Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 4, pp. 3062–3073, DOI 10.12694/scpe.v25i4.2990

PARAMETRIC DESIGN OF OFFICE FURNITURE PARTITION SPACE INTEGRATED WITH THE INTERACTIVE EVOLUTION ALGORITHM OF FNT AND TREE STRUCTURE

SHIDONG CHEN* AND HUIYUAN GUAN †

Abstract. Office furniture and its spatial layout design are playing an increasingly important role in improving work efficiency and employee comfort. However, the technology still faces some challenges. For instance, accurately simulating and evaluating the behavior and feelings of people in the office environment is difficult due to the high complexity of furniture spacing space design. It is important to address these issues. The study aims to explore the key technology and practical application of the parametric design of office furniture partition space based on the interactive evolution algorithm of tree structure. This paper proposes an improved version of the flexible neural tree model and corresponding algorithm. It also presents a design method based on the interactive differential evolution algorithm to optimize the automatic balance effect between global exploration and local development in the average shortening of the difference vector based on individual distribution. The results showed that all indexes were larger than or equal to other algorithms on 46 datasets. According to the Wilcoxon signed-rank test, the P-value was all less than 0.05, which is a significant advantage. Median, mean, and quartiles indicated that the overall performance of the algorithm was higher than the others. Furthermore, similarity evaluation-based Flexible Neural Tree algorithm had no outliers in the selected dataset, which also indicates the stability of the performance. The research results will support innovation and development in the field of office furniture design. This will promote intelligence, efficiency, and personalization in the design process, and meet the diverse needs of modern office environments.

Key words: Flexible neural tree; Similarity assessment; Interactive evolution; Indoor space; Parameterization

1. Introduction. In recent years, computer-aided design technology has made parametric design a hot topic in the fields of architecture and interior design. This technology provides designers with unprecedented flexibility and innovation. Office space serves not only as a place for employees to work but also as an important location for promoting communication, cooperation, and innovation [1-2]. Parametric design enables designers to explore creativity within pre-defined parameter spaces. Traditional parametric design methods still face problems such as low efficiency and unstable results when dealing with complex office environments. Interactive evolutionary algorithm is a computational method that draws inspiration from biological evolution mechanisms, which can adaptively search for the optimal or near optimal solution in the design space [3-4]. The Flexible Neural Tree (FNT) structure can better describe the relationship between office furniture and its configuration, thereby achieving more efficient and accurate design search and optimization. Through this method, designers can not only quickly explore design solutions that meet specific needs, but also continuously explore and optimize creativity based on algorithm feedback [5-6]. Therefore, this study proposes a parameterized design method for office furniture partitions that combines tree structure Interactive Differential Evolution (IDE) algorithm. The aim of this study is to provide designers with an efficient and flexible design tool, enabling office furniture design to meet functional needs while demonstrating outstanding aesthetic value.

The research innovatively combines the FNT structure and the IDE algorithm based on similarity evaluation to realize the parametric design of office furniture spacing space. The FNT model is optimized by introducing the similarity distance between nodes and intertree similarity partial function, improving genetic diversity. At the same time, the improved FNT is applied to the IDE algorithm to replace the complex scoring mechanism to

*College of Furnishings and Industrial Design, Nanjing Forestry University, Nanjing, 210037, China. School of Mechanical and Engineering, Yangzhou University, YangZhou, 225127, China.

†College of Furnishings and Industrial Design, Nanjing Forestry University, Nanjing, 210037, China. Co-innovation Center of Efficient Processing and Utilization of Forest Resources, Nanjing Forestry University, Nanjing 210037, China (stone8267@163.com)

realize the automatic balance between global exploration and local development. Moreover, an IDE algorithm based on the backtracking strategy is proposed to solve the problem of Local Optimal State (LOS).

The first part of the study presents the purpose and provides a literature review. The second part proposes a parametric design method of office furniture partition combined with IDE structure. The third part is algorithm performance testing and application analysis. The fourth part summarizes the method and experimental results and draws the research conclusion.

2. Related works. The principles of spatial layout design have been widely applied in modern engineering and daily life. Keshavarzi M et al. proposed three new layout methods through the interactive design system GenFloor and implemented them as part of the Dynamo software package. Experiments had shown that the GenFloor system performed well in residential floor plan planning tasks [7]. To optimize the layout of the sewer network, Hassan W H et al. proposed a Hybrid Genetic Algorithm (HGA), which solves the problem by generating feasible layouts of the network and determining the optimal design from them. Compared with previous methods, this model could achieve the optimal solution with the minimum number of generations [8]. Dong et al. proposed an automatic solution method for Ship Piping Route Design. The feasibility and effectiveness of this method were verified by combining A* algorithm and Genetic Algorithm (GA) [9]. Feng et al. proposed an effective B-spline parameterization method to improve the layout of stiffeners for the optimization of thin-walled shell structures in aerospace engineering. This method could produce design results with clear layout and no checkerboard effect [10]. To reduce the manual work of indoor floor plan design, He S et al. used a differentiable renderer as an optimizer to achieve design optimization by adjusting the parameters of grid primitives. The optimization process was constrained by input boundaries, overlap, and weight constraints to ensure parameterization of the design and reasonable spatial partitioning. The effectiveness of this method in floor plan design was verified through ablation and control experiments [11].

Tan et al. studied the existing research work of Evolutionary Transfer Optimization (ETO) in multitasking, complex, multi/multi-objective optimization, and machine learning applications in uncertain environments. At the same time, the article discussed the challenges faced in computational intelligence and explored the future research directions of ETO [12]. Zhang et al. proposed an Adaptive and Scalable Neural Structure Search method (AS-NSS), which utilizes the enhanced International Data Encryption Algorithm and a variable structure encoding strategy. The effectiveness and superiority of the AS-NSS method had been demonstrated [13]. Deng et al. proposed a population generation method based on target space, which generates new individuals in the target space and maps them to the decision variable space to synthesize new solutions. The algorithm outperformed traditional algorithms in large-scale decision variables and multi-objective problems [14]. Al Fugara et al. used an integrated model of support vector regression and GA to map groundwater spring potential for the Jerash and Agilent regions of Jordan. This study provided a new integrated model for the drawing of groundwater spring potential maps, with high accuracy and reliability [15]. Yi et al. proposed a new interval multi factor evolutionary algorithm to overcome the limitation of solving optimization tasks and simultaneously solve multi-objective optimization problems. This method had effectiveness and performance advantages in benchmark testing functions and robot path planning cases [16].

Although the studies mentioned above have yielded results, they still have limitations. Some methods rely too heavily on specific algorithms and are less universal. For instance, methods like GA, HGA, and neural structure search may incur high computational costs and long solution times when dealing with complex problems, particularly in scenarios with large-scale decision variables or multiple objectives. Applying parametric design to furniture spacing allows for the automatic generation of multiple layout schemes in a rule and parameter-driven manner, enabling rapid iteration and optimization when combined with IDE. This study focuses on the parametric design of office furniture spacing and explores the application of IDE structure. This research level and methodology is in sharp contrast to the research areas and methods of the literature, demonstrating unique innovations.

3. Parametric design of office furniture partition space based on tree structure IDE. The study first discusses the FNT structure based on similarity evaluation and applies it to the IDE algorithm to achieve parameterized design of furniture spacing space. The algorithm achieves automatic balance between global exploration and local development during the process of average shortening of the difference vector based on individual distribution.

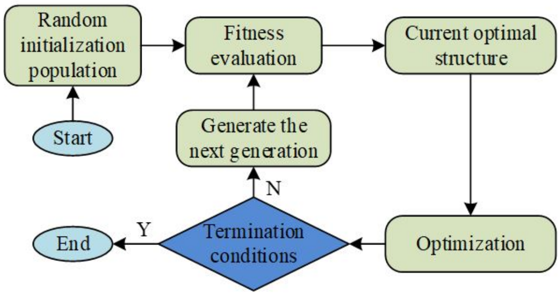

Fig. 3.1: The overall evolutionary framework of FNT

3.1. FNT structure based on similarity evaluation. Compared with conventional neural networks, FNT can find the optimal network structure by utilizing evolutionary algorithms. Hence, the modeling process of FNT optimization has two aspects: structure and parameter, as displayed in Figure 3.1.

In the FNT model, leaf nodes correspond to the input nodes of the neural network and are used to represent a certain feature in the dataset. Non leaf nodes (NLN) represent hidden layers in neural networks, representing an activation function. Therefore, two instruction sets are proposed for these 2 kinds: terminal and functional instruction sets, T and F. T has all the dataset features, while F owns the neurons' operating instructions, which together make up the set S for adjusting the tree structure, as shown in equation (3.1).

$$S = F \bigcup T = \{+_2, +_3, \cdots, +_n\} \bigcup \{x_1, x_2, \cdots, x_n\} \tag{3.1}$$

In equation (3.1), $+_n$ is the instruction function of a NLN with n parameters. $x_1, x_2, \cdots, x_n$, fitting to the feature meaning in the dataset, is the instruction of the leaf node. The output results of NLN of FNT need to be calculated based on FNT. $+_i$ is also represented as a flexible neural operator with i inputs. During creating FNT, an instruction is randomly selected from the F and T sets as a node. Besides, two optimized random parameters a_i and b_i are generated as parameters for the activation function. This commonly used function in the FNT model is equation (3.2).

$$f(a_i, b_i, x) = e^{-(\frac{x-a_i}{b_i})^2} \tag{3.2}$$

The input calculation of flexible neuron $+_i$ is equation (3.3).

$$net_n = \sum_{j=1}^{n} w_j \cdot x_j \tag{3.3}$$

In equation (3.3), leaf node $x_j (j = 1, 2, \cdots, n)$ is the input of NLN $+_n$. The output calculation of $+_n$ is equation (3.4).

$$out_n = f(a_i, b_i, net_n) = e^{-(\frac{net_n - a_i}{b_i})^2} \tag{3.4}$$

Figure 3.2 shows a classic FNT model that calculates recursively from left to right using a depth-first traversal method to obtain the overall output of FNT.

This study utilizes Particle Swarm Optimization (PSO) algorithm to optimize the parameters of FNT. In the FNT model, each node has three parameters, namely activation function parameters a_i, b_i, and the weights ω representing the strength of the connection with the parent node (PN). The range of values for these parameters is a real number between [0,1]. During the parameter optimization process, FNT nodes are traversed in the order of the root traversal. The parameters of each node are sequentially encoded and ultimately represented

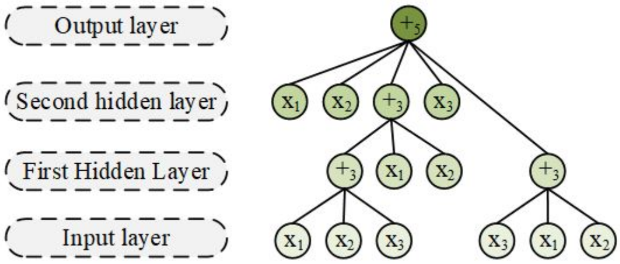

Fig. 3.2: Classic FNT model

as a particle vector in PSO. This study uses the standard PSO formula to update the velocity and position (V-P) of particles. The speed of particle updates is equation (3.5).

$$v_i(t+1) = \omega \times v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t)) \tag{3.5}$$

The position of particle updates is equation (3.6).

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{3.6}$$

In equations (3.5) and (3.6), ω is the inertia weight, and c_1 and c_2 are the learning factors. r_1 and r_2 are two random numbers that obey a uniform distribution within the range of [0,1]. $v_i(t)$ and $x_i(t)$ are the V-C of the i-th particle in the t-th generation. $v_i(t+1)$ and $x_i(t+1)$ are the iteration results for the $t+1$-th generation. $pbest_i$ represents the historical optimal position of the particle i during the current iteration process. $gbest$ represents the optimal position in the global particle record.

The standard FNT dose not consider the similarity between tree structures during the evolution process, which may significantly reduce the genetic diversity of tree structure populations [17]. Therefore, this study proposes an enhanced version of the FNT model. The Similar Distance between Nodes (SDN) function is a distance function proposed based on FNT nodes. For any two random nodes a and b of FNT T_1 and T_2, both have $a \in \{x_1, x_2, \cdots, x_n, +_2, +_3, \cdots, +_N\}$ and $b \in \{x_1, x_2, \cdots, x_n, +_2, +_3, \cdots, +_N\}$. This study uses $\varphi(a,b)$ to represent the a-b distance of nodes, as shown in equation (3.7).

$$\varphi(a,b) = \begin{cases} a \oplus b, \ \ if \ \ a \in T, b \in T \\ n(b) \times std(b), \ \ if \ \ a \in T, b \in F \\ h^{\alpha} \times \sum \varphi(a_c, b_c), \ \ otherwise \end{cases} \tag{3.7}$$

In equation (3.7), the function $n(x)$ is utilized to calculate the quantity of leaf nodes of x. $std(y)$ is used to measure the high number of words in the root node. h represents the maximum height of numbers T_1 and T_2. a_c and b_c are sub-trees of a and b, respectively. α can reflect the impact caused by the height difference between nodes. When two nodes are the same height, α=0, otherwise α=1.

Considering the differences in the position of nodes in the tree structure, the differences brought by nodes closer to the root node may be more significant than those far away from the root node. Two factors need to be considered, namely the size and height of the tree [18-19].

As exhibited in Figure 3.3, this study introduces two ideas when testing the trees' distance and combines them with SDN. Firstly, nodes near the root node have higher weights when calculating distance. Secondly, regarding the similar distance between two trees, various heights should be greater than the same height. The representation of the Similarity Dist between Trees (SDT) function is as follows. Supposing there is a tree $T_1 = \{st_1(T_1), st_2(T_1), \cdots, st_m(T_1)\}$, where T_1 represents the root node and $st_1(T_2), st_2(T_2), \cdots, st_n(T_2)$ represents the presence of m sub-trees. In another tree $T_2 = \{st_1(T_2), st_2(T_2), \cdots, st_n(T_2)\}$, T_2 represents the

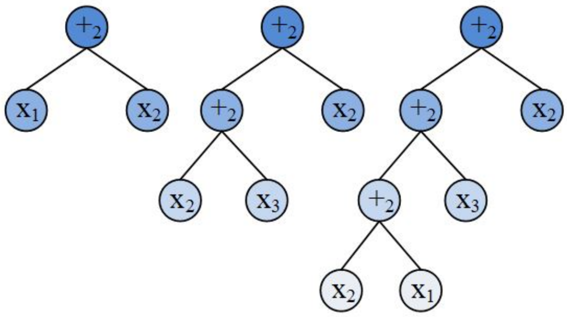

Fig. 3.3: Example of the impact of height difference between trees on distance calculation

root node. $st_1(T_2), st_2(T_2), \cdots, st_n(T_2)$ represents having n sub-trees. Therefore, the SDT between T_1 and T_2 is represented by equation (3.8).

$$siml(T_1, T_2) = \begin{cases} \varphi(T_1, T_2), & if\, T_1\, or\, T_2\, is\, a\, single\, node \\ C^{\Delta h} \times k \times \sum_{i=1}^{\min(m,n)} \varphi(s_i(T_1), s_i(T_2)), & otherwise \end{cases} \tag{3.8}$$

The calculation method for k in equation (3.8) is equation (3.9).

$$k = \frac{1}{1 + e^{-h}} \tag{3.9}$$

In equation (3.9), k is the weight of the node, and C is a constant.

The setting of fitness functions is crucial for the performance of evolutionary algorithms. Generally, a smaller fitness function value indicates better performance, while a larger AUC value indicates more accurate classification. Therefore, the non equilibrium fitness function proposed in the study is equation (3.10).

$$fitness(i) = 1 - AUC_i \tag{3.10}$$

In equation (3.10), i is the i-th individual in the population. AUC_i represents the classification performance when using the i to classify the dataset. This study proposes a Similarity Evaluation-based Flexible Neural Tree algorithm (SEFNT) by integrating SDN, SDT function, and imbalanced fitness function. The SEFNT algorithm first calculates the fitness values of all individuals in each generation of FNT population, and sorts them in ascending order according to the size of the fitness values. Then, individuals in adjacent positions are grouped in pairs. The structural similarity of each group of individuals is calculated and determined whether to delete them based on the preset similarity criteria. Finally, a new population is obtained after deleting similar individuals.

3.2. Parameterized design of furniture spacing space based on IDE. The IDE algorithm is an evolutionary algorithm that replaces the scoring mechanism. In the IDE, users only need to choose a more satisfactory individual between two individuals, without the need for complex comparative scoring processes. The IDE algorithm utilizes information on individual differences and indirect population distribution size to search for offspring of parent individuals around a base individual. The parent individual will only be replaced by the offspring if they exceed it in fitness. During the optimization process of the algorithm, the individual distribution gradually shrinks, and the difference vector shortens on average according to the individual distribution, thus achieving automatic balance between global exploration and local development. The standard IDE algorithm includes four steps: population initialization, crossover, mutation, and selection [20-22].

Firstly, here is a specific method for providing the initial population, which is represented by equation (3.11).

$$x_{j.i.0} = x_i^{\min} + (x_i^{\max} - x_i^{\min}) \times rand, \;\; \forall i \in 1, \cdots, D; \;\; \forall j \in 1, \cdots, N \tag{3.11}$$

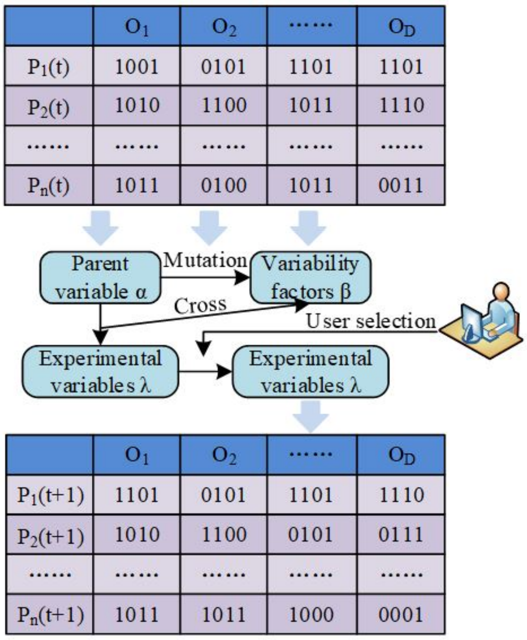

Fig. 3.4: The iterative process of IDE

In equation (3.11), $x_{j,i,0}$ is the gene in the i part of the chromosome of the j-th individual. $x_i^{\min}$ and $x_i^{\max}$ represent the lower and upper bound of the i-th component. $rand$ is a random number that follows a uniform distribution of [0,1]. In each iteration, a mutation vector is produced through mutation operations. The latter is vital in performing the differential evolution algorithms. The classic mutation operator $DE/rand/1$ is equation (3.12).

$$V_{j,t} = X_{r1,t} + F_j \times (X_{r2,t} - X_{r3,t}) \tag{3.12}$$

In equation (3.12), $X_{j,t}$ represents the j-th individual during the t-th iteration. $V_{j,t}$ is the related mutation factor. r_1, r_2 and r_3 are mutually exclusive integers chosen from array $[1, 2, 3, \cdots, N]$ at random and are not equal to j. F_j is the proportion factor. Then cross operation is performed to obtain the experimental vector. The crossover operator is calculated by equation (3.13).

$$U_{j,i,t} = \begin{cases} V_{j,i,t}, & if \ \ rand < CR \ and \ i = i_{rand} \\ X_{j,i,t}, & otherwise \end{cases} \tag{3.13}$$

In equation (3.13), $U_{j,i,t}$ is the i component of the experimental vector of the j individual during the t iteration. CR is the crossover rate. i_{rand} is a random number from $[1, 2, \cdots, D]$. After obtaining the test vector, a selection operator is used to choose the greater one from the PN and test vector to survive to the next generation. The operation is selected as shown in equation (3.14).

$$X_{j,t+1} = \begin{cases} U_{j,t}, & if \ f(U_{j,t}) < f(X_{j,t}) \\ X_{j,t}, & otherwise \end{cases} \tag{3.14}$$

Figure 3.4 is the IDE's iterative diagram.

As a heuristic algorithm, the differential evolution algorithm is essentially a greedy strategy. Selective pressure causes the population to gradually move towards the higher fitness region, which can cause the algorithm

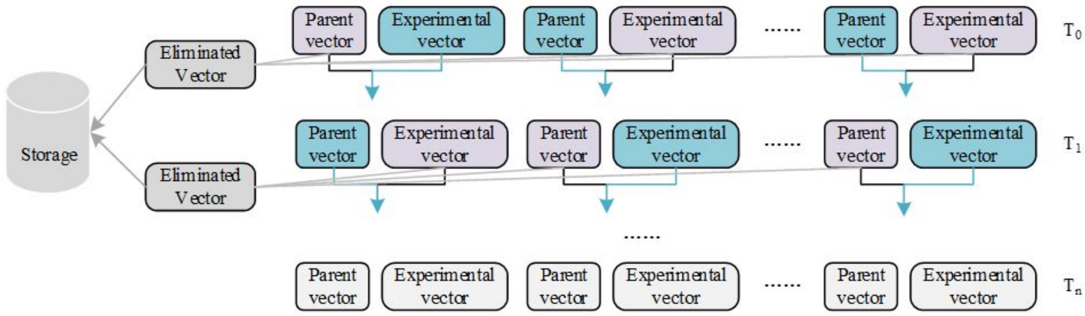

Fig. 3.5: The iterative process of IDE-BO

to converge prematurely to the local optimal solution. If the individuals in the population are too similar, the resulting difference vector may not be sufficient to lead the algorithm out of the local optimal region. To address this issue, this study introduces an IDE based on the backtracking strategy (BS-based) to address the LOS by gradually returning to the previous choice.

To address the shortcomings of greedy selection operators in LOS, the algorithm adopts a new BS-based selection operator. The first is to determine whether an individual has fallen into a LOS. When an individual slowly updates within a certain interval, it is in a stagnant state. It may have a certain rate of misjudgment but effective for determining if an individual has fallen into a LOS and can develop strategies to low down the misjudgment effect. The new operator utilizes backtracking and establishes a spatial warehouse in each iteration to store individuals that failed in the current round of competition and were discarded. When an individual is detected falling into a LOS, a series of vectors are continuously extracted from the storage space to survive to the next generation. To avoid calculating the fitness of random vectors and increase computational complexity, the PN is substituted with a discarded experimental vector. These vectors will not be passed on to the later generation, but individuals evolved built on them are more likely to aid individuals to overcome. Figure 3.5 is the process of IDE-BO.

4. Analysis of the parametric design effect of office furniture spacing space based on IDE-SEFNT algorithm. To verify the effectiveness of differential evolution algorithm and GA in indoor layout design, this study conducted simulation experiments. All eigenvalues are normalized to intervals (0.0, 1.0) during the experiment. To pre set parameters, with a number of individuals NP=12 for each iteration, a jump rate Jr=0.3, a scaling factor F=0.7, and a crossover probability CR=0.5. In the simulation experiment, it is assumed that the user cannot manually adjust the feature values, but can have a clear overall understanding of the layout design generated by the algorithm and accurately select the layout that suits their preferences. The target individual is set that the user is most satisfied with $O(o_1, o_2, \cdots, o_D)$. In the system, the distance between the individual and the satisfactory solution is considered as the fitness value. The experimental template and parameter values are shown in Figure 4.1.

This study selected 10 publicly available datasets from the UCI machine learning library as experimental data for performance testing of the SEFNT algorithm. The experiment divided the data into 10 equal parts, each containing two parts: the training and the test sets. The imbalanced rate (IR) is the sample ratio between the majority and minority class, indicating the imbalance degree. In the experiment, datasets with IR<9 and IR≥9 were considered low-imbalanced and high-imbalanced datasets. Table 4.1 is the UCI dataset.

The experiment compares the SEFNT algorithm with ten other algorithms. These algorithms can be divided into 9 non equilibrium methods and 1 standard FNT method. Among them, the imbalanced method includes 4 oversampling algorithms, 3 cost sensitive algorithms, and 2 ensemble algorithms. Non equilibrium algorithms are all based on random forests as the basic classifier. The AUC and GM experimental results of different algorithms on low and high imbalanced datasets are shown in Figure 4.2.

In Figure 4.2, the maximum, minimum, median, mean, and quartile of AUC/GM results of the SEFNT

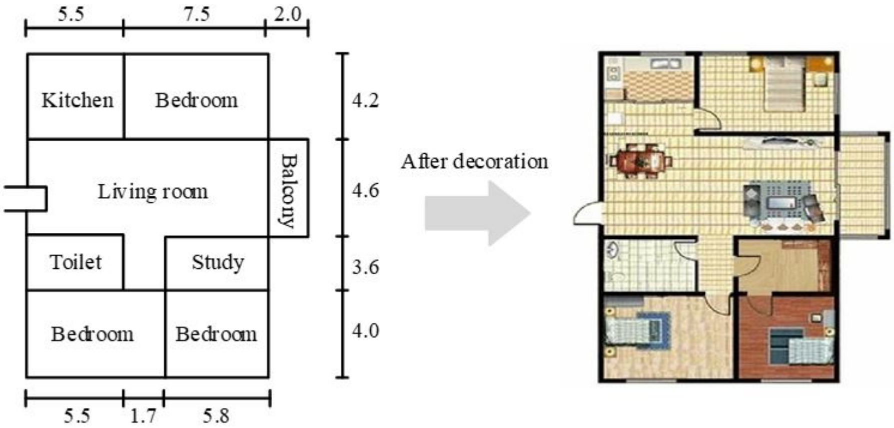

Fig. 4.1: Experimental template and parameter values

Table 4.1: UCI dataset

Dataset	Number of samples	Characteristic number	IR
Chesscking-Rook vs. King	28056	6	113.05
letter-recognition	20000	16	24.35
HTRU2	17898	9	67.44.
musk	6598	168	5.49
isolet	7797	617	24.99
waveform	5000	21	2.04
page-block0	5472	10	8.79
nursery	12960	8	2.00
parkinsons telemonitoring	5875	26	41.57
sat	6435	36	9.28

on 46 datasets are greater than or equal to those of other algorithms. The max and min values reflect the good performance of the SEFNT algorithm on certain datasets. The median, mean, and quartile indicate that the overall performance of the SEFNT algorithm is higher than other algorithms. In addition, SEFNT has no outliers in the selected dataset, which also indicates the performance stability. Compared with other algorithms, the SEFNT still performs well on five indicators and has no outliers. Compared to its performance on low imbalanced datasets, the SEFNT algorithm performs slightly better on high imbalanced datasets. Specifically, the box area in the figure is smaller, while the box area of other algorithms increases slightly. When the performance of other algorithms decreases, the SEFNT algorithm can maintain more stability and better performance on highly imbalanced datasets. Table 4.2 is the AUC results of the UCI dataset.

In Table 4.2, the SEFNT has AUC values greater than 0.85 on most datasets. On the Nursery dataset, the AUC of the EE is 0.1668, while the SEFNT's AUC is 0.6295. In addition, the SEFNT algorithm has an AUC value of up to 0.9960 on the Parkinsons Telemonitoring dataset. Table 4.3 is the GM results of the UCI dataset.

In Table 4.3, the SEFNT algorithm has a GM value greater than 0.85 on most datasets. On the Nursery dataset, the EE's GM value is 0.0622, while the SEFNT's GM is 0.6220. In addition, the SEFNT algorithm has a GM value of up to 0.9959 on the Parkinsons Telemonitoring dataset. Overall, the SEFNT has AUC and GM values greater than 0.85 on most datasets. These values indicate that the SEFNT algorithm not only maintains good classification performance, but also has more stable performance.

This study conducted simulation tests on this algorithm and other three. To accurately verify the convergence, the iterations in the simulation are set to 150. However, in real interactive environments, to avoid inaccurate results caused by user fatigue, the maximum number of iterations generally does not exceed 50. The

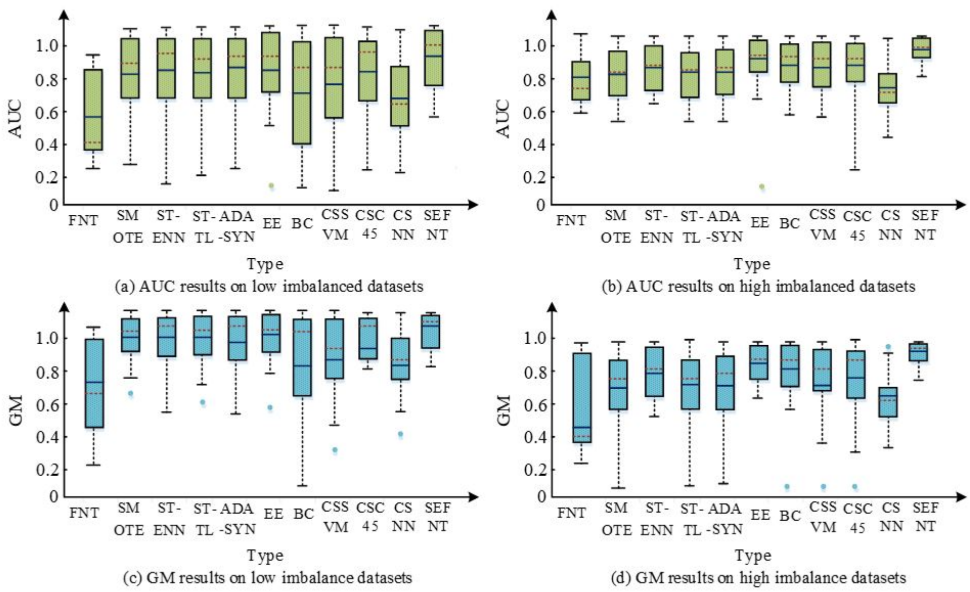

Fig. 4.2: AUC and GM test results

Table 4.2: AUC experimental results on the UCI dataset

Dataset Type	FNT	SMOTE	ST-ENN	ST-TL	ADASYN	EE	BC	CSSVM	CSC4.5	CSNN	SEFNT
Chesscking-Rockvs.King	0.5000	0.7632	0.8183	0.7857	0.7954	0.9758	0.9731	0.972	0.9858	0.6535	0.9849
HTRU2	0.7242	0.9340	0.9372	0.9327	0.9321	0.9427	0.9416	0.9431	0.9304	0.875	0.9445
Isolet	0.5000	0.8915	0.9122	0.8967	0.887	0.9752	0.9692	0.9697	0.9287	0.5000	0.9333
Letter-recognition	0.6513	0.9694	0.9705	0.9703	0.9704	0.9944	0.9931	0.9554	0.9808	0.5509	0.9428
Musk	0.5416	0.7604	0.7552	0.7584	0.7631	0.7853	0.7874	0.9252	0.9437	0.5526	0.829
Nursery	0.5000	-0.3023	0.2075	0.3008	0.2992	0.1668	0.2595	0.5871	0.5871	0.5160	0.6295
Page-block0	0.7348	0.9188	0.932	0.9221	0.9213	0.9397	0.9411	0.9219	0.9491	0.7036	0.8904
Parkinsons telemonitoring	0.5000	0.9785	0.9769	0.9799	0.982	0.9917	0.9867	0.9834	0.9920	0.4082	0.9960
Sat	0.5000	0.7587	0.8200	0.7574	0.7627	0.8571	0.8554	0.7275	0.7841	0.615	0.8543
Waveform	0.7529	0.8714	0.8871	0.8699	0.8713	0.9056	0.9010	0.8981	0.8502	0.8023	0.8943

Table 4.3: GM experimental results on the UCI dataset

Dataset Type	FNT	SMOTE	ST-ENN	ST-TL	ADASYN	EE	BC	CSSVM	CSC4.5	CSNN	SEFNT
Chesscking-Rockvs.King	0.0000	0.7279	0.8004	0.7578	0.7711	0.9757	0.973	0.9716	0.9858	0.6076	0.9848
HTRU2	0.6727	0.9327	0.9363	0.9313	0.9312	0.9423	0.9413	0.9423	0.9298	0.873	0.9441
Isolet	0.0000	0.885	0.9081	0.8909	0.8799	0.9753	0.9692	0.9690	0.9260	0.0000	0.9220
Letter-recognition	0.5566	0.9689	0.9700	0.9699	0.9699	0.9944	0.9931	0.9553	0.9806	0.4674	0.9422
Musk	0.3084	0.7582	0.7535	0.756	0.7578	0.7845	0.7862	0.9251	0.9431	0.3804	0.828
Nursery	0.0000	0.1138	0.1738	0.1070	0.1019	0.0622	0.0566	0.5548	0.4317	0.4582	0.6220
Page-block0	0.6916	0.9172	0.9315	0.9207	0.9199	0.9397	0.9049	0.9217	0.9487	0.6342	0.8901
Parkinsons telemonitoring	0.0000	0.9785	0.9768	0.9798	0.9819	0.9917	0.9866	0.9833	0.9919	0.2560	0.9959
Sat	0.0000	0.7317	0.8161	0.7299	0.7376	0.8569	0.8546	0.7064	0.7701	0.5174	0.8530
Waveform	0.7468	0.8708	0.8851	0.8692	0.8711	0.9040	0.9009	0.8957	0.8498	0.7805	0.8924

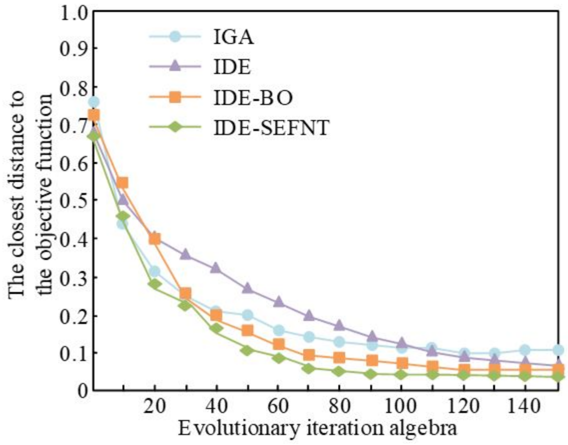

Fig. 4.3: Simulation experiment results

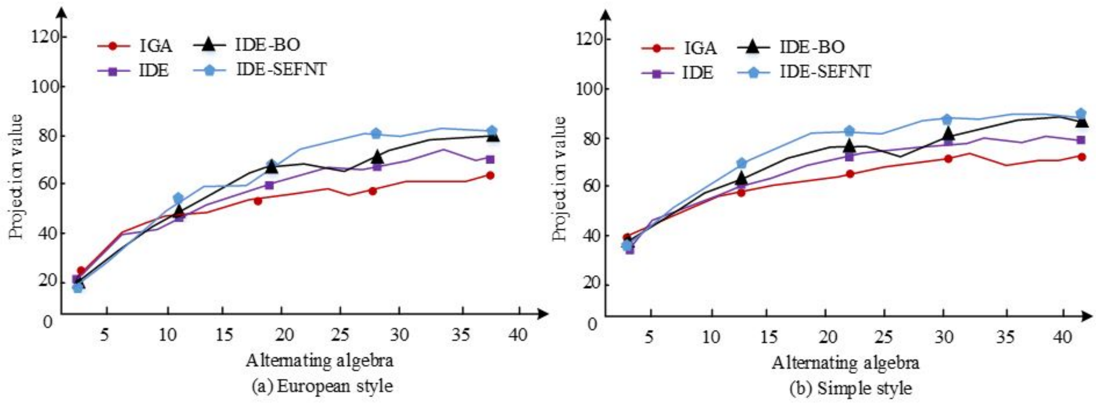

Fig. 4.4: Convergence test results

simulation evaluation results are shown in Figure 4.3.

In Figure 4.3, there is no obvious discrepancy among the four in the early iteration stage. In the later stage, IGA converges early. IDE and IDE-BO own better convergence effects. IDE-SEFNT is the best. When the iterations are 40, the convergence speed and accuracy of IDE-SEFNT are significantly greater than others.

In interactive algorithms, the subjective choices of users play a crucial role in the convergence of the system. Therefore, the convergence testing is used to test the user satisfaction and practicality of testing parameterized office furniture spacing space design schemes in the real world, and study the influence of user subjective choices. To ensure the diversity and representativeness of the results, the study selected 10 testers from different occupational backgrounds, age groups, and usage habits. These testers comprise office employees, designers, managers, and other individuals, to simulate the reactions of diverse user groups to the spacing design of office furniture. Due to the characteristics of the design and art industries, it is difficult to quantitatively describe convergence. Therefore, 10 testers were selected in the experiment and the maximum number of iterations was set to 40 to investigate their satisfaction with the individual. The convergence test results are shown in Figure 4.4 .

In Figure 4.4, at iteration number $t \in [0, 12]$, the four algorithms have no significant impact on the results. However, when t, the IDE algorithm exhibits more excellent convergence ability than IGA. The optimized IDE

Table 4.4: Wilcoxon sign rank test results

Index	Interval of t	Euclidean style				Simple style			
		IGA	IDE	IDE-BO	Ours	IGA	IDE	IDE-BO	Ours
P	[1,10]	0.873	1.000	0.872	0.825	0.378	0.470	0.471	0.764
	[11,30]	0.001	0.093	0.330	0.002	0.005	0.001	0.003	0.003
	[31,40]	0.003	0.003	0.003	0.001	0.004	0.004	0.166	0.002
	[1,40]	0.018	0.110	0.257	0.002	0.004	0.007	0.166	0.001

is easier to run from the LOS, thus achieving higher satisfaction. At the same time, the improved IDE algorithm dose not show a significant improvement in score at high iterations, but it can accelerate the convergence speed, which is beneficial for users to complete the evolution of the algorithm before low iterations and prevent user fatigue. The users' subjective requirements have a significant impact. Diverse style choices can also affect users' subjective judgments, thereby affecting the algorithm convergence. By comparing to "minimalist", the meaning of the term "European" is more complex, making it hard to make accurate judgments. Hence, the "minimalist" style owns more outstanding results.

To compare and study the performance differences between improved and traditional algorithms, Wilcoxon Signed Rank Test (WSRT) was used in the experiment, with a significance level set at 0.05. The WSRT results are shown in Table 4.

The experiment is divided into three stages based on the number of iterations: early stage, mid stage, and late stage. In Table 4, there was inconspicuous distinction in performance among the four algorithms in the previous stage. However, starting from the mid-term stage, the proposed IDE-SEFNT algorithm shows significant advantages, with all P-values less than 0.05 in the "minimalist" style. As the research progresses, the advantages of algorithms become more apparent. In summary, the IDE-SEFNT algorithm has achieved significant performance improvements compared to the other three algorithms.

5. Conclusion. This study has designed an improved version of the FNT algorithm and the IDE algorithm based on similarity evaluation. The structure and parameters of the neural tree were optimized by PSO, and BS was introduced to help the algorithm overcome the LOS. The results showed that the AUC and GM metrics of SEFNT were greater than or equal to the other algorithms on 46 datasets. Moreover, the SEFNT algorithm also showed some advantages in the AUC and GM values of 0.9960 and 0.9959 on the Parkinsons remote monitoring dataset, respectively. In terms of style selection, minimalist style showed superior performance than European style, so the research suggested more minimalist style in future applications. The results of the Wilcoxon signed-rank test also showed that the IDE-SEFNT algorithm has a significant performance advantage over the other algorithms. Especially in the "minimalist" style, the performance was more pronounced, with all P-values less than 0.05. Therefore, this research method has important influence on solving the problems existing in the traditional algorithm and furniture spacing space design and has wide practical applications. The management significance of the research lies in that it can help enterprises to improve work efficiency and employee satisfaction. By optimizing the partition space design of office furniture, the working environment is more comfortable, efficient and flexible, so as to improve the work efficiency and satisfaction of employees. The proposed methods may have performance limitations when dealing with highly complex or large-scale layout design issues. Additionally, user preferences and real-time feedback may not be fully integrated. To compensate for these shortcomings, future research can explore more effective combinations of optimization algorithms and parallel computing techniques to improve the performance and efficiency of algorithms in handling complex problems. In addition, further research can be conducted on how to effectively integrate user preferences and real-time feedback into algorithms to achieve more personalized and interactive layout design processes.

REFERENCES

[1] Slowik, A. & Kwasnicka, H. Evolutionary algorithms, and their applications to engineering problems. *Neural Computing And Applications.* **32** pp. 12363-12379 (2020)

[2] Abualigah, L. & Alkhrabsheh, M. Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing. *The Journal Of Supercomputing.* **78**, 740-765 (2022)
[3] Ameijde, J., Ma, C., Goepel, G., Kirsten, C. & Wong, J. Data-driven placemaking: Public space canopy design through multi-objective optimization considering shading, structural and social performance. *Frontiers Of Architectural Research.* **11**, 308-323 (2022)
[4] Ejegwa, P. & Agbetayo, J. Similarity-distance decision-making technique and its applications via intuitionistic fuzzy pairs. *Journal Of Computational And Cognitive Engineering.* **2**, 68-74 (2023)
[5] Watson, M., Leary, M. & Brandt, M. Generative design of truss systems by the integration of topology and shape optimisation. *The International Journal Of Advanced Manufacturing Technology.* **118**, 1165-1182 (2022)
[6] Chen, C., Chacón Vega, R. & Kong, T. Using genetic algorithm to automate the generation of an open-plan office layout. *International Journal Of Architectural Computing.* **19**, 449-465 (2021)
[7] Keshavarzi, M. & Rahmani-Asl, M. Genfloor: Interactive generative space layout system via encoded tree graphs. *Frontiers Of Architectural Research.* **10**, 771-786 (2021)
[8] Hassan, W., Attea, Z. & Mohammed, S. Optimum layout design of sewer networks by hybrid genetic algorithm. *Journal Of Applied Water Engineering And Research.* **8**, 108-124 (2020)
[9] Dong, Z. & Bian, X. Ship pipe route design using improved A* algorithm and genetic algorithm. *IEEE Access.* **8** pp. 153273-153296 (2020)
[10] Feng, S., Zhang, W., Meng, L., Xu, Z. & Chen, L. Stiffener layout optimization of shell structures with B-spline parameterization method. *Structural And Multidisciplinary Optimization.* **63** pp. 2637-2651 (2021)
[11] He, S., Fu, Q. & Li, X. Parameterized Interior Floor Plan Design Based on Differentiable Renderer. *Chinese Intelligent Systems Conference.* pp. 132-141 (2022)
[12] Tan, K., Feng, L. & Jiang, M. Evolutionary transfer optimization-a new frontier in evolutionary computation research. *IEEE Computational Intelligence Magazine.* **16**, 22-33 (2021)
[13] Zhang, T., Lei, C., Zhang, Z., Meng, X. & Chen, C. AS-NAS: Adaptive scalable neural architecture search with reinforced evolutionary algorithm for deep learning. *IEEE Transactions On Evolutionary Computation.* **25**, 830-841 (2021)
[14] Deng, Q., Kang, Q., Zhang, L., Zhou, M. & An, J. Objective space-based population generation to accelerate evolutionary algorithms for large-scale many-objective optimization. *IEEE Transactions On Evolutionary Computation.* **27**, 326-340 (2022)
[15] Al-Fugara, A., Ahmadlou, M., Al-Shabeeb, A. & AlAyyash, S. Spatial mapping of groundwater springs potentiality using grid search-based and genetic algorithm-based support vector regression. *Geocarto International.* **37**, 284-303 (2022)
[16] Yi, J., Bai, J., He, H., Zhou, W. & Yao, L. A multifactorial evolutionary algorithm for multitasking under interval uncertainties. *IEEE Transactions On Evolutionary Computation.* **24**, 908-922 (2020)
[17] Mekawy, M. & Gabr, M. Against a workplace contagion: a digital approach to support hygiene-conscious office space planning. *Open House International.* **46**, 391-400 (2021)
[18] Sarkar, A. & Bardhan, R. Improved indoor environment through optimised ventilator and furniture positioning: A case of slum rehabilitation housing, Mumbai, India. *Frontiers Of Architectural Research.* **9**, 350-369 (2020)
[19] Kheiri, F. Optimization of building fenestration and shading for climate-based daylight performance using the coupled genetic algorithm and simulated annealing optimization methods. *Indoor And Built Environment.* **30**, 195-214 (2021)
[20] Fang, Y., Luo, B., Zhao, T. & Others ST-SIGMA: Spatio-temporal semantics and interaction graph aggregation for multi-agent perception and trajectory forecasting. *CAAI Transactions On Intelligence Technology.* **7**, 744-757 (2022)
[21] Pervaiz, S., Ul-Qayyum, Z., Bangyal, W., Gao, L. & Ahmad, J. A systematic literature review on particle swarm optimization techniques for medical diseases detection. *Computational And Mathematical Methods In Medicine.* **2021** pp. 1-17 (2021)
[22] Bangyal, W., Ahmad, J. & Abbas, Q. Recognition of off-line isolated handwritten character using counter propagation network. *International Journal Of Engineering And Technology.* **5**, 227 (2013)

Edited by: Zhengyi Chai
Special issue on: Data-Driven Optimization Algorithms for Sustainable and Smart City
Received: Dec 7, 2023
Accepted: May 6, 2024