches of the

Table 2 Pareto Optimal Solutions from Case I Gray Code

Random Search		MOGA		MODCGA	
t	SMAE	t	SMAE	t	SMAE
		0.21	0.14613		
0.22	0.12512	0.22	0.11324	0.22	0.11340
0.23	0.10976	0.23	0.10090	0.23	0.10160
0.24	0.09433	0.24	0.08179	0.24	0.08670
0.25	0.07003	0.25	0.06801	0.25	0.06970
0.26	0.05950	0.26	0.05544	0.26	0.05360
0.27	0.05298	0.27	0.04156	0.27	0.04450

t - Trajectory time (second)

SMAE - Sum of mean absolute tracking errors (rad)

Table 3 Pareto Optimal Solutions from Case I - Integer - Based Code

Random Search		MOGA		MODCGA	
t	SMAE	t	SMAE	t	SMAE
		0.21	0.14403		
0.22	0.12512	0.22	0.12255	0.22	0.11200
0.23	0.10976	0.23	0.10061	0.23	0.09630
0.24	0.09433	0.24	0.08789	0.24	0.08710
0.25	0.07003	0.25	0.06955	0.25	0.06940
0.26	0.05950	0.26	0.05415	0.26	0.05630
0.27	0.05298	0.27	0.04084	0.27	0.04890

Table 4 Pareto Optimal Solutions from Case II - Gray Code

Random Search		MOGA		MODCGA	
t	SMAE	t	SMAE	t	SMAE
0.25	0.07003	0.25	0.07145	0.25	0.07000
0.26	0.05950	0.26	0.05799	0.26	0.05720
0.27	0.05298	0.27	0.04235	0.27	0.04180
0.28	0.03582	0.28	0.03104	0.28	0.03190
0.29	0.02312	0.29	0.02182	0.29	0.02130
0.30	0.02224	0.30	0.01649	0.30	0.01810

t - Trajectory time (second)

SMAE - Sum of mean absolute tracking errors (rad)

Table 5 Pareto Optimal Solutions from Case II - Integer- Based Code

Random Search		MOGA		MODCGA	
t	SMAE	*t*	SMAE	*t*	SMAE
0.25	0.07003	0.25	0.06986	0.25	0.06950
0.26	0.05950	0.26	0.05371	0.26	0.05610
0.27	0.05298	0.27	0.04686	0.27	0.04600
0.28	0.03582	0.28	0.03014	0.28	0.03540
0.29	0.02312	0.29	0.01994	0.29	0.01910
0.30	0.02224	0.30	0.01760	0.30	0.01700

t - Trajectory time (second)

SMAE - Sum of mean absolute tracking errors (rad)

MOGA and MODCGA in each simulation run are generated such that the resulting decision variables are the same. In other words, the initial populations used in the two approaches are equivalent in terms of the decision variables obtained after decoding the chromosomes.

7. Discussions and Conclusions

Prior to any analyses on the simulation results can be carried out, a number of points are required to be made clear. The Pareto front results are used to represent two main aims of the search; these are to find the range of variety in solutions and to locate the solutions which are close to the true Pareto optimal solutions of the problem. For this path planning problem, the exact range of variety in solutions is known. Such knowledge is gained by inspecting the non-dominated solutions and their corresponding objectives in the solution set itself. This statement will be made clearer later on in the discussions. Nonetheless, similar to the majority of engineering applications, the theoretical, or true, Pareto optimal solutions of the problem are not known. Of course, there will be a possibility that some of the Pareto optimal solutions found by one technique can be dominated by the solutions found by another technique. In order to compare the Pareto optimal solutions obtained from each technique objectively, both points of view on the variety in solutions found and the number of solutions found which cannot be dominated by the solutions obtained from other techniques need to be considered.

First of all, consideration is placed on the simulation results from the first case study. Both the MOGA with a Gray coding scheme and the MOGA with an integer-based coding scheme can locate seven distinct solutions while the random search and both approaches of the MODCGA fail to locate a solution with the trajectory time of 0.21 seconds. For this case study, there can be only seven distinct solutions in the Pareto optimal solution set. This is because the solution that has a trajectory time of 0.21 seconds and still has the tracking error within the target value is obtained for magnitudes of torque limits which are close to the actual limits on the actuator torque. In addition, there are only seven distinct solutions which can occupy the trajectory time solution space from t = 0.21 seconds to t = 0.27 seconds with an increment of 0.01 seconds (the sampling period). With a close inspection, it is noticeable that all solutions found by both approaches of the MOGA and MODCGA dominate all optimal solutions found by the random search. However, after comparing the results found by both approaches of the MOGA, it is found that the solutions with the trajectory times of 0.21, 0.23, 0.26 and 0.27 seconds found by the MOGA with a Gray coding scheme are dominated by the corresponding solutions found by the MOGA with an integer-based coding scheme. At the same time, the solutions found by the MOGA with an integer-based coding scheme which have trajectory times of 0.22, 0.24 and 0.25 seconds are dominated by the solutions obtained by the MOGA with a Gray coding scheme. In this respect, it can be said that the search performances of the two MOGA approaches are very close to one another. In addition, it is observable that the diversity control effect embedded in the MODCGA worsen the search performance in the case of Gray coding scheme while the same effect helps improving the search results in the case of integer-based coding scheme. From the simulation results, only the solution with the trajectory time of 0.25 seconds identified by the MODCGA with a Gray coding scheme dominates the corresponding solution found by the MOGA. In contrast, the solutions found by the MOGA with an integer-based coding scheme which have trajectory times of 0.22, 0.23, 0.24 and 0.25 seconds are dominated by the corresponding solutions obtained by the MODCGA. Hence, the use of diversity control in the multi-objective search is recommended only in the case of integer-based coding scheme for the first case study.

Moving onto the second case study: all search techniques are capable of locating six distinct solutions. Note that for this case study, there can be a maximum of six distinct solutions in the Pareto optimal solution set. This is concluded from the results obtained from the first case study which indicates that the solution which has the minimum allowable trajectory time and also has the tracking error which is smaller than 0.07854 radians is the one with the trajectory time of 0.25 seconds. With the maximum allowable trajectory time being limited to 0.3 seconds by the search target and the sampling period is set to 0.01 seconds, there are only six distinct solutions with the trajectory times ranging from 0.25 to 0.30 seconds that can cover the whole Pareto front. The simulation results in this case study also reveals that all solutions found by the MOGA with an integer-based coding scheme and both approaches of the MODCGA dominates all solutions found by the random search. In contrast, the MOGA with a Gray coding scheme can only find five solutions which dominate the solutions located by the random search. The only solution found by the MOGA with a Gray coding scheme which is dominated by the solution found by the random search is the one with the trajectory time of 0.25 seconds. Among the solutions found by the two MOGA approaches, two solutions found by the MOGA with a Gray coding scheme dominates the solutions located by the MOGA with an integer-based coding scheme.These two solutions are the solutions with the trajectory times of 0.27 and 0.30 seconds. In contrast, the MOGA with an integer-based coding scheme can locate four distinct solutions that dominates the solutions found by the MOGA with a Gray coding scheme: the solutions with the trajectory times of 0.25, 0.26, 0.28 and 0.29 seconds. In overall, it can be noticed that the performance of the MOGA with an integer-based coding scheme is slightly higher than that of the MOGA with a Gray coding scheme. In addition, it is noticeable that the diversity control effect helps improving the search results in both cases of chromosome coding schemes. From the results, the solutions found by the MOGA with a Gray coding scheme which have trajectory times of 0.25, 0.26, 0.27 and 0.29 seconds are dominated by the corresponding solutions obtained by the MODCGA. Moreover, the solutions identified by the MOGA with an integer-based coding scheme which have trajectory times of 0.25, 0.27, 0.29 and 0.30 seconds are dominated by the corresponding solutions generated by the MODCGA. In other words, the use of diversity control is recommended for both cases of coding schemes in the second case study.

From the above discussions, it can be concluded that for this closed-loop time-optimal path planning problem, the integer-based chromosome is more suitable than the Gray chromosome at representing the decision variables. In addition, the use of diversity control in conjunction with the integer-based coding scheme is also recommended. Nonetheless, the use of diversity control can also effect the range of variety in solutions identified by the genetic algorithm. Further investigation is required in order to eliminate this drawback.

References

1. Bobrow, J. E., Dubowsky, S. and Gibson, J. S. “Time-optimal control of robotic manipulators along specified paths.” *International Journal of Robotics Research*. 4, 3 (1985) : 3-17.

2. Shiller, Z. and Lu, H.-H. "Computation of path constrained time optimal motions with dynamic singularities." *Transactions of the ASME. Journal of Dynamic Systems, Measurement, and Control*. 114, 1 (1992) : 34-40.
3. Sahar, G. and Hollerbach, J. M. "Planning of minimum-time trajectories for robot arms." *International Journal of Robotics Research.* 5, 3 (1986) : 90-100.
4. Croft, E. A., Benhabib, B. and Fenton, R. G. "Near-time optimal robot motion planning for on- line applications." *Journal of Robotic Systems.* 12, 8 (1995) : 553-567.
5. Rana, A. S. and Zalzala, A. M. S. "Near time-optimal collision-free motion planning of robotic manipulators using an evolutionary algorithm." *Robotica*. 14, 6 (1996) : 621-632.
6. Shiller, Z.,Chang, H.and Wong,V."The practical implementation of time-optimal control for robotic manipulators." *Robotics and Computer-Integrated Manufacturing.* 12, 1 (1996) : 29-39.
7. Shiller, Z. "Time-energy optimal control of articulated systems with geometric path constraints." *Transactions of the ASME. Journal of Dynamic Systems, Measurement, and Control.* 118, 1 (1996) : 139-143.
8. Chaiyaratana, N. and Zalzala, A. M. S. "Hybridisation of neural networks and genetic algorithms for time-optimal control." In *The 1999 Congress on Evolutionary Computation (CEC'99)*, 389-396. Washington, DC, 1999.
9. Chaiyaratana, N. and Zalzala, A.M.S. "Time-optimal path planning and control using neural networks and a genetic algorithm." In *Proceedings of the 2001 ASME International Mechanical Engineering Congress and Exposition*, DSC-24512/1-8. New York, NY, 2001.
10. Fonseca, C. M. and Fleming, P. J. "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization." In *Genetic Algorithms: Proceedings of the 5th International Conference*, 416-423. Urbana-Champaign, IL, 1993.
11. Shimodaira, H. "DCGA: A diversity control oriented genetic algorithm." In *The Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'97)*, 444-449. Glasgow, UK, 1997.
12. Freund, E. "Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators." *International Journal of Robotics Research.* 1, 1 (1982) : 65-78.
13. Fonseca C. M. and Fleming, P. J. "Multiobjective genetic algorithms made easy : Selection, sharing and mating restriction." In *The First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95)*, 45-52. Sheffield, UK, 1995.
14. Fonseca, C. M. and Fleming, P. J. "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part 1: A unified formulation." *IEEE Transactions on Systems, Man, and Cybernetics.* 28, 1 (1998) : 26-37.
15. Shimodaira, H. "A diversity-control-oriented genetic algorithm (DCGA): Performance in function optimization." In *The 2001 Congress on Evolutionary Computation (CEC'01)*, 44-51. Seoul, Korea, 2001.
16. Baker, J. E. "An analysis of the effects of selection in genetic algorithms." Ph.D. Dissertation, Department of Computer Science, Vanderbilt University, Nashville, TN, 1989.