IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 30, 2021, accepted October 3, 2021, date of publication October 14, 2021, date of current version October 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120289

Multiple Geographical Feature Label Placement Based on Multiple Candidate Positions in Two Degrees of Freedom Space

JIQIU DENG, (Member, IEEE), ZHIYONG GUO, AND MOHAMMAD NASER LESSANI

School of Geosciences and Info-Physics, Central South University, Changsha 410083, China

Key Laboratory of Metallogenic Prediction of Nonferrous Metals and Geological Environment Monitoring (Central South University), Ministry of Education, Changsha 410083, China

Corresponding author: Mohammad Naser Lessani (naser11@csu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Project 2017YFC0601503 and Project 2018YFC0603902, and in part by the National Natural Science Foundation of China under Project 41401532.

ABSTRACT Automatic multiple geographical feature label placement (MGFLP) is a combinatorial optimization problem shown to be an NP-hard problem, and it is a challenge in automatic cartography. Many automatic label placement algorithms for point, line, and area features were put forward. It is a common way to use multiple candidate positions (MCP) for label placement, but the research in this way mostly focuses on point features and does not take all three types of features and all the possible candidate positions into account on the map. Therefore, in this paper, the concept of degrees of spatial freedom for feature label placement is proposed based on the idea of degrees of freedom of mechanical motion. We define the degrees of freedom (DOF) and its space for feature labels on a planar map so as the potential space, including all the optional candidate positions of each feature label, can be standardized. Based on two degrees of freedom (2-DOF) space, feature reference position (FRP), and certain buffer distance (CBD) from FRP, we studied the methods including generating, calculating, evaluating, and selecting MCP for feature label. By using and improving the discrete differential evolution genetic algorithm (DDEGA), we carried out MGFLP experiments on the same dataset used by DDEGA algorithm. The results show that: 1) although the MCP based on the 2-DOF space increase the complexity of the NP-hard problem, however, the obtained results by optimizing the performance of the algorithm and increasing the number of candidate positions are still better than the traditional 8-candidate positions model. 2) In the same 2-DOF space, increasing the candidate positions from less to more along each direction of the 2-DOF space improves the quality of label placement.

INDEX TERMS Feature label placement, NP-hard problem, discrete differential evolution and genetic algorithm, multiple candidate positions, two degrees of freedom space.

I. INTRODUCTION

Accurate positioning of the labels on the map is one of the fundamental steps to visualize map information clearly, and it requires great attention from the cartographers. Since the 1960s, many studies have been done on automatic label placement of features. However, automatic labeling is still a technical problem, which is also one of the hotspots in computer graphics. The most common challenges on automatic label placement are label conflict, label-feature conflict, and association of label features. Many attempts have been made through decades of research to address these obstacles in automatic cartography. In addition, it has been proven that label placement is a non-deterministic polynomial-time hard (NP-hard) problem [1]. Therefore, it is extremely difficult to estimate the ideal label position.

Geographical features on the planar map generally fall into three categories. Point features represent cities, well, sewers, etc. Line features show the linear infrastructures such as roads, streets, rivers, etc. Area features represent countries, buildings, etc. The problem of label placement is significantly different for the three categories.

High-quality map labeling is the main focus of automatic label placement. Whereas, multiple geographical features consist of different geographic feature types, the procedure of label placement is more challenging compared to label the

The associate editor coordinating the review of this manuscript and approving it for publication was Tu Ngoc Nguyen.

single type of geographic feature because of considering all three types of features during the process of label placement. However, most of the existed studies mainly focused on label placement of single type only or in limited space, and cannot handle the problem of multiple geographical feature label placement as cross-layer and it will be described in detail in the next section. Moreover, to the best of our knowledge, all the existed researches only considered the reference position as the possible space for candidate position of the label for each feature which makes their work less applicable to label geographical features. Therefore, due to the lack of the insufficient degree of freedom space, all generated positions of each feature are potentially in conflict with features and other label positions while taking into account exclusively the reference position, in this case, the elimination of label conflict and label-feature conflict decreases. Lu *et al.* [2] presented the DDEGA algorithm for multiple geographical feature label placement based on one degree of freedom space. However, the orientation of the label for area features and the problem of small-area features are not regarded which can cause significant ambiguity on the map. Additionally, the execution time of the algorithm is extremely long which disables their approach to obtain the best solution from the set of generated positions in a reasonable execution time.

This paper aims to improve the procedure of candidate position generation and take an appropriate optimization algorithm to find the best candidate position from a large-scale of solutions to improve the quality of multiple geographical feature label placement (MGFLP). The proposed algorithm is based on the idea of degrees of freedom of mechanical motion. By expanding the degrees of freedom space for feature label placement, the algorithm can explore the most appropriate label position in a certain distance and different orientation from the feature, and realize the cross-layer feature label placement finally. (The contribution of this work is described below:

1) Expanding the degrees of spatial freedom from one degree to two degrees of freedom (2-DOF) space for multiple geographical feature label placement.
2) Generation of multiple candidate positions for multiple geographical features in 2-DOF space.
3) A comprehensive quality evaluation model is established to evaluate and select the best generated positions. The quality evaluation model consists of label position priority for area features in addition to label conflict, label-feature conflict, ambiguity factor, and label position priority for point features.
4) The problem of small-area features is solved by placing the corresponding label outside the boundary of the features with respect to the neighboring labels.n 3, reviewer 2:related work from the introduction)

II. RELATED WORK

Automatic label placement has been a major focus in geographic information science and many approaches have been presented to address this problem. Most of the presented methods follow the same procedure which are candidate positions generation, evaluation of generated positions, and select one label position from the set of generated positions for each feature based on the evaluation model [3].

For point feature labeling, the most common model is the fixed-position model by using several fixed positions as candidate positions to label each point feature. A lot of attention has been given to use this model for label placement. Yoeli is one of the earliest researchers who studied automatic label placement, proposed an 8-bearing placement method in point feature labeling [4]. Zhou *et al.* put forward an oval multi-orientations and multi-levels cartographic potential label position scheme, which parameterized and diversified the candidate positions of point feature [5]. Qiao *et al.* regarded the candidate position space of the point feature as a set of innumerable circles, which includes multiple candidate positions, and further discussed candidate position priority [6]. Li *et al.* proposed the point feature label placement method by using movable areas and selected those free spaces for labels around point features that were not in conflict with features [7]. More recently, Lei *et al.* [8] used a hexagonal grid for point features and achieved high-quality label placement. Another common model for point feature labeling is the slider model, which can make better use of the blank area of the map through the continuous sliding strategy, but the position of the label is limited to this trajectory line. Recently, Ding *et al.* [9] proposed an algorithm based on a four-slider model to place the label of point features.

The label mode of line features and area features can fall into point positioning label mode and line positioning label mode. Doerschler and Freeman [10] developed a software system to label various features with consideration of feature label association. Barranult [11] presented an approach that could analyze the main axes which constrain label placement and place the labels of road administrative. Wolff *et al.* [12] introduced a method for high-quality line feature labeling. The method allowed curved labels with the runtime of $O(n^2)$ in the worst-case. Several studies placed the labels on the line features, or they generated parallel lines based on line features at first and then placed the labels on the parallel lines. Sun *et al.* [13] put forward an approach for line feature which include generation of candidate curve along the line features, producing a chain of points that present the candidate position, and elimination of overlaps and ambiguity from the set of generated positions. More studies can be found about line feature labeling in these works [14]–[17].

While placing labels for area features, the cartographers may encounter two situations. One situation is that the labels of area features can be placed inside the area features. In this case, for positioning the labels inside the area, the concept of the line labeling model is used. For example, Yoeli [4] introduced a method that placed the labels in a horizontal straight line passing through the center of area features. Ahn and Freeman [18] presented the method of adding the label to the skeleton line of area feature. Ebinger and Goulette [19] proposed an algorithm that used parallel lines to cut the area features to get the approximate skeleton line inside the area features and then positioned the labels of area features along the obtained skeleton line. Li *et al.* [20] applied the concept of deep learning for label placement of area features. Another situation is that the labels of area features which places outside the boundary of area features due to the lack of enough space inside the features. In this case, the area features are assumed as point features. For example, Rylov and Reimer [21] proposed a novel algorithm that placed the label outside the area feature if the feature did not have enough internal space. In many prior automatic label placement algorithms, the feature label was often restricted to a specific area around features. This fails to consider the distance from the label to its corresponding feature, and the quantity and orientation of the label. Without considering the aforementioned parameters, the improvement of label placement quality is limited.

Since it has been proven that label placement is an NP-hard problem [1], and the optimal position for labels has to be identified from a large set of solutions. The common algorithms applied to solve this problem include the Greedy algorithm [22], [23], Backtracking algorithm [24], Simulated Annealing algorithm [25], Ant colony optimization algorithm [26], and Tabu Search algorithm [27], [28]. Most of the above algorithms focused on automatic label placement of point features, and less attention has been given to label multiple geographical features jointly. In the study on the cross-layer feature label placement [29]–[32], it is rarely thought about the orientation and distance of the label from the related feature when selecting candidate position for the label. This causes ambiguity for the readers and has a negative effect on the map visualization. Lu *et al.* [2] proposed hybrid discrete differential evolution and genetic algorithm (DDEGA) to realize the cross-layer feature label placement. DDEGA algorithm can effectively solve the problem of multiple geographical feature label placement to a certain extent. However, the available space for label candidate positions in the DDEGA algorithm was still limited by only using eight candidate positions, and it restrains the improvement of label placement.

The rest of this paper is organized as follows. In Section III, the procedure of generating multiple candidate positions for multiple geographical feature and evaluation model based on 2-DOF space is explained; In Section IV, the experiments and results are discussed in detail; In Section V, the experiments analyses are discussed; In Section VI, the paper is concluded, and also some potential future works are mentioned.

III. MULTIPLE CANDIDATE POSITIONS GENERATION AND EVALUATION BASED ON TWO DEGREES OF FREEDOM SPACE

A. DEGREES OF SPATIAL FREEDOM FOR FEATURE LABEL PLACEMENT

The candidate position is a set of all possible label positions selected around the point and line features, around or inside the area features. Theoretically, an infinite number of positions can be generated for each feature, and the angle of the label on each candidate position is changeable. The changes of label position and angle for feature labeling are similar to the changes of position and angle for components moving in the mechanical system. The mechanical plane component has three degrees of freedom including the coordinates X and Y of any point, and the angle between the vertical and horizontal axes. Similarly, the plane feature label has three degrees of freedom including the coordinates X and Y of any point, and the angle between the label direction and the horizontal axis θ, as shown in Fig. 1.

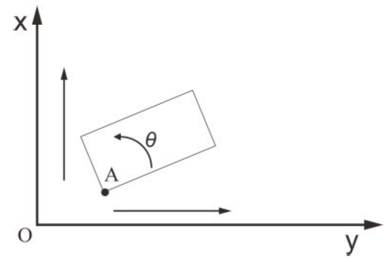


FIGURE 1. The direction of three degrees of freedom for feature label placement.

The label box can move along the direction of the x-axis, y-axis and rotate around a certain point (point A), as shown in Fig.1. Here, according to the degrees of freedom of mechanical motion as a reference, the degree of freedom space for the feature label placement is defined as the number of independent position parameters that must be given.

The relationship between the label and its corresponding feature cannot be directly reflected from the label coordinates. Therefore, referring to the definition of plane polar coordinates, the degrees of freedom of x and y axes are converted into two other expressions, reference position (R) and buffer distance (B). The degree of freedom of R is determined by the orientation relationship between the label and its corresponding feature. The degree of freedom of B is determined by the minimum and maximum distance between the label and the reference position or the feature. As shown in Fig. 2(a-d), R represents the reference position of different features, B represents the buffer distance, and the arrow indicates the direction of the two degrees of freedom space for feature label placement. Where b_{min} and b_{max} represent the minimum and maximum buffer distance, respectively, and the label can be placed in the buffer area from b_{min} to b_{max}. Thus, the degrees of freedom space for feature label placement can be converted into three degrees of freedom: the reference position R, the buffer distance B, and the rotation angle θ. The rotation angles for an individual label of different feature types should be different to ensure the beauty and clarity of the map based on cartographic standardization. The point label is generally placed horizontally. However, the labels of line and area features have three different directions, horizontal, vertical, and along the direction of the reference

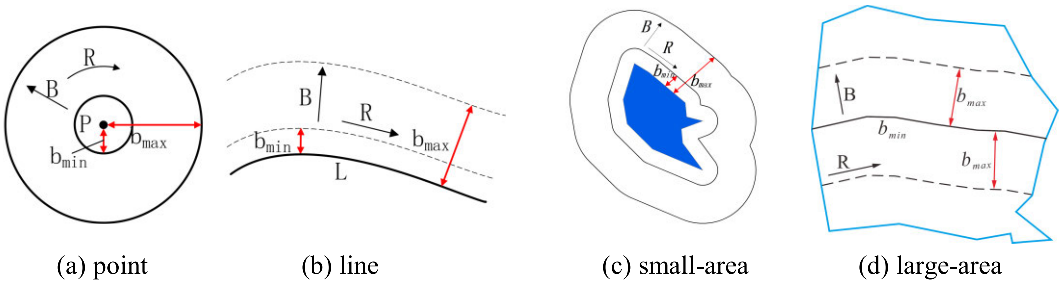


FIGURE 2. Degrees of spatial freedom for feature label placement.

line, at the same candidate position. Along a specific reference line of the line or the area feature, the direction of the candidate position can only be set to one of those three directions, see Section C. 3, according to the overall shape and trend of the corresponding feature. This shows that the label angle of a specific candidate position of the feature is generally fixed, and the third degree of freedom (rotation angle θ) of the two-dimensional map can be ignored. Therefore, in this study, only the reference position and the buffer distance are considered as the degrees of freedom space for feature label placement.

B. TWO DEGREES OF FREEDOM SPACE FOR PLANE MAP FEATURE LABEL

As discussed in the previous section, the feature label has three degrees of freedom space on a two-dimensional map. However, only the reference position and the buffer distance are considered in this study, which is called two degrees of freedom (2-DOF) space.

Prior studies for placing the labels of point features only considered 4, 8, and 16 candidate positions based on one degree of freedom space which corresponds to the reference position of 2-DOF. In these multiple candidate positions model of point feature, the researchers did not consider all possible candidate positions. So the choice of candidate position for the label is limited, and the labels may not have enough positions to be placed. Moreover, the idea of multiple candidate positions is rarely used for line and area features, and the selection of candidate positions for labeling these two types of features is more limited than point features. In this case, as the limited number of candidate positions, the overlap and conflict at each label position occur on maps with intensive features with a high probability. If more positions are generated for the feature label in the 2-DOF space, the probability of overlap and conflict can be reduced. Because generating more potential candidate positions for each feature is beneficial to improve the feature label placement quality, and the label has more positions to be placed.

C. GENERATION OF MULTIPLE CANDIDATE POSITIONS BASED ON 2-DOF SPACE

For generating multiple candidate positions (MCP), N points along the reference position (R) and M points along the buffer distance (B) are selected by the proposed algorithm. According to the 2-DOF, N * M candidate positions are generated for each feature. N and M are any integers greater than 0. Theoretically, if the values of N and M reach infinity, the entire 2-DOF space will be filled with the generated candidate positions.

1) GENERATING MULTIPLE CANDIDATE POSITIONS FOR POINT FEATURE

For generating MCP of point feature, a circle whose center is the point feature and radius is the minimum buffer distance was taken as the reference position of the point feature in the 2-DOF space. Based on reference position, M equally spaced buffer circles were obtained around the point feature, then N equally spaced candidate positions were generated on each buffer circle. The labels were placed horizontally at each generated position. In this way, the candidate positions of point features distribute more evenly and discretely in 2-DOF space. The specific method and process are as follows. The point on the label box with the shortest distance from the circle with radius b is called the near rounded point, and the position of the near rounded point of the label box is different in each quadrant. The distance from b_{min} to b_{max} was equally divided into (M-1) parts, then generated M different buffer circles according to the equal distance $(b_{max} - b_{min})/(M - 1)$. N points were evenly selected on each buffer circle, so N * M candidate positions were obtained around each point feature. To ensure that the label box does not overlap the point feature itself, the point closest to the center of the circle, the point feature, of the label box can be selected as the candidate position of the point feature, and the coordinates of the central point of the label box should be determined according to the length of feature label. The distance (b) from the circle center closest point of the label box to the corresponding point feature and the angle (β) between the line of the circle center closest point and point feature and the horizontal direction were used in this scheme to identify the priority of the labels around point features. The smaller the β and the closer the b distance, the higher the priority of the candidate position.

Fig. 3(a) is the schematic diagram of the candidate position for the point label. As the figure presents, the coordinates of the point feature **P** is (x_i, y_i), the distance between the point

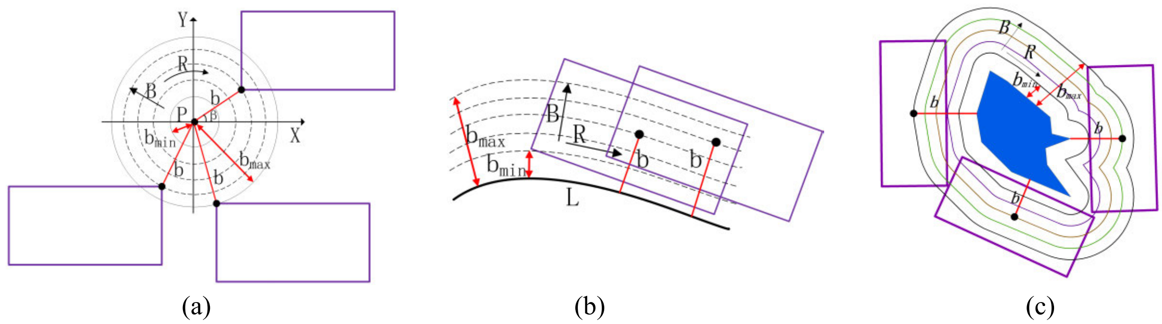


FIGURE 3. Candidate position scheme for feature label (a) point (b) line (c) small-area.

feature and the circle center closest point of the label box is b, and the angle between the point feature and the circle center closest point of the label box and the horizontal line of the map is β. According to the relationship of the coordinates, the coordinates of the circle center closest point of the label box are as follows.

$$X_i = x_i + b * \cos\beta, \quad Y_i = y_i + b * \sin\beta \tag{1}$$

2) MULTIPLE CANDIDATE POSITIONS GENERATION FOR LINE FEATURE

In the 2-DOF space of the line feature, the line feature itself was taken as the reference position. M equally spaced parallel lines were obtained based on line feature, and then N equally spaced candidate positions were generated on each parallel line by the proposed algorithm. The label was placed horizontally, vertically, or along the line trend direction based on the angle of the line feature, the suitable direction with the least conflict and the highest priority among the three placement modes will be select in the actual algorithm, and it uniformly distributes the candidate positions of line features above the feature in the 2-DOF space.

The parallel lines of line features were generated through the buffer method. The generated parallel lines were top-bottom and left-right parallel lines according to the angle α between the line feature and the horizontal direction of the line. When $\alpha \in [45°, 135°]$, it was as left-right parallel lines. Otherwise, it was defined as top-bottom. According to the line feature label placement, the top and right parallel lines are in higher priority than the bottom and left parallel lines. Therefore, only the top and the right parallel lines of the line feature were selected in the 2-DOF space for label placement of line features. The minimum and the maximum buffer distances are shown as b_{min} and b_{max}, respectively.

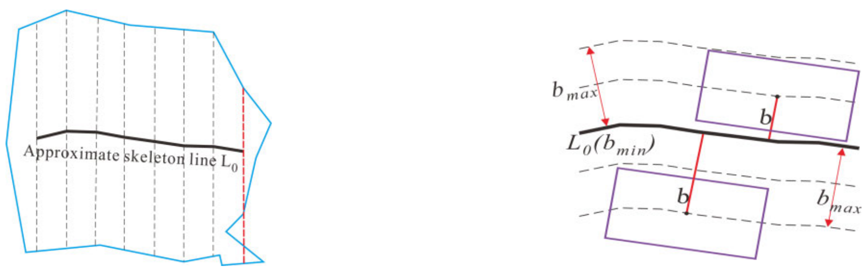
To generate multiple candidate positions for line feature, first, the distance between b_{min} and b_{max} was equally divided into (M-1) parts and buffered by the equal distance $(b_{max} - b_{min})/(\text{M} - 1)$ to obtain M parallel lines. Then each parallel line was divided into (N + 1) equal parts. Each line has N bisection points representing the center point of the label box of each candidate position, so a total of N * M candidate positions were generated. The distance between the parallel lines and the line feature is represented by b, and the schematic diagram of the candidate position scheme for the line feature label is shown in Fig. 3(b).

3) MULTIPLE CANDIDATE POSITIONS GENERATION FOR AREA FEATURE

For generating label position of area features, there are two ways to generate multiple candidate positions, including inside and outside of area feature in the 2-DOF space. If the area feature does not have enough interspace for its label, we term it as a small-area feature, and as a large-area feature conversely. To generate MCP for small-area features, a method similar to the point feature was used, and the reference position was distributed on the parallel curves of the area feature boundary. Fig. 3(c) presents the candidate position scheme for the small-area feature label. The blue area represents the small-area feature, and b_{min} and b_{max} show the minimum and maximum buffer distances, respectively. The space between the two buffer distances is the 2-DOF space of small-area features.

The candidate positions were generated based on approximate skeleton lines for large-area features. In this study, the approximate skeleton line was extracted using parallel lines to cut the area features in several segments. According to the length and width ratio of the area feature, the area features were cut by vertical or horizontal parallel lines, and then the midpoint of each parallel line was found. If a parallel line had multiple points which intersect with an area feature, the midpoint of the longest intersection segment in the area was used as the midpoint of the intersection of the parallel line and the area feature. A total of N midpoints were obtained in this way, and by connecting these midpoints, the approximate skeleton line was constructed. The label can be placed on the constructed approximate skeleton line (L_0), which is used as the reference position (b_{min}) in 2-DOF space, and L_0 is shown in Fig. 4(a). The maximum buffer distance (b_{max}) was used to generate parallel lines in the top and bottom parts of the approximate skeleton line, and the area between the top and bottom parallel lines can be used as the space to produce label position for the area feature.

In the region between the approximate skeleton line and the upper parallel line, $(\text{M} - 1)/2$ parallel lines were generated,



(a) (b)

FIGURE 4. The label placement scheme for large-area feature (a) extracting approximate skeleton line and (b) selecting candidate positions around approximate skeleton line.

Algorithm 1 Pseudo Code of N * M Candidate Positions Generation for Each Feature

Input features (F)
Define the maximum (max) and minimum (min) buffer distances
Set N and M values
for i=0 . . . F **THEN**
 Determine feature type (T)
 if T= Point **THEN**
  Get the length of the point label (l)
  Generate minimum buffer circle around T based on min value (b_{min})
  Generate maximum buffer circle around T based on max value (b_{max})
  D = ($b_{max} - b_{min}$ / M-1) // Equal distance
  Generate M buffer based on D distance between (b_{min} AND b_{max})
  N = (M / N+1)
  OUTPUT (N*M)
 elif T= Line **THEN**
  Get the length of the line label (l)
  line assumed as a minimum buffer (b_{min})
  Generate maximum buffer based on max value around T
  D = ($b_{max} - b_{min}$ / M-1) // Equal distance
  Generate M buffer between (b_{min} AND b_{max})
  N = (M / N+1)
  OUTPUT (N*M)
 else **THEN**
  Get the length of the area label (l)
  Check the width of area (w)
  if len (l) > w **THEN**
   Cut the area by vertical parallel lines (L)
  else **THEN**
   Cut the area by horizontal parallel lines (L)
  Find the midpoint of L (ML)
  Connect ML (approximate skeleton line (L_0))
  $b_{min} = L_0$
  Generate maximum buffer circle based on b_{max} around L_0
  D = ($b_{max} - b_{min}$ / M-1) // Equal distance
  Generate M buffer between (b_{min} AND b_{max})
  N = (M / N+1)
  if N*M conflict area itself **THEN**
   Generate N*M positions as point feature for this area feature
  OUTPUT (N*M)
END for

and the same number of parallel lines below the approximate skeleton line was obtained. Then each generated line was divided into (N + 1) parts, and N bisection points were generated on each line which presented the skeleton line closest point of the label box. Thus, a total of N * M candidate points of area features were achieved, which represented the midpoint of the label box. The area enclosed by the two lines of the topmost-bottommost and left-right vertical lines is the 2-DOF space of the large-area feature. The procedure of candidate position selection for area features follows the same principle of line label, which are three different modes, vertical, horizontal, or along the skeleton line trend. The procedure of candidate position selection around the skeleton line is shown in Fig. 4(b).

D. SCORING RULES

In the 2-DOF space, each feature has N * M candidate positions, which is a large number of solutions. Therefore, it is necessary to consider a comprehensive Scoring Rule that can accurately evaluate the candidate positions during the process of label placement.

While searching for the optimal solution, how to score each candidate position is the first issue to be considered in the 2-DOF space. The quality of feature label placement is directly affected by the quality evaluation model, which needs to be established extensively and it is the key factor to place the labels clearly. Many label placement criteria and Scoring Rules have been proposed over decades of research [1], [33]–[35]. Fan *et al.* [34] proposed the four basic evaluation factors of label position priority, relevance, overlap, and conflict. The Scoring Rules were formulated based on the research work of Fan in this study. Meanwhile, to avoid label conflict for the small-area feature and improve the beauty of map visualization, the Scoring Rules of each quality factor were redefined based on two degrees of freedom space. In the 2-DOF space, the Scoring Rule was defined differently based on the feature type, and four quality factors were considered in this study.

- Label ambiguity
- Label position priority
- Label conflict
- Label-feature conflict

The total number of input features is denoted as $\boldsymbol{Q}$, also to distinguish the feature type, $\boldsymbol{Q_1}$ stands for points, $\boldsymbol{Q_2}$ stands for lines, and $\boldsymbol{Q_3}$ stands for areas features. Each quality factor will be discussed in detail in the coming subsections.

1) LABEL AMBIGUITY FACTOR

The ambiguity factor addresses the association of label position and its corresponding feature. As a general rule, the closer the label is to its feature, the greater the correlation between label and feature, and vice versa. Here the distance between the center point of the label box and the midpoint of the line feature (D) was taken as the determinant to evaluate the ambiguity factor. The Scoring Rule is defined as follows.

$$T_{i1} = \left\{ \frac{D - D_{min}}{D_{max} - D_{min}}, \quad D_{min} \le D \le D_{max} \right. \tag{2}$$

$$S_1 = \sum\nolimits_{i=1}^{Q_2} T_{i1} \tag{3}$$

where D_{min} and D_{max} represent the minimum and maximum distances between the center of the label box and the midpoint of a line feature, respectively, T_{i1} represents the score of label ambiguity factor for the *i-th* label and S_1 represents the score of label ambiguity factor of all line features.

2) LABEL POSITION PRIORITY FACTOR

Label position priority refers to the orientation relationship between the label and its corresponding feature. This quality factor has a significant influence on the label placement of point and area features. Therefore, this quality factor was only considered for point and area features labels. The point feature label was scored according to the angle (β) between the circle center closest point of the label box and the point feature and the horizontal line of the map, as shown in Fig. 3(a). The Scoring Rule is defined as follows.

$$T_{i2} = \begin{cases} 0.25, & \beta \in [0^\circ, 90^\circ) \\ 0.5, & \beta \in [90^\circ, 180^\circ) \\ 0.75, & \beta \in [180^\circ, 270^\circ) \\ 1.0, & \beta \in [270^\circ, 360^\circ) \end{cases} \tag{4}$$

$$S_2 = \sum\nolimits_{i=1}^{Q_1} T_{i2} \tag{5}$$

where T_{i2} represents the score of label position priority factor for the *i-th* point feature, and S_2 represents the score of label position priority factor of all point features.

For the area feature, the score of label position priority was determined by the distance between the label and the midpoint of the approximate skeleton line of the area feature (L). The Scoring Rule is defined as follows.

$$T_{i3} = \left\{ \frac{L - L_{min}}{L_{max} - L_{min}}, \quad L_{min} \le L \le L_{max} \right. \tag{6}$$

$$S_3 = \sum\nolimits_{i=1}^{Q_3} T_{i3} \tag{7}$$

where L_{min} represents the minimum distance between the label and the midpoint of the skeleton line of the area feature, and L_{max} represents the maximum distance value. Where T_{i3} represents the score of label position priority factor for the *i-th* area feature, and S_3 represents the score of label position priority factor for all area features.

3) LABEL-FEATURE CONFLICT FACTOR

Another key challenge of map labeling is the overlap between labels and features. A map is limited to the length of it in the transmission of geographic information, and there is always the case that the features are overlapped. Since the elimination of label-feature conflict is unavoidable, however, it is necessary to reduce the number of label feature conflicts for the sake of legibility and clearness of the map. In addition, the label-feature conflict has a certain priority that a label cannot cover point features in order to avoid the loss of map information. Based on candidate position generation guidance in the 2-DOF space, the label of a small-area feature is placed outside the area feature. If the label overlaps the area feature itself, it is easy to mistakenly think that the label is for other features, which will cause more ambiguity. It is also necessary to ensure that the label of the small-area feature does not overlap with its area feature. To highlight the importance of point and small-area features, the score of label conflict of these two kinds of features should be set greater than the other quality factors in the Scoring Rule.

The label-feature conflict is categorized into three classes including scoring point, line, and area label-feature conflicts. The Scoring Rule for area features is categorized into label conflicts of small-area and large-area features. The Scoring Rule is defined as follows.

$$T_{i4} = \begin{cases} 0, & \text{Without label}-\text{feature conflict} \\ 1, & \text{Overlapping line feature} \\ 99, & \text{Overlapping point or small}-\text{area area feature} \\ \dfrac{99^{*}S_0}{S_r}, & \text{Overlapping large}-\text{area area feature} \end{cases} \tag{8}$$

where S_0 represents the ratio of overlapping of the label box of each large-area with the area feature, and S_r represents the area of each area label box.

Q_3 stands for area feature label, the number of overlaps between each area label and line feature is t_1, the number of overlaps between each small-area feature label and point features and its own area feature is t_2, the number of overlaps between each large-area feature label and point feature is t_3. Therefore, the label-feature conflict score of each area label is $P_{i1} = t_1 + 99^{*}t_2$ (small-area feature label) or $P_{i1} = t_1 + 99^{*}t_3 + (99^{*}S_0)/S_r$ (large-area feature label). The label-feature conflict score of all area features is S_A and the Scoring Rule is defined as follows.

$$S_A = \sum\nolimits_{i=1}^{Q_3} P_{i1} \tag{9}$$

When point and line labels overlapped the point feature, the ambiguity of label placement is larger. Therefore, the score of label conflict with point features was also set larger than the score of other label-feature conflicts, and the Scoring Rule is defined as follows.

$$T_{i5} = \begin{cases} 0, & \text{Without label}-\text{feature conflict} \\ 1, & \text{Overlapping line or area feature} \\ 99, & \text{Overlapping point feature} \end{cases} \tag{10}$$

There are a total of point features labels (Q_1) and line features labels (Q_2) on the map. The number of overlapping of line and area features with each point and line label is t_4, and the number of overlapping between labels and point features is t_5. Thus, the label-feature conflict score of each point and line label is $P_{i2} = (t_4 + 99^{*}t_5)$, and the label-feature conflict score of all point and line labels is S_{PL}.

$$S_{PL} = \sum\nolimits_{i=1}^{Q_1+Q_2} P_{i2} \tag{11}$$

The sum of label-feature conflict scores of all points, lines, and areas label is S_4.

$$\boldsymbol{S_4 = S_{PL} + S_A} \tag{12}$$

4) LABEL CONFLICT FACTORS

Label conflict indicates the conflict between two or more labels on the map, which significantly decreases the readability, clearness, and harmony of the map. Therefore, the number of label conflicts is the main criteria to evaluate the label conflict factor. In this paper, the Scoring Rule for the label conflict factor is defined as follows.

$$T_{i6} = \begin{cases} 0, & C_{i1} = 0 \\ 9 * C_{i1}, & C_{i1} \neq 0 \end{cases} \tag{13}$$

$$S_5 = \sum\nolimits_{i=1}^{Q} T_{i6} \tag{14}$$

where C_{i1} represents the number of overlaps between the *i-th* label minimum bounding rectangle (MBR) box and other label MBR boxes. Where T_{i6} represents the score of label conflict factor for the *i-th* label, and S_5 represent the score of label conflict factor of all features on the map.

5) A COMPREHENSIVE QUALITY MODEL

In this study, four quality factors were set to evaluate the effect of label placement and fully considered the label placement rules which are no ambiguity, legibility, a clear indication of the features, and no overlapping. Considering the mentioned factors can comprehensively reflect the advantages and disadvantages of automatic label placement results. To compare the quality of each generated position, it is necessary to evaluate them based on the given score value, and we refer to the corresponding Scoring Rules developed by Lu *et al.* [2]. Since label conflict and label-feature conflict are more important on automatic label placement, the weights of the two factors were deliberately increased. Also, the label position priority factor for the area feature is added in the Scoring Rule. The final quality evaluation model consists of five quality factors.

According to the weighted sum of five factors in the label quality evaluation model, the effect of label placement is evaluated based on Eq. (15), which is the final definition of the quality evaluation model in the 2-DOF space.

$$S = \sum\nolimits_{i=1}^{5} S_i * W_i \tag{15}$$

where S_i represents the score value of each quality factor, W_i represents the weight of each corresponding factor, and S represents the score of the quality evaluation model for the cartographic label placement.

In equation (15), the range value of the weight of each influencing factor is [0, 1], and the sum of the weight of all five factors is 1. The smaller the value of S is, the better

Algorithm 2 Pseudo Code of DDEGA Algorithm

```
Input features (F)
Set the buffer distance (b)
for i=0 ... F THEN
 if T = Point THEN // feature type (T)
     Generate the buffer circle around T based on value of (b)
     Generate 8 candidate positions based on the buffer circle (Q)
     OUTPUT (Q)
 elif T = Line THEN
      Generate the buffer line around T based on value (b)
      Generate 8 candidate positions based on the buffer line (Q)
      OUTPUT (Q)
 else THEN
      Generate the approximate skeleton line (L0) of area feature
      Generate 8 candidate positions based on the approximate skeleton line (Q)
      OUTPUT (Q)
Initialize the population (P) from generated positions (F*8)
Set mutation and crossover probability, and iteration number (iter)
Calculate the score value of P (S): including the score of label conflict (S1),
label-feature conflict(S2), label non-ambiguity(S3) and label priority(S4)
S= S1+ S2+ S3+ S4
Sort S value
i = 0
while (i<iter) THEN
      Select the top 50% individual (T) of P
      j=0
      while (j <0.2*P) THEN
        Randomly initialize two individuals from T and 1 new individual by GA
        j + = 1
      END while
      while (j<P) THEN
        Randomly initialize two individuals from T and 1 new individual by DDE
        j + =1
      END while
      Apply mutation on newly generated P to obtain P0
      P=P0
      Calculate the score value of P(S): including the score of label conflict (S1), label-feature conflict(S2), label
       non-ambiguity(S3) and label priority(S4)
       S = S1 + S2 + S3 + S4
      Select individual with the lowest score value
      i+ =1
OUTPUT (individual with lowest score value)
END while
```

the feature label placement result is based on label placement rules. In this current quality evaluation model, the label conflict factor and the label-feature conflict factor have the greatest weights due to the importance of the two quality factors on automatic label placement.

E. DDEGA IMPROVEMENT FOR N * M MULTIPLE CANDIDATE POSITIONS: DDEGA-NM

The second issue that needs to be considered is to choose an appropriate optimization algorithm to find the optimal candidate positions from large-scale solutions with reasonable execution time. Lu *et al.* [2] proposed the DDEGA algorithm and achieved better results in cross-layer feature label placement. DDEGA algorithm is a hybrid of discrete differential evolution algorithm [36] and genetic algorithm [37]. The advantages and disadvantages of the two algorithms are complementary. However, the algorithm has some major drawbacks including less candidate position selection, longer execution time, and label overlap of the small-area feature is not considered. Therefore, in this study, the procedure

Algorithm 3 Pseudo Code of Optimization of DDEGA-NM Algorithm

```
Form the initial population (P) from generated N*M positions of each feature
Define S, S0, S1, S2, S3, S4, and S5
Calculate the score of label ambiguity(S1), label position-priority(S2), label area-priority(S3) and label-feature conflict(S4)
  for N * M candidate positions of each feature, S0 = S1 + S2 + S3 + S4
Set mutation and crossover probability, and iteration number (iter)
Calculate the score value of P (S) and the score of label conflict (S5); S = S0 + S5
Sort S value
i = 0
while (i<iter) THEN
  Select the top 50% individual (T) of P
  j=0
  while (j < 0.3*P) THEN
      Randomly initialize two individuals from T and 1 new individual by GA
      j + = 1
  END while
  while (j<P) THEN
      Randomly initialize two individuals from T and 1 new individual by DDE
      j + = 1
  END while
  Apply mutation on newly generated P to obtain P0
  P=P0
  Calculate the score of label conflict (S5)
  Calculate the score value of P(S), S= S0+ S5
  Select individual with the lowest score value
  i+ =1
OUTPUT (individual with lowest score value)
END while
```

of candidate positions generation, and the algorithm optimization based on the DDEGA algorithm are improved, and the new algorithm is called the DDEGA-NM algorithm. Algorithm.2 is the pseudo code of the DDEGA algorithm.

1) CANDIDATE POSITIONS FROM 1-DOF N TO 2-DOF N * M

The DDEGA algorithm is equivalent to be a special form of the DDEGA-NM algorithm with $N = 8$ and $M = 1$, only selects 8 candidate positions in 1-DOF space which is the reference position, without considering that the label can be selected along another degree of freedom space. Here, the degree of freedom space is expanded from 1-DOF to 2-DOF space in the DDEGA-NM algorithm, and then $N * M$ candidate positions are generated along the two degrees of freedom space. The number of candidate positions of each feature is increased, which also expands the number of permutations and combinations between different label candidate positions. With the increment of candidate positions, fewer labels will be placed in overlap and conflicts theoretically. Therefore, the score value of the quality evaluation model can be reduced, which improves the quality of label placement.

2) OPTIMIZATION OF DDEGA-NM ALGORITHM

First, the problem of label placement for the small-area feature is solved in the DDEGA-NM algorithm. To determine whether an area feature is large enough or not, first, all the candidate positions are generated inside the area feature and then checks whether all the candidate positions overlap the area feature itself or not; if yes, the area feature does not have enough interspace for its label. In this case, the candidate positions are generated on the buffer curves, outside the area feature. If the generated positions do not overlap with the corresponding feature, then the label is placed on the buffer curves. If all generated positions outside the area feature have conflicts with its feature, the positions inside the area feature have higher priority even if the positions have overlap.

Second, by adding the priority factor for the label of area feature, the quality evaluation model of the DDEGA-NM algorithm is improved. It ensures that the label of the area feature is placed at the center of the feature as much as possible. Furthermore, the weights of label-feature conflict and label conflict are set larger than other weights in the evaluation model to highlight the importance of the two quality factors. Thus, the label-feature conflict and label conflict can be avoided as far as possible.

The DDEGA-NM optimization includes three procedures, which are the selection of candidate position, the quality evaluation of the label, and the sequential iteration. Compared to the original DDEGA algorithm, the new algorithm improved label placement quality and reduced the execution time. We ensure that under the premise of no overlap, the label

TABLE 1. Theoretical limit values corresponding to different N and M values.

N\M	1	2	3	4	5	6	7	8	9	10
8	7.109	1.857	1.770	1.211	1.474	1.147	1.138	1.172	1.151	1.134
16	2.677	1.082	1.029	0.978	0.987	0.952	0.944	0.962	0.955	0.950
24	2.115	1.059	0.783	0.977	0.743	0.958	0.736	0.721	0.710	0.721

with higher priority can be selected during the iteration cycle. Since the execution time of the DDEGA-NM algorithm is decreased, increasing the number of iterations can improve the comparability and optimize the algorithm results.

The probability of variation, hybridization probability, and genetic variation of the DDEGA-NM algorithm was set to 0.5, 0.8, and 0.1, respectively.

IV. EXPERIMENTS AND RESULTS

Three datasets were selected to verify the performance of the DDEGA algorithm for solving the problem of MGFLP, Lu *et al.* [2]. To compare the advantages and disadvantages of the two algorithms, a real dataset was selected for the experiments. The scale of the selected data was set to 1:4000000, and it is the Washington State map, which includes administrative areas, highways, downtown, and other layers. There are 71 features on the map, including 15 points, 17 lines, and 39 area features. The default height of a text is 11288.9114 at this scale. The DDEGA-NM algorithm was implemented in a machine with Intel® Core^{TM} i5-4210M CPU @ 2.60GHz 2.8GHz, running in Windows 10 professional x64 with 4.00 GB RAM installed. The code is written in python language, used Pycharm framework based on Arcpy template using ArcGIS 10.6.1.

The candidate positions were generated starting from a fixed distance of 4000m which accounts for about 0.354 of the text height, from the feature in the DDEGA algorithm. The minimum buffer distance in the DDEGA-NM algorithm is also set to 4000m, and the maximum buffer distance is three times the minimum buffer distance (about 1.063 of the height of a text). Therefore, the candidate positions generate between the minimum and the maximum buffer distance in 2-DOF space.

To avoid label conflict, the weight of this quality factor should be greater than the weight of the label-feature conflict factor. So the label conflict weight was set to the largest value in the label quality evaluation model of the DDEGA-NM algorithm. Also, to ensure the label of the area feature can be placed at the center of the feature, the label position priority factor is added in the quality evaluation model. In the quality evaluation model of the DDEGA-NM algorithm, the weights were as follows.

- Weight of label conflict ($W_{lc} = 0.5$)
- Weight of label-feature conflict ($W_{fc} = 0.3$)
- Weight of area label priority ($W_{ap} = 0.1$)
- Weight of point label priority ($W_{pp} = 0.05$)
- Weight of line label ambiguity ($W_a = 0.05$)

In this paper, to assess the result of the same evaluation model for the labeling results, we use the score value and the number of label-feature conflicts and labels conflicts as the two evaluation indices and comprehensively evaluated the label effect according to the two indices. For the labeling results of different evaluation models, the number of label-feature conflicts and label conflicts is used as evaluation indices.

A. CALCULATION AND COMPARISON OF THEORETICAL LIMIT VALUES OF DIFFERENT N AND M VALUES BASED ON 2-DOF SPACE

The ideal purpose of feature label placement is that each label should be placed at the ideal position without any conflicts. For this purpose, the quality evaluation model only needs to consider the label-feature conflict, label position priority, and ambiguity factor to label the features on the map. The lowest score value of each candidate position while considering the aforementioned factors is called the theoretical optimal permutation and combination, and the sum of the scores for a set of feature labels is called the theoretical limit value. The smaller the theoretical limit value is, the better the optimal permutation and combination is. The permutation and combination of candidate positions are increasing with the increment of N * M, which helps to reduce the possibility of overlap. To investigate the influence of different values on the theoretical limit value of multiple candidate positions, 8, 16, and 24 were selected for N, and 1 to 10 was chosen for M. The results are presented in Table 1.

The theoretical limit value of the quality evaluation model demonstrates a decreasing trend from 7.109 to 0.721, as presented in Table 1. However, as the values of N and M increased, the theoretical limit value also increased in some cases. For instance, when N = 8 and the value of M was changed from 4 to 5, the theoretical limit value increased from 1.211 to 1.474. Because the candidate positions are selected evenly within a certain distance, and the label might not overlap the feature when the distance is split into four parts. However, if the distance is split into five parts, the label overlapped with certain features. As a result, the value of the corresponding quality evaluation model decreased. By increasing the value of N and M, more candidate positions can be generated in the 2-DOF space. However, the 2-DOF space with a certain buffer distance is limited. When N and M reach a certain value, the candidate position can almost cover the entire 2-DOF space. In this case, increasing the value of N and M is not meaningless. The results of different N and

TABLE 2. The Score value of three algorithms running with10000 iterations (N = 8, M = 3).

Times	1	2	3	4	5	6	7	8	9	10	Mean
DDEG-NM	8.854	9.016	7.291	9.206	10.108	11.142	9.637	9.637	8.031	13.764	9.669
DDE-NM	40.052	31.964	30.497	30.232	33.907	33.961	33.112	39.256	36.212	35.667	34.486
GA-NM	10.098	13.635	11.200	12.087	12.084	14.997	13.195	14.045	9.777	12.478	12.360

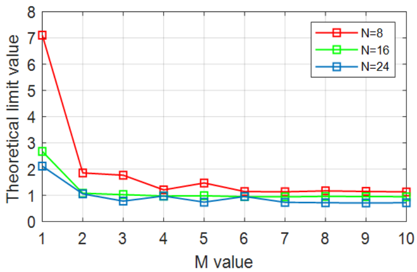


FIGURE 5. Changes of theoretical limit values.

M values obtained by the DDEGA-NM algorithm are shown in Fig. 5.

As shown in Fig. 5, the inflection point of the theoretical limit value is between M = 2 and M = 4 when N is 8, and at M = 2 when N is 16 and 24. The changes of theoretical limit values were not obvious when the value of M was set greater than 4 based on the experiments. This indicates that continuing to divide the 2-DOF space has an unobvious effect on improving the automatic label placement. The value of 3 was also taken as another parameter in two degrees of freedom space as the value of M. Therefore, based on the implemented experiments, the value of (N * M) 8 * 3 was selected as the preferred number of candidate positions for further analysis. The changes of the theoretical limit value of N equal to 16 and 24 with the value of M equal to 2 were also compared, the changes were not so obvious. Hence, the value of (N * M) 16 * 2 was selected as the second option to generate candidate positions for the current dataset. The DDEGA algorithm proposed by Lu *et al.* [2] is based on 8 candidate positions. To compare the obtained results with the DDEGA algorithm, N and M were also selected equal to 8 and 1, respectively, as a parameter in this experiment.

B. COMPARISON OF DIFFERENT ALGORITHMS

The DDEGA algorithm is the hybrid of GA and DDE algorithm, which is proven that the algorithm has a significant performance to solve the problem of automatic label placement and has better results than the single use of GA or DDE algorithm [2]. Hence, the DDEGA-NM (8 * 3) is improved based on the DDEGA algorithm, which changes from 1-DOF space to 2-DOF space. To study the efficiency of the DDEGA-NM algorithm with N = 8 and M = 3, the obtained results were compared with the results of DDE-NM (8 * 3) and the GA-NM (8 * 3) algorithms. Each algorithm ran 10 times, and the number of iterations was set to 10000. The smallest score values among 10 times of running were selected for comparison, as presented in Table 2.

As presented in Table 2, the average score of the DDEGA-NM algorithm, DDE-NM algorithm, and GA-NM algorithm is 9.669, 34.486, and 12.360, respectively, and the three algorithms are relatively stable. The score value of the DDEGA-NM algorithm is smaller than the two others. During 10 times of running, the maximum and minimum score values are 13.764 and 7.291, respectively, and the overall variance is not obvious. Furthermore, the score values of 10 runs, and the average score value of each algorithm, were taken as comparison parameters, as shown in Fig. 6(c-d) and Fig. 6(d).

Fig. 6 (a-c) shows the score values of each algorithm, and Fig. 6 (d) the average score value of the three algorithms with (N = 8, M = 3) when the same experimental dataset was used. The graphs show that the GA-NM algorithm has a faster convergence speed in the preliminary stage of the optimization, but the score value does not decrease in the middle and later stages. Although the score value of the DDE-NM algorithm has been converging and decreasing, the rate of decreasing is far slower than the two others. However, the convergence speed of the DDEGA-NM algorithm is fast and the value of the DDEGA-NM algorithm continues to converge. This confirms the superiority of the DDEGA-NM algorithm over both the GA-NM and the DDE-NM algorithms in the 2-DOF space.

C. COMPARISON OF DDEGA-NM WITH DDEGA AND MAPLEX

The DDEGA algorithm is a special form of the DDEGA-NM algorithm when N = 8 and M = 1, but the two algorithms are not equivalent because of using different models for candidate position generation and quality evaluation. To verify the practicality of the proposed method, we not only compared the obtained result with DDEGA, DDE, and GA results but also compared it with the optimal result of Maplex Label Engine. The Maplex Label Engine is a smart labeling module provided by ArcGIS Development that provides advanced label placement and conflict detection methods to help users improve the quality of labeling on the map.

The number of iterations was set to 300 in the DDEGA algorithm [2], which is expressed as the DDEGA-LU algorithm in this study. The value of 8 and 1 was set for N and M, respectively, in the DDEGA-NM algorithm, which is equivalent to the DDEGA algorithm. The algorithm ran 10 times with 10000 iterations. After the experiments were

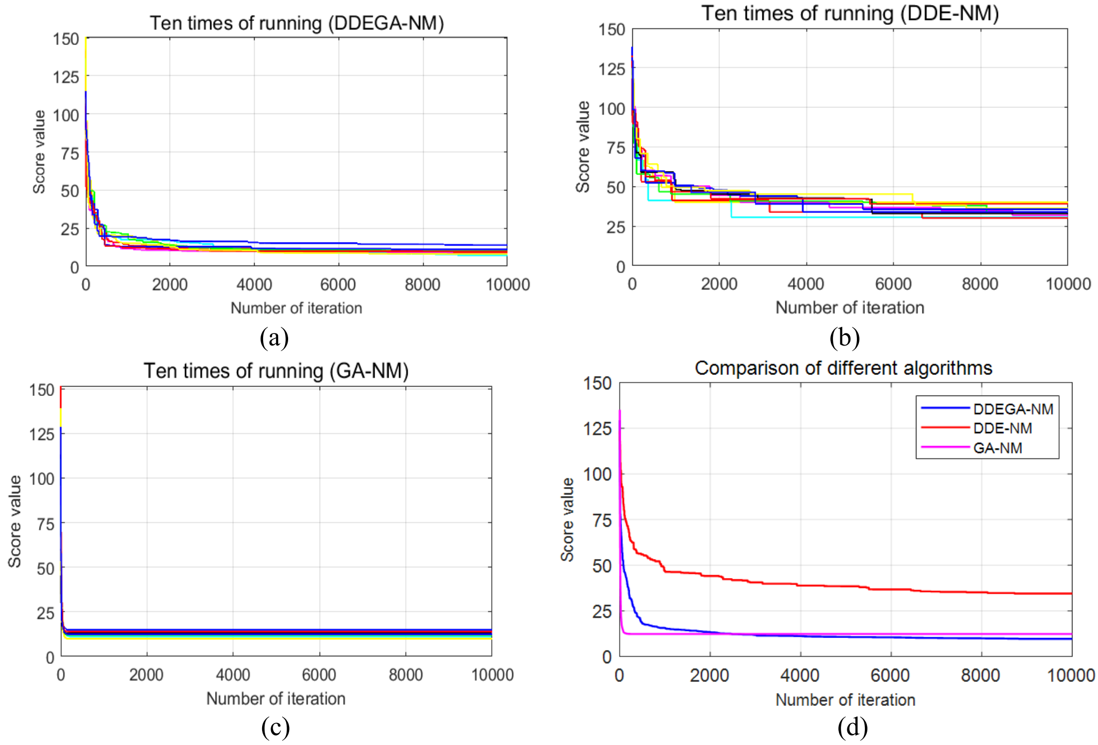


FIGURE 6. (a-d) Comparison of score values of the three algorithms (N = 8, M = 3).

run 10 times, the result with the smallest score value was selected for analysis. The results of 300 and 10000 iterations of the experiments were used as comparison parameters. The DDEGA-NM algorithm with the parameters of (N = 8, M = 3) was also used with 300 and 10000 iterations, which ran 10 times as well, and the lowest score values were chosen for comparison. Fig. 7(a) shows the result of the DDEGA-LU algorithm with 300 iterations, Fig. 7(b-c) shows the results of DDEGA-NM (8 * 1) with 300 and 10000 iterations, and the results of 300 and 10000 iterations of DDEGA-NM (8 * 3) are shown in Fig. 7(d-e). Fig. 7(f) shows the result of Maplex.

The below figures present that the DDEGA-NM algorithm is better than the original DDEGA-LU algorithm and Maplex, and the number of label conflicts and label-feature conflicts is reduced too. For comparing more clearly, the results were analyzed based on automatic label placement quality, the effect of different N * M values in the DDEGA-NM algorithm, and the number of label-feature conflicts. The obtained results are shown in Fig. 7, and the statistical results are presented in Table 3.

Table 3 presents that the obtained results by the DDEGA-NM algorithm, compared with the DDEGA-LU - algorithm and Maplex, are significantly improved. Also, the table illustrates that the result of the DDEGA-NM (8 * 3) algorithm is not as efficient as the DDEGA-NM (8 * 1) algorithm when the loop iteration is 300. However, in 10000 iterations, the label placement quality is better than achieved with the DDEGA (8 * 1) algorithm. This certifies that expanding the degrees of freedom space for feature label placement is an appropriate approach to reduce label conflicts and label-feature conflicts of multiple geographical features. In addition, it also confirms that the DDEGA-NM algorithm had better performance when the value of N = 8 and M = 3 was used.

D. COMPARISON OF DIFFERENT BUFFER DISTANCES OF DDEGA-NM ALGORITHM BASED ON 2-DOF SPACE

As shown in Table 3, the results generated by the DDEGA-NM algorithm comply with the label placement guidance. The minimum buffer distance is equal to 4000m and the maximum buffer distance is three times 4000m in the DDEGA-NM algorithm. This caused some labels to be placed far away from the features and leads to the ambiguity of the labels. It is important to find the appropriate minimum and maximum buffer distances in the 2-DOF space to reduce ambiguity. If the maximum buffer distance between the label and its feature exceeds the height of the label, the label of another feature may insert into this gap. Thus, the maximum buffer distance should preferably not exceed the height of a label. Furthermore, the minimum buffer distance cannot be equal to zero. Otherwise, the label with its corresponding feature will overlap. Based on these two facts, the two distances were defined as greater than zero and less than the height of the label in 2-DOF space.

To determine suitable values for minimum and maximum buffer distances in the DDEGA-NM algorithm, the reference

(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 7. The map of Washington State with labels generated: (a) DDEGA-LU [2]; (b) DDEGA-NM-300 iterations (N = 8, M = 1); (c) DDEGA-NM-10000 iterations (N = 8, M = 1); (d) DDEGA-NM-300 iterations (N = 8, M = 3); (e) DDEGA-NM-10000 iterations (N = 8, M = 3); (f) Maplex.

position and the height of the text were equally split into 8 segments. The minimum buffer distance was set to 1/8, and the maximum buffer distance was set to 2/8, 3/8, 4/8, 5/8, 6/8, 7/8 of a text height in turn. Under different N and M values, the theoretical limit values with different maximum buffer distances were calculated, as presented in Table 4.

Table 4 presents that as N and M continue to increase under the same maximum buffer distance, the theoretical limit value shows a decreasing trend. The statistical results comply with the theoretical limit values that are shown in Fig.5. To compare the impact of different maximum buffer distances in 2-DOF space, the DDEGA-NM (8 * 3) was taken as the

TABLE 3. Remarks on the statistics of conflicts.

Algorithm	(N*M) Candidate	Iterations	The number of label-feature conflicts					Label conflict	Score	Time (h)
			point	line	area	Multiple-features	Total			
DDEGA-LU	8	300	0	8	16	9	33	0	//	60.58
DDEGA-NM	(8*1)	300	0	8	13	4	25	0	12.159	1.98
DDEGA-NM	(8*3)	300	0	16	17	3	36	1	27.510	2.91
DDEGA-NM	(8*1)	10000	0	8	12	3	23	0	9.154	5.24
DDEGA-NM	(8*3)	10000	0	9	12	1	22	0	7.291	43.8
Maplex	//	//	0	16	10	11	37	0	//	//

TABLE 4. Theoretical limit values under different maximum buffer distances.

Maximum buffer	N\M	1	2	3	4	5	6	7	8	9	10
2/8	8	7.766	3.124	2.843	2.831	2.804	2.786	2.788	2.776	2.767	2.790
2/8	16	3.354	2.633	2.596	2.579	2.568	2.561	2.556	2.552	2.549	2.546
2/8	24	2.758	2.328	2.289	2.269	2.258	2.251	2.245	2.241	2.238	2.235
3/8	8	7.766	2.509	2.155	2.176	2.124	2.138	2.121	2.133	2.109	2.119
3/8	16	3.354	2.034	1.978	1.997	1.972	1.988	1.970	1.982	1.969	1.978
3/8	24	2.758	1.676	1.636	1.617	1.605	1.597	1.592	1.587	1.584	1.582
4/8	8	7.766	1.868	1.539	1.508	1.480	1.445	1.420	1.439	1.420	1.434
4/8	16	3.354	1.403	1.339	1.364	1.335	1.317	1.305	1.319	1.310	1.322
4/8	24	2.758	1.341	1.301	1.280	1.268	1.260	1.255	1.250	1.247	1.244
5/8	8	7.766	1.821	1.486	1.537	1.466	1.462	1.436	1.410	1.392	1.378
5/8	16	3.354	1.381	1.317	1.336	1.312	1.292	1.278	1.294	1.284	1.276
5/8	24	2.758	1.318	1.274	1.256	1.244	1.236	1.230	1.226	1.223	1.219
6/8	8	7.766	1.850	1.510	1.506	1.433	1.460	1.441	1.403	1.387	1.375
6/8	16	3.354	1.409	1.334	1.313	1.272	1.295	1.280	1.269	1.263	1.276
6/8	24	2.758	1.345	1.290	1.269	1.244	1.236	1.231	1.226	1.223	1.220
7/8	8	7.766	1.542	1.425	1.174	1.158	1.150	1.097	1.113	1.092	1.066
7/8	16	3.354	1.105	1.034	0.989	0.983	1.006	0.982	0.971	0.962	0.955
7/8	24	2.758	1.021	0.739	0.955	0.722	0.936	0.714	0.927	0.710	0.920

basic parameter, and the obtained theoretical limit values under different buffer distances are shown in Fig. 8.

As the maximum buffer distance reaches 4/8 of a text height, the theoretical limit value became stable, as shown in Fig. 8. Although increasing the maximum buffer distance can reduce the theoretical limit value, but the changes are not obvious, and even they can be ignored. Because the farther the label position is from the corresponding feature, the greater the ambiguity is. Therefore, the value of 4/8 of the text height is the most appropriate value for the maximum buffer distance

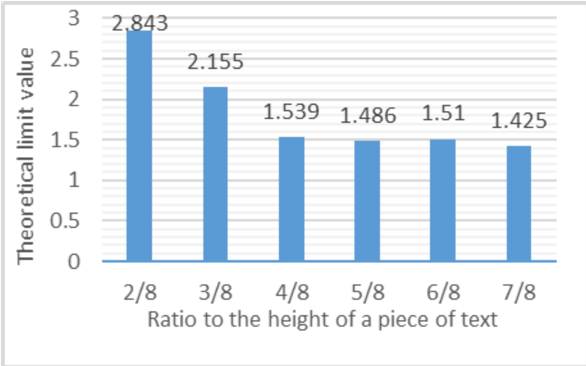


FIGURE 8. Theoretical limit values of different maximum buffer distances (N = 8, M = 3).

of the current dataset. The algorithm ran 10 times with this buffer distance, and each time iterated 10000 iterations. After 10 times of running, the result with the smallest score value was selected for comparison, as shown in Fig. 9.

The labels in Fig. 9 are closer to their corresponding features and have less ambiguity than the labels where the maximum buffer distance was three times 4000m, which is shown in Fig. 7(e). The number of label-feature conflicts and label conflicts in Fig. 9 were counted and compared with the result of the DDEGA-NM algorithm that the buffer distance was three times 4000m. The statistical results are presented in Table 5.

As can be seen in Table 5, after the values of minimum and maximum buffer distances were changed from (0.354-1.063) to (1/8-4/8) of a text height, the quality of map label placement and map visualization is improved. Additionally, Fig. 7(e) and Fig. 9 were overlaid to compare the differences of the two results with different maximum buffer distances on automatic label placement. As shown in Fig. 10(a), when the minimum and maximum buffer distances were set to 0.354 and 1.063, respectively, the labels are shown in gray, but with thevalue of 1/8 and 4/8, the labels are shown in black.

When the minimum and maximum buffer distances were set to 1/8 and 4/8, respectively, most of the labels were positioned closer to their corresponding features, as shown in Fig. 10(b-c). The area labels ''San Juan'' and ''island'' and the point labels ''Port Angeles'' and ''Bellingham'' within area A, as well as the line label ''U97 (1'', ''U97 (2'' and the point label ''Wenatchee'' within area B illustrate this clearly. In summary, the ambiguity of the label can be reduced by selecting proper values for the minimum and maximum buffer distances in 2-DOF space.

FIGURE 9. DDEGA (8 * 3) −10000 iterations (changing the minimum and maximum buffer distance).

*E. COMPARISON OF DDEGA-NM (8 * 3) AND DDEGA-NM (16 * 2)*

As discussed in the previous section, the inflection point of the theoretical limit value of the DDEGA-NM algorithm appears at 8 * 3 and 16 * 2. To determine an appropriate minimum and maximum buffer distance for the value of $N = 16$ and $M = 2$ in 2-DOF space, we set the values of 1/8 and 4/8 of a text height, respectively, for the algorithm, which ran 10 times. Each time the algorithm iterated 10000 times, and the lowest score value was selected for comparison with the result of DDEGA-NM (8 * 3). The statistical results are presented in Table 6, which shows that the value of (8 * 3) is the most appropriate for our dataset. As Table 6 presents, after 10 runs, there is no significant difference between the final score values of the two algorithms and they are relatively stable. However, because of increasing the number of candidate positions from 24 to 32 candidate positions for each feature, the DDEGA-NM (16 * 2) algorithm requires longer computational times. Therefore, the performance of DDEGA-NM (8 * 3) is better than DDEGA-NM (16 * 2) algorithm in terms of running time.

F. COMPARISON OF DIFFERENT WEIGHT VALUES ON LABEL PLACEMENT QUALITY

The weights of all quality factors were set based on the experiments in this study. However, here only the weights of label conflict and label-feature conflict are discussed due to the importance of these two factors. The label conflict and label-feature conflict have the greatest effect on label placement quality. Therefore, different weight values for these two factors were compared and discussed to determine suitable

TABLE 5. Comparison of the results before and after the DDEGA-NM changed the buffer distance.

Minimum	Maximum	The number of label-feature conflicts					Label-conflict	Score
		point	line	area	Multiple-features	Total		
0.354	1.063	0	8	13	1	22	0	7.291
1/8	4/8	0	4	16	1	21	0	6.129

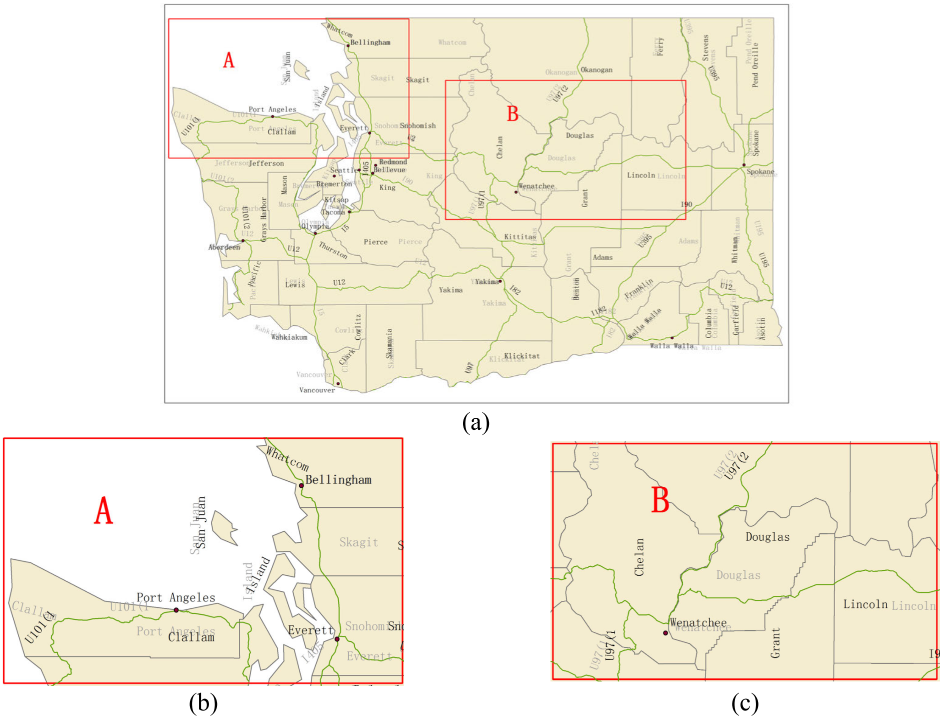


FIGURE 10. (a) Labels of different minimum and maximum buffer distances; (b) area A; (c) area B.

weight values. However, the values of the other three factors were kept unchanged.

For studying different weight values, the two quality factors were divided into five groups, with the range value of [0.2, 0.6]. The minimum and maximum buffer distances were set to 1/8 and 4/8 of a text height, respectively. Each group of weight experiments ran 10 times with 10000 iterations. The number of label conflicts and label-feature conflicts of 5 groups of experiments with different weights was counted. The results are presented in Table 7.

Table 7 presents that by increasing the weight of label conflict and decreasing the weight of the label-feature conflict, the number of label conflicts decreases, and the number of label-feature conflicts increases, and vice versa. The value of 0.4 was the optimal weight value for the two quality factors, with fewer label conflicts and label-feature conflicts based on the experiments.

V. EXPERIMENTAL ANALYSIS

A. THE IMPACT OF ADDING MORE CANDIDATE POSITIONS ON THE THEORETICAL LIMIT VALUE BASED ON 2-DOF SPACE

In the traditional model of multiple candidate positions (MCP) scheme, each point feature adopts 8 candidate positions. In this case, all 8 positions may overlap the features, or few positions are free of overlap. Line and area features rarely use the concept of MCP scheme. When the number of candidate positions is not adequate for the feature label, the probability of label-feature conflict and label conflict increase on the map. Therefore, expanding the degrees of freedom space for feature label placement increases the number of label candidate positions, and it reduces the possibility of label-feature conflicts and label conflicts on automatic label placement.

To evaluate the generated positions in the 2-DOF space, a comprehensive quality evaluation model is defined that assigns a score value for each solution which is called the theoretical limit value. Without considering label conflict, the theoretical limit value is the sum of the minimum scores of a set of labels positions that considering label-feature conflict, label position priority, and ambiguity factor, which can be used to evaluate the result of label placement. Therefore, when point, line, and area features are located far enough that their labels do not have the possibility of conflicts, the theoretical limit value is the only determinant to evaluate the optimal solution in such cases, as can be seen in Table 1 and Fig. 5. When the value of M was relatively small, and by increasing the value of N, the theoretical limit value

TABLE 6. Comparison between DDEGA-NM (8 * 3) and DDEGA-NM (16* 2).

(N*M)	1	2	3	4	5	6	7	8	9	10	Mean
(8*3)	6.129	8.043	7.260	9.696	7.946	8.590	8.025	7.342	7.814	7.498	7.834
(16*2)	7.676	8.040	9.428	9.828	8.890	7.126	9.751	8.534	8.899	9.178	8.735

TABLE 7. Comparison of the label and label-feature conflict with different weights.

Group	Weight		Number of label-conflicts			Number of label-feature conflicts			Mean score
	Label conflict	Label-feature conflict	Min	Max	Mean	Min	Max	Mean	
1	0.2	0.6	18	25	22.1	0	1	0.2	13.336
2	0.3	0.5	17	25	22.3	0	1	0.1	12.779
3	0.4	0.4	19	25	22.3	0	0	0	9.871
4	0.5	0.3	21	27	23.4	0	0	0	8.242
5	0.6	0.2	21	29	25.2	0	0	0	6.893

decreased. For example, when M = 1 and N was increased, the theoretical limit value decreased rapidly, which indicates that it is effective to increase the reference position in the first degree of freedom space.

If N is small, increasing the value of M has an obvious effect on reducing the theoretical limit value. For example, when N = 8 and M was increased, the theoretical limit value decreased. It indicates that increasing the buffer position is beneficial in 2-DOF space. However, when M exceeds a certain value, the value of M has little significance on the reduction of the theoretical limit value, for instance, when M > 6, as shown in Fig. 5. This points out that the value of M should not be too large. Increasing the values of N and M helps to decrease the theoretical limit value in general, which is helpful to improve the quality of automatic label placement.

B. THE EFFECT OF THE NUMBER OF ITERATIONS ON THE APPROXIMATION OF THE ACTUAL RESULTS TO THE OPTIMAL SOLUTION

Each feature has N * M label candidate positions in the DDEGA-NM algorithm. When the number of features to be labeled on a map is F, the permutation and combination of the label can reach to $(N^*M)^F$. In this experimental dataset, F = 71, N = 8, and M = 3, thus the number of combinations between labels is 24^{71}. Therefore, an extensive optimization algorithm should be designed to obtain a satisfactory solution from large-scale solutions.

The DDEGA-NM optimization starts with 100 initial solutions and then sorts them according to their calculated score values. To select the best solution in each iteration, 50 percent of the best initial solutions are selected based on their lowest score values. Then they participate in the DDEGA-NM algorithm operation to generate another 99 new solutions for the next iteration. One out of the initial solutions only takes part in the next iteration without participating in the DDEGA-NM algorithm operation. This occurs based on the lowest score value and ensures that the best solution from the obtained result is not lost in the next iteration. There are 100 solutions in the next iteration as well. 99 of them are generated by the DDEGA-NM algorithm operation and one is from the previous iteration based on its lowest score value. The rest of the loop follows the same steps as the first iteration until reaching the last number of the iteration cycle.

By increasing the number of iterations, the quality of the label placement can be improved because the probability of getting a new solution is increasing with the increment of the iteration cycle, as shown in Fig. 6 (c). However, as the score value reaches a certain level, it will be difficult for the proposed algorithm to reduce the score value because of the possibility of finding a solution with a smaller score value than the obtained solution decreases, noting that the score value is the value of the quality evaluation model that evaluates each solution. It explains the reason why the score value of the algorithm gradually tends to be flat in the later stage in Fig. 6 (c). As can be seen in Table 2, the score value of 10000 iterations is 7.291 because, at this stage of iteration, there are fewer label permutations and combinations with a score smaller than 7.291 among the combination of candidate positions. Therefore, the probability of achieving the optimal solution from 24^{71} number of combinations is almost zero (about $10000 \times 99 \div 24^{71} \approx 1.00e - 10^{-90}$) in the DDEGA-NM algorithm with 10000 iterations, and only a satisfactory result can be obtained with this number of iterations.

C. THE EFFICIENCY OF THE ALGORITHM

Two phases are designed in the DDEGA-NM algorithm: generation of candidate positions, and the optimization phase. The optimization phase includes the selection of candidate positions, evaluation of label quality, and the iteration cycle. Comparing to the DDEGA algorithm, the speed of label placement and label position accuracy of the DDEGA-NM algorithm is significantly improved. In addition, the obtained results are compared with Maplex Label Engine results to verify the practical application of the proposed algorithm. The results have shown significant improvement compared to Maplex result, too.

The execution time of the DDEGA algorithm is 60.58 hours for 300 iterations. However, the execution time of the DDEGA-NM algorithm is 43.8 hours for 10000 iterations. When the number of candidate positions increased from 8 to 8 * 3, the complexity of the NP-hard problem increased as well. However, the achieved results are better than the DDEGA algorithm that used the traditional method of candidate position generation with 8 candidate positions. Therefore, the results prove that increasing the number of candidate positions and improving the efficiency of the algorithm is helpful to solve the problem of label placement which is proven to be an NP-hard problem.

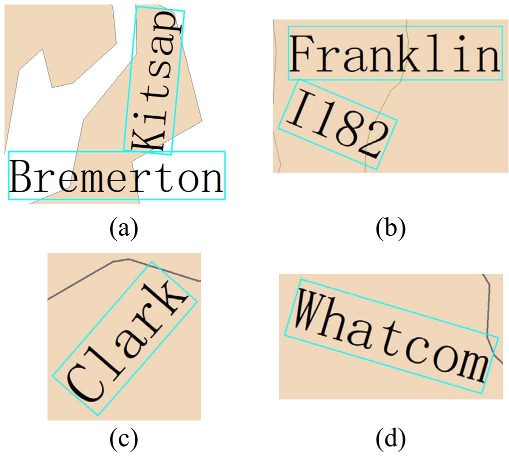


FIGURE 11. **(a-b) False label conflict; (c-d) false label-feature conflict.**

It should also be explicitly noted that the efficiency of the DDEGA-NM algorithm is limited by two factors which are the number of map features (F) and the number of candidate positions (N * M) for each feature. In this study, the value of N and M is 8 and 3, respectively, thus the total number of permutations and combinations is 24^F. If the number of features (F) increases on the map, the permutation and combination of the labels increase exponentially (24^F), and it increases the difficulty of obtaining the optimal result with a smaller number of iterations. Theoretically, the algorithm can be improved by increasing the degree of freedom space, which can get better results than using one degree of freedom space, and reduces the label-feature conflict and label conflict. However, by expanding the degree of freedom space for feature label placement, the time complexity of the label placement problem increases. Therefore, further optimization algorithms are needed to be studied to enhance the performance of the algorithm in terms of execution time, such as parallel processing which enables the algorithm to run in parallel by decomposing the search space in several sub-domains.

D. ANALYSIS OF QUALITY EVALUATION MODEL

The weights of label-feature conflict and label conflict factor were deliberately increased in the Scoring Rule, which reduced the possibility of label-feature conflict and label conflicts. It should be noted that all label conflicts are eliminated in our experimental result, and the number of label-feature conflicts is reduced compared with the previous study. In the Scoring Rules, a new quality factor is also added for the area features to ensure that their labels are placed at the center of the feature as much as possible. As shown in Fig. 9, most of the labels of area features are placed at the center of them, which enhances the beauty of map visualization.

Despite defining a comprehensive Scoring Rule in the DDEGA-NM algorithm, the quality evaluation model still has some drawbacks. In this study, the label-feature conflict and label conflict scores are evaluated based on the minimum bounding rectangle (MBR) box of the label. Sometimes the MBR box overlaps the feature or label, however, the letters of the labels do not overlap each other or with features, resulting in false label conflict and false label-feature conflict, as shown in Fig. 11 (a-d). Such conditions are scored as label conflict and label-feature conflict by the DDEGA-NM algorithm which is not conducive to find the optimal solution. In the future study, M such instances should be regarded as false label conflict and false label-feature conflict which can be assigned the score value of zero in the quality evaluation model. This will lead the algorithm to find a solution close to the optimal in a lower number of the iteration cycle.

VI. CONCLUSION AND FUTURE WORK

Due to the lack of the insufficient degree of freedom space and the uncertainty of multiple candidate positions (MCP), the degree of freedom space for feature label placement is expanded from one degree to two degrees of freedom (2-DOF) space. In the 2-DOF space, the methods of generating, calculating, evaluating, and selecting MCP for feature label placement are studied to determine a proper method of generating multiple candidate positions for multiple geographical features.

The reference position and the buffer position are defined as the degree of freedom space for feature label placement in 2-DOF space. After defining the degree of freedom space, the method and the process of generating multiple candidate positions for multiple geographical features in 2-DOF space are studied based on N equal division of reference position and M equal division of buffer position. To evaluate different candidate positions during the process of label placement, an extensive quality evaluation model is established based on multiple candidate positions in 2-DOF. The quality evaluation model includes label ambiguity, label position priority, label-feature conflict, and label conflict. We studied the theoretical limit value and the process of finding a suitable solution for feature label placement based on the DDEGA-NM algorithm. In addition, the problem of label placement for small-area features is solved in the DDEGA-NM algorithm. The labels of these area features are positioned outside the boundary of features, preventing the labels from being placed in conflict with their corresponding features.

The efficiency of the DDEGA-NM algorithm is verified by comparing the obtained results with the previous study that used 8 candidate positions in one degree of freedom space. The comparison analysis indicates that the DDEGA-NM algorithm achieved superior results. Though the complexity of the NP-hard problem increased in 2-DOF space, however, the labels are positioned more accurately, and the execution time of the proposed algorithm is reduced. In addition to the DDEGA algorithm, the obtained results are compared with Maplex Label Engine results, too. The result achieved by

Maplex has 37 label-feature conflicts, however, the obtained result of the proposed algorithm only has 22 label-feature conflicts. Therefore, the proposed method can effectively position the labels of multiple geographical features in a limited space around or inside features with respect to the automatic label placement guidance.

A satisfying result is achieved by the DDEGA-NM algorithm based on 2-DOF space. However, there are still some conflicts between features and labels due to the lack of enough space around the feature to generate candidate positions without feature conflict. In further studies, based on N * M candidate positions, the researchers can consider the third degree of freedom space for feature label placement that the label can be rotated to a certain angle, and select a label angle with the minimum number of overlap and conflict as the third degree for labels. The degree of freedom space for feature label placement expands from 2 degrees to 3 degrees of freedom space. Alternatively, the maximum buffer distance of 2-DOF space can be appropriately increased for those labels with overlap and conflict. Also, longer line features are only labeled in one segment, which may cause ambiguity when there are many features. Hence, longer line features should be labeled in multiple segments according to their length in further studies.

Since feature label placement is an NP-hard problem, by increasing the number of input features, the running time of the algorithm increases exponentially. Therefore, it is necessary to improve the efficiency of the algorithm considering the complexity of the problem in terms of the NP-hard. Moreover, the optimization algorithm can be designed in such a way that it starts the iteration cycle with a solution that has a lower score value. In addition, during the iteration cycle, it is also possible to intervene in the random selection of each iteration to obtain a result with a lower score value in every iteration of the algorithm.

REFERENCES

[1] M. Formann and F. Wagner, "A packing problem with applications to lettering of maps," in *Proc. 7th Annu. Symp. Comput. Geometry (SCG)*, 1991, pp. 281–288, doi: 10.1145/109648.109680.

[2] F. Y. Lu, J. Q. Deng, S. Y. Li, and H. Deng, "A hybrid of differential evolution and genetic algorithm for the multiple geographical feature label placement problem," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 5, p. 237, May 2019, doi: 10.3390/ijgi8050237.

[3] S. Edmondson, J. Christensen, J. Marks, and S. M. shieber, "A general cartographic labelling algorithm," *Cartographica: Int. J. Geographic Inf. Geovis.*, vol. 33, no. 4, pp. 13–24, Dec. 1996, doi: 10.3138/U3N2-6363-130N-H870.

[4] P. Yoeli, "The logic of automated map lettering," *Cartographic J.*, vol. 9, no. 2, pp. 99–108, 1972, doi: 10.1179/000870472787352505.

[5] X. X. Zhou, Z. H. Sun, C. B. Wu, and Y. Ding, "Automatic label placement of point feature: Using ant colony algorithm based on group clustering," *J. Geo-Inf. Sci.*, vol. 17, no. 8, pp. 902–908, Aug. 2015, doi: 10.3724/SP.J.1047.2015.00902.

[6] J. J. Qiao, F. W. Hu, and H. W. Zhang, "Point annotation labelling based on the model of detecting information," *Bull. Surv. Mapping*, no. 6, pp. 35–40, Jun. 2016, doi: 10.13474/j.cnki.11-2246.2016.0185.

[7] L. Li, H. Zhang, H. H. Zhu, and W. Hu, "A point-feature labeling algorithm based on movable regions," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 43, no. 8, pp. 1129–1137, Aug. 2018, doi: 10.13203/j.whugis20160289.

[8] Y. Lei, T. Ai, X. Zhang, and J. Li, "A parallel annotation placement method for dense point of interest labels using hexagonal grid," *Cartography Geographic Inf. Sci.*, vol. 48, no. 2, pp. 95–104, Mar. 2021, doi: 10.1080/15230406.2020.1833761.

[9] Y. Ding, N. Jiang, C. Wu, and X. Zhou, "A two-phase algorithm for point-feature cartographic label placement," *Earth Sci. Informat.*, vol. 11, no. 2, pp. 183–203, Jun. 2018, doi: 10.1007/s12145-017-0320-8.

[10] J. S. Doerschler and H. Freeman, "A rule-based system for dense-map name placement," *Commun. ACM*, vol. 35, no. 1, pp. 68–79, Jan. 1992, doi: 10.1145/129617.129620.

[11] M. Barrault, "An automated system for name placement which complies with cartographic quality criteria: The hydrographic network," in *Proc. Int. Conf. Spatial Inf. Theory*, vol. 12. Berlin, Germany: Springer, pp. 321–330, 1995, doi: 10.1007/3-540-63623-4_71.

[12] A. Wolff, L. Knipping, M. V. Kreveldz, and P. K. Agarwal, "A simple and efficient algorithm for high-quality line labelling," *Dept. Geography Univ. Southampton*, vol. 11, pp. 146–150, Apr. 2008, doi: 10.1.1.24.6970.

[13] S. Sun, H. Zhao, J. Fang, Z. Cheng, and Y. Zhao, "A pratical method for line labeling," in *Proc. 18th Int. Conf. Geoinform.*, Jun. 2010, pp. 18–20, doi: 10.1109/GEOINFORMATICS.2010.5567478.

[14] A. Gemsa, B. Niedermann, and M. Nöllenburg, "Label placement in road maps," in *Proc. Int. Conf. Algorithms Complex. (CIAC)*, May 2015, pp. 221–234, doi: 10.1007/978-3-319-18173-8_16.

[15] T. Lan, Z. Li, Q. Peng, and X. Gong, "Automated labeling of schematic maps by optimization with knowledge acquired from existing maps," *Trans. GIS*, vol. 24, no. 6, pp. 1722–1739, Aug. 2020, doi: 10.1111/tgis.12671.

[16] F. Hong, Z. X. Zhang, and D. S. Du, "The algorithm design and implementation of adding annotation to map for linear feature automatically," *Acta Geodaetica et Cartographic Sinica*, vol. 28, no. 1, pp. 88–91, Feb. 1999, doi: 10.3321/j.issn:1001-1595.1999.01.017.

[17] Y. Yin, C. M. Li, and S. T. Ding, "The algorithm and realization of linear feature automatic label placement," *Chin. Acad. Eng.*, vol. 15, no. 5, pp. 30–36, May 2013, doi: 10.3969/j.issn.1009-1742.2013.05.006.

[18] J. Ahn and H. Freeman, "A program for automatic name placement," in *Proc. 6th Int. Symp. Automated Cartography (Auto-Carto Six*, 1984, vol. 21, no. 2, pp. 101–109, doi: 10.3138/0646-Q262-6636-3681.

[19] L. R. Ebinger and A. M. Goulette, "Noninteractive automated names placement for the 1990 decennial census," *Cartography Geographic Inf. Syst.*, vol. 17, no. 1, pp. 69–78, Mar. 2013, doi: 10.1559/152304090784005877.

[20] Y. Li, M. Sakamoto, T. Shinohara, and T. Satoh, "Automatic label placement of area-features using deep learning," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLIII, pp. 117–122, Aug. 2020, doi: 10.5194/isprs-archives-XLIII-B4-2020-117-2020.

[21] M. Rylov and A. Reimer, "A practical algorithm for the external annotation of area features," *Cartographic J.*, vol. 54, no. 1, pp. 61–76, Jan. 2017, doi: 10.1179/1743277414Y.0000000091.

[22] S. P. Gomes, G. M. Ribeiro, and L. A. N. Lorena, "Dispersion for the point-feature cartographic label placement problem," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5878–5883, Nov. 2013, doi: 10.1016/j.eswa.2013.04.035.

[23] G. L. Cravo, G. M. Ribeiro, and L. A. N. Lorena, "A greedy randomized adaptive search procedure for the point-feature cartographic label placement," *Comput. Geosci.*, vol. 34, no. 4, pp. 373–386, Apr. 2008, doi: 10.1016/j.cageo.2007.01.007.

[24] Y. Chen, Z. Wang, and X. Liu, "Automated point feature label placement using backtracking algorithm with an adjacent graph," in *Proc. 18th Int. Conf. Geoinform.*, Jun. 2010, pp. 1–5, doi: 10.1109/GEOINFORMATICS.2010.5567901.

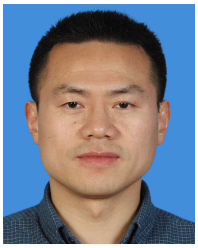
[25] S. Zoraster, "Practical results using simulated annealing for point feature label placement," *Cartography Geographic Inf. Syst.*, vol. 24, no. 4, pp. 228–238, Jan. 1997, doi: 10.1559/152304097782439259.

[26] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MLACO: A multi-label feature selection algorithm based on ant colony optimization," *Knowl.-Based Syst.*, vol. 192, Mar. 2020, Art. no. 105285, doi: 10.1016/j.knosys.2019.105285.

[27] M. Yamamoto, G. Camara, and L. A. N. Lorena, "Tabu search heuristic for point-feature cartographic label placement," *GeoInformatica*, vol. 6, no. 1, pp. 77–90, Mar. 2002, doi: 10.1023/A:1013720231747.

[28] Y. Yang, S. D. Deng, L. Li, and H. H. Zhu, "The research of intelligent point-feature cartographic label placement base on Tabu search algorithm," *Sci. Surv. Mapping*, vol. 32, no. 6, pp. 46–48, Nov. 2007, doi: 10.3771/j.issn.1009-2307.2007.06.014.

[29] K. G. Kakoulis and I. G. Tollis, "A unified approach to automatic label placement," *Int. J. Comput. Geometry Appl.*, vol. 13, no. 1, pp. 23–59, Feb. 2003, doi: 10.1142/S0218195903001062.
[30] J. Zhao, X. G. Luo, and R. Y. Zhang, "Novel label placement algorithm in digital map—Grid method," *Comput. Eng.*, vol. 34, no. 7, pp. 278–279, Apr. 2008, doi: 10.3969/j.issn.1000-3428.2008.07.098.
[31] Z. Q. Chen and X. Wu, "Study on automatic cartographic labeling based on multi-objective evolutionary algorithm," *Geomatics Spatial Inf. Technol.*, vol. 35, no. 4, pp. 217–220, Apr. 2012, doi: 10.3969/j.issn.1672-5867.2012.04.073.
[32] C. Wu, Y. Ding, X. Zhou, and G. Lu, "A grid algorithm suitable for line and area feature label placement," *Environ. Earth Sci.*, vol. 75, no. 20, pp. 1–11, Oct. 2016, doi: 10.1007/s12665-016-6190-4.
[33] E. Imhof, "Positioning names on maps," *Amer. Cartographer*, vol. 2, no. 2, pp. 128–144, Mar. 2013, doi: 10.1559/152304075784313304.
[34] H. Fan, D. S. Du, and Z. X. Zhang, "The study on the principles of automated placement of map name and its implementation approach," *J. Wuhan Tech. Univ. Surv. Mapping*, vol. 24, no. 2, pp. 3–5, Jun. 1999.
[35] S. V. Dijk, M. V. Kreveld, T. Strijk, and A. Wolff, "Towards an evaluation of quality for names placement methods," *Int. J. Geographical Inf. Sci.*, vol. 16, no. 7, pp. 641–661, Nov. 2002, doi: 10.1080/13658810210138742.
[36] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
[37] O. V. Verner, R. L. Wainwright, and D. A. Schoenefeld, "Placing text labels on maps and diagrams using genetic algorithms with masking," *INFORMS J. Comput.*, vol. 9, no. 3, pp. 266–275, Aug. 1997, doi: 10.1287/ijoc.9.3.266.



JIQIU DENG (Member, IEEE) received the Ph.D. degree in earth exploration and information technology from the Central South University, in 2006. He was qualified as a Senior Programmer in 1998. He is currently an Associate Professor and the Vice Dean of the Department of Geographic Information, Central South University. His current research interests include geoscience dig data and artificial intelligence, resources and environment information systems, network and mobile GIS, and geoscience 3-D modeling and visualization. He is a Senior Member of the China Computer Federation and a member of the Committee on Big Data and Digital Geosciences of the Chinese Society of Mineral, Petrological, and Geochemistry.

ZHIYONG GUO received the bachelor's degree in surveying and mapping engineering from Xiangtan University, in 2019. He is currently pursuing the master's degree in geological resources and geological engineering with Central South University. His current research interest includes geographic information systems.

MOHAMMAD NASER LESSANI received the bachelor's degree in surveying and mapping engineering from Kabul Polytechnic University, Kabul, Afghanistan, in 2019. He is currently pursuing the master's degree with Central South University, Changsha, China. His current research interests include spatial data analysis, big data analysis, and parallel computing.