International Journal of Computational Intelligence Systems, Vol. 8, No. 6 (2015) 1044-1052

A New Adaptive Genetic Algorithm and Its Application in the Layout problem

Wu Lei[*]
College of mechanical and electronic engineering, China University of Petroleum, No. 66, Changjiang West Road, Huangdao District, Qingdao, Shangdong Province, China
E-mail: wuleiupc@163.com

Xiao Wensheng
College of mechanical and electronic engineering, China University of Petroleum, No. 66, Changjiang West Road, Huangdao District, Qingdao, Shangdong Province, China

Wang Jingli
College of mechanical and electronic engineering, China University of Petroleum, No. 66, Changjiang West Road, Huangdao District, Qingdao, Shangdong Province, China

Zhou Houqiang
College of mechanical and electronic engineering, China University of Petroleum, No. 66, Changjiang West Road, Huangdao District, Qingdao, Shangdong Province, China

Tian Xue
College of mechanical and electronic engineering, China University of Petroleum, No. 66, Changjiang West Road, Huangdao District, Qingdao, Shangdong Province, China

Received 10 January 2015

Accepted 14 September 2015

Abstract

Genetic algorithm (GA) is a search algorithm based on the theory of Darwin. For the purpose of improving the convergent rate and maintaining the population diversity in GA, this paper presents a new genetic operator called trisecting group and directional selection mechanism (TDGA), in which the worst 2/3 of parent individuals are removed from the population before other manipulations. With 1/3 individuals that are selected randomly from the removed parent individuals, the best 1/3 of the parent individuals is manipulated to reproduce offspring. Simulation results based on 10 test functions show that TDGA is feasible and effective. In addition, inspired by the graph of the function $f(x)=e^{-x^C}$, a new self-adaptation adjusting the tactics of crossover operator and mutation operator (SAGA) is proposed so that individuals with higher fitness cross each other with smaller values of crossover probability, and individuals with lower fitness cross each other with larger values of crossover probability. Combining the two improvements, TA-SAGA is applied to study the layout of drilling equipment in semi-submersible drilling platforms. In addition, the simulated best centroid transverse deviator just only is 0.120 m, which is far less than the allowable value 0.7m.

Keywords: genetic algorithm; trisecting group and directional selection mechanism; self-adaptation adjusting tactics; TD-SAGA; drilling equipment layout; centroid transverse deviator

---

[*] Corresponding author: wuleiupc@163.com.

Co-published by Atlantis Press and Taylor & Francis

1. Introduction

Layout problem is a challenging non-linear combinatorial optimization problem encountered in many services and manufacturing organizations such as machine manufacturing, shipbuilding, glassmaking, transportation, aerospace industry, large-scale integrated circuit design and medical physics applications [1][2][3][4][5]. The layout problem has been proved to be an NP-hard problem [6]. With the increased number of layout objects, the time that is required for problem solving increases sharply. Therefore, deterministic optimization methods are not suitable for solving this kind of problem. Thus, most scholars use intelligent optimization algorithms, such as GA (genetic algorithm) [7], PSO (particle swarm optimization) [8], ACO (ant colony optimization) [9], etc., to solve these problems. In particular, genetic algorithm has shown good performance in solving layout problems because of its better global optimization performance.

Genetic algorithm is a search algorithm that is based on the biological principles of selection, reproduction, and mutation [7] [10]. Genetic algorithm uses the theory of Darwin based on the survival of the fittest and simulates natural selection and genetic recombination [10]. In the process of genetic algorithm reproduction, genetic crossover and genetic mutation will occur in each iteration. According to some indicators, a set of candidate solutions are retained and used to produce a new generation of candidate solutions [11][12][13]. This process will repeat until some convergence indicators are meet. The object of genetic algorithm processing is the code of the parameters, rather than the parameters themselves. The search process is not restrained by the continuity of the optimization function, and does not require the continuity and differentiability of optimization function [14] [15]. The basic idea of the genetic algorithm is simple, and the operation mode and achievement steps are normative. However, the efficiency of the standard genetic algorithm is low, and it is easy to fall into local optimal solutions [16] [17]. Thus, many kinds of genetic operators such as the parent-offspring competition mechanism (POGA) [18] and elitist genetic algorithm (EGA) [19] [20], have been used in GA to improve performance. Referring to the crossover operator and mutation operator, Srinivas [21] introduced a new method in which the values of crossover probability and mutation probability change linearly with the fitness of individuals (AGA). Shi [22] designed a new method in which the values of crossover probability and mutation probability change automatically based on the cosine function (CAGA).

In this paper, a new genetic operator called trisecting group and directional selection mechanism (TDGA) is proposed; test results show that TDGA is feasible and effective. In addition, a new self-adaptation adjusting tactics of crossover operator and mutation operator (SAGA) is proposed to quickly determine the global optimization. Combining the two improvements, TA-SAGA is applied to study the layout of drilling equipment in semi-submersible drilling platforms.

The rest of the paper is organized as follows. Section 2 describes the improvement of the genetic operator. Section 3 describes the self-adaptation adjusting tactics of crossover operator and mutation operator. Section 4 introduced the layout of drilling equipment based on the improved genetic algorithm and Section 5 covers conclusions.

2. Improvement of the genetic operator

Because of the defects of the existing genetic algorithm, an improved genetic operator is proposed that is based on the trisecting group and directional selection (TDGA). The number of population is N .

The specific process of the trisecting group and directional selection mechanism is as follows:

Let us define the population of generations

$$P(g)=(P_{g1},P_{g2},\cdots,P_{aN}) \tag{1}$$

Where ***g*** is the number of generations. The fitness of P_{gn} is expressed by $F_{P_{gn}}$.

Define ***M***

$$M=\text{int}(\frac{N}{3}) \tag{2}$$

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

According to the fitness function, the individuals' fitness are arranged on the order of high to low, to get $P(g)'$

$$P(g)'=(P_{g1}',P_{g2}',\cdots,P_{aN}') \tag{3}$$

and stipulating $F_{P_{g1}'} \geq F_{P_{g2}'} \geq \cdots \geq F_{P_{gN}'}$

The highest fitness of $N/3$ parent individuals is genetically manipulated to reproduce offspring, and the other $N/3$ manipulated parent individuals are selected randomly from the remaining $2N/3$ parent individuals.

Currently, the number of genetically manipulated parent individuals is 2***M***. Thus, the number of offspring individuals is 2***M,*** and the total number of two generations is 4***M***.

Sorting the 4*M* individuals according their fitness, the worse (4*M*–*N*) individuals will be removed from the population. Then the remaining *N* individuals make up the next generation population $P(g+1)$.

$$P(g+1)=(P_{(g+1)1},P_{(g+1)2},\cdots,P_{(g+1)N}) \qquad (4)$$

With the above method, the iterative process of TDGA is carried out gradually until the convergence condition is satisfied.

This method can not only maintain the continuity of highest-fitness individuals in the population, but also ensure the diversity of the population by randomly selecting non-optimal individuals. The diversity of the population makes significant contributions to avoid falling into local optimal solutions.

In order to verify the excellent properties of the TDGA as proposed in this article, POGA and EGA were compared. Ten common test functions that are shown in Table 1 were selected to contrast the performance of the three methods.

In order to ensure that the results were comparable, same parameters were used when solving the problem through POGA, EGA and TDGA. Specific parameters were set as in Table 2.

Table1. List of benchmark functions

Test functions	Functional expression	Region of search	Extreme value	Coordinates of extreme value point
f_1	$f(x_1,x_2)=100({x_1}^2-x_2)^2+(1-x_1)^2$	[-2,2]	0	[1,1]
f_2	$f(x_1,x_2)=0.5-\frac{\sin^2(\sqrt{{x_1}^2+{x_2}^2}-0.5)}{(1+0.001({x_1}^2+{x_2}^2))^2}$	[-100,100]	1	[0,0]
f_3	$f(x_1,x_2)=-\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}$	[-100,100]	-1	[π,π]
f_4	$f(x)=\sum_{i=1}^{3}{x_i}^2$	[-5.12,5.12]	0	[0,0,0]
f_5	$f(x)=\sum_{i=1}^{3}(x_i-10\cos(2\pi x_i)+10)$	[-5,5]	0	[0,0,0]
f_6	$f(x)=\sum_{i=1}^{5}\lvert x_i\rvert+\prod_{i=1}^{5}\lvert x_i\rvert$	[-10,10]	0	[0,0,0,0,0]
f_7	$f(x)=1-\cos(2\pi\sqrt{\sum_{i=1}^{5}{x_i}^2})+0.1\sqrt{\sum_{i=1}^{5}{x_i}^2}$	[-100,100]	0	[0,0,0,0,0]
f_8	$f(x)=\frac{1}{4000}\sum_{i=1}^{15}x_i^2-\prod_{i=1}^{15}\cos(\frac{x_i}{\sqrt{i}})+1$	[-10,10]	0	[0,0,…,0]
f_9	$f(x)=\sum_{i=1}^{15}\lvert x_i\sin(x_i)+0.1x_i\rvert$	[-10,10]	0	[0,0,…,0]
f_{10}	$f(x)=\frac{1}{30}\sum_{i=1}^{30}(10(x_{i+1}-x_i^2)^2+(x_i-1)^2)$	[-10,10]	0	(1,1,…,1)

Table2. List of specific parameters

Choice mechanism	Encoding mechanism	Number of bits	Number of individual species	Number of evolution	Crossover probability	Mutation probability	Operation selection
POGA	binary encoding	40	200	500	0.9	0.01	Roulette wheel
EGA	binary encoding	40	200	500	0.9	0.01	Roulette wheel
TDGA	binary encoding	40	200	500	0.9	0.01	Roulette wheel

One hundred running times were used to obtain the reality effect of three GA methods. All three algorithms were implemented using Matlab 2012a. The algorithms were run on the same Inter (R) i5-2400CPU @3.0 GHz with 4.0 GB RAM in the Windows 2007 Operating System. The test results are displayed in Table 3, where “Best” means the optimal objective function value, “Mean” reflects the precision and rate of convergence, “CP” means convergence probability, and “ACI” means the Average convergence iteration.

From the simulation results we know that TDGA has the best performance in finding the extremum of the ten test functions compared to POGA and EGA. In detail, TDGA had the best performance in “Best”, “Mean”, “CP” and “ACI” of all of the test functions. Thus we conclude that TDGA is feasible and predominant.

Table 3. Simulation results of different algorithms for f1-f10

Choice mechanism	Stats	f_1 (0)	f_2 (1)	f_3 (-1)	f_4 (0)	f_5 (0)	f_6 (0)	f_7 (0)	f_8 (0)	f_9 (0)	f_{10} (0)
POGA	Best	0.0	0.99992	-1.0	1.5E -8	3.4E -7	4.1E -6	0.0	1.8E -3	8.9E -3	1.5E -2
	Mean	3.5E -8	0.99756	-0.99711	8.5E -7	5.8E -5	3.5E -4	4.4E -5	1.24578	1.35476	3.77130
	CP	81%	75%	70%	90%	77%	79%	72%	68%	56%	45%
	ACI	62	89	120	83	76	159	267	296	385	406
EGA	Best	0.0	0.99998	-1.0	2.6E -8	4.8E -7	0.0	6.7E -6	3.0E -3	1.8E -2	6.0E -3
	Mean	2.7E -8	0.99633	-0.99815	1.2E -7	6.7E -5	4.6E -4	8.9E -5	2.54891	2.14366	2.54162
	CP	83%	70%	75%	88%	86%	73%	65%	65%	69%	49%
	ACI	78	101	145	95	88	178	304	336	310	399
TDGA	Best	0.0	1.0	-1.0	0.0	1.8E -7	0.0	0.0	5.9E -4	4.5E -3	4.1E -3
	Mean	1.5E -8	0.99942	-0.99907	4.5E -8	8.4E -6	1.8E -4	1.3E -5	1.14632	1.24711	1.01194
	CP	92%	86%	83%	98%	96%	91%	85%	85%	88%	72%
	ACI	45	80	103	72	49	120	192	214	255	346

3. Self-adaptation adjusting tactics of crossover operator and mutation operator

In genetic algorithms, the crossover operator and mutation operator play important roles in promoting the ability to determine global optimization. Corresponding to nature evolution procedures, crossover operation is similar to genetic recombination, and mutation operation is similar to genetic mutation. In order to quickly determine the global optimization, we hope that individuals with higher fitness cross each other with smaller values of crossover probability and that individuals with lower fitness cross each other with greater values of crossover probability. For the same reason, the situation is suitable for the mutation operator.

Inspired by the graph of function $f(x)=e^{-x^C}$, which is shown in Fig.1, a new self-adaptation adjusting tactics of the crossover operator and mutation operator (SAGA) is proposed in this section.

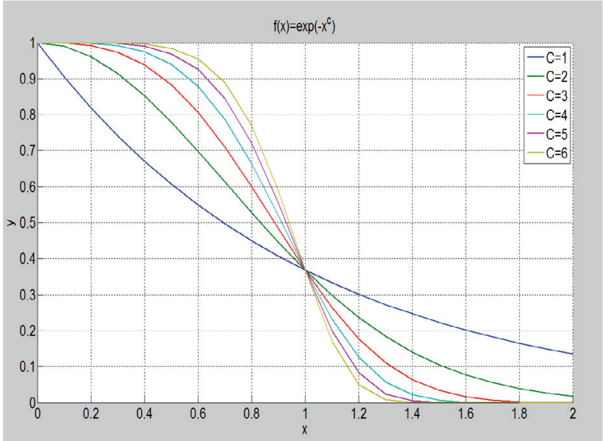

Fig.1. Graph of the function $f(x)=e^{-x^C}$

From Fig.1, we know that the top and bottom of the graph are smooth, but the graph changes rapidly near $x=1$. With the increased value of constant C, the steepness of the graph increases. With the feature of the function, the values of the crossover operator P_c and mutation operator P_m can be expressed as follows (the equations are enumerated when finding the maximum, and they are similar when finding the minimum):

$$P_c = (P_{c\max} - P_{c\min})e^{-[2\frac{f^{'} - f_{\min}}{f_{\max} - f_{\min}}]^C} + P_{c\min} \qquad (5)$$

$$P_m = (P_{m\max} - P_{m\min})e^{-[2\frac{f - f_{\min}}{f_{\max} - f_{\min}}]^C} + P_{m\min} \qquad (6)$$

Where $P_{c\max}$ and $P_{c\min}$ are the top and bottom limits of crossover probability, respectively; $P_{m\max}$ and $P_{m\min}$ are the top and bottom limits of mutation probability, respectively; $f_{\max}$ is the largest fitness of population individuals; $f^{'}$ is the bigger fitness of the two individuals that take part in the crossover; and f is the fitness of the mutation individuals.

From the graph of function $f(x) = e^{-x^C}$, Eq. (5) and Eq. (6), it is obvious that the constant C has a significant impact on SAGA. A discussion about the suitable value of constant C is provided in this section. Ten common test functions that are shown in Table 1 are used to test the influence of different values of the constant C with SAGA. The Specific parameters are set as in Table 2. The simulation results are listed in Table 4.

Table 4. Simulation results of different constant C with SAGA

Constant C	Stats	f_1 (0)	f_2 (1)	f_3 (-1)	f_4 (0)	f_5 (0)	f_6 (0)	f_7 (0)	f_8 (0)	f_9 (0)	f_{10} (0)
C =1	Mean	7.3E -5	0.98564	-0.91766	2.0E -5	4.5E -4	4.2E -2	5.6E -3	5.47126	8.46127	7.51362
	ACI	102	158	256	197	176	289	346	409	399	421
C =2	Mean	3.5E -6	0.94125	-0.94118	7.7E -6	6.5E -5	5.6E -3	4.8E -4	4.12578	5.66214	4.55126
	ACI	89	145	179	158	124	204	259	342	345	405
C =3	Mean	7.7E -7	0.99773	-0.99475	4.1E -7	3.1E -5	4.2E -4	9.4E -5	3.64595	3.33458	3.11648
	ACI	56	124	144	106	84	153	220	276	307	365
C =4	Mean	2.6E -7	0.99984	-0.99811	5.6E -8	1.25 -5	2.8E -4	4.5E -5	1.94851	2.00149	2.00149
	ACI	49	86	135	87	62	149	202	243	287	355
C =5	Mean	4.8E -6	0.99987	-0.99746	8.1E -7	2.5E -5	3.5E -4	2.8E -5	2.04651	1.64719	1.85647
	ACI	51	84	121	92	78	164	243	235	294	362
C =6	Mean	7.2E -6	0.99846	-0.99102	6.6E -7	4.7E -5	7.7E -4	8.6E -5	2.43849	2.74156	2.11457
	ACI	66	90	164	104	94	189	238	279	300	389
C =7	Mean	4.8E -5	0.99741	-0.98553	1.5E -6	8.8E -5	3.0E -3	2.0E -4	3.46912	3.22544	2.59476
	ACI	78	99	159	112	88	201	278	345	326	378
C =8	Mean	5.1E -5	0.99655	-0.98422	3.2E -6	7.9E -5	4.5E -3	4.9E -4	4.66215	4.21657	3.02469
	ACI	80	105	197	155	105	194	294	309	348	402
C =9	Mean	5.5E -5	0.99548	-0.97843	7.4E -6	1.4E -4	8.8E -3	1.2E -3	5.49874	4.65849	4.55166
	ACI	82	146	183	149	142	215	315	348	336	399
C =10	Mean	7.2E -5	0.99673	-0.98072	1.2E -5	1.8E -4	7.5E -3	8.9E -4	4.91439	5.44618	3.94561
	ACI	86	150	202	177	166	206	307	379	357	408

As shown in Table.4, SAGA has a better performance when the value of C ranges from 3 to 5. Thus, as a general rule, the constant C is assigned the value of 4.

Now an improved genetic algorithm (TD-SAGA) is formed by combining the “trisecting group and directional selection operator (TDGA)” and “self-adaptation adjusting tactics of crossover operator and mutation operator (SAGA)”. Fig.2 displays the flow diagram of TD-SAGA.

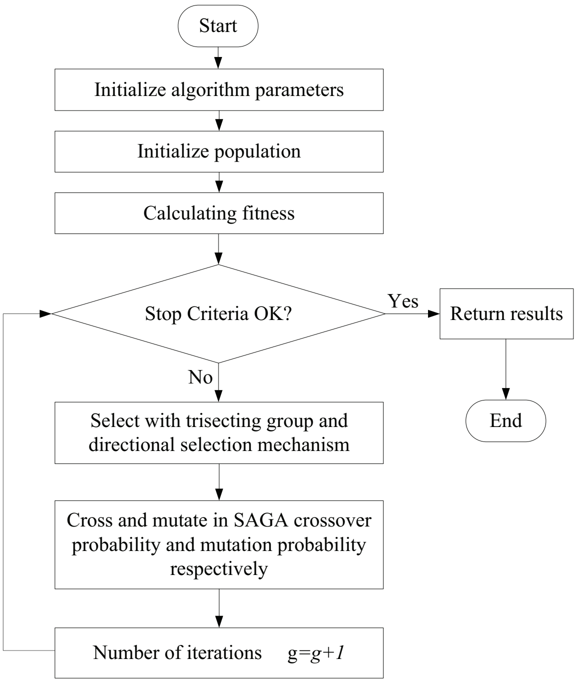
Start
Initialize algorithm parameters
Initialize population
Calculating fitness
Stop Criteria OK?
Yes
Return results
End
No
Select with trisecting group and directional selection mechanism
Cross and mutate in SAGA crossover probability and mutation probability respectively
Number of iterations g=g+1

Fig.2. Flow diagram of TD-SAGA

4. Layout of drilling equipment based on the improved Genetic algorithm

In this section, the improved genetic algorithm (TD-SAGA) is applied to research the layout of drilling equipment in deep water semi-submersible platforms. Drilling equipment layout is an important part of the general design of drilling rig systems. The layout scheme of the drilling rigs system should ensure the high efficiency and smooth operation of the drilling operation. A reasonable layout scheme retains the drilling platform stability, security, reliability and other indicators in a better state. In addition, the influence of layout on the overall performance of the drilling platform should be considered.

4.1. *Mathematical model*

According to engineering practice, the operation cost, operating efficiency and stability of the platform are the key considerations, and the optimization objective is 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.

$$F(x)=\frac{\sqrt{\left(\sum_{i=1}^{n} m_i x_{ic}\right)^2+\left(\sum_{i=1}^{n} m_i y_{ic}\right)^2}}{\sum_{i=1}^{n} m_i} \tag{7}$$

In Eq. (7), m_i indicates the quality of the i-th equipment with the unit KG, x_{ic} and y_{ic} indicate the barcentric abscissa and ordinate respectively.

The layout of the drilling equipment is limited by many conditions. In this paper, we focus on the boundary constraint, interference constraint and process constraint.

The layout mathematical model can be expressed as follows:

$$\begin{aligned}
&\text{Min } F(x)\\
&\text{s.t. } g_1(x)=|x_{ic}|+|y_{ic}|+\frac{l_i+w_i}{2}-\frac{L+W}{2}\le 0\\
&g_2(x)=\begin{cases}|x_{ic}-x_{jc}|-\dfrac{p_{11}+p_{12}}{2}>0\\ or\\ |y_{ic}-y_{jc}|-\dfrac{p_{21}+p_{22}}{2}>0\end{cases}\\
&g_3(x)=\begin{cases}-a\le x_i\le a\\ -b\le y_i\le b\end{cases}\\
&g_4(x)=\begin{cases}\dfrac{w_{gl}+w_z}{2}-a\le x_{gl}\le\dfrac{w_{gl}+w_z}{2}+a\\ -b\le y_{gl}\le b\end{cases}\\
&g_5(x)=\begin{cases}-\dfrac{l_{gz}+w_z}{2}-a\le x_{gz}\le-\dfrac{l_{gz}+w_z}{2}+a\\ -b\le y_{gl}\le b\end{cases}\\
&g_6(x)=\begin{cases}-a\le x_{sx}\le a\\ -b+\dfrac{l_z+w_{sx}}{2}\le y_{sx}\le b+\dfrac{l_z+w_{sx}}{2}\\ \text{or}\\ -b-\dfrac{l_z+w_{sx}}{2}\le y_{sx}\le b-\dfrac{l_z+w_{sx}}{2}\end{cases}
\end{aligned} \tag{8}$$

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 the process constraints. All meanings of mathematical quantities can be found in the literature [23].

4.2. *Numerical examples*

In order to verify the effectiveness of the improved Genetic algorithm, a specific example is used to verify result. The experimental data come from the literature [23]. The length of the deck is 78 meters, the width of the deck is 70 meters, and the size and weight of the each module is shown in Table 5. It is notable that the location of the drill floor area center coincides with the drilling well location.

Table 5. 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 NO.1	9.60	8.33	1.63	188.36
4	drill pipe area NO.2	9.60	8.33	1.63	188.36
5	30in drive pipe area	12.19	2.63	2.63	60.46
6	20in drive pipe area	12.19	3.35	3.35	99.64
7	13-3/8in drive pipe area	12.19	3.90	3.90	115.96
8	9-5/8in drive pipe area	11.99	6.72	5.89	575.93
9	7in 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	18	16.1	2	115.39
16	living quarters	38.00	10.70	11.00	342

To compare the performance of the proposed genetic algorithm with that of other GAs, several contrasting groups are listed in this section.

First, three different GAs (POGA, EGA and TDGA) were used to solve the mathematical model and the numerical examples that are referred to in section 4.1 and section 4.2. The algorithm parameter is similar to the parameter in the literature [23]. 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 are 0.70 and 0.35, respectively; and the top limitation $P_{m\max}$ and the bottom limitation $P_{m\min}$ of crossover probability are 0.10 and 0.05, respectively. The concrete solving process was implemented using MATLAB 2012a 300 times. The simulation results are listed in Table 6, where 'Best' means the optimal value of centroid transverse deviator, 'Worst' means the worst optimal value of centroid transverse deviator, 'Mean' means the average value of centroid transverse deviator running 300 times, 'MCI' means Minimum convergence iteration, and 'ACI' means the Average convergence iteration.

Table 6. Simulation results with different GAs

centroid transverse deviator	POGE	EGA	TDGA	TD-SAGA
Best	0.177	0.185	0.125	0.120
Worse	0.567	0.774	0.353	0.348
Mean	0.284	0.301	0.189	0.183
MCI	327	296	241	155
ACI	418	457	367	306

As we can see in Table 6, the "Best", "Worse", and "Mean" of TDGA are superior to POGA and EGA. In addition, the convergence speed of TDGA is better than POGA and EGA. Compared to TDGA which is less self-adaptation crossover and mutation operator, the accuracy of the solution of TD-SAGA has a little promotion, but major advances can be found in the solution speed and solution efficiency. Therefore, we conclude that the improved genetic algorithm TD-SAGA is more efficient.

A special layout case shown in Fig.3 was selected from the computational results.

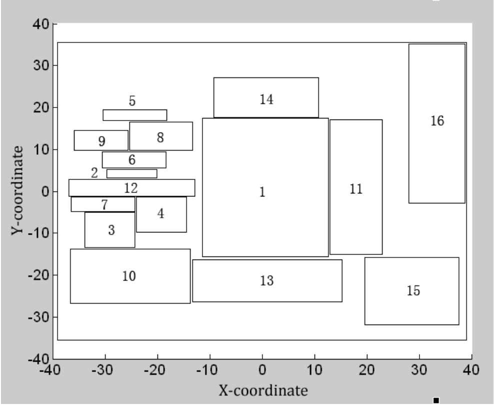

Fig.3. Special case of optimal layout scheme

In Fig.3, the coordinate planes represents the layout plane of the platform, and the coordinate center represents the drilling center. The centroid transverse deviator of the special case in Fig.3 is 0.120m, and the allowable value of centroid transverse deviator is 0.7m.

5. Conclusions

To improve the performance of the existing genetic algorithm, this article introduce a new improved GA that includes two major improvements:

1. A new genetic operator called trisecting group and directional selection mechanism (TDGA) is recommended. In TDGA, the highest fitness of $N/3$ parent individuals is genetically manipulated to reproduce offspring, and another $N/3$ manipulated parent individuals are selected randomly from the remaining $2N/3$ parent individuals. Simulation results based on 10 test functions show that TDGA is feasible and effective.

2. A new self-adaptation adjusting tactics of crossover operator and mutation operator (SAGA) is proposed so that individuals with higher fitness cross each other with smaller values of crossover probability, and individuals with lower fitness cross each other with greater values of crossover probability. Therefore, individuals with different fitness levels show different crossover probabilities and mutation probabilities, and the convergence rate of the population is guaranteed.

Finally, TD-SAGA is used to solve layouts for deep-water, semi-submersible platform drilling equipment. Combined with numerical examples, the results indicate that the best centroid transverse deviator is only 0.120 m, which is far less than the allowable value.

Acknowledgements

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), and Project of scientific and technological achievements of Jiangsu province “Research and industrialization of the key techniques of drilling string used in marine deep water”.

References

1. J. F. Gonçalves and M. G. C. Resende, A biased random-key genetic algorithm for the unequal area facility layout problem, *European Journal of Operational Research*. **246**(1) (2015)86-107.
2. R. D. Meller, W. Chen, and H. D. Sherali, Applying the sequence-pair representation to optimal facility layout designs. *Operations Research Letters*. **35**(5)(2007)651-659.
3. K. Y. Gau and R. D. Meller, An iterative facility layout algorithm, *International Journal of Production Research*. **37**(16) (1999)3739-3758.
4. Y. A. Bozer and C. T. Wang, A graph-pair representation and MIP-model-based heuristic for the unequal-area facility lay out problem, *European Journal of Operational Research*. **2**18(2) (2012)382-391.
5. M.T. Niemier and P.M. Kogge, Problems in designing with QCAs: Layout equals timing, *International Journal of Circuit Theory and Applications*. **29**(1)(2001)49-62.
6. J. Diaz, J. Petit and M.Serna, A survey of graph layout problems, *ACM Computing Surveys*. **34**(3)(2002)313-356.
7. J. H. Holland, *Adaptation in Natural Artificial Systems*. (Ann Arbor: The University of Michigan Press,1975).
8. Z. H. Zhan, J. Zhang, Jun and Y. Li, Adaptive Particle Swarm Optimization, *IEEE Transactions on Systems Man and Cybernetics Pate B-Cybernetics*. **39**(6)(2009)1362-1381.

9. M. Dorigo, V. Maniezzo and A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems Man and Cybernetics Pate B-Cybernetics*. **26**(1)(1996)29-41.
10. M. K. Mayer, A network parallel genetic algorithm for the one machine sequencing problem, *Computers and Mathematics with Applications*. **37**(1999) 71-78.
11. A. Konaka, D. W. Coitb and A. E. Smithc, Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering and System Safety*. **91**(9)(2006) 992–1007.
12. K. Deb, A. Pratap and S. Agarwal et al, A fast and elitist multi objective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*. **6**(2)(2002) 182-197.
13. G. Renner and A. Ekart, Genetic algorithms in computer aided design, *Computer-aided Design*. **35**(8)(2003)709-726.
14. L. M. Schmitt, Theory of genetic algorithms, *Theoretical Computer Science*. **259**(1)(2001) 1-61.
15. F. Herrera, M. Lozano and A.M. Sanchez, A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study, *International Journal of Intelligent Systems*. **18**(3)(2003)309-338.
16. R. Ruiz, C. Maroto and J. Alcaraz, Two new robust genetic algorithms for the flowshop scheduling problem, *Omega-International Journal of Management Science*. **34** (5)(2006)461-476.
17. B. B. Li and L. Wang, A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling, *IEEE Transactions on Systems Man and Cybernetics Pate B-Cybernetics*. **37**(3)(2007)576-591.
18. Z.B. Xu, Z.K. Nie, W.Z. Zhang, Almost sure strong convergence of a class of genetic algorithms with parent-offsprings competition, *Acta Mathematicae Applicatae Sinica*. **25**(1) (2002)167-175.(in chinese)
19. C A C. Coello, An Updated Survey of GA-Based Multi-objective Optimization Techniques, *ACM Computing Surveys*. **32**(2)(2000)109-140.
20. L. He, K.J. Wang, G.B. Li, et cl, Elitist preserved genetic algorithm and its convergence analysis, *Control and Decision*. **15**(1)(2000)63-66.
21. M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics.* **24**(4) (1994)656-667.
22. S. Shi, Q.F. Li, X.H. Wang, Design Optimization of Brushless Direct Current Motor Based on Adaptive Genetic Algorithm, *Journal of Xi'An JiaoTong University*. **36**(12)(2002)1215-1218.(in chinese)
23. K.K. Wei, Research on Layout Design of Drilling Rig System on Deep-water Semi-submersible Drilling Platform, (Qingdao China: China university of petroleum(Huadong), 2010).(in chinese)