

Multi-Criteria Manipulator Trajectory Optimization Based on Evolutionary Algorithms

E. J. Solteiro Pires, P. B. de Moura Oliveira, J. A. Tenreiro Machado

Abstract This paper proposes a method, based on a genetic algorithm, to generate smooth manipulator trajectories in a multi-objective perspective. The method uses terms proportional to the integral of the squared displacements in order to eliminate the jerk movement. In this work, the algorithm, based on NSGA-II and maximin sorting schemes, considers manipulators of two, three and four rotational axis ($2R$, $3R$, $4R$). The efficiency of the algorithm is evaluated, namely the extension of the front and the dispersion along the front. The effectiveness and capacity of the proposed approach are shown through simulations tests.

1 Introduction

Genetic algorithms (GAs) are one of the most popular evolutionary inspired search and optimization technique. This popularity is shown by the large number of successful applications in many scientific areas [1]. One of the advantages of GAs over classical techniques is that it can be adopted in optimization applications without requiring specific knowledge about the working problem. Initially, it was mainly used in single-objective problems. However, in few years became clear that GAs could be applied in multi-objective optimization. Taking advantage of using a population, GA can determine a set of non-dominated solutions in just one execution of the algorithm [2, 3, 4, 5]. Moreover, GAs are less susceptible to the Pareto front shape

E. J. Solteiro Pires and P.B. de Moura Oliveira
Centro de Investigação e de Tecnologias Agro-Ambientais e Biológicas, Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro, Quinta de Prados, 5000–801 Vila Real, Portugal, e-mail: {epires,oliveira}@utad.pt

J. A. Tenreiro Machado
Instituto Superior de Engenharia do Porto, Instituto Politécnico do Porto, 4200–072 Porto, Portugal, e-mail: jtm@isep.ipp.pt

or continuity than classical optimization techniques. Due to these factors, multi-objective evolutionary algorithms (MOEAs) have become increasingly popular in a vast number of areas. For example in electrical engineering, hydraulics, robotics, control scheduling, physics, medicine and computer science [6].

In robotics, the problems of trajectory planning, collision avoidance and manipulator structure design considering a single criteria has been solved using several techniques [7, 8, 9, 10]. However, trajectory planning adopting multiple objectives was somewhat overlooked and only a few articles analyzing this topic can be found. Pires *et al.* [11] proposed a MOEA to optimize a manipulator trajectory considering multiple objectives, namely: space and joint arm displacements and the energy required to perform the route without colliding with the obstacles. Ramabalan *et al.* [12] proposed two MOEAs to generate a manipulator trajectory with multiple objectives. In their work, they compare the results obtained by different algorithms. Liu *et al.* [13] considers the planning of a space manipulator taking account the joint angle, joint velocity and torque constraints. They use a weighted fitness function where the weights are randomly selected. They decompose the trajectory considering several segments. In each segment a suitable polynomial is used. The inter-knots parameters (angle, velocity and torque) of each trajectory segment are optimized by the proposed MOEA.

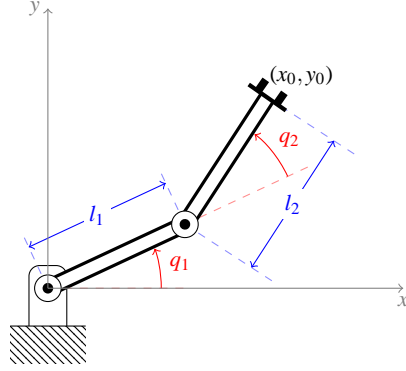
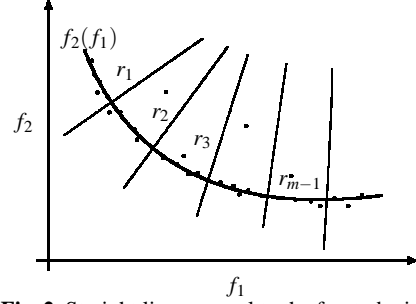
In this line of thought, this paper proposes the use of a multi-objective method to optimize the trajectory of manipulators with 2, 3 and 4 rotational degrees of freedom (dof). The method is based on the NSGA-II and the maximin sorting scheme [14] adopting the direct kinematics. The non-dominated trajectories are those that minimize the joint and gripper displacement. In a second phase, non-dominated solutions are analyzed in order to measure the solution spread along the front.

The paper is organized as follows. Section 2 introduces the problem and the GA based scheme for its resolution. Sections 3 describes the method to measure the solution spread along the Pareto front. Based on this formulation, section 4 presents the results for several simulations involving 2, 3, and 4 link manipulators. Finally, section 5 outlines the main conclusions.

2 Problem formulation

This work considers robotic manipulators with 2, 3 and 4 links which are required to move between two coordinates (Figure 1) in the operational space. The work considers two objectives, namely the joint displacement (O_q) and the gripper displacement (O_p). It is intended to determine a representative set of non-dominated solutions belonging to the Pareto optimal front. To measure the quality of the algorithm, the solutions spread along the front is analyzed. The decision maker chooses the solution taking into account the compromise between the objectives that he finds more appropriate.

The experiments consist on moving a iR robotic arm, $i = \{2, 3, 4\}$, between configurations defined by the points $A \equiv \{1.2, -0.3\}$ and $B \equiv \{-0.5, 1.4\}$. To find out

**Fig. 1** Two joint (link) manipulator (2R robot)**Fig. 2** Straight lines normal to the front obtained

the initial and final 2R manipulator configurations the inverse kinematics is used. For the 3R and 4R manipulators, the initial joint values are determined to obtain a configuration close to the one observed for the 2R robot. In the work, the rotational joints are free to rotate 2π rad.

The GA parameters are: population size $pop_{dim} = 300$, number of generations $T_t = 1500$; crossover and mutation probabilities $p_c = 0.6$ and $p_m = 0.05$, respectively, link length $l = 2/i$ [m] and mass of $m = 2/i$ [Kg].

The two indices $\{O_q, O_p\}$ presented in (1) quantify the quality of the evolving trajectories of the robotic manipulators. The indices represent the joint displacement, O_q (1a), and the Cartesian gripper displacement O_p (1b):

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

$$O_p = \sum_{j=2}^n d(p_j, p_{j-1})^2 \quad (1b)$$

The path for a iR manipulator ($i = 2, 3, 4$), at generation T , is directly encoded as a string in the joint space to be used by the GA. This string is represented by expression (2), where i represents the number of dof and Δt the sampling time between two consecutive configurations. Therefore, one potential solution is encoded as:

$$[\{q_1^{(\Delta t, T)}, \dots, q_i^{(\Delta t, T)}\}, \dots, \{q_1^{(2\Delta t, T)}, \dots, q_i^{(2\Delta t, T)}\}, \dots, \{q_1^{((n-2)\Delta t, T)}, \dots, q_i^{((n-2)\Delta t, T)}\}] \quad (2)$$

where the joints values $q_l^{(j\Delta t, 0)}$, $j = 1, \dots, n-2$; $l = 1, \dots, i$, are randomly initialized in the range $[-\pi, +\pi]$ [rad]. The robot movement is described by $n = 8$ configurations. However, the initial and final configurations are not encoded into the string, because they remain unchanged throughout the trajectory search. Without losing

generality, for simplicity, it is adopted a normalized time of $\Delta t = 0.1$ s, but it is always possible to perform a time re-scaling.

The proposed algorithm is based on the NSGA-II [3]. The individual solution fitness takes into account all the neighboring solutions independently of their rank (sharing parameters are $\sigma = 0.01$ and $\alpha = 2$). Moreover, the maximin sorting scheme is adopted [14], replacing the crowding distance used within NSGA-II, at the end of each iteration, to determine the progenitors which will be part of the next population.

3 Method to measure the solution distribution and the extension of the front

In this section a method to determine the distribution of solutions along the Pareto optimal front and the extension of the front are described.

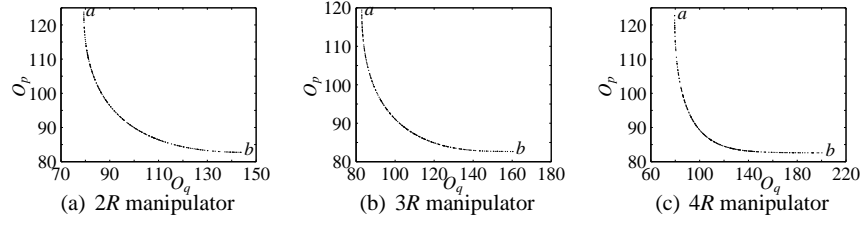
The method begins by finding a function that models the Pareto optimal front in the appropriate range. This function should be valid between the extreme solutions obtained by the MOEA. Next, the function adopted to represent the Pareto front is divided through normal straight lines (figure 2). Between each two consecutive normal straight lines (r_i and r_{i+1}) a range I_i is defined. Finally, all solutions located in a specific range are counted. The dispersion is given by the solution number of the ranges.

The extension of the front is measured through the curve length of the modeled function taking into account the two closest points to the two extreme solutions of the Pareto front found by the MOEA.

4 Results

This section develops several tests. For each type of test multiple experiments are performed to achieve $n_{\text{exp}} = 21$ valid simulations. This means that many distinct experiments are executed until 21 successful convergences to the optimal Pareto front are obtained, for the 2R, 3R and 4R manipulators. Figure 3 depicts one optimal Pareto front for each manipulator type. The convergence rates were 95%, 57% and 38% for the 2R, 3R and 4R manipulators, respectively. It can be observed that as the number of links increases the convergence rates decrease. This is due to the exponential increase of the number of local fronts, with the number of robot links.

In all cases, the fronts can be modeled by equation (3) with the parameter set $\{\kappa, \alpha, \beta\} \in \mathbb{R}$. For each front obtained in a valid simulation, the parameters are estimated. Next, the median, mean and standard deviation of the parameters κ , α and β are calculated (Table 1). The front modelling procedure through equation 3 is only valid between the non-dominated extreme solutions a and b .

**Fig. 3** Pareto optimal fronts.**Table 1** Statistical parameters for the modeled fronts

	2R robot front			3R robot front			4R robot front		
	κ	α	β	κ	α	β	κ	α	β
Median	77.90	-66.83	-71.21	78.68	-71.36	-75.12	80.28	-71.82	-74.44
Mean	77.99	-66.74	-71.13	78.76	-71.33	-75.19	80.15	-71.49	-74.34
Std Dev.	0.44	0.71	0.42	0.58	1.67	1.22	0.60	2.21	1.62

$$O_p(O_q) = \kappa \frac{O_q + \alpha}{O_q + \beta} \quad (3)$$

From Table 1 it can be seen that the mean and the median values are almost similar. Additionally, the values of the standard deviation are relatively small, leading to the conclusion that the algorithm always converges to the same front, which is likely to be the Pareto optimal front. Moreover, with the increasing of the manipulator number of links the standard deviation increases. By other words, as the number of links increases the problem complexity becomes higher and the algorithm has more difficulty to converge to the same non-dominated front.

The extension (*i.e.*, the length) of the front for the iR robot manipulators, $i = \{2, 3, 4\}$, has an average $\mu_{\text{Ext}} = \{86.57, 105.61, 200.49\}$ and a standard deviation $\sigma_{\text{Ext}} = \{2.00, 17.48, 48.94\}$. It can be concluded that, with the increasing problem complexity, the algorithm has more difficulty in obtaining always the same front extension.

The solution diversity along the non-dominated front is presented in figures 4-6. The final percentage average of solution number belonging to the non-dominated front is $\mu_{\text{Div}} = \{96.15\%, 92.53\%, 88.25\%\}$ and the standard deviation is $\sigma_{\text{Div}} = \{1.26, 5.21, 4.28\}$. The figures reveal that the solutions are distributed over all intervals. However, this distribution is not uniform and the uniformity decreases as the number the manipulator links increases. This phenomena occurs because the solutions percentage of the non-dominated front decreases with the number of links. This is due to the fact that the algorithm favors non-dominated solutions, and only then enters into account with diversity. Consequently, the proposed algorithm reveals its potential only when all population elements are within the non-dominated front. Figures 4-6 also show that regions with less non-dominated solutions are com-

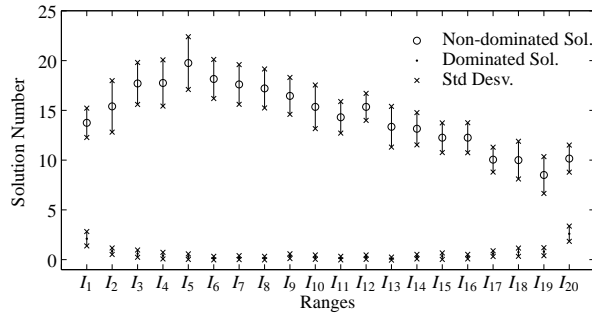


Fig. 4 The 2R robot solution distribution along the Pareto front

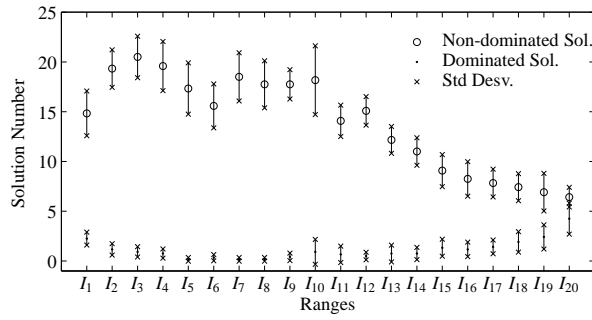


Fig. 5 The 3R robot solution distribution along the Pareto front

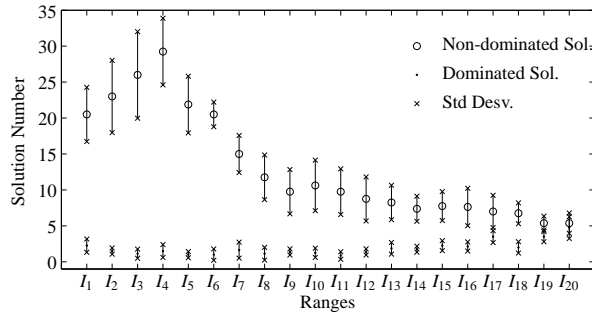


Fig. 6 The 4R robot solution distribution along the Pareto front

compensated by the algorithm with more dominated solutions, in order to keep a good solution distribution in all intervals.

Figure 7 depicts the extreme solutions, a and b , corresponding to the optimal fronts Pareto illustrated in Figure 3. The figure includes the successive configurations and the angular displacements for the 2R, 3R and 4R manipulators.

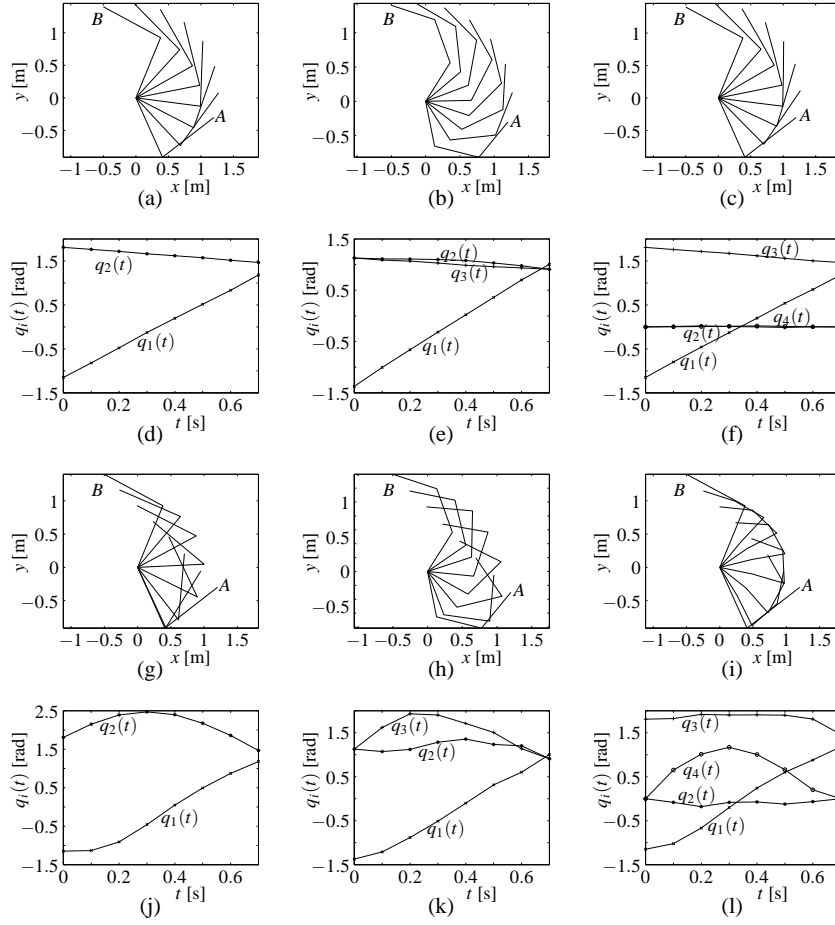


Fig. 7 Pareto optimal trajectories. Successive configurations of solution *a* (Figures 3(a)-3(c)) for robot: (a) 2R, (b) 3R, (c) 4R. Successive configurations of solution *b* for robot: (g) 2R, (h) 3R, (i) 4R. Joint position trajectory vs. time of solution *a* for robot: (d) 2R, (e) 3R, (f) 4R. Joint position trajectory vs. time of solution *b* for robot: (j) 2R, (k) 3R, (l) 4R.

5 Conclusions

This paper solves the manipulator trajectory planning problem in a multi-objective perspective. This work considered manipulators with two, three and four rotational joints with trajectories solved by an evolutionary algorithm based on NSGA-II and maximin sorting schemes. To study the efficiency of the algorithm the extension of the front and the dispersion along the front were carried out. The results show that it is possible obtain different solutions according to the weight of the objectives. Moreover, with only one execution of the GA, it was possible to obtain a consider-

able number of non-dominated solutions with good diversity along the front. Each solution represents a possible manipulator trajectory yielded by the weights of the objectives envisaged by the decision maker.

References

1. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: Comments on the history and current state. *IEEE Trans. on Evolutionary Computation* **1**(1) (April 1997) 3–17
2. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation Journal* **3**(1) (1995) 1–16
3. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, LTD (2001)
4. Horn, J., Nafploitis, N., Goldberg, D.: A niched pareto genetic algorithm for multi-objective optimization, *Proceedings of the First IEEE Conference on Evolutionary Computation* (1994) 82–87
5. Coello, C.A.C.: A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems* **1**(3) (August 1998) 269–308
6. Coello Coello, C.: Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE* **1**(1) (Feb. 2006) 28–36
7. Davidor, Y.: *Genetic Algorithms and Robotics, a Heuristic Strategy for Optimization*. Number 1 in Series in Robotics and Automated Systems. World Scientific Publishing Co. Pte Ltd (1991)
8. Kubota, N., Fukuda, T., Shimojima, K.: Trajectory planning of cellular manipulator system using virus-evolutionary genetic algorithm. *Robotics and Autonomous systems* **19** (1996) 85–94
9. Luo, X., Wei, W.: A new immune genetic algorithm and its application in redundant manipulator path planning. *Journal of Robotic Systems* **21**(3) (2004) 141–151
10. Ridao, M.A., Camacho, E.F., Riquelme, J., Toro, M.: An evolutionary and local search algorithm for motion planning of two manipulators. *Journal of Robotic Systems* **18**(8) (2001) 463–476
11. Solteiro Pires, E.J., de Moura Oliveira, P.B., Tenreiro Machado, J.A.: Multi-objective genetic manipulator trajectory planner. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: *EvoWorkshops*. Volume 3005 of *Lecture Notes in Computer Science*, Springer (2004) 219–229
12. Ramabalan, S., Saravanan, R., Balamurugan, C.: Multi-objective dynamic optimal trajectory planning of robot manipulators in the presence of obstacles. *The International Journal of Advanced Manufacturing Technology* **41**(5-6) (2009) 580–594
13. Liu, Z., Huang, P., Yan, J., Liu, G.: Multi-objective genetic algorithms for trajectory optimization of space manipulator. In: *4th IEEE Conference on Industrial Electronics and Applications*, 2009. ICIEA 2009, Xi'an, China (25-27 May 2009) 2810–2815
14. Solteiro Pires, E.J., de Moura Oliveira, P.B., Tenreiro Machado, J.A.: Multi-objective Max-iMin Sorting Scheme. In Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E., eds.: *Evolutionary Multi-Criterion Optimization*. Third International Conference, EMO 2005, Guanajuato, México, Springer. *Lecture Notes in Computer Science* Vol. 3410 (March 2005) 165–175