CHAOTIC OPTIMIZATION

Donald Davendra
Ivan Zelinka
Faculty of Applied Informatics
Tomas Bata University in Zlin
Czech Republic
{davendra, zelinka}@fai.utb.cz

Godfrey Onwubolu
School of Engineering and Physics
University of the South Pacific
Laucala Campus
Suva, Fiji Islands
onwubolu_g@usp.ac.fj

KEYWORDS
Evolutionary heuristics, chaos, optimization.

ABSTRACT

A novel approach symbolizing the principles of chaos and its application to evolutionary heuristics is discussed. The concept of local convergence and its symbolism to chaotic attractors is discussed. A novel approach of having a population driven evolutionary heuristic is then proposed combining the principle of chaotic attractors and edges.

INTRODUCTION

The creation of evolutionary systems has largely been in response to the explosion of complex tasks and applications. In an engineering sense, many problems now exist which require the use of advanced algorithms to find optimal and concise results. Problems in transportation, logistics, production and task scheduling, telecommunications and financial planning, have such large search spaces that it become computationally impossible to address all the search points within the search space (Onwubolu 2002). Scheduling problems requiring multiple machines such as the job shop family cannot be solved exactly with any known mean for over thirty sized problems (Yamada 2003). For the traveling salesman problem, some solutions would not be acquired in the lifespan of a human being. The theory of optimization evolved in order to solve these complex problems.

The theory of optimization encompasses the quantitative study of optima and the methods of finding them (Onwubolu 2002). In order to optimize a function the critical point to address is the optimal point which is the goal and the means of reaching the optimum. Traditionally, the main objective has been the *convergence criteria*; the finding of the optimal, regardless of the means. New research has delved into recognizing the *provisionary improvements* that drive the process of optimization towards the optimal.

In order to optimize, scientists observed naturally occurring optimization and phenomena and tried to mimic their attribute into artificial systems. New and emerging optimization techniques are usually classified into three distinct classes; natural phenomena, physical phenomena and mathematical-computational phenomena. Goldberg (1989) devised genetic algorithms (GA) through the observation of evolution. Glover (1989) created tabu search (TS) and scatter search (SS) with the hallmark of memory retention capabilities. Dorigo (1992) came up with the ant colony (ACS) approach after the observation of natural ants and their work in foraging for food in a natural setting. Simulated annealing (SA) mimics the heating and cooling of metals in order to optimize (Van Laarhoven and Aarts 1987). Differential evolution (DE) by Price (1999) and its discrete variant (Onwubolu and Davendra 2006) uses vector differentials to find optimal values in a search space.

This paper proposes a new generic evolutionary optimization technique for finding global minimal solutions. This approach looks at three critical issues in its operation. The first issue is the critical importance on initial conditions to the successful propagation of the population. The second is population dynamics which is included in the solutions, with regards to its interaction and behavior in the solution space. The population dynamics give rise to the third issue which is the attraction of variables within the population and its behavior which can be termed chaotic and random.

CHAOS IN EVOLUTIONARY ALGORITHMS

The emergence of chaotic systems was initially described by Lorenz (1993) and by Henon (1976). The two famous chaotic attractor bearing their names are the cornerstone of chaos theory in modern literature.

The emergence of chaotic nature has been described in EAs in multiple literatures, but in the terms of *local optima convergence*. Premature convergence towards local minima instead of a global minima and the subsequent “freeze” on evolution was described with the application of enhanced GA with evolved agents (Optimization and Automaton Group 2004).

Richter (2002) and Zelinka (2005, 2006) discuss the possibility of local control of chaos system using evolutionary algorithms. Chen and Huang (2005) outline how chaos can be introduced in an evolutionary system and its behavior observed. Extensive work has been done to manipulate the conversion schema of DE to introduce and describe chaos (Zhenyu *et al.* 2006). Zelinka (2006) investigated real time control of chaotic systems using evolutionary heuristics. Till date only chaos in systems is observed, however this research

Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

aims to establish a heuristic based on the principle of chaos attractor and population dynamics.

CHAOTIC OPTIMIZATION

Chaos states disorder and irregularities within a system. In order to enforce non-chaotic behavior, it is imperative to design a control of chaos. Two possibilities exist in order to accomplish a system that does not converge to an attractor or diverge to an edge as given in Figure 1.

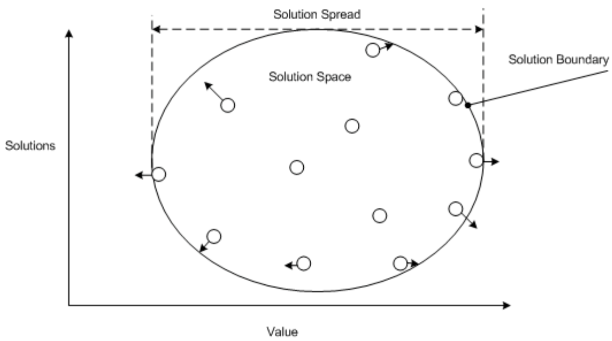

Figure 1: Conceptual diagram of attraction and edge

The first possibility is to detect whenever a chaotic system is about to arise and design a feedback system in order to bypass the chaotic region

In order to find the global minima, the population needs not converge, but stay robust. Robustness is critical in order to map the solution space. Even when the objective function has converged, the ordering of the individual solutions is diverse. Therefore the approach proposed is to keep the solutions diverse throughout the evolution, by generating a distance between the solutions spread instead of the objective function of the solution. Inorder to do this, intelligence has to be incorporated within the solutions. The overriding approach is to incorporate population dynamics within the solutions in order to organize a feasible propagation approach.

The processes required to have a controlled propagation is described in the following sections. The methodology introduces the approach in terms of discrete optimization, specifically permutation based as a means to describe the different processes.

Initial Population

Chaos theory stipulates that the emergence of chaotic behavior is invariably linked to initial conditions of the system. When observing all EA's, it becomes clear that little attention is paid to the initial conditions like population. The overriding approach is to have a population created using random generation, which the search heuristic will guide towards the global minima.

The fallibility of this approach is that a lot of emphasis is given to the random generator. Propagation will only occur, if a "good" starting point is achieved. If the initial population is very far away from the global minima, then a large number of generations would be required in order to find the correct route.

The first issue to be addressed in this research is the creation of the initial population. Using chaos theory as a guide, the initial conditions has to be such which can be mapped and its structure identified.

To archive this purpose, the population P is divided into four sub-populations (SP). The size of the population P_n which will be user defined is dived into four segments:

$$k = \left\lfloor P_n / 4 \right\rfloor \quad (1)$$

where k represents the size of each SP. The population can now be represented as a collection of four SP's.

$$P = \{SP_1, SP_2, SP_3, SP_4\} \quad (2)$$

Two SP's are created using a structured approach and the other two using random generation.

Structured Approach
As outlined previously, a structured approach is essential in order to have a control over the solution spread. Problems like traveling salesman problem (TSP) and quadratic assignment problem (QAP) amongst others have an in built structure and the solution also has to exhibit some structure, in order to find good solutions quickly.

Two SP are identified as having a structured population. One has a *forward approach*, while the other has a *backward approach.*

The *forward approach* takes the value from the lower order to the higher order. The SP is given as:

$$SP_1 = \{\delta_1, \delta_2, ..., \delta_k\} \quad k = \left\lfloor P_n / 4 \right\rfloor \quad (3)$$

where each solution δ is represented as:

$$\delta = \{x_1, x_2, x_3, ..., x_n\} \quad (4)$$

The user defined variable $\boldsymbol{n}$ represents the solution size.

The first solution is a direct ascend from the lower bound x^{lo} to the upper bound x^{hi}.

$$\delta_1 = \{x^{lo}, x^{lo+1}, ..., x^{hi}\} \quad (5)$$

In order to obtain a structured solution, the first solution is segmented and recombined in different orders to produce different combinations. The first segmentation occurs at $n/2$, and the two half's are swapped to produce the second solution. The second fragmentation occurs by the factor 3; $n/3$. The general representation is given as:

$$k \geq 1 + 2! + 3! + ... + z! \quad (6)$$

where z is the total number of permutations possible.

The second approach is the backward approach. In the backward approach the solutions are aligned with high x^{hi} to low x^{lo} order.

$$\delta_1 = \{x^{hi}, x^{hi-1}, \ldots, x^{lo}\} \quad (7)$$

Random Approach

It is very critical to observe that a random approach is only as good as the random generator used. A very good random generator is required to produce solutions which have a good spread. A random population is very easy to create. Simply generate a value between the lower x^{lo} and higher x^{hi} bounds:

$$x = rnd[x^{lo}, x^{hi}] \quad (8)$$

This value is checked against the values already in the solution and added to the solution if it is unique.

$$x \rightarrow \{x_1, x_2, \ldots., x_i\} \quad i \in \{1, 2, \ldots., k\}$$
$$if \quad x \notin \{x_1, x_2, \ldots., x_i\} \quad (9)$$

The two SP's are created using the random generation where each solution is unique.

POPULATION DYNAMICS

Chaos theory is based on two principles. The first principle is that simple systems will exhibit complex behavior which cannot be explained using conventional theory. The second principle is that complex systems will exhibit behavior which will seem random and unstructured, but it has an underlying order.

The application of this theory to EA is that EA by comparison with dynamic systems are simple systems, which exhibit complex behavior. So in order to understand complex behavior it is essential to have in built intelligence within the population.

Most systems have no intelligence within the population. GA, DE and Mematic Algorithm (MA) have no group dynamics where as SS, Particle Swamp Optimization (PSO) and ACS due to their memory adaptive programming (MAP) has some level of group dynamics. The core issue that is used with group dynamics is the issue of memory adaptive programming. Most emerging EA's like SS and PSO have inbuilt MAP, which is used to find good search space. SS has a singular reference set which it creates from the main population, which has intensified and diversified solutions. Using this approach, solution space is mapped to find better routes.

This proposed approach takes this application to the next level. SS uses a singular reference set to invoke MAP, where as this approach proposes four distinct SP's. The advantage is that small groups will have greater competition, in lieu of large groups. Also small grouping are easier to manage.

Operational Variables

The proposed algorithm proposes five distinct *operational variables* to order to invoke and expedite evolved agents, to enforce cooperative evolution and compute feasibility of solution.

Lifespan: Number o generation completed by solution

Propagate: Nmber of valid solutions created by a singular solution.

Successful propagate χ: Frequency with which the solution will be combined with another solution

$$\chi = \frac{\sum Propagate}{\sum Successful\ propagate} \quad (10)$$

Rank: The rank gives the standing of the solution in the SP. Rank is solely determined by the objective function.

Action: Three actions are defined; dormant, active and expired. Action is based on the performance of the solution in the SP. "Dormant" refers to an inactive solution, "Active" refers to a vaible solution and "Expired" refers to an inactive solution.

Individual Variables

In addition to operational variables which keep track of the feasibility of the solution within the population, individual characteristics of the solutions have to be mapped.

A new class of *individual variables* is defined in this proposal in order to map and keep track of individual variables. Each solution is computed in accordance to the difference between adjacent values. The *variance factor* ς defined here is given as:

$$\varsigma = \left(\frac{\sum_{i=1}^{n-1} |x_i - x_{i+1}|}{n} \right) \quad x_i \in \{x_1, x_2, \ldots, x_n\} \quad (11)$$

The ς gives the ordering of the values within the solution. The higher the factor the larger the spread of the values, and the more diverse the solutions.

$$\varsigma \leq 1.0 \quad for\ condensed\ solutions$$
$$\varsigma \geq 1.0 \quad for\ expanded\ solutions \quad (12)$$

The second individual variable defined is the *spread factor* ∂. The ∂ gives an overall identification as to the difference between individual solutions; however the hierarchy between adjacent values in a solution is not indicated. The spread factor s defined as:

$$\partial = \begin{cases} +1 & if\,(x_{i+1} - x_i) \geq 1 \\ -1 & if\,(x_{i+1} - x_i) \leq 1 \end{cases} \quad (13)$$
$$where \ i \in \{1, 2, \ldots., n\}$$

The conclusions drawn from the ∂ is given as:

$$\partial = \begin{cases} > 0 & backward\ spread \\ = 0 & uniform\ spread \\ < 0 & forward\ spread \end{cases} \quad (14)$$

The spread factor is critical in order to manage the propagation of the SP.

Group Variables

EAs are characterized by their population based approach, where "many are better than one" approach is used. Also a group of solutions offers in addition to a larger search space, a better probability of finding a global minimal solution. However as stated earlier, the emergence of unpredictable behavior hinders the propagation of the population. To study group behavior, it is essential to understand group characteristics. The three critical factors of *group variables* which are required are:

Average: The average value is the cumulative average of the SP in terms of the objective function. The formulation for the average value is given as:

$$SP_{avg} = \frac{\sum_{i=1}^{k} f(\delta_i)}{k} \quad where\ \ \delta_i \in \{\delta_1, \delta_2, ..., \delta_k\} \quad (15)$$

Range: The range of the SP is given as the difference between the highest objective function and the lowest objective function given as:

$$SP_{range} = f\left(\delta^{(hi)}\right) - f\left(\delta^{(lo)}\right) \quad (16)$$

Deviation: The *spread* of the solution in the solution space which is given as:

$$SP_{dev} = \frac{\sum_{i=1}^{k}\left(f(\delta_i)^2\right) - \left(\sum_{i=1}^{k} f(\delta_i) \bullet \sum_{i=1}^{k} f(\delta_i)\right) \Big/ k}{k-1} \quad (17)$$

The deviation is the most important feature since it is used to control the solution space in order to avoid the chaos edges. Deviation controls the solution spread as it defines the range a solution occupies. According to the deviation of the SP, new solutions are incorporated in the SP, and old solutions discarded. This process is further discussed in the following section.

POPULATION PROPAGATION

The key issue in all optimization techniques in the issue of propagation of the population from one generation to the next. The primary issue in this approach is not the propagation, but the avoidance of chaotic features in the population as it's propagates from one generation to the next.

Two chaotic features are identified in EA's; chaotic attractors and chaotic edges. The attractors pull together the solutions around a common value, whereas the edges tend to pull solutions away from the minima and into infeasible paths. A schematic of the chaos features is given in Figure 2.

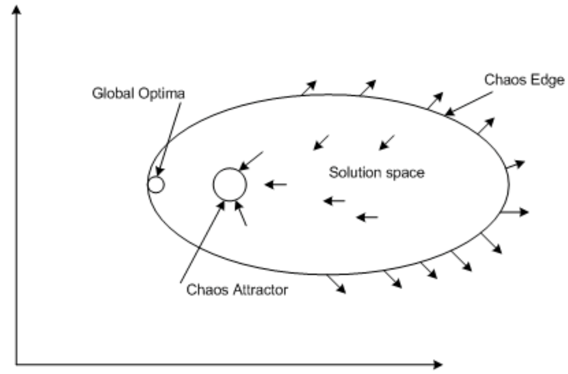

Figure 2: Chaos Features.

In order to drive the population and avoid chaos, it is essential to have population control.

When the population is generated, all the solutions are mapped and evaluated. The solutions are assigned according to their rank in the SP. By evaluating the group variables, it then becomes possible to compare each SP with each other in order to see if the problem is structured or random based on their average values. The *problem classification* φ is set as structured α if the problem is classified as structured or set as random β if classified as random.

$$\varphi = \begin{cases} \alpha\ (boolean\ 1) & if\ SP_{avg}(structured) < SP_{avg}(random) \\ \beta\ (boolean\ 0) & otherwise \end{cases} \quad (18)$$

If a true classification is not achieved in the initial population, then the population is allowed to iterate taking the average of the two approaches.

CHAOS VARIABLES

Whenever propagation occurs, two phenomena that are created are either attractors or edges. These are regions where unpredictable behavior occurs, and regions which should be avoided. In order for the propagation to occur two chaos control variables are now defined.

Chaos Attractor C_A

The distance that each segment of solution has to differ from each other. The C_A is given as:

$$C_A \in [0, 1+] \quad (19)$$

Attractors form when solutions tend to converge towards a particular solution. Attraction can be classified in two ways, either through the use of objective functions or the ordering of solutions. The contemporary approach is to evaluate solutions in regards to their objective functions; however, two solutions may have the same objective functions, but different spread.

The SP is segmented into four regions, each region having a distinct spread, differing from its neighbor by at least one C_A.

$$\left(\delta_1,\delta_2,\ldots,\delta_{k/5}\right)\overset{C_A}{\leftrightarrow}\left(\delta_{(k/5)+1},\delta_{(k/5)+2},\ldots,\delta_{2(k/5)}\right)\overset{C_A}{\leftrightarrow}\ldots\overset{C_A}{\leftrightarrow}\left(\delta_{4(k/5)+1},\delta_{4(k/5)+2},\ldots,\delta_k\right) \quad (20)$$

This approach looks at the variance of the solutions in order to check for stagnation. This enforces separation of the solutions; enforcing non-convergence of the SP. C_A is the distance between two segments in the region spread.

During the propagation of the population, the minimum spread is to be maintained. As outlined C_A is an element of ς

$$C_A \in \varsigma \quad (21)$$

This enforces that no attractor forms, by isolating solution away from each other.

Chaos Edge C_E

Whereas the Chaos Attractor C_A deals with individual solutions variables, the Chaos Edge deals with the entire SP variables. The Chaos Edge C_E implies that the population be kept robust, in order to keep away from the chaotic edges. A tight grouping has to be initiated for the algorithm, since multiple intensified variables will lead to attraction, and highly diversified variables will lead to unguided behavior at the edges.

The solutions in the population cannot differ by more than the specified C_E at any point during the generations. The deviation of the entire SP has to be set at a threshold point where the SP will not become too diverse. This is a better approach than simply specifying the range between intensified and diversified solutions, since now the entire SP is taken into consideration when the computations is conducted.

$$C_E \in [0,1+] \quad (22)$$

As with all operational variables, C_A and C_E have to be tuned in order to have optimal operation of the algorithm. The initial population however gives a good indication of how the variables are to be tuned.

CROSSOVER APPROACH

The solutions are combined using a *two point crossover* with respect to the *variance factor* ς and the *spread factor* ∂. In order to propagate two conditions are vital. Firstly MAP structures have to be utilized. This requires the merging of intensified solution and diversified solutions. Secondly for rapid progression, the best solutions have to be exploited.

The solutions are mated according to their individual variables. A hig varience ς solution is mated with a low one and vice versa. As each crossover occurs, the resulting solution is evaluated with respect to its objective function, and a counter is set as to how successful different mating strategies are. Generally opposing ς and ∂ will give ideal results since a greater space will be evaluated in order to find solutions.

SOLUTION VALIDITY

Once a solution is generated, it is evaluated and checked against the other solutions in the SP. If it improves upon the best, it is then included in the SP. If it improves upon the first cluster of high performing solutions, it then replaces the worst solution within its cluster.

The most critical issue is that the population integrity has to be maintained. In order to avoid chaos attraction, C_A has to be maintained between the solutions. Once a solution is added to the SP, the C_A between the inserted cluster and the next cluster is checked.

The edge boundary C_E is also checked in order to avoid chaos edges. If the boundary is within tolerable limits then the solution is allowed to remain in the population.

DYNAMIC REPLACEMENT

Survival in the population is performance based. The first ten iteration of the population are free iterations, since no penalty functions are operational. After ten iterations, the performance is checked in terms of the *lifespan, offspring* and *propagate factor.* If the propagate factor χ is high, then the solution is non-performing, since the number of offspring's will be low. This solution is then tagged as dormant. If upon another set number of iterations, the solution does not improve, it is then tagged as expired.

As other solutions are created then solutions to be removed are the expired and dormant solutions, unless their rank is within the top five of the SP.

GENERATION

The solution iterates for a set number of generations, set in the range of:

$$100 \le G_{\max} \le 300 \quad (23)$$

However this value is user dependent. Upon the completion of the iterations, the top ranked solutions within each cluster is printed and the best within them is taken as the minima produced in the iteration.

CONCLUSION

The proposed heuristic of chaotic optimization addresses the issue of local optima convergence and stagnation. Premature convergence is shown as an attribute of Chaos theory, and its underlying principles.

The evolution of attractors and edges within the populations are indications of chaotic behavior.

By using standard spread techniques and applying selective crossover techniques, it is possible to keep the evolution progressing towards global minima. Anti-convergence criteria (C_A) along with anti-divergence criteria (C_E) are introduced to enforce strict distribution of the population.

The end result is a population which is multi-functional based, where for propagation to occur multiple conditions, like average spread, variance and rank have to be satisfied, instead of the generic approach of objective function evaluation. By using multiple functions, it then becomes possible to introduce intelligence within the population, and expedite population dynamics.

REFERENCE

Chen, Y. and Huang, X. 2005. "An Optimization Method Based on Chaotic Immune Evolutionary Algorithm". *International Conference on Natural Computation*. 890-894.

Dorigo, M. and Gambardella, L. 1997. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computations*. 1, 1, 53-65.

Dorigo, M. 1992, Ottimizzaazione, aprendimento automatico, et algoritmi basati su metafora naturale [Optimization, learning and natural algorithms]. PhD dissertation, Dipartimento Elettronica e Informazione, Politecnico di Milano, Italy.

Glover, F. 1989. Tabu Search –Part I. *ORSA Journal on Computing*. 1. 190-206.

Goldberg, D. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc.

Hénon, M. 1976. A Two-dimensional Mapping with a Strange Attractor. C*ommunications in Mathematical Physics*. 50. 69-77.

Kauffman, S. 1993. *The Origin of Order: Self-Organizational and Selection in Evolution*. Oxford University Press. New York.

Lorenz, E. 1993. *The Essence of Chaos.* University of Washington Press, Seattle.

Optimization and Automation Group, 2004. [http://www.upl.cs.wisc.edu/~creed/Theory-EV.html]

Onwubolu, G. C., and Davendra, D., 2006, Scheduling flow shops using differential evolution algorithm, *European Journal of Operational Research*, 171, 674-679.

Onwubolu, G., C. 2002. *Emerging Optimization Techniques in Production Planning and Control.* Imperial Collage Press, London.

Onwubolu, G., C. and Babu, B., V. 2004. *New Optimization Techniques in Engineering*. Springer Verlag, Germany.

Price, K. 1999. An introduction to differential evolution. In: Corne, D., Dorigo, M., and Glover editor's, *New Ideas in Optimization*, McGraw Hill International, UK, 79-108.

Richter, H., 2002. "An Evolutionary Algorithm for Controlling Chaos: The use of Multi-objective Fitness Functions". *Parallel Problem Solving from Nature.* 2439, 308-317.

Taillard, E., Gambardella, L., Gendreau, M. and Potvin, J. 2001. Adaptive memory programming: A unified view of metaheuristic. *Invited Review; European Journal of Operations Research.* 135, 1-16.

Van Laarhoven, R. and Aarts, E. 1987. *Simulated Annealing: Theory and applications.* Kluwer Academic Publishers.

Zelinka, I. 2005. "Investigation on evolutionary deterministic chaos control" , *Proc. IFAC*, Prague, Czech Republic, paper No. 03187

Zelinka I. 2006. "Investigation on realtime deterministic chaos control by means of evolutionary algorithms", *Proc. 1st IFAC Conference on Analysis and Control of Chaotic Systems*, Reims, France, 211-217.

Zhenyu, G., Bo, Cheng., Min, Ye. and Binggang Cao. 2006. "Self-Adaptive Chaos Differential Evolution", *International Conference on Natural Computation.* 972-975.

DONALD D. DAVENDRA is a Master of Science graduate in Computing Science from the University of the South Pacific. He is currently a doctoral candidate in Technical Cybernatics at the Tomas Bata University in Zlin, Czech Republic. His email address is davendra@fai.utb.cz

IVAN ZELINKA holds the position of Associate Professor**,** Head of Department of Applied Informatics and Vice Dean of Research at the Faculty of Applied Informatics at the Tomas Bata University in Zlin. He is the inventor of Analytic Programming and SOMA, which won him the Siemens Award for outstanding doctoral dissertation in Europe in 2001. His email address is zelinka@fai.utb.cz

GODFREY C. ONWUBOLU is the Professor and Chair of Engineering at the University of the South Pacific. He obtained his PhD from Aston University and has research interest in Optimization, Advance Manufacturing and Mechatronics. His email address is onwubolu_g@usp.ac.fj