Decision Making: Applications in Management and Engineering

Journal homepage: www.dmame-journal.org
ISSN: 2560-6018, eISSN: 2620-0104

Improved Multi-objective Particle Swarm Optimization in Software Engineering Supervision

Ping Yue[1], Zhiguo Wang[1*]

[1] Hangzhou Weishi Technology Company, Hangzhou, 315000, China

ARTICLE INFO

Article history:
Received 22 December 2023
Received in revised form 15 February 2024
Accepted 1 March 2024
Available online 12 March 2024

Keywords:
Software engineering supervision; IDMPSO; Multi-objective network planning optimization; Pareto-optimal set.

ABSTRACT

In the 21st century, the software industry has achieved great development, and the development complexity and volume of software projects are also continuously increasing. Therefore, the design of software engineering supervision network plans is becoming increasingly important. The Pareto optimal solution set construction method, global extremum selection method, and fitness value determination method of the multi-objective particle swarm optimization algorithm have been improved in response to the poor optimization performance and poor convergence and distribution of optimal solutions in existing network planning algorithms. However, traditional methods only optimize one or two objectives of network planning, resulting in inconsistency with actual engineering. Thus, this study establishes a multi-objective model based on resources, duration, cost, and quality for comprehensive optimization. Based on the results, the Pareto optimal solution curves obtained by the proposed algorithm on three classic test functions were consistent with the actual theoretical Pareto frontier curves. The proposed method was applied to engineering project examples. 10 solutions that met the schedule requirements were obtained. Most engineering projects had a quality of over 80%, which verified the practicality of the algorithm. The algorithm has achieved good results in optimizing engineering quality. Therefore, this model can consider various indicators such as resources and costs to obtain software engineering quality improvement plans. It has certain application potential.

1. Introduction

With the rapid progress of the Internet, the software industry has become the pillar industry of the information industry today and the development complexity and volume of software projects are also continuously increasing. Therefore, it has become very important to introduce a software engineering supervision network plan in software engineering projects to guide engineering projects and achieve effective control of engineering resources, construction period, costs, and quality. Software engineering supervision network planning is a Multi-objective Optimization (MOO) problem. If traditional single-objective methods are used for network supervision optimization, conflicts can

*Corresponding author.
E-mail address: hrgzgd@163.com

https://doi.org/10.31181/dmame7220241074

easily occur between various engineering objectives. The goals of the shortest construction period, minimum resource utilization, and minimum cost cannot be achieved simultaneously, resulting in inconsistency with actual engineering projects [1-2]. Based on this, a MOO model is designed to comprehensively optimize multiple objectives of the software engineering supervision network plan. At the same time, network planning optimization problems are very complex, making it difficult to achieve multi-objective network optimization goals through a single algorithm. Moreover, existing MOO methods also have low convergence and uneven distribution of optimal solutions [3]. Therefore, to solve the multi-objective conflict problem in software engineering supervision network optimization, a Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is studied for optimization. A MOO model for network engineering supervision has been established, and by constructing a MOPSO model, comprehensive optimization of engineering resources, schedule, cost, and quality has been achieved, improving the convergence and distribution of the algorithm. This study has the following four aspects. The first part introduces the current research status of software engineering supervision network optimization technology and DMPSO algorithm. The second part constructs an improved method for DMPSO, as well as the specific design of the DMPSO model for software engineering supervision. The third part mainly focuses on the simulation experimental results of the research model. Then, the example application results of the software engineering supervision optimization network are analyzed. The fourth part is a summary and analysis of the entire text.

2. Related Works

With the rapid development of the software industry, the research on multi-objective network planning optimization methods for software engineering supervision is receiving increasing attention. Many scholars have researched multi-objective network planning optimization. Considering the nonlinear MOO global supply chain network models, Hasani et al., [4] proposed a new hybrid heuristic algorithm based on the improved Strength Pareto Evolutionary Algorithm 2 (SPEA2) for optimization. The results indicated that this method integrated economic and environmental factors and responded to customer needs. However, there was still poor convergence of the optimal solution. To improve the resource utilization and cost-effectiveness of urban sewage treatment projects, Ye et al., [5] proposed a multi-agent hybrid particle swarm optimization to optimize the network design of the project. The genetic algorithm was introduced for comparative experiments. The results indicated that this method had good performance in optimizing engineering costs. Based on numerical research on network planning and optimization problems in 5G telecommunications systems, Tun [6] proposed a specific area wireless network planning algorithm based on linear programming technology to optimize the source and target nodes in the G network. The feasibility of the method was verified through simulation experiments. Zeidan et al., [7] proposed a heuristic MOO method for segmenting and operating water distribution systems. This method divided the network into clusters based on connectivity analysis. It was applied to balance operating costs, overpressure, and water age. The feasibility of the algorithm was verified through simulation experiments. However, the algorithm still had poor convergence.

In multi-objective network planning optimization, the DMPSO algorithm is widely used. However, existing algorithms still have poor convergence. Some scholars have also conducted relevant research on the improvement of DMPSO. Devaraj et al., [8] proposed a new load-balancing algorithm to address the uneven distribution of load resources in computer network planning. This algorithm combined the Firefly algorithm and improved DMPSO technology. The minimum search space was obtained through the Firefly algorithm. The DMPSO algorithm was applied to identify enhanced responses. The algorithm was verified to have good optimization performance through simulation experiments. Xu et al., [9] addressed the poor convergence in existing DMPSO algorithms and transformed the global convergence of the original DMPSO into the convergence metric sequence. The defined convergence metric method was used to analyze the global convergence of the original DMPSO from the probability theory. It indicated that this method had some effectiveness in improving convergence. However, it was not specifically implemented. Yuen et al., [10] proposed an improved competitive mechanism DMPSO algorithm for MOO problems. It applied a competition mechanism to inertia weights. The most appropriate balance between the exploration and development capabilities of algorithms during the search process was explored. According to the findings, it was superior to the other four common DMPSO algorithms.

For multi-objective decision-making problems in the real world, Rasoulzadeh et al., [11] proposed a new group model that combined Intuitionistic Fuzzy Sets (IFS) to handle uncertainty in practical problems and solved it using Non-dominated Sorting Genetic Algorithm II (NSGA-II). The comparison with existing methods indicated that the proposed model was effective in selecting the optimal investment portfolio. Nafei et al., [12,13] proposed an authoritarian strategy based on a Neuromorphic Set (NS) to address uncertainty, ambiguity, and ambiguity in the real world. This method was successfully applied to supplier selection problems by transforming different management decisions and weight matrices into a unified evaluation matrix. The study also proposed a new Neuromorphic Multi-choice Objective Programming (NMCGP) model to simulate real-world problems more realistically. The effectiveness and computational complexity of the introduced method in dealing with multi-objective decision-making problems were demonstrated through numerical and mathematical examples. Akram et al., [14] proposed a method for Multi-objective Transportation Problems (MOTP) with Fermatéan fuzzy costs by treating each arc as a Decision Unit (DMU) for Data Envelopment Analysis (DEA) and using the DEA model to obtain the Fermatéan fuzzy efficiency score. Example analysis showed that this method was effective and more accurate than existing technique. Mekawy [15] focused on multi-objective linear fractional programming problems in fuzzy environments, introducing all parameters into piecewise quadratic fuzzy numbers and transforming them into brittle problems. The effectiveness of the method was verified through examples, and it had the potential for application in practical problems. Farnam and Darehmiraki [16] established a model for multi-objective linear fractional programming problems in a fuzzy environment, which fully considered uncertainty. The applicability of this method in practical problems was verified through examples, providing a new perspective for the study of multi-objective fractional programming in hesitant fuzzy environments. Ghasemi et al., [17] proposed a multi-objective mathematical location routing model aimed at reducing logistics costs and improving reliability. The model solved problems of different scales using Epsilon constraints and NSGA-II methods. The results indicated that the model had correct performance in solving location and routing problems.

In summary, in the optimization of software engineering supervision network plans, traditional single-objective methods are prone to conflicts between various engineering objectives, resulting in inconsistencies with actual engineering projects. However, existing multi-objective network plan optimization methods also have low convergence and distribution uniformity of the optimal solution. Based on this, by improving the DMPSO, the software engineering supervision multi-objective network plan is optimized. The Pareto optimal curve obtained is closer to reality, providing better solutions for engineering project decision-makers.

3. Design of Software Engineering Supervision Optimization Network based on Improved DMPSO

The existing MOO algorithms for software engineering supervision network planning have low convergence and uniform distribution of optimal solutions. Based on the Improved Multi-Objective Particle Swarm Optimization (IDMPSO), a MOO model based on resources, duration, cost, and quality is established for network engineering supervision, improving convergence and distribution.

3.1 Specific Design of IDMPSO

The existing MOO algorithms have low convergence and uneven distribution of optimal solutions. Combined with the characteristics of network planning optimization problems in software engineering supervision, the proposed DMPSO has been theoretically explored and analyzed. By improving several main parameters that affect the algorithm, the algorithm is improved to enhance the convergence performance, making the Pareto optimal curve closer to reality and providing better solutions for decision-makers. The flow diagram of the existing DMPSO is shown in Figure 1 [18-19].

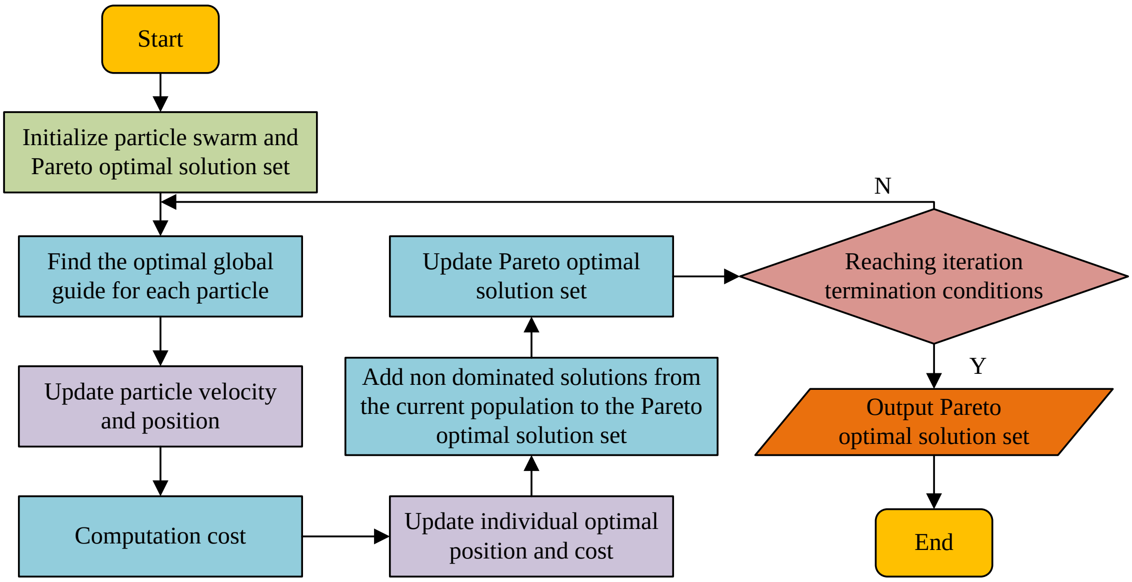


Fig. 1. Flowchart of DMPSO method

In Figure 1, the DMPSO introduces the Pareto Optimal Solution Set (POSS) for multi-objective search. The general process of the algorithm is as follows. Firstly, the particle swarm and Pareto dataset are initialized. Then a global optimal guide is found for each particle, updating particle velocity and position information. The individual extremum is calculated to update the optimal position and cost of the individual. The non-dominated solution is added to the POSS, further updating the Pareto solution set. Whether the end condition is reached is judged in the end. If it reaches, the POSS is output. Among them, particle velocity and position are shown in Eq. (1-2) [20].

$$V_{eiv}^{t+1} = \omega V_{eiv}^{t} + c_1 r_1 (P_{eiv}^{t} - X_{eiv}^{t}) + c_2 r_2 (P_{eiv}^{t} - X_{eiv}^{t}) \tag{1}$$

In Eq. (1), t stands for the iterations. V_{ej}^{t+1} stands for the velocity of the e particle during the $t+1$ -th optimization in the j -th dimensional space. ω stands for inertia weight. c_1 and c_1 stand for acceleration factors. r_1 and r_2 stand for random numbers between 0~! . V_{ej}^{t} stands for the speed generated by the current update of the particle. X_{ej}^{t} represents the position of the e at the t -th iteration in the j -th dimensional space. P_{ej}^{t} stands for the individual optimal position.

$$X_{eiv}^{t+1} = V_{eiv}^{t+1} + X_{eiv}^{t} \quad (2)$$

In Eq. (2), X_{ej}^{t+1} stands for the position of the e during the $t+1$-th optimization in the j-th dimensional space. The research will optimize the four basic operators in DMPSO to improve the convergence performance. The construction method of POSS individual extreme value selection, global optimal value selection method, and individual file update method are improved. The construction of the POSS is improved. The construction method is shown in Figure 2.

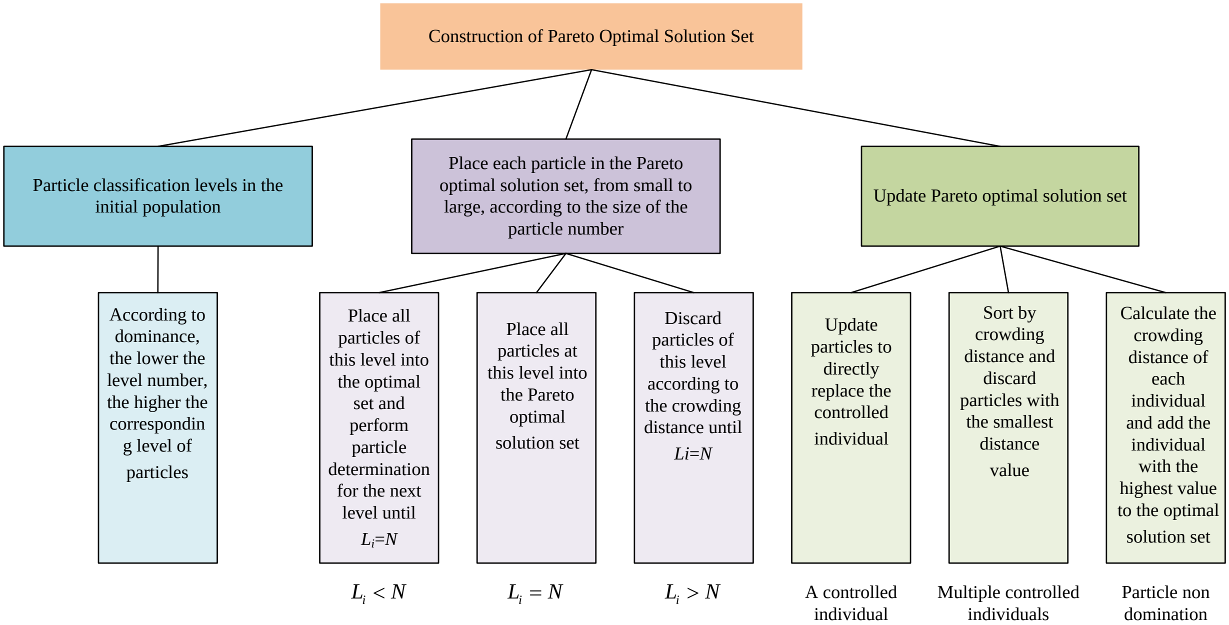


Fig. 2. The construction method of the POSS

In Figure 2, the construction and improvement process of the POSS is as follows. According to the dominance relationship, particles are graded and numbered. Then, according to the number from small to large, the crowding distance is introduced for judgment. Particles with small crowding distance are screened. The screened particles are placed in the POSS. Finally, the POSS is updated based on the crowding distance. Among them, retaining particles with large crowding distances is to broaden the search range of the algorithm and maintain the diversity of the final solution. The expression formula for the defined crowding distance is shown in Eq. (3).

$$\begin{cases} d_l(x_n) = \left| f_1(x_{n1}) - f_l(x_n) \right| + \left| f_l(x_{n2}) - f_l(x_n) \right| \\ D(x_n) = \sum_{l=1}^{m} d_l(x_n) \end{cases} \quad (3)$$

In Eq. (3), $d_l(x_n)$ represents the crowding distance of the l-th object function corresponding to the x_n particle. $D(x_n)$ represents the overall crowding distance of x_n particles. $f_l(x_n)$ represents the position of the x_n particle. $f_1(x_{n1})$ and $f_l(x_{n2})$ represent the two closest points. $l = 1, 2, ..., m$ represents the number of objective functions. For global extremum construction, the optimal position of each individual is obtained from the non-dominated solutions of the POSS. The particle with the highest dominance level is considered as the global extremum of that individual. If multiple solutions dominate the individual, the solution with the maximum crowding distance is selected as the global extremum to enhance the diversity and uniformity of particles. To eliminate the impact of individual extremum on algorithm convergence and diversity, Euclidean distance is introduced to construct individual extremum. The particle with the smallest Euclidean distance is selected as the individual extremum. The specific calculation formula for Euclidean distance is shown in Eq. (4).

$$d = \sqrt{(f_l(x_p) - f_l(x_b))^2 + (f_m(x_p) - f_m(x_b))^2} \quad (4)$$

In Eq.(4), d represents the Euclidean distance. $f_l(x_p)$ and $f_m(x_p)$ represent the function values of the globally optimal particle x_p on the l and m objective functions, respectively. $f_l(x_b)$ and $f_m(x_b)$ represent the function values of particle x_b on the l and m objective functions, respectively. At the same time, the optimal position of an individual and the global optimal position of the population are mainly determined through the fitness function. It is further improved to enhance the iterative optimization performance. Specifically, the particle fitness function is determined by counting the number of particles dominating other particles to determine the level of each particle. After improving the construction method of the POSS, global extremum selection method, and fitness value determination method, the DMPSO process is shown in Figure 3.

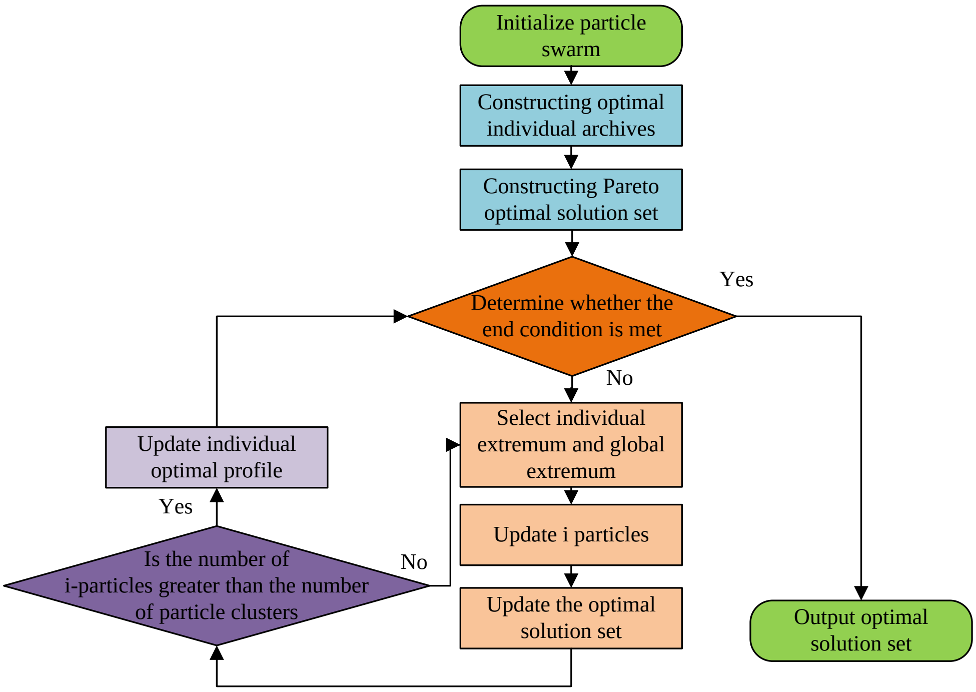


Fig. 3. Flow chart of improved DMPSO

3.2 Design and Implementation of IDMPSO for Software Engineering Supervision

Based on the IDMPSO and combined with the main objectives of actual software engineering supervision network planning, a specific MOO model is established to achieve network planning optimization of software engineering supervision. The main goal of optimizing the network plan for software engineering supervision is to achieve effective control over the construction period, cost, resources, and quality. Thus, the project achieves the shortest time, lowest cost, most balanced resource utilization intensity, and highest quality. Therefore, a resource-time-cost-quality integrated optimization model is established to comprehensively optimize the four objectives. This provides project management personnel with a more effective supervision plan. The relevant indicator requirements for the comprehensive optimization model constructed through research are shown in

Figure 4. In Figure 4, in terms of selecting resource indicators for the model, the amounts of resources at each moment need to be quantified. There are no special requirements that cannot be interrupted. The final obtained resources are the total resources after synthesis. The resource optimization function is used to analyze the resource utilization intensity at every moment, balancing various resources. The optimization equation for resource utilization intensity is shown in Eq. (5) [21].

$$f = \min \sqrt{\frac{1}{T}\sum_{t=1}^{T}[R(t) - R_m]^2} \tag{5}$$

In Eq. (5), f represents the real-time usage intensity of the resource. T represents the total duration of the engineering project. $R(t)$ represents the total resources consumed by all activities during time t . $t = 1, 2, ..., T$ represents any time during the construction period. R_m represents the average consumption resource intensity of the project. The mathematical expression of $R(t)$ is shown in Eq. (6).

$$R(t) = \begin{cases} \sum_{k=1}^{m} R_t(k), ES(i) + O(i) < t \leq ES(i) + O(i) + T(i) \\ 0, t \leq ES(i) + O(i) or t > ES(i) + O(i) + T(i) \end{cases} \tag{6}$$

In Eq. (6), $ES(i)$ represents the earliest start time of activity i . $O(i)$ represents the start time of activity i relative to $ES(i)$. $T(i)$ represents the execution time of activity i . m represents the total number of resources. $k = 1, 2, ..., m$ represents the k -th resource.

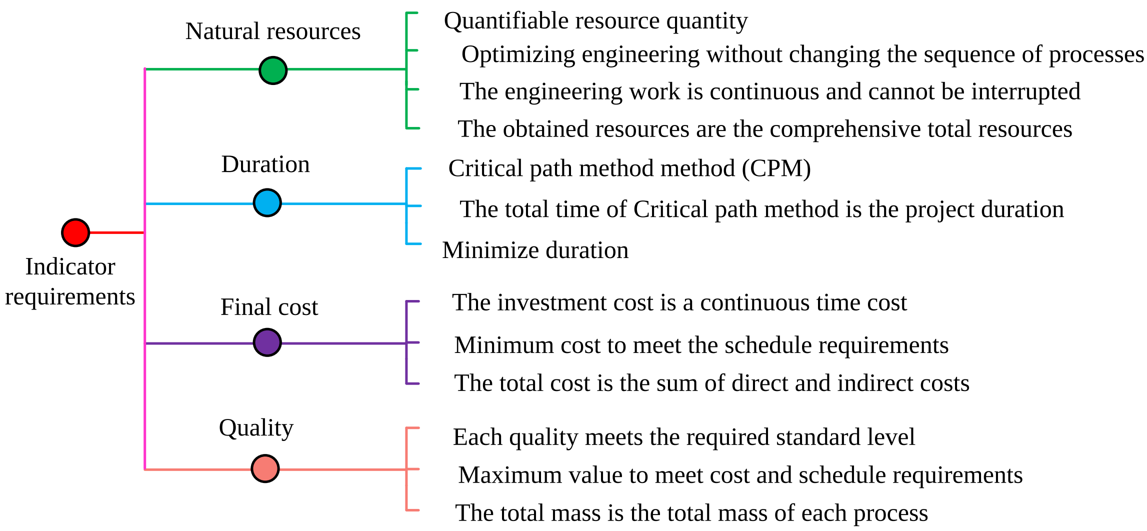


Fig. 4. Flow chart of IDMPSO

In Figure 4, the optimization of project duration is mainly achieved through the critical path method to minimize the duration. The specific time optimization function expression is shown in Eq. (7).

$$T = Min \sum_{i=1}^{N} T_i \tag{7}$$

In Eq. (7), T represents the total time spent on the critical path. T_i represents the time spent on the i-th process on the critical path. N stands for the operations on the critical path. $i = 1, 2, ..., N$ displays the i-th process. The project needs to meet the minimum cost of the construction period. The optimization functions for project duration and cost are shown in Eq. (8-9).

$$\min c = \min \sum_{i=1}^{N} [cs_i + \partial_i * (t_i - ts_i)] + c_\xi \quad (8)$$

In Eq. (8), $\min c$ represents the objective function with the lowest cost. ξ represents the indirect cost rate of the project. c_ξ represents the indirect cost of the project. cs_i represents the time spent on work i in a rush state. ∂_i represents the change rate in indirect costs of work. t_i represents the duration of i work. ts_i represents the duration of work in a rush state.

$$\min T = TE(1) - TE(0) \quad (9)$$

In Eq. (9), $\min T$ represents the objective function of the minimum time. $TE(0)$ stands for the start time of the engineering project. $TE(1)$ stands for the end time of the engineering project. In Figure 4, the optimization model requires each quality to reach the standard level and meet the maximum duration cost. The specific quality optimization function is shown in Eq. (10).

$$Q = \sum_{i=1}^{N} \omega_i \sum_{j=1}^{K} \omega_{i,j} * Q_{i,j} \quad (10)$$

In Eq. (10), Q represents the objective function of all masses. ω_i represents the weight ratio of the i-th sub-task. N stands for the total sub-tasks. $\omega_{i,j}$ represents the i-th quality weight of i jobs. K represents the total quality of sub-tasks. $Q_{i,j}$ represents the required value for the j-th quality of i works.

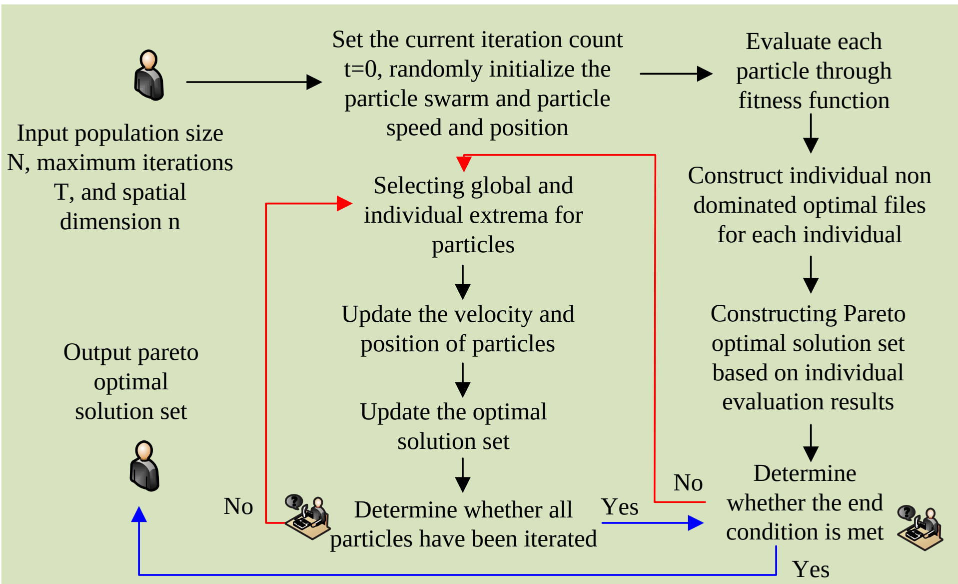


Fig. 5. Actual algorithm optimization process

The research will combine the actual software engineering supervision network planning model to program the software engineering supervision plan. Based on the improvement of the DMPSO optimization process, optimization design is carried out. The optimization process of the designed algorithm is shown in Figure 5. The algorithm first optimizes each network planning scheme by calculating the critical path of the network plan. Then, based on the sliding on the critical path, the total project duration is calculated, ultimately achieving effective optimization of the project duration. At the same time, based on the four indicators of duration, resources, cost, and quality, the asynchronous update method is used to determine the global extremum corresponding to each particle. The velocity and position are updated. The optimized particle velocity and position update formulas are shown in Eq. (11-12).

$$V_{eiv}^{t+1} = \omega V_{eiv}^{t} + c_1 r_1 (P_{eiv}^{t} - X_{eiv}^{t}) + c_2 r_2 (P_{eiv}^{t} - X_{eiv}^{t}) \tag{11}$$

In Eq. (11), V_{eiv}^{t+1} represents the amount of change in the v-th indicator attribute of the e-th particle in the i. X_{eiv}^{t} represents the v-th index attribute value of the e-th particle in the i. B_{eiv}^{t} represents the individual extreme value of particle e. P_{eiv}^{t} represents the global optimal value of e. The other parameters are shown in Eq. (12).

$$X_{eiv}^{t+1} = V_{eiv}^{t+1} + X_{eiv}^{t} \tag{12}$$

To test the performance of the IDMPSO, the standard detection functions commonly used in DMPSO are introduced to test the convergence and distribution. The specific test functions are shown in Table 1 [22].

Table 1
Tested functions

Problem	Dimension	Range	Objective function	Convergence or not	Continuity
ZDT1	30	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1-\sqrt{x_1 / g(x)}\right]$ $g(x) = 1+9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	Convergence	Continuous
ZDT3	30	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1-\sqrt{x_1 / g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1+9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	Convergence	Nonsequence
ZDT6	30	[0,1]	$f_1(x) = 1-\exp(-4x_1)\sin^6(4\pi x_1)$ $f_2(x) = g(x)\left[1-(f_1(x)/g(x))^2\right]$ $g(x) = 1+9\left[\left(\sum_{i=2}^{n} x_i\right)/(n-1)\right]^{0.25}$	Convergence	Continuous

In Table 1, this test function performs algorithm performance testing through three test functions. The POSS is obtained through the corresponding objective function. For the measurement of algorithm convergence, the generation distance is introduced to test the convergence. The definition of the generation distance is shown in Eq. (13).

$$GD = \left(\frac{1}{n_{pa}} \sum_{i=1}^{n_{pa}} d_i^2 \right)^2 \tag{13}$$

In Eq. (13), GD represents the generation distance. n_{pa} represents the number of non-inferior solutions in the Pareto solution set. d_i represents the Euclidean distance between the i-th solution in the target space and the nearest solution in the real Pareto front end. When $GD = 0$, the convergence performance of the algorithm is optimal. The closer the GD value is to 0, the better the convergence performance of the algorithm. For the distribution uniformity of the solution, the spatial measurement index is used to measure it. The specific calculation for the spatial measurement index is shown in Eq. (14).

$$S = \frac{\left[\frac{1}{n_{pa}} \sum_{i=1}^{n_{pa}} (d_i' - \overline{d'}) \right]^{\frac{1}{2}}}{\overline{d'}} \tag{14}$$

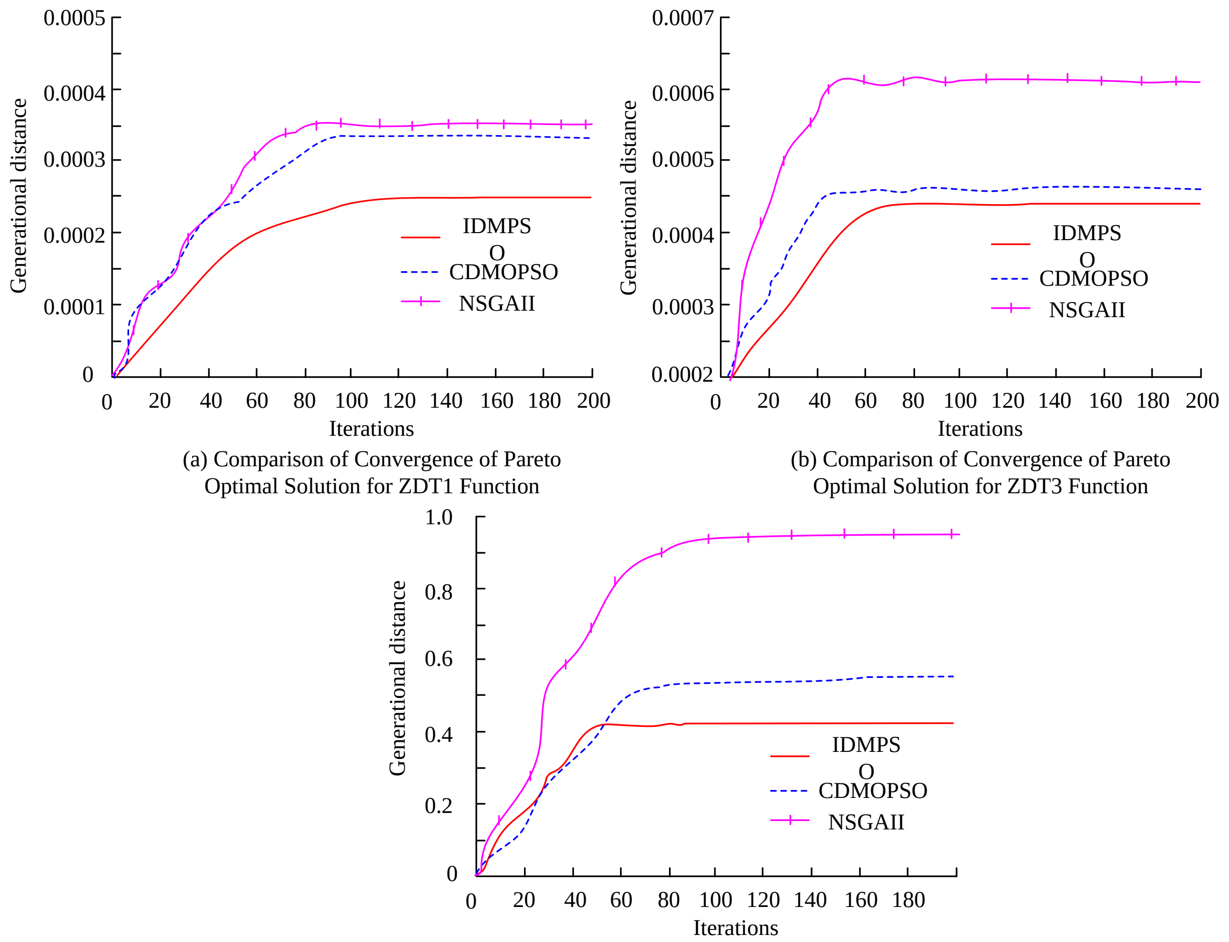
In Eq. (14), S represents the spatial measurement indicator. $\overline{d'} = \frac{1}{n_{pa}} \sum_{i=1}^{n_{pa}} d_i'$. d_i' represents the Euclidean distance between the i-th particle in the POSS and the nearest particle. n_{pa} represents the final number of non-dominated solutions in the POSS. At $S = 0$, the nondominated solutions in the POSS are uniformly distributed. The closer S is to 0, the more uniform the distribution of solutions.

4. Optimization Results of Software Engineering Supervision Based on IDMPSO

This chapter conducted simulation experiments on the IDMPSO. The MOPSO standard test function was applied to test the performance of the IDMPSO. Related algorithms were introduced for comparative analysis. At the same time, the improved algorithm design software engineering supervision model was applied to engineering project examples to verify the practicality. The engineering case project data used in the research was sourced from "Multi-objective network planning optimization based on genetic algorithm". The construction period of this project was required within 125 days [23-24].

4.1 Evaluation Results of IDMPSO

To detect the performance more intuitively, the high-performance NSGA-II and Cluster Density Multiple Objective Particle Swarm Optimization (CDMOPSO) algorithms are introduced for comparative experiments with the proposed IDMPSO. Based on expert experience, the particle population size of the algorithm is 200. POSS size is 20. Each algorithm undergoes 200 iterations under three classic test functions. Figure 6 displays the convergence results of the optimal solution.



(a) Comparison of Convergence of Pareto Optimal Solution for ZDT1 Function

(b) Comparison of Convergence of Pareto Optimal Solution for ZDT3 Function

(c) Comparison of Convergence of Pareto Optimal Solution for ZDT6 Function

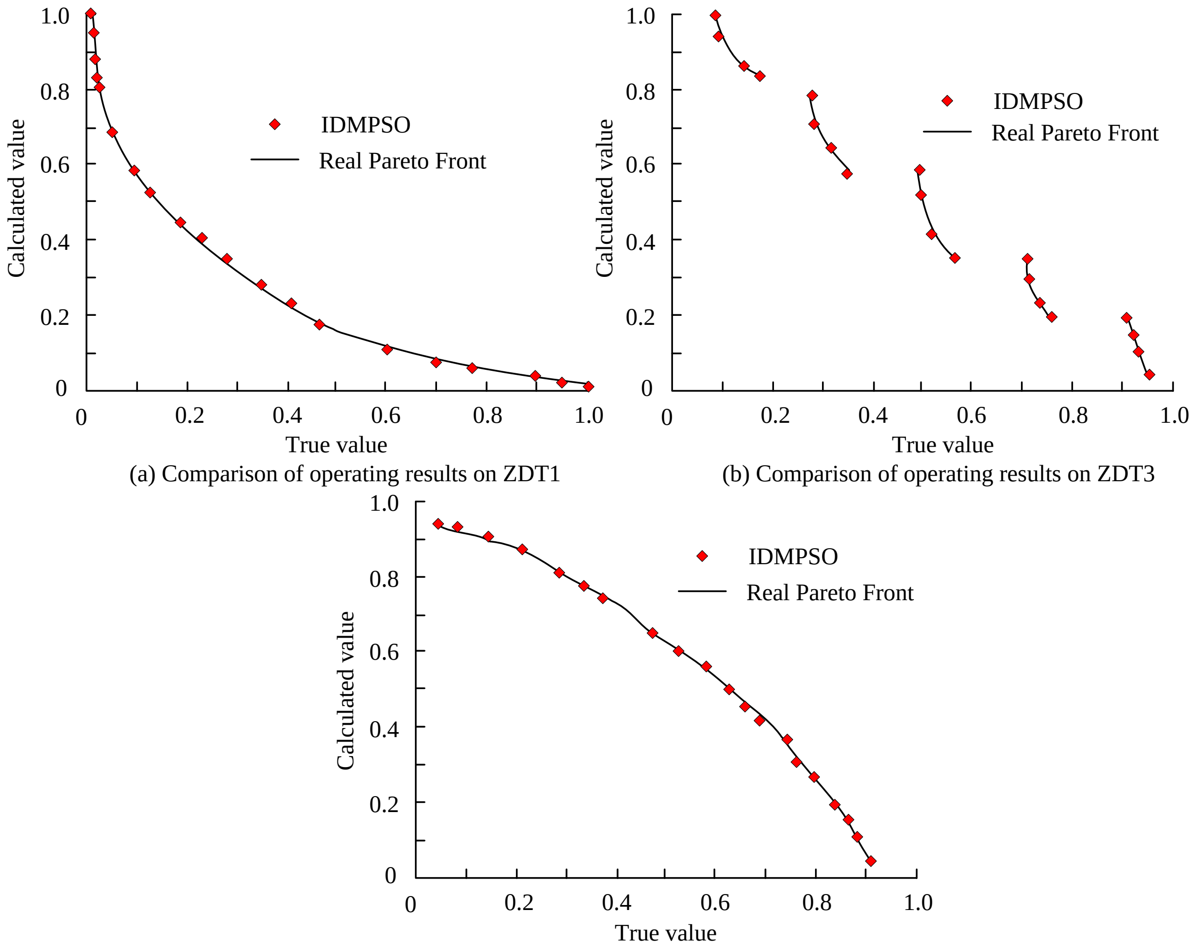
Fig. 6. Comparison chart of convergence of pareto optimal solution

In Figure 6, the IDMPSO algorithm has the smallest GD values on three classic test functions, with values of 0.00025, 0.000445, and 0.4136, respectively. It indicated that IDMPSO had the smallest generation distance and the best convergence. Meanwhile, the IDMPSO algorithm achieved convergence faster than the other two algorithms. Through comparative experiments between NSGA-II and CDMOPSO, it was found that the improved IDMPSO algorithm converged in the first 60 iterations and reached the convergence state as quickly as possible. This indicated that the IDMPSO algorithm had high performance and sensitivity and was suitable for solving MOO problems. The GD values of the three algorithms on the ZGT1 test function were relatively minimal, indicating that the algorithm had good convergence on the ZGT1 test function. Under the same parameter settings, the uniformity of the optimal solution distribution of the proposed algorithm was tested. The uniformity test results under the three test functions ZGT1, ZGT3, and ZGT6 are shown in Table 2.

Table 2
Comparison of distribution uniformity of pareto optimal solutions

Algorithm	ZDTI	ZDT3	ZDT6
NSGAII	0.0047	0.0075	0.1441
CDMOPSO	0.0041	0.004	0.1062
IDMPSO	0.0042	0.0035	0.0043

In Table 2, the spatial metric of the IDMPSO algorithm is only 0.0001 larger than the CDMOPSO algorithm in the ZDTI test function. On the ZDTI function, the optimal solution distribution uniformity of the IDMPSO was slightly smaller than the CDMOPSO. However, the algorithm proposed in the research had the smallest spatial metric on both ZDT3 and ZDT6 functions. The optimal solution distribution of the IDMPSO algorithm on the ZDT3 and ZDT6 functions was better than the other two algorithms. Overall, the algorithm proposed in the study improved the convergence and distribution performance of POSS. To provide a more intuitive analysis of algorithm performance, the Pareto optimal solution curve obtained from the IDMPSO was compared with the theoretical Pareto frontier curve. The comparison curves on three classic test functions are shown in Figure 7.



(a) Comparison of operating results on ZDT1
(b) Comparison of operating results on ZDT3
(c) Comparison of operating results on ZDT6

Fig. 7. Comparison between test results and the actual pareto optimal solution curve

In Figure 7, the Pareto optimal solution curves obtained by the proposed algorithm on ZDT1, ZDT3, and ZDT6 functions are consistent with the actual theoretical Pareto frontier curves. It indicated that the IDMPSO optimized the convergence and distribution of multi-objective algorithms. It was applied to solve the MOO software engineering supervision problems. The obtained curve better approximated the theoretical Pareto curve, proving the effectiveness of the algorithm. To further validate the performance of the multi-objective algorithm proposed in the study, it was applied in practical engineering, including engineering design, production scheduling, and financial investment, along with the other two algorithms. By comparing the performance of algorithms in practical engineering problems, the performance of IMOPSO algorithms was comprehensively evaluated. The performance comparison of different algorithms in practical engineering applications is shown in Table 3.

Table 3
Performance comparison of different algorithms in practical engineering aplications

Performance index	Algorithm	Engineering design	Production scheduling	Financial investment
F1 value	NSGAII	81%	83%	84%
	CDMOPSO	91%	90%	89%
	IDMPSO	95%	96%	94%

In Table 3, the IDMPSO algorithm performs better than NSGAII and CDMOPSO algorithms in engineering design, production scheduling, and financial investment. The F1 values of the IDMPSO algorithm in three fields were 95%, 96%, and 94%, respectively. Therefore, the IDMPSO algorithm had higher performance in practical engineering applications.

4.2 Application Analysis of Software Engineering Supervision Optimization Network Example

After proving the effectiveness of the algorithm, the software engineering supervision model designed based on the improved algorithm was applied to engineering project cases to verify the practicality of the algorithm in software engineering supervision network planning problems. The engineering case project data used in the research is shown in Table 4.

Table 4
Quality weight table for each process of the project

Activity i	Quality index 1 (%)	Quality index 2 (%)	Quality index 3 (%)
1	30	50	20
2	40	40	20
3	15	70	15
4	35	50	15
5	20	60	20
6	25	50	25
7	30	30	40
8	0	100	0
9	50	50	0
10	40	60	0
11	30	70	0
12	35	50	15
13	40	40	20
14	10	80	10
15	30	70	0
16	30	30	40
17	30	30	40
18	20	70	10

In Table 4, this case project includes 18 activities. Each software engineering network plan had 18 sub-tasks to complete. The spatial dimension of each particle was 18. Each dimension represented one sub-task. Matlab software was used for network planning experiments. The four-dimensional matrix was used to represent the corresponding data of activities Z= [T, C, R, Q]. Each dimension represented the indicator values of the project duration, cost, resources, and quality of each activity. According to the expert experience method, the swarm particles were 100, the size of the POSS was 30, and the iterations were 200. The relationship results of the project duration, cost, resources, and quality are shown in Figure 8.

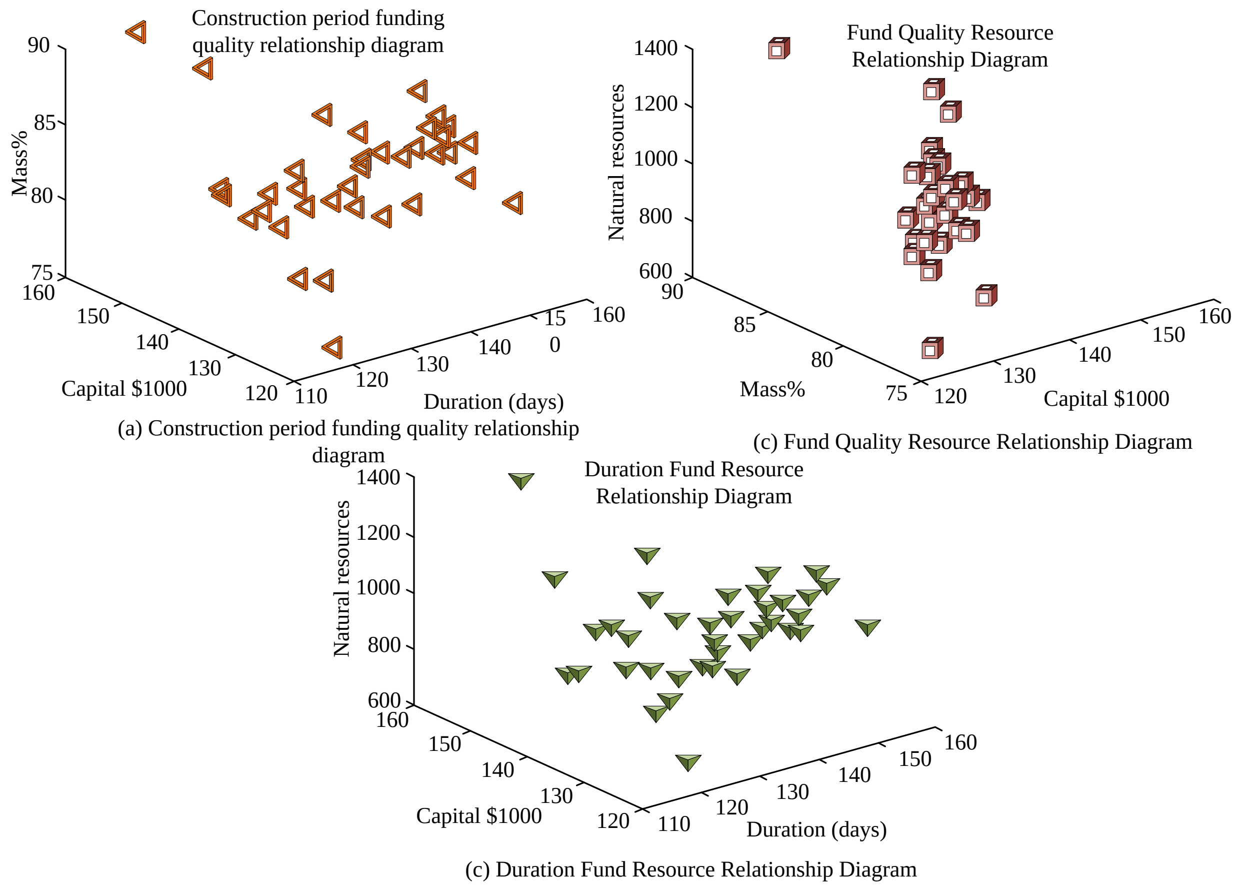


Fig. 8. Duration cost resources quality relationship diagram

Figure 8(a) shows the relationship of project duration-funding quality. Among the three indicators, there was a strong correlation between quality and funding. The indicators were mainly clustered between 130-150 ($1000) in funds, 80-90 (%) in quality, and 120-150 (days) in project duration. Figure 8(b) shows the relationship of funds-quality sources. Among them, quality and funding indicators are relatively concentrated, mainly concentrated in the 75-80(%) in quality, 120-130 ($1000) in funds, and 800-1200 in resources. Figure 8(c) shows the relationship of project duration-funds resources. The indicators mainly focused on funds ranging from 120-140 ($1000), resources ranging from 800 to 1200, and project durations ranging from 120 to 150. The algorithm proposed in the research was applied to practical projects to find the optimal solution. The applied engineering case project required a project duration of less than 125 days. The network planning scheme that met the requirements was selected from the experimental results in Figure 8. The combined scheme is shown in Table 5.

Table 5
Results of software engineering supervision network plan combination scheme

Programme	Duration days	Capital $1000	Mass %	Average resource variance	Scheme combination mode
1	116	138.95	83.826	870.62	{152311315112311211}
2	117	139.26	83.372	872.27	{252311315112311211}
3	120	12Fig 5.3.25	75.963	693.72	{213342323121321112}
4	121	136.25	82.364	1031.64	{412242121131131312}
5	121	139.54	81.318	979.26	{342321212232331112}
6	124	129.05	78.125	826.38	{513242251132111521}
7	124	137.25	80.642	973.24	{122321343211221423}
8	124	150.26	88.69	1069.35	{311231155114121211}
9	124	150.51	88.919	1071.65	{311231145114121211}
10	125	134.25	77.82	723.56	{213342323121321112}

In Table 5, 10 supervision schemes can be combined based on the experimental results. Most of these network plans achieved project quality of over 80%. The lowest cost option 3 was 123.25 $1000. The scheme with the highest engineering quality was Scheme 9, which was 88.91%. There were plans 8 and 9 for projects with a quality of over 85%. There were plans 1, 2, 4, 5, 7, 8, and 9 with engineering quality exceeding 80%. The scheme with the lowest average resource variance was Scheme 10, which was 723.56. If the actual engineering project required high quality requirements, options 8 and 9 were chosen arbitrarily. The difference in construction period, resources, and funding between the two schemes was relatively small. If the quality of the engineering project only needed to meet the 80%, options 1 and 2 were selected. These two plans had relatively low construction period and resource consumption while ensuring quality requirements. The improvement methods studied can be extended to MOO problems in other fields, such as project management, engineering scheduling, supply chain management, etc. These fields all have multiple optimization objectives that need to be considered simultaneously and often require optimizing resource allocation, time scheduling, cost control, and other aspects of network planning. Therefore, the optimization method based on MOPSO proposed in this study can provide reference and inspiration for network planning optimization in these fields. When applying this algorithm to large-scale engineering, its scalability becomes a key issue. For large-scale engineering, it is possible to consider optimizing the parameters in the algorithm, such as the number of particles in the population, the size of POSS, and the number of iterations, to meet the needs of problems of different scales. Based on the data characteristics of large-scale engineering, more efficient computing methods can be studied, such as using parallel computing, distributed computing, and other technologies to improve the computational speed of algorithms in large-scale engineering. To further validate the effectiveness of the algorithm, the experiment applied it to different practical scenarios. The evaluation of the application effect of the IDMPSO algorithm in different scenarios is shown in Table 6.

Table 6
Evaluation of the application effect of algorithm

Actual scenario	Accuracy (%)	Recall (%)	F1 value (%)
Energy [24]	91%	93%	92%
Medical treatment [25]	88%	89%	79%
Network security [26]	89%	92%	81%

In Table 6, the accuracy, recall, and F1 values of the IDMPSO algorithm in energy scenarios are 91%, 93%, and 92%, respectively. In the medical service scenario, the accuracy, recall, and F1 values of the algorithm were 88%, 89%, and 79%, respectively. In network security scenarios, the accuracy, recall, and F1 values of the algorithm were 89%, 92%, and 81%, respectively. The results showed that the algorithm had a good application effect in energy and network security scenarios, but its performance was slightly insufficient in medical service scenarios.

5. Conclusion

With the continuous development of software engineering projects, software engineering supervision is also receiving increasing attention. Especially for large-scale software engineering projects, software engineering supervision is indispensable. The optimization problem of software engineering supervision network plan involves multiple factors such as construction period, cost, resources, quality, etc. The traditional single-objective method used for network supervision optimization may not be consistent with actual engineering. By improving the construction method of POSS, global extremum selection method, and fitness value determination method of DMPSO, a MOO model based on resources, duration, cost, and quality is established for network engineering supervision, improving the convergence and distribution. At the same time, a standard detection function is introduced to detect the convergence and distribution. The improved IDMPSO is applied to engineering project cases to verify the practicality of the algorithm in software engineering supervision network planning problems. Based on the experimental results, firstly, the Pareto optimal solution curves obtained by the proposed IDMPSO on ZDT1, ZDT3, and ZDT6 functions were consistent with the actual theoretical Pareto frontier curves, verifying the effectiveness of the IDMPSO. Secondly, it was applied to engineering project cases. 10 solutions that met the schedule requirements were obtained. The engineering quality of most schemes was above 80%. The algorithm achieved better solutions in optimizing engineering quality by comprehensively considering indicators such as resources, costs, and duration. The limitation of the research is that it only considered four aspects: project duration, cost, resources, and quality, and did not involve other important issues such as safety. Future work will expand on this basis and conduct research on network plan optimization problems that exceed four objectives. The global extremum selection strategy should be optimized to reduce its impact on the algorithm and the impact of other operators on the MOPSO algorithm should be further explored to propose more improved algorithms.

Author Contributions

Conceptualization, methodology, formal analysis, writing-original draft preparation: P.Y.; data curation, resources, writing—review and editing: Z.W. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data Availability Statement

The datasets generated or analyzed during this study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare no potential conflict of interests.

References

[1] Rabbani, M., Oladzad-Abbasabady, N., & Akbarian-Saravi, N. (2022). Ambulance routing in disaster response considering variable patient condition: NSGA-II and MOPSO algorithms. Journal of Industrial & Management Optimization, 18(2), 1035-1062. https://doi.org/10.3934/jimo.2021007

[2] Habib, H., Menhas, R., & McDermott, O. (2022). Managing engineering change within the paradigm of product lifecycle management. Processes, 10(9), 1770. https://doi.org/10.3390/pr10091770

[3] Eito-Brun, R., Gómez-Berbís, J. M., & de Amescua Seco, A. (2022). Knowledge tools to organise software engineering data: Development and validation of an ontology based on ECSS standard. Advances in Space Research, 70(2), 485-495. https://doi.org/10.1016/j.asr.2022.04.052

[4] Hasani, A., Mokhtari, H., & Fattahi, M. (2021). A multi-objective optimization approach for green and resilient supply chain network design: a real-life case study. Journal of Cleaner Production, 278, 123199. https://doi.org/10.1016/j.jclepro.2020.123199

[5] Ye, X., Chen, B., Jing, L., Zhang, B., & Liu, Y. (2019). Multi-agent hybrid particle swarm optimization (MAHPSO) for wastewater treatment network planning. Journal of environmental management, 234, 525-536. https://doi.org/10.1016/j.jenvman.2019.01.023

[6] Tun, H. M. (2021). Radio network planning and optimization for 5G telecommunication system based on physical constraints. Journal of Computer Science Research, 3(1), 1-15. https://doi.org/10.30564/jcsr.v3i1.2701

[7] Zeidan, M., Li, P., & Ostfeld, A. (2021). DMA segmentation and multiobjective optimization for trading off water age, excess pressure, and pump operational cost in water distribution systems. Journal of Water Resources Planning and Management, 147(4), 04021006. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001344

[8] Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. Journal of Parallel and Distributed Computing, 142, 36-45. https://doi.org/10.1016/j.jpdc.2020.03.022

[9] Xu, G., Luo, K., Jing, G., Yu, X., Ruan, X., & Song, J. (2020). On convergence analysis of multi-objective particle swarm optimization algorithm. European Journal of operational research, 286(1), 32-38. https://doi.org/10.1016/j.ejor.2020.03.035

[10] Yuen, M. C., Ng, S. C., & Leung, M. F. (2020). A competitive mechanism multi-objective particle swarm optimization algorithm and its application to signalized traffic problem. Cybernetics and Systems, 52(1), 73-104. https://doi.org/10.1080/01969722.2020.1827795

[11] Rasoulzadeh, M., Edalatpanah, S. A., Fallah, M., & Najafi, S. E. (2022). A multi-objective approach based on Markowitz and DEA cross-efficiency models for the intuitionistic fuzzy portfolio selection problem. Decision Making: Applications in Management and Engineering, 5(2), 241-259. https://doi.org/10.31181/dmame0324062022e

[12] Nafei, A., Huang, C. Y., Chen, S. C., Huo, K. Z., Lin, Y. C., & Nasseri, H. (2023). Neutrosophic Autocratic Multi-Attribute Decision-Making Strategies for Building Material Supplier Selection. Buildings, 13(6), 1373. https://doi.org/10.3390/buildings13061373

[13] Nafei, A., Huang, C. Y., Azimi, S. M., & Javadpour, A. (2023). An optimized model for neutrosophic multi-choice goal programming. Miskolc Mathematical Notes, 24(2), 915-931. https://doi.org/10.18514/MMN.2023.4020

[14] Akram, M., Shah, S. M. U., Al-Shamiri, M. M. A., & Edalatpanah, S. A. (2023). Extended DEA method for solving multi-objective transportation problem with Fermatean fuzzy sets. Aims Math, 8, 924-961. https://doi.org/10.3934/math.2023045

[15] Mekawy, I. M. (2022). A novel method for solving multi- objective linear fractional programming problem under uncertainty. Journal of Fuzzy Extension and Applications, 3(2), 169-176. https://doi.org/10.22105/jfea.2022.331180.1206

[16] Farnam, M., & Darehmiraki, M. (2021). Solution procedure for multi-objective fractional programming problem under hesitant fuzzy decision environment. Journal of Fuzzy Extension and Applications, 2(4), 364-376. https://doi.org/10.22105/jfea.2021.288198.1152

[17] Ghasemi, P., Hemmaty, H., Pourghader Chobar, A., Heidari, M. R., & Keramati, M. (2023). A multi-objective and multi-level model for location-routing problem in the supply chain based on the customer's time window. Journal of Applied Research on Industrial Engineering, 10(3), 412-426. https://doi.org/10.22105/jarie.2022.321454.1414

[18] Liu, Z., Xiang, B., Song, Y., Lu, H., & Liu, Q. (2019). An improved unsupervised image segmentation method based on multi-objective particle swarm optimization clustering algorithm. Computers, Materials & Continua, 58(2), 451-461. https://doi.org/10.32604/cmc.2019.04069

[19] Shuxiao, M., Yun, T., Binhe, C., & Yibo, Z. (2021). Optimization of Development Intensity Index of Regulatory Land under the Constraint of Bearing Capacity of Road Network: A Case Study of Xingtang County. Journal of Landscape Research, 13(1), 73-80.

[20] Hajgató, G., Paál, G., & Gyires-Tóth, B. (2020). Deep reinforcement learning for real-time optimization of pumps in water distribution systems. Journal of Water Resources Planning and Management, 146(11), 04020079. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001287

[21] Aikhuele, D. (2023). Development of a statistical reliability-based model for the estimation and optimization of a spur gear system. Journal of Computational and Cognitive Engineering, 2(2), 168-174. https://doi.org/10.47852/bonviewJCCE2202153

[22] Choudhuri, S., Adeniye, S., & Sen, A. (2023). Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation. Artificial Intelligence and Applications, 1(1), 43-51. https://doi.org/10.47852/bonviewAIA2202524

[23] Devi Priya, R., Sivaraj, R., Abraham, A., Pravin, T., Sivasankar, P., & Anitha, N. (2022). Multi-Objective Particle Swarm Optimization Based Preprocessing of Multi-Class Extremely Imbalanced Datasets. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 30(05), 735-755. https://doi.org/10.1142/S0218488522500209

[24] Salehi, N., & Askarzadeh, H. R. (2018). Optimum solar and wind model with particle optimization (PSO). International Journal of Research in Industrial Engineering, 7(4), 460-467. https://doi.org/10.22105/riej.2018.148836.1059

[25] Dirik, M. (2022). Type-2 fuzzy logic controller design optimization using the PSO approach for ECG prediction. Journal of fuzzy extension and applications, 3(2), 158-168. https://doi.org/10.22105/jfea.2022.333786.1207

[26] Rajeshkumar, G., Kumar, M. V., Kumar, K. S., Bhatia, S., Mashat, A., & Dadheech, P. (2023). An Improved Multi-Objective Particle Swarm Optimization Routing on MANET. Computer Systems Science & Engineering, 44(2), 1187-1200. https://doi.org/10.32604/csse.2023.026137