K-means clustering and Chaos Genetic Algorithm for Nonlinear Optimization

Cheng Min-Yuan and Huang Kuo-Yu

Dept. of Construction Engineering, National Taiwan University of Science and Technology, Taipei, 106, Taiwan, R.O.C

Abstract

To reduce the computational amount and improve estimation accuracy for nonlinear optimizations, a new algorithm, K-means clustering with Chaos Genetic Algorithm (KCGA) is proposed, in which initial population are generated by chaos mapping and refined by competition. Within each iteration of this approach, in addition to the evolution of genetic algorithm (GA), the K-means Clustering algorithm is applied to achieve faster convergence and lead to a quick evolution of the population.

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 KCGA approach can produce much more accurate estimates of the true optimum points than the other two optimization procedures, the chaos genetic algorithm (CGA) and GA. Further, the proposed hybrid KCGA approach exhibits superior convergence characteristics when compared to other algorithms in this paper separately. On the whole, the new approach is demonstrated to be extremely effective and efficient at locating optimal solutions and verified by an empirical example from construction.

Keywords: Chaos; optimization; K-means clustering; Genetic Algorithm

1. Introduction

In order to find a global or near-global optimal solution, the search by Genetic algorithm (GA) was a group base instead of the point-to-point search. The group, which contains several solution points, is named population and is represented by P(t) with t denoting the number of generations. The well known GA was introduced by Holland in 1970s as optimization approach. The main concept of this approach was derived from biological evolution in a competitive environment (Holland, 1975). Nowadays, many industrial applications have been developed with the aid of this tool (Hugget, Se´bastian & Nadeau, 1999).

At the meantime, GA are highly parallel randomly searching algorithms that imitate the life evolution as proposed in Darwinian survival of the fittest principle (Hibbert, 1993; Lavine and Moores, 1999). 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 (Holland, 1975; Zhao, Chen & Hu, 2000).

The population diversity of GA would be greatly reduced after some generations, and might lead to a premature convergence to a local optimum. Actually, it tends 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. In order to overcome these flaws, the key point is to maintain the population diversity and prevent the incest leading to misleading local optima (Syswerda, 1989; Eshelmen and Schaffer, 1991).

To maintain the population diversity of GA, the concept of chaos is introduced in this paper. Chaos being radically different from statistical randomness, especially the inherent ability to search the space of interest efficiently, could improve the performance of optimization procedure. Chaos 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 is reserved for problems in which to know the input forces are not necessary, but some statistical measures of the parameters are enough. However, chaotic 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 (Lorenze, 1963; Kim and Stringer, 1992). Sensitive dependence on initial conditions is 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.

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 (Forgy, 1965). 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 (Han & Kamber, 2001). K-means is famous for its simplicity and computational efficiency in clustering techniques. As aforementioned Chaotic algorithm is for population diversity in GA, and K-means is for convergence efficiency in evolution. The former will keep the system accuracy, and the later will decrease iteration times of GA significantly.

The remainder of this paper is organized as follows. In Section 2, gives an overview of the theorem and algorithm which will be encountered in this study later. In Section 3, a K-means clustering algorithm for Chaos GA is presented. And Section 4, KCGA is employed to search the optimization solution of a construction management issue. Section 5 provides some concluding remarks.

2. Theorem and Algorithm

2.1The Chaotic Concept and Logistic Mapping

Chaos 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 (May, 1976; Feigenbaum, 1978) 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 (May, 1976):

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

in which μ stands for a control parameter, X for a variable and n =0,1,2,3…. It is easy to find that equation (1) is a deterministic dynamic system. The variable X is also called as chaotic variable. The basic characteristic of chaos could be presented by Eq. (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 Eq. (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 μ = 4. Then, a tiny difference in initial value of the chaotic variable would 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 (Bountis, 1995; Li & Jiang, 1998; Ohya, 1998).

Logistic equation as shown in equation (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 μ ≧ 3.0. The system will enter the chaos domain, if μ reaches a critical point of 3.5699456…. Finally, when μ =4.0 the values of X_{n+1} would take any real numbers between 0.0 and 1.0 and no redundant value would present again while having turned up already. In this study, 'μ =4.0' was taken to have the advantages of diversity during evolution.

2.2 The concept of GA and CGA

Genetic algorithms (GA) 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 could be created by an objective or fitness function or based 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 (Wright, 1991; Michalewicz, Janikow & Krawczyk, 1992), a real-coded GA was proposed and the individual vector was coded as the same as the solution vector. The evolution usually starts from a population of randomly generated individuals and continued by selection, crossover, mutation in each iteration.

In every iteration, 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 was to choose two individuals as parents randomly. This study applies one-point crossover process in which the point is randomly selected in the list of fields. All the fields lying after this point was exchanged between the two parents to create two new offspring.

(4) Mutation: The mutation process is a probability-based procedure in which a heuristic operation was employed to find shortest path from a random point. Then, a correction action is taken to keep individuals meeting the legal requirements, in case of necessary.

The above four steps are iterated in this study until a satisfactory solution is found or the terminating criterion is met.

In this study, while a crossover has finished, the new generated offspring may not follow the designed rule to visit every node once and move back to the starting point. A new offspring will compare with the swapped and original portion to verify if the members are identical. Same members lead to a sound crossover while duplicated members with parents need to be legalized. For instance, a one-point crossover was introduced; the random selected point of field is 3 and 2 shown on the following two tables.

Table 1 Legalization to crossover with identical members

Parents	Selected field	Swapping	Operation	Offspring
1 2 3 [5 4 1]	Crossover on field 3	1 2 3 [4 5 1]	Equal to	1 2 3 4 5 1
1 3 2 [4 5 1]		1 3 2 [5 4 1]		1 3 2 5 4 1

Table 2 Legalization to crossover with un-identical members

Parents	Selected field	Swapping	Operation	Offspring
1 2 [3 5 4 1]	Crossover on field 2	1 2 [2 4 5 1]	Legalization	1 3 2 4 5 1
1 3 [2 4 5 1]		1 3 [3 5 4 1]		1 2 3 5 4 1

To improve the performance of GA search, it should keep individuals scattered in the whole searching space. After adopting the nature of the chaotic process, a new GA search method was formed. Chaos-Genetic Algorithms (CGA), integrating GA with chaotic variable, was proposed in this work and would be improved by incorporating clustering techniques later. CGA holds both advantages of GA and the chaotic variable. It can keep the individuals distributed ergodically in the defined space and avoid from the premature of generations. And, CGA also takes the inherent advantage of GA over convergence to overcome the randomness of the chaotic process and hence to increase the probability of finding the global optimal solution.

2.3 The K-means Clustering concept in 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 Custom Relationship Management, and other business applications. 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) (Mac- Queen, 1967). In general, in this study the primary operating procedures for K-means 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 with either standardized or un-standardized observations) and re-calculate the centroid for the existing or updated clusters.
(4) Repeating Step (3) until no more reassigning.

3. Proposed K-means clustering and Chaos in Genetic Algorithm

Assume that the working individual of independent variables is denoted by x consisting of n elements. They are named and denoted by x1, x2, . . ., xn . Thus, a problem of searching minimum could be described as:

Min f (x1; x2; . . . xn)
s.t. xi∈(ai, bi) i=1, 2,3, . . . , n (2)

Function f is related to the value of dependent variables x, which is subject to be optimized. The lower and upper limit of xi in function f are [a1, a2, ..., an] and [b1, b2, …, bn], respectively. The chaotic process could be defined through the following equation as the same as Eq. (1) (Li & Jiang, 1998; May, 1976):

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

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 could be described as:

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

in which x_i^{k+1} is the *i*th working variable, and (k) and (k+1) denote the number of iterations. a_i and b_i are the lower and upper limits.

K-means plays a critical role in convergence of GA. Chaos algorithm can keep GA population diversity and avoid from premature. To take advantages of the above described benefits in GA, a novel algorithm combined K-means clustering and the Chaos algorithms with genetic algorithms was proposed as a powerful hybrid algorithm called KCGA (K-means and Chaos in Genetic Algorithm). Initial population of KCGA should be generated from chaotic algorithm, and then chaos function would adjust the individuals after mutation with the same proability. After mutation, K-means clustering in this study will help to group population in several clusters as pre-defined. Thus, location information of each centroid of cluster would be treated as candidate individuals for next generation. A competing procedure was employed to eliminate lower fitness value individuals, and reserved the others to create formal population for KCGA iteration.

During the convergence, GA generates a certain rule to direct population's migration. In particular, K-means was used with GA to thoroughly explore the entire search space so that to find out the most possible migration way and potential individuals for conventional GA. First, each individual in a population of GA denotes a set of feasible solution generated by chaos algorithm. Second, given all individuals as input, the K-means clustering algorithm can locate the centroid of each cluster. Third, the new formed centroids of each cluster will convert to candidate individuals appending to the existing population. These new formed centroids also indicate the moving centers of current iteration. Fourth, fitness values of individuals are evaluated and by a competing algorithm to keep enough individuals for next iteration. And, the flow chart of K-means Chaos Genetic Algorithm is described as following in figure 1.

4. Experimental result

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) in 2008. In this study, a simulated case of ten building-construction sites was used for an auditor of safety and health. The auditor should start from one of the building-construction sites and travel to every site before returning back to the same place. The target is to find out the shortest path along every construction site.

After assigning each construction site an integer number, the distances between each site could be recorded and create a matrix. The fitness function was designed to calculate the total distance along the path. Any set of random integer number within [1, 10] may stands for a different path. To comply with the real world, it is critical to legalize offspring during the KCGA iteration, especially after the crossover and mutation procedure. Chaotic algorithm will impact the visit priority of each site. However, each centroid of cluster derived from K-means may not be integer. They need to re-legalize again to get their integer sequence as one of the quasi individuals for next generation. All experiments are completed on Core 2 CPU T5500 @ 1.66GHz PCs with 2GB memory. The results reported are all averaged over 50 independent runs. The parameters, such as mutation rate, crossover rate, generation limit, are given under the results.

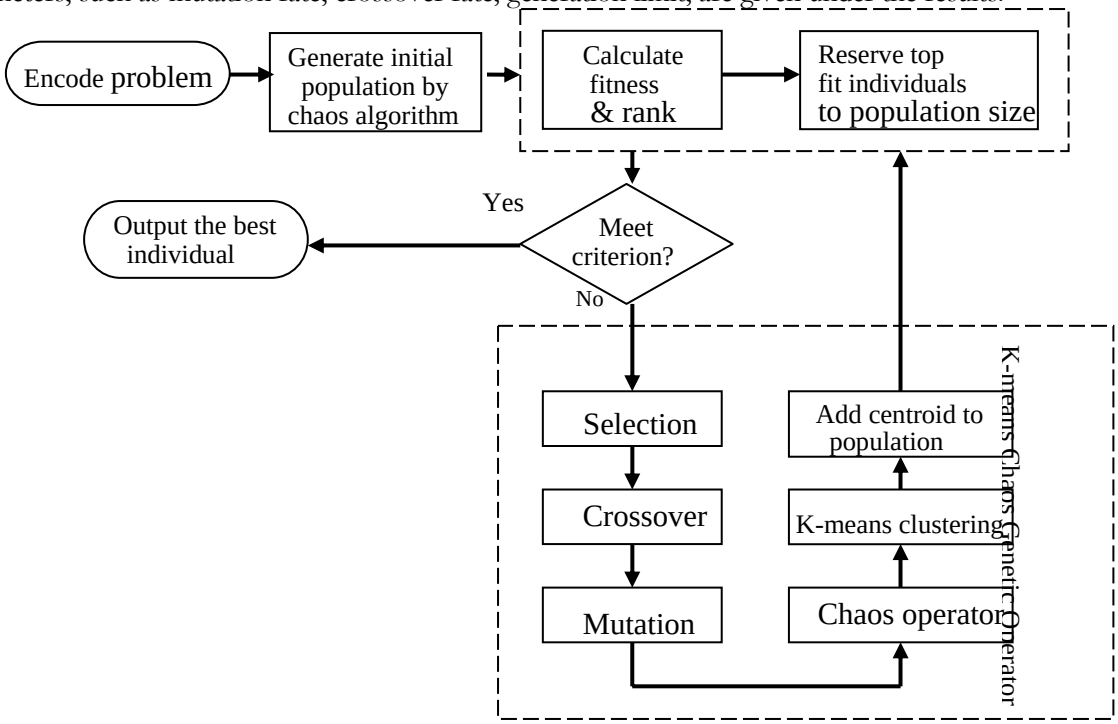

Fig. 1. Flow chart of K-means chaos genetic algorithm

From GA to KGA, a K-means clustering technique adopted by genetic algorithm can speed up its convergence rate by saving seventy percent of iterations and keep the accuracy over eighty percent. It is easy to find that the standard error of iteration has tremendously descended from GA (or CGA) to KGA in table 3 listed as above. The result has strongly recommended that a speeding convergence of searching in ten-dimension space can be smoothly realized by K-means clustering technique efficiently.

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 values in minimum and maximum, defined by GA as a criterion of termination.

GA, integrated with K-means clustering technique and chaos algorithm, could promote its accuracy and reduce the converging time. Migration from GA to KCGA, listed in table 3, has shown that KCGA improves the accuracy of GA, and diminishes the amount of evolution runs significantly.

Table 3 The performance of four methods

	KCGA (0.90)		KGA (0.80)		CGA (0.85)		GA (0.85)	
	Avg.	Std. E.	Avg.	Std. E.	Avg.	Std. E.	Avg.	Std. E.
Iteration	23.9	5.0	31.0	38.3	71.8	132.5	105.0	171.0
Time(sec.)	1.3	0.3	1.5	2.0	1.1	1.6	1.1	1.5
Min.	38.1	0.3	38.3	0.6	38.1	0.3	38.1	0.2
Max.	38.1	0.3	38.3	0.6	38.2	0.5	39.2	3.0
Fitness	2286.0	18.5	2295.0	33.0	2289.0	22.0	2291.0	26.8

Notes: mutation rate = 0.01, crossover rate = 0.8, population size = 60, generation limit = 500, Avg.: Average, Std. E.: Standard Error

(*) = accurate ratio

5. Conclusions

This study has proposed a procedure which joins K-means and chaos attributes based on genetic algorithm. The proposed procedure is not only to enhance the diversity of GA for more accuracy but also to extract clustering rules for achieving a potential trend of evolution. Additionally, it can effectively improve some drawbacks of traditional GA, such as long running time and getting trapped in local optima. Furthermore, this proposed procedure can really contribute to construction management in real world.

References

[1] Bountis, T. (1995). Fundamental concepts of classical chaos. Part I. Information Dynamics and Open Systems, 3 (23) 23_/96.

[2] Feigenbaum, M.J., (1978). J. Stat. Phys. 19, 25.

[3] Forgy, E. (1965). Cluster analysis of multivariate data: Efficiency versus interpreability of classifications. Biomertrics, 21, 768.

[4] Han, J., & Kamber, M. (2001). Data mining: Concepts and techniques. San Francisco: Morgan Kaufmann Publishers.

[5] Hibbert, D.B., (1993). Chem. Intel. Lab. Syst. 19, 277.

[6] Holland, J.H., (1975). Adaptation in Natural and Artificial Systems. University of Michigan, Ann Arbor.

[7] Hugget, A., Se´bastian, P., & Nadeau, J. P. (1999). Global optimization of a dryer by using neural networks and genetic algorithms. American Institute of Chemical Engineering Journal , 45 (6) 1227_/1238.

[8] Kim, J.H., Stringer, J., (1992). Applied Chaos. Wiley, New York.

[9] Li, B., & Jiang, W. S. (1998). Optimizing complex functions by chaos search. Cybernetics and Systems: An International Journal 29, 409-419.

[10] Lorenze, E.N., (1963). J. Atmos. Sci. 20, 130.

[11] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of 5th berkeley symposium on mathematical statistics and probability (pp. 281–297). Berkeley: University of California Press.

[12] May, R., (1976). Simple mathematical model with very complicated dynamics. Nature 261, 45–67.

[13] Michalewicz, Z., Janikow, C. Z., & Krawczyk, J. B. (1992). A modified genetic algorithm for optimal control problems. Computer and Mathematical Applications 23 (12), 83_/94.

[14] Ohya, M. (1998). Complexities and their applications to characterization of chaos. International Journal of Theoretical Physics , 37 (1) 495-505.

[15] Syswerda, G., (1989). Uniform in Genetic Algorithms, ICGA' 89. Kaufmann (Morgan), California, p. 2.

[16] Wright, A. H. (1991). In G. J. E. Rawlins (Ed.), Genetic algorithms for real parameter optimization. Foundations of genetic algorithms (pp. 205_/218). California: Morgan Kaufmann.
[17] Zhao, W. X., Chen, D. Z., & Hu, S. X. (2000). Optimizing operating conditions based on ANN and modified gas. Computers and Chemical Engineering 24, 61/65.