Niche Method Complementing the Nearest-better Clustering

Yuhao Li
Graduate School of Design
Kyushu University
Fukuoka, Japan 815-8540
Email: li.yuhao.698@s.kyushu-u.ac.jp

Jun Yu
Faculty of Design
Kyushu University
JSPS Research Fellow
Fukuoka, Japan 815-8540
Email: yujun@kyudai.jp

Hideyuki Takagi
Faculty of Design
Kyushu University
Fukuoka, Japan 815-8540
Email: h.takagi.457 @ m.kyushu-u.ac.jp

Abstract—We propose a two-stage niching algorithm that separates local optima areas in the first stage and finds the optimum point of each area using any optimization technique in the second stage. The proposed first stage has complementary characteristics to the shortcoming of Nearest-better Clustering (NBC). We introduce a weighted gradient and distance-based clustering method (WGraD) and two methods for determining its weights to find out niches and overcome NBC. The WGraD creates spanning trees by connecting each search point to other suitable one decided by weighted gradient information and weighted distance information among search points. Since weights influence its clustering result, we propose two weight determination methods 1 and 2. The weight determination method 1 We combine these methods into WGrad, i.e. $WGraD_1$ and $WGraD_2$, and compare the characteristics of NBC, $WGraD_1$, and $WGraD_2$ using differential evolution (DE) as a baseline search algorithm for obtaining the optimum of each niche after clustering local areas. We design a controlled experiment and run (NBC + DE), ($WGraD_1$ + DE) and ($WGraD_2$ + DE) on 8 benchmark functions from CEC 2015 test suite for single objective multiniche optimization. The experimental results confirmed that the proposed strategy can overcome the shortcoming of NBC and be a complementary niche method of NBC.

Keywords—niche, local optima, clustering, evolutionary computation, nearest-better clustering

I. Introduction

Many real-world optimization problems are multimodal problem, and decision makers frequently want to compare multiple candidate search points to make the final decision rather than only one best search point. Truss-structure optimization design problem [1], second Toyota paradox [2], drug molecule design [3] and protein structure prediction [4] are good examples to show the importance of obtaining multiple candidate search points. When we can have multiple search points, we can replace one search point with others if the search point is inadequate. Most traditional optimization methods can only get one optimal search point in a single execution run except niche methods.

Local area segmentation methods, as a niching technique, can help traditional optimization methods to find out multiple optimal search points. Unlike population-based niche methods finding local optima, the objective of local area segmentation methods is to detect different local optima areas. It is based on the assumption that a fitness landscape consists of multiple peaks and valleys [5], and optimal search points are located on different peaks (or niches) [6]. It always firstly separates a search space into several non-overlapping tiny niches by clustering the search points on the same niche into one cluster. After then, traditional optimization methods searches the local optimum in each niche.

Nearest-better clustering (NBC) [7] and its improved version, NBC2 [8], are local area segmentation methods, and their niching algorithm (the nearest-better evolutionary algorithm) consisting of NBC and CMA-ES [9] won the CEC2013 niching competition. Although NBC has shown its effectiveness in splitting the search space in low-dimensional problems, for high-dimensional function, especially those where local optima locate far from each other, a few studies have indicated that NBC becomes hard to separate a whole search space [10], [11].

There are two main objectives of this paper. The first one is to propose local area segmentation methods for detecting niche areas which local optima are far from each other in highdimensional search spaces. The proposed method constructs multiple spanning trees that represent potential niches using information of a weighted gradient and a weighted distance. The second objective is to analyze the applicability of the proposed methods under controlled experimental conditions and introduce some topics for open discussion.

Following this introductory section, we briefly summarize some related works in the Section II. The detailed implementation of our proposed clustering method and two method for determining weights are proposed in the Section III. We evaluate our proposals using eight benchmark functions in the Section IV. Then, we discuss the effect of the proposed clustering method and analyze the effect of its components in the Section V. Finally, we conclude our works and point out some potential research directions in the Section VI. Throughout this paper, without loss of generality, we explain our methods using continuous maximization problems.

II. Related Works

One approach to solve multi-modal optimization problems is to increase population diversity to locate multiple search points, for example pre-selection [12], fitness sharing [13],

crowding [14] and species [15], etc. [16], [17]. RS-CMSA-ES [18], the winner of the CEC2016 niching competition, maintains several subpopulations to explore a search space in parallel. It marks the previously identified local peaks as taboo points and repels the subpopulation to prevent convergence to the same peak. Even though these niching techniques have been applied to several real-world applications and shown their effectiveness in solving multi-modal optimization, they also have a disadvantage. Their performance is sensitive to two niching parameters: the number of niches and the distances among niches. However, most optimization problems are black box, and it is hard to set appropriate values to them.

Another approach is to use two-stage search algorithms. They acquire location information of niche areas in a search space in the first stage and then applies an optimization method to obtain the local optimum of each niche area in the second stage. The difficulty of this two-stage search is to obtain knowledge of niche location on a fitness landscape.

The Hill-Valley test [19] can determine whether two search points are in the same hill (niche). To achieve that, t internal interpolation points are needed to be sampled evenly between two search points, and their fitness is compared. When the fitness of all interpolation points are worse than the fitness of both search points, there is a valley between the two search points, and it is assumed that the two search points belong to different peaks (niches). This test is a simple and an effective clustering method of peak detection, but it increases additional fitness evaluation cost to test all interpolation points. The multi-national evolutionary algorithm [19] used the Hill-Valley test to each of newly sampled search points and determined whether it belongs to an existing niche or a new niche is found.

Topographical selection [20] uses uniform random search points and can determine a subset S of search points that approximates local optima. S is determined to approximate distinct local optima. To achieve that, each search point connects to its k nearest neighbors to draw k directed edges to build a directed graph. The direction of the edges is always pointing toward to the search point that has better fitness. The search points without outgoing edge are selected as a subset S and are used as initial individuals for local search to obtain multiple local optima.

NBC and NBC2 construct nearest better spanning trees and clusters the initial population using the trees. First, it creates a spanning tree by connecting each search point to the nearest search point with better fitness. Afterwards, the spanning tree is split into several connected components by removing its *long* edges. Each component identifies a distinct niche of multimodal functions. There are two rules for detecting long edges. The first rule simply removes all edges which lengths are longer than a threshold, $\varphi\times$ the average length of edges [7], and a suggested value of the φ in the reference [7] is 2. The second rule is applied to the edges that have more than equal three incoming edge connections [7], [8]. This rule removes an edge(s) if its length is more than $\beta\times mdist$, where *mdist* is the median of the incoming edges of the same search point, and β is calculated by the number of search points and

the dimension of an optimization task.

III. PROPOSED METHOD

A. Basic Idea

Firstly, we define terminology and symbols used in this paper in the Table I. They are visualized in the Fig. 1.

TABLE I
DEFINITION OF TERMINOLOGY AND SYMBOLS USED IN THIS PAPER.

| Terminology	Definition
niche	A local peak area containing only one local optimum.
local best	The highest search point in a *niche*.
edge	Connection line between two search points.
clustering, cluster	Grouping search points in a local peak area.
N	Initial population size.
$CV_{i,j}$	Climbing value of the edge between the i-th and the j-th search points.
$d_{i,j}$ and $g_{i,j}$	Distance and gradient values between the i-th and the j-th search points.
$\hat{d}_{i,j}$ and $\hat{g}_{i,j}$	Normalized $d_{i,j}$ and $g_{i,j}$.
w_g and w_d	Weights of gradient and distance information.
L	Directed graph list.
L_i	The i-th edge in L.
C	Clusters in form of connected components of a graph.
α	Step size of the weight determination method 1, $WGraD_1$.
D	Dimension of a search space.
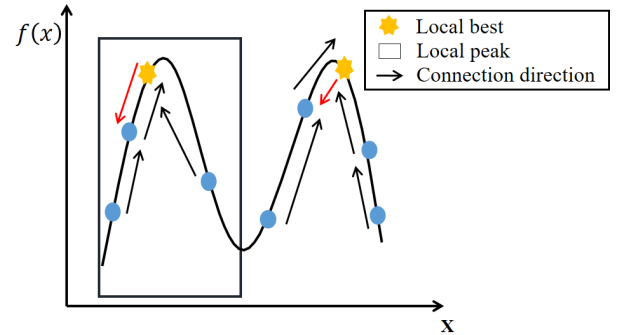
Fig. 1. Visualized terms used in this paper.

Our proposed method for the first stage, weighted gradient and distance-based clustering (WGraD), firstly initializes N search points and creates spanning trees (clusters) by connecting each point to a near one not to generate long edges according to the following two rules:

Rule 1: A search point climbs up in a niche until it meets a near better search point and connects the searched point.

Rule 2: The local best climbs down in a niche until it meets a neighbor search point and connects the search point.

The outline procedure of WGraD is shown in the Algorithm 1. The WGraD connects the i-th search point with the search point that has the maximum climbing function value among the remaining $N-1$ search points; the climbing function,

Algorithm 1 Outline procedure of WGraD.
___
Input: initial individuals of population size N, and parameters w_g and w_d.
Output: clusters in form of connected components of a graph C, directed graph list L
1: Create a directed graph L;
2: **for** all $i \in \{1, ..., N\}$ **do**
3:     Create $N-1$ directed edges between the i-th point and the remaining $N-1$ point and calculate the distance value and the gradient value of each edge by Eq. (3);
4:     Normalize distance and gradient by Eq. (2);
5:     Calculate the climbing value of each edge using normalized distance and gradient by Eq. (1);
6:     Copy the edge with the maximum climbing value to L;
7: **end for**
8: Find n connected components $C = \{C_1, ..., C_n\}$ by list L;
___

$CV_{i,j}$, is a linear combination of gradient information and distance information defined by the Eq. (1).

$$CV_{i,j} = w_g \times \hat{g}_{i,j} + w_d \times \hat{d}_{i,j} \qquad (1)$$

$$\hat{g}_{i,j} = \frac{g_{i,j} - \min(d_{i,j})}{\max(g_{i,j}) - \min(g_{i,j})}, \ \hat{d}_{i,j} = \frac{\max(d_{i,j}) - d_{i,j}}{\max(d_{i,j}) - \min(d_{i,j})} \qquad (2)$$

$$g_{i,j} = (f(\overrightarrow{x_j}) - f(\overrightarrow{x_i}))/d_{i,j}, \ d_{i,j} = \|\overrightarrow{x_i} - \overrightarrow{x_j}\| \qquad (3)$$

where $\overrightarrow{x_i}$ and $f(\overrightarrow{x_i})$ are the i-th search point and its fitness value, respectively; $\max(d_{i,j})$, $\min(d_{i,j})$, $\max(g_{i,j})$, and $\min(g_{i,j})$ $(1 \le i, j \le N, i \ne j)$ are the maximum distance, minimum distance, maximum gradient, and minimum gradient among $N-1$ search points, respectively. See other symbols in the Table I. Gradient metric is proportional to a gradient value between two search points, while distance metric is in inverse proportion to a distance between them. Gradient value and distance value can be calculated by Eq. (2) and Eq. (3), respectively. Note that $0 \le \hat{d}_{i,j} \le 1$ and $0 \le \hat{g}_{i,j} \le 1$.

Clustering results must be influenced by the trade-off control of w_g and w_d. When w_g is bigger than w_d, each search point trends to connect a better search point to obtain a smaller number of spanning trees; on the contrary, when w_g is smaller than w_d, each search point tends to be connected with a closer search point to produce a larger number of spanning trees that only cover tiny areas.

Our proposed clustering algorithm differs from NBC in regard to the below three points:

1) WGraD uses both values of gradient and distance to determine which search point should be connected,
2) our method forms multiple spanning trees by making the local best connect a worse internal search point to reduce the risk of merging two niches into one, and
3) controlling w_g and w_d can change clustering pattern.

How should we determine the weighting ratio of w_g and w_d? We propose two weight determination methods 1 and 2, combine each of them with WGraD, i.e. WGraD$_1$ and

WGraD$_2$, and solve the difficulty of determining weights in the next two subsections.

B. Weight Determination Method 1: WGraD$_1$

The climbing function Eq. (1) can be rewritten as Eq. (4) using only one parameter w_d. The bigger parameter w_d becomes, the more important distance factor becomes, and more spanning trees are constructed.

$$CV_{i,j} = (1 - w_d) \times \hat{g}_{i,j} + w_d \times \hat{d}_{i,j}, \quad w_d \in [0,1] \qquad (4)$$

We propose a dynamic process of increasing w_d gradually to determine a suitable w_d without any heuristic information about the fitness landscape of a problem (see the Fig. 2). It breaks up edges and applies the Hill-Valley test to these edges to merge tiny subtrees that belong to the same niche and improve the accuracy of niche detection. Concretely speaking, w_d is increased from 0 to 1 with a step size α. The increase of the value of w_d relatively increases the importance of a distance factor, and therefore some search points tend to connect a nearer but worse point instead of a far but better one. It results to correspond to *cut* some edges, and therefore one spanning tree is divided into several subtrees.

The Fig. 3 illustrates how search points are connected according to the weighs of w_d and w_g. When smaller w_d and bigger w_g $(= 1 - w_d)$ are given, the point B is connected to the *better* point A at the next step w_d $(= w_d + \alpha)$, and one big tree is made. However, when bigger w_d and smaller w_g are given, the point B is connected to the *near* ponit C at the next step, and two trees are formed.

We determine whether edges can be connected according to the Algorithm 2 in each iteration, apply the Hill-Valley test to two searched endpoints, and prune the edge or add a new edge if necessary. When the two endpoints belong to the same niche, the two subtrees containing the two endpoints are merged into one spanning tree and internal interpolation points in the test are added into search points; otherwise, we ignore the test points and detect a new potential niche.

___
Algorithm 2 Broken Edge Detection
___
Input: information of the current clusters $C = \{C_1, ..., C_n\}$, the number of clusters n, directed graph list L obtained in previous iteration with the weight of one step before, $w_d - \alpha$.
Output: broken edges E
1: create an edge list E;
2: **for** all $i \in \{1, ..., n\}$ **do**
3:     $J \leftarrow$ indices of the points in the i-th cluster C_i;
4:     **for** all $j \in J$ **do**
5:         $p \leftarrow$ the j-th point in the i-th cluster C_i;
6:         $q \leftarrow$ the search point that connected by p in L;
7:         **if** q is not a member in the i-th cluster C_i **then**
8:             Copy the edge between p and q to E;
9:         **end if**
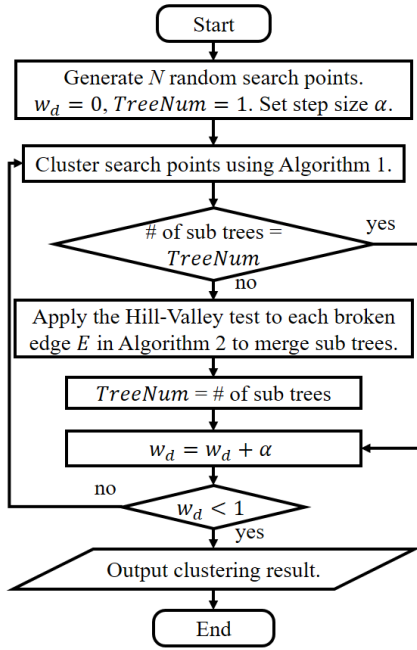10:   **end for**
11: **end for**
___

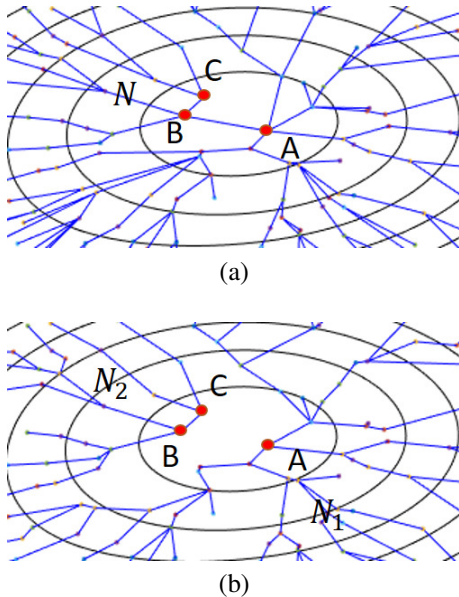Fig. 2. Flow of a proposed WGraD$_1$.

(a)

(b)

Fig. 3. Fig.3 Influence of weighs, w_d and w_g, on search point connection. (a) The case of smaller w_d and bigger w_g ($= 1 - w_d$): each search point is connected to a *better* one, and one big spanning tree is formed. (b) The case of bigger w_d and smaller w_g: each search point is connected to a *near* one, and two spanning trees are formed.

C. Weight Determination Method 2: WGraD$_2$

We propose another method for determining weights by assigning different weights to different search points based on their distance factor. We rewrite Eq. (1) as Eq. (5) to give different weights to different search points.

We assign a low weight to a distance factor and a high weight to a gradient factor to a near search point each by each

to connect better search points in a local area. Conversely, we assign a high weight to a distance factor but a low weight to a gradient factor of a far search point not to connect better but far search points. See the Eq. (2) to determine $\hat{d}_{i,j}$.

$$CV_{i,j} = (1 - \hat{d}_{i,j}) \times \hat{g}_{i,j} + \hat{d}_{i,j} \times \hat{d}_{i,j} \qquad (5)$$

D. Two-stage Search Niching Algorithm

To evaluate the performance of our proposal, we combine NBC, WGraD$_1$ and WGraD$_2$ with differential evolution (DE) and make two-stage search niching algorithms, respectively. The first stage clusters the initial N search points using each of these three algorithms to separate local optima areas, and the second stage searches the local optimum of each local area using any optimization methods. We use DE with NP population size for the second stage in this paper.

IV. EXPERIMENTAL EVALUATIONS

We use eight benchmark functions with six dimensional settings: D =2, 4, 6, 8, 10, and 20 from the CEC2015 multi-niche benchmark test suite [21] in our experimental evaluations. They were designed for real parameter bound constrained single-objective optimization. Table III shows their types, characteristics and category types. These landscape characteristics include shifted, rotated, combination and multimodal. These benchmark functions are classified into two categories; (category 1) local optimal peaks locate far away each other, and (category 2) those locate close each other.

Since these benchmark functions do not contain all the dimensions that need to be tested, we generate a rotation matrix for excluded dimensions according to the guideline of the benchmark. For a D dimensional problem, we rotate the (x_k, x_{k+1})-plane ($k = 1, 2, ..., D - 1$) θ degrees (here, $\theta = 30°$) to rotate a whole search space.

The parameter α described in the section III-B is set as $\alpha = 0.1$ in this experiments. DE parameter settings are described in the Table II, where the symbols are taken from the original reference [22]. There are three DE termination conditions: (1) the number of fitness calls reaches to $20,000 \times D$ times, where D is dimensions, (2) any distances among individuals become less than $1.0e^{-8}$, and (3) the fitness improvement over the last T ($= 50$) generations becomes less than $1.0e^{-8}$.

When the Hill-Valley test detects that the best search point of a cluster belong to any niche that has been already found, we do not optimize the cluster because the niche area would be already searched. Finally, a whole optimization is terminated when the DE termination condition (1) is satisfied or all core search processes are finished.

We evaluate convergence against the number of fitness evaluations rather than generations for fair evaluations. The maximum number of fitness function calls is set to $20,000 \times D$. The search space is $[-100, 100]^D$ for each D-dimensional test function based on the guideline of CEC2015 benchmark test suit. The tolerance value (level of accuracy) is set to 0.1, which means that if the distance between a found optimal search point and an actual optimal search point is shorter than the

Parameter	Value
# of search points for clustering	$N = 200D$
DE population size	$NP = 80$
DE scale factor	$F = 0.5$
DE crossover rate	$Cr = 0.5$
DE operations	DE/current-to-best/1/bin
max. # of fitness evaluations	20,000D
# of trial runs	30

TABLE III
CEC2015 NICHING BENCHMARK FUNCTIONS. D IS A DIMENSION SIZE, AND THE CATEGORIES 1 AND 2 ARE FUNCTIONS WHICH LOCAL OPTIMAL PEAKS LOCATE FAR AWAY EACH OTHER AND CLOSE EACH OTHER, RESPECTIVELY.

No.	Functions	Category
F_1	Shifted and Rotated Expanded Two-Peak Trap	1
F_2	Shifted and Rotated Expanded Five-Uneven-Peak Trap	2
F_3	Shifted and Rotated Expanded Equal Minima	2
F_4	Shifted and Rotated Expanded Decreasing Minima	2
F_5	Shifted and Rotated Expanded Two-Peak Trap	2
F_6	Shifted and Rotated Expanded Himmelblau's Function	1
F_7	Shifted and Rotated Expanded Six-Hump Camel Back	1
F_8	Shifted and Rotated Modified Vincent Function	2

tolerance value, we assume that the optimal search point is found out.

We count the average number of found optimum/optima and the average number of clusters after 30 trial runs and analyze the performance of these algorithms. Since we only know the information of global/local optima, we compare the average number of found global optimum/optima and the average number of clusters. If a method can obtain more clusters, it is likely to obtain more local/global optima. We apply the Wilconxon signed-rank test and Holm's multiple comparison for the two performance measurements at the stop condition of three methods to check the significance difference among them. The Tables IV and V show the statistical test result of the two performance measurements, respectively.

TABLE IV
STATISTICAL TEST RESULTS OF THE WILCONXON SIGNED-RANK TEST AND HOLM'S MULTIPLE COMPARISON FOR AVERAGE NUMBER OF FOUND GLOBAL OPTIMUM OR OPTIMA AMONG THREE ALGORITHMS AT THE STOP CONDITION. A ≪ B AND A < B MEAN THAT B IS SIGNIFICANTLY BETTER THAN A WITH SIGNIFICANT LEVELS OF 1% AND 5%, RESPECTIVELY. A ≈ B MEANS THAT THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN THEM. ~ MEANS THAT THERE IS NO SIGNIFICANT DIFFERENCE AMONG THE THREE METHODS. 0: NBC2 + DE; 1: WGRAD$_1$+ DE; 2: WGRAD$_2$+ DE; D: DIMENSION.

	2-D	4-D	6-D	8-D	10-D	20-D
F_1	~	~	~	0<2≈1	~	~
F_6	~	0≪1≈2	0≪1≪2	0≪1≪2	0≪1<2	0≪1<2
F_7	~	0≪2≈1	0≪1≪2	0≪1≪2	0≪1<2	0≪2≈1
F_2	~	0≪1≈2	0≪1<2	0≪2≈1	0≪1≈2	0≪1≈2
F_3	2≪1<0	~	~	~	~	~
F_4	~	0≈2<1	~	~	~	~
F_5	2≪1≈0	0≈2≪1	~	~	~	~
F_8	~	0≪1≪2	0≪1≪2	0≪1≈2	0≪1≈2	0≪1≈2

TABLE V
STATISTICAL TEST RESULTS OF THE WILCONXON SIGNED-RANK TEST AND HOLM'S MULTIPLE COMPARISON FOR AVERAGE NUMBER OF CLUSTERS AMONG THREE ALGORITHMS AT THE STOP CONDITION. A ≪ B AND A < B MEAN THAT B IS SIGNIFICANTLY LARGER THAN A WITH SIGNIFICANT LEVELS OF 1% AND 5%, RESPECTIVELY. A ≈ B MEANS THAT THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN THEM. ~ MEANS THAT THERE IS NO SIGNIFICANT DIFFERENCE AMONG THE THREE METHODS. D IS A DIMENSION. 0, 1, AND 2 MEAN NBC2, WGRAD$_1$, AND WGRAD$_2$, RESPECTIVELY.

	2-D	4-D	6-D	8-D	10-D	20-D
F_1	2≪1≪0	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1
F_6	1≪2≈0	0≪1≪2	0≪1≪2	0≪1≪2	0≪1≪2	0≪1≪2
F_7	0≈1≪2	0≪1≪2	0≪1≪2	0≪1≪2	0≪1≪2	0≪1≪2
F_2	2≪1≪0	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1
F_3	2≪1≪0	0<2≪1	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1
F_4	2≪1≪0	0<2≪1	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1
F_5	2≪1≪0	0≈2≪1	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1
F_8	2≪1≪0	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1	0≪2≪1

V. DISCUSSION

Firstly, we discuss on the characteristics of the local peak detection algorithms. NBC uses distance information and fitness information and has shown its effectiveness in low-dimensional functions, i.e. almost 2-D functions. It constructs multiple spanning trees by cutting long edges using only distance information. However, it is hard for NBC to obtain many clusters in a high dimensional search space. Some researches [11], [19] indicated that distance between search points did not vary significantly when there are few search points compared to the size of the search space, which means that the number of limited search points become sparse especially in high-dimensional problems. It becomes hard for NBC to prune long edges when they are longer than expected lengths in high-dimensional problems.

Unlike the pruning method of NBC, the core mechanism of WGraD is to use both weighted distance information and weighted gradient information to connect a *local best* to a neighbor search points to avoid creating a long edge connecting two *local best*'s. This is a reason why our WGraD is significantly better than NBC for F_1, F_6 and F_7 of the (category 1). However, a *local best* easily connects another *local best* and loose a peak, when multiple *local best*'s are very close to the *local best*. This is a reason why there are not always significant different between WGraD and NBC for other (category 2) functions.

Fig. 4 explains these characteristics of our WGraD. The *local best* A selects a worse search point B among its neighbor points instead of other *local best* C locating far away, makes an edge between the A and the B, and constructs a local spanning tree. On the other hand, the connection of C-E-F-G illustrates the case that two *local best*'s are connected and actual multiple local peaks become one big local peak because they locate closely.

Secondly, we compare the characteristics of benchmark functions. The low dimensional category 1 functions have the characteristic that a *local best* can find multiple searching points belonging to this niche in its local area, because local
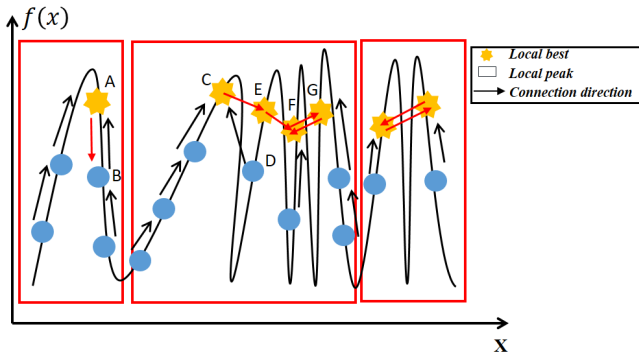
Fig. 4. A 2-D demonstration of missing niches using the proposed method. Connecting a *local best* with a near better *local best* can result in missing a niche.

	Low dimensional tasks	High dimensional tasks
Category 1	NBC and WGrad both show good performance.	WGraD shows better performance.
Category 2	NBC shows better performance.	

peak is far away from each other. Both NBC and WGraD can detect all niches easily. However, for the low dimensional category 2 functions, local peaks are close to each other, which means that *local best*'s are close to each other. NBC can cut long edges in this case and construct more spanning trees than WGraD of this kind of problem, which means that NBC is more likely to find more optima.

To analyze the performance of our proposal, the Wilconxon signed-rank test and Holm's multiple comparison are applied to check the significant difference at the stop condition. From the results of the statistical tests, we have found the following four results:

1) WGraD obtained more clusters of high dimensional benchmark functions (4-D, 6-D, 8-D, 10-D, and 20-D) and showed better performance for F_1, F_2, and F_6 - F_8 than NBC2.
2) NBC2 achieved better performance and obtain more niches than WGraD for 2-D F_2 and F_3 (low dimension in the category 2).
3) None of three algorithms converged to the global optimal area of F_3-F_5 in high dimensional case. (The global optimum might be hard to be found or the search algorithm were trapped at local optima.
4) All three methods found all global optima of 2-D F_1, F_6, and F_7 (low dimension in the category 1).

For the above reasons, we summarize the pros and cons in the Table VI. Since both NBC and WGraD use distance information and fitness (gradient) information, and the two methods are complementary from the application point of view, there are two potential ways to overcome the shortcoming of the two methods to achieve better performance. The first potential method is to switch local area segmentation method according to the problem category. The second one is to first create spanning trees using WGraD and then apply the two rules to prune long edges in spanning trees.

Finally, we would like to show some topics and potential improvements. Since it is difficult to prepare enough number of search points covering a whole search space, it is necessary to develop more efficient sampling methods to detect every

peaks. It is important for two-stage search algorithms to use algorithms having a powerful exploitation ability to find the optimum in a tiny local area. Thus, another improvement of the two-stage search algorithms is to develop powerful local search algorithms. High-level control strategy would be able to improve the performance of the two-stage niching algorithms, such as running a two-stage niching algorithm repeatedly until a stop condition or running a local area segmentation method to cluster individuals in their respective populations in every generation.

VI. CONCLUSION

We introduce a weighted gradient and distance-based clustering to detect niches in the fitness landscape. The experiments have confirmed that our method and NBC are complementary to each other, which the one can overcome the shortcoming of the other. In our future work, we will consider a high-level control strategy to improve the performance of our clustering method and a switching method that can decides which method to use in a black box optimization problem.

ACKNOWLEDGMENT

REFERENCES

[1] G.-C. Luh and C.-Y. Lin ,"Optimal design of truss-structures using particle swarm optimization," Comput. Struct., vol. 89, nos. 23–24, pp. 2221–2232, Dec. 2011.
[2] A. Ward, J. K. Liker, J. J. Cristiano, and D. K. Sobek, "The second Toyota paradox: How delaying decisions can make better cars faster," Sloan Manag. Rev., vol. 36, no. 3, pp. 43–61, 1995.
[3] J. W. Kruisselbrink et al., "Enhancing search space diversity in multi-objective evolutionary drug molecule design using niching," 11th Annual Conf. on Genetic and Evolutionary Computation (GECCO2009), Montreal, QC, Canada, pp. 217–224, 2009.
[4] K.-C. Wong, K.-S. Leung, and M.-H. Wong, "Protein structure prediction on a lattice model via multimodal optimization techniques," 12th Annual Conf. Companion on Genetic and Evolutionary Computation (GECCO2010), Portland, OR, USA, pp. 155–162, 2010.
[5] K. M. Malan and A.P. Engelbrecht, "A survey of techniques for characterisingfit-ness landscapes and some possible ways forward," Information Sciences, vol. 241, pp. 148–163, 2013.
[6] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple search points: An updated survey on niching methods and their applications," IEEE Trans. on Evolutionary Computation, vol. 21, no. 4, pp. 518–538, Aug. 2017.
[7] M. Preuss, "Niching the CMA-ES via nearest-better clustering," 12th Annual Conf. Companion on Genetic and Evolutionary Computation (GECCO2010), Portland, OR, USA, pp. 1711–1718, 2010.
[8] M. Preuss, "Improved Topological Niching for Real-Valued Global Optimization," Applications of Evolutionary Computation, Berlin, Heidelberg: Springer, pp. 386–395, 2012.

[9] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," IEEE Int. Conf. on Evolutionary Computation (ICEC1996), Nagoya, Japan, pp.312–317, 1996.

[10] S. C. Maree, T. Alderliesten, D. Thierens, and P.A.N. Bosman, "Real-valued evolutionary multimodal optimization driven by hill-valley clustering," Genetic and Evolutionary Computation Conference (GECCO2018), Kyoto, Japan, pp. 857–864, 2018.

[11] M. W. Pereira, G. S. Neto, and M. Roisenberg. "A Topological Niching Covariance Matrix Adaptation for Multimodal Optimization". IEEE Congress on Evolutionary Computation (CEC2014), Beijing, China, pp. 2562–2569, July 2014.

[12] D. J., Cavicchio, "Adaptive search using simulated evolution," Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1970.

[13] E. G. David and R. Jon, "Genetic algorithms with sharing for multimodal function optimization," in Proceedings of the Second Int. Conf. on Genetic algorithms and their application, Hillsdale, NJ, USA, pp. 41–49, July 1987.

[14] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," in PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975.

[15] J. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," Evolutionary Computation, vol. 10, no. 3, pp. 207–234, 2002.

[16] A. Ahrari, K. Deb, and M. Preuss, "Multimodal optimization by co-variance matrix self-adaptation evolution strategy with repelling subpopulations," Evolutionary Computation, vol. 25, no. 3, pp. 439–471, Sep. 2017.

[17] M. Bessaou, A. Petrowski, and P. Siarry, "Island model cooperating with speciation for multimodal optimization," in 6th Int. Conf. on Parallel Problem Solving from Nature, 2000, pp. 437–446.

[18] A. Ahrari, K. Deb, and M. Preuss, "Multimodal optimization by co-variance matrix self-adaptation evolution strategy with repelling subpopulations," Evolutionary Computation, vol. 25, no. 3, pp. 439–471, Sep. 2017.

[19] R.K. Ursem, "Multinational evolutionary algorithms," in Proceedings of the IEEE Congress on Evolutionary Computation - CEC, Washington, DC, USA, USA, vol. 3, pp. 1633–1640, 1999.

[20] A. Törn and S. Viitanen, "Topographical Global Optimization," In: C. A. Floudas and P. M. Pardalos (eds.), Recent Advances in Global Optimization, Princeton University Press, pp. 384–398, 1992.

[21] B. Y. Qu, J. J. Liang, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on single objective multi-niche optimization," Technical Report 201411B, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, Nov., 2014.

[22] R. Storn and K. Price, "Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.