Research Article

Optimum Design of Pile Foundation by Automatic Grouping Genetic Algorithms

Xiaofeng Liu,[1] Gengdong Cheng,[2] Bo Wang,[2] and Shuzhi Lin[3]

[1] *Goldwind Science & Technology Company Ltd, Beijing 100176, China*
[2] *State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics,*
  Dalian University of Technology, Dalian 116024, China
[3] *Xiamen Construction and Administration Bureau, Xiamen 361003, China*

Correspondence should be addressed to Xiaofeng Liu, xfl_999@sina.com

This paper studies the optimum conceptual design of pile foundations at the initial design stage. A modular method is proposed, which divides the foundation into modules and each module is identified by its characteristics of pile length, diameter, number and layout. Modules with the same characteristics may be packed and represented by a design variable. A minimum-cost optimization model with multiple design constraints based on Chinese code and a cardinality constraint is built to achieve the concurrent optimization of pile size and layout. The model is solved by the improved automatic grouping genetic algorithms to obtain the design with optimal variables and optimal variable grouping. A practical example demonstrates the effectiveness of the proposed approach.

1. Introduction

Pile foundations widely used in high-rise buildings often arrange identical piles on a regular grid pattern with constant spacing between them. Such design is very conservative and uneconomical. Several design strategies for pile foundations [1–3] are presented to achieve an economic design.

Pile design optimization may be defined as minimum cost of the foundation, while maintaining satisfactory performance. Comparing with the wide study and application of optimization technique in structural engineering domain, the development of the optimization of pile foundations is relatively late for three main difficulties. Firstly, accurate performance prediction of pile foundation is almost impossible because of the uncertainty of soil parameters, the complexity of pile-soil-raft interaction, and the inaccurate constitutive law of layered soil. Even with many available studies based on the elastic-plastic theory [4–6], the nonlinear analysis needs various simplifications and assumptions which may not fit the real situation. As Poulos pointed out, "engineering theory should be based initially on the experience and extended or

modified in the light of further experience" [7], the theoretical analysis results of pile foundations should be modified according to experience in practical design. Secondly, for the discrete nature of pile characteristics (number, diameter, and length), pile optimization is a discrete problem. In addition, the objective function and constraint conditions may be discontinuous, nondifferentiable or even difficultly expressed mathematically in terms of design variables [8]. As a result, pile optimization must be solved by an effective method. Thirdly, piles of a practical design must be grouped because designs with too many different piles will significantly increase the construction and management cost. The experience-based predefined grouping configuration of piles leads to a different optimization problem with a potentially substantially different optimal solution [9]. Therefore, the pile characteristics and piles grouping should be optimized simultaneously. The grouping optimization is discrete, and should be solved by the discrete optimization methods.

Some researches [10–15] introduced the concept and theory of structural optimization into the pile design process, and employed the gradient-based methods with prerequisite

for differentiability and continuity of constraints/objective to solve the optimization problem. In addition, other efforts [8, 16] have been done based on genetic algorithms (GAs) which have no prerequisite for differentiability and continuity.

This paper studies the pile optimization problem at the initial design stage using an improved automatic grouping genetic algorithm (AGGA). Pile characteristics (number, length, and diameter) and pile layout are considered by the proposed modular method to achieve concurrent optimization of pile size and layout. The main contributions of the paper are the proposal of the novel representation for pile design optimization problem based on the modular method, as well as the use of improved AGGA in solving the problem. Minimum-cost pile optimization model with practical design constraints and cardinality constraint is presented in Section 2. The design constraints are evaluated by the Chinese code JGJ 94-2008 [17] that combines both the theoretical researches and engineering experiences, and provides a standard for practical pile foundation design in China. Section 3 applies AGGA with improvements in penalty function and crossover operator to handle the cardinality constraint representing the requirement of pile grouping. Section 4 presents detailed flow on the application of the improved AGGA to the pile foundation optimization. A practical example in Section 5 demonstrates the effectiveness of proposed approach. In the end, some conclusions are discussed.

2. Formulation of Pile Foundation Optimization

The pile foundation optimization can be formulated as

$$\text{find} \quad \mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s\} = \left\{ \left\{ l_1, d_1, n_1, \text{layout}_1 \right\}, \right.$$
$$\left\{ l_2, d_2, n_2, \text{layout}_2 \right\},$$
$$\left. \ldots, \left\{ l_s, d_s, n_s, \text{layout}_s \right\} \right\}, \tag{1}$$

$$\text{minimize} \quad F(\mathbf{X}) = \sum_{i=1}^{s} C(l_i, d_i) \times V(\mathbf{x}_i)$$
$$= \sum_{i=1}^{s} C(l_i, d_i) \times \frac{\pi d_i^2}{4} l_i n_i p_i \tag{2}$$

subject to

$$N_i \le R_i, \quad i = 1, 2, \ldots, s, \tag{3}$$

$$\tilde{N}_i \le \varphi_c f_c A_{psi}, \quad i = 1, 2, \ldots, s, \tag{4}$$

$$s_i \le s_u, \quad i = 1, 2, \ldots, s, \tag{5}$$

$$\frac{|s_{\min} - s_c|}{\Delta l} \le \bar{s}_u, \tag{6}$$

$$\left| c_p - c_g \right| \le c_u, \tag{7}$$

$$\sigma_z + \gamma_m z \le f_{az}, \tag{8}$$

$$\sum_{i=2}^{s} H(\mathbf{x}_i) + 1 \le \text{Ca}, \quad H(\mathbf{x}_i) = \begin{cases} 0, & \text{if } \mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\} \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

The physical meanings of (1)–(9) are given in the following sections.

2.1. Design Variables. Pile foundation optimization involves the optimization of pile length, diameter, number, and layout. To treat such factors simultaneously, the concepts of module and package are introduced in the present study. Firstly, the pile foundation is divided into some modules with a certain rule. The module characteristics include both pile attributes (number, diameter, and length) and pile layout. Then, based on engineering experience, the modules with same characteristics may be packed, and each package corresponds to a design variable. For example, the pile foundation with a symmetrical superstructure or symmetrical applied loads often uses a symmetrical design. Then, the modules corresponding to the pile foundation may be packed in symmetrical patterns. Referring to two packed patterns suggested by [14], that is, the row variation pattern and the squared variation pattern, the latter is used in this paper. As an illustration, let us consider the pile foundation of a symmetrical frame-corewall structure in Figure 1. The foundation containing 43 modules is only subjected to the vertical loads. The loads acting on the foundation beneath the corewall are distributed uniformly, and those acting on the foundation beneath every column are different from each other. As such, the foundation beneath the corewall may use a symmetrical design and 25 corresponding modules are packed from outside to inside in turn, where the outmost 16 modules belong to the first package, 8 modules in the second outer circle are assigned as the second package, see the shaded part of Figure 1, and one innermost module is the third package. Each frame column on the foundation periphery corresponds to a module which belongs to a package. As a result, all the 43 modules are assigned to 21 packages.

Since the modules in each package share the same characteristics, that is, they have identical pile number, pile length, pile diameter and pile layout, the *i*th package is defined as the *i*th design variable $\mathbf{x}_i$. Here, $\mathbf{x}_i$ is not a scalar but a vector with $\mathbf{x}_i = \{l_i, d_i, n_i, \text{layout}_i\}$. l_i, d_i are the pile length and diameter in the *i*th package. n_i, layout_i refer to the pile number and layout in each module of the *i*th package. Moreover, layout_i is defined graphically. As such, the three packages beneath the corewall may be represented by three design variables $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, respectively. Other 18 packages beneath the frame columns are represented by design variables of $\mathbf{x}_4$–$\mathbf{x}_{21}$, shown as Figure 1. For a large scale pile foundation, many packages, that is, many design variables, included in the design should be further grouped to save the construction cost. The detailed discussion for the grouping method will be given in Section 3.1.
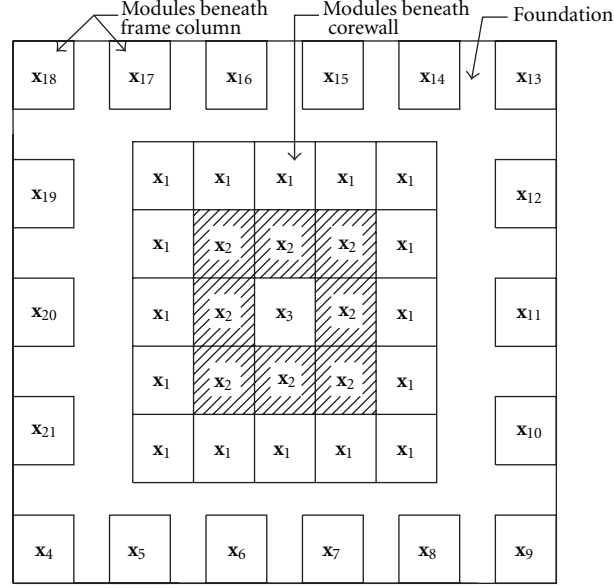
Figure 1: Module division and package.

Table 1: The ranges of pile diameter d for a given module length L in the four pile layouts in Figure 2.

Pile numbers	1	2	3	4
Pile spacing	L	$\sqrt{2}L/2$	$2L/3$	$L/2$
Pile diameter	$L/6 \le d \le L/3$	$\sqrt{2}L/12 \le d \le \sqrt{2}L/6$	$L/9 \le d \le 2L/9$	$L/12 \le d \le L/6$

A module may include many piles, here only the modules with the pile numbers from 1 to 4 are considered. Based on the practice that piles are often arranged in a regular triangle or a square pattern, only four layouts in Figure 2 are considered. Referring to clauses 3.3.3 and 5.5.6 of JGJ 94-2008, the pile spacing D and pile diameter d should meet $3d \le D \le 6d$ to reduce the adverse effect of interaction of pile-to-pile on the pile group's bearing capacity and accommodate the analysis approach. Based on the inequality, the range of d corresponding to four pile layouts are listed in Table 1 for a specified module length L. In addition, due to the limitation of pile-driving equipment and site condition, only discrete values could be assigned to the pile lengths and diameters.

2.2. Objective Function. The total cost of pile foundation includes the installation cost, the material cost, and so on. Factors such as site location (vibration and noise), geotechnical and hydrogeologic characteristics, seismic zone, loading plans and available equipment decide the suitable pile driving method. Each method has a specific installation cost and may have a global limit in terms of pile lengths and/or pile diameters. For example, the continuous flight auger method limits the maximum pile lengths of 20 m and pile diameters of 1.2 m, the bored piles method using stabilizing fluids and temporary or definitive casing limits the pile diameters below 0.5 m, and the bored piles method with unsupported excavation limits the pile diameters below 1.5 m [18]. For simplification's sake, it is assumed in this paper that the

total cost of the foundation $F(\mathbf{X})$ is expressed as (2) where $V(\mathbf{x}_i)$ is the total volume of piles with solid round cross-sections, p_i is the number of modules in the ith package, and s is the total number of packages. The cost per unit volume $C(l_i, d_i)$ is a function of the pile lengths and diameters and may include the cost of driving the piles into the soil in addition to the material cost. $C(l_i, d_i)$ could be defined in discrete form from the local price index or engineering experience. For convenient study, $C(l_i, d_i)$ is defined as a constant in this paper. However, more complex function even in discrete form could be easily accommodated with the present method.

2.3. Constraint Conditions. As was pointed out in [19], pile foundation design should follow five important issues. For a conceptual design at the initial design stage, we mainly consider four of them, which involve the vertical ultimate bearing capacity of pile (3), the vertical load for the structural design of pile (4), and the maximum and differential settlement (5) and (6). The ultimate bearing capacity of pile for lateral and moment loads, the shear and moment for the structural design of raft are not taken into account to simplify analysis. In addition, the eccentricity between the center of gravity of superstructure and the stiffness center of piles is considered (7). So does the checking of softer compressible strata lying beneath the pile base (8) and a cardinality constraint (9) which describes the grouping of design variables.

In (3), N_i is the vertical load with a nominal combination applying in a single pile of the ith package. Here we assume
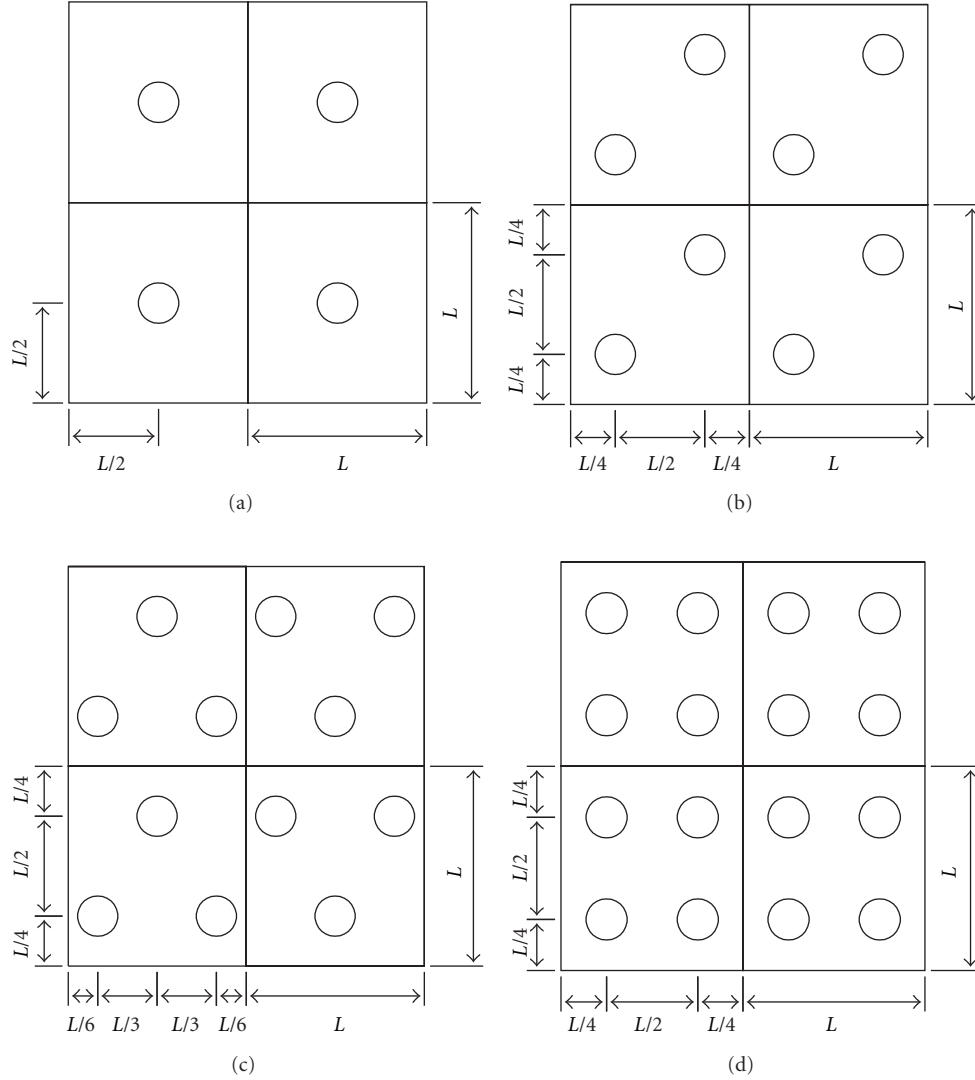
Figure 2: Pile layouts for different pile numbers in a module. (a) 1 pile; (b) 2 piles; (c) 3 piles; (d) 4 piles.

that piles within a package have equal vertical load. R_i is the characteristic value of the vertical ultimate bearing capacity of a single pile in the ith package. For a piled raft foundation, the bearing capacity contribution of raft is considered according to clause 5.2.5 of JGJ 94-2008. Such contribution is the product of the ultimate bearing capacity of raft and a reduction factor which describes the effect of pile-soil-raft interaction.

In (4), $\tilde{N}_i$ is the vertical load with a fundamental combination applying in a single pile of the ith package, and piles within a package have the same vertical load. φ_c is a reliability coefficient that reflects the effect of pile installation methods on the pile strength. f_c is the axial compressive strength of concrete, A_{psi} is the cross-sectional area of one pile in the ith package.

In (5), s_i is the settlement of the ith package, and s_u is the upper bound of settlement. s_c in (6) is the maximum settlement of pile foundation. For a frame-corewall structure, the pile group beneath the corewall carries most applied loads,

and the maximum settlement s_c appears at the pile group center, which may be evaluated by the following:

$$s_c = 4\varphi\varphi_e p_0 \sum_{i=1}^{n_s} \frac{z_i\alpha_i - z_{i-1}\alpha_{i-1}}{E_{si}}. \tag{10}$$

The above equation is different from the equivalent pier method [7], where the pile group is replaced by a pier containing the piles and soil between them. Instead of treating the pile group as a deep foundation to compute the settlement, (10) evaluates the settlement on the basis of the Mindlin equation [20] and uses an equivalent settlement coefficient φ_e, which represents the ratio of the pile group's settlement based on Mindlin equation [20] to that based on equivalent pier method. φ_e considers the effect of pile-soil-raft interaction on the settlement of pile foundation. Its value depends on the number of piles, the ratio of pile diameter to pile spacing, pile diameter to pile length, and raft length to raft width, and can be directly obtained by JGJ 94-2008

for easy use. To reduce the error between practical settlement and computed one based on elasticity theory, an experience coefficient φ is used in (10), which describes the ratio of the measured settlement to computed settlement with respect to various soils. Furthermore, p_0 is the average value of additional pressure of the raft with a quasi-permanent load combination. For other symbols, z_i is the distance from pile base level to the bottom of the ith soil layer. α_i obtained from JGJ 94-2008 is the coefficient of average additional stress corresponding to the ith soil layer. E_{si} is the confined compression modulus of the ith soil layer, n_s is the total number of soil layers used to assess the settlement. As other settlement evaluation methods of pile group, for example, the interaction factor method [21] or the analytical approach [22], (10) only allows the analyses of piles with equal length. Therefore, piles in a pile group should have a same length in the optimization process.

The peripheral frame carries a little proportion of applied loads. The settlement s_f of piles below the columns is evaluated by (11) which is based on Mindlin equation [20] and used to compute the settlement of a single pile by JGJ 94-2008. In (11), the pile is acted upon by a system of uniform vertical shear stresses around the periphery and uniform vertical stresses at the base. s_f comprises three parts: the compressive deformation of subsoil under the pile base which is caused by the pressure of raft; the compressive deformation of subsoil under the pile base which is caused by this pile and other piles within a range of 0.6 times pile length; the compressive deformation of pile self s_e. In (11), the additional stress σ_{zci} caused by raft and the sum σ_{zi} of additional stresses caused by active piles in the middle of the ith soil layer correspond to the first and second part, respectively. In addition, Δz_i is the thickness of the ith soil layer. φ, n_s, and E_{si} have the same meanings as (10):

$$s_f = \varphi \sum_{i=1}^{n_s} \frac{\sigma_{zi} + \sigma_{zci}}{E_{si}} \Delta z_i + s_e. \tag{11}$$

In this paper, we assume that the settlement of a module beneath the frame is equal to the maximum pile settlement in the module. The modules corresponding to the pile group beneath the corewall are separated from other modules below the frame. Also, the modules below the frame are separated from each other. As a result, every module may have an independent settlement, and the differential settlement is evaluated referring to the minimum settlement of peripheral modules under the frame ($s_{\min}$) and the settlement of the center of pile group (s_c). In (6), Δl is the distance between the center of the peripheral module with the minimum settlement and the center of pile group. $\bar{s}_u$ is the upper bound of the scaled differential settlement. It should be noticed that the practical differential settlement may be smaller than the result from (6) due to the neglect of module connections and superstructure stiffness.

In (7), c_g, c_p are the gravity center of superstructure and the stiffness center of piles. c_u is the upper bound of eccentricity. This constraint limits the load eccentricity.

Equation (8) checks the bearing capacity of the softer compressible strata beneath the pile base according to JGJ 94-2008. σ_z is the additional stress acting on the top of the softer strata. γ_m, z are the average weight and thickness of the soil layers overlying the softer strata. f_{az} is the characteristic value of the bearing capacity of the softer strata.

Equation (9) is a cardinality constraint, in which Ca is the upper bound of the grouping and may be determined by the engineer. Modules in the packages with the same group index are identical, that is, they have the same pile length, pile diameter, pile number, and pile layout.

3. Automatic Grouping Genetic Algorithms (AGGA) and Two Improvements

GAs are a stochastic search procedure based on the mechanics of Darwin's evolutionary theory of survival of the fittest and natural genetics. GAs simulate the evolutionary process of living organisms, and start with an encoding operator that encodes an individual, that is, a design point in the search space, as a chromosomal string. Then, a fitness function relating to objective function is defined to describe the individual quality that is a measure of adaption to environment. Finally, the genetic operators (e.g., reproduction, crossover, and mutation in classical GAs) are executed to generate successive generations. The critical parameters in GAs are population sizes, chromosome strings lengths, and probability parameters of genetic operators, see [23]. Unlike other methods working from a single point, GAs work from a population, where the optimization process simultaneously updates a set of points. Response characteristics from various parts of the search space are considered in the update scheme, thereby improving the probability of locating a global optimum. Furthermore, GAs are not limited in search space assumptions such as continuity, convexity, existence of derivatives, and unimodality, which makes it robust for solving discrete engineering problems [24].

To meet the cardinality constraint of (9), the grouping may be predefined based on the engineer's experience. For the quality of grouping that affects final results directly, a trivial grouping will result in the solution which may be far from the optimal solution. An ideal solution would be to provide the designer with the possibility of deciding only the upper bound of grouping Ca, and let the optimization program search also for the optimal grouping configuration in addition to the optimal values of design variables. As such, the optimization model of (1)–(9) may be described as: for a given Ca, we optimize both design variable values and design variable grouping. This problem is discrete in nature and can be solved by AGGA [25]. Section 3.1 details the encoding method of AGGA. Section 3.2 briefly describes the fitness function. Since the standard reproduction with elitism preservation and the mutation operators are adopted, see [23], they are not detailed here. To increase the computational efficiency, some improvements involving the adaptive penalty function in Section 3.2 and a crossover operator in Section 3.3 are made. The effectiveness of both improvements is illustrated by some examples in Section 3.4.

3.1. Encoding. Let us store the available module characteristics in a table. To solve the optimization problem in the
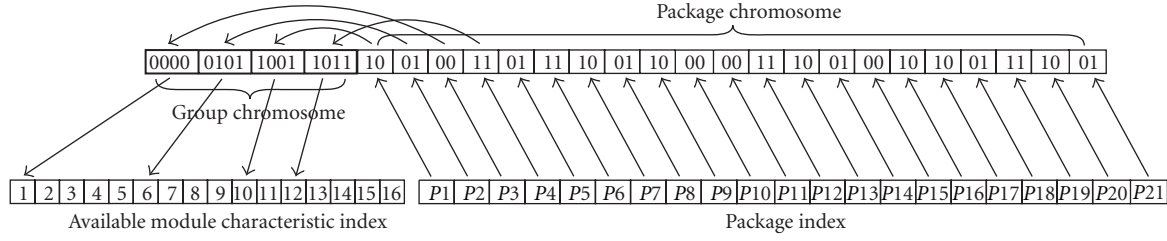
Figure 3: The encoding rule of AGGA.

Table 2: Available module characteristics.

Module number	1	2	3	4	5	6	7	8
Pile number	1	2	3	4	1	2	3	4
Pile length (m)	10	10	10	10	10	10	10	10
Pile diameter (m)	1.0	1.0	1.0	1.0	1.5	1.5	1.5	1.5
Pile layout								
Module number	9	10	11	12	13	14	15	16
Pile number	1	2	3	4	1	2	3	4
Pile length (m)	15	15	15	15	15	15	15	15
Pile diameter (m)	1.0	1.0	1.0	1.0	1.5	1.5	1.5	1.5
Pile layout								

above paragraph, a design of pile foundation is described by the package variables $pack(1: Nm)$ and the group variables $group(1: Ca)$, where Nm is the number of packages. Both variables are integers and play the role of pointer. The package variable $pack(i)$ of the ith package is a pointer to one of the Ca groups. The group variable $group(j)$ of the jth group is a pointer to one module characteristic allowed to be chosen from the table of available module characteristics. As such, the module characteristic of the ith package can be obtained by $characteristic(i) = table(group(pack(i)))$. It should be noted that the module characteristic including the pile number, length and diameter is a vector here. Corresponding to group and package variables, the chromosome of each individual in AGGA comprises two sections: group and package chromosome. As a binary code is used, for group chromosome, the string length of every group variable is controlled by the upper bound of available modules. For package chromosome, the string length of every package variable is controlled by the upper bound of groups. Both chromosomes are involved in genetic operation.

The encoding method of AGGA is illustrated by a 21-package pile foundation example in Figure 1. There are 16 available modules in this example, the characteristics of which are listed in Table 2. The upper bound of grouping is 4. Then the optimization problem has 4 group variables and 21 package variables. Based on the 16 available modules, each group variable is encoded by a four-bit binary string, shown in Figure 3 as the *Group chromosome*. As the upper bound

of grouping is only four, each package variable is encoded by a two-bit binary string, shown in Figure 3 as the *Package chromosome*.

Figure 3 and Table 2 together represent a possible design of 21-package pile foundation, which meets the cardinality constraint with the upper bound Ca = 4. The binary package variable points to the group index. Similarly, the group variable points to the module characteristic in Table 2. For example, the 1st string 10 of package chromosome in Figure 3 with a decoded value of 2 belongs to the 3rd group, that is, $pack(1) = 3$ (here, string 00 with a decoded value of 0 points to the 1st group). Furthermore, the 3rd string 1001 of group chromosome with a decoded value of 9 points to the 10th available module characteristic of Table 2, that is, $group(3) = 10$ (here, string 00000 with a decoded value of 0 points to the 1st available module characteristic of Table 2). By $characteristic(1) = table(group(pack(1))) = \{2, 15, 1.0\}$, the 1st package $P1$ has 2 piles of 15 m long and diameter of 1.0 m. Similarly, the 2nd package $P2$ points to the 2nd group and has 2 piles of 10 m length and diameter of 1.5 m, and so on. As a result, one can decode based on one chromosome the module characteristics: the $P1$, $P7$, $P9$, $P13$, $P16$, $P17$, and $P20$ have the module with 2 piles of 15 m length and diameter of 1.0 m. The $P2$, $P5$, $P8$, $P14$, $P18$, and $P21$ have the module with 2 piles of 10 m length and diameter of 1.5 m. The $P3$, $P10$, $P11$, and $P15$ have the module with 1 pile of 10 m length and diameter of 1.0 m. The $P4$, $P6$, $P12$, and $P19$ have the module with 4 piles of 15 m length and diameter of 1.0 m.

3.2. Fitness Function and Improved Adaptive Penalty Function. Fitness measures the goodness of individuals within a population in GAs and provides a basis for the reproduction operation. For unconstrained maximization optimization, the fitness function is often identical or proportional to the objective function. For constrained optimization problems, it needs modification to consider the satisfactory degree of the constraint conditions. The modified fitness function of the optimization problem in (1)–(9) is defined as

$$F(\mathbf{X}_i) = f_{\max}^p - (f(\mathbf{X}_i) + \text{penal}(\mathbf{X}_i)), \tag{12}$$

where $F(\mathbf{X}_i)$, $f(\mathbf{X}_i)$, and $\text{penal}(\mathbf{X}_i)$ are the fitness, objective, and penalty functions of the ith design $\mathbf{X}_i$ in the population, respectively. $f_{\max}^p$ is the largest $f(\mathbf{X}_i) + \text{penal}(\mathbf{X}_i)$ in the current population.

Though many penalty methods have been proposed (see the introduction of [26]), [26] develops an adaptive penalty function

$$\text{penal}(\mathbf{X}_i) = \sum_{j=1}^{m} k_j v_j(\mathbf{X}_i) = \langle f(\mathbf{X}) \rangle \sum_{j=1}^{m} \frac{\overline{v}_j}{\sum_{j=1}^{m} \left[\overline{v}_j \right]^2} v_j(\mathbf{X}_i), \tag{13}$$

where $\langle f(\mathbf{X}) \rangle$ is the average of $f(\mathbf{X}_i)$ over current population, which is defined as (14). $v_j(\mathbf{X}_i)$ by (15) is the violation of the jth constraint of the ith design, $\overline{v}_j$ is the violation of the jth constraint averaged over the current population, k_j is the violation coefficient of the jth constraint that considers the different difficulties of constraints to be satisfied in the current generation, n_p is the size of population, and m is the number of total constraints:

$$\langle f(\mathbf{X}) \rangle = \frac{1}{n_p} \sum_{i=1}^{n_p} f(\mathbf{X}_i), \tag{14}$$

$$v_j(\mathbf{X}_i) = \max \left\{ 0, g_j(\mathbf{X}_i) \right\}, \tag{15}$$

$$\overline{v}_j = \frac{1}{n_p} \sum_{i=1}^{n_p} v_j(\mathbf{X}_i), \tag{16}$$

$$k_j = \langle f(\mathbf{X}) \rangle \frac{\overline{v}_j}{\sum_{j=1}^{m} \left[\overline{v}_j \right]^2}. \tag{17}$$

The proposed fitness function by [26] is

$$F(\mathbf{X}_i) = \begin{cases} f_{\max}^p - f(\mathbf{X}_i), & \text{if } \mathbf{X}_i \text{ is feasible,} \\ f_{\max}^p - \overline{f}(\mathbf{X}_i) - \text{penal}(\mathbf{X}_i), & \text{otherwise,} \end{cases} \tag{18}$$

where

$$\overline{f}(\mathbf{X}_i) = \begin{cases} f(\mathbf{X}_i), & \text{if } f(\mathbf{X}_i) > \langle f(\mathbf{X}) \rangle, \\ \langle f(\mathbf{X}) \rangle, & \text{otherwise.} \end{cases} \tag{19}$$

As is stated in [26], the penalty of (13) has three features, that is, adaptive penalty capability; not requiring any parameter; automatically defining a different penalty coefficient which varies along the run according to the feedback received from the evolutionary process for each constraint. Such features relieve the user from the burden of having to determine sensitive parameters when dealing with every new constrained optimization problem. Numerical comparisons in [26] show that (13) is equal to or more effective than many other penalty methods.

Here we propose an alternative penalty function based on our numerical studies, which also has the three features of (13):

$$\text{Penal}'(\mathbf{X}_i) = \langle f(\mathbf{X}) \rangle \left[\left(1 + \sum_{j=1}^{m} \alpha_j v_j(\mathbf{X}_i) \right)^3 - 1 \right], \tag{20}$$

$$\alpha_j = \frac{\overline{v}_j}{\sum_{j=1}^{m} \overline{v}_j}, \tag{21}$$

where α_j represents the violation percent of the jth constraint over current population. In comparison with (13), the penalty function equation (20) keeps those near-optimum designs with only slight constraint violation in the population and increases the probability of achieving the optimum.

3.3. Crossover. Crossover is a major genetic operator that allows for an exchange of design characteristics among the mating individuals to produce new designs in the optimization process. The standard two-point crossover operator in [25] includes three steps: first, two mating parents called *Parent* 1 and *Parent* 2 are selected from the mating pool at random. Second, a random number between 0 and 1 is generated. Third, if the random number is less than the defined probability of crossover p_c, the crossover is performed. Two sites along the chromosomal string are selected at random between 1 and the string length less 1. And two new strings called *Child* 1 and *Child* 2 are created by swapping characters between the two chosen sites between the parents, shown as Figure 4(a).

With the grouping approach, the group and the package chromosome corresponding to the group and the package variables, respectively, play different roles. A group variable represents all the packages which belong to this group, and strongly influences the objective function. A package variable, by contrast, only represents a specific package, and has relatively small influence on objective function. In the above two-point crossover operator, two sites are randomly chosen along the whole chromosome (a sum of group and package chromosome) and the bits between them are swapped. Obviously, this crossover does not consider the difference between the group and the package chromosomes, and may leave the two sites in either group chromosome segment (Figure 4(b)) or package chromosome segment (Figure 4(c)). To address such limitation, a crossover operator allowing both the group and the package chromosomes to cross concurrently is used. Two pairs of crossover sites are chosen first, which are located in the group and the package chromosomes, respectively. This process differs from the multiple point crossover, for the latter chooses multiple points along the whole chromosome at random so that such points may be only located in the
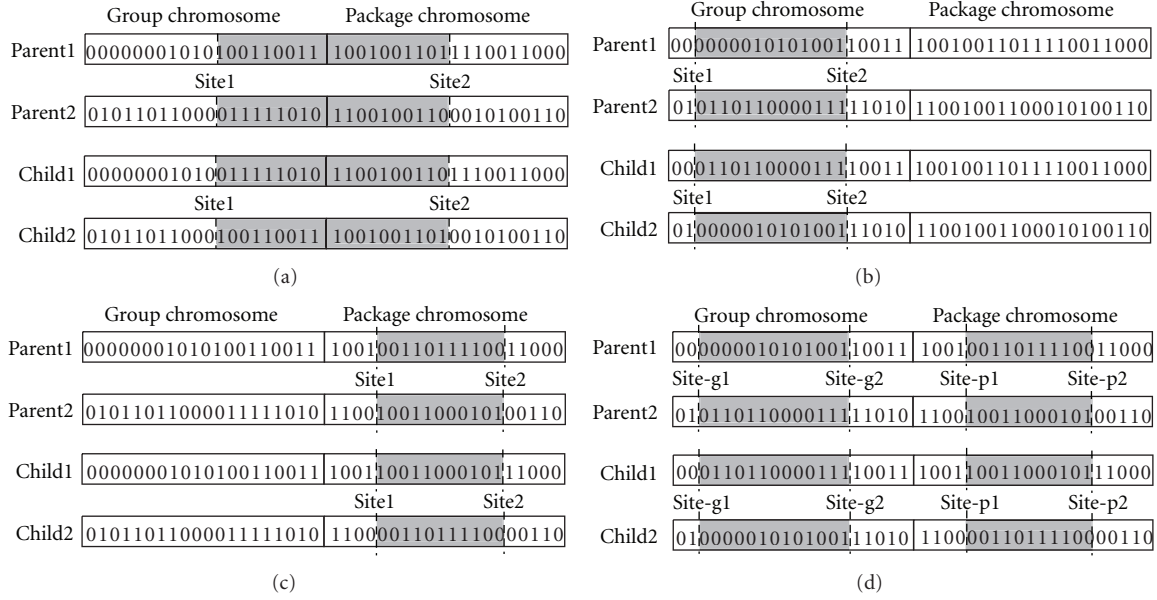
Figure 4: Crossover operation. (a) Standard two-point crossover; (b) two points are located in group chromosome; (c) two points are located in package chromosome; (d) proposed crossover.

group or the package chromosome. Then, the mating parents swap the bits between each pair of sites, shown as Figure 4(d). By the proposed crossover operator, both the group and the package variables are operated so that the search efficiency is improved.

Additionally, diversity within a population is also significant for search efficiency when the population size is fixed. To keep the population diversity, we further modify the crossover operator to produce two offspring different from both their parents and the elitism in their parent population. This modification is achieved by adding two steps, which are the checking of parent pair and child pair, to the whole crossover operation.

In summary, the new crossover operator proceeds in four steps: first, two parent individuals in the mating pool are mated at random, on condition that the pair are different. This step assures the crossover will explore new schema. Second, a random number is generated. Third, if the random number is less than the defined probability of crossover p_c, the crossover is performed. Two pairs of crossover sites are selected at random from group and package chromosome, respectively, and both parent individuals are crossed to create two child individuals in a manner as described above. Lastly, both new created children are checked, and the second step is repeated if any of them is identical to the elitism of parent generation or their parents.

3.4. The Efficiency of Improved AGGA.

For the problems we studied, the proposed penalty function equation (20) together with the proposed crossover operator improves the numerical efficiency of AGGA. For comparison, three benchmark cases from the literature of [27] were tested by different approaches and more test cases comparisons are presented in [28]. In approach 1, original AGGA with the

penalty function equation (13) and the standard two-point crossover operator is used, which is referred to as OAGGA in comparison results. In approach 2 or 3, AGGA only with the improvement in penalty function equation (20) or in proposed crossover operator is used, which is referred to as AGGA-P or AGGA-C. In approach 4, AGGA with the improvements both in penalty function and in crossover operator is used and referred to as IAGGA. All approaches use the same parameters: a population size of 100, a crossover rate of 0.7, and a mutation rate of 0.002. A binary code, rank-based reproduction with elitism preservation and standard mutation in [23] are used.

These benchmark cases have been repeatedly used as a test bed in the evolutionary computation literature. Here an additional cardinality constraint is included in every case and, presented in Table 3. Each variable of the three cases is coded with 20 binary bits. OAGGA, AGGA-C, AGGA-P and IAGGA are used with the maximum number of generations set to 200 and 500. Each approach performs 50 independent runs for each case. The run results are arranged in ascending order and shown in Figure 5.

All results of test cases T_1 by OAGGA, AGGA-C, and IAGGA are feasible solutions. It can be noted that the results from AGGA-C are better than those from OAGGA, especially in the comparison of 200 generations. The results from IAGGA are better than those from AGGA-C. For test case T_2 and, T_3, the results of 50 runs both by OAGGA and by AGGA-C are not presented in Figures 5(b) and 5(c) due to many infeasible solutions in them. However, all the results by AGGA-P and IAGGA are feasible and shown in Figures 5(b) and 5(c). Again, the results from IAGGA are better than those from AGGA-P which does not perform the proposed crossover operator, especially in the comparison of 200 generations.

TABLE 3: Definition of test-cases T_1–T_3.

Objective function	Constraints
$\text{Max}: T_1(\mathbf{x}) = \left\| \dfrac{\sum_{i=1}^{20} \cos^4(x_i) - 2\prod_{i=1}^{20} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{20} i x_i^2}} \right\|$	$\prod_{i=1}^{n} x_i \geq 0.75$ $\sum_{i=1}^{n} x_i \leq 150$ $\sum_{i=2}^{20} H(\mathbf{x}_i) + 1 \leq 4, \qquad H(\mathbf{x}_i) = \begin{cases} 0, & \text{if } \mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\} \\ 1, & \text{otherwise} \end{cases}$ $0 \leq x_i \leq 10, \quad i = 1, \ldots, 20$
$\begin{aligned} \text{Min}: T_2(\mathbf{x}) \ = \ & x_1^2 + x_2^2 + x_1 x_2 - 14x_1 - 16x_2 \\ & + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\ & + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 \\ & + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \end{aligned}$	$105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \geq 0$ $-10x_1 + 8x_2 + 17x_7 - 2x_8 \geq 0$ $8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0$ $3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0$ $-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \geq 0$ $-x_1^2 - 2(x_2 - 2)^2 + 2x_1 x_2 - 14x_5 + 6x_6 \geq 0$ $-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0$ $-0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \geq 0$ $\sum_{i=2}^{10} H(\mathbf{x}_i) + 1 \leq 4, \qquad H(\mathbf{x}_i) = \begin{cases} 0, & \text{if } \mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\} \\ 1, & \text{otherwise} \end{cases}$ $-10 \leq x_i \leq 10, \quad i = 1, \ldots, 10$
$\begin{aligned} \text{Min}: T_3(\mathbf{x}) = \ & 5(x_1 + x_2 + x_3 + x_4) \\ & -5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \end{aligned}$	$2x_1 + 2x_2 + x_{10} + x_{11} \leq 10$ $2x_1 + 2x_3 + x_{10} + x_{12} \leq 10$ $2x_2 + 2x_3 + x_{11} + x_{12} \leq 10$ $-8x_1 + x_{10} \leq 10$ $-8x_2 + x_{11} \leq 10$ $-8x_3 + x_{12} \leq 10$ $-2x_4 - x_5 + x_{10} \leq 0$ $-2x_6 - x_7 + x_{11} \leq 0$ $-2x_8 - x_9 + x_{12} \leq 0$ $\sum_{i=2}^{13} H(\mathbf{x}_i) + 1 \leq 2, \qquad H(\mathbf{x}_i) = \begin{cases} 0, & \text{if } \mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\} \\ 1, & \text{otherwise} \end{cases}$ $0 \leq x_i \leq 1, \quad i = 1, \ldots, 9$ $0 \leq x_i \leq 100, \quad i = 10, 11, 12$ $0 \leq x_{13} \leq 1$

In summary, the proposed penalty function in (20) and crossover operator constitute two effective techniques for this test suite. Both techniques lead to better results than OAGGA, especially when the generation number is not large.

4. Improved AGGA for Design Optimization of Pile Foundations

The optimization procedure of pile foundations for each run can be concluded as follows:

(1) Divide the foundation into modules. The characteristics of a module include the pile number, length, diameter, and layout.

(2) Pack the modules, and each package corresponds to a design variable.

(3) Set the parameter values of AGGA, such as the population size, generation number, crossover, and mutation rate.

(4) Define a set of initial designs which satisfy the cardinality constraint. The initial characteristics of the modules can be randomly selected from the available module library.

(5) Carry out structural computation, that is, the computation of bearing capacity, settlement, differential settlement, eccentricity, and softer compressible strata.

(6) Evaluate every individual's fitness according to the analysis result.

(7) Execute the genetic operators: selection, crossover, and mutation.

(8) Repeat (5)–(7) until the convergence criterion is satisfied.

5. Practical Examples

Two pile foundation examples for buildings in Xiamen, China are optimized to demonstrate the effectiveness of
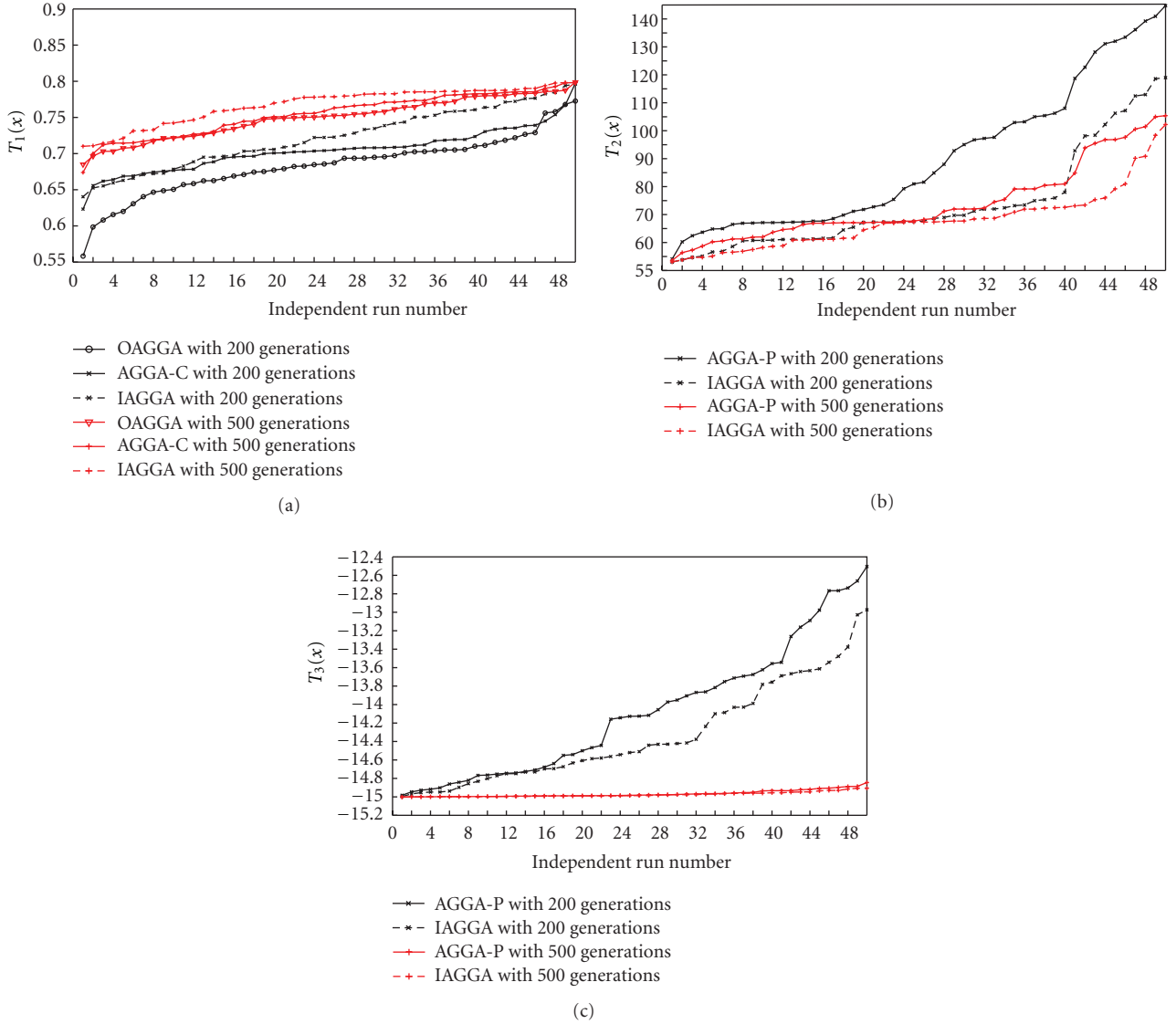
FIGURE 5: Optimization results of three test cases with different approaches. (a) results of T_1; (b) results of T_2; (c) results of T_3.

AGGA for the large-scale practical foundation designs. The first building with a frame-corewall structure system has a square foundation plan, and comprises 3 stories (14.8 m) underground and 50 stories (220 m) overground. The second with a shear-wall structure system has a polygon foundation plan, and comprises 3 stories (9.3 m) underground and 32 stories (100 m) overground. All the structural loads are transferred to foundations by the columns and walls in both examples. Among various load combinations, only the dominating gravity load alone is considered in the study. The bounds of several constraint conditions are set same in both examples, that is, the upper bounds of the maximum settlement s_u and the scaled differential settlement $\bar{s}_u$ are 0.2 m and 0.002 in accordance with JGJ 94-2008; the upper bounds of the eccentric distance c_u are 0.5 m. However, the cardinality constraints Ca are 3 and 2 in the first and second examples. Both examples use the same simulation settings: a population size of 100, a crossover rate of 0.7, and a mutation

rate of 0.002. The iteration terminates when the generation number reaches 200. Two optimized designs are obtained in 5.1 and 5.2, which provide the excellent initial designs for the engineers.

5.1. Pile Foundation Optimization of 50-Story Building. Figure 6 shows the vertical loads acting on every column or wall and the pile layout of the original design. The whole raft with an area of 49.2×49.2 m^2 has a thickness of 4.5 m under the corewall and a thickness of 3.0 m under the peripheral frame. 68 piles with the diameter of 1.2 m and the length of 55 m are located under the corewall, and 60 piles with the diameter of 1.2 m and the length of 45 m are located under the frame. The soil under the raft consists of 7 layers shown as Table 4. Where E_s is the confined compression modulus of soil, f_s, f_b are the ultimate pile shaft frication and ultimate pile end bearing capacity. "Capacity" represents the characteristic value of bearing capacity of soil, that is,
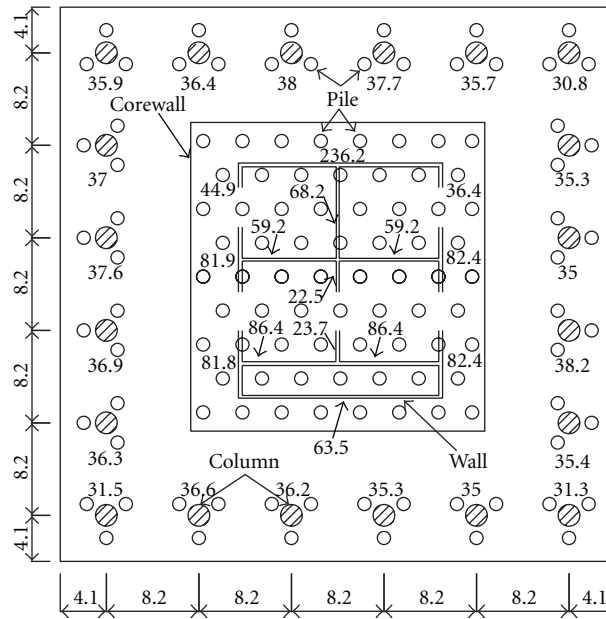
FIGURE 6: Pile layout in original design, and vertical loads acting on columns and walls (m, MN).

the maximum compressive stress permitted acting on soil. According to Table 4, all the 128 piles of the original design have their base location in the 5th soil layer, that is, rock-b.

To perform the optimization design, the foundation is divided into modules, firstly. Considering clause 3.3.3 of JGJ 94-2008 that the piles should be located directly under the columns or walls, here, we arrange the modules as Figure 7. Furthermore, the modules under the corewall are packed with the squared variation pattern, while each module below the column corresponds to a package. For every package that is a design variable, there are two variables underlying the corewall, and twenty variables underlying the columns. Such variables are represented by x_1–x_{22} and shown as Figure 7.

For module characteristics, the pile numbers are from 1 to 4 with an interval of 1, the pile lengths are from 20 to 52 m with an interval of 1 m, and the pile diameters are from 1.0 to 2.0 m with an interval of 0.1 m. Moreover, piles with different numbers are arranged in a module as Figure 2. Referring to Table 1, there are 792 available modules. The piles under the corewall have a same length to meet the requirement of (10). Raft sizes of the original design are used in the optimization process.

For such optimization problem with discrete design variables, discrete available sectional library and grouping constraint, the number of alternative designs which can be estimated by classical probability theory is total more than 2.58×10^{18}. In every run, AGGA obtains an optimum solution by only 2.0×10^4 function evaluations (200 generations, with a population of 100 for each generation) that less than 1.29×10^{-14} of design space is searched. 30 independent runs are executed and each run needs about 30 minutes on a workstation (CPU of AMD quad core 2.6 G and 4 G RAM).

The optimal design among 30 runs with three different modules is shown in Figure 8. 16 modules under the corewall have the same characteristics, that is, every one includes 3
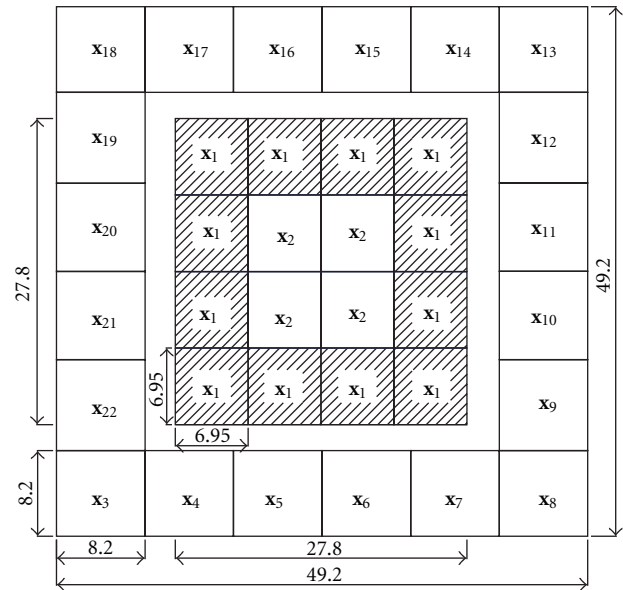

FIGURE 7: Division of modules and assignment of design variables (m).

piles with the diameter of 1.2 m and the length of 52 m. However, 20 modules under the frame are classified as two types: *Frame module* 1 and *Frame module* 2. Each *Frame module* 1 has four piles with the diameter of 1.0 m and the length of 37 m, while each *Frame module* 2 has four piles with the diameter of 1.1 m and the length of 37 m. Those *Frame module* 2 are mainly located in the upper side of the foundation to make the eccentric distance (0.46 m) between the gravity center $G(24.44, 25.34)$ and the stiffness center $K(24.55, 24.89)$ less than the upper bound of 0.5 m. The optimal design has the same number of piles, which is 128, as

TABLE 4: Summary of soil properties under the raft.

Layer	1	2	3	4	5	6	7
Soil	Clay-a	Clay-b	Clay-c	Rock-a	Rock-b	Rock-c	Rock-d
Thickness (m)	3.5	13.0	15.3	4.9	23.6	8.4	5.8
Weight (kN/m^3)	18.3	18.5	19.0	19.5	21.0	22.5	24.0
Capacity (kPa)	530	540	580	650	700	800	2000
E_s (MPa)	12	15	18	34	60	95	—
f_s (kPa)	50	55	60	65	90	140	220
f_b (kPa)	1300	1700	1700	1800	3000	13230	41320

TABLE 5: Summary of soil properties under the raft.

Layer	1	2	3	4	5
Soil	Clay-a	Clay-b	Rock-a	Rock-b	Rock-c
Thickness (m)	3.8	8.4	12.2	33.3	7.3
Weight (kN/m^3)	18.8	18.6	19.5	20.5	23.5
Capacity (kPa)	250	280	400	700	3000
E_s (MPa)	6.5	15	30	70	—
f_s (kPa)	60	70	90	120	800
f_b (kPa)	500	1200	1500	4000	10000

the original one but a different pile layout. Comparing with the original design, the optimal design arranges fewer piles under the corewall and more under the frame. All the pile bases are located in the 5th layer, that is, rock-b. The bearing capacities R of three different piles shown in Figure 8 meet the bearing capacity constraints. The maximum settlement s_c located in the center of the corewall is 0.111 m that is less than the upper bound of 0.2 m. The minimum settlement s_{min} with a value of 0.065 m is located in the upper right of the foundation. The maximum-scaled differential settlement of 0.0016 also is less than the upper bound of 0.002. The total piles volume of the optimal design is 5259 m^3 that saves the concrete of 2021 m^3, which is 27.8% of the original design (7280 m^3).

5.2. Pile Foundation Optimization of 32-Story Building. In the second example, the piled raft foundation with a polygon plan shape shown as Figure 9 is optimized. The raft has a constant thickness of 3.0 m, and is subjected to the actions of vertical loads transmitted via the wall. The soil under the raft consists of five layers presented in Table 5. 30 piles with the same diameter of 1.2 m and the length of 14 m are arranged in the original design as illustrated in Figure 9. All of them have their base locations in the 3rd soil layer, that is, rock-a.

The optimization begins with discretizing the raft plan into 22 modules. 10 shaded modules shown in Figure 10 are assigned to 5 packages and represented by variables x_1–x_5, while each of the other modules is assigned to a package and represented by a design variable. All the 22 modules are considered as a pile group foundation and have the same pile length to consist with the requirement of (10). The differential settlement selects the maximum one occurring between the center and any corner of the raft. For module characteristics, the pile numbers are from 1 to 4 with an interval of 1, the pile lengths are from 10 to 34 m with

an interval of 1 m, and the pile diameters are from 0.9 to 1.7 m with an interval of 0.1 m. Piles layout in a module with different numbers conforms to the criteria described in Figure 2 and Table 1. As a result, there are 390 available modules. Raft sizes of the original design are used in the optimization process. 30 independent runs are executed and each run needs about 25 minutes on a workstation (CPU of AMD quad core 2.6 G and 4 G RAM).

The optimal design among 30 runs has two different modules and is shown in Figure 11. Five modules represented by x_1, x_7, x_9, and x_{11} have the same characteristics, that is, every one includes two piles with the diameter of 0.9 m, and the length of 13 m. However, each of the other 17 modules has one pile with a diameter of 0.9 m and a length of 13 m. The total number of piles decreases from 30 in the original design to 27, the pile diameter from 1.2 m to 0.9 m and the pile length from 14 m to 13 m. As a result, only 223 m^3 of concrete is used and 251.6 m^3 of concrete that is 53.0% of the original 474.7 m^3 is saved. The bearing capacity R of the pile shown in Figure 11 meets the bearing capacity constraints. The maximum settlement s_c located in the center of the foundation is 0.056 m that is less than the upper bound of 0.2 m. The minimum settlement s_{min} with a value of 0.018 m is located in the lower left of the foundation. The maximum scaled differential settlement of 0.00196 also is less than the upper bound of 0.002. The optimal pile layout sites the stiffness center at $K(16.40, 8.61)$ that is 0.079 m distant from the gravity center $G(16.38, 8.53)$ to meet the eccentric constraints.

6. Conclusions

The paper proposed a modular approach to deal with the multiplicity of design variables in the optimization of pile foundation. The improved AGGA is used to handle
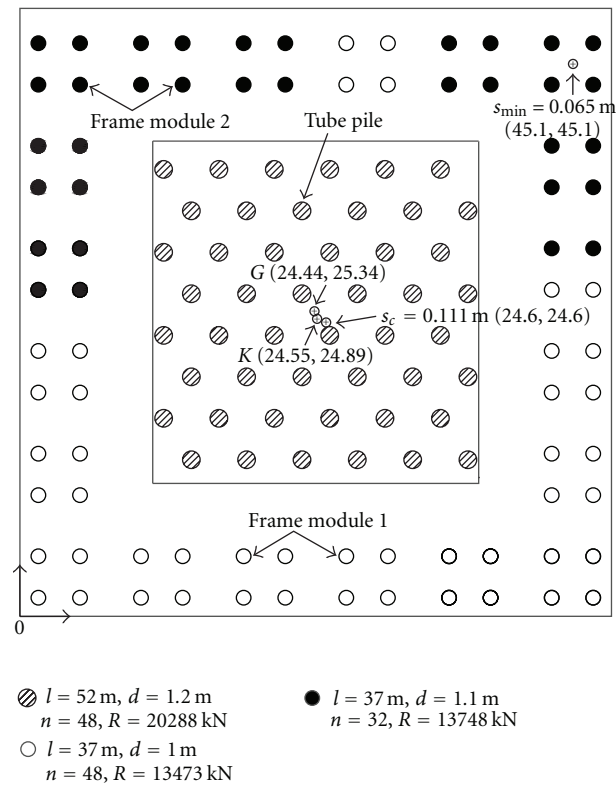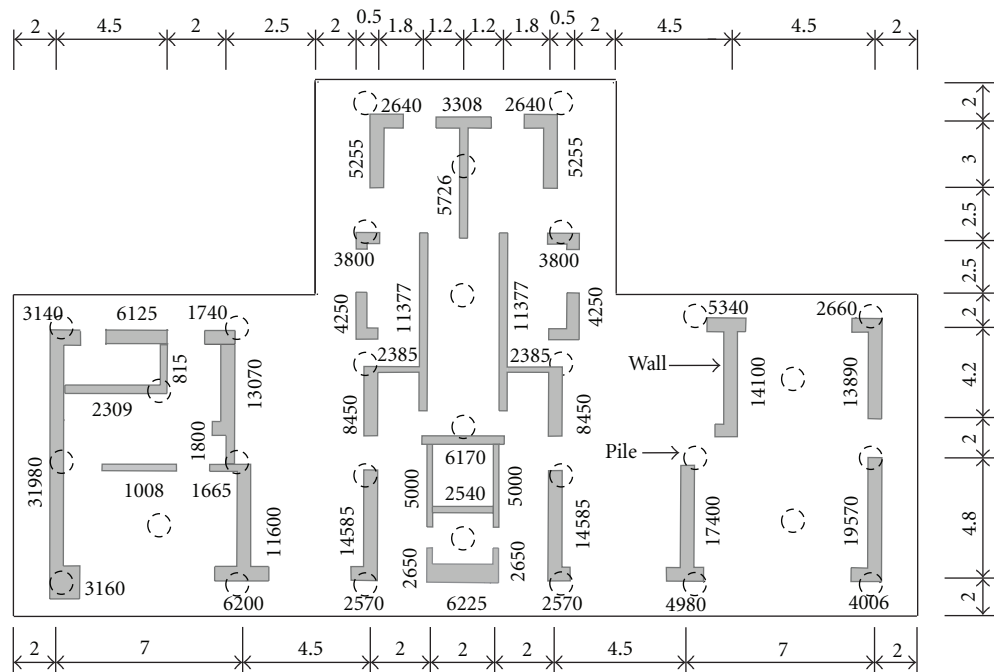
Figure 8: Optimum design among 30 runs.

Figure 9: Pile layout in original design and vertical loads acting on walls (m, kN).
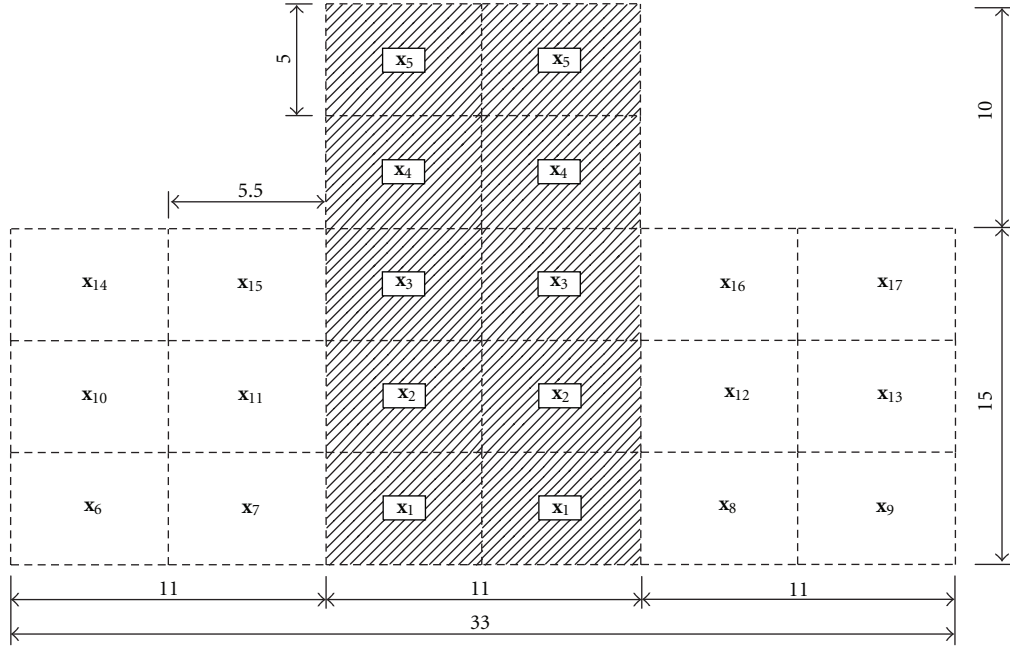
FIGURE 10: Division of modules and assignment of design variables (m).

$s_c = 0.056$ m (16.5, 9.77)

G (16.38, 8.53) K (16.4, 8.61)

$s_{\min} = 0.018$ m
(2.75, 2.5)
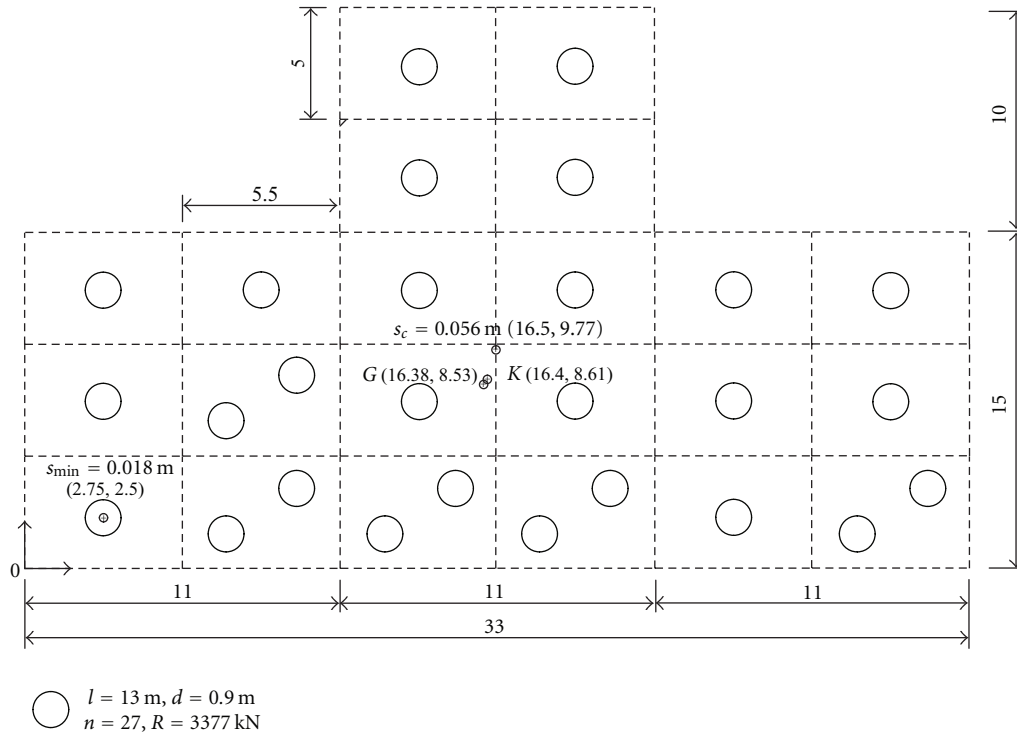
$l = 13$ m, $d = 0.9$ m
$n = 27$, $R = 3377$ kN

FIGURE 11: Optimal design among 30 runs (m).

the discrete optimization problem. Based on the modular method and the improved AGGA, the optimum conceptual design of pile foundations at the initial design stage is studied. The objective is to minimize the cost of pile foundation, and the design requirements of bearing capacity, maximum settlement, differential settlement, eccentricity, and softer compressible layers underlying the pile base are considered with the methods of JGJ 94-2008. Moreover, the requirement of pile grouping also is considered in the proposed optimization model. This study is novel for the pile design optimization problem, because the combination of modular method and improved AGGA achieves

the optimization of pile size, layout and grouping. Some of the conclusions obtained in the study are summarized as follows.

(1) The modular approach handles the pile number, diameter, length, and layout simultaneously to achieve the concurrent optimization of pile size and layout.

(2) The calculation methods specified by JGJ 94-2008 avoid the great computational cost of the numerical methods. Therefore, the large-scale practical optimization of pile foundation is feasible.

(3) The automatic grouping genetic algorithm (AGGA) achieves the optimization of pile foundation and design variable grouping with a remarkably effective search method. The improvements with the crossover and penalty function increase the efficiency of AGGA.

(4) The effectiveness of the proposed approach is illustrated by two practical projects, and the innovative and economical conceptual designs which fulfill all constraints and minimize the cost are obtained.

(5) It should be noted that in practical project a reasonable cost function $C(l_i, d_i)$ might be adopted in order to consider the construction cost.

Acknowledgment

References

[1] R. P. Cunha, H. G. Poulos, and J. C. Small, "Investigation of design alternatives for a pilled raft case history," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 127, no. 8, pp. 635–641, 2001.

[2] K. Horikoshi and M. F. Randolph, "A contribution to optimum design of piled rafts," *Geotechnique*, vol. 48, no. 3, pp. 301–317, 1998.

[3] O. Reul and M. F. Randolph, "Design strategies for piled rafts subjected to nonuniform vertical loading," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 130, no. 1, pp. 1–13, 2004.

[4] E. M. Comodromos, "Response evaluation of axially loaded fixed head pile groups using 3D nonlinear analysis," *Soils and Foundations*, vol. 44, no. 2, pp. 31–39, 2004.

[5] E. M. Comodromos and S. V. Bareka, "Response evaluation of axially loaded fixed-head pile groups in clayey soils," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 33, no. 17, pp. 1839–1865, 2009.

[6] R. Katzenbach, U. Arslan, C. Moormann, and O. Reul, "Piled raft foundation-interaction between piles and raft," *Darmstadt Geotechnics*, vol. 4, pp. 279–296, 1998.

[7] H. G. Poulos and E. H. Davis, *Pile Foundation Analysis and Design*, Wiley, New York, NY, USA, 1980.

[8] C. M. Chan, L. M. Zhang, and J. T. M. Ng, "Optimization of pile groups using hybrid genetic algorithms," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 135, no. 4, pp. 497–505, 2009.

[9] H. J. C. Barbosa and A. C. C. Lemonge, "A genetic algorithm encoding for a class of cardinality constraints," in *Proceedings of the genetic and evolutionary computation conference (GECCO '05)*, H. G. Beyer, Ed., pp. 1193–1120, ACM Press, New York, NY, USA, June 2005.

[10] B. B. Budkowska, "Sensitivity analysis of short piles embedded in homogeneous soil. Part I (theoretical formulation)," *Computers and Geotechnics*, vol. 21, no. 2, pp. 87–101, 1997.

[11] Y. K. Chow and V. Thevendran, "Optimisation of pile groups," *Computers and Geotechnics*, vol. 4, no. 1, pp. 43–58, 1987.

[12] A. J. Hurd and K. Z. Truman, "Optimization method of pile foundations," in *Advances in Engineering Structures, Mechanics and Construction*, M. Pandey, W. C. Xie, and L. Xu, Eds., vol. 140, pp. 653–661, Springer, Amsterdam, The Netherlands, 2006.

[13] K. N. Kim, S. H. Lee, K. S. Kim, C. K. Chung, M. M. Kim, and H. S. Lee, "Optimal pile arrangement for minimizing differential settlements in piled raft foundations," *Computers and Geotechnics*, vol. 28, no. 4, pp. 235–253, 2001.

[14] Y. F. Leung, A. Klar, and K. Soga, "Theoretical study on pile length optimization of pile groups and piled rafts," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 136, no. 2, Article ID 003002QGT, pp. 319–330, 2010.

[15] K. Z. Truman and A. S. Hoback, "Optimization of steel piles under rigid slab foundations using optimality criteria," *Structural Optimization*, vol. 5, no. 1-2, pp. 30–36, 1992.

[16] H. T. Kim, H. K. Yoo, and I. K. Kang, "Genetic algorithm-based optimum design of piled raft foundations with model tests," *Geotechnical Engineering*, vol. 33, no. 1, pp. 1–11, 2002.

[17] Ministry of Construction, *Technical code for building pile foundations*, Ministry of Construction, Beijing, China, 2008.

[18] L. M. Penteado and J. de Brito, "Expert knowledge-based selection methodology for optimizing the construction of concrete piles," *Journal of Perfomance of Constructed Facilities*, vol. 26, no. 1, pp. 95–103, 2012.

[19] H. G. Poulos, "Piled raft foundations: design and applications," *Geotechnique*, vol. 51, no. 2, pp. 95–113, 2001.

[20] R. D. Mindlin, "Force at a point in the interior of a semi-infinite solid," *Journal of Applied Physics*, vol. 7, no. 5, pp. 195–202, 1936.

[21] H. G. Poulos, "Analysis of the settlement of pile groups," *Geotechnique*, vol. 18, no. 4, pp. 449–471, 1968.

[22] M. F. Randolph and C. P. Wroth, "analysis of the vertical deformation of pile groups," *Geotechnique*, vol. 29, no. 4, pp. 423–439, 1979.

[23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, NY, USA, 1989.

[24] J. P. B. Leite and B. H. V. Topping, "Improved genetic operators for structural engineering optimization," *Advances in Engineering Software*, vol. 29, no. 7–9, pp. 529–562, 1998.

[25] H. J. C. Barbosa, A. C. C. Lemonge, and C. C. H. Borges, "A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization," *Engineering Structures*, vol. 30, no. 12, pp. 3708–3723, 2008.

[26] A. C. C. Lemonge and H. J. C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 703–736, 2004.

[27] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evolutionary computation*, vol. 7, no. 1, pp. 19–44, 1999.

[28] X. F. Liu, G. D. Cheng, J. Yan, and L. Jiang, "Singular optimum topology of skeletal structures with frequency constraints by AGGA," *Structural and Multidisciplinary Optimization*, vol. 45, no. 3, pp. 451–466, 2012.