Journal of Marine Science and Technology

Volume 18 | Issue 3 Article 15

GENETIC ALGORITHM-BASED CHAOS CLUSTERING APPROACH FOR NONLINEAR OPTIMIZATION

Min-Yuan Cheng
Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C

Kuo-Yu Huang
Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C, as09@ms31.hinet.net

Recommended Citation
Cheng, Min-Yuan and Huang, Kuo-Yu (2010) "GENETIC ALGORITHM-BASED CHAOS CLUSTERING APPROACH FOR NONLINEAR OPTIMIZATION," *Journal of Marine Science and Technology*: Vol. 18 : Iss. 3 , Article 15.
DOI: 10.51400/2709-6998.1891
Available at: https://jmstt.ntou.edu.tw/journal/vol18/iss3/15

GENETIC ALGORITHM-BASED CHAOS CLUSTERING APPROACH FOR NONLINEAR OPTIMIZATION

Min-Yuan Cheng* and Kuo-Yu Huang*

Key words: chaos, optimization, K-means clustering, genetic algorithm.

ABSTRACT

In this paper, a n-dimension convergence algorithm was employed to track the potential trend of evolution in traditional genetic algorithm (GA) by K-means clustering technique. And, chaotic algorithm was exploited to prevent the new approach from premature. By means of the proposed approach, not only the basic search capability was maintained but also the flexibility and efficiency of parametric modeling were improved.

The main purpose of the paper is to demonstrate how the GA optimizer can be improved by incorporating a hybridization strategy. Experimental studies revealed that the hybrid chaotic approach with genetic algorithm (CGA) procedure could produce much more accurate estimates of the true optimum points than other optimization procedures. Furthermore, including K-means clustering into CGA, named KCGA, exhibited superior convergence performance than other algorithms. And, the proposed approach, KCGA, had 84 percent of probability to get optimized. On the whole, the new approach was demonstrated to be extremely effective and efficient at locating optimal solutions and verified by an empirical example from construction.

I. INTRODUCTION

The well known genetic algorithms (GA) were introduced by Holland in 1970s as optimization approaches. To find a global or near-global optimal solution, the search by GA was a group base instead of the point-to-point search. The main concept of this approach was derived from biological evolution in a competitive environment [12]. Nowadays, many industrial applications have been developed with the aid of this tool [15]. For instance, Davies [4] proposed a genetic algorithm to generate an optimal (shortest distance) path plan, and successfully guided an actual X80 mobile robot to all its waypoints without colliding with any obstacles in a test environment. And Fung [8] have developed the extended hybrid genetic algorithm (EHGA) to solve nonlinear programming (NLP) problems with equality and inequality constraints.

At the meantime, GA can deal with the problem which has highly nonlinear objective function and upper and lower limits of variables [30]. In 2001, Lu and Fang [21] proposed a genetic algorithm to solve a nonlinear single objective problem with fuzzy relation equation constraints. In fact, GA is highly parallel randomly searching algorithms that imitate the life evolution as proposed in Darwinian survival of the fittest principle [11, 17]. Critical genetic operations such as the encoding of the solution of optimizing problem, the designing of the fitting function according to its application, and the crossover and mutation for offspring, play important roles in GA [12, 36].

The population diversity of GA will be greatly reduced after some generations, and may lead to a premature convergence to a local optimum. Actually, GA with excellent capabilities solves difficult nonlinear optimization problems [9]; nevertheless, it tends to take long running time to converge prematurely and the optimization may get stuck at a local optimum. For example, the population is not always sufficiently huge in size to typical GA problem solving within limited iterations or times. In order to overcome these flaws, the key point is to maintain the population diversity and prevent the incest leading to misleading local optima [5, 33]. At the mean time, an efficient convergence over optimization search is needed.

To maintain the population diversity of GA, the chaos procedure was introduced in this paper. Chaos being radically different from statistical randomness, especially the inherent ability to search the space of interest efficiently, can improve the performance of optimization procedure. Chaotic motion can be considered as an irregular motion, seemingly unpredictable random behavior under deterministic conditions. Random and chaotic motions should be distinguished here by their features. The former was reserved for problems in which

Paper submitted 02/03/09; revised 07/17/09; accepted 07/19/09. Author for correspondence: Kuo-Yu Huang (e-mail: as09@ms31.hinet.net).
*Department of Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.

to know the input forces were not necessary, but some statistical measures of the parameters were enough; however, chaos is reserved for deterministic problems in which there are no random or unpredictable inputs or parameters.

In chaos, a small difference in the initial conditions may produce an enormous error in the final phenomena. It is extremely sensitive to the initial conditions, and its property sometimes referred to as the instability in the so-called butterfly effect or Liapunove's sense [16, 20]. Sensitive dependence on initial conditions was often exhibited by multiple elements with nonlinear interactions in the systems. Owing to chaos characteristic, the system could be designed as an efficient approach for maintaining the population diversity in the problem of interest.

An efficient convergent approach was founded by integrating moving centers into population evolution to speed up optimization search in GA. To locate these moving centers, a K-means clustering technique was employed in this study. Clustering is one of the most important and the most challenging of classifying algorithms. A successful clustering algorithm is able to reliably find true natural groupings in the data set. K-means is one of the well-known algorithms for clustering, originally known as Forgy's method [7]. K-means is famous for its simplicity and computational efficiency in clustering techniques. K-means clustering is the process of dispatching a set of objects into groups or clusters of similarities. Objects collected in the same cluster have similar features, but others are not [10].

In this study, chaotic algorithm was for population diversity while K-means clustering technique was for population grouping. The former would contribute to locate the optimum points, and the latter would diminish iteration runs of GA significantly.

The remainder of this paper was organized as follows. In Section II, gave an overview of the theorem and algorithm which would be encountered in this study later. In Section III, a K-means clustering algorithm for chaos GA was presented. And Section IV, KCGA was employed to search the optimization solution of a construction management issue. Section V provided some concluding remarks.

II. LITERATURE REVIEW

1. Chaos and Logistic Mapping

Chaotic mappings can be considered traveling particles within a limited range occurred in a deterministic nonlinear dynamic system. There is no definite regularity for such a traveling path. Such a movement is very similar to a random process, but extremely sensitive to the initial condition. Chaotic dynamic mappings have been defined as noninvertible mappings of the (0, 1) interval onto itself. Logistic mapping [6, 24] is one of the most important chaotic dynamic mappings which defines the simplest mapping for studying the period-doubling bifurcation (vide infra). In the well-known logistic equation [24]:

$$X_{n+1} = f(\mu, X_n) = \mu X_n (1 - X_n) \tag{1}$$

In which μ stands for a control parameter, X for a variable and $n = 0, 1, 2, 3....$ It is easy to find that (1) is a deterministic dynamic system. The variable X is also called as chaotic variable. The basic characteristic of chaos can be presented by (1), for a very small difference in the initial value of X will cause large difference in its long-term behavior.

The variation of control parameter μ of (1) will directly impact the behavior of X greatly. Usually, [0, 4] has been defined as domain area of control parameter μ. Different value in domain area of μ will determine whether X stabilizes at a constant size or behaves chaotically in an unpredictable pattern. The track of chaotic variable looks like in disorder; however, it can travel ergodically over the whole space of interest especially under the condition of $\mu = 4.0$. Then, a tiny difference in initial value of the chaotic variable will result in considerable differences of the values of chaotic variable later. Generally, there are three primary characteristics of the variation of the chaotic variable, i.e. ergodicity, irregularity and pseudo-randomness [1, 18, 28].

Logistic equation as shown in (1) can be distinguished by four intervals in accordance with the value of μ. First, when the value of μ is smaller than 1.0, the chaotic variable X_{n+1} converges to a stable point 0.0. Then, if the value of μ is between 1.0 and 3.0, no mater what initial value for X_0 between 0.0 and 1.0 was taken, X_{n+1} would converge to a certain value between 0.0 and 0.63665. And, the bifurcation occurs from $\mu \geq 3.0$. The system will enter the chaos domain, if μ reaches a critical point of 3.5699456…. Finally, when $\mu = 4.0$ the values of X_{n+1} will take any real numbers between 0.0 and 1.0 and no redundant value will present again while having turned up already. In this study, '$\mu = 4.0$' was taken to have the features of diversity during evolution.

2. Conventional GA and Legalization

Genetic algorithms are designed by randomized search and optimization techniques. The principles of evolution and natural genetics are built in functions to GA accompanied with a large amount of implicit parallel features. GA contains a fixed-size population of potential solutions over the search space. The idea population can be created by an objective or fitness function or base on the domain knowledge of GA. These potential solutions are named individuals or chromosomes. GA consists not only of binary strings-individuals but other encodings are also possible. For instance, in the literature [25, 35], a real-coded GA was proposed and the individual vector was coded as the same as the solution vector. And Chang [3] also applied a real-coded genetic algorithm to the system identification and control for a class of nonlinear systems. The evolution usually starts from a population of randomly generated individuals and continued by selection, crossover and mutation in iterations.

In typical GA iterations, a new population is created and based on the following four steps:

(1) Evaluation: each individual of the population will be evaluated and assigned a value derived from fitness function.
(2) Selection: individuals with higher fitness value will be more likely to be selected for next generation. Here, a competitive strategy was used to selection to improve its performance.
(3) Crossover: the crossover process is to choose two individuals as parents randomly. This study applied one-point crossover process in which the point was randomly selected in the list of fields. All the fields lying after this point were exchanged between the two parents to create two new offspring.
(4) Mutation: in this study, the mutation process was a probability-based procedure in which the random mutation operator selected a gene as starting point. Then, all points would be connected together along the nearest path. And, a correction action was taken to assure individuals meeting the legal requirements, in case of necessary.

The above four steps of iterations will hold in genetic algorithms until a satisfactory solution is found or the terminating criterion is met.

Nanayakkara [26] had successfully found out the shortest, least congested route by a GA based route planning algorithm. In 2008, Liang [19] designed a new immune genetic algorithm based on elitist strategy to get the shortest path of China's 31 provincial capital cities. Furthermore, GA can deal with the multi-depot vehicle route problem whose objective was to minimize both traveling cost and vehicle acquisition cost [31]. In this study, the designed operation rule for GA was to visit every node once and move back to the same point. Once the new generated offspring did not follow the designed rule, a correction work would be necessary. Thus, a new offspring would compare with the swapped and original portion to verify if the members were identical. Unique members leaded to a sound crossover while duplicated members needed to be legalized. For instance, a one-point crossover operation was exhibited in Tables 1 and 2. The random selected points of field 3 and 2 are shown on the following two tables. After legalization, no duplicated code was allowed except terminal points.

To improve the performance of optimization search, GA should keep individuals scattered in the whole searching space. Integrating GA with chaotic process, named chaos genetic algorithms (CGA), was proposed in this work and would be improved by incorporating clustering techniques. CGA held both advantages of GA and the chaotic process, and therefore could assure the individuals distributed ergodically in the defined space and avoided from premature. CGA also took the inherent advantages of GA over convergence after combining the diversity features of chaotic process and hence to increase the probability to find out the global optimal solution.

Table 1. Legalization to crossover with unique members.

Parents	Selected field	Swapping	Operation	Offspring
1 2 3 $\boxed{5\ 4\ 1}$	Crossover on field 3	1 2 3 $\boxed{4\ 5\ 1}$	Equal to	1 2 3 4 5 1
1 3 2 $\boxed{4\ 5\ 1}$		1 3 2 $\boxed{5\ 4\ 1}$		1 3 2 5 4 1

Table 2. Legalization to crossover with non-unique members.

Parents	Selected field	Swapping	Operation	Offspring
1 2 $\boxed{3\ 5\ 4\ 1}$	Crossover on field 2	1 2 $\boxed{2\ 4\ 5\ 1}$	Legalization	1 3 2 4 5 1
1 3 $\boxed{2\ 4\ 5\ 1}$		1 3 $\boxed{3\ 5\ 4\ 1}$		1 2 3 5 4 1

3. Incorporating K-means Clustering Technique into GA

Clustering is the process of grouping a set of physical or abstract items into clusters by similar features. K-means is one of the well-known algorithms for clustering, and it has been employed extensively in various fields including exploring studies: such as data mining, statistical data analysis: such as customer relationship management, and other business applications.

The K-means clustering technique adopted in GA evolution, named KGA, could easily conduct an efficient convergence of GA. K-means clustering technique introduced in this study was intended to track the main stream of population movement during GA evolution. Each center of clusters could be treated approximately as one of the items in the main stream of evolution, and reserved for population as candidate individuals.

The K-means algorithm for clustering is based on the mean value of items in the group. It is suggested to assign each item to the cluster with the nearest centroid (mean) [23]. Generally, in this study the primary operating procedures for K-means clustering technique are presented as follows:

(1) Defining how many clusters are to be created.
(2) Randomly assigning initial items to different clusters.
(3) Assigning new items to the cluster whose location to centroid is the nearest (by Euclidean distance) and recalculate the centroid for the existing or updated clusters.
(4) Repeating Step (3) until no more reassigning.

III. PROPOSED K-MEANS CLUSTERING AND CHAOS IN GENETIC ALGORITHM

1. Combination Logic

Assume that the working individual of independent variables is denoted by x consisting of n elements. They are named and denoted by x_1, x_2, ..., x_n. Thus, a problem of searching minimum can be described as:

$$\text{Min } f(x_1; x_2; \ldots x_n)$$
$$\text{s.t. } x_i \in (a_i, b_i) \quad i = 1, 2, 3, \ldots, n \tag{2}$$

Function f is related to the value of dependent variables x, which is subject to be optimized. The lower and upper limit of x_i in function f are $[a_1, a_2, \ldots, a_n]$ and $[b_1, b_2, \ldots, b_n]$, respectively. The chaotic process can be defined through the following equation as the same as (1) [18, 24]:

$$cx_i^{k+1} = 4cx_i^{(k)}(1 - cx_i^{(k)}) \quad i = 1, 2, \ldots, n, \tag{3}$$

In which cx_i is the ith chaotic variable, and (k) and (k + 1) denote the number of iterations. Then a linear mapping function was used to convert chaotic variable to a certain interval. In this study the linear mapping function can be described as:

$$x_i^k = a_i + cx_i^{(k)}(b_i - a_i) \quad i = 1, 2, \ldots, n, \tag{4}$$

In which cx_i is the ith chaotic variable, x_i^k is the ith working variable, and (k) denotes the number of iterations. a_i and b_i are the lower and upper limits.

K-means clustering technique played a critical role in speeding up convergence of GA evolution while chaos algorithm could assure GA population diversity and avoid from premature. To take advantages of speeding convergence and global optima features in GA, a hybrid algorithm combined K-means clustering, chaos algorithms and genetic algorithms was proposed as a new algorithm named KCGA.

Initial population of KCGA should be generated from chaos algorithm, and then chaotic function would review the individuals after mutation with a decreasing probability to improve diversity in the beginning iterations and diminish impact to convergence at the end of evolution.

After chaos operator, K-means clustering in this study would help to group population in several clusters as predefined. Thus, each centroid of cluster would be treated as a candidate individual of population. A competing procedure was employed to eliminate individuals with lower fitness value, and reserved the others for creating appropriate population for KCGA.

Centroid of each cluster was derived from Euclid distance in population, and hence to locate in the center of cluster. During evolution each cluster centroid would keep migrating within population and therefore could create a track of centroid movement. This track could be treated as a potential trend of population centers movements directed by a certain rule of GA. Incorporating moving centers or tracks into population was an efficient way toward convergence in GA. The proposed approach was proved to be effective in the experiment later.

2. Migration Algorithm

During evolution, GA generated a certain rule to direct population migration. In particular, K-means and chaotic algorithms were exploited in GA to thoroughly explore the entire search space so that to point out the most possible migration way and potential individuals for conventional GA.

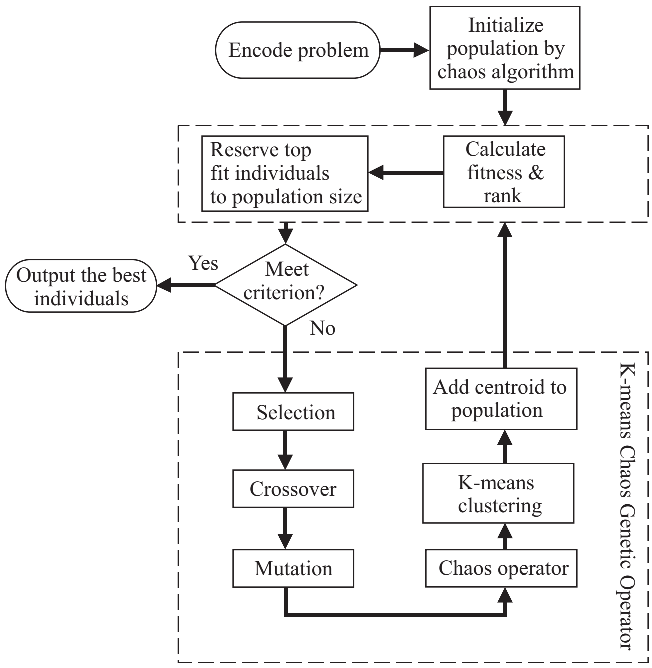

Fig. 1. Flow chart of K-means chaos genetic algorithm.

First, each individual initialized for GA population denoted a set of feasible solution by chaos algorithm. Second, given all individuals as input, the K-means clustering algorithm could group and locate the centroid of each cluster. Third, the new formed centroids of each cluster would convert to candidate individuals appending to the existing population. These new formed centroids also indicated the moving centers of current iteration. Fourth, fitness values of individuals were evaluated by a competing algorithm to keep enough individuals for next iteration. And, the flow chart of K-means chaos genetic algorithm is described as shown in Fig. 1.

3. Limitations of the Proposed Approach

Chaos is nonlinear in nature [22], and can help global searching by its diversity characteristic; however, it will cause much more computation time before getting convergent. To perform effectively, chaos algorithm employed in this study was a probability based function, that was, chaos function would be triggered with high probability in the beginning and decreasingly by iteration times. At the meantime, chaos function contributed global searching to GA evolution from initial, and reduced its affecting range by times to comply with GA convergent procedure at end.

K-means is a grouping technique which explores existing searching space, points out the centers of population groups and offers a short cut to convergence during GA evolution. Unfortunately, k-means efficient convergent feature could not improve heterogeneous population creation to enlarge the possible searching space for global optimization in this study.

Hybrid chaos and k-means with GA could combine their characteristics and merits together; therefore, a GA based global searching with effectively convergent could be reached. Apparently, it is a nonlinear approach to search in global space with less fit for linear problems.

IV. EXPERIMENTAL RESULT OF CONSTRUCTION MANAGEMENT

1. Background

Construction work includes many inherently hazardous conditions and tasks such as work at noise, dust, height, excavations, etc. For example, construction has about 6% of U.S. workers, but 20% of the fatalities - the largest number of fatalities reported for any of the industry sectors. These were announced by National institute for occupation safety and health (NIOSH) on Dec. 2008 [27].

In this study, a simulated case of a safety and health auditor patrolling model was built. An example of ten building-construction sites was employed in this case. The auditor should start from one of the building-construction sites and travel to every site before returning back to the same place to appraise their performances of safety and health in construction management. That is an optimum route problem which can be described as nonlinear mixed integer programming model [14]. The purpose of this example was to point out the shortest path along every construction site to meet the auditor patrolling model requirements in real construction cases.

After assigning each construction site a specific code, the distances between each site could be recorded in a lower triangular matrix as shown in Table 3. The fitness function was designed to calculate the total distance along the path. Any set of randomized code might stand for a different path. To comply with the real world, it was critical to legalize offspring during iterations of proposed algorithms, especially after the crossover and mutation procedure. The legalization example was exhibited in Section II.2.

In fact, it is a traveling salesman problem (TSP), and is a typical combinational optimization problem [32]. TSP is known as the classical combinatorial optimization problem. The basic concept of TSP is to find the shortest closed tour that connects a number of cities in a region. GA is used to find the optimal combination of these heuristic rules [13, 29]. Hopfield, Tank and other scholars applied Hopfield-model to solve TSP, the typical Nonlinear Programming (NP) problem of combination optimization [34]. Sometimes the developed nonlinear integer programming model of TSP was partially linearized and solved by enumerating a series of solutions of the TSP sub-problems [2]. In this study, KCGA, a GA based approach incorporating nonlinear chaos mechanism and K-means function, was proposed to solve the same problem with a satisfied performance.

Table 3. Distances between building-construction sites.

1	0									
2	6	0								
3	9	8	0							
4	7	7	11	0						
5	8	7	8	5	0					
6	9	8	9	6	7	0				
7	2	5	6	2	3	6	0			
8	7	9	2	7	12	3	6	0		
9	5	12	8	11	7	11	2	11	0	
10	11	6	3	8	6	6	8	7	2	0
Sites	1	2	3	4	5	6	7	8	9	10

Table 4. The performances of four models.

Item \ Model	KCGA		KGA		CGA		GA	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Optimization rate	0.84		0.70		0.74		0.70	
Iteration	43.1	9.4	32.7	31.3	61.5	43.0	56.3	53.4
Time (sec.)	5.6	1.9	3.2	3.0	9.7	9.4	5.4	5.8
Min.	38.1	0.3	38.3	0.4	38.2	0.4	38.3	0.5
Max.	38.1	0.3	38.3	0.5	42.0	11.0	39.3	4.6
Fitness	2289.6	22.2	2298.0	27.8	2300.7	37.4	2300.6	33.3

Notes: mutation rate = 0.01, crossover rate = 0.8,
population size = 60, generation limit = 150,
Avg.: Average, Std.: Standard Deviation

2. Solutions

All experiments were completed on Core 2 CPU T5500 @ 1.66GHz PCs with 2 GB memory. The results reported in Table 4 are all averaged over 50 independent runs. The parameters, such as mutation rate, crossover rate, generation limit, are given under the results.

From GA to KGA, a K-means clustering technique adopted by genetic algorithm can speed up its convergence rate. It is easy to find that the tremendous iterations decreases from GA (or CGA) to KGA in Table 4 listed as above. The result has strongly recommended that a speeding convergence of searching in ten-dimension space can be realized by K-means clustering technique effectively. While the number of iterations has sharply decreased, the KGA takes the shortest time, and maintains approximately the same optimization rate as conventional GA.

Chaos algorithm improved the probability of GA to get optimized; however, it needed more time and iterations to search for the optimal solution. Actually, chaotic diversity contributions to GA, from GA to CGA, provided a four percent optimization increase in current work. Furthermore, during KCGA verification, 42 from 50 independent runs got the shortest path leading to an optimization rate of 84 percent.

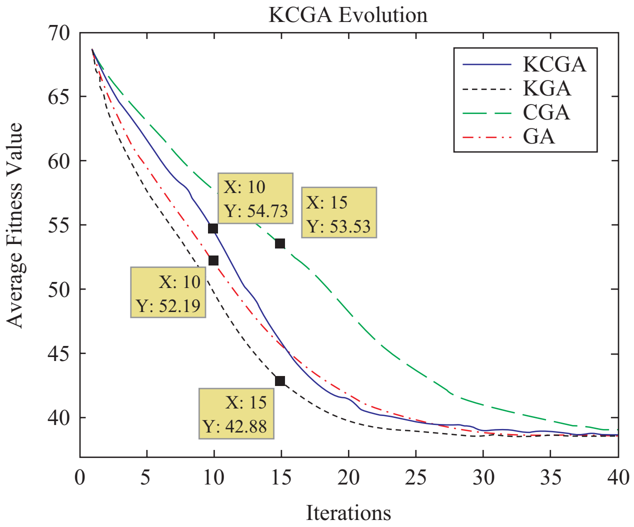

Fig. 2. Convergent curves of KCGA etc.

Combining K-means clustering technique with GA could assure to converge. It was shown that KCGA and KGA had never failed to converge during their experimental procedures for they had identical fitness values in minimum and maximum, defined by GA as a criterion of termination. KGA in this experiment held the lowest rate of 70 percent of getting optimized. But, after incorporating chaos algorithm into KGA, 84 percent of KCGA solutions could be optimized.

GA, integrated with K-means clustering technique and chaos algorithm, could promote its accuracy and reduce the converging time. Migrations from GA to KCGA, listed in Table 4, have shown that KCGA improves the accuracy of GA, and diminishes the amount of iteration runs significantly.

The mean performances of four optimization approaches were calculated and shown in Fig. 2. It can be seen from Fig. 2 that the KGA convergent capability is superior to others for its average fitness values always being the least at any iteration. And CGA is not as sensitive to the iteration runs as KGA and others, due to the fact that the mean performance of CGA with different operation probability fluctuates within a limited size. Although CGA has the worst convergent performance, it holds higher probability to get optimized than GA and KGA as shown in Table 4.

To illustrate the performances of optimization approaches with generation clearly, four specified points were used as indicated in Fig. 2. Thus, when comparing the performance of CGA with that of KGA at 15th iteration, CGA takes fitness value 53.53 and KGA takes 42.88 only. The same iteration with different average fitness value leaded to various convergent performances.

V. CONCLUSIONS

The proposed algorithms had effectively mitigated some drawbacks of traditional GA, such as long running time and getting trapped in local optima. Based on GA, the designed approach, KCGA, had a 14 percent probability increase to get optimization. Thus, 84 percent of KCGA solutions could be optimized. This approach, joined K-means clustering technique and chaos attributes based on genetic algorithm, had successfully conquered the underlying premature by diversifying population and reduced iteration times by tracking moving centers.

The proposed approach was not only to enhance the diversity of GA for more accuracy but also to extract clustering rules for achieving a potential moving track of evolution to improve the convergence performance. Owing to the similarities between some heuristic optimization algorithms, the proposed approach could be easily modified to fit various heuristic methodologies (e.g., particle swarm optimization, P.S.O.; ant colony optimization, A.C.O.) to improve their performances.

REFERENCES

1. Bountis, T., "Fundamental concepts of classical chaos. Part I," *Information Dynamics and Open Systems*, Vol. 3, No. 23, pp. 23-96 (1995).
2. Cao, D. and Chen, M. Y., "A mixed integer programming model for a two line CONWIP-based production and assembly system," *International Journal of Production Economics*, Vol. 95, Issue 3, pp. 317-326 (2005).
3. Chang, W. D., "Nonlinear system identification and control using a real-coded genetic algorithm," *Applied Mathematical Modelling*, Vol. 31, Issue 3, pp. 541-550 (2007).
4. Davies, T. and Jnifene, A., "Multiple waypoint path planning for a mobile robot using genetic algorithms," *Proceedings of 2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, CIMSA 2006*, art. No. 4016817, pp. 21-26 (2006).
5. Eshelman, L. J. and Schaffer, J. D., "Preventing premature convergence in genetic algorithms by preventing incest," *Proceedings of the 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, California, pp. 115-122 (1991).
6. Feigenbaum, M. J., "Quantitative universality for a class of nonlinear transformations," *Journal of Statistical Physics*, Vol. 19, pp. 25-52 (1978).
7. Forgy, E., "Cluster analysis of multivariate data: Efficiency versus interpretability of classifications," *Biomertrics*, Vol. 21, pp. 768 (1965).
8. Fung, R. Y. K., Tang, J. F., and Wang, D. W., "Extension of a hybrid Genetic Algorithm for nonlinear programming problems with equality and inequality constraints," *Computers & Operations Research*, Vol. 29, Issue 3, pp. 261-274 (2002).
9. García, S., Saad, M., and Akhrif, O., "Nonlinear tuning of aircraft controllers using genetic global optimization: A new periodic mutation operator," *Canadian Journal of Electrical and Computer Engineering*, Vol. 31, Issue 3, pp. 149-158 (2006).
10. Han, J. and Kamber, M., *Data mining: Concepts and techniques*, Morgan Kaufmann Publishers, San Francisco (2001).
11. Hibbert, D. B., "Genetic algorithms in chemistry," *Chemometrics and Intelligent Laboratory Systems*, Vol. 19, No. 3, pp. 277-293 (1993).
12. Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan, Ann Arbor, MI (1975).
13. Huang, H.-W., Lu, C.-H., and Fu, L.-C., "Lot dispatching and scheduling integrating OHT traffic information in the 300 mm wafer fab," *Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering*, IEEE CASE 2007, Article number 4341833, pp. 495-500 (2007).
14. Huang, Y., Liang, C., and Yang, Y., "The optimum route problem by

genetic algorithm for loading/unloading of yard crane," *Computers and Industrial Engineering*, Vol. 56, No. 3, pp. 993-1001 (2009).

15. Hugget, A., Sébastian, P., and Nadeau, J. P., "Global optimization of a dryer by using neural networks and genetic algorithms," *American Institute of Chemical Engineering Journal*, Vol. 45, No. 6, pp. 1227-1238 (1999).
16. Kim, J. H. and Stringer, J., *Applied Chaos*, Wiley, New York (1992).
17. Lavine, B. K., Ritter, J., Moores, A. J., Wilson, M., Faruque, A., and Mayfield, H. T., "Source identification of underground fuel spills by solid-phase microextraction/high-resolution gas chromatography/genetic algorithms," *Analytical Chemistry*, Vol. 72, pp. 423-431 (2000).
18. Li, B. and Jiang, W. S., "Optimizing complex functions by chaos search," *Cybernetics and Systems: An International Journal*, Vol. 29, pp. 409-419 (1998).
19. Liang, Y. and Yang, K., "Immunity genetic algorithm based on elitist strategy and its application to the TSP problem," *Proceedings - 2nd 2008 International Symposium on Intelligent Information Technology Application Workshop*, IITA 2008 Workshop 2008, art. no. 4731867, pp. 3-6 (2008).
20. Lorenze, E. N., "Deterministic nonperiodic flows," *Journal of the Atmospheric Sciences*, Vol. 20, pp. 130-141 (1963).
21. Lu, J. and Fang, S.-C., "Solving nonlinear optimization problems with fuzzy relation equation constraints," *Fuzzy Set and Systems,* Vol. 119, pp.1-20 (2001).
22. Lu, Z., Shieh, L.-S., Chen, G., and Coleman, N. P., "Adaptive feedback linearization control of chaotic systems via recurrent high-order neural networks," *Information Sciences,* Vol. 176, No. 16, pp. 2337-2354 (2006).
23. MacQueen, J. B., "Some methods for classification and analysis of multivariate observations," *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, California, pp. 281-297 (1967).
24. May, R., "Simple mathematical model with very complicated dynamics," *Nature,* Vol. 261, pp. 45-67 (1976).
25. Michalewicz, Z., Janikow, C. Z., and Krawczyk, J. B., "A modified genetic algorithm for optimal control problems," *Computer and Mathematical Applications*, Vol. 23, No. 12, pp. 83-94 (1992).
26. Nanayakkara, S. C., Srinivasan, D., Lup, L. W., German, X., Taylor, E., and Ong, S. H., "Genetic algorithm based route planner for large urban street networks," *2007 IEEE Congress on Evolutionary Computation,* CEC 2007, Article number 4425056, pp. 4469-4474 (2007).
27. "NIOSH safety and health topic: Construction." <http://www.cdc.gov/niosh/topics/construction/> (Dec. 10, 2008).
28. Ohya, M., "Complexities and their applications to characterization of chaos," *International Journal of Theoretical Physics*, Vol. 37, No. 1, pp. 495-505 (1998).
29. Pongcharoen, P., Chalnate, W., and Thapatsuwan, P., "Exploration of genetic parameters and operators through travelling salesman problem," *ScienceAsia,* Vol. 33, Issue 2, pp. 215-222 (2007).
30. Shi, F., Luo, Q., and Song, C. H., "Genetic algorithms for optimization of complex nonlinear system," *International Conference on Computer Science and Software Engineering,* Vol. 1, Article number 4721766, pp. 378-381 (2008).
31. Skok, M., Skrlec, D., and Krajcar, S., "The genetic algorithm method for multiple depot capacitated vehicle routing problem solving," *Proceedings of 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies,* Brighton, UK, pp. 520-526 (2000).
32. Sun, M., Zhao, L., Yan, C., and Yao, Q. X., "Wavelet chaotic neural network with nonlinear self-feedback and its application to traveling salesman problem," *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pp. 4537-4541 (2008).
33. Syswerda, G., "Uniform in genetic algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann, California, pp. 2-9 (1989).
34. Tan, Z. and Sang, H., "ANN-based combination optimization model and application in mineral industry," *Computer Applications in the Minerals Industries,* pp. 817-820 (2001).
35. Wright, A. H. and Rawlins, G. J., "Genetic algorithms for real parameter optimization," *Foundations of Genetic Algorithms*, Morgan Kaufmann, California, pp. 205-218 (1991).
36. Zhao, W. X., Chen, D. Z., and Hu, S. X., "Optimizing operating conditions based on ANN and modified gas," *Computers and Chemical Engineering*, Vol. 24, pp. 61-65 (2000).