

A Multi-Objective Trajectory Optimization Algorithm for 7-DOF Robotic Arms

Chenxi Zhao

Nanjing University of Information Science & Technology, Nanjing, China
202283320100@nuist.edu.cn

Abstract: With the rapid advancement of robotics technology, robotic arms are increasingly required to execute precise tasks, such as industrial handling and human-robot collaboration. This work propose a novel multi-objective trajectory planning approach to ensure smooth and efficient motion for a 7-DOF robotic arm. Specifically, our method employs inverse kinematics to transform the trajectory from workspace to joint space, followed by a seventh-order B-spline function to generate a smooth trajectory. Then, we introduce Trajectory Competition Two-phase Framework Combined with NSGA-II (TTToP) to optimal this trajectory, which solve the Pareto optimal solution set of the trajectory based on the robot constraints and select the optimal solution according to the task requirements. Experimental results demonstrate that the proposed method effectively generates high-quality trajectories that meet practical energy and time constraints.

Keywords: multi-objective optimization, robotic arm, Evolutionary Algorithm, NSGA II.

1. Introduction

With the advancement of robotics technology, 7-degree-of-freedom (7-DOF) robotic arms, also called redundant robots, have gained attention due to their increased flexibility. Unlike traditional 6-DOF robotic arms, 7-DOF robots have one more degree of freedom, allowing them to perform more complex and adaptable operations in constrained environments. This redundancy enables better obstacle avoidance, improved manipulable, and optimized workspace utilization. However, efficient trajectory interpolation and optimization methods are needed to fully leverage these advantages.

Trajectory interpolation generates intermediate points between way-points to ensure smooth motion. Common methods include linear interpolation [1], polynomial interpolation [2] (e.g., cubic, quintic), and spline interpolation [3] (e.g., B-spline, Bézier curves). While these methods ensure smooth transitions, they often ignore dynamic constraints, leading to high joint torques or abrupt velocity changes. Trajectory optimization refines trajectories by considering factors like energy efficiency and execution time. For example, Liu et al. [4] focuses on energy-optimal trajectory, while Shen et al. [5] considers minimum jerk optimization.

However, existing research primarily focus on specific-objective trajectory planning, which is struggle to meet the demands of practice 7-DOF robots trajectory planning. As a result, some researchers began to pay attention to multi-objective trajectory planning. For example, Gasparetto et al. [6] propose hybrid time-energy optimization approaches, where multiple objectives (e.g., execution time and energy consumption) are integrated into a single objective function through weighting coefficients. However, this approach is prone to falling into the trap of local optimal

solutions. Lan et al. [7] proposes an effective method for robot trajectory planning, which uses multi-objective particle swarm optimization algorithm to solve the optimal trajectory. However, this method focuses on 6-DOF robots and is difficult to be directly applied to 7-DOF robots.

In this paper, we propose a simple and effective multi-objective trajectory planning method for 7-DOF robots. We employ a high-order B-spline interpolation method instead of traditional interpolation methods to ensure smoother planned trajectories. To optimize the trajectory interpolation, we propose Trajectory Competition Two-phase Framework Combined with NSGA-II (TToP), which can identify Pareto optimal solutions for robot trajectory and select the solution that meets the requirements. The experimental results demonstrate that the method enables the robotic arm to perform effectively.

2. Kinematics analysis of robotic arm

As shown in Fig. 1, we use the modified Denavit–Hartenberg (DH) method [8] to build the kinematic model, which provide a foundation for analyzing the robot's motion and behavior.

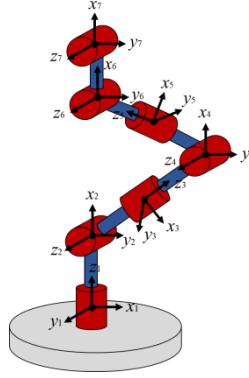


Figure 1: 7-DOF robotic arm

Then, the kinematic transformation matrix is derived by applying the improved DH parameters as

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^3_4T = \\
 &\begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^4_5T &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & d_5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^5_6T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^6_7T = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & -d_7 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)
 \end{aligned}$$

where ${}^i_{i+1}T$ is the matrix between links i and $i+1$, s_i represents $\sin\theta_i$, and c_i represents $\cos\theta_i$.

The transformation matrix is multiplied by the next transformation matrix,

$${}^0_7T = {}^0_1T(\theta_1){}^1_2T(\theta_2){}^2_3T(\theta_3){}^3_4T(\theta_4){}^4_5T(\theta_5){}^5_6T(\theta_6){}^6_7T(\theta_7) \quad (2)$$

and the forward kinematics equation is expressed as:

$${}^0_7T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The approach in [9] is applied to multiply and invert the transformation matrices. It enables the calculation of the joint coordinates. Given the end-effector pose, only θ_4 can be uniquely determined, while the other joint angles $\theta_i (i = 1, 2, 3, 5, 6, 7)$ are transformed into an expression with arm angle γ as the parameter.

$$\begin{aligned} \theta_1 &= \arccos \left(\frac{x_4}{\sqrt{x_4^2 + y_4^2}} \right) \theta_2 = \arccos \left(\frac{z_4 - d_1}{d_3} \right) \theta_3 = \arccos \left(\frac{\bar{v}_{31} \cdot \bar{v}_{32}}{|\bar{v}_{31}| \cdot |\bar{v}_{32}|} \right) \theta_4 = \\ &\arccos \left(-\frac{d_3^2 + d_5^2 - |O_2 O_6|^2}{2d_3 d_5} \right) \\ \theta_5 &= \arccos \left(\frac{\bar{v}_{51} \cdot \bar{v}_{52}}{|\bar{v}_{51}| \cdot |\bar{v}_{52}|} \right) \theta_6 = \arccos \left(-\frac{d_5^2 + d_\gamma^2 - |O_4 O_\gamma|^2}{2d_5 d_\gamma} \right) \theta_7 = \arccos \left(\frac{Pr_{31} + Qr_{32}}{P^2 + Q^2} \right), \end{aligned} \quad (4)$$

where

$$\begin{cases} P = -s_2 c_3 c_4 c_5 c_6 + s_2 c_3 s_4 s_6 + s_2 s_3 c_5 s_6 - c_2 s_4 c_5 c_6 - c_2 c_4 s_6 \\ Q = s_2 c_3 c_4 s_5 + s_2 s_3 c_5 + c_2 s_4 s_5 \end{cases} \quad (5)$$

where O_i is the center point of the i -th joint.

3. Multi-objective trajectory planning

In this section, we first analyze the problem and then employ B-spline interpolation to construct smooth joint trajectories. To address the multi-objective optimization challenges, we propose a novel TToP method. This method incorporates dynamic constraints and an improved crowding distance calculation to ensure the generation of feasible and diverse Pareto solution sets. Within the set, a normalized weight function is used to adjust objective priorities, thereby generating optimal solutions that meet practical requirements.

3.1. Problem Statement

The motion path of the end-effector, represented as T_i , is calculated when moving from its initial position to the desired target position. Then, the position-time sequence is

$$Q = \{(q_{ij}, t_i) | i \in [0, n], j \in [1, N]\} \quad (6)$$

where q_{ij} denotes the position of joint j and time i , n denotes the number of time nodes, N denotes the number of joints.

To ensure smooth transitions between trajectory points, a specialized curve is employed for interpolation. We have defined a trajectory function and established optimization objectives: total movement time, acceleration, and jerk. They align with the optimization goals in the key areas: efficiency, energy consumption, and smoothness. The problem is defined as

$$P1: \min_i \sum_{i=1}^n \Delta t_i \quad (7)$$

$$P2: \min_i \sum_{j=1}^N \sqrt{\frac{1}{T} \int_0^T a_i^2 dt} \quad (8)$$

$$P3: \min_i \sum_{j=1}^N \sqrt{\frac{1}{T} \int_0^T \tau^2 dt} \quad (9)$$

$$|v_{ij}| \leq v_{max} \quad (10.1)$$

$$|a_{ij}| \leq a_{max} \quad (10.2)$$

$$|j_{ij}| \leq \tau_{max} \quad (10.3)$$

Where v_{max} denotes the maximum velocity, a_{max} denotes the acceleration, jerk τ_{max} and time interval Δt_i as optimization variable, and v_{ij} , a_{ij} , j_{ij} are velocity, acceleration and jerk of joint j and time i .

3.2. Trajectory Planning

To design the joint trajectory for the manipulator, B-spline curve [10] is employed. In the process of trajectory interpolation, the junction points between curve segments are precisely aligned with the nodes of the B-spline curve to ensure that the starting and ending points coincide exactly with the initial and final data points, respectively. We use the method in [7] to normalize the time nodes t to obtain internal node values, i.e.,

$$u_i = u_{i-1} + \left| \Delta t_{i-k-1} \right| / \sum_{j=0}^{n-1} \Delta t_j, i \in [k+1, n+k-1] \quad (11)$$

where $\Delta t_k = t_{k+1} - t_k$ is chordal length between consecutive points.

Then, the equations that satisfy the interpolation conditions are expressed as

$$p(u_{i+k}) = \sum_{j=i}^{i+k} d_j N_{j,k}(u_{i+k}) = P_i \quad (12)$$

where $u_{i+k} \in [u_k, u_{n+k}]$, $i = 0, 1, \dots, n$.

To determine the control vertices d_i , $k-1$ additional conditions are necessary. For a seven-order B-spline curve, we introduce six tangent boundary conditions. Furthermore, the convex hull property of the B-spline ensures that the constraints on the control vertices align with v_{jmax} , a_{jmax} , and j_{jmax} .

$$\begin{cases} \max \{ |d_{ij}^1| \} \leq v_{max} \\ \max \{ |d_{ij}^2| \} \leq a_{max} \\ \max \{ |d_{ij}^3| \} \leq j_{max} \end{cases} \quad (13)$$

where d_{ij}^1 , d_{ij}^2 and d_{ij}^3 are the control vertices for velocity, acceleration, and jerk curves of joint j at point i .

Through this process, we calculate the control vertices and enable the completion of the B-spline trajectory interpolation. This approach ensures smooth and dynamically feasible joint trajectories for the robotic arm.

3.3. Trajectory competition two-phase framework combined with NSGA-II

To tackle the challenges in trajectory optimization, it is crucial to develop a multi-objective optimization approach that can generate a Pareto-optimal solution set [11]. However, existing methods often fail to incorporate the specific constraints of robot trajectory planning, limiting their ability to produce targeted optimal solutions. To address this gap, we introduce TToP, a novel multi-objective evolutionary algorithm that combines the Two-phase Optimization (ToP) strategy with NSGA-II [12]. Originally designed for multi-objective optimization, ToP employs a two-stage strategy to balance conflicting objectives. By harnessing its staged optimization and multi-objective trade-off capabilities, we adapt ToP for robot trajectory optimization, enabling the generation of smooth and stable motion trajectories for robotic arms.

TToP integrates feasibility rules and kinematic constraints to ensure that the generated solutions are not only optimal but also physically feasible, aligning with the real-world motion limitations of robotic systems. Dynamic constraints are embedded into the objective function to enforce physical feasibility, while an enhanced crowding distance calculation method is introduced to improve population diversity. This approach effectively avoids local optima and enhances the algorithm's global optimization performance, making it well-suited for complex robotic applications.

To further enhance the flexibility of the optimization process, we introduce a normalized weight function to balance the trade-offs between objectives. The weighted objective is defined as a linear combination of normalized objective values, scaled to the interval $[0,1]$, and multiplied by customizable weight coefficients. If higher efficiency is needed, the value of σ_1 can be increased. Conversely, if lower energy consumption and greater smoothness are desired, larger values can be assigned to σ_2 and σ_3 .

$$f = \sigma_1 \frac{P1 - P1_{\min}}{P1_{\max} - P1_{\min}} + \sigma_2 \frac{P2 - P2_{\min}}{P2_{\max} - P2_{\min}} + \sigma_3 \frac{P3 - P3_{\min}}{P3_{\max} - P3_{\min}} \quad (14)$$

where σ_1 , σ_2 and σ_3 are the weight coefficients, and $P1_{\min}$, $P2_{\min}$, $P3_{\min}$, $P1_{\max}$, $P2_{\max}$, $P3_{\max}$ are the minimum and maximum values of objective $P1$, $P2$ and $P3$.

Core steps of applying TToP for trajectory optimization are outlined in Algorithm 1.

Algorithm 1 Trajectory Competition Two-phase Framework Combined With NSGA-II(TToP)

- 1: **Input:**
 - Population size N
 - Crossover and mutation rates
 - Number of generations M
 - Weighting Factors w
 - 2: **Output:** Pareto optimal P
 - 3: Initialize population P .
 - 4: Evaluate the objective functions and rank the solutions using dominance and non-dominance relations.
 - 5: Use Non-dominated Sorting to sort the population, and employ the Fast Non-dominated Sorting method. These solutions are ranked based on their dominance relationships, resulting in multiple fronts.
 - 6: Choose solutions as parents using a uniform selection process.
 - 7: Apply crossover and mutation operators to the selected parents to produce an offspring population Q_0 .
-

- 8: Assess the new offspring by evaluating their performance based on the defined objective functions.
 - 9: **If** $|P_{t+1}| + |F_i| \geq N$.
 - 10: Calculate crowding distance in F_i .
 - 11: Sort $(F_i, < n)$.
 - 12: $P_{t+1} = P_{t+1} \cup F_i [1 : (N - |P_{t+1}|)]$.
 - 13: **If** stopping condition is satisfied.
 - 14: Terminate the algorithm.
 - 15: **Else** back to step 4.
 - 16: Use a normalized weighted function to adapt to specific task requirements.
-

4. Experiments

4.1. Experimental setup

To validate the effectiveness of the proposed approach, we utilized MATLAB to model the robotic arm and plotted the corresponding experimental results. As illustrated in Table 1, we use a practice joint space trajectory for this experiment.

Table 1: Joints space trajectory

Node	Joint 1(°)	Joint 2(°)	Joint 3(°)	Joint 4(°)	Joint 5(°)	Joint 6(°)	Joint 7(°)
1	16.17	-24.64	34.55	29.34	88.57	-18.77	31.56
2	22.54	-10.97	37.65	59.58	110.84	-31.67	35.67
3	24.18	-25.77	61.52	71.27	99.63	-14.27	47.86
4	17.04	-31.66	69.72	77.33	100.11	-10.01	35.88
5	15.22	-33.62	59.63	66.89	88.04	11.55	50.41
6	10.18	-49.57	69.89	78.55	86.12	14.58	54.88
7	-10.888	-38.74	60.74	55.66	69.78	25.27	44.66
8	-25.78	-40.34	82.17	47.80	78.16	13.55	64.43

4.2. Multi-objective optimization

We select time intervals along the trajectory as the key variables for optimization after the joint trajectory curve is generated. Then, that algorithm identifies the most suitable values to achieve optimal performance. These intervals are bounded, with an upper limit of six and a calculated lower limit. The population size is 100 and the maximum evolutionary generations number is 10,000.

Fig. 2 presents the Pareto front derived from the proposed algorithm. Solutions near point P show improved energy and jerk performance but reduced time efficiency, while solutions closer to point Q achieve better time performance at the cost of energy and jerk efficiency. This indicates a positive correlation between energy efficiency and jerk performance, whereas there exists a negative correlation between energy efficiency and temporal performance.

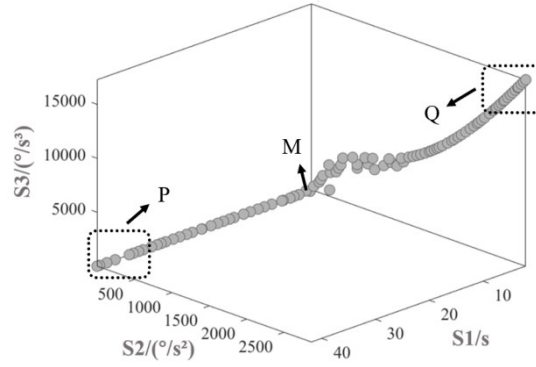


Figure 2: Pareto front of the algorithm

Both P and Q lie near the constraint boundaries, and the proposed algorithm effectively covers the entire solution space. In practical robot operations, a balanced consideration of all three objectives is necessary to determine the optimal solution. For specific tasks, the optimal values must simultaneously satisfy predefined maximum ranges.

4.3. Trajectory planning

To prioritize stability and smoothness while maintaining operational efficiency, the point M is selected as the optimal solution. We use this parameter to generate trajectories on a 7-DOF robot.

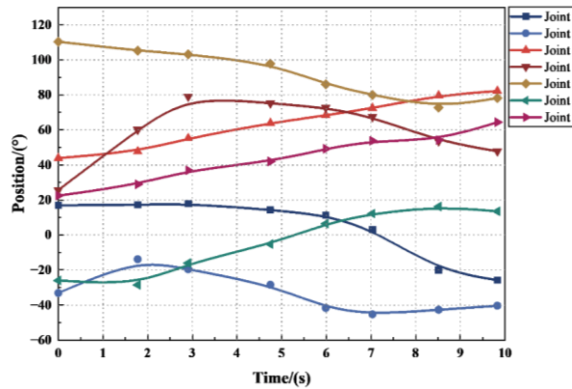
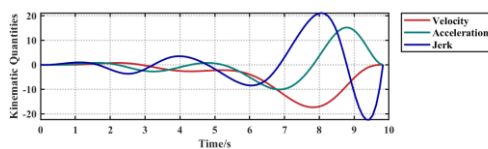


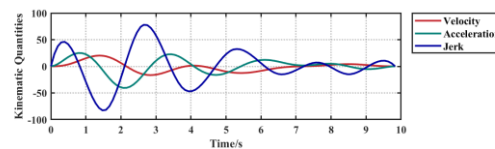
Figure 3: The trajectory sequence of each joint

As shown in Fig.3, the trajectory position of each joint are smooth and continuous. The time nodes are [0, 1.8001, 2.9112, 4.7471, 6.0139, 7.0355, 8.5227, 9.9107] and the optimal objective values are [9.8199, 8.0386, 39.8993].

Fig. 4 present the velocity, acceleration, and jerk of each joint. The results show that the velocity, acceleration, and jerk are also smooth and continuous, and their ranges align with the robot's kinematic constraints. Moreover, the velocity, acceleration, and jerk values at the start and end points are zero, in order to ensure smooth operation and prevent sudden torque changes.



(a) Joint 1



(b) Joint 2

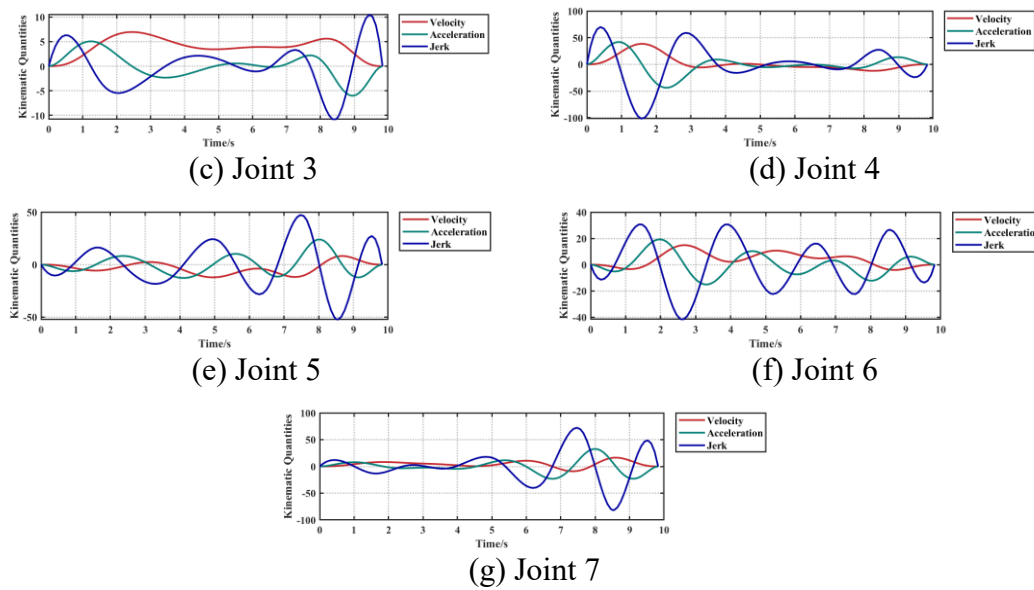


Figure 4: The trajectory of each joint

5. Conclusions

This study introduces a multi-objective trajectory planning approach for robotic arms. We use seventh-order B-splines, continuous and smooth motion trajectories are generated. The proposed TToP algorithm is then applied to derive the Pareto optimal solution set. To adapt to varying application needs, a normalized weight function is implemented for solution selection. Experimental results have demonstrated that the trajectory of the robot is smooth and stable, which can be widely used in robotic arms and has provided a new idea for the trajectory optimization algorithm. For future work, we plan to extend this method to more complex robotic systems and explore its application in dynamic environments with real - time constraints.

References

- [1] de Boor, C. (1978). *A practical guide to spline*. <https://doi.org/10.2307/2006241>
- [2] Guan, Y., Yokoi, K., Stasse, O., & Kheddar, A. (2005). *On robotic trajectory planning using polynomial interpolations*. 2005 IEEE International Conference on Robotics and Biomimetics - ROBIO, Hong Kong, China, 111-116. <https://doi.org/10.1109/ROBIO.2005.246411>
- [3] Fritsch, F. N., & Carlson, R. E. (1980). *Monotone piecewise cubic interpolation*. SIAM Journal on Numerical Analysis, 17(2), 238-246.
- [4] Liu, S., & Sun, D. (2011). *Optimal motion planning of a mobile robot with minimum energy consumption*. 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Budapest, Hungary, 43-48. <https://doi.org/10.1109/AIM.2011.6027010>
- [5] Shen, Y., Li, Y., Shen, Y., et al. (2023). *Trajectory optimization algorithm for agricultural UAV operations based on improved minimum snap*. Transactions of the Chinese Society of Agricultural Engineering, 39(17), 51-59. <https://doi.org/10.11975/j.issn.1002-6819.202212003>
- [6] Gasparetto, A., & Zanotto, V. (2007). *A new method for smooth trajectory planning of robot manipulators*. Mechanism and Machine Theory, 42(4), 455-471.
- [7] Lan, J., Xie, Y., Liu, G., & Cao, M. (2020). *A multi-objective trajectory planning method for collaborative robot*. Electronics, 9(5), 859.
- [8] Sanz, P. (2009). *Robotics: Modeling, planning, and control* (Siciliano, B. et al; 2009) [On the shelf]. IEEE Robotics & Automation Magazine, 16(4), 101-101. <https://doi.org/10.1109/MRA.2009.934833>
- [9] Paul, R. P., & Shimano, B. (1978). *Kinematic control equations for simple manipulators*. 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, San Diego, CA, USA, 1398-1406. <https://doi.org/10.1109/CDC.1978.268148>

- [10] Wang, H.-F., & Z.S. (2012). *INSGA-II based multi-objective trajectory planning for manipulators*.
- [11] Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). *PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum]*. *IEEE Computational Intelligence Magazine*, 12(4), 73-87. <https://doi.org/10.1109/MCI.2017.2742868>
- [12] Eftekharian, S., Shojafar, M., & Band, S. (2017). *2-Phase NSGA II: An optimized reward and risk measurements algorithm in portfolio optimization*. *Algorithms*, 10(4), 130. <https://doi.org/10.3390/a10040130>