

Research Article

Applying a New Adaptive Genetic Algorithm to Study the Layout of Drilling Equipment in Semisubmersible Drilling Platforms

Wensheng Xiao, Lei Wu, Xue Tian, and Jingli Wang

College of Mechanical and Electronic Engineering, China University of Petroleum (East China), Qingdao 266580, China

Correspondence should be addressed to Lei Wu; wuleiupc@163.com

Received 17 February 2015; Accepted 11 May 2015

Academic Editor: Jason Gu

Copyright © 2015 Wensheng Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study proposes a new selection method called trisection population for genetic algorithm selection operations. In this new algorithm, the highest fitness of $2N/3$ parent individuals is genetically manipulated to reproduce offspring. This selection method ensures a high rate of effective population evolution and overcomes the tendency of population to fall into local optimal solutions. Rastrigin's test function was selected to verify the superiority of the method. Based on characteristics of arc tangent function, a genetic algorithm crossover and mutation probability adaptive methods were proposed. This allows individuals close to the average fitness to be operated with a greater probability of crossover and mutation, while individuals close to the maximum fitness are not easily destroyed. This study also analyzed the equipment layout constraints and objective functions of deep-water semisubmersible drilling platforms. The improved genetic algorithm was used to solve the layout plan. Optimization results demonstrate the effectiveness of the improved algorithm and the fit of layout plans.

1. Introduction

GA (genetic algorithms) [1] are a type of optimization method that is based on biological evolution. Due to their strong robustness and global optimization ability, GA have been widely used in artificial intelligence, engineering optimization, machine learning, expert system, and many other fields [2–4]. However, the efficiency of standard genetic algorithm computation is low, and it is easy to fall into local optimal solutions. Many researchers have proposed strategies to improve these algorithms. These studies have investigated the design of encoding genetic algorithms [5, 6], genetic operations improvement [7, 8], combination with other optimization algorithms [9–11], and so forth.

Many studies have focused on layout problems. Such problems arise in a variety of situations, including pallet loading, container stuffing, and placement problems [12]. The drilling equipment layout of semisubmersible platforms represents a typical complex system layout problem, in which layout scheme significantly affects platform performance.

In recent years, genetic algorithms have been shown to be particularly effective for solving NP-hard layout problems. In this study, the selection operator of GA was improved

based on existing genetic algorithm. This study proposed that a genetic algorithm crossover and mutation probability adaptive methods are based on the characteristics of arc tangent function. An improved GA was used to solve the layout problem of drilling equipment for deep-water, semisubmersible platforms. Calculation results confirm that the improved algorithm is quite efficient for solving the NP-hard layout problem.

In standard GA, offspring is generated by parent individuals through the selection operator, crossover operator, and mutation operator. The parent-offspring competition mechanism (POGA) is often used in GA to produce excellent offspring [13]. However, all parent individuals participate in the generation procedure of offspring. Therefore, this is not useful for excellent individuals' evolution at rapid speed. The elitist genetic algorithm (EGA) was introduced [14, 15]. In EGA, the best offspring remains in the next generation without being destroyed. However, the EGA leads to locally optimal solutions with a higher possibility. Other common selection operators include area selection, group selection, selection based on population gender, and selection based on model extraction and compensation.

The remainder of the paper is organized as follows. Section 2 describes the improvement of the genetic operator. Section 3 describes improvement of adaptive crossover operator and mutation operator. Section 4 describes convergence performance of T-AAGA and simulation testing results are listed. Drilling equipment layout of semisubmersible drilling platforms is introduced in Sections 5 and 6 that cover conclusions.

2. Improvement of Genetic Operator

This work proposes the trisection population method selection operator (TGA). Excellent two-thirds parent individuals were selected to participate in the genetic operation and the lower fitness one-third offspring individuals were eliminated. The remaining individuals of parent and offspring generation constituted the next generation of population. This selection mechanism ensures that excellent individual genes are stable through next generation. It also guarantees the diversity of the species gene pool and prevents algorithms from becoming stuck in local optimal solution.

The specific process of trisection population method selection mechanism is as follows.

Let us define the population of generations:

$$S(a) = (S_{a1}, S_{a2}, \dots, S_{aN}), \quad (1)$$

where a is the number of generations and N is the number of individuals. The fitness of S_{a1} is expressed by $\overline{S_{a1}}$.

To ensure the superiority of offspring, the parent individuals whose fitness is higher were selected for operating to generate new individuals. Based on calculation of the fitness function, the fitness of individuals is ranked from highest to lowest. Rank results are expressed by $S'(a)$

$$S'(a) = (S'_{a1}, S'_{a2}, \dots, S'_{aN}). \quad (2)$$

In the formula (2), $\overline{S'_{a1}} \geq \overline{S'_{a2}} \geq \dots \geq \overline{S'_{aN}}$.
Let us define M

$$M = \text{int}\left(\frac{N}{3}\right), \quad (3)$$

where $\text{int}()$ is the integral function.

Then the M individuals with lower fitness would be eliminated:

$$S''(a) = (S'_{a1}, S'_{a2}, \dots, S'_{a(N-M)}). \quad (4)$$

Individual species $S'''(a)$ are produced by $S''(a)$ through the selection operator, crossover operator, mutation operators, and other genetic forms of manipulation. The number of $S'''(a)$ individual is $(N - M)$:

$$S'''(a) = (S''_{a1}, S''_{a2}, \dots, S''_{a(N-M)}). \quad (5)$$

The total number of $S''(a)$ and $S'''(a)$ is $2(N - M)$. Then all of the $2(N - M)$ individuals are ranked by the fitness value, and the $(N - 2M)$ individuals with lower fitness are

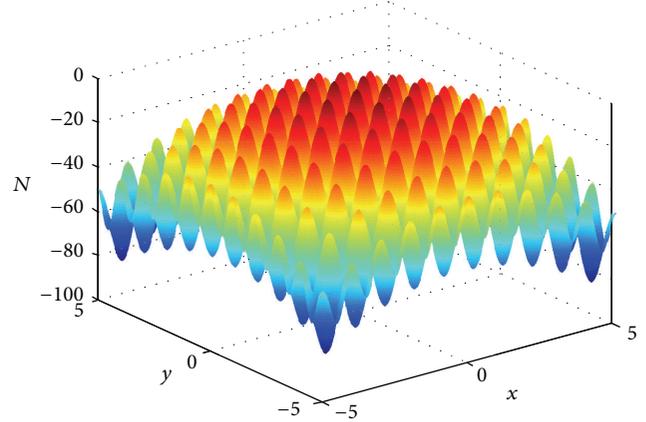


FIGURE 1: Graph of Rastrigin's function.

eliminated. The N individuals with higher fitness are retained to constitute the next generation of population $S(a + 1)$:

$$S(a + 1) = (S_{(a+1)1}, S_{(a+1)2}, \dots, S_{(a+1)N}). \quad (6)$$

The iterative process of GA carries on according to the above method until the condition of convergence is satisfied.

In TGA, offspring individuals are generated from parent individuals with higher fitness through the trisection population selection operator. TGA can quickly lead the solution procedure to the globally optimal solution.

For comparison with the parent-offspring competition mechanism (POGA) and the elitist genetic algorithm (EGA), this study used Rastrigin's function to test the effectiveness of three methods.

The function expression of Rastrigin's function is listed as:

$$f(x, y) = -\left[20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)\right] \quad (7)$$

$$-5 < x < 5, \quad -5 < y < 5.$$

Figure 1 shows the graph of Rastrigin's function. From Figure 1, we know that there are many local maximum points, with just one global maximum point. The global maximum point is point $(0, 0)$, and the maximum is 0. The progress of the genetic algorithm is easy to fall into local maximum points when finding globe maximum of this function.

This study uses binary encoding for the encoding mechanism of the three GAs (POGA, EGA, and TGA). Specific parameters are set as follows: the number of bits in the chromosomes was 40, the number of individual species was 100, the number of evolution was 200, the crossover probability P_c was 0.9, and the mutation probability P_m was 0.01. Operation selection was performed using the roulette wheel. One hundred running times were used to obtain the reality effect of three GA methods. The threshold value was set as $1.0E - 004$ so that the algorithm was convergent when the convergences value was always larger than the threshold value.

All three algorithms are implemented using Matlab 2012a. The algorithms were run on the same Intel Core i5-2400CPU

TABLE 1: Test results of different selection methods used in gas.

Choice mechanism	POGA	EGA	TGA
Running times	100	100	100
Number of convergence	77	79	89
Convergence probability	77%	79%	89%
Minimum convergence iteration	8	15	10
Average convergence iteration	56	55	45

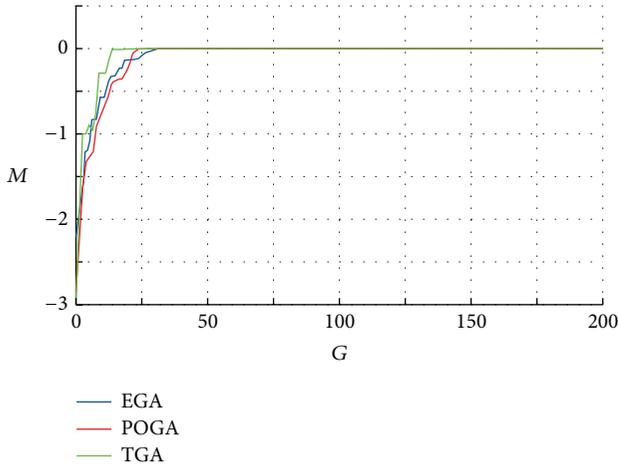


FIGURE 2: Change of the largest fitness value of Rastrigin's function.

@3.0 GHz with 2.0 GB RAM in the Windows XP Operating System. Table 1 displays test results.

Figure 2 shows the change in the largest fitness value for Rastrigin's function in three GA methods, in which the abscissa is evolutionary algebra G and the ordinate is the maximum fitness M .

Table 1 indicates that the convergence probability and average convergence algebra of the trisection population method selection operator genetic algorithm (TGA) proposed in this study was far superior to that of POGA and EGA. Figure 2 shows that the shape of the curve of TGA is rapid on its way to the global maximum of Rastrigin's function. Therefore, we can conclude that TGA is feasible and predominant.

3. Improvement of Adaptive Crossover Operator and Mutation Operator

In genetic algorithms, the crossover operator plays a central role in global search ability. New individuals are generated by exchanging the structure of part of the parent individuals. The location of the cross is randomly selected from one or more genes in the parent individuals' genes. This indicates that the search capability of GA can be improved tremendously. Genetic operations depend on the selection operator, the crossover operator, and mutation operator of the genetic operator. The values of crossover probability and mutation probability significantly affect performance of the algorithm. The crossover probability and mutation probability of basic genetic algorithm are fixed, while different optimization

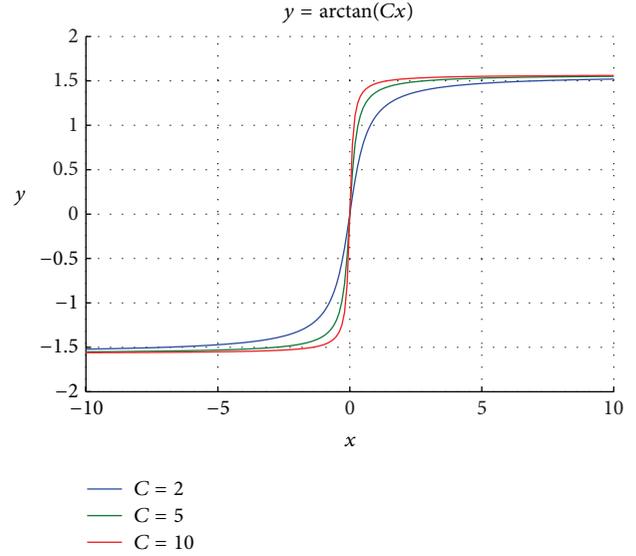


FIGURE 3: Graph of arc tangent function $y = \arctan(Cx)$.

problems progress through trial and error in order to improve P_c and P_m . However, in the optimization process of GA, different P_c and P_m are needed for individuals with different fitness.

If the value of P_c is too large, new individuals are generated quickly. At the same time, genetic models may be destroyed. However, if the value of P_c is too small, the search process is too slow. Genetic algorithms are easy to degenerate into the pure random search method if the value of P_m is too large. However, it is difficult to produce new individuals if the value of P_m is too small. In light of this, many scholars have developed new adaptive genetic operators. The values of P_c and P_m are distributed along the fitness of different individuals though these adaptive genetic operators.

Srinivas and Patnaik [16] introduced a new method in which the values of crossover probability and mutation probability change automatically along the fitness of individuals (adaptive GA, AGA). However, the AGA cannot improve individual evolution in the early part of the process, and it easily becomes trapped in local optimal solution.

Shi et al. [17] designed a new method in which the values of crossover probability and mutation probability change automatically based on the cosine function (CAGA). In CAGA, P_c and P_m change nonlinearly, but the adaptive adjustment curve nears the straight line when differentials of f_{avg} and f_{max} are too large.

This work developed a new adaptive genetic operator based on arc tangent function (AAGA). We know that the arc tangent function $y = \arctan(x)$ has the following characteristics.

- (1) The function value of arc tangent function is between $(-\pi/2, \pi/2)$.
- (2) The arc tangent function is an odd function whose domain is $(-\infty, +\infty)$.

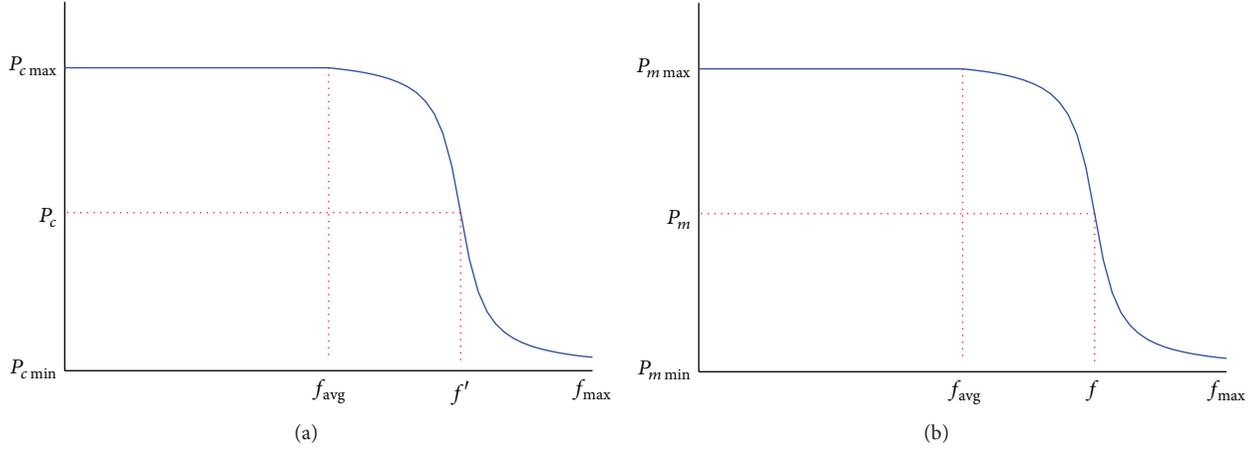


FIGURE 4: Curve of adaptive adjustment crossover probability and mutation probability in AAGA.

- (3) The value of the arc tangent function's derived function tends to zero when $x \rightarrow +\infty$ and tends to maximum when $x \rightarrow 0$.

As shown in Figure 3 which displays graphs of the arc tangent function, the top and bottom of the graph are smooth, but the value of the function changes rapidly when constant C grows. With the feature of the function, a new

adaptive frame function of P_c and P_m is formed, so that individuals with higher fitness cross each other with smaller values of crossover probability, and individuals with fitness close to f_{avg} cross each other with larger values of crossover probability.

The expression of P_c and P_m in AAGA are as follows. The formulas are enumerated when finding the maximum, and they are similar when finding the minimum:

$$P_c = \begin{cases} \frac{P_{c \min} + P_{c \max}}{2} + \frac{P_{c \min} - P_{c \max}}{\pi} \cdot \arctan\left(C \cdot \frac{2f' - f_{\max} - f_{\text{avg}}}{2(f_{\max} - f_{\text{avg}})}\right), & f' \geq f_{\text{avg}}, \\ P_{c \max}, & f' < f_{\text{avg}}, \end{cases} \quad (8)$$

$$P_m = \begin{cases} \frac{P_{m \min} + P_{m \max}}{2} + \frac{P_{m \min} - P_{m \max}}{\pi} \cdot \arctan\left(C \cdot \frac{2f - f_{\max} - f_{\text{avg}}}{2(f_{\max} - f_{\text{avg}})}\right), & f \geq f_{\text{avg}}, \\ P_{m \max}, & f < f_{\text{avg}}, \end{cases}$$

where $P_{c \max}$ and $P_{c \min}$ are the top and bottom limitation of crossover probability, $P_{m \max}$ and $P_{m \min}$ are the top and bottom limitation of mutation probability, f_{avg} is the average fitness of the entire population, f_{\max} is the largest fitness of population individuals, f' is the bigger fitness of the two individuals that take part in the crossover, f is the fitness of the mutation individuals, and C is a constant that can control the changing rate when the value of the function is close to f_{avg} . Usually the value of C is between 20 and 60. In this work, the value of C is 40.

Figure 4 displays the curves of (8).

As shown in Figure 4, crossover probability P_c and mutation probability P_m change adaptively between the maximum adaptation degree f_{\max} and the average adaptation degree f_{avg} based on arc tan function. Compared to AGA and CAGA, individuals whose fitness degree is located as $[f_{\text{avg}}, (f_{\text{avg}} + f_{\max})/2]$ can be operated with bigger crossover probability P_c and mutation probability P_m , while individuals whose fitness degree is located at $[(f_{\text{avg}} + f_{\max})/2, f_{\max}]$ can

be operated with the smaller crossover probability P_c and mutation probability P_m . This means that AAGA tends to the optimal solution quickly, ensuring that excellent individuals are not destroyed.

Now a new genetic algorithm (T-AAGA) is formed by combining trisection population method selection operator and adaptive genetic operator based on the arc tangent function. Figure 5 displays the flow diagram of T-AAGA.

4. Convergence Performance of T-AAGA and Simulation Testing

Form the improving process of T-AAGA, we can know that T-AAGA is a special case of EGA. Compared to EGA, more excellent individuals are retained to the next generation of population. Early in the 1990s, Eiben and Rudolph have proven that EGA can be convergent to globally optimal solution in a probability value of 100% [18, 19]. So we can draw a conclusion that T-AAGA is convergent.

TABLE 2: List of benchmark functions.

Test functions	Functional expression	Dimension	Region of search	Extreme value	Coordinates of extreme value point
f_1	$f(x) = 0.5 - \frac{\sin^2(\sqrt{x^2 + y^2} - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$	2	[-100, 100]	1	[0, 0]
f_2	$f(x) = \sum_{i=1}^3 (x_i - 10 \cos(2\pi x_i) + 10)$	3	[-5, 5]	0	[0, 0, 0]
f_3	$f(x) = \frac{1}{4000} \sum_{i=1}^{15} x_i^2 - \prod_{i=1}^{15} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	15	[-10, 10]	0	[0, 0, ..., 0]
f_4	$f(x) = \frac{1}{15} \sum_{i=1}^{15} (x_i^4 - 16x_i^2 - 5x_i)$	15	[-10, 10]	-78.332314	(2.9051, 2.9051, ..., 2.9051)
f_5	$f(x) = \frac{1}{30} \sum_{i=1}^{30} (10(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	[-10, 10]	0	(1, 1, ..., 1)

TABLE 3: Simulation results of different algorithms for f_1-f_5 .

Algorithm	Stats	f_1 (1.0)	f_2 (0.0)	f_3 (0.0)	f_4 (-78.332314)	f_5 (0.0)
PSO	Best	1.0	0.0	0.0	-78.332302	8.6E - 04
	Mean	0.999984	0.11968	0.003795	-68.367517	1.798256
	Std.	7.0E - 06	8.6E - 03	3.0E - 03	7.2E - 00	8.8E - 01
CMAES	Best	0.999925	0.029721	0.0	-78.332289	0.0
	Mean	0.996273	0.070160	0.001008	-71.123059	0.254623
	Std.	1.8E - 03	2.9E - 02	8.9E - 04	6.6E - 00	1.0E - 01
SaDE1	Best	0.999998	0.000527	0.0	-78.332297	3.0E - 06
	Mean	0.997836	0.045876	0.002766	-66.549080	0.387544
	Std.	1.7E - 03	3.2E - 02	2.0E - 03	9.9E - 00	3.1E - 01
SaDE2	Best	1.0	0.000561	0.0	-78.332305	7.4E - 04
	Mean	0.999328	0.001673	0.007731	-72.207516	0.913510
	Std.	4.3E - 04	7.3E - 04	6.4E - 03	5.8E - 00	6.9E - 01
T-AAGA	Best	1.0	0.0	0.0	-78.332306	0.0
	Mean	0.999973	0.007468	5.22E - 05	-77.453346	0.118650
	Std.	4.05E - 06	1.44E - 04	6.05E - 05	2.36E - 00	2.03E - 02

To test the performance of the proposed IAFOA algorithm, we consider a total of 5 benchmark functions. The details of these 5 functions are found in Table 2.

Four widely used evolutionary algorithms are compared with the proposed T-AAGA approach: they are the particle swarm optimization (PSO) algorithm [20], the covariance matrix adaptation evolution strategy (CMAES) [21], the self-adaptive differential evolution (SaDE1) algorithm [22], and the self-adaptive differential evolution (SaDE2) algorithm [23]. The parameter settings and results of PSO, CMAES, SaDE1, and SaDE2 come from the literature [20–24]. The main parameter settings of T-AAGA are as follows: the number of bits in the chromosomes was 40, the number of individuals was 200, the number of evolution was 1000, $P_{c \max} = 0.7$, $P_{c \min} = 0.35$, and $P_{m \max} = 0.1$, $P_{m \min} = 0.05$. Stopping iterative criteria is that the maximum number of iterations equals 1000.

Owing to their stochastic nature, evolutionary algorithms may arrive at solutions that are better or worse than solutions they have previously reached during their search for new solutions. For this reason, it is beneficial to use statistical tools to compare the problem-solving success of one algorithm with that of another [24]. All five of these algorithms were run 50 times for 5 test functions.

The comparison of the results is reported in Table 3. The “Best” means the optimal objective function value, the “Mean” reflects the precision and rate of convergence, and the “Std.” reflects the stability and robustness of the algorithm. From Table 3, we can know that T-AAGA have best performance in “Best,” “Mean,” and “Std.” of f_3 , f_4 , and f_5 . Besides, T-AAGA have best performance in “Best” and “Std.” of f_1 and f_2 . It can be concluded that the proposed T-AAGA has made great improvement in astringency, accuracy, and stability in finding the optimal value.

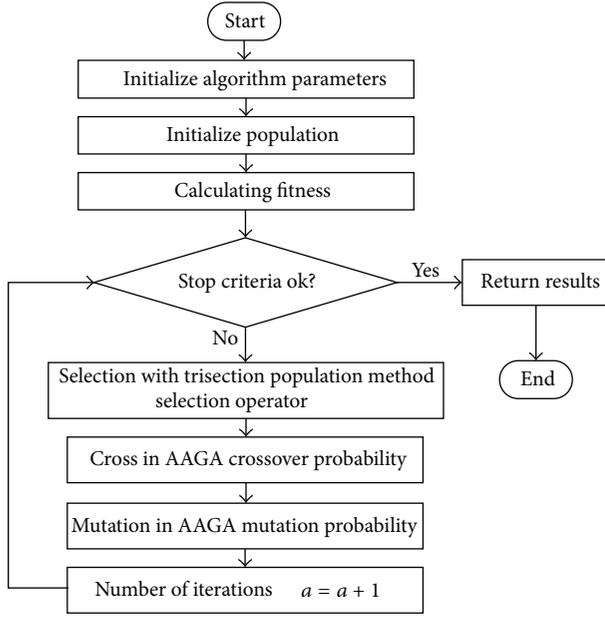


FIGURE 5: Flow diagram of T-AAGA.

5. Drilling Equipment Layout of Semisubmersible Drilling Platforms

This section describes how T-AAGA was used to examine the drilling equipment layout of semisubmersible drilling platforms. Drilling equipment layout is an important part of the general design of drilling rig systems. It has important effects on operation performance and drilling efficacy and forms the main basis of further detailed designs of drilling rig systems. An effective layout ensures smooth drilling efficiency and considers effects on the overall performance of the drilling platform. A reasonable layout scheme retains drilling platform stability, security, reliability, and other indicators in a better state. This paper briefly discusses the layout mathematical model of drilling equipment in semisubmersible platforms.

5.1. Mathematical Model. In real-world engineering, factors such as operation cost, operation efficiency, and stability of the platform are the main focus, with the optimization goal to minimize the centroid transverse deviator of the platform. The centroid transverse deviator refers to the distance between the actual barycenter position of all equipment and the center of the centroid. To improve the stability of the platform, the centroid transverse deviator should be as small as possible, with the unit in meters:

$$F(x) = \frac{\sqrt{(\sum_{i=1}^n m_i x_{ic})^2 + (\sum_{i=1}^n m_i y_{ic})^2}}{\sum_{i=1}^n m_i}, \quad (9)$$

where m_i is the weight (unit: KG) of the i th equipment and x_{ic} and y_{ic} , respectively, are the barycentric abscissa and ordinate.

The layout of drilling equipment is limited by many constraint conditions. In this work, the significant constraints include the boundary constraint, interference constraint, and process constraint. The layout mathematical model can be expressed as follows:

$$\text{Min } F(x)$$

$$\text{s.t. } g_1(x) = |x_{ic}| + |y_{ic}| + \frac{l_i + w_i}{2} - \frac{L + W}{2} \leq 0,$$

$$g_2(x) = \begin{cases} |x_{ic} - x_{jc}| - \frac{P_{11} + P_{12}}{2} > 0 \\ \text{or} \\ |y_{ic} - y_{jc}| - \frac{P_{21} + P_{22}}{2} > 0, \end{cases}$$

$$g_3(x) = \begin{cases} -a \leq x_i \leq a \\ -b \leq y_i \leq b, \end{cases}$$

$$g_4(x)$$

$$= \begin{cases} \frac{(w_{gl} + w_z)}{2} - a \leq x_{gl} \leq \frac{(w_{gl} + w_z)}{2} + a \\ -b \leq y_{gl} \leq b, \end{cases} \quad (10)$$

$$g_5(x)$$

$$= \begin{cases} -\frac{(l_{gz} + w_z)}{2} - a \leq x_{gz} \leq -\frac{(l_{gz} + w_z)}{2} + a \\ -b \leq y_{gl} \leq b, \end{cases}$$

$$g_6(x)$$

$$= \begin{cases} -a \leq x_{sx} \leq a \\ -b + \frac{(l_z + w_{sx})}{2} \leq y_{sx} \leq b + \frac{(l_z + w_{sx})}{2} \\ \text{or} \\ -b - \frac{(l_z + w_{sx})}{2} \leq y_{sx} \leq b - \frac{(l_z + w_{sx})}{2}, \end{cases}$$

where $g_1(x)$ is the expression of the boundary constraint, $g_2(x)$ is the expression of the interference constraint, and $g_3(x)$, $g_4(x)$, $g_5(x)$, and $g_6(x)$ are the expression of process constraints. All meanings of mathematical quantities are found in the literature [25].

5.2. Numerical Examples. To verify the effectiveness of T-AAGA, a specific example is adopted; the experimental data is covered in the literature [25]. In this example, the length of the deck was 78 meters, and the width of deck was 70 meters. The details on size and weight of allocated objects are listed in Table 4. It is notable that the location of the drill floor area center must be coincident with the drilling well location.

The improved genetic algorithm is used to solve numerical examples. For comparing results, the algorithm parameter is similar to parameter in the literature [25]. The details are as follows: the population size is 100; the number of iterations is 600; the top limitation $P_{c \max}$ and bottom limitation $P_{c \min}$ of crossover probability, respectively, are 0.70 and 0.35;

TABLE 4: Size and weight of allocated objects.

Serial number	Layout module	Length/m	Width/m	Height/m	Quality/t
1	Area of drill floor	33	24	10	2860
2	Drill collar storage area	9.60	1.95	1.07	86.51
3	Drill pipe area number 1	9.60	8.33	1.63	188.36
4	Drill pipe area number 2	9.60	8.33	1.63	188.36
5	30 in drive pipe area	12.19	2.63	2.63	60.46
6	20 in drive pipe area	12.19	3.35	3.35	99.64
7	13-3/8 in drive pipe area	12.19	3.90	3.90	115.96
8	9-5/8 in drive pipe area	11.99	6.72	5.89	575.93
9	7 in drive pipe area	10.36	4.78	4.05	366.23
10	Flatwise marine riser area	22.86	13	7	670.07
11	Vertical marine riser area	32	10	22.86	2,113.79
12	Pipe conveyor area	24	4	10	35
13	Bop area	28.5	10	3.8	400
14	Christmas tree area	20	9.5	3.8	130
15	Mud purification area	25	16.1	2	115.39
16	Living quarters	38.00	10.70	11.00	342

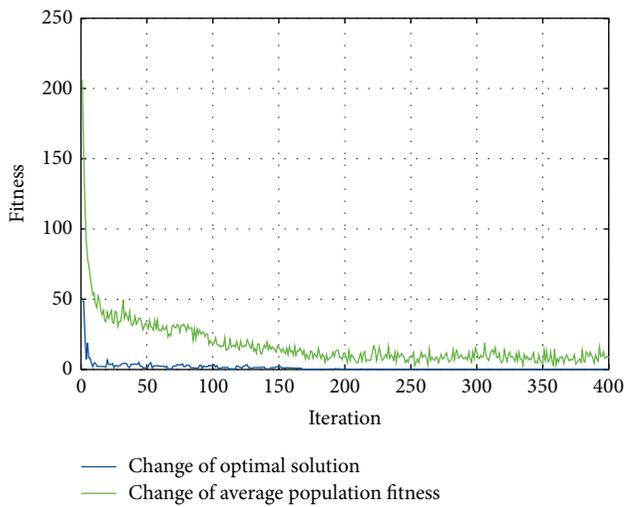


FIGURE 6: Performance tracking of objective function value and the average value of population fitness.

top limitation $P_{m,max}$ and the bottom limitation $P_{m,min}$ of crossover probability, respectively, are 0.10 and 0.05.

The concrete solving process is implemented using Matlab 2012a for 300 times. Figure 6 displays performance tracking of the objective function value and the average value of the population.

As shown in Figure 6, simulation results demonstrate that the algorithm has fast convergence performance and high accuracy. A special layout case shown in Figure 7 was selected from the computational results.

In Figure 7, the coordinate planes represent layout plane of platform, and coordinate center represents the drilling center.

The centroid transverse deviator of the special case in Figure 7 is 0.128 m, and the allowable value of centroid transverse deviator is 0.7 m. Compared with [26] that uses hybrid

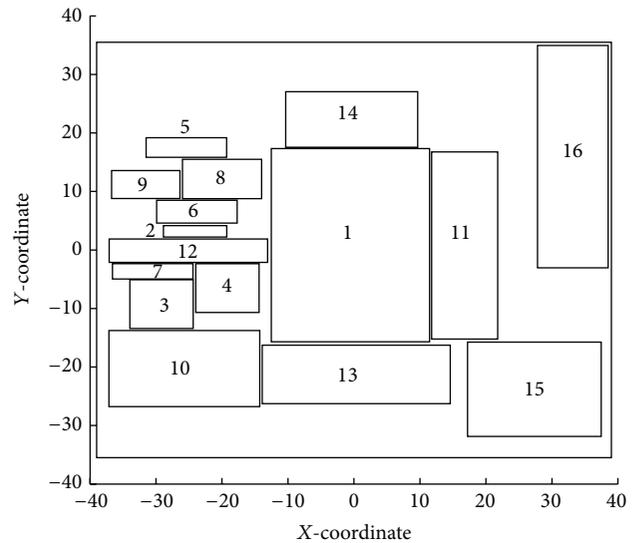


FIGURE 7: Special case of optimal layout scheme.

genetic algorithms (HCGA), we know that 0.128 meters is far less than the results 0.392 m, 0.687 m, and 0.495 m in [26]. The genetic annealing algorithm used for solving the value of centroid transverse deviator and the results 0.5845 m and 0.4398 m are larger than 0.128 m. Therefore, we conclude that the improved genetic algorithm T-AAGA is more efficient.

6. Conclusions

- (1) This study introduced a selection operator and proposed a trisection population method. Using the competition test function, this study compared the elitist genetic algorithm and the parent-offspring competition mechanism. Results show that

the improved genetic algorithm (T-AAGA) is superior in convergence probability and convergence speed. At the same time, the iterative process of T-AAGA is relatively smooth and steady.

- (2) Compared with AGA and CAGA, a new genetic operator based on arc tangent function can change the values of crossover probability and mutation probability adaptively. Therefore, individuals with different fitness levels show different crossover probability and mutation probability, and the convergence rate of the population is guaranteed.
- (3) This study used the improved genetic algorithm T-AAGA to solve layouts for deep-water, semisubmersible platform drilling equipment. Combined with numerical examples, the results show considerable improvement compared with references. The centroid transverse deviator is just 0.128 m, which is far less than the allowable value.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the project foundation of China Ministry of Industry and Information Technology “Research of Gordian Technique of Deep-Water Semi-Submersible platforms,” and the National 863 Plan Project Foundation of China “Key technology research of automated processing of deep-water drilling rig and string” (no. 2012AA09A203).

References

- [1] J. H. Holland, *Adaptation in Natural Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [2] S. Soares, C. H. Antunes, and R. Araújo, “Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development,” *Neurocomputing*, vol. 121, pp. 498–511, 2013.
- [3] C. Lance, *The Practical Handbook of Genetic Algorithms: Applications*, Chapman & Hall, Boca Raton, Fla, USA, 2001.
- [4] A. M. Abazari, M. Solimanpur, and H. Sattari, “Optimum loading of machines in a flexible manufacturing system using a mixed-integer linear mathematical programming model and genetic algorithm,” *Computers and Industrial Engineering*, vol. 62, no. 2, pp. 469–478, 2012.
- [5] D. Mei, X. Du, and Z. Chen, “Optimization of dynamic parameters for a traction-type passenger elevator using a dynamic byte coding genetic algorithm,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 3, pp. 595–605, 2009.
- [6] T. Zhang, Y.-J. Zhang, and S.-X. Liu, “A mixed integer programming model and improved genetic algorithm for order planning of iron-steel plants,” *International Journal of Information and Management Sciences*, vol. 19, no. 3, pp. 413–435, 2008.
- [7] K. Deb, “A population-based algorithm-generator for real-parameter optimization,” *Soft Computing*, vol. 9, no. 4, pp. 236–253, 2005.
- [8] F. Herrera and M. Lozano, “Gradual distributed real-coded genetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 1, pp. 43–62, 2000.
- [9] J. A. Hageman, R. Wehrens, H. A. van Sprang, and L. M. C. Buydens, “Hybrid genetic algorithm-tabu search approach for optimising multilayer optical coatings,” *Analytica Chimica Acta*, vol. 490, no. 1-2, pp. 211–222, 2003.
- [10] S.-K. Oh, W. Pedrycz, and H.-S. Park, “Genetically optimized fuzzy polynomial neural networks,” *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 1, pp. 125–144, 2006.
- [11] R. Teek, K. Kruustük, R. Žordania et al., “Hearing impairment in Estonia: an algorithm to investigate genetic causes in pediatric patients,” *Advances in Medical Sciences*, vol. 58, no. 2, pp. 419–428, 2013.
- [12] K. A. Dowsland and W. B. Dowsland, “Packing problems,” *European Journal of Operational Research*, vol. 56, no. 1, pp. 2–14, 1992.
- [13] Z. B. Xu, Z. K. Nie, and W. X. Zhang, “Almost sure strong convergence of a class of genetic algorithms with parent-offspring competition,” *Acta Mathematicae Applicatae Sinica*, vol. 25, no. 1, pp. 167–175, 2002.
- [14] C. A. Coello Coello, “An updated survey of GA-based multiobjective optimization techniques,” *ACM Computing Surveys*, vol. 32, no. 2, pp. 137–143, 2000.
- [15] L. He, K. J. Wang, G. B. Li et al., “Elitist preserved genetic algorithm and its convergence analysis,” *Control and Decision*, vol. 15, no. 1, pp. 63–66, 2000 (Chinese).
- [16] M. Srinivas and L. M. Patnaik, “Adaptive probabilities of crossover and mutation in genetic algorithms,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
- [17] S. Shi, Q. F. Li, and X. H. Wang, “Design optimization of brushless direct current motor based on adaptive genetic algorithm,” *Journal of Xi’an JiaoTong University*, vol. 36, no. 12, pp. 1215–1218, 2002 (Chinese).
- [18] A. E. Eiben, E. H. L. Aarts, and K. M. van Hee, “Global convergence of genetic algorithms: a Markov chain analysis,” in *Parallel Problem Solving from Nature*, vol. 496 of *Lecture Notes in Computer Science*, pp. 3–12, Springer, Berlin, Germany, 1991.
- [19] G. Rudolph, “Convergence analysis of canonical genetic algorithms,” *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.
- [20] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, “Particle swarm optimization: hybridization perspectives and experimental illustrations,” *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5208–5226, 2011.
- [21] C. Igel, N. Hansen, and S. Roth, “Covariance matrix adaptation for multi-objective optimization,” *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.
- [22] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, “Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [23] A. K. Qin and P. N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’05)*, pp. 1785–1791, Edinburgh, Scotland, September 2005.

- [24] X. F. Yuan, X. S. Dai, J. Y. Zhao, and Q. He, "On a novel multi-swarm fruit fly optimization algorithm and its application," *Applied Mathematics and Computation*, vol. 233, pp. 260–271, 2014.
- [25] K. K. Wei, *Research on Layout Design of Drilling Rig System on Deep-Water Semi-Submersible Drilling Platform*, China University of Petroleum, Qingdao, China, 2010 (Chinese).
- [26] G. R. Lu, *Key Technology Research on Layout Design of Semi-Submersible Platform Drilling System*, China University of Petroleum (East China), Qingdao, China, 2009, (Chinese).



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

