Crimson Publishers
Wings to the Research

Mini Review

Multi-Objective Optimization for Collaborative Robot Working Environment

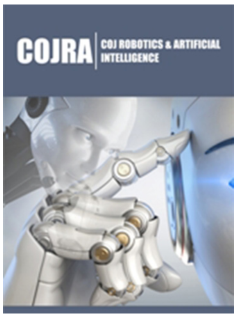
Sivakumar R[1], John Alexis S[2]* and Manickavasagam A[3]

[1]Department of Computer Science and Engineering, Ahalia School of Engineering and Technology, India

[2]Department of Mechanical Engineering, Ahalia School of Engineering and Technology, India

[3]Department of Electrical and Electronics Engineering, Ahalia School of Engineering and Technology, India

ISSN: 2832-4463



***Corresponding author:** John Alexis S, Department of Mechanical Engineering, Ahalia School of Engineering and Technology, Kerala 678557, India

Submission: September 19, 2023
Published: November 21, 2023

Volume 3- Issue 3

How to cite this article: Sivakumar R, John Alexis S* and Manickavasagam A. Multi-Objective Optimization for Collaborative Robot Working Environment. COJ Rob Artificial Intel. 3(3). COJRA. 000564. 2023. DOI: 10.31031/COJRA.2023.03.000564

Abstract

Multi-objective Optimization algorithms like Swarm Intelligence Algorithms are becoming more useful in complex, dynamic collaborative robot working environments. These environments allow humans and robots to work securely and productively. Collaborative robot behaviour is aimed to be optimised via swarm intelligence techniques. Social insect-inspired swarm intelligence algorithms are robust and adaptive to multi-objective Optimization approaches. They enjoy exploring the solution space and balance the competing goals. These algorithms let collaborative robots balance task productivity, safety, human-robot interaction, energy utilization, and other related issues. This paper is aimed to build swarm intelligence algorithms for multi-objective Optimization in a collaborative robot environment. Define objectives, formulate the problem, choose the algorithm, encode and decode robot configurations, fitness assignment, initialization, iterative Optimization, convergence, and post-processing discoveries. Swarm intelligence algorithms help collaborative robots navigate their workplace, adapt, and safely interact with humans. Decision-makers can choose Pareto front or non-dominated solutions for optimal trade-offs. Validation and simulation enable Pareto-optimal and safe and collaborative robot behaviours. Fine-tuning algorithm parameters enhance performance and convergence. Collective intelligence solves complex, dynamic multi-objective Optimization issues. As collaborative robots transform many industries, swarm intelligence algorithms will shape safe, efficient, and productive human-robot interactions.

Keywords: Swarm Intelligence; Multi-objective Optimization approaches; Collaborative robots; Multi-objective Optimization; Pareto-optimal; Particle Swarm Optimization

Introduction

Multi-objective Optimization (MOO) is a powerful and versatile field that optimizes many conflicting objectives. Multi-objective Optimization seeks trade-off solutions rather than an ideal one. Cost reduction, efficiency, safety, and profit are examples. Decision-makers often face real-world challenges when maximizing one objective is not enough or may harm other areas. A lighter, less durable device may cost more in engineering design [1]. Multi-objective Optimization offers solutions that balance all objectives in such cases. Multi-objective Optimization uses Pareto optimality. Pareto-optimal methods improve one goal without hurting another. The trade-off region between objectives is the Pareto front, or Pareto optimal solutions. Multi-objective Optimization problems are hard to solve due to the huge search space and complex trade-offs. Multi-objective Optimization is effective. GA, PSO, EA, ACO, and others optimize multi-objectively. Fitness assignment, elitism, and dominance relations identify Pareto optimum solutions in these algorithms. Engineering, finance, logistics, resource allocation, portfolio Optimization, and more use multi-objective Optimization. It optimizes technical design, controls, and parameters. Multi-objective Optimization enhances financial portfolio management and risk analysis. It optimizes logistics and supply chain management. Innovative cobots, or collaborative robots, work alongside people in shared offices. Collaborative cobots are safer and more human-friendly, allowing humans and machines to work together. Collaborative robots were created to fulfil the demand for flexible and adaptable automation technologies that can operate with people.

Traditional robots work alone and require substantial safety measures. Collaborative robots improve workplaces by working with people. Collaborative robot working environments must prioritize safety [2]. These robots recognize humans and avert accidents through force/torque sensors, collision detection, and speed decrease. Collaboration robots are intuitive. Non-experts can programme and control them. This simplifies deployment and integration, enabling human-robot collaboration. Collaborative robots are flexible. Their fast reprogramming and deployment enhance small-batch production and flexible manufacturing. Lightweight, small, and transportable robots reduce operator strain [3]. They reduce repetitive strain injuries in strenuous work. Humans and machines can collaborate. The robot's precision and repeatability improve the operator's decision-making and cognition, enhancing productivity and efficiency. Collaborative robots perform mundane tasks, so that people may focus on higher-value tasks like product quality and speed increase [4]. Manufacturing, logistics, healthcare, agriculture, research, and even homes use collaborative robots. They assemble, inspect, and help with healthcare. Collaborative robots allow human-robot collaboration. They improve global industry efficiency, adaptability, and creativity by working safely and efficiently with humans. Collaborative robots will improve productivity and peace in the future [5].

Multi-objective optimization for a collaborative robot in a working environment

Multi-objective Optimization for a collaborative robot in a workplace ensures safe and efficient collaboration with people and other robots. To reduce human-robot collisions, dangerous movements, and compliance with safety standards and laws, safety limits should be addressed during Optimization. The robot should reduce energy, resources, and task completion time. Optimizing the robot's path, mobility, and work distribution could enhance productivity. The robot's conduct should be designed for smooth and intuitive collaboration with humans. This includes predictable, natural actions, clear communication, and avoiding surprises. The optimizer should delegate jobs to several robots based on their capabilities and the system's aims. Coordinating task execution and resource sharing requires multi-robot communication to optimize data and transmission overhead. To optimize for workplace change, the robot should adapt to new challenges. NSGA-II, NSGA-III, SPEA2, and MOEA/D optimize many objectives simultaneously. These algorithms solve objective trade-offs Pareto-optimally. Dynamic collaboration demands real-time Optimization. The robot should quickly adapt to changing conditions to maintain efficiency and safety. User preferences or utility functions let operators or supervisors optimize. This ensures solutions match stakeholder needs.

Implementation

Multi-objective Optimization methods like swarm intelligence algorithms for collaborative robot work require numerous steps. Process overview:

A. Specify the various collaborative robot working environment objectives that is required to optimize like Safety, task efficiency, human-robot interaction, energy consumption, etc.

B. Model multi-objective Optimization. Define collaborative robot environment limitations, decision variables, and objective functions.

C. Select a swarm intelligence algorithm for multi-objective Optimization. PSO, ACO, GA, and DE are common evolutionary algorithms. Evaluate the problem and each algorithm.

D. Use functions to evaluate each candidate solution (robot behaviour/configuration) based on the given objectives. Swarm intelligence algorithm fitness functions will be these.

E. Encode decision variables for the swarm intelligence algorithm and decode them back to the robot's configuration during Optimization. This phase corrects the Optimization algorithm's robot parameters.

F. Use the swarm intelligence algorithm to award fitness scores to each candidate solution based on their performance across several objectives. The algorithm should favour Pareto-front non-dominated solutions.

G. Create a random or well-distributed swarm of candidate solutions within the viable range.

H. Optimize the swarm intelligence algorithm iteratively. Each iteration, the swarm updates candidate solution positions based on fitness and dynamics.

I. Set an algorithm stopping criterion, such as a maximum number of iterations or convergence, after meeting the requirement, the algorithm stops.

J. Analyze algorithm-converged solutions. Extract the Pareto front or a representative subset of non-dominated solutions that represent objective trade-offs.

K. Test the results in a collaborative robot environment. Based on the chosen answers, assess the robot's performance.

To increase Optimization performance and convergence time, fine-tune algorithm parameters including population size, convergence criteria, and inertia weights for PSO. Swarm intelligence methods for multi-objective Optimization in a collaborative robot working environment must consider the problem, objectives, and robot behaviour. Validate the optimised solutions to ensure they meet the collaborative environment's safety and efficiency requirements.

Improved particle swarm optimization algorithm for collaborative robot environment

An improved Particle Swarm Optimization (PSO) algorithm is a version of the regular PSO algorithm that has been made better and bigger. Its goal is to get around the problems that come with the normal PSO method and make it better at solving Optimization problems. According to the standard Particle Swarm Optimization (PSO) method, this is a population-based Optimization method that is based on how birds flock or fish school. Using this method is based on the idea that particles move through a solution space to find the best answer. In Improved Particle Swarm Optimization (PSO) algorithms, extra features, changes, or strategies are added to improve the performance of PSO in a variety of situations. Possible improvements to a better Particle Swarm Optimization (PSO) method could be included.

The use of dynamic inertia weight schemes might help find a good mix between exploring and exploiting when it comes to controlling inertia weight. At different points in the planning process, the inertia weight is changed to fit the properties of the search space. Constriction factors are very important for making the Particle Swarm Optimization (PSO) method more stable and helping it settle. By constantly searching the areas around the best sites, local search techniques built into search engines can help people who get stuck in local optima. The use of adaptive processes to change Particle Swarm planning (PSO) parameters, like cognitive and social weights, to improve convergence during the planning process is known as adaptive parameter tuning. The main topic of this study is how to make Particle Swarm Optimization (PSO) work better for problems with more than one goal.

This method makes it possible to find Pareto front solutions by letting you work on multiple goals at the same time. The focus of this study is on handling constraints, especially coming up with methods for doing this in a way that makes sure the algorithm can come up with possible solutions. When you combine Particle Swarm Optimization (PSO) with other Optimization methods or strategies, you can take advantage of their strengths and work around their weaknesses. This is called hybridization. The study's goal is to show methods that make it easier to remember and handle past successful solutions, so that they can guide the swarm's exploration process. Topology Variations is looking into different neighbourhood topologies that go beyond the usual global and local topologies. These include structures like the ring, star, and fully connected topologies. Adaptive swarm size means that the size of a swarm changes on the fly during the Optimization process based on how much progress has been made in achieving agreement. This research is mainly about dynamic environment adaptation, more specifically coming up with ways for particles to change how they act when the problem space changes. Improvements to Stability and Robustness: Adding ways to deal with problems like early convergence, slow convergence, or swings that happen during the search process.

Improved Particle Swarm Optimization (PSO) methods are made to work with certain types of problems and might be able to give better Optimization abilities. How the additions and changes are chosen depends on the specifics of the Optimization problem at hand as well as the speed gains that are wanted. Researchers and practitioners often try out different methods to see which ones work best for their specific uses.

Pseudocode

Initialize PSO parameters:

population_size, max_iterations, inertia_weight,cognitive_weight, social_weight

- robot_positions[] // Current positions of collaborative robots

- robot_velocities[] // Current velocities of collaborative robots

- pBest_positions[] // Best-known positions for each robot

- gBest_position // Global best-known position

- task_assignments[] // Assignment of tasks to robots

Initialize collaborative robot positions and tasks:

- Assign initial positions and tasks to robots

- Evaluate the performance using an objective function

Initialize pBest_positions and gBest_position:

- Set pBest_positions[robot] and gBest_position based on the initial evaluations

for iteration in max_iterations:

for each robot in population:

Update robot_velocities[robot] using the PSO equations:

- new_velocity = inertia_weight * current_velocity

+ cognitive_weight * rand() * (pBest_positions[robot] - robot_positions[robot])

+ social_weight * rand() * (gBest_position - robot_positions[robot])

Update robot_positions[robot] using the new velocities:

- robot_positions[robot] = robot_positions[robot] + new_velocity

Update task_assignments[robot] based on the robot's position:

- Determine the tasks the robot can reach based on its position

- Update task_assignments[robot] accordingly

End of robot loop

Evaluate robot performance using an objective function:

- Consider task completion time, energy consumption, etc.

Update pBest_positions and gBest_position:

- If a robot's performance is better than its pBest_position:

- Update pBest_positions[robot]

- If any robot's performance is better than gBest_position:

- Update gBest_position

End of iteration loop

Return the gBest_position as the optimized task allocation for collaborative robots

It is possible that the Particle Swarm Optimization (PSO) method may need to have its parameters optimised and the pseudocode will need to be modified, to accommodate the specific requirements of your application. This includes overcoming restrictions, making it easier for robots to communicate with one another, adapting to dynamic changes in the surrounding environment, and taking precautions to prevent collisions.

Conclusion

The multi-objective Optimization collaborative hybrid with swarm intelligence algorithms is very effectively used to optimize the collaborative robot behaviour in dynamic working environments. These multi-objective Optimization techniques balance multiple opposing goals. Collaborative robot workplaces must balance safety, task efficiency, human-robot interaction, and other performance indicators. Swarm intelligence algorithms can search the vast solution space for Pareto optimal solutions that trade-off these objectives. Swarm intelligence algorithms can boost productivity, safety, and ergonomics in collaborative robots. These algorithms let robots adapt to environmental changes and adjust their behaviour in real time, making them ideal for agile manufacturing and flexible production. Swarm intelligence systems also aid decision-makers. The Pareto front or non-dominated set of options helps people understand objective trade-offs and pick actions that satisfy their needs. Successful implementation requires rigorous examination of the collaborative robot working environment's challenges and goals. Validation and simulation guarantee optimum behaviours meet safety and collaborative norms. Fine-tuning algorithm parameters speeds convergence and performance. Swarm intelligence algorithms will change human-robot collaboration as collaborative robots transform several industries. These algorithms optimize complex and dynamic activities to make robot-human interactions safe, efficient and productive.

References

1. Deb K (2014) Multi-objective optimization. In: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer, Boston, Massachusetts, USA, pp. 403-449.
2. Faccio M, Granata I, Menini A, Milanese M, Rossato C, et al. (2022) Human factors in cobot era: A review of modern production systems features. J Intell Manuf 34: 85-106.
3. Kanazawa A, Kinugawa J, Kosuge K (2019) Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency. IEEE Transactions on Robotics 35(4): 817-832.
4. Romero D, Stahre J (2021) Towards the resilient operator 5.0: The future of work in smart resilient manufacturing systems. Procedia CIRP 104: 1089-1094.
5. Zanchettin AM, Rocco P, Chiappa S, Rossi R (2019) Towards an optimal avoidance strategy for collaborative robots. Robotics and Computer-Integrated Manufacturing 59: 47-55.