



## On the effects of smoothing rugged landscape by different toy problems: A case study on UBQP

Wei Wang, Jialong Shi, Jianyong Sun, Arnaud Liefoghe, Qingfu Zhang, Ye Fan

### ► To cite this version:

Wei Wang, Jialong Shi, Jianyong Sun, Arnaud Liefoghe, Qingfu Zhang, et al.. On the effects of smoothing rugged landscape by different toy problems: A case study on UBQP. CEC 2024 – IEEE Congress on Evolutionary Computation, Jun 2024, Yokohama, Japan. pp.1-8, 10.1109/CEC60901.2024.10612165 . hal-04692908

**HAL Id: hal-04692908**

**<https://hal.science/hal-04692908v1>**

Submitted on 10 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# On the Effects of Smoothing Rugged Landscape by Different Toy Problems: A Case Study on UBQP

1<sup>st</sup> Wei Wang

*School of Mathematics and Statistics  
Xi'an Jiaotong University  
Xi'an, China  
ww123@stu.xjtu.edu.cn*

2<sup>nd</sup> Jialong Shi

*School of Mathematics and Statistics  
Xi'an Jiaotong University, Xi'an, China  
Sichuan Digital Economy Industry  
Development Research Institute, Chengdu, China  
jialong.shi@xjtu.edu.cn*

3<sup>rd</sup> Jianyong Sun

*School of Mathematics and Statistics  
Xi'an Jiaotong University  
Xi'an, China  
jy.sun@xjtu.edu.cn*

4<sup>th</sup> Arnaud Liefoghe

*LISIC  
Université du Littoral  
Côte d'Opale (ULCO)  
Calais, France  
arnaud.liefoghe@univ-littoral.fr*

5<sup>th</sup> Qingfu Zhang

*Department of Computer Science  
City University of Hong Kong  
Hong Kong, China  
qingfu.zhang@cityu.edu.hk*

6<sup>th</sup> Ye Fan

*School of Electronics and Information  
Northwestern Polytechnical University, Xi'an, China  
Research and Development Institute of  
Northwestern Polytechnical University in Shenzhen  
Shenzhen, China  
fanye@nwpu.edu.cn*

**Abstract**—The hardness of the Unconstrained Binary Quadratic Program (UBQP) problem is due its rugged landscape. Various algorithms have been proposed for UBQP, including the Landscape Smoothing Iterated Local Search (LSILS). Different from other UBQP algorithms, LSILS tries to smooth the rugged landscape by building a convex combination of the original UBQP and a toy UBQP. In this paper, our study further investigates the impact of smoothing rugged landscapes using different toy UBQP problems, including a toy UBQP with matrix  $\hat{Q}^1$  (construct by “+/-1”), a toy UBQP with matrix  $\hat{Q}^2$  (construct by “+/-i”) and a toy UBQP with matrix  $\hat{Q}^3$  (construct randomly). We first assess the landscape flatness of the three toy UBQPs. Subsequently, we test the efficiency of LSILS with different toy UBQPs. Results reveal that the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) exhibits the flattest landscape among the three, while the toy UBQP with  $\hat{Q}^3$  (construct randomly) presents the most non-flat landscape. Notably, LSILS using the toy UBQP with  $\hat{Q}^2$  (construct by “+/-i”) emerges as the most effective, while  $\hat{Q}^3$  (construct randomly) has the poorest result. These findings contribute to a detailed understanding of landscape smoothing techniques in optimizing UBQP.

**Index Terms**—Unconstrained Binary Quadratic Programming (UBQP), Landscape smoothing, Homotopic convex (HC) transformation.

## I. INTRODUCTION

The Unconstrained Binary Quadratic Program (UBQP) [1] has attracted much attention in the field of combinatorial optimization due to its practical applications in financial analysis [2], molecular conformation [3], traffic management [4] and so on. The aim of this NP-hard problem is to determine binomial decision variables that will maximize a quadratic objective function by choosing binary decision variables. The

UBQP problem can be formalized as follows [5]:

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j \\ \text{subject to} \quad & \mathbf{x} \in \{0, 1\}^n \end{aligned} \quad (1)$$

where  $\mathbf{Q} = [Q_{ij}]$  is an  $n \times n$ -dimensional matrix and  $\mathbf{x}$  is an  $n$ -dimensional vector with binary variables. UBQP serves as a common model for a wide range of combinatorial optimization problems, such as maximum cut problems [6] and set partitioning problems [7], among many others [8].

There exist difficulties in finding a global optimal solution within polynomial time because of the NP-hardness of UBQP [9]. Numerous heuristics and metaheuristics have been developed over the years in order to cope with this computational challenge. UBQP is characterized by a rugged and irregular fitness landscape, which makes the task of solving it challenging, particularly when there are numerous local optima within the search landscape [20]. However, to the best of our knowledge, very few research efforts have been committed to smoothing the landscape of UBQP.

In an effort, Wang et al. [10] introduce a landscape smoothing method called Homotopic Convex (HC) transformation for the UBQP by employing a toy UBQP problem formed by 1s and -1s. They design the Landscape Smoothing Iterated Local Search (LSILS) algorithm based on this HC transformation. However, no study has been conducted on the smoothing effects of HC transformation using different toy problems. This paper extends the investigation into the smoothing effects of HC transformation by introducing two additional toy UBQPs, where the assumption is that toy problems with different landscape flatness have different landscape smoothing effects. Subsequently, employing the previously proposed LSILS al-

gorithm, we evaluate the performance of it using the three toy UBQPs and compare it against a baseline ILS [11]. Results indeed indicate that different toy UBQPs do have different effects on smoothing the landscape.

The rest of this paper is organized as follows. Section II reviews the related work. In Section III, we briefly review the LSILS presented in [10] and introduce two LSILS algorithms with different toy problems. Detailed experimental results are reported in Section IV. Section V presents the conclusion.

## II. RELATED WORK

Generally speaking, there exist two types of algorithms for UBQP: exact methods and heuristics. Since the UBQP problem is NP-hard [1], [9], exact algorithms tend to be time-consuming for large-scale problems. They are not able to obtain the global optimum within limited computing time. Different from exact algorithms, heuristics can obtain near-optimal solutions within a reasonable time for large-scale UBQP instances. Among those are algorithms based on tabu search [12], [13], simulated annealing [14] [15], iterated local search [16], evolutionary algorithm (EA) [17], among others. However, we find that the previous algorithms are rarely based on landscape smoothing. In [10], Wang et al. propose a landscape smoothing method HC transformation for UBQPs and then design a LSILS algorithm based on this method.

The behavior of heuristics is significantly influenced by the characteristics of fitness landscapes. Therefore, nowadays, the characteristics of fitness landscapes have been widely studied in many combinatorial optimization problems [18]. The difficulty to explore a landscape lies in its ruggedness which is directly related to the number of local optima in the landscape. A great amount of efforts have been made on fitness landscape analysis. With regard to UBQP, Tari et al. found a big-valley structure in UBQP landscapes [19] [20]. They point out that local optima tend to be grouped in a sub-region of the landscape. Merz and Katayama [21] conducted a landscape analysis of UBQP and observed that local optima are contained in a small fraction of the search space, which corroborates previous research. Based on this feature, they proposed a memetic algorithm which is very effective in solving UBQP problems with up to 2 500 variables. In the work of Tari et al. [22], they used a recently proposed method, the monotonic local optima networks [23], to study the induced fitness landscapes for understanding the difference in performance among five pivoting rules.

One of the main challenge of combinatorial optimization comes from rugged landscapes. Some studies have proposed landscape smoothing based algorithms to improve algorithm performance. To the best of our knowledge, there are few landscape smoothing methods for UBQP, so we briefly introduce some related methods for other combinatorial optimization problems below. Gu and Huang [24] proposed a landscape smoothing method by edge cost manipulation. Liang et al. [25] considered using local search to smooth the fitness landscape and thus preventing sticking in local optima for a long time. One year later, Coy et al. [26] proposed the sequential

smoothing algorithm (SSA) that alternates between convex and concave smoothing function to avoid being trapped in poor local optima. Hasegawa and Hiramatsu [27] suggested that metropolis algorithm can be used effectively as a local search algorithm in search-space smoothing strategies. Recently, Shi et al. [28] proposed a new landscape smoothing method called HC transformation for the TSP, and this approach was recently extended to UBQP [10].

## III. METHODS

Initially proposed in [28], HC transformation is a technique to smooth the rugged landscape of the traveling salesperson problem (TSP), with the detailed process illustrated in Fig. 1. The TSP can be smoothed by a convex combination of the original TSP and the unimodal TSP which is generated based on a known local optimum of the original TSP. This method was later generalized to UBQP [10]. Given a known locally optimal solution for the considered UBQP instance, a toy UBQP of same size is created, represented by a binary matrix  $Q$  constructed with  $-1$ s and  $1$ s values. Hence, we say that the toy UBQP used in [10] is constructed by the “+/-1” method. The unimodal nature of the toy UBQP landscape is theoretically established in [10], ensuring that any local search process consistently converges towards its single global optimum, mirroring the known local optimum of the original UBQP. Subsequently, a convex combination of this constructed toy UBQP and the original UBQP is implemented, introducing a smoothing effect on the fitness landscape of the original UBQP. Governed by a coefficient denoted as  $\lambda$  within the  $[0, 1]$  range, this process establishes a path from the original UBQP to the unimodal toy UBQP, allowing for an exploration of landscape characteristics and the impact of smoothing on search performance.

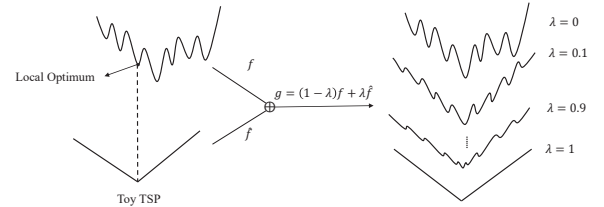


Fig. 1. Effect of HC transformation for the TSP [10].

The LSILS algorithm represents an optimization approach that integrates the HC transformation technique with ILS. The HC transformation serves as a main component of LSILS, contributing to the landscape smoothing of the target UBQP instance. At the same time, ILS is used to explore and exploit the smoothed landscape, while concurrently updating the smoothed UBQP landscape iteratively to maintain the best solution found so far in the original UBQP. The entire procedure of LSILS is depicted in Algorithm 1, where the algorithm executes an iterative process of a local search procedure and a perturbation procedure. This iterative process ensures the smoothing effect of the solution space and the enhancement of optimization outcomes.

---

**Algorithm 1:** Landscape Smoothing Iterated Local Search for UBQPs

---

```

1  $\mathbf{x}_{ini} \leftarrow$  Randomly generated solution;
2  $\mathbf{x}_{(0)} \leftarrow \text{ILS}(\mathbf{x}_{ini}|f_o)$ ;
3  $\mathbf{x}^* \leftarrow \mathbf{x}_{(0)}$ ;
4  $j \leftarrow 0$ ;
5 while stopping criterion is not met do
6   Construct the unimodal UBQP  $\hat{f}$  based on  $\mathbf{x}^*$ ;
7    $g \leftarrow (1 - \lambda)f_o + \lambda\hat{f}$ ;
8    $\mathbf{x}'_{(j)} \leftarrow \text{Perturbation}(\mathbf{x}_{(j)})$ ;
9    $\{\mathbf{x}_{(j+1)}, \mathbf{x}^*\} \leftarrow \text{LS}(\mathbf{x}'_{(j)}, \mathbf{x}^*|g)$ ;
10   $j \leftarrow j + 1$ ;
11   $\lambda = \text{Update}(\lambda)$ ;
12 return  $\mathbf{x}^*$ 

```

---

Lines 1-4 initiate the LSILS algorithm, initializing the current best solution denoted as  $\mathbf{x}^*$  with respect to  $f_o$ , where  $f_o$  represents the original objective function. Subsequently, LSILS iteratively executes the ensuing steps until the termination criterion is satisfied (lines 5-11). At each iteration, the procedure starts by constructing the objective function  $g$  for the smoothed UBQP based on the current best solution  $\mathbf{x}^*$  (line 7). This objective function  $g$  is obtained by integrating the original UBQP objective function  $f_o$  and the objective function  $\hat{f}$  of the toy UBQP, employing a smoothing factor  $\lambda$ . After that, a local search operation is applied to the smoothed UBQP, initiating from a perturbed solution  $\mathbf{x}'_{(j)}$  derived from the current solution  $\mathbf{x}_{(j)}$  (line 8). Here,  $\text{LS}(\mathbf{x}'_{(j)}, \mathbf{x}^*|g)$  denotes the execution of a local search from  $\mathbf{x}'_{(j)}$  on the smoothed UBQP  $g$ , concurrently updating  $\mathbf{x}^*$  by tracking the original objective function  $f_o$ . The resultant solution, denoted as  $\mathbf{x}_{(j+1)}$ , represents the local optimum for the smoothed UBQP  $g$  (line 9) and is utilized as the input for the next local search after perturbation (line 10). The final step involves the update of the smoothing coefficient  $\lambda$  (line 11), with the specific methodology for this update being user-defined.

In [10], the construction of the toy UBQP (line 6 in Algorithm 1) is based on a simple method called “+/-1”, which is detailed as follows. Given a solution vector  $\mathbf{x}^*$  of dimension  $n$ , the generation of the specialized matrix  $\hat{Q}^1$  is executed through the following process,

$$\hat{Q}^1_{ij} = \begin{cases} 1, & \text{if } x_i^* x_j^* = 1 \\ -1, & \text{if } x_i^* x_j^* \neq 1 \end{cases}, \quad i, j \in \{1, 2, \dots, n\} \quad (2)$$

where the  $i$ -th and  $j$ -th binary variables (0-1) of solution  $\mathbf{x}^*$  are denoted by  $x_i^*$  and  $x_j^*$ , respectively.  $\hat{Q}^1 = [\hat{Q}^1_{ij}]$  represents an  $n \times n$  matrix of the constructed toy UBQP. From the construction process of  $\hat{Q}^1$ , we can see that only when the solution of the toy problem is  $\mathbf{x}^*$ , all the positive elements in  $\hat{Q}^1$  are obtained, while for any solution except  $\mathbf{x}^*$ , a certain number of negative elements will be taken. This reflects the unimodality of the toy UBQP. In [10], it is theoretically proven

that the toy UBQP  $\hat{f}^1$  defined by  $\hat{Q}^1$  exhibits unimodal behavior, indicating that a local search algorithm will consistently converge towards the unique global optimum of  $\hat{f}^1$ , i.e.,  $\mathbf{x}^*$ . The unimodal characteristic of  $\hat{f}^1$  can be used to smooth the landscape of the original UBQP while preserving valuable information in the high quality solution  $\mathbf{x}^*$ , thereby facilitating a more efficient search towards optimal solutions.

Given the unique features of the problem landscape, the efficiency of an algorithm is influenced not only by the smoothness of the landscape but also by its flatness. While smoothness denotes the global regularity within the solution space, flatness focuses on the horizontal characteristics of local regions. In this paper, in order to further examine the impact of the HC transformation on smoothing rugged landscapes by employing different toy problems, we offer two new constructions of the toy UBQP with different flatness, designated as  $\hat{Q}^2$  and  $\hat{Q}^3$ .

Similar to  $\hat{Q}^1$ , the matrix  $\hat{Q}^2$  is constructed based on a method called “+/-i” as follows:

$$\hat{Q}^2_{ij} = \begin{cases} i, & \text{if } x_i^* x_j^* = 1 \\ -i, & \text{if } x_i^* x_j^* \neq 1 \end{cases} \quad (3)$$

where  $i \in \{1, 2, \dots, n\}$  and  $j \in \{1, 2, \dots, i\}$ ,  $i$  is the  $i$ -th row of matrix  $\hat{Q}^2$ . Let  $\hat{Q}^2_{ji} = \hat{Q}^2_{ij}$  to construct the symmetric  $\hat{Q}^2$ . By contrast, the matrix  $\hat{Q}^3$  is constructed based on a so-called “randomly” method:

$$\hat{Q}^3_{ij} = \begin{cases} u_{ij}^1, & \text{if } x_i^* x_j^* = 1 \\ u_{ij}^2, & \text{if } x_i^* x_j^* \neq 1 \end{cases} \quad (4)$$

where  $i \in \{1, 2, \dots, n\}$  and  $j \in \{1, 2, \dots, i\}$ ,  $u_{ij}^1$  is a randomly generated integer from a discrete uniform distribution in the range  $[1, 100]$ , that is  $P(u_{ij}^1 = k_1) = \frac{1}{100}, k_1 \in \{1, 2, \dots, 100\}$  and  $u_{ij}^2$  is a randomly generated integer from a discrete uniform distribution in the range  $[-100, -1]$ , that is  $P(u_{ij}^2 = k_2) = \frac{1}{100}, k_2 \in \{-100, -99, \dots, -1\}$ . Then let  $\hat{Q}^3_{ji} = \hat{Q}^3_{ij}$  to construct the symmetric  $\hat{Q}^3$ .

Since  $\hat{Q}^2$  (Eq. (3)) and  $\hat{Q}^3$  (Eq. (4)) have the same positive/negative element distribution as  $\hat{Q}^1$  (Eq. (2)), which is determined by  $\mathbf{x}^*$ , we can say that the landscapes defined by  $\hat{Q}^2$  and  $\hat{Q}^3$  are both unimodal with  $\mathbf{x}^*$  as the unique optimum.

As an example, for a 5-dimensional UBQP, given  $\mathbf{x}^* = (0, 1, 0, 1, 1)$ , we construct three toy UBQPs based on the above constructions. Initially, it can be seen that  $x_1^* = 0, x_2^* = 1, x_3^* = 0, x_4^* = 1, x_5^* = 1$ , according to Eq. (2), Eq. (3) and Eq. (4), we have  $\hat{Q}^1_{22}, \hat{Q}^1_{44}, \hat{Q}^1_{55}, \hat{Q}^1_{24}, \hat{Q}^1_{42}, \hat{Q}^1_{25}, \hat{Q}^1_{52}, \hat{Q}^1_{45}, \hat{Q}^1_{54} > 0$ , the remaining elements are all negative.

$$\hat{Q}^1 = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 1 \end{pmatrix}$$



$$\hat{Q}^2 = \begin{pmatrix} -1 & -2 & -3 & -4 & -5 \\ -2 & 2 & -3 & 4 & 5 \\ -3 & -3 & -3 & -4 & -5 \\ -4 & 4 & -4 & 4 & 5 \\ -5 & 5 & -5 & 5 & 5 \end{pmatrix}$$

$$\hat{Q}^3 = \begin{pmatrix} -33 & -27 & -52 & -46 & -40 \\ -27 & 62 & -72 & 95 & 11 \\ -52 & -72 & -24 & -18 & -44 \\ -46 & 95 & -18 & 1 & 17 \\ -40 & 11 & -44 & 17 & 21 \end{pmatrix}$$

The aforementioned toy UBQP instances defined by  $\hat{Q}^m$ ,  $m \in \{1, 2, 3\}$  can be used to smooth the original UBQP. After smoothing, the objective function ( $g$ ) is defined as:

$$g(\mathbf{x}|\lambda, \alpha, m) = (1 - \lambda)f_o(\mathbf{x}) + \lambda\alpha\hat{f}^m(\mathbf{x})$$

$$= (1 - \lambda)\mathbf{x}^T \mathbf{Q} \mathbf{x} + \lambda\alpha\mathbf{x}^T \hat{\mathbf{Q}}^m \mathbf{x}, \quad m \in \{1, 2, 3\},$$

where  $f_o$  is the objective function of the original UBQP,  $\mathbf{Q}$  is the matrix of  $f_o$ ,  $\hat{f}^m$  is the objective function of the toy UBQP defined by matrix  $\hat{\mathbf{Q}}^m$ ,  $m \in \{1, 2, 3\}$ .  $\lambda \in [0, 1]$  controls the strength of smoothing and  $\alpha$  is a scaling factor to make sure that the original UBQP  $f_o$  and the toy UBQP  $\hat{f}^m$  are on the same scale. We can see that, when  $\lambda = 0$ , the smoothed UBQP  $g$  degenerates to the original UBQP  $f_o$ , when  $\lambda = 1$ , it is smoothed to the unimodal UBQP  $\hat{f}^m$  and when  $0 < \lambda < 1$ , the original UBQP  $f_o$  can be gradually smoothed to the unimodal UBQP  $\hat{f}^m$ . In fact, The smoothed objective function  $g$  is a homotopic transformation from  $f_o$  to  $\hat{f}^m$ .

#### IV. EXPERIMENTAL ANALYSIS

To comprehensively investigate the impact of the HC transformation on smoothing rugged landscapes using various toy problems, a series of experiments are conducted. To better understand the solution distributions of different toy UBQPs, we first conduct landscape flatness experiments, and then we apply LSILS algorithms to the smoothed UBQPs, which have been modified by means of the HC transformation.

In order to carry out the experiments, we select 10 UBQP instances with a size of  $n = 2500$  from the ORLIB [29]. These instances possess a density of 0.1, where density denotes the proportion of non-zero elements in the matrix  $\mathbf{Q}$ . All 10 instances have known optimal solutions. The experiments are conducted using GNU C++ with the -O2 optimization option on the Tianhe-2 supercomputer, which is equipped with 17 920 computer nodes, each comprising two Intel Xeon E5-2692 12C (2.200 GHz) processors.

The parameter setting is as follows: the maximum value of  $\lambda$  is set to 0.004. The time point to increase  $\lambda$  is based on the CPU time, changing from 0 to 0.001 at 200 s, from 0.001 to 0.002 at 400 s, and so forth. This is because the smoothing effect of the HC transformation depends highly on the quality of the local optimum used to construct the toy UBQP. At the beginning of the search, the quality of the solution is poor, so we set a small  $\lambda$ . As search goes by, better solutions are found, a larger  $\lambda$  is set to make full use

of the high-quality solutions. Considering the zero elements of the original matrix  $\mathbf{Q}$  and the non-sparse nature of the toy UBQPs, to smooth the landscape of the original UBQP without causing great damage, we introduce  $\alpha$  to constrain the elements of  $\hat{\mathbf{Q}}^m$  to be less than or equal to 5. Notably, the choice of 5 is proximate to the average absolute value of the original matrix  $\mathbf{Q}$ . Specifically, for the three different toy UBQPs,  $\alpha$  is set to be 5 for  $\hat{\mathbf{Q}}^1$  (construct by “+/-1”), 0.002 for  $\hat{\mathbf{Q}}^2$  (construct by “+/-i”) and 0.05 for  $\hat{\mathbf{Q}}^3$  (construct randomly). The stopping criterion is at 1 000 seconds of CPU runtime, with information logging conducted every 10 seconds. Each algorithm was executed for 20 independent runs on each benchmark instance, employing different random initial solutions. The local search implementation incorporates the *1-bit-flip* local search, while perturbation is executed randomly, flipping  $n/4$  bits at each call, where  $n$  is the instance size.

##### A. Landscape Analysis of Toy UBQPs

Before the implementation of LSILS on UBQP instances, an initial experimental analysis is conducted to evaluate the landscape flatness of the three different toy problems. This comprehensive experiment aims to provide a thorough understanding of the landscape characteristics.

As a first step of the experiment, we use a 18-dimensional UBQP problem as an example. A 18-dimensional symmetric matrix is generated randomly as an original UBQP problem, whose elements follow a discrete uniform distribution in the range  $[-100, 100]$ . Based on a solution  $\mathbf{x}^* = (1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0)$ , we build the corresponding  $\hat{\mathbf{Q}}^1$  (construct by “+/-1”),  $\hat{\mathbf{Q}}^2$  (construct by “+/-i”) and  $\hat{\mathbf{Q}}^3$  (construct randomly) for the 18-dimensional randomly generated UBQP. Since the instance size is small, we traverse all solutions from the search space.

We first focus on the number of 1-bit-flip local optimum in the solution space of the four UBQPs which can reflect the ruggedness of the landscape. As shown in TABLE I, the original randomly generated  $\mathbf{Q}$  has 5 1-bit-flip local optima, while  $\hat{\mathbf{Q}}^1$  (construct by “+/-1”),  $\hat{\mathbf{Q}}^2$  (construct by “+/-i”) and  $\hat{\mathbf{Q}}^3$  (construct randomly) all have only 1, that is  $\mathbf{x}^*$ . This experiment confirms the unimodality of the three toy UBQPs. We also conduct HC transformation using the three toy UBQPs on the original randomly generated problem with different  $\lambda$  values. To make sure the original UBQP and the three toy UBQPs are in the same scale, we let  $\alpha = 50$  in the HC transformation with  $\hat{\mathbf{Q}}^1$  (construct by “+/-1”),  $\alpha = 2.8$  in the HC transformation with  $\hat{\mathbf{Q}}^2$  (construct by “+/-i”),  $\alpha = 1$  in the HC transformation with  $\hat{\mathbf{Q}}^3$  (construct randomly), where 50 is proximate to the average absolute value of the original randomly generated matrix  $\mathbf{Q}$ . Fig. 2 shows how the number of local optimum of the smoothed UBQP landscape changes against  $\lambda$ . In general, we can see that the number of local optima is approximately negatively related to  $\lambda$  and when  $\lambda > 0.8$ , all the smoothed UBQPs tend to have only 1 local optimum. Based on the above observations, we can conclude that  $\hat{\mathbf{Q}}^1$  (construct by “+/-1”),  $\hat{\mathbf{Q}}^2$  (construct by “+/-i”) and  $\hat{\mathbf{Q}}^3$

(construct randomly) are unimodal and the HC transformation can indeed smooth the UBQP landscape, where the smoothing effect is controlled by  $\lambda$ .

TABLE I  
LOCAL OPTIMA OF FOUR DIFFERENT 18-DIMENSIONAL UBQPs.

Instance	Number of local optimum	1-bit local optimum
Original randomly generated $Q$	5	$x_1 = (0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0)$ $x_2 = (1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0)$ $x_3 = (1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0)$ $x_4 = (0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0)$ $x_5 = (1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1)$
$\hat{Q}^1$ (construct by “+/-1”)	1	$x_1 = (1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0)$
$\hat{Q}^2$ (construct by “+/-i”)	1	$x_1 = (1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0)$
$\hat{Q}^3$ (construct randomly)	1	$x_1 = (1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0)$

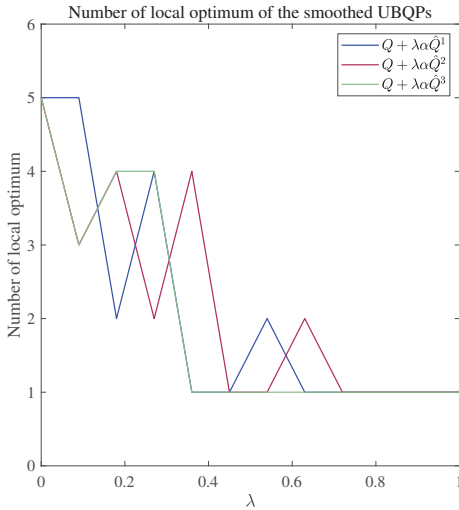


Fig. 2. Number of local optima of the smoothed UBQPs with different  $\lambda$  values.

We then focus on the flatness of the original randomly generated  $Q$ ,  $\hat{Q}^1$  (construct by “+/-1”),  $\hat{Q}^2$  (construct by “+/-i”) and  $\hat{Q}^3$  (construct randomly). Fig. 3a gives a histogram of the original  $Q$  with function values as x axis and the corresponding frequency of each function value as y axis. From Fig. 3a, we can see that the function values tend to cluster around the central region and relatively few values are distributed at both ends, with a range of function values between -1935 and 922. To examine the distributions of function values of the three different toy UBQPs, an 18-dimensional vector is randomly generated from the solution space of the original UBQP, serving as the basis for constructing the toy UBQPs. By following the steps specified for constructing toy

UBQPs in Eqs. (2), (3) and (4), three different toy UBQP instances are obtained. The distributions of function values in the solution space for these three toy UBQPs are illustrated in Fig. 3b, Fig. 3c, and Fig. 3d, respectively. Results reveal that the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) exhibits a relatively flat landscape, with more solutions sharing the same function values. In contrast,  $\hat{Q}^3$  (construct randomly) possesses a relatively non-flat landscape, with fewer solutions with the same function values. The landscape flatness of the toy UBQP with  $\hat{Q}^2$  (construct by “+/-i”) falls between  $\hat{Q}^1$  (construct by “+/-1”) and  $\hat{Q}^3$  (construct randomly), being flatter than  $\hat{Q}^3$  (construct randomly).

To quantify landscape flatness more precisely, the possibility of obtaining two solutions with different function values is calculated. This involves first summing the function value possibilities and then squaring the result. The resulting possibilities for the original UBQP, toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”), toy UBQP with  $\hat{Q}^2$  (construct by “+/-i”), and toy UBQP with  $\hat{Q}^3$  (construct randomly) are 0.000795, 0.054152, 0.001924, and 0.000413, respectively. This metric reflects that though the landscape of the toy UBQP with  $\hat{Q}^3$  (construct randomly) is not as flat as the original UBQP, it remains unimodal. Conversely, the landscape of the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) appears excessively flat. The varying flatness of the toy UBQPs will imply diverse landscape smoothing effects on the original UBQP while owning the unimodal structure. The landscapes of the four UBQP problems are sketched in Fig. 4.

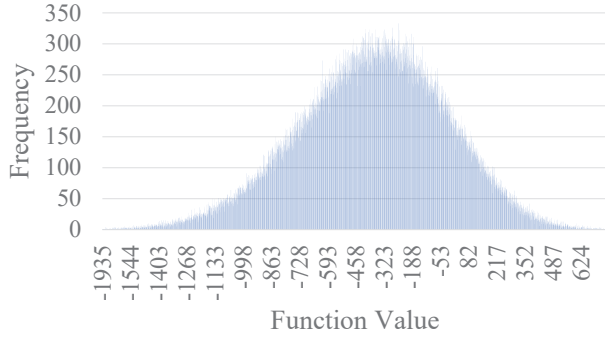
### B. Performance of LSILS with Different Toy UBQPs

According to the previous analysis, three different toy UBQPs we designed have different landscapes. To study the landscape smoothing effects of the HC transformation with different toy UBQPs, we then conduct experiments on 10 large UBQP instances using LSILS. To measure the quality of a solution, we use its relative deviation to the optimum, which is defined as:

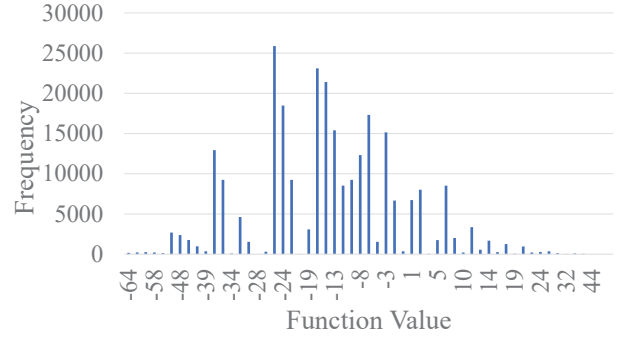
$$\text{excess} = \frac{f(x_{\text{LO}}) - f(x_{\text{opt}})}{f(x_{\text{opt}})}, \quad (5)$$

where  $f(x_{\text{opt}})$  denotes the known global optimum and  $f(x_{\text{LO}})$  is the solution returned by the algorithm. In order to compare the effect of LSILS algorithms with different toy UBQPs, we use the result of ILS and LSILS using the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) from [10]. ILS iteratively executes a local search procedure and a perturbation procedure on the original UBQP landscape until the stopping criterion is met. The experimental settings are the same as before.

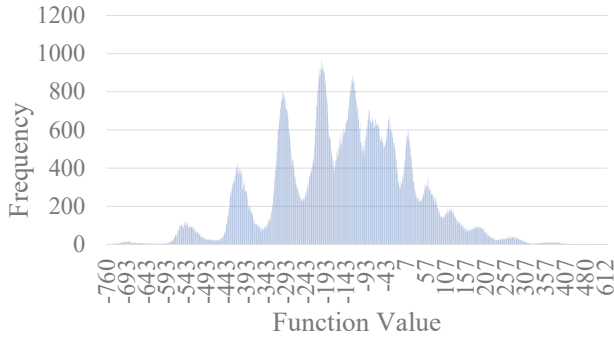
Fig. 5 reports the average relative deviation of the best solution found by LSILS and ILS against CPU time. A lower value is better. From Fig. 5, we can see that in all instances, LSILS using the toy UBQP with  $\hat{Q}^2$  (construct by “+/-i”) performs better than ILS. LSILS using the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) performs better than ILS on 8 instances except bqp2500.5 and bqp2500.9. In 9 out of 10 instances,



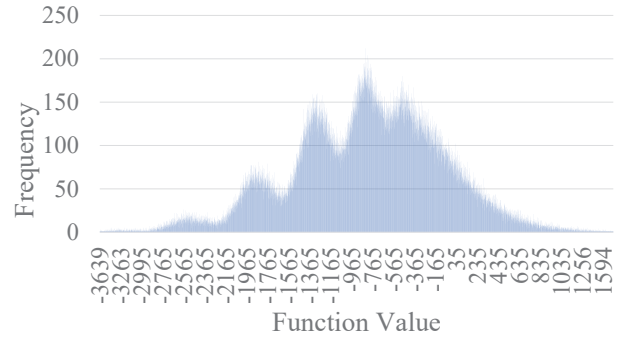
(a) Original randomly generated  $Q$



(b)  $\hat{Q}^1$  (construct by "+/-1")



(c)  $\hat{Q}^2$  (construct by "+/-i")



(d)  $\hat{Q}^3$  (construct randomly)

Fig. 3. Distribution of function values in the solution space of different 18-dimensional UBQPs, frequency versus the function value. (a) Original randomly generated  $Q$ . (b)  $\hat{Q}^1$  (construct by "+/-1"). (c)  $\hat{Q}^2$  (construct by "+/-i"). (d)  $\hat{Q}^3$  (construct randomly).

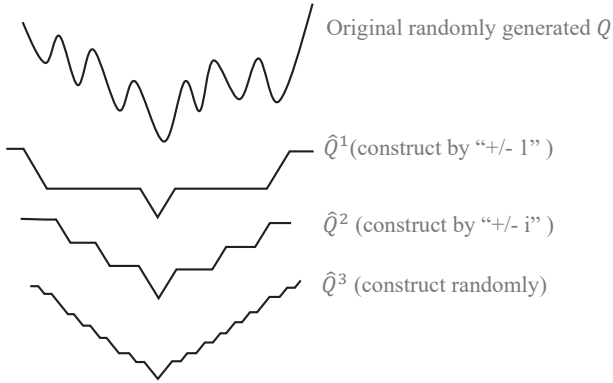


Fig. 4. A diagram of the landscapes of different 18-dimensional UBQPs.

LSILS using the toy UBQP with  $\hat{Q}^3$  (construct randomly) performs worse than ILS. These results compared with ILS confirm the effectiveness of LSILS using the toy UBQP with  $\hat{Q}^1$  (construct by "+/-1") and  $\hat{Q}^2$  (construct by "+/-i") on smoothing the landscape of the original UBQP. In contrast, the toy UBQP with  $\hat{Q}^3$  (construct randomly) has no landscape

smoothing effect and even makes the original UBQP more difficult to solve. From previous landscape flatness analysis, we can find that the toy UBQP with  $\hat{Q}^3$  (construct randomly) possesses a non-flat landscape, with a lower likelihood of generating solutions with identical function values than the original UBQP.

In addition, Fig. 5 shows that among the 10 instances, LSILS using the toy UBQP with  $\hat{Q}^2$  (construct by "+/-i") ultimately achieves better performance than with  $\hat{Q}^1$  (construct by "+/-1"), except for the bqp2500.7 instance. This indicates that the toy UBQP with  $\hat{Q}^2$  (construct by "+/-i") has better landscape smoothing effect than with  $\hat{Q}^1$  (construct by "+/-1"). Based on the landscape flatness analysis mentioned previously, a possible reason is that the toy UBQP with  $\hat{Q}^1$  (construct by "+/-1") has flatter landscape and these flat areas actually have almost no smoothing effect on the original UBQP by HC transformation, thus remaining the rugged landscape of the original UBQP landscape. These results reveal that the smoothing effect of the HC transformation is not only affected by the smoothness of the landscape but also by the flatness of the landscape, therefore, striking a balance between smoothness and flatness of the toy UBQP is crucial for HC

transformation to guide the landscape smoothing algorithms.

In summary, the experimental results highlight that the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) displays the flattest landscape among the three toy UBQPs and the toy UBQP with  $\hat{Q}^3$  (construct randomly) demonstrates the most non-flat landscape while remaining unimodal. This landscape feature indeed makes the LSILS algorithm, which is based on the HC transformation, performs differently for different toy UBQPs. We can not simply assume that the flatter the landscape, the better the smoothing effect. To have a better landscape smoothing effect, choosing a suitable toy UBQP is a key issue.

## V. CONCLUSION

In this study, we extended the HC transformation based landscape smoothing method for UBQP by studying its impact when employing different toy problems: the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”), the toy UBQP with  $\hat{Q}^2$  (construct by “+/-i”), and the toy UBQP with  $\hat{Q}^3$  (construct randomly). Firstly, landscape analysis experiments were conducted to assess the flatness of the three toy UBQPs. The results demonstrate that the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) has the flattest landscape, while the toy UBQP with  $\hat{Q}^3$  (construct randomly) exhibits the most non-flat landscape among the three. Secondly, the LSILS algorithm was applied to 10 UBQP benchmark instances utilizing each of the aforementioned toy UBQPs. Our findings show a different performance of LSILS with the three toy UBQPs, where LSILS using the toy UBQP with  $\hat{Q}^1$  (construct by “+/-1”) and with  $\hat{Q}^2$  (construct by “+/-i”) both having beneficial smoothing effects on the original UBQP. Among the two, the latter consistently outperforms the former. However, LSILS using the toy UBQP with  $\hat{Q}^3$  (construct randomly) even has a poorer performance than ILS. This reinforces the notion that different toy UBQPs exercise varying landscape smoothing effects on the original UBQP. The smoothness and flatness are both important landscape characteristics of the toy problem, emphasizing the important role of selecting an appropriate toy UBQP in enhancing the performance of LSILS. In conclusion, our study contributes valuable insights into the area of landscape smoothing through the utilization of the HC transformation with different toy problems. The observed variations in landscape flatness among the toy UBQPs directly influence the effectiveness of LSILS. In the future, we plan to investigate deep learning (DL) models in order to smooth difficult optimization problems whose solution spaces are rugged.

## ACKNOWLEDGMENT

The work described in this paper was supported in part by Sichuan Science and Technology Program (No. 2022NS-FSC1907), in part by National Natural Science Foundation of China (grant no. 62076197, 61903294), in part by Key Research and Development Project of Shaanxi Province (grant no. 2022GXLH-01-15), in part by Research Grants Council of the Hong Kong Special Administrative Region, China (GRF Project No. CityU11215622), in part by Guangdong Basic and

Applied Basic Research Foundation (No.2021A1515110077) and in part by Fundamental Research Funds for Central Universities (No.D5000210692).

## REFERENCES

- [1] G. Kochenberger et al., “The unconstrained binary quadratic programming problem: a survey,” *J Comb Optim*, vol. 28, no. 1, pp. 58–81, Jul. 2014.
- [2] R. D. McBride and J. S. Yormark, “An Implicit Enumeration Algorithm for Quadratic Integer Programming,” *Management Science*, vol. 26, no. 3, pp. 282–296, Mar. 1980.
- [3] A. T. Phillips and J. B. Rosen, “A quadratic assignment formulation of the molecular conformation problem,” *J Glob Optim*, vol. 4, no. 2, pp. 229–241, Mar. 1994.
- [4] G. Gallo, P. L. Hammer, and B. Simeone, “Quadratic knapsack problems,” *Combinatorial Optimization*, pp. 132–149, 1980.
- [5] A. Liefvooghe, S. Verel, and J.-K. Hao, “A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming,” *Applied Soft Computing*, vol. 16, pp. 10–19, Mar. 2014.
- [6] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, “The Maximum Clique Problem,” in *Handbook of Combinatorial Optimization: Supplement Volume A*, D.-Z. Du and P. M. Pardalos, Eds., Boston, MA: Springer US, 1999, pp. 1–74.
- [7] M. Lewis, G. Kochenberger, and B. Alidaee, “A new modeling and solution approach for the set-partitioning problem,” *Computers & Operations Research*, vol. 35, no. 3, pp. 807–813, Mar. 2008.
- [8] G. A. Kochenberger, F. Glover, B. Alidaee, and C. Rego, “A unified modeling and solution framework for combinatorial optimization problems,” *OR Spectrum*, vol. 26, no. 2, pp. 237–250, Mar. 2004.
- [9] M. R. Garey, and D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” W. H. Freeman & Co Ltd, 1979.
- [10] W. Wang, J. Shi, J. Sun, A. Liefvooghe, and Q. Zhang, “A New Parallel Cooperative Landscape Smoothing Algorithm and Its Applications on TSP and UBQP,” *arXiv e-prints* arXiv:2401.03237, 2024.
- [11] H. R. Lourenço, O. C. Martin, and T. Stützle, “Iterated Local Search: Framework and Applications,” in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., in International Series in Operations Research & Management Science. , Cham: Springer International Publishing, 2019, pp. 129–168.
- [12] F. Glover, Z. Lü, and J.-K. Hao, “Diversification-driven tabu search for unconstrained binary quadratic problems,” *4OR-Q J Oper Res*, vol. 8, no. 3, pp. 239–253, Oct. 2010.
- [13] Z. Lü, F. Glover, and J.-K. Hao, “A hybrid metaheuristic approach to solving the UBQP problem,” *European Journal of Operational Research*, vol. 207, no. 3, pp. 1254–1262, Dec. 2010.
- [14] T. M. Alkhamis, M. Hasan, and M. A. Ahmed, “Simulated annealing for the unconstrained quadratic pseudo-Boolean function,” *European Journal of Operational Research*, vol. 108, no. 3, pp. 641–652, Aug. 1998.
- [15] K. Katayama and H. Narihisa, “Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem,” *European Journal of Operational Research*, vol. 134, no. 1, pp. 103–119, Oct. 2001.
- [16] K. Katayama and H. Narihisa, “On fundamental design of parthenogenetic algorithm for the binary quadratic programming problem,” in *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, pp. 356–363.
- [17] A. Lodi, K. Allemand, and T. M. Liebling, “An evolutionary heuristic for quadratic 0–1 programming,” *European Journal of Operational Research*, vol. 119, no. 3, pp. 662–670, Dec. 1999.
- [18] K. M. Malan, “A survey of advances in landscape analysis for optimisation,” *Algorithms*, vol. 14, no. 2, pp. 40, 2021.
- [19] S. Tari, M. Basseur, and A. Goëffon, “Sampled Walk and Binary Fitness Landscapes Exploration,” in *Artificial Evolution*, E. Lutton, P. Legendre, P. Parrend, N. Monmarché, and M. Schoenauer, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 47–57.
- [20] S. Tari, M. Basseur, and A. Goëffon, “Worst Improvement Based Iterated Local Search,” in *Evolutionary Computation in Combinatorial Optimization*, A. Liefvooghe and M. López-Ibáñez, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 50–66.



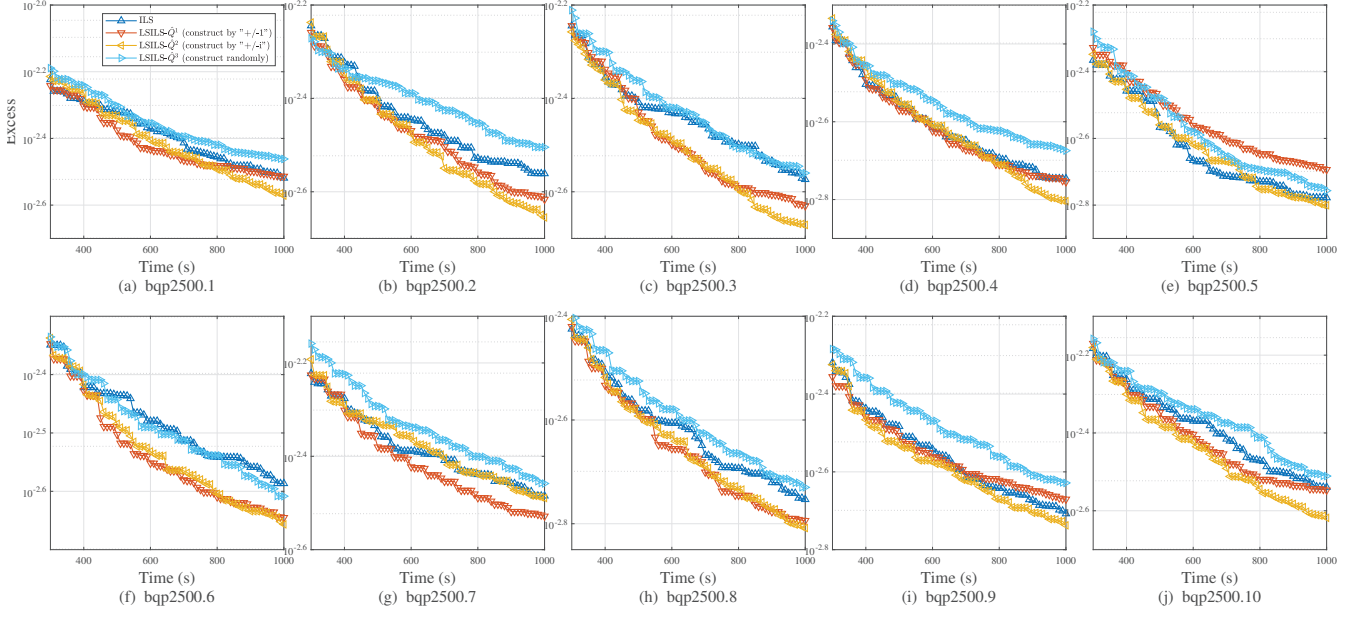


Fig. 5. Comparison results of ILS, LSILS with different toy UBQPs. (a) bqp2500.1. (b) bqp2500.2. (c) bqp2500.3. (d) bqp2500.4. (e) bqp2500.5. (f) bqp2500.6. (g) bqp2500.7. (h) bqp2500.8. (i) bqp2500.9. (j) bqp2500.10.

- [21] P. Merz and K. Katayama, "Memetic algorithms for the unconstrained binary quadratic programming problem," *Biosystems*, vol. 78, no. 1, pp. 99–118, Dec. 2004.
- [22] S. Tari and G. Ochoa, "Local search pivoting rules and the landscape global structure," in *Proceedings of the Genetic and Evolutionary Computation Conference, in GECCO '21*, Jun. 2021, pp. 278–286.
- [23] G. Ochoa, N. Veerapen, F. Daolio, and M. Tomassini, "Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study," in *Evolutionary Computation in Combinatorial Optimization*, B. Hu and M. López-Ibáñez, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 233–248.
- [24] J. Gu and X. Huang, "Efficient local search with search space smoothing: a case study of the traveling salesman problem (TSP)," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 5, pp. 728–735, May 1994.
- [25] K. H. Liang, X. Yao, and C. Newton, "Combining landscape approximation and local search in global optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Washington, DC, USA: IEEE, 1999, pp. 1514–1520.
- [26] S. P. Coy, B. L. Golden, and E. A. Wasil, "A computational study of smoothing heuristics for the traveling salesman problem," *European Journal of Operational Research*, vol. 124, no. 1, pp. 15–27, Jul. 2000.
- [27] M. Hasegawa and K. Hiramatsu, "Mutually beneficial relationship in optimization between search-space smoothing and stochastic search," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 19, pp. 4491–4501, Oct. 2013.
- [28] J. Shi, J. Sun, Q. Zhang, and K. Ye, "Homotopic Convex Transformation: A New Landscape Smoothing Method for the Traveling Salesman Problem," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 495–507, Jan. 2022.
- [29] J. E. Beasley, "Obtaining test problems via Internet," *J Glob Optim*, vol. 8, no. 4, pp. 429–433, Jun. 1996.