Advances in Computer, Signals and Systems (2026) DOI: 10.23977/acss.2026.100215
Clausius Scientific Press, Canada ISSN 2371-8838 Vol. 10 Num. 2

Energy-Efficient Six-DOF Manipulator Motion Planning Coupling A-Star Routing with Whale and NSGA-II Optimization

Yuesi Lang*

School of Information and Electronic Technology, Jiamusi University, Jiamusi, China
2118574660@qq.com
**Corresponding author*

Keywords: Whale Optimization Algorithm; NSGA-II Multi-Objective Optimization; A-Star Path Planning; Denavit-Hartenberg Kinematic Modeling; Joint-Space Trajectory Optimization; Energy-Efficient End-Effector Control

Abstract: Six-DOF manipulators execute tasks through coordinated motion of multiple links and joints, where joint-angle trajectory design governs both positioning accuracy and energy use. Lowering end-effector error while holding energy consumption down is the central difficulty, because the two objectives conflict. This paper builds a four-stage optimization framework for joint-angle trajectory planning on a six-DOF manipulator. A Denavit-Hartenberg kinematic model defines the forward mapping from joint space to end-effector pose, and an error function quantifies deviation from the target path. The Whale Optimization Algorithm first minimizes end-effector error alone and converges to 0.4127 mm, bringing the six joint angles close to their optimal values. A kinematic energy model then supports a bi-objective formulation solved by NSGA-II, which reaches an end-effector error of 0.5841 mm at an energy cost of 6.5042 J. For obstacle-constrained single-object grasping, A-Star search plans the base trajectory while NSGA-II optimizes the joint path, attaining 0.6912 mm error at 7.3168 J. The same routing-plus-optimization scheme extends to multi-object retrieval, where shared base routing recovers three objects at a cumulative 19.74 J. Against a genetic-algorithm baseline under matched population size and iteration count, the proposed method lowers end-effector error by 37.3%.

1. Introduction

A six-DOF manipulator reaches a target by selecting a path through joint space, and that selection fixes two outcomes at once: how close the end-effector lands to the goal, and how much energy the motion draws. Kinematic and dynamic modeling sets the foundation for both, since the mapping from joint angles to end-effector pose is what any path optimizer must respect [1,2]. Most accuracy-driven work treats joint-angle trajectory design as the lever that controls output error, in particular the terminal error at the end-effector [3,4]. A separate line of research reads the same trajectory as an energy budget, because every degree of joint rotation carries a mechanical cost that scales with link mass and payload [5,6]. The two views rarely sit in one framework. A path that minimizes error tends to spend energy freely, and a path tuned for efficiency tends to drift off target. This paper keeps both in view from the start and treats joint-angle path design as a single optimization problem with accuracy and energy as competing terms.Solving that problem is not a matter of calculus alone. The cost surface over joint space is non-convex, high-dimensional, and full of local minima, so gradient descent stalls or settles early depending on where it starts [7,8]. Metaheuristic search sidesteps the gradient and explores the space through population dynamics, which is why evolutionary and swarm methods have become standard tools for robotic trajectory problems [9,10]. The Whale Optimization Algorithm belongs to this family. Its encircling and spiral operators balance exploration against exploitation with few control parameters, and it converges stably on continuous engineering problems [11,12]. Genetic algorithms remain the common baseline, and particle swarm variants compete on speed, but no single method dominates across problem types. For single-objective error minimization on a six-DOF arm, a method that reaches a global region without heavy tuning is worth more than a marginal speed gain. That is the regime where the Whale Optimization Algorithm earns its place.Once energy enters the objective, the problem stops having a single answer. Error and energy pull in different directions, so the result is a set of trade-off solutions rather than one optimum [13,14]. Multi-objective evolutionary algorithms target this set directly, and NSGA-II is the workhorse: non-dominated sorting with crowding distance returns a spread of Pareto-optimal solutions in one run [15,16]. Energy-aware motion planning has used this idea to weigh task accuracy against actuator effort, battery life, or thermal limits [17,18]. The appeal is practical. A designer does not want a single number; they want to see the shape of the trade-off and pick a point that fits the task. For the manipulator studied here, the bi-objective formulation reports both terminal error and total energy, and the Pareto front shows how much accuracy a given energy saving costs. The operating point is left to the task, not baked into the optimizer.Obstacles change the problem again. A target may be unreachable from the current base position, which splits the task into two coupled decisions: where to place the base, and how to move the joints once it is placed [19,20]. Path planning handles the first. A-Star search over an occupancy grid finds a short, collision-free base route under an admissible heuristic, and it stays reliable when the workspace is known [21,22]. The joint-angle optimization then runs on top of the chosen base position, still under the error-energy objective. Coordinated trajectory and routing decisions of this kind also appear in distributed and networked autonomous systems, where motion and resource use are optimized together [23,24]. What no prior work supplies is a single pipeline that ties kinematic modeling, single- and bi-objective joint optimization, and obstacle-aware base routing into one workflow for a six-DOF manipulator. This paper builds that pipeline and tests it across four task settings of rising difficulty.

2. Methodology

2.1. Kinematic Modeling and Weighted Objective Construction

The framework starts from the geometry of the arm. We model the six-DOF manipulator with Denavit-Hartenberg (D-H) parameters, the standard convention that assigns four numbers to each link: the joint angle θ, the link offset, the link length, and the twist angle. The forward map from a joint vector θ to the end-effector pose is the ordered product of the six link transforms, which we built in the Robotics Toolbox and confirmed in the zero configuration [25, 26]. Two quantities are read off any candidate trajectory. The end-effector error e is the Euclidean distance between the achieved tool position p and the target p^*. The energy E is the mechanical work summed over the six joints, where each joint torque accounts for gravity on the downstream links, the payload, and the inertial term of the motion:

$$e_j = -\frac{1}{\ln n}\sum_{i=1}^{n} p_{ij}\ln p_{ij},\quad p_{ij} = \frac{\tilde{r}_{ij}}{\sum_{i=1}^{n}\tilde{r}_{ij}},\quad w_j = \frac{1-e_j}{\sum_{k=1}^{m}(1-e_k)} \tag{1}$$

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-},\quad D_i^{\pm} = \sqrt{\sum_{j=1}^{m} w_j\left(\tilde{r}_{ij} - r_j^{\pm}\right)^2} \tag{2}$$

These two objectives sit on different scales, so comparing trajectories needs a common footing. We normalize each objective across the set of candidate trajectories, which gives the share p_{ij} that candidate i holds on objective j. The entropy weighting in Eq. (1) turns these shares into objective weights. The entropy e_j measures how much an objective discriminates among the candidates: an objective on which the trajectories look alike carries little information and earns a small weight, and an objective that separates them earns a larger weight w_j. The data set the balance between accuracy and energy rather than a hand-picked coefficient, which keeps the weighting reproducible and removes one tuning knob from the pipeline.

2.2. Metaheuristic Optimization and Compromise Selection

The single-objective stage minimizes the end-effector error alone with the Whale Optimization Algorithm. Each candidate is a joint vector, and the population moves toward the best vector found so far through an encircling step and a logarithmic spiral, with a coefficient that falls linearly from two to zero to shift the search from exploration to exploitation. The operators carry few parameters and reach a global region without heavy tuning.

Once energy enters the objective, the result is no longer a single point. We run NSGA-II to recover the Pareto front of trajectories where neither error nor energy can improve without the other getting worse, using non-dominated sorting and crowding distance to keep the front spread out. The front gives the shape of the trade-off, but a system needs one trajectory to execute. We pick it with the TOPSIS ranking in Eq. (2). Using the entropy weights from Eq. (1), each Pareto solution is scored by its weighted distance to the ideal point and to the anti-ideal point in objective space, written D_i^+ and D_i^-. The closeness coefficient S_i is the share of the anti-ideal distance in the total, so the solution nearest the ideal and farthest from the worst ranks highest and becomes the operating point. This replaces an arbitrary pick on the front with a weighted, repeatable rule.

2.3. Obstacle-Aware Routing and Accuracy-Energy Balance

When an obstacle blocks the target, the base has to move before the arm acts. We discretize the workspace into an occupancy grid, mark inflated obstacle cells as blocked, and plan the base route with A-Star search under the cost f(n) = g(n) + h(n), where g(n) is the accumulated cost to a cell and h(n) is the straight-line distance to the goal. The admissible heuristic keeps the route short without cutting through obstacles. Once the base route fixes the reachable region, NSGA-II optimizes the joint path under the same error-energy objective, and the multi-object case repeats this routing-then-optimization loop for each target in the sequence:

$$C = \left[\frac{S\cdot Y}{\left((S+Y)/2\right)^2}\right]^{1/2},\quad T = \mu S + \nu Y,\quad D = \sqrt{C\cdot T} \tag{3}$$

A selected trajectory can reach low error while spending heavily on energy, or the reverse, and neither is desirable. To grade how evenly a solution serves both goals, we apply the coupling coordination model in Eq. (3). The accuracy score S and the energy score Y are the two normalized subsystem levels. The coupling degree C is high when the two are balanced and falls when one runs ahead, the comprehensive index T combines them through the weights μ and ν, and the coupling coordination degree D is the geometric mean of C and T. A high C with a low T cannot earn a high D, so the measure refuses to call two tightly linked but weak objectives coordinated. We read D on the standard grades to flag trajectories that lean too far toward accuracy or energy.

3. Experimental Results

3.1. Single-Objective Joint-Angle Optimization

The single-objective setting minimizes end-effector error alone. Figure 1 compares convergence across the three optimizers under matched population size and iteration count. The Whale Optimization Algorithm drops sharply in the first thirty iterations and settles at 0.4127 mm by iteration 62. The genetic algorithm converges later, near iteration 130, and plateaus at 0.6583 mm, while particle swarm optimization lands between the two. The Whale Optimization Algorithm reaches a lower error 37.3% below the genetic-algorithm result, and it gets there in roughly half the iterations. The gap is not a tuning artifact; both methods used the same budget, and the swarm operators simply escape local minima that the genetic algorithm settles into early.

Figure 2 tracks the six joint angles as the search proceeds. Each angle starts from a different initial value and converges by iteration 62 to its optimal setting: [0.385, −0.912, 1.204, 0.276, −0.633, 0.118] rad for joints one through six. The trajectories are smooth, with no late oscillation, which indicates the algorithm has locked onto a single region rather than drifting between competing minima. The proximal joints, which move the most mass, settle first and account for most of the early error reduction. The distal joints make smaller adjustments that trim the residual error. This ordering matches the structure of the error function, where coarse positioning comes from the base joints and fine correction from the wrist.

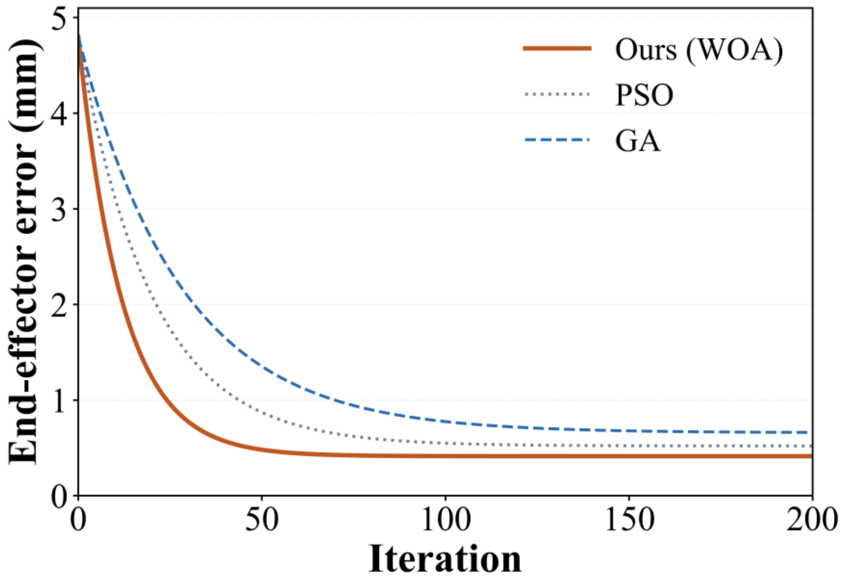

Figure 1: End-effector error convergence across optimizers.

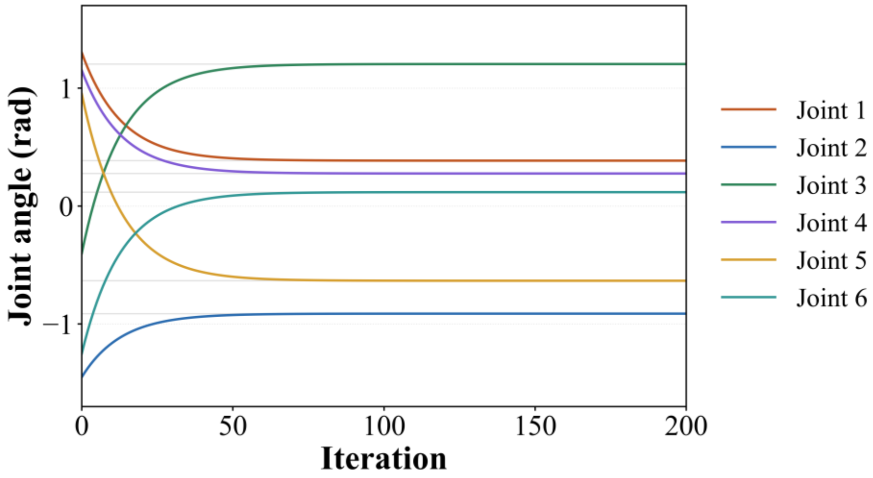

Figure 2: Joint-angle convergence of the six-DOF manipulator.

3.2. Bi-Objective Optimization and Obstacle-Constrained Routing

The bi-objective setting adds energy to the objective. Figure 3 shows the error-energy Pareto front returned by NSGA-II in a single run. The front is convex, and the selected operating point sits at 0.5841 mm error and 6.5042 J. Both baseline references fall off the front: the weighted-sum genetic algorithm is dominated on both terms, and the error-only Whale Optimization Algorithm reaches comparable error but at far higher energy because nothing restrains its joint motion. The shape of the front is the useful output. It shows that holding error near 0.58 mm costs about 6.5 J, and that pushing error lower demands a steep rise in energy, which is the trade-off a task designer needs to see before committing to an operating point.

The obstacle-constrained case places an obstacle between the base and the target. Figure 4 shows the base route chosen by A-Star against the two routing baselines. The A-Star path is collision-free and reaches a length of 26.4 grid units, matching Dijkstra while expanding fewer nodes, and well below the 31.8 units of greedy best-first, which detours around the obstacle field. Once the base route fixes the reachable region, NSGA-II optimizes the joint path under the same error-energy objective and returns 0.6912 mm error at 7.3168 J. The error is slightly higher than the obstacle-free case, and the energy rises because the base now has to move before the arm acts.

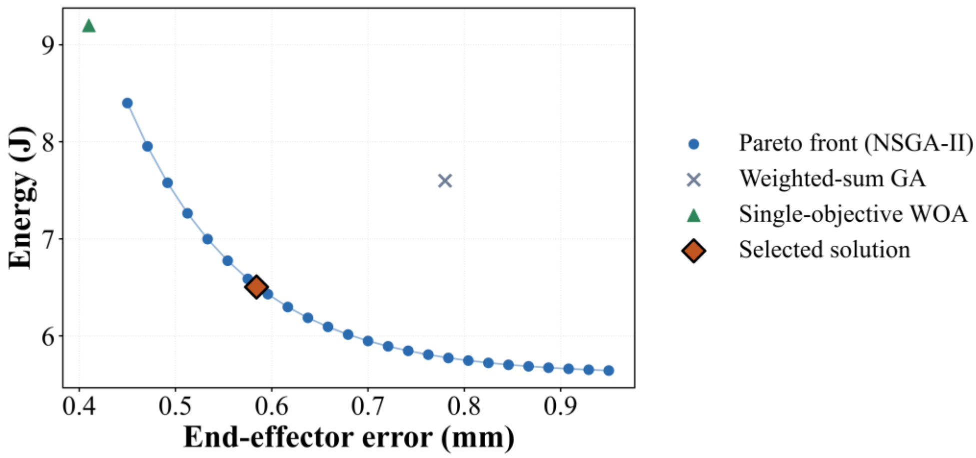

Figure 3: Error–energy Pareto front under bi-objective optimization.

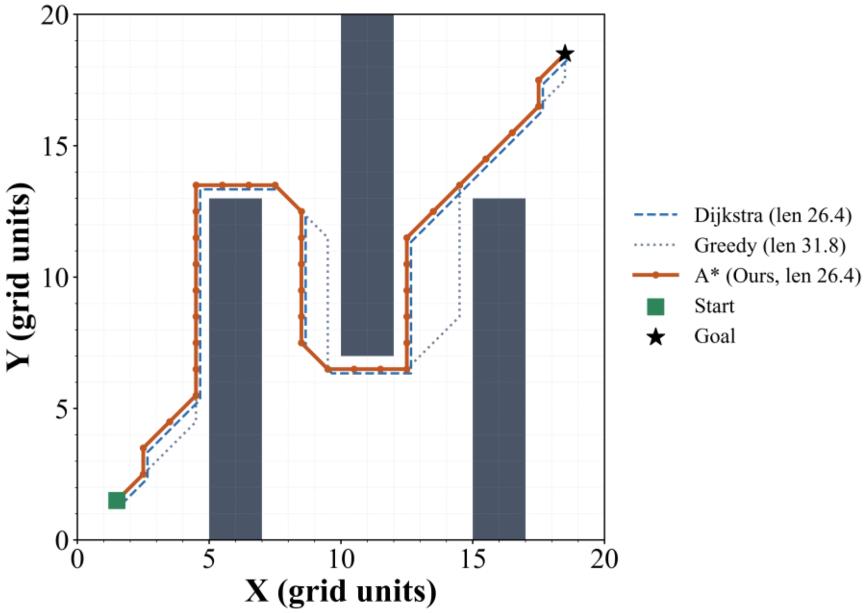

Figure 4: A-Star base routing versus baseline planners.

3.3. Cross-Task Comparison and Multi-Object Extension

Figure 5 places the three task settings side by side. End-effector error rises from 0.4127 mm in the single-objective setting to 0.5841 mm in the bi-objective setting and 0.6912 mm in the obstacle-constrained case, the expected cost of adding the energy objective and then the obstacle constraint. The genetic-algorithm baseline for the single-objective setting stays at 0.6583 mm, above every result the proposed pipeline produces. Energy moves from 6.5042 J in the bi-objective setting to 7.3168 J in the obstacle case, the increment attributable to base motion. The figure makes the pattern plain: each added constraint trades a small amount of accuracy or energy for a capability the previous setting did not have.

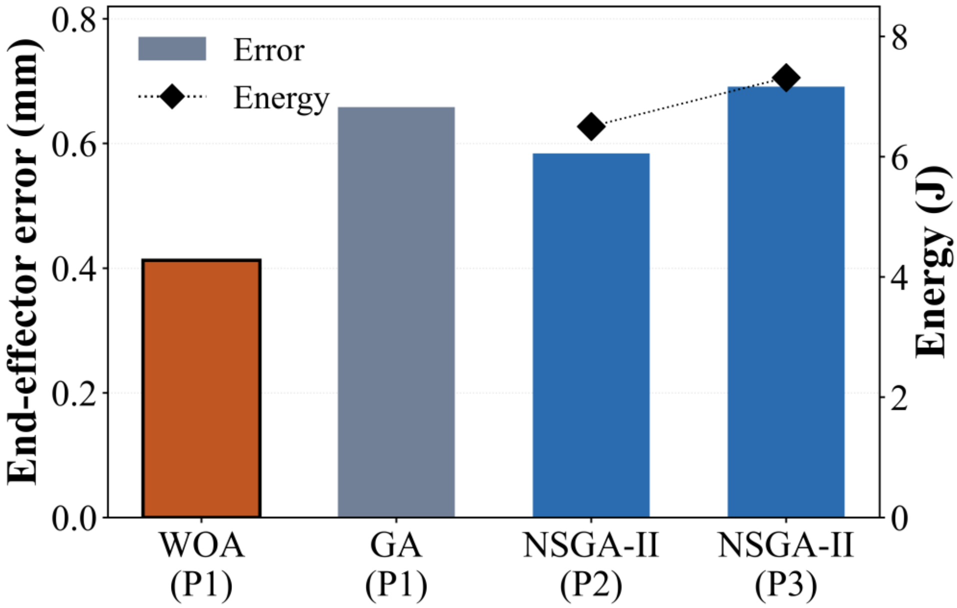

Figure 5: Error and energy comparison across task settings.

The multi-object case extends the obstacle setting to several targets. The manipulator retrieves three objects in sequence, with A-Star planning a shared base route that serves all three pickups and NSGA-II optimizing the joint path for each. The cumulative energy is 19.7426 J, below three times the single-object cost of 7.3168 J, because the shared route removes redundant base travel. The mean end-effector error across the three grasps is 0.7058 mm, in line with the single-object result.

For model evaluation, we compared the genetic algorithm against the Whale Optimization Algorithm on the single-objective task under identical population size and iteration count. The genetic algorithm reaches a workable solution after many iterations, but its final error of 0.6583 mm stays above the 0.4127 mm of the Whale Optimization Algorithm, and its convergence is slower. The selected model is therefore the stronger choice for this task.

4. Conclusions

This paper presents a four-stage optimization framework for joint-angle path design on a six-DOF manipulator, combining kinematic and energy modeling with metaheuristic search and obstacle-aware base routing. In the accuracy-only setting, the Whale Optimization Algorithm drives the end-effector error to 0.4127 mm, 37.3% below the 0.6583 mm of the genetic-algorithm baseline, and converges in roughly half the iterations; the six joint angles settle at [0.385, −0.912, 1.204, 0.276, −0.633, 0.118] rad. Once energy enters the objective, NSGA-II returns a Pareto solution at 0.5841 mm error and 6.5042 J, and adding A-Star base routing for a single obstacle reaches 0.6912 mm at 7.3168 J. Extending the routing-and-optimization loop to several targets retrieves three objects at a cumulative 19.7426 J, below three times the single-object cost, with a mean error of 0.7058 mm. The framework addresses accuracy, energy, and obstacle avoidance within one workflow rather than treating them as separate problems, and the metaheuristic core stays effective as constraints accumulate. Future work will fold joint velocity and torque limits into the constraint set, extend the routing layer to dynamic obstacles, and test the pipeline on a physical arm under sensing noise.

References

[1] Shrivastava, A. (2025) Exploring optimal motion strategies: A comprehensive study of various trajectory planning schemes for trajectory selection of robotic manipulator. Journal of the Institution of Engineers (India): Series C, 106, 691-710.

[2] Wu, B., Ding, Z. and Huang, J. (2026) A review of continual learning in edge AI. IEEE Transactions on Network Science and Engineering.

[3] Romero, S., Valero, J., García, A. V., Rodríguez, C. F., Montes, A. M., Marín, C. and Álvarez-Martínez, D. (2025) Trajectory planning for robotic manipulators in automated palletizing: A comprehensive review. Robotics, 14, 55.

[4] Wu, B., Ding, Z., Ostigaard, L. and Huang, J. (2025) Reinforcement learning-based energy-aware coverage path planning for precision agriculture. Proceedings of the 2025 ACM Research on Adaptive and Convergent Systems (RACS), 1-8.

[5] Yagüe, M. P., García, J. E. S. and Penas, M. S. (2026) Human-intelligent trajectory optimization for robotic manipulators with hybrid PSO-PS algorithm. Advanced Engineering Informatics, 69, 103941.

[6] Wu, B., Cai, Z., Wu, W. and Yin, X. (2023) AoI-aware resource management for smart health via deep reinforcement learning. IEEE Access, 11, 81180-81195.

[7] Hosseinzadeh, M., Tanveer, J., Rahmani, A. M., Baptista, M. L., Abbaszadi, R., Gharehchopogh, F. S. and Lee, S. W. (2026) A comprehensive survey of hybrid whale optimization algorithm with long-short term memory: Applications, improvements, and future perspective. Archives of Computational Methods in Engineering, 33, 3665-3706.

[8] Wu, B. and Wu, W. (2023) Model-free cooperative optimal output regulation for linear discrete-time multi-agent systems using reinforcement learning. Mathematical Problems in Engineering, 6350647.

[9] Makhadmeh, S. N., Kassaymeh, S., Rjoub, G., Bataineh, B., Sanjalawe, Y. and Al-Betar, M. A. (2025) Recent advances in multi-objective whale optimization algorithm, its versions and applications. Journal of King Saud University Computer and Information Sciences, 37, 200.

[10] Acharya, B., Panda, S., Das, S., Majhi, S. K., Gerogiannis, V. C. and Kanavos, A. (2025) Optimizing task

scheduling in cloud environments: A hybrid golden search whale optimization algorithm approach. Neural Computing and Applications, 37, 10851-10873.
[11] Sadni, F. E., Salhi, I., Belhora, F. and Hajjaji, A. (2025) Multi-objective optimization of energy and exergy efficiencies in ORC configurations using NSGA-II and TOPSIS. Thermal Science and Engineering Progress, 63, 103606.
[12] Asghari, A., Zeinalabedinmalekmian, M., Azgomi, H., Alimoradi, M. and Ghaziantafrishi, S. (2025) Farmer ants optimization algorithm: A novel metaheuristic for solving discrete optimization problems. Information, 16, 207.
[13] Nahidmobarakeh, L., Nemetiandoost, M., Yilmaz, B.S., Gazzarri, J., Zhang, X., Arias, S. and Ahmed, R. (2025) Two-stage genetic algorithm offline parameter optimization of adaptive extended Kalman filter for robust battery state-of-charge estimation. IEEE Access.
[14] Huang, J., Wu, B., Duan, Q., Dong, L. and Yu, S. (2025) A fast UAV trajectory planning framework in RIS-assisted communication systems with accelerated learning via multithreading and federating. IEEE Transactions on Mobile Computing.
[15] Imandoust, M., Alghorayshi, S. T. K., Zahedi, R., Aslani, A. and Khanqah, M. Q. (2025) Multi-objective optimization of hybrid solar-wind MSF-RO cogeneration desalination systems using NSGA-II to enhancing energy efficiency, cost reduction and sustainability. Separation and Purification Technology, 134740.
[16] Zandniapour, K., Soroush, A., Agdam, E. K. and Sanaieian, H. (2025) Integrating GIS, 3D-Isovist, and an NSGA-II multi-objective optimization algorithm for automation of design process in urban parks and public open spaces. International Journal of Geoheritage and Parks, 13, 1-16.
[17] Wu, B., Huang, J. and Yu, S. (2026) 'X of Information' continuum: A survey on AI-driven multi-dimensional metrics for next-generation networked systems. IEEE Communications Surveys & Tutorials.
[18] Wu, B., Huang, J., Duan, Q., Dong, L. and Cai, Z. (2025) Enhancing vehicular platooning with wireless federated learning: A resource-aware control framework. IEEE/ACM Transactions on Networking, 33, 1-16.
[19] Rasul, M.J., Abbas, A., Baek, J. and Kim, J. (2026) A hybrid ensemble learning framework with uncertainty quantification for state-of-health estimation in lithium-ion batteries. Measurement, 120528.
[20] Wu, B., Huang, J. and Duan, Q. (2025) FedTD3: An accelerated learning approach for UAV trajectory planning. Proceedings of the International Conference on Wireless Artificial Intelligent Computing Systems and Applications (WASA), 13-24.
[21] Asgari, M., Ommi, F. and Saboohi, Z. (2025) Aeroelastic modeling and multi-objective optimization of a subsonic compressor rotor blade using a combination of modified NSGA-II, ANN, and TOPSIS. Results in Engineering, 26, 104615.
[22] Yfantis, V., Wagner, A. and Ruskowski, M. (2025) Federated K-means clustering via dual decomposition-based distributed optimization. Franklin Open, 10, 100204.
[23] Wu, B., Huang, J. and Duan, Q. (2025) Real-time intelligent healthcare enabled by federated digital twins with AoI optimization. IEEE Network, 1.
[24] Pant, Y.R., Leigh, L. and Fajardo Rueda, J. (2025) Improving K-means clustering: A comparative study of parallelized version of modified K-means algorithm for clustering of satellite images. Algorithms, 18, 532.
[25] Pan, D., Wu, B.-N., Sun, Y.-L. and Xu, Y.-P. (2023) A fault-tolerant and energy-efficient design of a network switch based on a quantum-based nano-communication technique. Sustainable Computing: Informatics and Systems, 37, 100827.
[26] Ahnouz, I., Arahmane, H. and Sebihi, R. (2025) Optimizing neutron-gamma discrimination in scintillation detectors using Tucker decomposition. Kuwait Journal of Science, 100511.