Mathematical Biosciences and Engineering

MBE, 20 (7): 12169–12193.
DOI: 10.3934/mbe.2023542
Received: 07 February 2023
Revised: 30 April 2023
Accepted: 30 April 2023
Published: 16 May 2023

http://www.aimspress.com/journal/MBE

Research article

A point-feature label placement algorithm based on spatial data mining

Wen Cao[1,*], Jiaqi Xu[1], Feilin Peng[2], Xiaochong Tong[3], Xinyi Wang[4], Siqi Zhao[1] and Wenhao Liu[1]

[1] School of Geoscience and Technology, Zhengzhou University, Zhengzhou 450001, China
[2] Zhongke Yungu Technology, Changsha 201306, China
[3] School of Geospatial Information, University of Information Engineering, Zhengzhou 450001, China
[4] College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100101, China

* **Correspondence:** Email: zzdx_edifier@zzu.edu.cn.

Abstract: The point-feature label placement (PFLP) refers to the process of positioning labels near point features on a map while adhering to specific rules and guidelines, finally obtaining clear, aesthetically pleasing, and conflict-free maps. While various approaches have been suggested for automated point feature placement on maps, few studies have fully considered the spatial distribution characteristics and label correlations of point datasets, resulting in poor label quality in the process of solving the label placement of dense and complex point datasets. In this paper, we propose a point-feature label placement algorithm based on spatial data mining that analyzes the local spatial distribution characteristics and label correlations of point features. The algorithm quantifies the interference among point features by designing a label frequent pattern framework (LFPF) and constructs an ascending label ordering method based on the pattern to reduce interference. Besides, three classical metaheuristic algorithms (simulated annealing algorithm, genetic algorithm, and ant colony algorithm) are applied to the PFLP in combination with the framework to verify the validity of this framework. Additionally, a bit-based grid spatial index is proposed to reduce cache memory and consumption time in conflict detection. The performance of the experiments is tested with 4000, 10000, and 20000 points of POI data obtained randomly under various label densities. The results of these experiments showed that: (1) the proposed method outperformed both the original algorithm and recent literature, with label quality improvements ranging from 3 to 6.7 and from 0.1 to 2.6, respectively. (2) The label efficiency was improved by 58.2% compared with the traditional grid index.

Keywords: metaheuristics; point-feature label placement; data mining; spatial distribution characteristics; label correlation

1. Introduction

With the arrival of the big data era, people have an urgent demand and expectation for the intrinsic mechanism and high-value information contained in big data. Visualization technology [1], as an important tool of big data analysis, has become a hot spot of people's attention and research [2]. In the field of data visualization, geographic information technology is often used to integrate various industrial data into a unified geospatial space for analysis and representation. Therefore, geospatial data visualization is an important foundation for big data analysis and visualization technologies [3]. It can transfer information to users by transforming geospatial data into understandable symbols such as maps [4]. In the process, the label is the key factor for users to obtain valuable information correctly and quickly. Hence, obtaining more readable and clear labels becomes the focus and difficulty of research.

Spatial objects in data visualization are generally classified into points [5], lines [6], areas [7], and bodies [8]. Among them, the point-feature label placement (PFLP) problem is the most complicated. The point-feature label placement is the placement of labels near the point features of a map according to certain rules such as maximizing conflict-free, ambiguity-free, and aesthetics, and belongs to the NP-hard combination optimization problem [9]. Scholars at home and abroad have conducted a lot of research on this challenge of finding the optimal solution from the large-scale feasible solution set in combinatorial optimization problems. Metaheuristic is a method that uses high-level strategies to explore the search space and finds approximate optimal solutions in a short time [10]. It has been widely used in the study of PFLP problem, including tabu search [11], simulated annealing algorithm [12], genetic algorithm [13], ant colony algorithm [14], and greedy randomized adaptive search [15]. Most scholars mainly focus on improving these metaheuristic algorithms or adopting hybrid metaheuristic algorithms. Rabello et al. [16] presented a Clustering Search (CS) metaheuristic algorithm to solve the PFLP problem; Araujo et al. [17] proposed a CS-based Density Search Clustering (DCS) metaheuristic algorithm as a new alternative to PFLP; Ding [18] et al. combined exact and heuristic algorithms to construct the initial solution and used a backward greedy approach to improve the initial solution to obtain maximally conflict-free labels. Li et al [19]. combine genetic algorithm and tabu search to solve point-feature label placement. Lu [20] and DENG [21] et al. used discrete differential evolution and genetic algorithm (DDEGA) to solve the multi-geographic feature notation configuration problem.

Some other scholars have applied data mining techniques to point-feature label placement, mainly including clustering grouping, extracting elite set patterns to guide subsequent iterations, and mining data set information to plan the label ordering. The clustering method refers to analyzing the features and dividing the whole dataset into several sub-datasets to reduce the computational complexity of the problem and thus improve the quality and efficiency of the label. Alvim & Taillard [22] proposed POPMUSIC heuristics to solve the PFLP problem. It divided the original instance into several subparts for tabu search and was suitable for medium-sized instance problems. Moreover, they presented a new set of instances with 13,206 points to be labeled. Zhou et al. [23] combined the ant colony algorithm and Density-based spatial clustering of applications with noise (DBSCAN) to solve the PFLP problem. This method is suitable for maps with large point scales and large variations in point cluster density. Cao et al. [24] divided a single dataset into several independent sub-datasets based on the label association model, but this method had limited effectiveness in enhancing dense point features. Due to the ability of clustering to reduce the complexity of the problem, the method has been well applied to different problems such as traveling salesman [25,26] and shop floor scheduling [27,28] problems. However, while clustering can improve label quality to some extent, the improvements through clustering are low when the number of solution space searches is large enough. Using parallel computing can increase computational power requirements, while not using parallel computing results in severe computational time consumption. In addition, due to the good combination of data mining and greedy randomized adaptive search procedures [29,30], Guerine [31] et al. borrowed this idea to combine data mining techniques to cluster search and simulated annealing for the first time by extracting the maximum frequent item set for the elite set to guide the subsequent iterations of cluster search to obtain higher quality solutions. Besides, the label results can also be optimized by mining data set information to plan the label order. Luo & Xu [32] used Voronoi diagram to describe the spatial proximity and interaction of point features in the labeling. On this basis, the labeling order of degree of freedom from small to large was planned to control the labeling process in the direction of simplicity, efficiency, and high quality, but the labeling speed of this method is slow. Li et al [33] proposed to annotate the point elements according to their size from small to large, which got the more conflict-free labels. These methods can improve the quality of labels to some extent but do not take into account the interference among features, and the solution quality may still be poor when solving dense and complex datasets. Cao et al. [24] quantified the interference ability of point features using the number of other point features associated with the point feature to guide the label placement. Although the method takes into account the interference among features, it considers more the global spatial characteristics of point elements, i.e., the number of rectangles associated with the point features. The quantified interference ability of point feature labels is more ambiguous, and the process of solving the label placement can still fall into the local optimum.

Moreover, in the label placement process, it is required that there should be no overlap or conflict between labels and features. Therefore, a corresponding conflict detection method is needed. There are various methods such as traversal, R-tree spatial index [34], and grid spatial index [35] to detect conflict. These methods have the problem of being time consuming.

Regarding the problems of planning label order and conflict detection time, we proposed a PFLP algorithm based on spatial data mining which used data mining to discover the inherent laws of point datasets. Our main contributions are:

I. The label frequent pattern framework of point features describes the interference among features in the labeling placement by fully mining the local spatial distribution characteristics of features and the correlation among labels. This pattern can provide a judgment basis for designing placement priorities.

II. A flexible framework consisting of an ascending label ordering method is proposed for PFLP, which reduces the interference among label placement.

III. In the conflict detection of automatic label placement, the bit-based grid spatial index is suggested to reduce cache memory and consumption time.

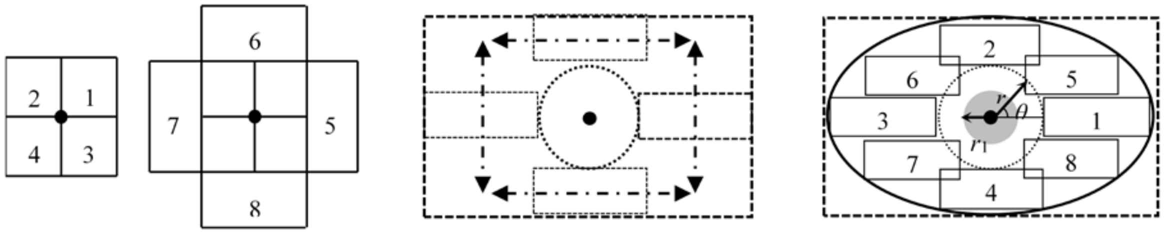
This paper is structured as follows. In Section 2, detailed descriptions of the label frequent pattern framework of point features, a flexible framework consisting of label ordering based on label frequent pattern framework, and conflict detection of the bit-based grid spatial index as well as multiple metaheuristic algorithms are provided. In Section 3, we make experiments to test the proposed algorithm, analysis, and discussion of the results. In Section 4, the conclusions are summarized.

2. The proposed method

2.1. Label frequent pattern framework of point features

2.1.1. Label candidate position model

The selection of a label candidate position model is one of the important foundations of label placement. It directly affects the quality and efficiency of labeling. At present, there are three main types of label candidate position models, as shown in Figure 1. (1) Fixed position model [36], which is an abstraction of label orientation, and it has low flexibility and space utilization; (2) sliding model [37], which can utilize the blank area more efficiently by moving in fixed steps, but the computational amount will increase significantly as the step length decreases; (3) multi-level and multi-orientation model [23], which improves the flexibility of candidate positions and the utilization of blank area by adjusting the radius *r* and inclination angle *θ*. The inclination angle $\theta \in [0^\circ, 360^\circ]$ under different levels of radius *r* can be divided into *n* equal divisions by $\Delta\theta$, and *n* candidate positions can be generated.



(a) fixed-position model (b) slider model (c) multi-level and multi-orientation model

Figure. 1. Label candidate position model.

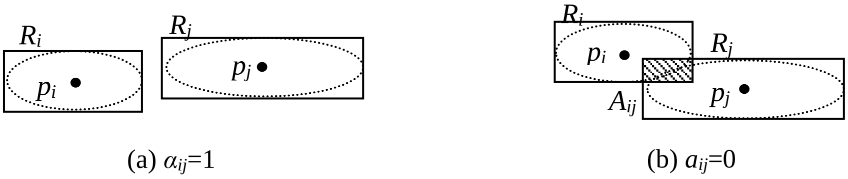
Considering the aesthetics, clarity, and non-ambiguity of labels, the symbolic circumcircle of the point feature can be added to the multi-level and multi-orientation model, and the symbolic circumcircle radius r_1 of the point feature is required to be smaller than the model radius *r*. it is suggested that the value of $\delta = r - r_1$ should not be too large to avoid ambiguity. The multi-level and multi-orientation model unifies the label candidate position models. This model can make full use of the space by adjusting these parameters, while it maybe brings a larger computation. Therefore, to balance the labeling quality and computing time, we selected the 8-position model with the parameters *δ* and $\Delta\theta$ as 5 pixels and 45º, respectively. In Figure 1(c), the smaller numerals indicate a higher priority of label candidate position, and the solid ellipse and the dashed rectangle are the circumscribed ellipse and the minimum bounding rectangle (MBR) composed of different positions, character heights, and string lengths, respectively.

The point-feature label placement algorithm chooses different positions of points with some rules to obtain solutions. In this process, each labeled point directly will affect the labeling positions of subsequent points, so it is worthwhile to focus on how to use their diversity and mutual interference to guide the solution of PFLP.

2.1.2. Mining and analysis based on local spatial distribution characteristics of point features

Data mining [38] refers to mining knowledge from data using classification, regression analysis, cluster analysis, association rules, feature analysis, etc. It aims to transform a large amount of data into useful information and knowledge expressed as patterns or rules, such as frequent item sets and association rules, which help people better understand the actual data. When solving the PFLP problem, various factors such as position, character height, string length, and sequence of labels can cause strong mutual interference. The interference causes the labeling algorithm to fall into the local optimum trap and slow down the solution speed. From the label candidate position model, it can be seen that the label candidate position minimum bounding rectangle (LBP-MBR) is driven by various factors such as different positions of labels, character height, string length, and sequence of labels. Therefore, we can use the association rule technique in data mining to discover the influence law among point labels and construct the corresponding label frequent pattern framework to quantify the influence of different point features on labeling.

Let $\mathbf{P} = \{p_1, p_2, \mathrm{K}\ , p_N\}$ represent the set of points to be labeled and LBP-MBR of point p_i and point p_j are respectively R_i and R_j. α_{ij} a binary variable represents whether there is mutual interference between them. α_{ij}=0 indicates that point p_i and p_j have no correlation in the labeling process, as shown in Figure 2(a), that is $R_i \cap R_j = \phi$. In this case, we do not need to consider the order of labels. Otherwise, as shown in Figure 2(b), α_{ij}=1 indicates that p_i and p_j have a correlation, i.e., $R_i \cap R_j \neq \phi$. Their influence on the automatic label placement needs to be considered. if α_{ij}=1, there exists an intersecting part of LBP-MBR between p_i and p_j, which is defined as the label influence surface A_{ij}, $A_{ij} = R_i \cap R_j$. Obviously, both α and A have symmetry, i.e., $\alpha_{ij} = \alpha_{ji}$ and $\mathrm{A}_{ij} = \mathrm{A}_{ji}$.



(a) α_{ij}=1

(b) a_{ij}=0

Figure 2. Label influence surface pattern.

Label influence surface between point features will produce corresponding interference in their label placement. We used association rules in data mining to explore local spatial distribution characteristics of point features and label correlation. Based on the calculation of LBP-MBR of point feature, two functions, label support *sup(p_i)* and label confidence *conf(p_i)*, are defined to construct the label frequent pattern framework (LFPF) to quantitatively describe the interference intensity of features in the label placement process.

(1) label support $sup(p_i)$

Label support $sup(p_i)$ is calculated based on the LBP-MBR of point p_i. As shown in Figure 3 and Eq. (1), the label conflict frequency value g_{xy} ($\mathrm{g_{xy}}$ is initialized to 0) is calculated for each grid by judging whether there is a label influence surface x_{ij} at the label position. g_{xy} reflects the frequency that the position of the gird is susceptible to be selected by label. $sup(p_i)$ describes the interference intensity of the point p_i to other features by accumulating all label conflict frequent values in the range of LBP-MBR of p_i, as shown in Eq. (2).

$$g_{xy} = \begin{cases} g_{xy} + 1 & x_{ij} = 1, g_{xy} \in S_{ij} \\ g_{xy} & x_{ij} = 0 \end{cases} \tag{1}$$

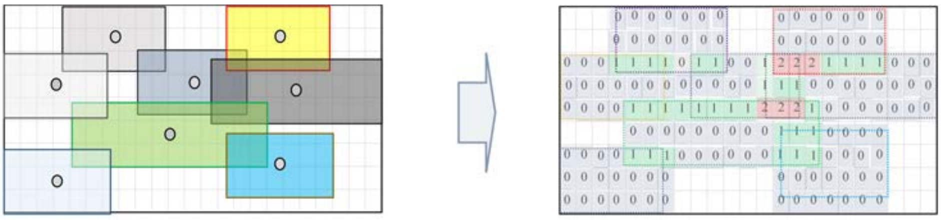
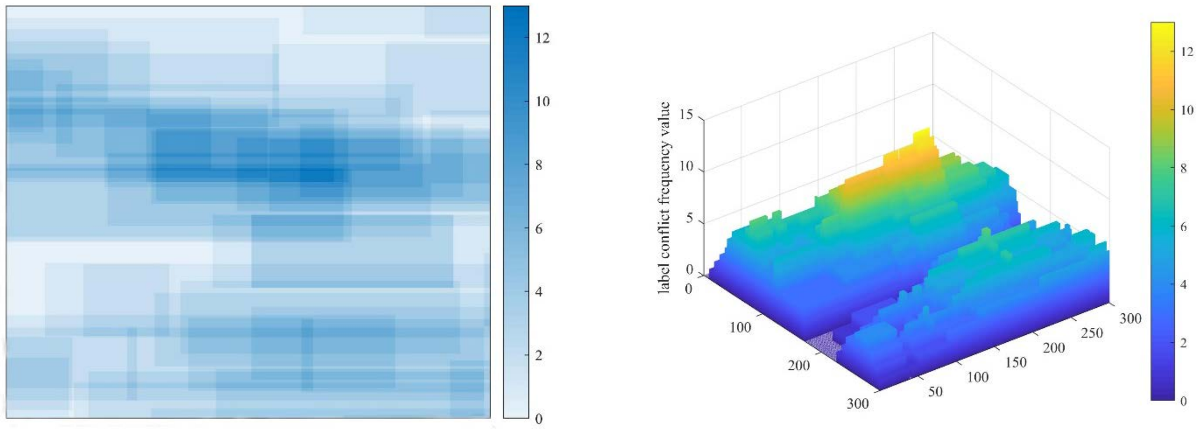


Figure 3. Schematic diagram of calculation principle of label conflict frequent value.

$$sup(p_i)=\sum_{x=1}^{w_i}\sum_{y=1}^{h_i}g_{xy} \tag{2}$$

Where w_i and h_i are the width and height of the label candidate rectangle of point p_i respectively.

sup(p_i) quantifies the interference intensity by counting all the frequent values of LBP-MBR of p_i. The larger value indicates that LBP-MBR of p_i appears frequently in LBP-MBR of other point features, reflecting that the local spatial distribution around the point is denser and the spatial competition is intense, as shown in Figure 4. If we take it as a priority labeling point will reduce the selectivity of the label position of the subsequent points to be labeled, resulting in the solution of the labeling algorithm easily falling into the local optimum trap.



(a) Label density diagram

(b) Statistical histogram

Figure 4. Label density diagram and statistical histogram of conflict frequent values.

In our early process [24], we expressed *sup*(p_i), which quantifies the interference of point p_i, by using the number of point features associated with LBP-MBR of the point element p_i. However, we did not take into account the size of the LBP-MBR. The effect of affecting one label position rectangle of point feature is the same as that of multiple label position rectangles of point feature, take Figure 5 as an example. For instance, as shown in the Figure 6, according to the method described in reference 24, the interference ability of point p_i is greater than that of point p_k. point p_k is labeled before point p_j. In fact, the number of label candidates affected by p_k is 14, while the number of annotation candidates affected by point p_j is only 7. If p_k is labeled first, it could affect the label possibility of subsequent point features, potentially leading to a local optimum trap. However, this paper uses the label conflict frequency value within the label candidate area of the point feature to represent *sup*(p_i). Considering the size of the label influence surface, the interference capability of the point feature is more reasonably represented. the label order of p_j is prior to p_k, avoiding the local optimal trap caused by p_k being labeled first.

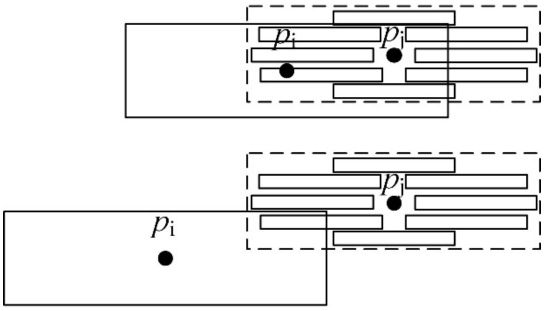
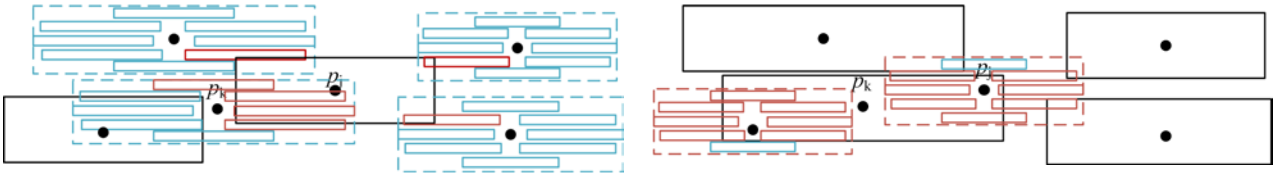


Figure 5. Limitations of label support in reference 24.



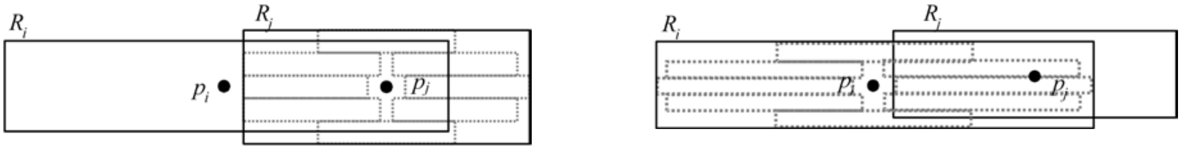
(a) interference ability of p_j

(b) interference ability of p_k

Figure 6. Comparison with reference 23 quantifying interference ability.

(2) label confidence *conf*(p_i)

Although *sup*(p_i) describes the local spatial distribution characteristics of p_i, it ignores the interference between correlation points. For instance, the label influence surfaces A_{ij} and A_{ji} between p_i and p_j are equal, *sup*(p_i)=*sup*(p_j), but the interference between them is different. As shown in Figure 7**,** figure (a) indicates that the LBP-MBR of p_i interferes with eight label positions of p_j, while figure (b) indicates that the LBP-MBR of p_j affects five label positions of p_i. Therefore, *conf*(p_i) needs to be introduced to describe the correlation among the labels of point features.



(a) Interference of point p_i to point p_j

(b) Interference of point p_j to point p_i

Figure 7. Schematic diagram of label correlation principle.

From the above representation, it is known that the impact of the same label influence surface on the label placement of different point features is different, and the mathematical essence is the size of the LBP-MBR of the two points is different. As a result, an association probability P_{ij} needs to be defined to represent the interference intensity of labeled point p_i to point p_j to be labeled, see Eq. (3).

$$P_{ij} = \frac{\alpha_i}{w_i \times h_i} \tag{3}$$

Similarly, P_{ji} can be obtained as the effect of the preferred labeled point p_j on subsequent point p_i to be labeled. In dense point feature label placement, spatial competition is intense, and a point feature often interferes with multiple other point features. *conf*(p_i) is the sum of the associated probabilities of other point features that have a label influence surface with the point p_i, but directly accumulating their association probabilities ignores the characteristics associated with each pair of labels. Therefore, the association probabilities are first weighted and corrected according to the number of label positions of the label influence surface overlapping point p_j, and then the *conf*(p_i) can be obtained by accumulating weighted association probability of all other point features that have interference with the point feature p_i, as shown in Eq. (4).

$$conf\,(p_i) = \sum_{i=1, i\neq j}^{k} \omega_t P_{ij} \tag{4}$$

Where ω_t is the weight corresponding to equal intervals of association probability, and $\omega_1 \leq \cdots \leq \omega_t \leq \cdots \leq \omega_n$ (n is the number of candidate positions for the label).

The use of association probability to calculate the *conf*(p_i) not only described the interference intensity to other point features in its local area after preferential labeling on point p_i, but also counteracted the over-smoothing problem caused by the summation calculation of association probabilities.

In summary, we used the two quantitative criteria of *sup* (p_i) and *conf*(p_i) to mine the frequent patterns in the dataset to characterize the local spatial distribution and label correlation of point feature p_i, to provide priori auxiliary decision information for the solution.

2.2. Label ordering based on label frequent pattern framework

The core of solving the point set is to obtain the maximum conflict-free labels in a limited space, and correlation among labels of point features actually reflects the spatial competition among these points, so PFLP problem can also be regarded as a trade-off problem of labels. In the process of label placement, the choice of point feature label is to judge whether there is an appropriate label position in the empty space according to the previous labeled results. Meanwhile, its labeling position will also affect the label position selection of subsequent point features. The PFLP algorithm usually randomly selects point features for labeling, ignoring the internal spatial structure and order of datasets, increasing the uncertainty and ambiguity in the solution process.

The label frequent pattern framework describes the local spatial distribution characteristics of point p_i and the correlation of its directly correlated point labels by defining the label support *sup*(p_i) and the label confidence *conf*(p_i) respectively. In this way, we designed the corresponding sequential rules to guide solutions to maximize the number of conflict-free labels. According to the function definition, *sup*(p_i) has a strong correlation with the interference of points, while *conf*(p_i) has a weak correlation. Therefore, the solution should be guided by the order of *sup*(p_i) first and then *conf*(p_i), and the label ordering of an ascending order label frequent pattern framework (A-LFPF) is constructed, as shown in Figure 8. All points are first sorted from small to large with the value of *sup*(p_i); then sort the points with the same value of *sup*(p_i) according to the value of *conf*(p_i) from small to large; and finally, sort the points with the same value of two functions randomly. compared with the label ordering of descending order label frequent pattern framework (D-LFPF), the label ordering of A-LFPF can not only effectively reduce the interference of the labeled points to the subsequent points to be labeled, but also improve the possibility of searching the empty space for the points to be labeled. Therefore, this label ordering can guide the label placement process, thereby improving the optimization rate of the algorithm and making the approximate optimal solution closer to the optimal solution. During the process of labeling point features, it's common to encounter different levels of detail (LOD). To prioritize the labeling process, we can rank the levels in order of importance and use the A-LFPF for ordering labels within the same level. However, in the experiment conducted to verify the superiority of the algorithm proposed in this paper, all point features were treated as having the same level of detail.

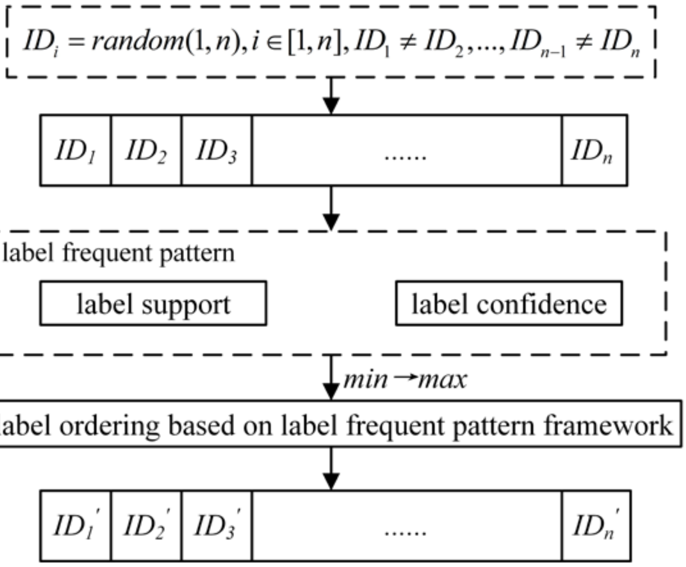

$ID_i = random(1,n), i \in [1,n], ID_1 \neq ID_2, ..., ID_{n-1} \neq ID_n$
ID_1
ID_2
ID_3
......
ID_n
label frequent pattern
label support
label confidence
min→max
label ordering based on label frequent pattern framework
ID_1'
ID_2'
ID_3'
......
ID_n'

Figure 8. Ascending order label frequent pattern framework.

2.3. Solving label placement problem based on label frequent pattern framework

The label frequent pattern framework (LFPF) is composed of label orderingbased on the label frequent pattern framework to mine the spatial distribution information and label correlation of point features, and then guide the solution of the label placement algorithm. The label placement based on LFPF obtains more reasonable results by establishing labeling quality evaluation, conflict detection method, and multi-hierarchy metaheuristic algorithm.

2.3.1. Labeling quality evaluation

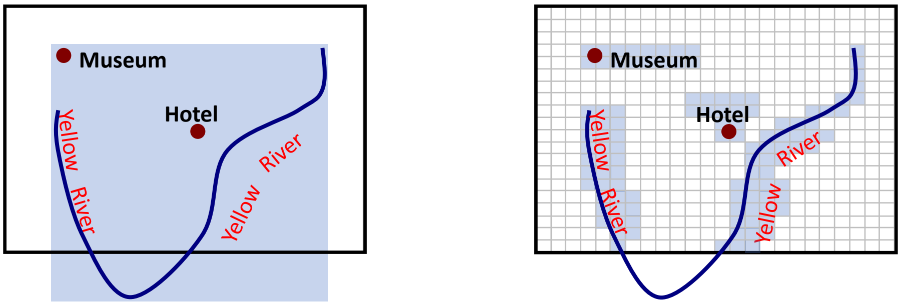
The label placement aims to obtain the maximum number of clear, aesthetically pleasing, and readable conflict-free labels [39,40]; thus, three factors need to be considered: (1) conflict or overlap between labels and features; (2) label candidate position preference; (3) association of labels with corresponding points. Based on these criteria, users can optimize the labeling results using the evaluation function. In this paper, the labeling quality evaluation is defined as:

$$E(l) = \lambda_1 E_{conflict}(l) + \lambda_2 E_{peference}(l) + \lambda_3 E_{association} \quad (5)$$

Where l represents the labeling result of a map. The functions $E_{conflict}()$, $E_{peference}()$, and $E_{association}()$ quantify the conflict, the preference, and the association factors of labeling result respectively, with λ_1,λ_2 and λ_3 as the weights. Due to the label association factor being related to the human eye resolution, it was not considered, i.e., λ_3=0. Assuming that a smaller quantitative score indicates a higher quality labeling result, then the goal of PFLP is to obtain the smallest score.

2.3.2. Conflict detection

Because one of the criteria of PFLP is no overlap or conflict between labels and features, it is necessary to develop a reasonable conflict detection method. There are various methods such as traversal, R-tree spatial index [34], and grid spatial index [35] to detect conflict. The traversal approach is to perform conflict detection between the label to be placed and the placed labels one by one. Its speed is extremely low and does not meet the needs of label placement. R-tree spatial index (RI) takes the minimum bounding rectangle of geographical objects as the basis of data storage. It divides neighboring geographic objects together by spatial aggregation and allows direct indexing of spatial objects that occupy a certain range. As the data volume increases, the deepening and reorganization of the spatial tree will take up more memory space and slow down the query speed. Moreover, it is not accurate for irregular features and rotation, wasting a lot of graphic space, as shown in Figure 9. The grid spatial index adopts row arrangement coding, and it only needs to determine whether the grid is where a label is located. This method is equivalent to a local small-scale traversal addressing and has strong adaptability and robustness to irregular ground features. Therefore, allowing for the possible irregularity of point symbols, the grid spatial index is suggested for conflict detection.



(a) R-tree spatial index

(b) grid spatial index

Figure 9. Comparison chart of index methods.

The traditional grid space index (GI) is built in a way that one byte corresponds to one grid. It is prone to the problems of large memory space and time consuming in the case of large visualization range and high accuracy of grid division. Therefore, this paper takes full advantage of the 8-bit per byte feature of the computer to propose a bit-based grid spatial index (BI), i.e., one grid corresponding to one bit. The grid spatial index based on x-bit mainly includes 8-bit, 16-bit, 32-bit and 64-bit, as shown in Figure 10. Theoretically speaking, it not only reduces the memory space to about 1/8 times, but also reduces the number of searches for one label to about 1/x times.

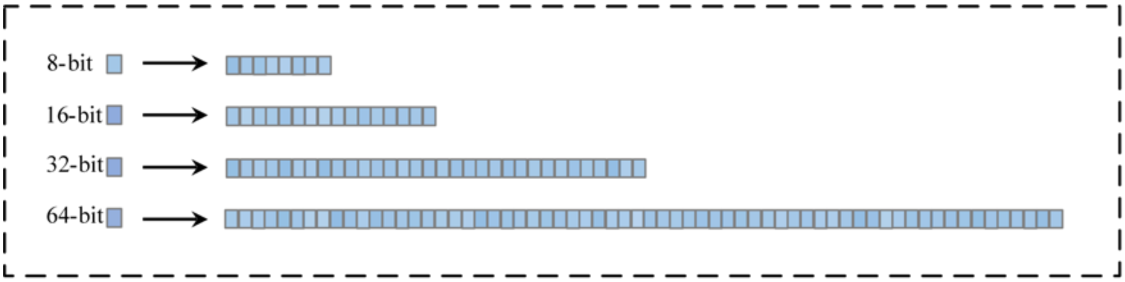


Figure 10. Schematic diagram of bit-based grid spatial index calculation unit.

For the 8-bit example of conflict avoidance for point labels, assume that the height of the label is H pixels and the width is W pixels. When the horizontal direction is bitwise, the label is divided into the left starting part, the middle byte part, and the right end part, as shown in Figure 11. The bit operation is performed to determine whether the grid is occupied by taking the low bit for the left start part and the high bit for the end part. the search times of a conflict-free label can be reduced by nearly 1/8 times compared with the byte index, which effectively improves the space utilization and reduces the search time. In addition, if the label text is arranged horizontally, usually W > H, which means that the horizontal bit grid can make full use of the bit characteristics to reduce the number of searches, the bit grid can be used horizontally or vertically according to the main way of text arrangement.

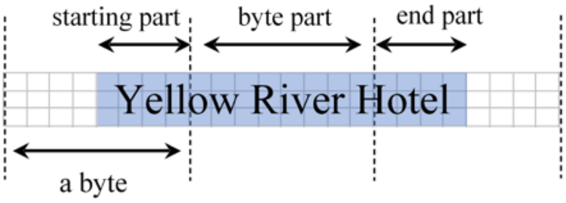


Figure 11. Schematic diagram of bit-based grid spatial index calculation unit.

2.3.3. Metaheuristic algorithm based on label frequent pattern framework of point features

The PFLP problem is a typical combinatorial optimization problem, and meta-heuristic algorithms are usually used to solve the point feature label placement problem. Simulated annealing algorithm (SA), genetic algorithm (GA), and ant colony algorithm (ACA) are the classical algorithms to solve the PFLP problem, which have good performance in solving the notation configuration. The current label placement algorithm usually randomly selects point features as label ordering, but solving in random order ignores the intrinsic spatial structure and order information of the point datasets, which increases the uncertainty and ambiguity in the process of PFLP. We construct the label frequent pattern framework by mining the local spatial characteristics and label correlation of point features and obtain a label ordering of an ascending order label frequent pattern framework pattern (A-LFPF) according to this pattern and use the A-LFPF to guide the meta-heuristic algorithm to solve the point feature label placement to jump out of the local optimum. For these three metaheuristic algorithms, we provide some introduction to them as follows:

1. Simulated Annealing Algorithm

SA is based on solid annealing. First, at a high initial temperature, the solution is diverse and volatile due to the acceptance of new states with large energy differences from the current one. Then, as the temperature slowly cools down, the probability of accepting a worse solution slowly decreases and eventually converges to a stable value, i.e., the approximate optimal solution. SA has a good global search capability, and the specific steps are as follows:

(1) Initialize the temperature T to a high initial temperature T_0 and randomly initializes the solution.

(2) Repeat the following steps until the temperature T falls below a given threshold value Tc:

① If the set number of annealing iterations is reached decrease the temperature T according to an annealing schedule otherwise perform steps 2 and 3.

② Randomly select a certain number (mv×n) of points and make them reselect a new candidate position to get a new solution. mv is the labeling transformation probability.

③ Calculate △E, the change of the labeling quality evaluation caused by changing the label. If the new solution is worse, it will accept the difference solution with the probability of Eq. (6).

$$p = e^{-(\Delta E/T)} \tag{6}$$

However, using too many annealing iterations can make the SA algorithm extremely time-consuming. Therefore, in this paper, we start with a smaller number of annealing iterations at the beginning of annealing and gradually increase the number of iterations as the temperature cools.

2. Genetic algorithm

GA is a search algorithm that simulates natural selection and population genetic mechanisms. In the application of PFLP, genes represent the candidate labeling position, and these are randomly generated in {1,2,3,4,5,6,7,8} by the integer encoding to form individuals. A certain number of individuals formed the population. According to the genetic mechanism, individuals of the parent generation are crossed or mutated to produce offspring, and the population evolves according to superiority and inferiority to obtain the optimal population. We obtain the optimal solution by using an elite selection strategy, uniform crossover, and variation.

The specific steps are as follows ：

(1) Generate a certain number of individuals and initialize the population.

(2) An elite retention strategy is used to replicate the elite population of the parent directly to the next generation. Individuals from the parent population are crossed and genetically mutated to obtain new individuals. Crossover parents are randomly selected from individuals, one of which must be from an elite individual, and the other from the entire population, performing uniform crossover to produce new offspring. Mutation takes place at a single point of variation.

(3) Select the best individuals among the parent population and all new individuals as the offspring population.

Repeat the above steps until the termination condition is satisfied to obtain the best population, and the best individual in this population is the approximate optimal solution for the secondary dataset.

3. Ant colony algorithm

The ant colony algorithm simulates the process of foraging by ants. The ant colony is constructed following the principle that the size of data is the same as the size of the ant colony [23], The right path is selected using the roulette wheel method with the pheromone concentration remaining in the ant colony during path selection. Since the ant colony algorithm has better guidance on the label direction, we develop the penalty mechanism based on label position preference to update pheromone concentration. The higher the priority of a conflict-free label position label is, the more the pheromone concentration increases. More ants will choose this path with a positive feedback mechanism. The specific steps are as follows:

(1) Initialization: the size of the ant colony is the same as the point dataset, the pheromone τ_{ij} of each point feature is initialized to $\xi\left(\xi > 0\right)$, the rightmost label priority η_{ij} is set to 8, and the other directions of the label are set to -1 in turn.

(2) Ant search: Since we set a fixed label ordering based on the LFPF in the point feature label placement, we do not need to set a taboo table, and only need ants to traverse through all point element sets.

(3) Calculate the probability of each label candidate position: Calculate the selection probability of each label candidate position based on the pheromone concentration and label priority of each point feature, where α is the information heuristic factor and β is the expectation heuristic factor, and the calculation formula is shown in Eq. (7).

(4) Ant selection location: Based on the selection probability of each position calculated in step (3), a roulette selection method is used to select the candidate location of the label.

(5) Pheromone update: After the ants select the position, the pheromone concentration is updated by setting a penalty mechanism based on the conflict situation and the direction priority of the label. The higher the label priority of a conflict-free label, the higher the increase in pheromone concentration of that label, where the label position directly to the right is +9, and the pheromone of labels in other directions is -1 in turn. If the label has a label conflict then -5. Each label position pheromone has a minimum of 10 to ensure that each label orientation can be selected.

(6) The iteration of the algorithm ends when the number of iterations reaches r_{ACA}.

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{i\in allow}[\tau_{is}]^{\alpha}[\eta_{is}]^{\beta}} \tag{7}$$

All three algorithms output results when certain termination conditions are met. In this paper, when the algorithm reaches a specified number of iterations or the evaluation function value reaches multiple times within a certain range, one of them is satisfied to stop the computation process. The experimental parameters and definitions of the three algorithms are shown in Table 1.

Table 1. Parameters and values used in methods.

Algorithm	Parameters	Parameters Definition	Experimental value
SA	T_0	Initial temperature	40000
	λ	Annealing speed	0.95
	SA_{max}	Number of annealing iterations	4000
	T_c	Termination temperature	1.00
	r_{SA}	Maximum number of repetitions of SA	30000
GA	m_{v}	Labeling transformation probability	0.001
	P_c	Population size	100
	p_m	Elite population probability	0.10
	p_e	Chromosomal crossover probability	0.70
	p_v	Chromosomal variation probability	0.20
	r_{GA}	Maximum number of repetitions of GA	50
ACA	α	Information heuristic factor	0.50
	β	Expectation heuristic factor	0.50
	r_{ACA}	Maximum number of repetitions of ACA	5000

3. Experiment and discussion

3.1. Instance and experiment environment

In our experiments, we use crawler technology to obtain 4000, 10000, and 20000 points of random POI data from Kaifeng city, Zhengzhou city, and Beijing city respectively to test the LFPF algorithm, which is representative of the third-tier, second-tier, and first-tier cities in China. The symbol size R_1 and the distance R_2 from the coordinate point to the label are 5 and 10 pixels, respectively, and the label font is 12 pixels. All methods are implemented by Microsoft Visual C++ 6.0 in C++, and experiments were carried out on Intel (R) Core (TM) i5-8500 3.0 GHz processor with 8GB of RAM.

3.2. Comparison experiment

To verify the superiority of the above algorithms, some comparative experiments with the label frequent pattern framework (LFPF) are conducted through metaheuristics including ACA, GA, and SA. It discusses and analyzes the effect of the proposed method on improving the quality and efficiency of point-feature label placement.

3.2.1. Index comparison experiment

To verify the effectiveness of the proposed bit-grid based index BI, the time consumption statistics of GI, BI and RI obtained by simulated annealing algorithm from 40,000°C annealed to 0.01℃ under the eight commonly used label densities ρ of 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40% are shown in Table 2–4. The Rule in the table represents the order of labels. *P* is the ratio of the sum of the symbol and label area to the map area, indicating the deneseness of the distribution of features and labels. The experimental data are 4000, 10000, and 20000 points of the actual point datasets. Figure 12 is the time cumulative histogram of the three index methods in Tables 2–4.

Table 2. Time consuming statistics of index methods for 4000 points under different label densities.

Index	label ordering	ρ=5%	ρ=10%	ρ=15%	ρ=20%	ρ=25%	ρ=30%	ρ=35%	ρ=40%
GI	R	12059	8229	6881	7285	7018	6586	5902	5802
	D-LFPF	12312	8225	6546	8022	6566	5778	5320	5153
	A-LFPF	11600	7785	6492	7864	6145	6088	5334	4777
BI	R	3835	3216	2724	2695	2628	2437	2296	2334
	D-LFPF	3779	3239	2964	2574	2417	2356	2398	2215
	A-LFPF	3721	2922	2784	2500	2425	2340	2315	2198
RI	R	6213	5681	5345	6868	6440	5377	5064	5161
	D-LFPF	5820	5082	4809	5528	6326	5409	5005	4518
	A-LFPF	6306	5582	5373	6783	5806	5966	5943	5541

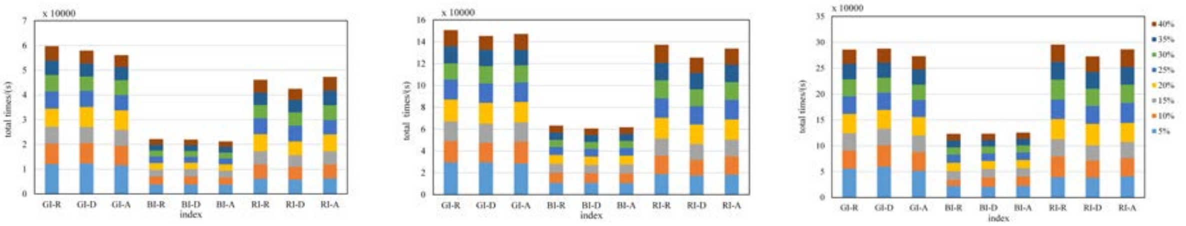
The following two conclusions can be drawn from Figure 12 and Table 2-4: (1) BI takes the shortest time, followed by RI, and GI takes the longest time. the efficiency of BI is 58.2% higher than GI. Because the grid index is equivalent to a local traversal index, search and query times of BI can be reduced by nearly 7 times, greatly improving the conflict detection efficiency. Due to the corresponding spatial regions of the brother nodes of the R-tree can overlap, its index has redundant paths for searching thus wasting some search time. (2) When the scale of points reaches 10000 and 20000, the time consumption of method RI has an obvious relative growth trend compared with BI. From Table 3–4 of 10000 and 20000 experimental statistics, it can be found that RI partially takes longer than GI. RI starts from the root node to find out whether there is a node rectangle intersecting with it according to the position of the label rectangle, and it searches down to the leaf node layer by layer. With the points increasing and the distribution of data aggregation, the R-tree structure will be deeper, which will increase the search time for querying a label rectangle and lead to a decrease in indexing efficiency. For GI, no matter how many points participate in the label placement, the grid index theoretically queries the pixel grid at the location of the label rectangle. Therefore, as points increase, the time increase of RI is relatively the largest.

Table 3. Time consuming statistics of index methods for 10000 points under different label densities.

Index	label ordering	ρ=5%	ρ=10%	ρ=15%	ρ=20%	ρ=25%	ρ=30%	ρ=35%	ρ=40%
GI	R	29527	19704	17767	20066	18320	14794	15682	14750
	D-LFPF	29568	17892	17486	19066	17835	15883	14772	12782
	A-LFPF	28390	20125	17574	18820	17835	15594	14183	14594
BI	R	10804	9167	8119	8260	7196	6639	6561	6493
	D-LFPF	10747	8923	7664	7563	7034	6364	6221	6157
	A-LFPF	10588	8888	8117	8169	6902	6566	6327	6189
RI	R	18913	16734	15729	18908	17980	16458	15839	16652
	D-LFPF	17490	14282	14454	18013	16892	15340	15094	13974
	A-LFPF	18308	16769	15273	18491	18001	16280	15849	14783

Table 4. Time consuming statistics of index methods for 20000 points under different label densities.

Index	label ordering	ρ=5%	ρ=10%	ρ=15%	ρ=20%	ρ=25%	ρ=30%	ρ=35%	ρ=40%
GI	R	55635	35566	33004	37460	34243	32485	30011	27707
	D-LFPF	59638	41401	31953	36540	32940	29101	29101	27258
	A-LFPF	51981	35946	32127	35796	32544	29896	29232	25762
BI	R	22572	11445	16953	16357	15715	14028	13296	12853
	D-LFPF	20740	18125	16335	15590	14584	13328	12681	12083
	A-LFPF	22762	17978	16352	15515	14681	13630	12473	12276
RI	R	39438	40170	33004	39081	37439	38634	34167	33790
	D-LFPF	38190	33536	29299	41392	34717	33146	31945	30634
	A-LFPF	41137	35323	31095	36568	38990	35138	33805	34628



4000 10000 20000

Figure 12. Cumulative time chart of different indexes.

3.2.2. Comparison experiments of point feature label placement based on label frequent pattern framework

In this paper, we fully exploited the local spatial distribution characteristics of the point dataset and the label correlation to construct the LFPF and applied it to the metaheuristics to conduct the labeling experiments. Additionally, the bit-based grid index is also used to reduce conflict detection time. In order to verify the effectiveness and applicability of the new algorithm, we designed to analyze the generalizability of LFPF algorithm and compare the label effect of LFPF algorithm with the primitive metaheuristic algorithm and annotation association model framework (AAMF) algorithm [24] by using simulated annealing algorithm, genetic algorithm, and ant colony algorithm. We used eight commonly used label densities (ρ) of 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40% for the experiment. The experiment is based on the premise that all point features are considered to the same level of detail. The comparative experiments between LFPF and the original algorithm mainly include: (1) LFPF-SA, SA, and AAMF-SA; (2) LFPF-GA, GA, and AAMF-GA; (3) LFPF-ACA, ACA and AAMF-ACA. The experimental data include 4000, 10000, and 20000 points of the actual point datasets, respectively, coming from Kaifeng（a third-tier city), Zhengzhou (a second-tier city), and Beijing (a first-tier city) in China. These three datasets are denoted by I_1, I_2, and I_3 respectively. We repeated the experimental results 10 times to take the average value. We recorded the time consumed (T) and the quantitative score (E) for each group of experiments. The experimental results are shown in Table 5–10. R-SA represents the original SA, A- LFPF -SA represents the SA based on A-LFPF, D- LFPF -SA represents the SA based on D-LFPF, A-AAMF-SA represents the SA based on A-AAMF, and D-AAMF-SA represents the SA based on D-AAMF. The same applies to GA and ACA.

Table 5. The statistical results of SA with label density range from 5% to 20%.

Instance	Algorithm	ρ=5%		ρ=10%		ρ=15%		ρ=20%	
		T	E	T	E	T	E	T	E
I_1	R-SA	3835	195.3	3216	239	2724	265.6	2695	281.1
	A- LFPF -SA	3715	191.6	3170	234.8	2758	259.7	2562	276.8
	D- LFPF -SA	3779	200.1	3239	245.1	2964	269.7	2574	286.5
	A-AAMF-SA	3721	192.5	2922	235.7	2784	259.8	2500	277.4
	D-AAMF-SA	3748	198.3	3282	243.1	2700	267.7	2565	284.4
I_2	R-SA	10804	188.3	9167	234.2	8119	263.6	8260	284.1
	A- LFPF -SA	10588	185.1	8888	230.2	8117	258.9	8169	278.8
	D- LFPF -SA	10747	190.3	8923	236.6	7664	266.4	7563	286.7
	A-AAMF-SA	10864	185.3	9003	230.4	9935	259.5	7707	279.4
	D-AAMF-SA	11103	189.2	9094	235.4	9055	265.1	8052	285.1
I_3	R-SA	22572	220.7	11445	261	16953	285.3	16357	302.7
	A- LFPF -SA	22762	216.5	17978	255.6	16352	279.8	15515	297.1
	D- LFPF -SA	20740	222.5	18125	263.5	16335	288.4	15590	306.1
	A-AAMF-SA	23121	217.6	18061	257.1	16545	281.6	15720	298.7
	D-AAMF-SA	20744	222.4	18470	262.3	16519	286.8	16233	304.8

Table 6. The statistical results of SA with label density range from 25% to 40%.

Instance	Algorithm	ρ=25%		ρ=30%		ρ=35%		ρ=40%	
		T	E	T	E	T	E	T	E
	R-SA	2628	295.7	2473	307.2	2296	316.1	2334	326.2
	A- LFPF -SA	2417	291.5	2421	303.1	2374	311.6	2193	322.1
I_1	D- LFPF -SA	2417	301.7	2356	313.3	2398	322.8	2215	333.3
	A-AAMF-SA	2425	291.9	2340	303.4	2315	312.5	2198	322.7
	D-AAMF-SA	2388	299.7	2395	311.2	2435	320.1	2193	322.1
	R-SA	7196	300.2	6639	314.9	6561	325.7	6493	336.1
	A- LFPF -SA	6902	294.8	6566	308.9	6327	319.6	6189	330.4
I_2	D- LFPF -SA	7034	303.5	6364	317.6	6221	329.1	6157	340.2
	A-AAMF-SA	6996	295.5	6580	310.3	6496	320.5	6333	331.5
	D-AAMF-SA	7025	301.9	6588	316.1	6628	327.6	6331	338.7
	R-SA	15715	315.7	14028	328.6	13296	337.5	12853	346.2
	A- LFPF -SA	14681	310	13630	323	12473	331.6	12276	340.2
I_3	D- LFPF -SA	14584	319.8	13328	332.8	12681	341.8	12083	350.9
	A-AAMF-SA	14896	311.5	13925	324.3	12894	332.7	12766	341.4
	D-AAMF-SA	15064	318.3	13866	331.2	13075	339.9	12470	348.9

Table 7. The statistical results of GA with label density range from 5% to 20%.

Instance	Algorithm	ρ=5%		ρ=10%		ρ=15%		ρ=20%	
		T	E	T	E	T	E	T	E
	R-GA	730	267.5	548	303.4	424	324.6	448	339.5
	A- LFPF -GA	686	261.3	498	299.1	403	318.7	385	333.8
I_1	D- LFPF -GA	670	272.4	541	308.5	444	329.7	300	345.8
	A-AAMF-GA	641	262.8	501	300	433	320	380	334.7
	D-AAMF-GA	585	272.9	525	307.5	437	328.5	316	344.3
	R-GA	1805	258.4	1247	302.7	1073	329.2	1025	346.7
	A- LFPF -GA	1746	253.9	1236	297.5	1016	324	854	342.3
I_2	D- LFPF -GA	1868	260.2	1154	306.7	952	333.3	918	350.1
	A-AAMF-GA	1639	256.3	1359	297.8	1005	325	864	342.8
	D-AAMF-GA	1613	260.7	1344	303.4	929	332	819	349.3
	R-GA	2618	271.6	3399	309.5	1827	331.6	1333	347.7
	A- LFPF -GA	2337	268.2	2101	304.9	1707	326.7	1436	341.5
I_3	D- LFPF -GA	2418	274.8	2103	314.3	1729	336.7	1556	351.3
	A-AAMF-GA	2423	268.4	2128	306.7	1540	328.1	1381	343.5
	D-AAMF-GA	2615	273	2053	313.1	1603	335.1	1428	350.8

Table 8. The statistical results of GA with label density range from 25% to 40%.

Instance	Algorithm	ρ=25%		ρ=30%		ρ=35%		ρ=40%	
		T	E	T	E	T	E	T	E
	R-GA	355	351.8	344	361.7	275	369.3	272	375.9
	A- LFPF -GA	338	346.6	244	358.7	239	364.6	247	371.3
I_1	D- LFPF -GA	300	358	244	368.8	266	375.3	228	382.7
	A-AAMF-GA	347	347	245	360.5	260	364.9	212	373.9
	D-AAMF-GA	274	358.2	282	366.1	243	374.5	227	381
	R-GA	768	360.7	739	371.9	541	382.5	660	389.8
	A- LFPF -GA	705	355.4	649	367.1	599	376.3	514	384.6
I_2	D- LFPF -GA	781	364.3	517	378.5	551	385.3	500	385.5
	A-AAMF-GA	660	357.3	710	367.8	550	376.7	508	385.4
	D-AAMF-GA	774	362.6	627	374.9	566	384.6	478	393.1
	R-GA	1259	359.3	1072	370.8	1052	378	953	386.1
	A- LFPF -GA	1226	353.7	1019	364.9	1028	372.6	909	380.3
I_3	D- LFPF -GA	1108	364.8	1019	376.4	1043	383.2	853	391.8
	A-AAMF-GA	1209	354.7	1130	366.1	1024	373.2	957	380.8
	D-AAMF-GA	1128	362.8	1079	373.6	876	382	908	389.3

Table 9. The statistical results of ACA with label density range from 5% to 20%.

Instance	Algorithm	ρ=5%		ρ=10%		ρ=15%		ρ=20%	
		T	E	T	E	T	E	T	E
	R-ACA	292.8	266.5	248	303.9	254	324.6	237	339.8
	A- LFPF -ACA	331	260.2	312	297.4	249	318.4	292	333.9
I_1	D- LFPF -ACA	282.5	273.8	241.4	312	229	332.1	179	346.9
	A-AAMF-ACA	345	361.7	257	298.7	247	319.7	210	335.7
	D-AAMF-ACA	351.2	272.2	315	309.9	260	330.8	259	345.3
	R-ACA	887	236.3	825	282	695	311	719	330.6
	A- LFPF -ACA	777	231.3	809	277.6	839	305.6	855	324.9
I_2	D- LFPF -ACA	880	241.4	647	287.5	691	316.2	726	335.2
	A-AAMF-ACA	916	233	821	278.3	980	307.4	659	326.9
	D-AAMF-ACA	874	239.3	817	285.8	754	314.9	575	335.1
	R-ACA	2126	239.2	2168	280.3	1535	305.5	1722	323.5
	A- LFPF -ACA	2089	234.2	1775	274.7	1524	299.2	1792	316.8
I_3	D- LFPF -ACA	1762	244.7	1888	286.4	1547	311.8	1519	329.8
	A-AAMF-ACA	1926	236	2090	276.3	1703	301.4	2006	318.2
	D-AAMF-ACA	1982	242.9	1760	284.9	1354	310.1	1598	328.4

In Table 5–10, LFPF achieved higher improvement compared with the original algorithm under the eight label densities of the experiment, which indicates that LFPF is obviously superior to the original algorithm in performance and robustness. LFPF-SA-A, LFPF-GA-A, and LFPF-ACA-A label quality scores were reduced by 3.2~6.1, 3~6.2, and 4.3~6.7 compared with the original algorithm respectively. Besides, compared with A-AAMF-SA, A-AAMF-GA, and A-AAMF-ACA, label quality scores were reduced by 0.1~1.8, 0.2~2.6, 0.4~2.2, respectively. In addition, among the five label ordering. The A-LFPF algorithm achieved the best label placement results, followed by the A-AAMF algorithm, then the original algorithm, and the D-LFPF algorithm with the D-AAMF algorithm achieved the worst label results. This shows that the proposed label ordering (A-LFPF) in this paper can effectively help the algorithm jump out of the local optimum trap and guides the label placement toward a better direction efficiently, getting a better labeling result. Giving priority to the points with low interference intensity can greatly reduce the impact of the labeled point on the points to be labeled.

Table 10. The statistical results of ACA with label density range from 25% to 40%.

Instance	Algorithm	ρ=25%		ρ=30%		ρ=35%		ρ=40%	
		T	E	T	E	T	E	T	E
I_1	R-ACA	224	352	226	362.1	225	368.9	290	376.7
	A-PFCLP -ACA	231	346.9	233	357.3	192	363.4	187	371.3
	D-PFCLP -ACA	214	359.2	244	369.2	204	376.3	189	383.7
	A-AAMF-ACA	174	348.1	167	358.2	217	364.8	221	371.4
	D-AAMF-ACA	197	358.3	228	368	188	375.2	262	382
I_2	R-ACA	768	345.6	678	358.8	571	368.7	711	378
	A-PFCLP -ACA	605	340.7	639	353.7	475	364	646	372.6
	D-PFCLP -ACA	523	351	454	364.2	506	374	513	383.8
	A-AAMF-ACA	772	341.4	564	355	775	364.1	608	373.5
	D-AAMF-ACA	672	349.1	633	362.9	605	373	524	382.9
I_3	R-ACA	1817	336.3	2029	349.3	1518	358.5	1883	367.1
	A-PFCLP -ACA	1894	329.7	1140	343.5	1293	352.1	1676	360.7
	D-PFCLP -ACA	1284	343.5	1458	356.4	1452	365.2	1350	374.1
	A-AAMF-ACA	1718	331.5	1558	344.6	1500	352.8	1481	361.8
	D-AAMF-ACA	1368	342.1	1779	354.4	1218	363.8	1328	372.7

Figure 13 represents the plot of the difference in scores between A-LFPF-SA and SA, A-LFPF-GA and GA, and A-LPFPF-ACA and ACA at different densities. The blue dots in the graph indicate the mean value, and the ends of the yellow line indicate the maximum and minimum value of the score difference. The average difference values of SA, GA, and ACA are 4.9, 5.1, and 5.6 respectively, and the average difference values are greater than 0. And the minimum difference values of ACA and SA are greater than 0, which has high stability. The ACA has a tendency to label at a higher label priority because it considers the label priority to develop a label conflict penalty mechanism to update the pheromone concentration, which tends to label at a higher label priority regardless of the label order used and therefore has a higher stability. SA is more stable due to its larger initial temperature, smaller termination temperature, and larger maximum number of iteration repetitions. However, GA has a minimum difference of partial label density of less than 0 and is slightly less stable. So in terms of stability, A-LFPF is the best in combination with SA and ACA. However, due to the instability of GA, the maximum difference values of GA improve the most compared to SA and ACA. The crossover and mutation operations of the GA make it capable of jumping out of the local optimum trap. Both A-LFPF-GA and GA have large randomness, and the difference between the minimum and maximum values of the final results obtained is large, so the difference between A-LFPF-GA and GA under partial label density exists less than 0.

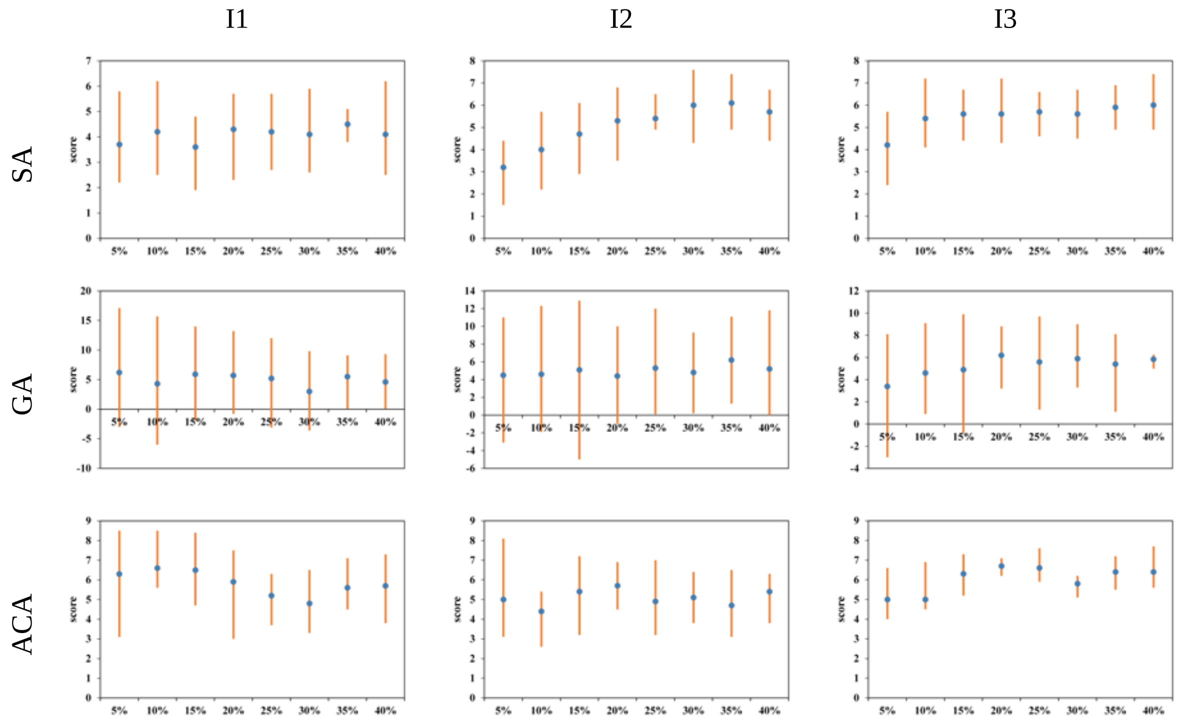
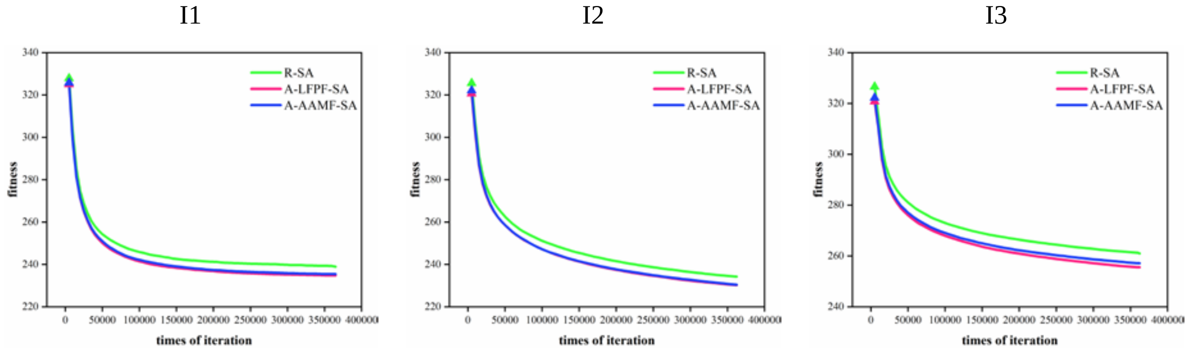
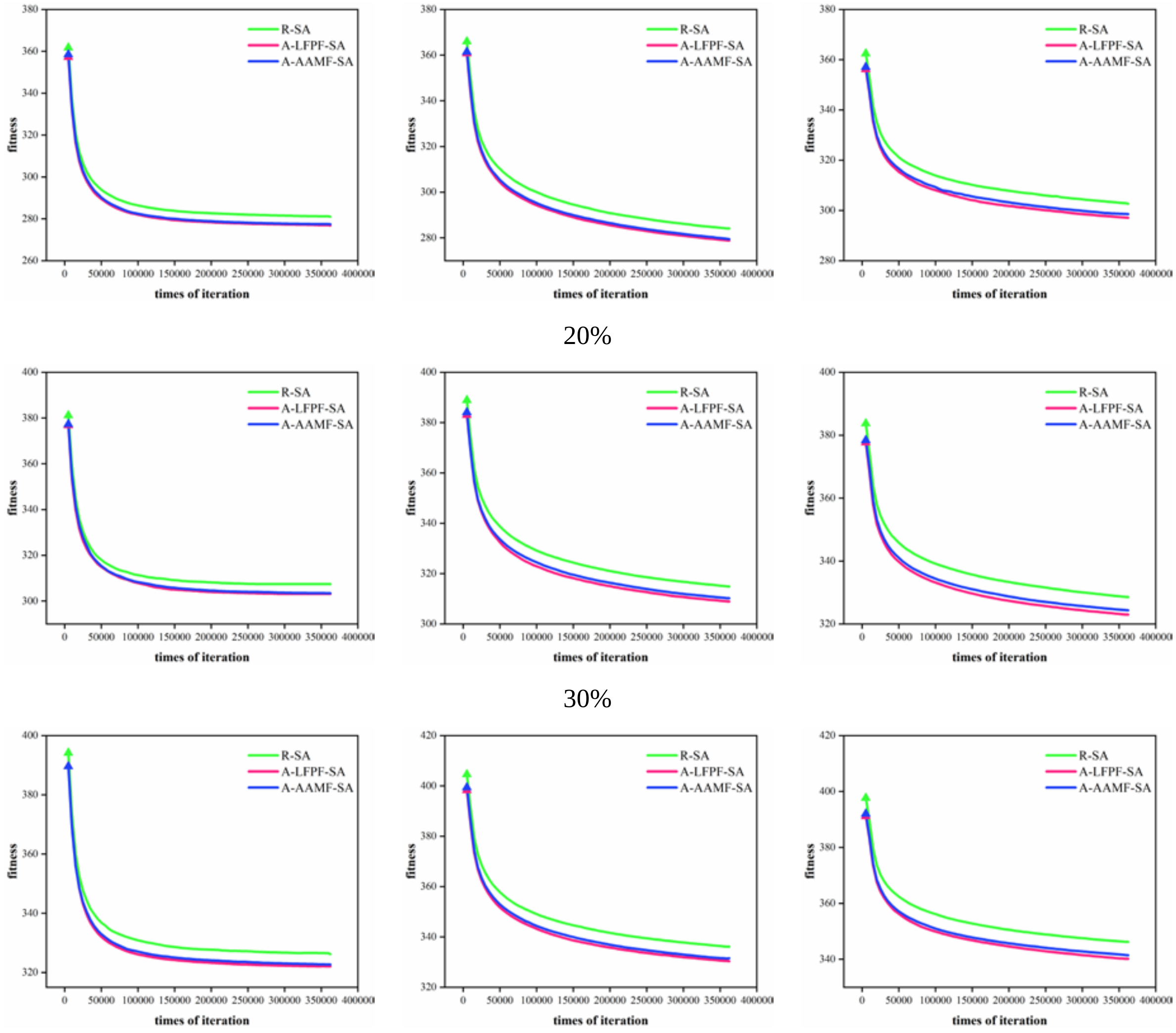


Figure 13. The difference in fitness scores between A-LFPF and the original algorithm.

The three cases in Figure 14 show the comparison of the trend (f) of the evaluation function values of the three algorithms for A-LFPF-SA, A-AAMF-SA, and SA at different densities. We use this to compare and analyze A-AAMF, A-LFPF, and R, and we can find: (1) Comparing the algorithm of this paper with SA, we can find that the algorithm A-LFPF-SA of this paper is significantly better than SA for different label densities and cases. A-LFPF-SA produced better initial solutions and outperformed SA in every iteration. Preferentially placing the intrusive points can reduce the interference on the later to be placed labels to obtain a better label placement result. (2) Compared with A-AAMF, A-LFPF scores were lower and the quality of the label result was slightly improved. Furthermore, in most cases, the initial solution of A-LFPF is better than A-AAMF and performs better than A-AAMF in each iteration. This suggests that LFPF has a more reasonable ability to quantify conflicts between point features than AAMF, providing a better label ordering for point feature label placement. A-LFPF makes the point features with less interference to be labeled first for better label results.



10%



20%

30%

40%

Figure 14. Comparison chart of labeling quality scores of three algorithms under different densities.

3.3. Discussion

The proposed algorithms in this paper have been evaluated through experimental results, which are discussed in detail in the following sections. These discussions aim to provide insights into the performance of the algorithms for PFLP and their importance in solving this problem.

In the first section, we discuss the label ordering based on LFPF and random label ordering. Our experiments show that the optimal label result is obtained by constructing the label ordering in ascending order (A-LFPF) according to the label frequent pattern framework. D-LFPF makes more interference points be labeled first, which leads to more points in the later labeling order being affected and unable to find an empty space to label. On the contrary, giving priority to the points with low interference intensity can greatly reduce the impact of the labeled point on the points to be labeled. It guides the label placement in a better direction efficiently, getting a better labeling result. Random label ordering ignores the internal spatial structure and order of datasets, increasing the uncertainty and ambiguity in the solution process. This demonstrates the importance of constructing a label ordering that considers the frequent pattern of the label in solving PFLP. It can effectively help the algorithm jump out of the local optimum trap and guides the label placement toward a better direction efficiently, getting a better labeling result.

In the second section, we compare the annotation association model framework (AAMF) and label frequent pattern framework (LFPF). Compared with the algorithm A-AAMF, the label quality of A-LFPF was slightly improved. AAMF quantifies the interference of the point feature by using the number of point features associated with LBP-MBR of the point feature. The effect of affecting one label position rectangle of the point feature is the same as that of multiple label position rectangles of the point feature. However, LFPF quantifies the interference intensity by counting all the frequent values of LBP-MBR of the point feature. LFPF takes into account the size of the LBP-MBR, quantifying interference capacity is more reasonable. By proposing a more reasonable interference intensity calculation method, this work presents a more effective label ordering construction algorithm, which can provide better performance for PFLP.

In the third section, we discuss the stable integration of the algorithms. Our experiments show that the combination of A-LFPF with the SA and ACA algorithms is more stable, while it is less stable when combined with the GA algorithm. Some results from the A-LFPF-GA experiment are inferior to those obtained using the GA algorithm alone. This is because the crossover and variation of the GA algorithm make it more random and therefore less stable. A suitable and stable metaheuristic algorithm should be selected in the actual label placement process.

In the fourth section, we discuss the performance evaluation of the indexes used in our experiments. We compared the performance of three indexes (GI, BI, and RI) in terms of conflict detection efficiency. The grid index is equivalent to a local traversal index, search and query times of BI can be reduced by nearly 7 times, greatly improving the conflict detection efficiency. Due to the corresponding spatial regions of the brother nodes of the R-tree can overlap, its index has redundant paths for searching thus wasting some search time. This demonstrates the effectiveness of the proposed index in conflict detection.

Overall, the algorithms presented in this paper can help improve the accuracy and efficiency of labeling, which is crucial for point-feature label placement. However, there are still some shortcomings: (1) The constructed order of the ascending label is static. In the future, it can be considered to update the label order in real time during the label process according to its impact capacity. (2) Data mining is not performed on the specific label candidate positions, and further guidance is needed for point label candidate positions to improve the efficiency and quality of the algorithm. (3) A more reasonable ambiguity factor based on the human eye resolution should be developed in the quality evaluation function. (4) The combination with the genetic algorithm is somewhat unstable, and subsequent attempts can be made to add local search into it to improve the stability of the algorithm.

4. Conclusions

Aiming at the problem of automatic label placement of dense and complex points, we proposed a PFLP algorithm based on spatial data mining. In this algorithm, we construct the label frequent pattern framework of point features to describe the interference among features in the labeling placement by fully mining the spatial distribution characteristics of features and the correlation among labels. This pattern can provide a judgment basis for designing placement priorities. In addition, an ascending label order based on the label frequent pattern framework is constructed to reduce the interference between point features. Prioritizing labels on features of lesser impact can reduce the impact on subsequent point features. Moreover, in the label placement process, we proposed a bit-based grid space index method, which can not only reduce the cache space occupied by the dataset but also improve the indexing efficiency of conflict detection, and further improve the labeling efficiency.

The label frequent pattern framework is combined with simulated annealing, genetic algorithm, and ant colony algorithm to verify the superiority of the algorithm in label placement under different label densities in test instances. The experimental results show that the proposed label frequent pattern framework has some universality, and the quality and efficiency of the label for each instance under different label densities are improved with the compared algorithms.

Acknowledgments

This work was supported by The Excellent Youth Foundation of Henan Municipal Natural Science Foundation (212300410096), Program of Song Shan Laboratory (Included in the Management of Major Science and Technology Program of Henan Province) under Grant number 221100211000-03, and The National Key R&D Plan of China (2018YFB0505304).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. X. Qin, Y. Luo, N. Tang, G. Li, Making data visualization more efficient and effective: A survey, *VLDB. J.*, **29** (2020), 93–117. https://doi.org/10.1007/s00778-019-00588-3
2. M. Aparicio, C. J. Costa, Data visualization, *Commun. Design Quart. Rev.*, **3** (2015), 7–11. https://doi.org/10.1145/2721882.2721883
3. S. Elwood, Geographic Information Science: Visualization, visual methods, and the geoweb, *Prog. Hum. Geogr.*, **34** (2011), 401–408. https://doi.org/10.1177/0309132510374250
4. A. C. Robinson, U. Demšar, A B. Moore, A. Buckley, B. Jiang, K Field, et al. Geospatial big data and cartography: Research challenges and opportunities for making maps that matter, *Int. J. Cartogr.*, **3** (2017), 32–60. https://doi.org/10.1080/23729333.2016.1278151
5. A. Lhuillier, M. V. Garderen, D. Weiskopf, Density-based label placement, *Vis. Comput.*, **35** (2019), 1041–1052. https://doi.org/10.1007/s00371-019-01686-7
6. J. She, J. Liu, C. Li, J. Li, Q. Wei, A line-feature label placement algorithm for interactive 3D map, *Comput. Graph-UK.*, **67** (2017), 86–94. https://doi.org/10.1016/j.cag.2017.06.002
7. Y. Li, M. Sakamoto, T. Shinohara, T. Satoh, Automatic label placement of area-features using deep learning, *Int. Arch. Photogr. Remote Sensing Spat. Inform. Sci.*, **43** (2020), 117–122. https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-117-2020
8. J. She, X. Li, J. Liu, Y. Chen, J. Tan, G. Wu, A building label placement method for 3D visualizations based on candidate label evaluation and selection, *Int. J. Geogr. Inf. Sci.*, **119** (2019), 123–138. https://doi.org/10.1080/13658816.2019.1606431
9. J. Christensen, J. Marks, S. Shieber, An Empirical Study of Algorithms For Point-Feature Label Placement, *ACM Transactionson Graphic*, **14** (1995), 203–232. https://doi.org/10.1145/212332.212334
10. I. H. Osman, J. P. Kelly, Meta-Heuristics Theory and Applications, *J. Oper. Res. Soc.*, **48** (1997), 657–657. https://doi.org/10.1007/978-1-4613-1361-8

11. M. Yamamoto, G. Camara, L. A. N. Lorena, Tabu search heuristic for point-feature cartographic label placement, *GeoInformation*, **6** (2002), 77–90. https://doi.org/10.1023/A:1013720231747
12. S. Zoraster, Practical results using simulated annealing for point feature label placement, *Cartogr. Geogr. Inf. Sci.*, **24** (1997), 228–238. https://doi.org/ 10.1559/152304097782439259
13. S. P. Gomes, L. A. N. Lorena, G. M. Ribeiro, A constructive genetic algorithm for discrete dispersion on point feature cartographic label placement problems, *Geogr. Anal.*, **48** (2016), 43–58. https://doi.org/10.1111/gean.12082
14. S. Peng, Y. Song, F. Wu, The research of intelligent point-feature cartographic label placement base on ant colony algorithm, *Sci. Survey. Map.*, **32** (2007), 80–81, https://doi.org/10.3771/j.issn.1009-2307.2007.05.029
15. G. L. Cravo, G. M. Ribeiro, L. A. N. Lorena, A greedy randomized adaptive search procedure for the point-feature cartographic label placement, *Comput. Geosci.*, **34** (2008), 373–386. https://doi.org/10.1016/j.cageo.2007.01.007
16. R. L. Rabello, G. R. Mauri, G. M. Ribeiro, L. A. N. Lorena, A clustering search metaheuristic for the point-feature cartographic label placement problem, *Eur. J. Oper. Res.*, **234** (2014), 802–808. https://doi.org/ 10.1016/j.ejor.2013.10.021
17. E. J. Araújo, A. A. Chaves, L. A. N. Lorena, Improving the Clustering Search: An application to cartographic labeling, *Appl. Soft. Comput.*, **77** (2019), 261–273. https://doi.org/10.1016/j.asoc.2018.11.003
18. Y. Ding, N. Jiang, C. Wu, X. Zhou, A two-phase algorithm for point-feature cartographic label placement, *Earth. Sci. Inform.*, **11** (2018), 183–203. https://doi.org/ 10.1007/s12145-017-0320-8
19. J. Li, Q. Dong, a genetic taboo search algorithm for point-feature label placement considering the constrain of network, *Bull. Survey. Map.*, **2** (2019), 80–85. https://doi.org/10.13474/j.cnki.11-2246.2019.0048
20. F. Lu, J. Deng, S. Li, H. Deng, A hybrid of differential evolution and genetic algorithm for the Multiple Geographical Feature Label Placement Problem, *ISPRS Int. J. Geo-Inf.*, **8** (2019), 237. https://doi.org/10.3390/IJGI8050237
21. J. Deng, Z. Guo, M. N. Lessani, Multiple geographical feature label placement based on multiple candidate positions in two degrees of freedom space, *IEEE Access*, **9** (2021), 144085–144105. https://doi.org/10.1109/ACCESS.2021.3120289
22. A. C. F. Alvim, E. D. Taillard, POPMUSIC for the point feature label placement problem, *Eur. J. Oper. Res.*, **192** (2009), 396–413. https://doi.org/10.1016/j.ejor.2007.10.002
23. X. Zhou, Z. Sun, C. Wu, Y. Ding, Automatic Label Placement of Point Feature: Using Ant Colony Algorithm Based on Group Clustering, *J. Geo-Inform. Sci.*, **17** (2015), 902–990. https://doi.org/10.3724/SP.J.1047.2015.00902
24. W. Cao, F. Peng, X. Tong, H. Dai, Y. Zhang, A point-feature label placement algorithm considering spatial distribution and label correlation, *Acta Geodaetica et Cartographica Sinica*, **51** (2022), 301–311. https://doi.org/ 10.11947/j.AGCS.2022.20210247
25. Z. Zhang, J. Yang, A discrete cuckoo search algorithm for traveling salesman problem and its application in cutting path optimization, *Comput. Ind. Eng.*, **169** (2022), 108157. https://doi.org/10.1016/j.cie.2022.108157
26. J. Zheng, Y. Hong, W. Xu, W. Li, Y. Chen, An effective iterated two-stage heuristic algorithm for the multiple Traveling Salesmen Problem, *Comput. Ind. Eng.*, **143** (2022), 105772. https://doi.org/10.1016/j.cor.2022.105772

27. M. Huang, F. Wang, S. Wu, The implementation of multiobjective flexible workshop scheduling based on genetic simulated annealing-inspired clustering algorithm, *Wirel. Commun. Mob. Comput.*, **2022** (2022), 1–11. https://doi.org/10.1155/2022/7452638
28. J. Mou, K. Gao, P. Duan, J. Li. A machine learning approach for energy-efficient intelligent transportation scheduling problem in a real-world dynamic circumstances, *IEEE Trans. Intell. Transp. Syst.*, **99** (2022), 1–13.
29. Í. Santana, A. Plastino, I. Rosseti, Improving a state-of-the-art heuristic for the minimum latency problem with data mining, *Int. Trans. Oper. Res.*, **29** (2022), 959–986. https://doi.org/10.1111/itor.12774
30. D. Martins, G. M. Vianna, I. Rosseti, S. L. Martins, A. Plastino, Making a state-of-the-art heuristic faster with data mining, *Ann. Oper. Res.*, **263** (2018), 141–162. https://doi.org/10.1007/s10479-014-1693-4
31. M. Guerine, I. Rosseti, A. Plastino, A hybrid data mining heuristic to solve the point-feature cartographic label placement problem, *Int. Trans. Oper. Res.*, **27** (2020), 1189–1209. https://doi.org/10.1111/itor.12666
32. G. Luo, B. Xu, The study on automatic name placement around point features based on Voronoi, *J. Chang'an University (Earth Science Edition)*, **2** (2003), 63–65. https://doi.org/10.3969/j.issn.1672-6561.2003.02.016
33. L. Li, H. Zhang, H. Zhu, W. Hu, A point-feature labeling algorithm based on movable regions, *Geom. Inform. Sci. Wuhan University*, **43** (2018), 1129–1137. https://doi.org/10.13203/j.whugis20160289
34. J. Qi, Y. Tao, Y. Chang, R. Zhang, Packing R-trees with Space-filling curves: Theoretical optimality, empirical efficiency, and bulk-loading parallelizability, *ACM Trans. Database Syst.*, **45** (2020), 1–47. https://doi.org/10.1145/3397506
35. X. Tong, C. Cheng, R. Wang, L. Ding, Y. Zhang, G. Lai, et al., An efficient integer coding index algorithm for multi-scale time information management, *Data Knowl. Eng.*, **119** (2019), 123–138. https://doi.org/10.1016/j.datak.2019.01.003
36. A. Marín, M. Pelegrín, Towards unambiguous map labeling – Integer programming approach and heuristic algorithm, *Expert Syst. Appl.*, **98** (2018), 221–241. https://doi.org/10.1016/j.eswa.2017.11.014
37. T. Strijk, M. V. Kreveld, Practical Extensions of point Labeling in the Slider Model, *Geo. Inform.*, **6** (2002), 181–197. https://doi.org/ 10.1145/320134.320148
38. C. Chee, J. Jaafar, I. A. Aziz, M. H. Hasan, W. Yeoh, Algorithms for frequent itemset mining: A literature review, *Artif. Intell. Rev.*, **52** (2019), 2603–2621. https://doi.org/10.1007/s10462-018-9629-z
39. S. V. Dijk, M. V. Kreveld, T. Strijk, A. Wolff, Towards an evaluation of quality for names placement methods, *Int. J. Geogr. Inf. Sci.*, **16** (2002), 641–661. https://doi.org/10.1080/13658810210138742
40. M. A. Rylov, A, W. Reimer, Improving label placement quality by considering basemap detail with a raster-based approach, *GeoInformatica*, **19** (2015), 463–486. https://doi.org/10.1007/s10707-014-0214-6