Multi-Objective Genetic Manipulator Trajectory Planner

E. J. Solteiro Pires[1⋆], P. B. de Moura Oliveira[1], and J. A. Tenreiro Machado[2]

[1] Universidade de Trás-os-Montes e Alto Douro, Dep. de Engenharia Electrotécnica, Quinta de Prados, 5000–911 Vila Real, Portugal,
epires@utad.pt, http://www.utad.pt/~epires
oliveira@utad.pt,http://www.utad.pt/~oliveira

[2] Instituto Superior de Engenharia do Porto, Dep. de Engenharia Electrotécnica, Rua Dr. António Bernadino de Almeida, 4200-072 Porto, Portugal,
jtm@dee.isep.ipp.pt, http://www.dee.isep.ipp.pt/~jtm

Abstract. This paper proposes a multi-objective genetic algorithm to optimize a manipulator trajectory. The planner has several objectives namely the minimization of the space and join arm displacements and the energy required in the trajectory, without colliding with any obstacles in the workspace. Simulations results are presented for robots with two and three degrees of freedom, considering the optimization of two and three objectives.

1 Introduction

In the last decade genetic algorithms (*GA*s) have been applied in a plethora of fields such as in control, system identification, robotics, planning and scheduling, image processing, pattern recognition and speech recognition [1]. This paper addresses the planning of trajectories, that is, the development of an algorithm to find a continuous motion that takes the manipulator from a given starting configuration to a desired end position in the workspace without collision with any obstacle.

Several single-objective methods for trajectory planning, collision avoidance and manipulator structure definition have been proposed. A possible approach consists in adopting the differential inverse kinematics, using the Jacobian matrix, for generating the manipulator trajectories [2, 3]. However, the algorithm must take into account the problem of kinematic singularities that may be hard to tackle. To avoid this problem, other algorithms for the trajectory generation are based on the direct kinematics [4–8].

Chen and Zalzala [2] propose a *GA* method to generate the position and the configuration of a mobile manipulator. The authors study the optimization of the least torque norm, the manipulability, the torque distribution and the obstacle avoidance, through the inverse kinematics scheme. Davidor [3] also applies *GA*s to the trajectory generation by searching the inverse kinematics solutions to pre

⋆ This paper is partially supported by the grant Prodep III (2/5.3/2001) from FSE.

defined end effector robot paths. Kubota *et al.* [4] study a hierarchical trajectory planning method for a redundant manipulator with a virus-evolutionary *GA* running simultaneously, two processes. One process calculates some manipulator collision-free positions and the other generates a collision free trajectory by combining these intermediate positions. Rana and Zalzala [5] developed a method to plan a near time-optimal, collision-free, motion in the case of multi-arm manipulators. The planning is carried out in the joint space and the path is represented as a string of via-points connected through cubic splines. Chocron and Bidaud [9] proposes an evolutionary algorithm to perform a task-based design of modular robotic systems. The system consists in a mobile base and an arm that may be built with serially assembled links and joints modules. The optimization design is evaluated with geometric and kinematic performance measures. Kim and Khosha [10] presents the design of a manipulator that is best suited for a given task. The design consists of determining the trajectory and the length of a three degrees of freedom (*dof*) manipulator. Han *et al* [11] describe a design method of a modular manipulator that uses the kinematic equations to determine the robot configuration and, in a second phase, adopts a *GA* to find the optimal length.

Multi-objective techniques using *GA*s have been increasing in relevance as a research area. In 1989, Goldberg [12] suggested the use of a *GA* to solve multi-objective problems and since then other investigators have been developing new methods, such as multi-objective genetic algorithm (*MOGA*) [13], non-dominated sorted genetic algorithm (*NSGA*) [14] and niched Pareto genetic algorithm (*NPGA*) [15], among many other variants [16].

In this line of thought, this paper proposes the use of a multi-objective method to optimize a manipulator trajectory. This method is based on a *GA* adopting the direct kinematics. The optimal manipulator front is one that minimizes both the path trajectory length and the energy required to perform a task, without any collision with the obstacles in the workspace. Following this introduction, the paper is organized as follows: section 2 formulates the problem and the *GA*-based method for its resolution. Section 3 presents several simulations results involving different robots, objectives and workspace settings. Finally, section 4 outlines the main conclusions.

2 Problem and algorithm formulation

This study considers robotic manipulators that are required to move from an initial point up to a given final configuration. In the experiments are used two and three *dof* planar manipulators (*i.e.* $2R$ and $3R$ robots) with link lengths of one meter and rotational joints free to rotate 2π *rad*. To test a possible collision between the manipulator and the obstacles, the arm structure is checked in order to verify if it is inside any obstacle. The trajectory consists in a set of strings representing the joint positions between the initial and final robot configurations.

2.1 Representation

The path for a iR manipulator ($i = 2, 3$), at generation T, is directly encoded as vectors in the joint space to be used by the GA as:

$$[\{q_1^{(\Delta t,T)}, .., q_i^{(\Delta t,T)}\}, \{q_1^{(2\Delta t,T)}, .., q_i^{(2\Delta t,T)}\}, .., \{q_1^{((n-2)\Delta t,T)}, .., q_i^{((n-2)\Delta t,T)}\}] \tag{1}$$

where i is the number of dof and Δt the sampling time between two consecutive configurations.

The values of joints $q_l^{(j\Delta t,0)}$ ($j = 1, \ldots, n-2$; $l = 1, \ldots, i$) are randomly initialized in the range $]-\pi, +\pi]$ *rad*. It should be noted that the initial and final configurations have not been encoded into the string because they remain unchanged throughout the trajectory search. Without losing generality, for simplicity, it is adopted a normalized time of $\Delta t = 0.1$ *sec*, because it is always possible to perform a time re-scaling.

2.2 Operators in the multi-objective genetic algorithm

The initial population of strings is generated at random. The search is then carried out among these populations. The three different operators used in the genetic planning are selection, crossover and mutation, as described in the sequel.

In what concerns the selection operator, the successive generations of new strings are reproduced on the basis of a Pareto ranking [12] with $\sigma_{\text{share}} = 0.01$ and $\alpha = 2$. To promote population diversity the count metric is used. This metric uses all solutions in the population independently of their rank to evaluate every fitness function. For the crossover operator it is used the simulated binary crossover (SBX)[14]. After crossover, the best solutions (among the parents and children) are chosen to form the next population. The mutation operator replaces one gene value with a given probability using the equation:

$$q_i^{(j\Delta t,T+1)} = q_i^{(j\Delta t,T)} + N(0, 1/\sqrt{2\pi}) \tag{2}$$

at generation T, where $N(\mu, \sigma)$ is the normal distribution function with average μ and standard deviation σ.

2.3 Evolution criteria

Three indices $\{q, p, E_a\}$ (3) are used to qualify the evolving trajectory robotic manipulators. Some of these criteria are used by the planner to find the optimal Pareto front. Before evaluating any solutions all the values such that $|q_i^{(j+1,T)} - q_i^{(j,T)}| > \pi$ are eliminated from the strings.

$$q = \sum_{j=1}^{n} \sum_{l=1}^{i} \left(\dot{q}_l^{(j\Delta t,T)}\right)^2 \tag{3a}$$

$$p = \sum_{j=2}^{n} d\left(p_j,\ p_{j-1}\right)^2 \tag{3b}$$

$$E_a = (n-1)T\ P_a = \sum_{j=1}^{n}\sum_{l=1}^{i} |\tau_l.\Delta q_l^{(j\Delta t,T)}| \tag{3c}$$

The joint distance q (3a) is used to minimize the joints manipulator traveling distance. For a function $y = g(x)$ the curve length is $\int[1 + (\mathrm{d}g/\mathrm{d}t)^2]\mathrm{d}x$ and, consequently, to minimize the distance curve length it is adopted the simplified expression $\int(\mathrm{d}g/\mathrm{d}t)^2\mathrm{d}x$. The cartesian distance p (3b) minimizes the total arm trajectory length, from the initial point up to the final point, where p_j is the robot j intermediate arm cartesian position and $d(\cdot,\cdot)$ is a function that gives the distance between the two arguments. Finally, the energy in expression E_a (3c), where τ_l are the robot joint torques, is computed assuming that power regeneration is not available by motors doing negative work, that is, by taking the absolute value of the power.

3 Simulation results

In this section results of various experiments are presented. In this line of thought, subsections 3.1 and 3.2 show the optimization of trajectories for the $2R$ and $3R$ robots respectively, for two objectives ($2D$). Finally, subsection 3.3 shows the results of a three dimensional ($3D$) optimization for a $2R$ robot.

3.1 $2R$ Robot trajectory with $2D$ optimization

The experiments consist on moving a $2R$ robotic arm from the starting configuration defined by the joint coordinates $A \equiv \{-1.149, 1.808\}$ *rad* up to the final configuration $B \equiv \{1.181, 1.466\}$ *rad* in a workspace without obstacles. The objectives used in this section are joint distance q (3a) and cartesian distance p (3b).

The simulations results achieved by the GA, with a $T_t = 15000$ generations and $pop_{\text{size}} = 300$ strings, converge to two optimal fronts. One of the fronts (fig. 1(a)) corresponds to the movement of the manipulator around its base in the clockwise direction. The other front (fig. 1(b)) is obtained when the manipulator moves in the counterclockwise direction. The solutions a and b represent the best solution found for a given objective.

In 71.4% of the total number of runs the algorithm the Pareto optimal front was found. In all simulations of the two cases the solutions converged to a front type which can be modelled approximately by the equation ($\kappa, \alpha, \beta \in \mathbb{R}$):

$$p(q) = \kappa\frac{q+\alpha}{q+\beta} \tag{4}$$

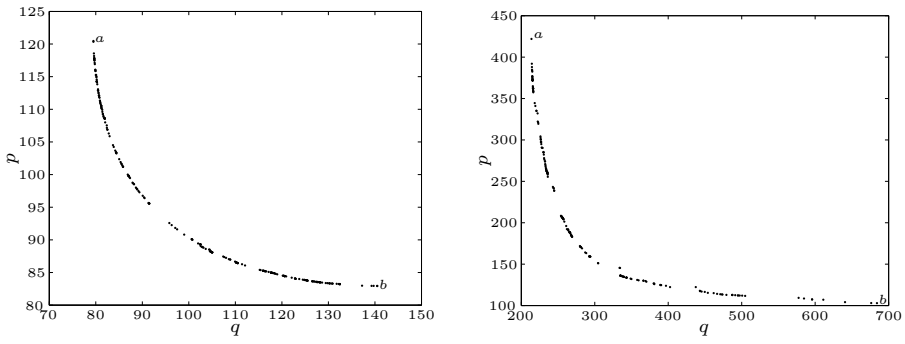
The achieved median, average and standard variation for the parameters κ, α and β of (4) are shown in table 1, both for the Pareto optimal and local fronts.

Table 1. Statistics of the fronts parameters

	Pareto front			Local front		
	κ	α	β	κ	α	β
Median	77.80	−66.31	−70.74	82.50	34.73	−173.02
Average	77.76	−66.16	−70.71	83.27	29.88	−173.09
Standard Deviation	0.53	1.05	0.70	2.63	20.87	3.41

To study the diversity of the front solution, the approximated front was split into several intervals, limited by normal straight lines r_m (fig. 2), such that the front length is identical for all intervals. For any two consecutive normal straight lines is associated an interval I_m ($m = 1, \ldots, 19$), and the solutions located between these lines are counted. Figure 3 shows the solution distribution achieved by one simulation run. From the chart, it can be seen that the solutions are distributed by all intervals. However, the distribution is not uniform. This is due to the use of a sharing function in the parameter domain in spite of the objective domain. Moreover, the algorithm does not incorporate any mechanism to promote the development of well distributed solutions in the objective domain.

The results obtained for solutions a and b, of the Pareto optimal front in figure 1(a), are presented in figures 4 to 6. Comparing figures 4(a) and 6(a) with figures 4(b) and 6(b) it is clear that the joint/cartesian distance for the optimal solutions a and b, respectively, is significantly different due to the objective considered. Between these extreme optimal solutions several others were found, that have a intermediate behavior, and which can be selected according with the importance of each objective.



(a) Pareto optimal front (b) local optimal front

Fig. 1. Optimal fronts for the $2R$ robot

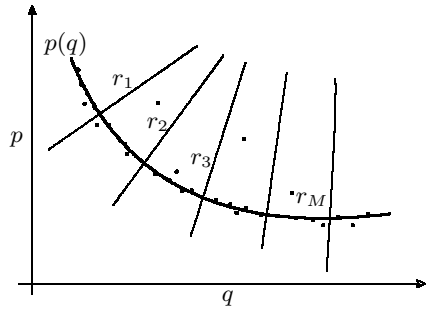


Fig. 2. Normal straight lines to the front obtained with $p(q)$ function

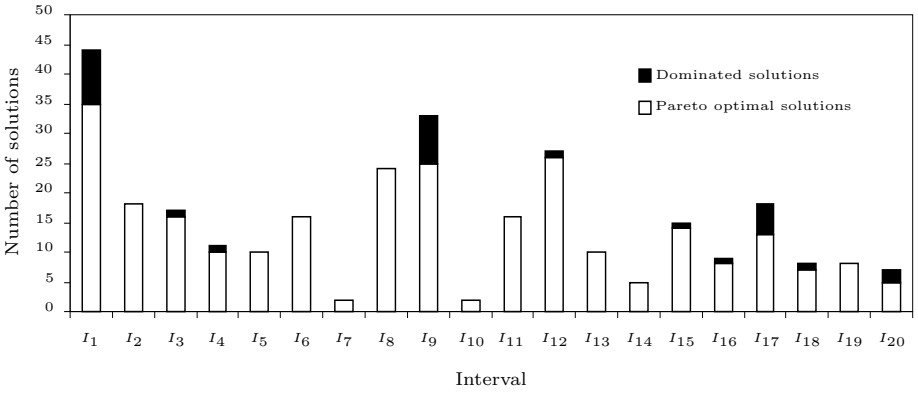


Fig. 3. Solution distribution achieved in one experiment for the $2R$ robot

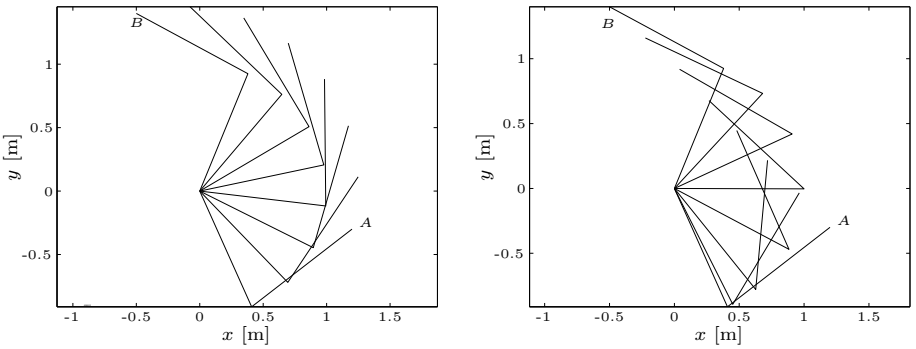
3.2 $3R$ Robot trajectory with $2D$ optimization

In this subsection a $3R$ robot trajectory is optimized in a workspace which may include a circle obstacle with center at $(x, y) = (2, 2)$ and radius $\rho = 1$. The initial and final configurations are $A \equiv \{-1.15, 1.81, -0.50\}$ *rad* and $B \equiv \{1.18, 1.47, 0.50\}$ *rad*, respectively. The T_t and pop_{size} parameters used are identical to those adopted in the previous subsection. The trajectories witch collide with the obstacle are assigned a very high fitness value.

For an optimization without any obstacle in the workspace is obtained the $f_2 = \widehat{ab}$ front (fig. 7). However, when the obstacle is introduced the front is reduced to the $f_1 = \widehat{cd}$. Thus, only the objective q if affected by the introduction of the obstacle (figures 8 and 9). The solutions a to d represent the best solution found for a given objective experiment.

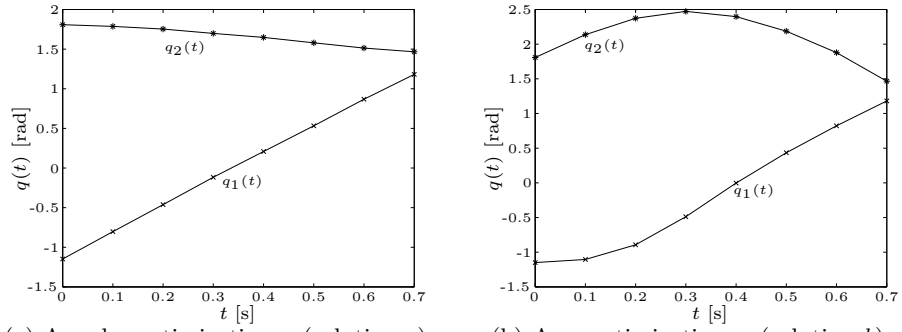
3.3 $2R$ Robot trajectory with $3D$ optimization

Here, the $2R$ manipulator trajectory is optimized considering three objectives, namely the joint distance, the cartesian distance and the energy required by the



(a) Angular optimization, q (solution a) (b) Arm optimization, p (solution b)

Fig. 4. Successive $2R$ robot configurations



(a) Angular optimization, q (solution a) (b) Arm optimization, p (solution b)

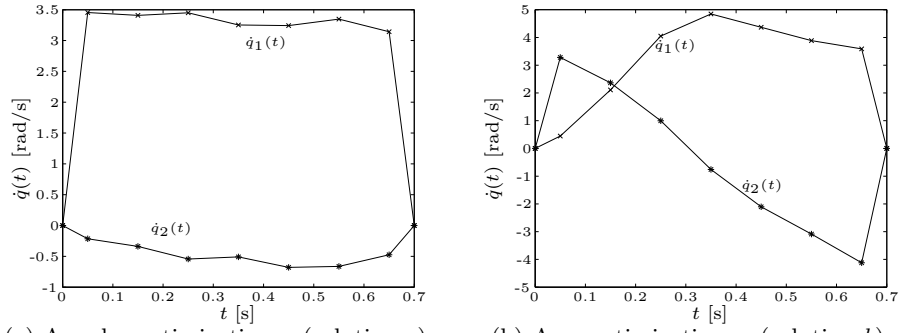
Fig. 5. Joint positions versus time for the $2R$ robot

manipulator to perform the task. Figure 10(a) shows the optimization results achieved with $T_t = 30000$ generations and $pop_{\text{size}} = 600$ string.

Figure 10 shows the $\{q, p\}$, $\{q, E_a\}$ and $\{p, E_a\}$ planner projections of the $3D$ optimization. Additionally, in each figure is included the correspondeding $2D$ optimization obtained previously.

4 Summary and conclusions

A multi-objective *GA* robot trajectory planner, based on the kinematics, was proposed. The algorithm is able to reach a determined goal with a reduced ripple both in the space trajectory and the time evolution. Moreover, any obstacles in the workspace do not represent a difficulty for the algorithm to reach the solution. Since the *GA* uses the direct kinematics the singularities do not constitute a problem. Furthermore, the algorithm determines the robot non-dominated front in order to the given objectives, maintaining a good solution distribution along



(a) Angular optimization, q (solution a) (b) Arm optimization, p (solution b)

Fig. 6. Joint velocities versus time for the $2R$ robot

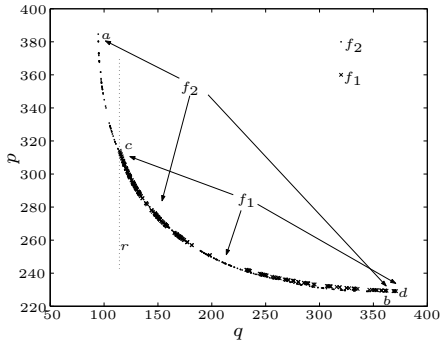
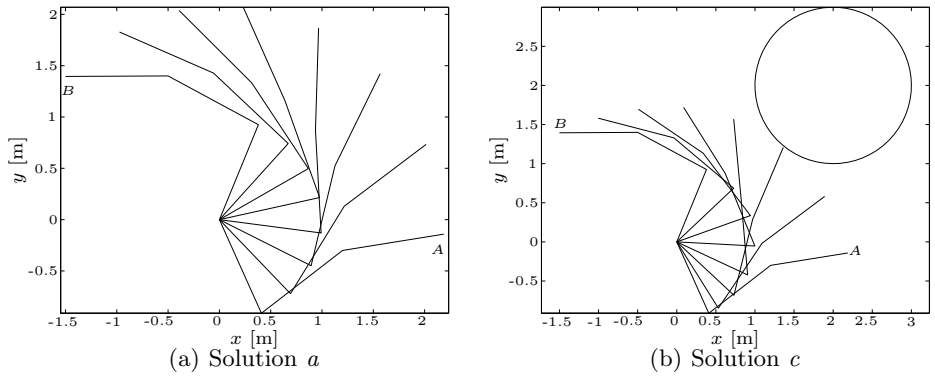


Fig. 7. Pareto optimal fronts, angular distance *vs.* cartesian distance optimization: $f_1 = \widehat{ab}$ – workspace without obstacles; $f_2 = \widehat{cd}$ – workspace with one obstacle

the front. The results shows that the algorithm reaches the Pareto optimal front or a very close one.

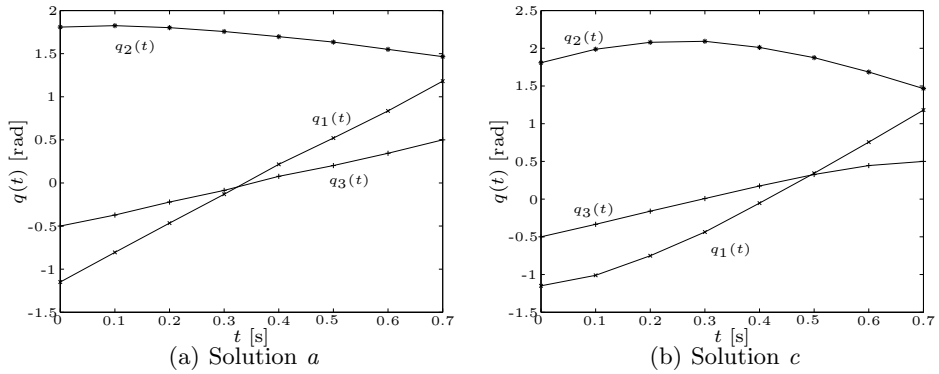
References

1. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: Comments on the history and current state. IEEE Trans. on Evolutionary Comp. **1** (1997) 3–17
2. Chen, M., Zalzala, A.M.S.: A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration. Journal Robotic Systems **14** (1997) 529–544
3. Davidor, Y.: Genetic Algorithms and Robotics, a Heuristic Strategy for Optimization. World Scientific (1991)
4. Kubota, N., Arakawa, T., Fukuda, T.: Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm, Albuquerque, New Mexico, IEEE Int. Conf. on Robotics and Automation (1997) 205–210



(a) Solution a (b) Solution c

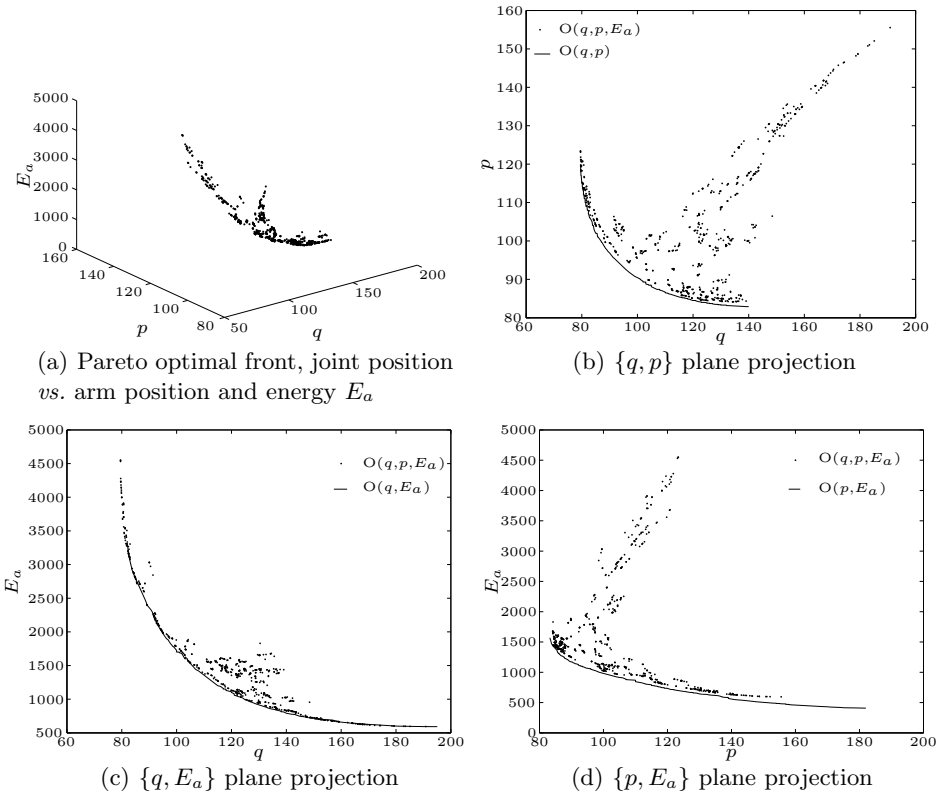
Fig. 8. Successive $3R$ robot configurations



(a) Solution a (b) Solution c

Fig. 9. Joint position of trajectory *vs.* time for the $3R$ robot

5. Rana, A., Zalzala, A.: An evolutionary planner for near time-optimal collision-free motion of multi-arm robotic manipulators. Volume 1., UKACC International Conference on Control (1996) 29–35
6. Wang, Q., Zalzala, A.M.S.: Genetic control of near time-optimal motion for an industrial robot arm, Minneapolis, Minnesota, IEEE Int. Conf. On Robotics and Automation (1996) 2592–2597
7. Pires, E.S., Machado, J.T.: Trajectory optimization for redundant robots using genetic algorithms. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2000), Las Vegas, USA, Morgan Kaufmann (2000) 967
8. Pires, E.J.S., Machado, J.A.T.: A GA perspective of the energy requirements for manipulators maneuvering in a workspace with obstacles. In: Proc. of the 2000 Congress on Evolutionary Computation, Piscataway, NJ, (2000) 1110–1116
9. Chocron, O., Bidaud, P.: Evolutionary algorithms in kinematic design of robotic system, Grenoble, France, IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems (1997) 279–286
10. Kim, J.O., Khosla, P.K.: A multi-population genetic algorithm and its application to design of manipulators, Raleight, North Caroline, IEEE/RSJ Int. Conf. on



(a) Pareto optimal front, joint position *vs.* arm position and energy E_a

(b) $\{q, p\}$ plane projection

(c) $\{q, E_a\}$ plane projection

(d) $\{p, E_a\}$ plane projection

Fig. 10. Pareto optimal front $\{q, p, E_a\}$ and Pareto optimal front plane projections: $\{q, p\}$, $\{q, E_a\}$ and $\{p, E_a\}$

Intelligent Robotics and Systems (1992) 279–286
11. Han, J., Chung, W.K., Youm, Y., Kim, S.H.: Task based design of modular robotic manipulator using efficient genetic algorithm, Albuquerque, New Mexico, IEEE Int. Conf. on Robotics and Automation (1997) 507–512
12. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison – Wesley (1989)
13. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation Journal **3** (1995) 1–16
14. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley-Interscience Series in Systems and Optimization. (2001)
15. Horn, J., Nafploitis, N., Goldberg, D.: A niched pareto genetic algorithm for multi-objective optimization, Proc. of the First IEEE Conf. on Evolutionary Computation (1994) 82–87
16. Coello, C., Carlos, A.: A comprehensive survey of evolutionary-based M.-O. optimization techniques. Knowledge and Information Systems **1** (1999) 269–308