

Maintaining Knowledge Distribution System's Sustainability Using Common Value Auction

Anas Al-Tirawi

Computer Science Department, Wayne State University,
Detroit, MI 48202, U.S.A
aaltirawi@wayne.edu

Robert G. Reynolds

Computer Science Department, Wayne State University,
Detroit, MI 48202, U.S.A
reynolds@cs.wayne.edu

Abstract

In Cultural Systems there are many ways to collect and distribute problem solving knowledge within social networks. Such mechanisms include games, auctions, and various voting mechanisms. Here, a new auction mechanism, Common Value Auctions, is presented. In this paper Common Value Auctions are used to distribute problem solving knowledge within a given model of social systems. These mechanisms are compared with the other distribution mechanisms in the solution of dynamic real-valued optimization problems. Specifically, their relative abilities to support the robustness and resilience of Cultural Systems in environments that vary in their dynamic complexity from static to chaotic are assessed. The Cultural Algorithms Toolkit (CAT) is used as a vehicle to generate real-valued dynamic problem landscapes of varying complexities. The results show that using the Common Value Auction in CAT4 has significant improvements over Weighted Voting methods (CAT2) in terms of both robustness and resilience across complexities that range from static to chaotic.

Keywords—cultural algorithm, sustainability, evolutionary algorithm, common value auction, robustness, resilience.

I. INTRODUCTION

Cultural systems provide a framework for human existence. One key observation that can be made is that certain cultures are more sustainable over time than others. Robustness and reliance are key factors behind the **sustainability** of cultural systems. These two factors are needed, so the system can handle a wide range of inputs/ perturbations while maintaining its integrity, structure, and reducing the severity of the impact that these perturbations can have on a system. **Robustness** is the property of a complex system to withstand the impact of a dynamic change or perturbation in its environment. Like a boxer in the ring, robustness is the quality of a system to endure a series of blows but still continue to function at a certain level or above. **Resilience** on the other hand is the ability of the system to adapt to the dynamics of its

environment so that it can maintain or improve its performance over time [1].

A cultural system will devote some of its resources to each of these two properties. If too many resources are devoted to robustness in the short term, it may impact its ability to be resilient in the long term and vice versa. So there needs to be a balance between the two in order for a system to be sustainable over the long term.

One key aspect of a Cultural System is how information can be distributed throughout its social networks in order to support both robustness and resilience. In this paper, the impact that various knowledge distribution mechanisms in a system will have on the systems robustness and resilience will be assessed. These mechanisms include voting schemes, auctions, games, and pure random processes. They will be studied through the lens of a computational model of cultural evolution, Cultural Algorithms.

In the next section the basic knowledge distribution mechanisms currently available for Cultural Algorithms are discussed. In section III the new knowledge distribution mechanism, Common Value Auction, is described. Section IV describes the dynamic landscape in which the performance of the new mechanism will be assessed. Section V provides a description of the experimental framework through which the sustainability of the Cultural Algorithm systems will be assessed. In the following section the performance of Common Value Auctions will be assessed in terms of the systems relative sustainability. Section VII presents the conclusions and suggestions for future work.

II. KNOWLEDGE DISTRIBUTION MECHANISMS IN CULTURAL ALGORITHMS

The Cultural Algorithm (CA) was introduced by Reynolds [2] as a computational model of Cultural Systems and their Evolution. It has been applied to many practical applications since then, one of which is: modeling the origins of agriculture in the valley of Oaxaca, Mexico [3]. In addition, CAs have been applied to concept learning [4], decision trees [5], software testing [6] and other hybrid approaches [7].

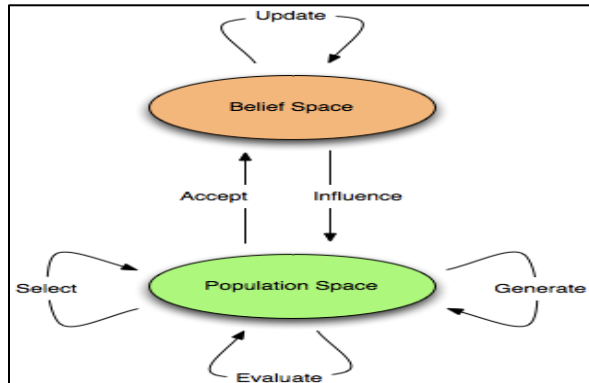


Fig. 1 Cultural Algorithm s framework [3]

As shown in Figure 1, the CA is a knowledge intensive evolutionary framework. First, the individuals in the population space are evaluated in terms of their performance in a problem space. Next, a subset of individuals is selected via the acceptance function and their performance is uploaded into the Belief Space which is a network of Knowledge Sources (KS). After updating the Belief Space network, the KS's can direct the next generation of individuals in the population space via the influence function. The knowledge sources (KS) utilize a variety of distribution mechanisms in order to circulate their influences among the individual agents in the population space.

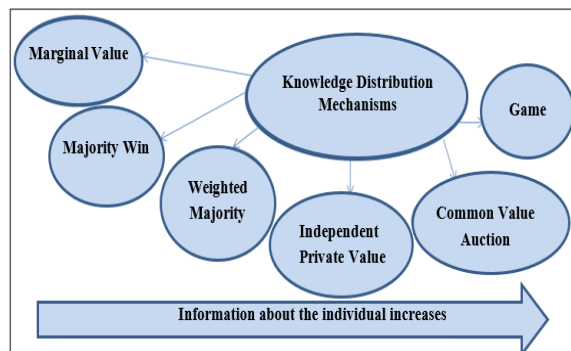


Fig. 2. The big picture for all Knowledge Distribution Mechanisms that utilize the CA.

Previous mechanisms have utilized CAs to distribute the influence of the KS over the individuals

in the population space. Fig. 2 shows the different knowledge distribution mechanisms. It is important to note that the amount of information that a distribution algorithm knows about a problem solution the fidelity), increases from left to right [8].

The first mechanism was called the Marginal Value Approach by Peng [9]. Every individual was controlled or directed by one KS in each generation. Peng in her approach [10], integrated the five KSs in the belief space into a single influence function as shown in Fig. 3. Peng used a random process based on their relative performance to select a KS to influence an individual. A KS roulette wheel, with proportional areas, based on relative performance, allows for an informed random selection process. However, Peng did not account for the influence among neighbors of a social network.

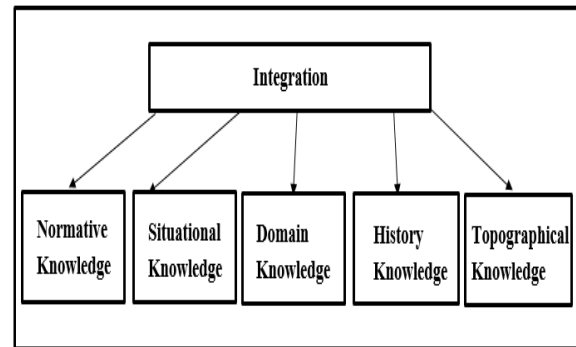


Fig. 3. Integration of multiple KSs [10]

Next, Ali and Reynolds [11], [12] developed a majority win approach. In their approach the influence of the neighbors is considered when selecting a KS. The social fabric is the connection between the individuals in a population. A conflict resolution process allows individuals to select the KS by which they are influenced, if their neighbors are influenced by one or more different ones. First, each individual was assigned a direct influence based upon the relative performance of the KSs using a roulette wheel as suggested above by Peng. Next, Ali used a conflict resolution strategy based on majority win in order to calculate the controlling knowledge source for each individual. They summed up the direct influence of the adjacent individuals in the social fabric and those of its current neighborhood. After that, the KS with the majority of the votes won the influence over this individual in that generation of the systems.

Another approach is the Weighted Majority Win proposed by Che [8]. Che used the average fitness of each KS to determine how much weight this KS deserves in the vote. The key to determining the weight for each KS is the average fitness value of the individuals that have recently been influenced by the knowledge source in the population. Figure 4 gives an example of the weighted voting process. The

individual, A0, has information about five competing KSs. They are represented in the figure as follows: S: Situational, D: Domain, H: History, T: Topographical, N: Normative. AO: represents the individual. The number of votes for each KS is given as x (number of votes). For Situational it is x3, or 3 individuals have it as a direct influence. The weight along each arc is the normalized relative performance for each KS.

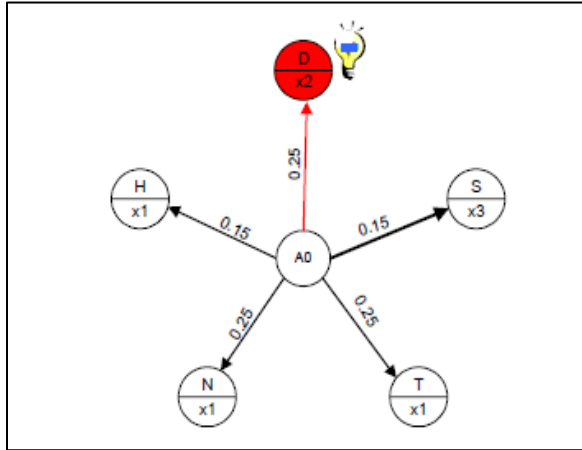


Fig. 4. Weighted Majority win in belief space through the social network [8]

In Fig. 4, the winning KS is the Domain KS, even though it does not have the most votes (votes=2). However, (D) does have the greater weight, which is the key factor in the weighted-majority win approach. Che has used many network topologies in his system including LBest, Square, Hexagon, Octagon, Hexadecagon, and Gbest. In addition to that he has also tested his system on different problem complexity levels [13]. As a result, Che concluded that when the performance function is of higher fidelity, the weighted approach can spread new information faster through a population than the majority win approach.

When the signal strength of information about the problem becomes even stronger, the auction approach can be effectively employed to find a solution. Kinnaird-Heather and Reynolds' [14], [15] embedded auction mechanisms into a CA. The new version was called CAT3. In CAT3, the production of the bidding tokens is the starting point. The process starts by producing bidding tokens and uses them to form biddings wheels for each KS. These tokens are generated by listing all of the individuals that were recently influenced by a KS over a given previous time window, t. The individual's fitness values are the bidding tokens and are normalized so that each previous result takes up a relative proportion of the token bidding wheel. The bid of a KS corresponds to the performance associated with the result of spinning the bidding wheel. Since the KS's do not have specific common knowledge about the location of the individual in the population network, the process must be stochastic based upon past performances.

As with the previous mechanisms the process for a generation starts with the assignment of a knowledge source to each individual as their direct influence. The next step is the selection of the bidders, those KS's who will be participating in the auction, from the KSs list. The KS's compete to influence each individual (x). The algorithm currently only allows the immediate adjacent neighbors of the individual (x) to participate in the auction. The actual auction takes place as shown in Fig. 5, where the system requests the selected bidders to submit their bidding values. Each selected KS will spin its correspondent wheel to get the bidding value. Finally, the auction system will determine the winner and assign the winner KS to influence individual (x). It may take several iterations to do so as shown in Fig. 5.

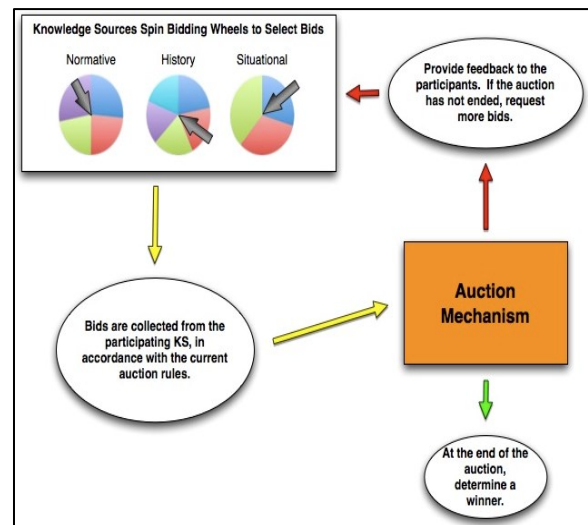


Fig 5. Conducting the Auction [14].

In the auction mechanisms above the bidders did not know anything about the properties of the individuals upon which they were making bids. Those properties can be the location in the network, the number of immediate neighbors, and the strength of their connections, what knowledge sources have influenced it in the past, among others. In the next section an approach, the Common Value Auction, is discussed. This approach provides a common set of parameters that are available to all bidders. These parameters can be used to condition the bids made by the participants.

III. THE COMMON VALUE AUCTION DISTRIBUTION MECHANISM

The new mechanism, Common Value Auction Toolkit (CAT 4) is an extension of CAT3. CAT4 propagates the influence using the Common Value knowledge. The Common Value knowledge is a set of parameters that every KS can know about the individuals in the social network. These include the individual's location in the network and the KS(s) that

influenced the individual in the past previous approaches.

The first step is to build the KS wheel (one wheel for all KSs), by normalizing the KS average score, where every KS will have a wheel's share that reflects its average (score). Each KS will have a portion of the wheel that reflects the average performance of those individuals who have been influenced by the correspondent KS. Next the algorithm assigns a direct influencer KS randomly using the roulette wheel approach discussed previously to each individual in the population. This step is the same as that for all other mechanisms discussed so far.

In the second step each KS constructs a bidding strategy wheel that will be used later to determine their bidding decision on a specific individual. This is done by selecting a subset of recent individual performances directed by that KS over a given past time window. A wheel is constructed such that each score comprises an area that is proportional to its contribution to the total score of the subset for the KS. In addition, a set of rules is selected to determine whether the KS will bid on an individual based upon common value knowledge about the location of the individual in the social fabric.

Next, the direct influence for each individual in the population is compared against those of its neighbors, here, just the directly adjacent neighbors are used. If the direct influence of an individual agrees with those of all of its neighbors, its direct influence is then chosen to guide it during that generation. Otherwise an auction is conducted between those KS's who directly influence that individual and its neighbors.

In order to do this the bidding strategy for each of the competing KSs is checked to see if it will bid on that individual based upon the common value information. The rule set associated with the KS is checked to see if taken together they support a bid on the current individual. This "expert system" can technically be comprised of many rules. For the experiments here, the same one rule is used for all KSs. To do so, the following distributing mechanism based upon just one subset of common values, the extent to which the individual and its neighbors have been influence by the KS in the past:

If KS[j] has influenced individual (i) in the past m generation or

If KS[j] is influencing currently the neighbors of individual

Then increase the bidding value by a bonus as shown in the equation below:

$$\text{Bidding value} = \text{KS's bidding value} + 0.5 \text{ (Boost)} \quad (1)$$

This is where the Common Value information is used to determine the winner. We simply give

incentive for the KS that influenced the individual in the past and for those KSs that were able to influence the individual's neighbors.

In the fourth step, the influencers that satisfy their bidding rules are then chosen to participate in the auction. The bidding wheel is spun for each to determine their bid. In the experiments conducted here each KS had a bidding wheel comprised of a single average value for the performance of the selected subset in order simplify computations at this stage.

In the fifth step, the bidding strategy rules that are satisfied for a KS are then applied to the bid as shown in the rule above to give a final bid for that KS. The bids are then compared with each other and the winner is selected to control the individual for that generation. If there are no bidders, then the direct influencer of the individual is retained. This redistribution process is then repeated for all individuals in the network.

Fig 6 covers the big picture of CAT4. First, the KS roulette wheel is spun to generate the direct influencer for each individual. If one or more of the individual's neighbors possess a different KS then each decides whether it wishes to bid for that individual using the common value information about individual. The selection process is governed by a rule-based expert system associated with each KS. The selected KSs then participate in the bidding for the auction as described above.

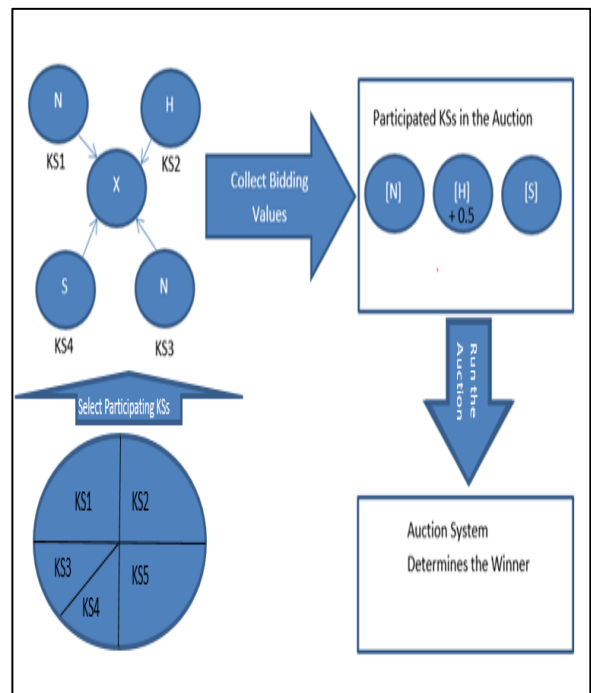


Fig. 6. Big picture of CAT4 Algorithms

IV. THE DYNAMIC PERFORMANCE ENVIRONMENT: THE CONES WORLD

To analyze the results and test the performance on the different levels of complexity, a robust problem generator (Cones World) was used in both CAT2 and CAT4. The Cones world framework was inspired by the work of Morrison and De Jong [16]. This tool has the ability to generate dynamic problem environment over various landscape complexities. A given cone world configuration can be described as follows:

$$f(\langle x_1, x_2, \dots, x_n \rangle) = \max_{j=1, k} \left(H_j - R_j \cdot \sqrt{\sum_{i=1}^n (x_i - C_{j,i})^2} \right) \quad (1)$$

Where: K: the number of the Cones. H_j : the cone height, R_j : the cone slope, N: the dimensionality. $C_{j,i}$: Coordinates of the cone j in dimension i ., (X_i, Y_i) : determine the location of the cones on the landscape. The values for the cone height, slope, and coordinates can be assigned randomly through the problem generator or logistic function. However, the values would be selected from the ranges below:

$H_j \in (H_{base}, H_{base} + H_{range})$; $R_j \in (R_{base}, R_{base} + R_{range})$; and $C_{j,i} \in (-1, 1)$.

The Max function here is used to handle the combination of the cones when they overlap. For example, if two cones overlap, the Max function will choose the height of the combined cone to be the height of the highest cone for the two overlapped cones. Fig 7 shows how the landscape looks like with the following parameters: $k = 15$, $H_{base} = 1$, $H_{range} = 9$, $R_{base} = 8$, and $R_{range} = 12$. To determine the dynamic changes of the system Morrison and De Jong used the logistics function below:

$$Y_i = A * Y_{i-1} * (1 - Y_{i-1}) \quad (2)$$

$A = \text{Constant value}$, Y_i is value of Y at iteration i .

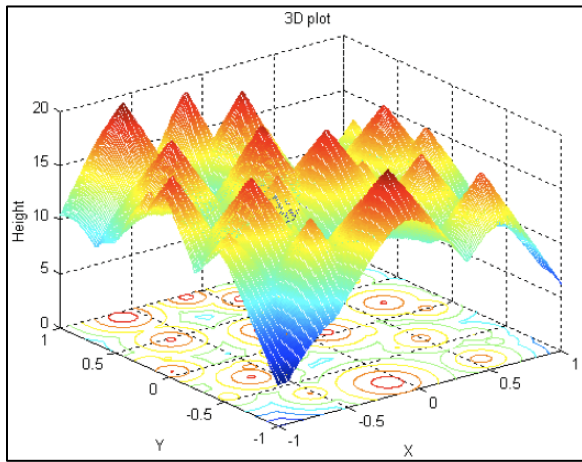


Fig.7. an Example Landscape In two-dimensional space ($n = 2$) bound by $x \in (-1.0, 1.0)$, $y \in (-1.0, 1.0)$ with $k = 15$, $H \in (1, 20)$, and $R \in (8, 20)$ [9].

As the value of A increases, the system generates more complicated behavior. Figure 8 shows how Y will change as a result of A for a sequence of landscapes. The x -axis gives the number of generations, the z axis gives the A value, and the Y axis gives the Y -value produced over the given generations for a specific A . Each of the Y trajectories is color coded with the A value that produces it. The color code is in the legend on the right side of the graph. Low values of A produce gradual linear changes while high values produce wildly oscillating values for Y . In the next section we discuss the experimental framework of CAT4. Also, we explain how the dynamic environment can affect the learning curve of the whole system and consequently the produced results.

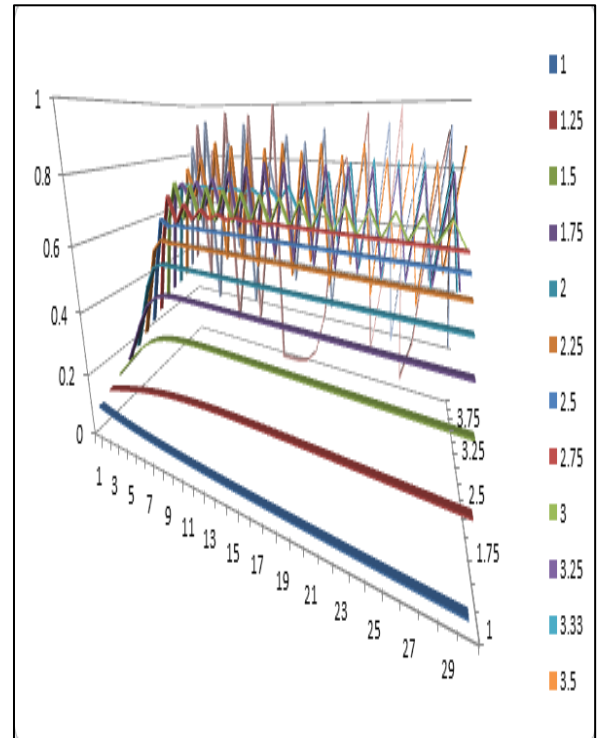


Fig. 8. The value for Y (on the Y -axis as a function of A (z axis) over the number of generations, x axis. The Y curves are color coded with A -value that generated them. The color code is on the right.

V. DYNAMIC EXPERIMENTAL FRAMEWORK

In these experiments, the performance of the Common Value Auction was compared with the Weighted Majority algorithm for three complexity levels of $A = \{1.01, 3.35, \text{ and } 3.99\}$. These three A -Values were selected because they represented a wide spectrum of complexities over which to test CAT4 against. The full list of experimental framework parameters are summarized in the table below. The types of social fabrics are explained here [17].

TABLE I. EXPERIMENTAL FRAMEWORK PARAMETERS

Parameter Name	Value
Complexity Class	1.01, 3.35, and 3.99
Number of Runs Per Complexity	300
Number of landscapes	50
Max. number of generation per landscape	800
Number of cones	100
Number of agents	50
Social fabrics	{L-Best, Square, Hexagon, Octagon, Sixteengon, Global}
Max fitness value	20
Precision of solution	0.001

The key hypothesis to be tested here is whether the Common Value Auction mechanism is able to produce a more sustainable cultural system than the weighted majority voting mechanism. The extent to which this is accomplished will be observed in terms of the two system's relative robustness and resilience over the course of 40,000 generations for each of the 300 runs for the 3 complexity classes.

Robustness will be assessed in terms of the ability of the system to bounce back after each of the 50 landscape shifts for a given run. The standard deviation over the set of 300 runs will provide an indicator of the need for each system to bounce back from a landscape change. **Resilience** on the other hand will be observed in terms of the extent to which the systems are able to adapt to these landscape shifts by reducing the time needed to achieve the optimum in the next landscape. The systems will then be compared in terms of how the complexity of the environment impacts their relative **sustainability** as the environmental complexity shifts from static, then to cyclic, and finally to chaotic.

VI. A COMPARISON OF THE RELATIVE SUSTAINABILITY OF THE COMMON VALUE AUCTION AND THE MAJORITY WIN KNOWLEDGE DISTRIBUTION MECHANISMS

The main difference between the two algorithms is that CAT4 uses information about the individuals before the auction starts. The CAT4 algorithm is an informative algorithm that provides crucial information for the bidders about past behavior of the individuals in the population space. This information is used to trigger bidding strategies for each of the knowledge sources. While many different factors can

be used to affect bidding strategies, the focus here will be in just a single set of factors, the KS previously used to influence an individual and its neighbors. The goal will be to show that the addition of just this new information can make a substantial difference in the performance of the cultural system.

The first dynamic landscape to be assessed was that produced by $A=1.0$. As seen in the previous section, that landscape involves a series of small linear shifts in the locations of the cones. Fig. 9 gives the standard deviation of the two systems over all three environments. For the linear dynamic landscape, the two systems each was perturbed by an average of around 85 generations for each landscape change. So their relative level of **robustness** is about the same for this environment.

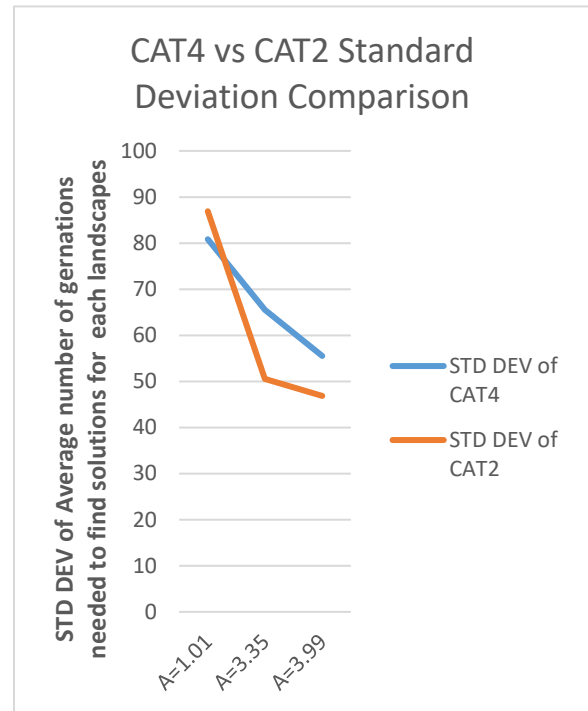


Fig. 9 CAT4 vs CAT2 standard deviation comparison.

The relative **resilience** of each of the two systems in the linear landscape is illustrated in Fig. 10 and 11. Both systems are able to significantly reduce the number of generations needed to find the new optimum over time. The CAT4 system was able to produce a correlation of (0.662) between the number of generations needed to solve the changed landscape and landscape number. The corresponding coefficient of determination, the percentage of the total variance, explained by the correlation is (0.43). CAT2 exhibited a coefficient of determination of (0.289). As shown in Table II the correlations were significantly different

from each other at the (0.05) level of significance. So CAT4 was able to do a better job of adapting to the changing linear environment than CAT2.

While CAT4 exhibited a significant level of learning within an environment with linear dynamics, the next question is how it would adapt to an environment in which the changes were non-linear from landscape to landscape. A nonlinear shift in cone location was produced by the landscape generated for $A=3.35$ as shown in Fig. 8 above. The relative change in robustness produced by the shift to a non-linear dynamic for the two systems is given in Fig. 12 and Fig.13. CAT4 exhibited an approximately 15 generation increase in terms of its response to a perturbation compared to CAT2. That is a significant difference in its ability to rebound from a perturbation in this environment. On the one hand, a non-linear environment required CAT4 to respond more robustly than before.

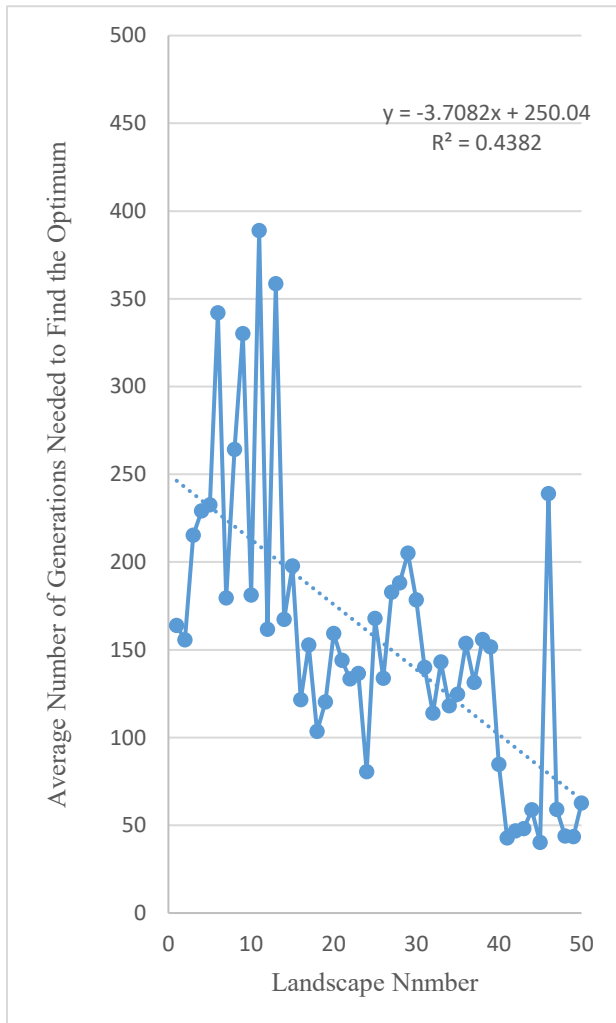


Fig. 10. CAT4 Regression line over 50 runs for complexity $A = 1.0$

On the other hand, CAT4 improved on its ability to adjust to the change in landscapes as reflected in an improved correlation coefficient (0.73) and coefficient of determination (54%) as shown in Table II. The Weighted Majority system exhibited a much lower overall coefficient of determination, (0.17). Again, the two systems exhibited a significant difference in adaptability over time, but now in a nonlinear environment.

Overall when the environment switched from a linear to a non-linear one the CAT4 mechanism produced a distinctly more robust and resilient behavior than CAT2. The next question is how the two systems would adapt to an extremely “chaotic” environment that was characterized by the superposition of numerous non-linear patterns of behavior?

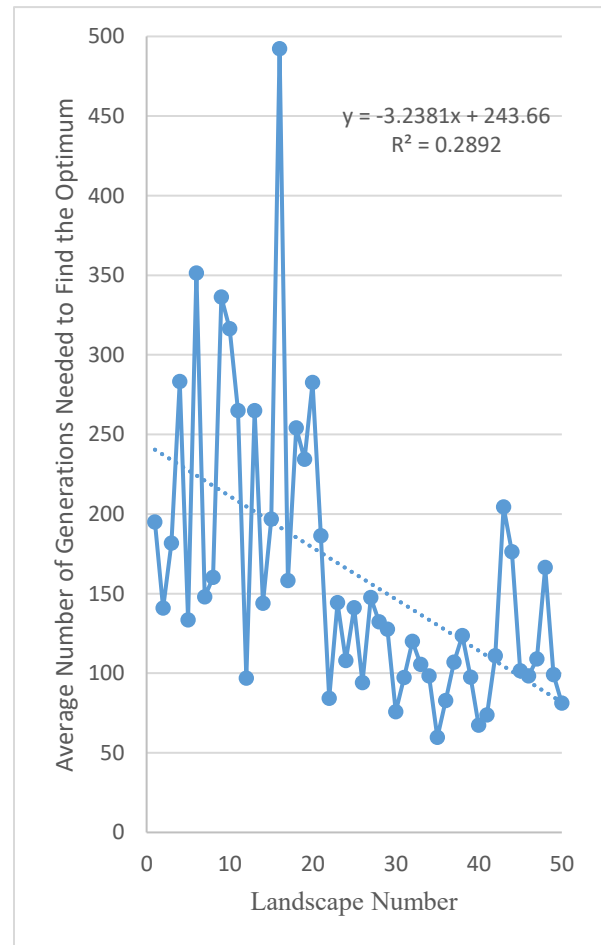


Fig. 11. CAT2 Regression line over 50 runs for complexity $A = 1.0$

Since the generating process was deterministic in nature, all of the information needed to provide a perfect prediction of the environment’s dynamics is there, it is just a matter of extracting all of the intertwined threads.

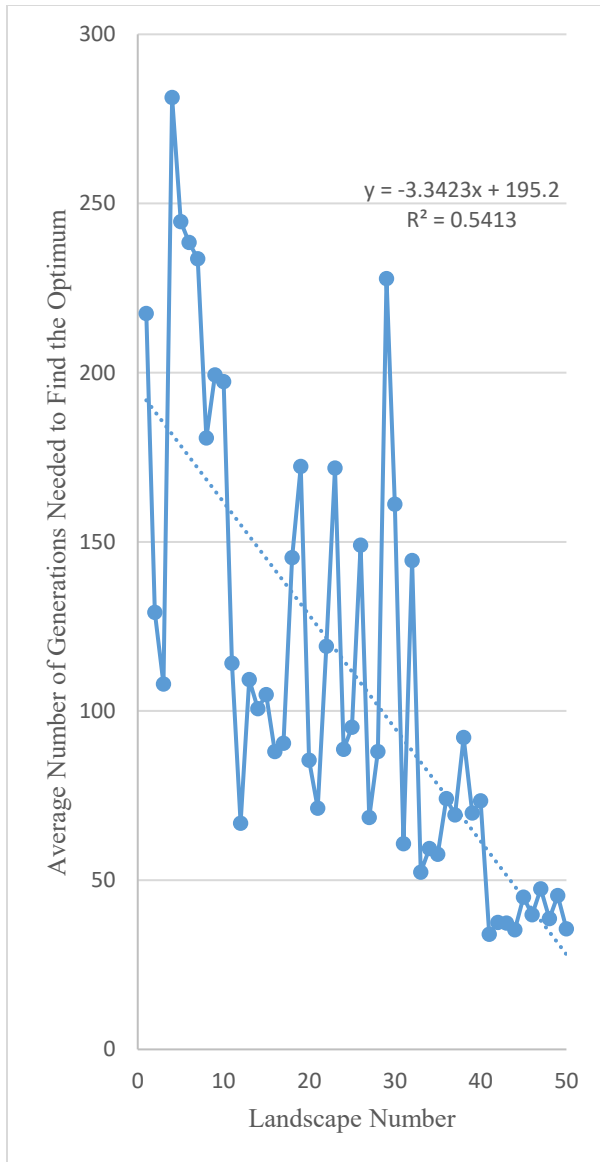


Fig. 12. CAT4 Regression line over 50 runs for complexity, $A=3.35$

As demonstrated in Fig. 14 and 15, the robustness of the CAT4 system is still significantly greater than that for CAT2. The difference in the number of additional generations needed to respond to a perturbation is now 10. That is down from 15 before, but still a significant difference in system robustness.

In such a chaotic environment learning is less of an issue than sustainability. As shown in Table II the two systems now exhibit a much lower level of resilience. The coefficient of determination for CAT2 is now significantly greater than that for CAT4 but notice that the relation between the number of generations needed to solve the problem is now increasing with increased landscape number. The rate of increase is now higher for CAT2 than CAT4 which means that its performance is more susceptible to degradation in this

environment. While both system's behavior is now clearly nonlinear, the regression line provides a general indicator of the additional stress that is placed on each system over time.

In the first two environments, the systems were not only able to survive the perturbations but to adapt to them. This produced a strong sense of sustainability. Of the two, CAT4 was more able to exploit the nonlinear environment. In the chaotic environment the theme was less on adaptability but survivability over time. Both systems displayed symptoms of stress over time.

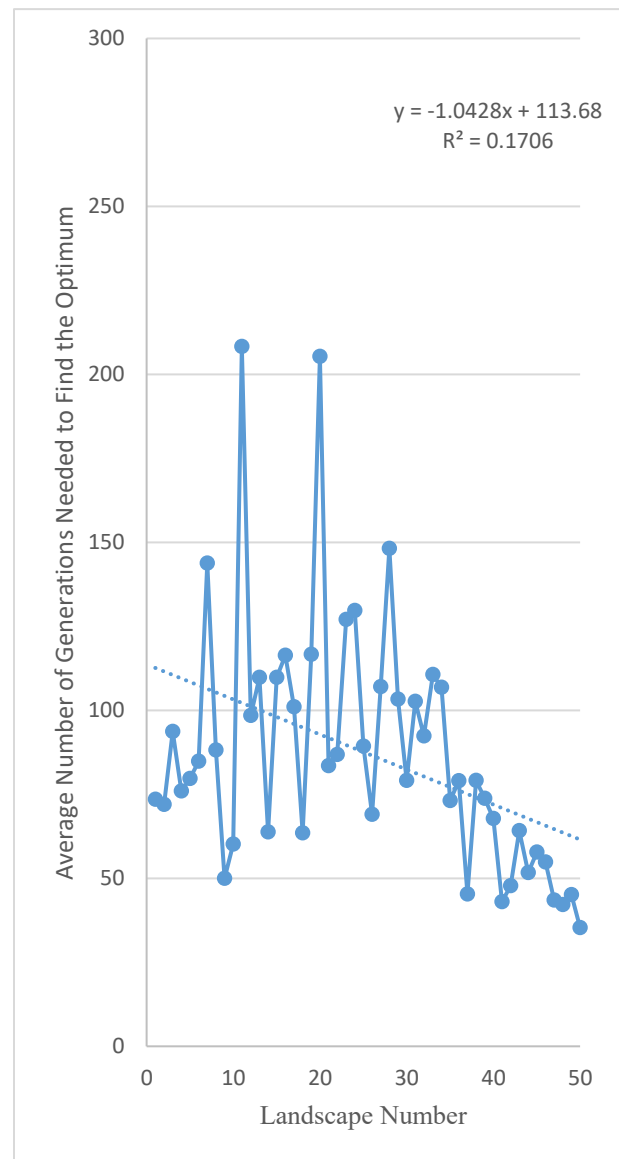


Fig. 13. CAT2 Regression line over 50 runs for complexity, $A = 3.35$

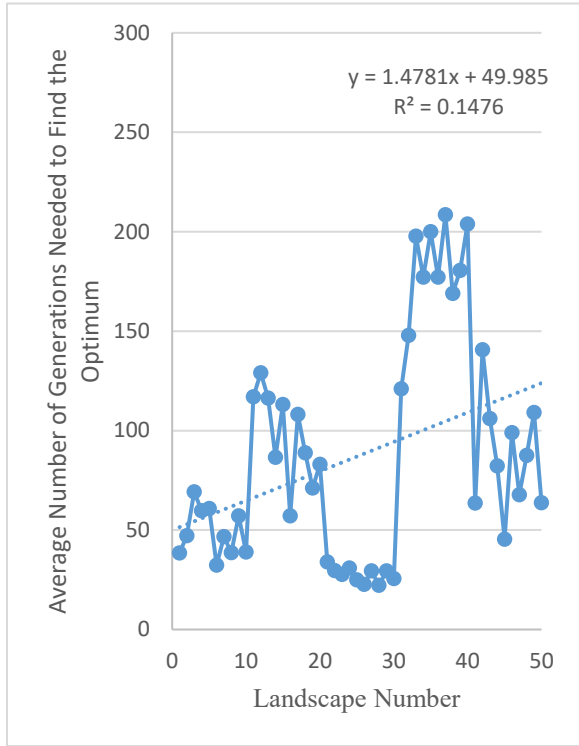


Fig. 14. CAT4 Regression line over 50 runs for complexity, $A = 3.99$

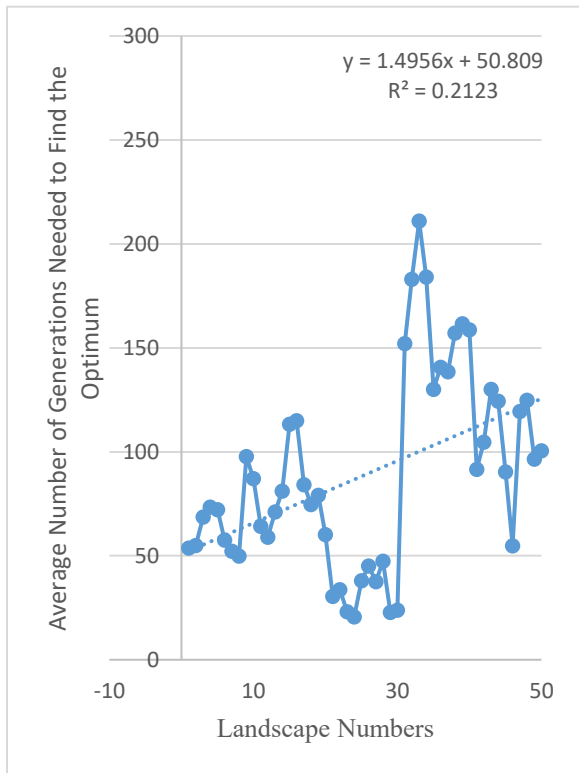


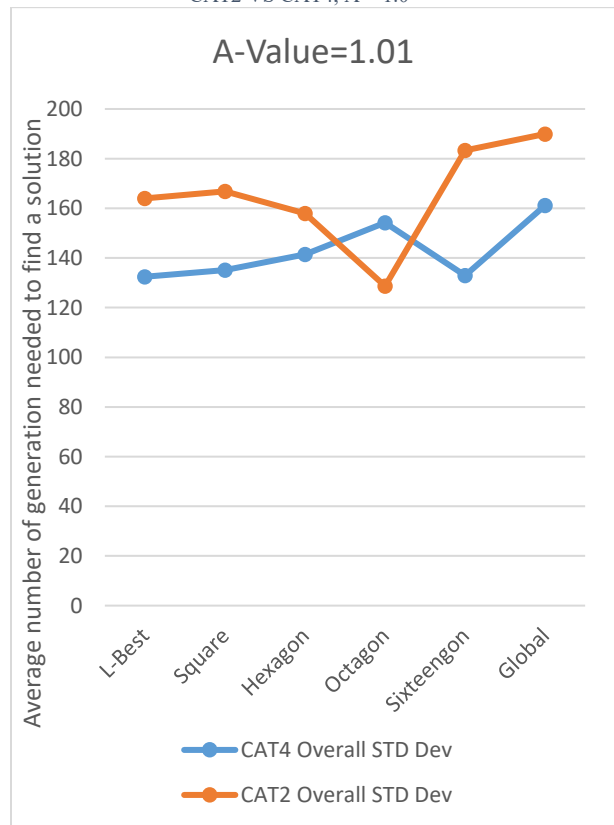
Fig. 15. CAT2 Regression line over 50 runs for complexity, $A = 3.99$

TABLE II. COMPARING CAT4 AND CAT2 REGRESSION THROUGH DIFFERENT A-VALUES COMPLEXITIES

	A-Value	R	CAT4(R ²)	CAT2(R ²)	Sig F Change
Static	1.01	0.662	0.438	0.289	0.000
Periodic	3.35	0.736	0.541	0.170	0.000
Chaotic	3.99	0.384	0.147	0.212	0.006

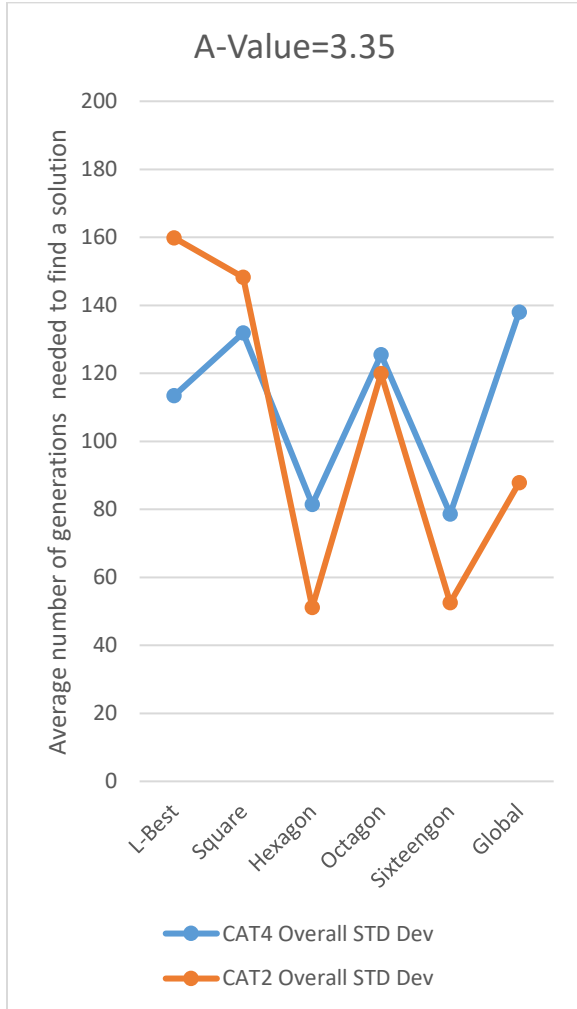
Another way of comparing the two algorithms is to compare the standard deviation for average number of generation needed to find the solution for given problem. The three tables below are showing the comparison for the three different complexity values { $A=1.01, 3.35, 3.99$ }. As showing in Table III, CAT4 needed less number of generations to find the solution for the same number of problems. Except for Octagon topology, CAT4 was more efficient than CAT2.

TABLE III: COMPARING THE STANDARD DEVIATION FOR CAT2 VS CAT4, $A = 1.0$



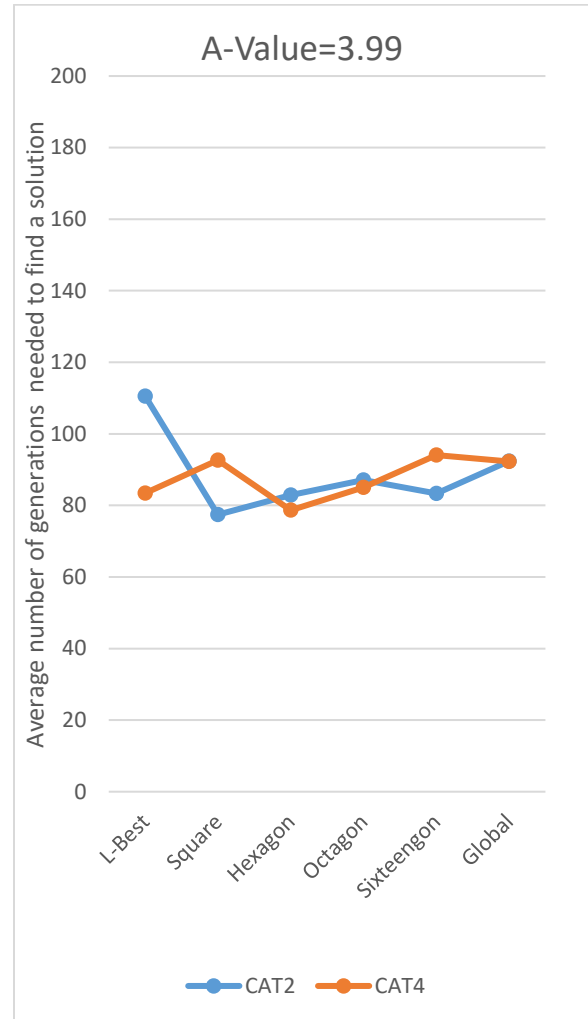
When the complexity increases to $A=3.35$, CAT4 was outperformed by CAT2. With the exception of the first two topology (L-Best, and Square), CAT2 was more efficient. CAT2 needed less number of generations to solve the same given problems when compare with CAT4.

TABLE IV: COMPARING THE STANDARD DEVIATION FOR CAT2 VS CAT4, $A = 3.35$



Interestingly for Complexity level $A=3.99$, When the system is in complete chaotic situation, CAT4 outperform CAT2.

TABLE V: COMPARING THE STANDARD DEVIATION FOR CAT2 VS CAT4, $A = 3.99$



VII. CONCLUSIONS AND FUTURE WORK

In society, there are many ways to collect and distribute problem solving knowledge. Such mechanisms include games, auctions, and various voting mechanisms. Previous work has focused on Independent value auctions. KSs did not have knowledge about the individuals on who they were bidding and did not have consistent bidding strategies. In this paper, Common Value Auctions were presented. This framework provided common knowledge to all KSs about each individual and supported rule based systems that were used to house individual KS bidding strategies T

The experimental results suggest that adding the common value auction to the CAs can enhance the robustness and the resilience of the algorithm relative to the commonly used Weighted Majority vote distribution mechanism. The differences in resilience

were significant across a wide range of dynamic environments tested, from linear to chaotic. The results effectively demonstrate the impact that knowledge about social networks can have on the sustainability of a Cultural system.

However, it was clear that as the environment of the Cultural Algorithm became increasingly chaotic, there was a shift from the need to sustain the culture through adaptations to that of survival. The presence of additional knowledge in CAT helped in that regard. The question remains as to what type of information about social networks will be particularly useful in guiding complex social systems into even more complex global environments. That is the focus of future work.

REFERENCES

- [1] R. Morrison and K. D. Jong, "A Test Problem Generator for Nonstationary Environments," in *Evolutionary Computation, CEC*, pp. 25-31, Washington, DC, 1999.
- [2] J. Husdal, *Robustness and flexibility as option to reduce uncertainty and risk*, Molde, Norway: Molde University College, 2004.
- [3] R. G. Reynolds, *An Adaptive Computer Model of the Evolution of Agriculture*, Ann arbor , MI: University of Michigan, 1979.
- [4] T. W. Jayyousi, *Bringing to Life an Ancient Urban Center at Monte Albán, Mexico: Exploiting the Synergy between the Micro, Meso, and Macro Levels in a Complex System*. Thesis, Detroit, MI: Wayne State Univerity, 2012.
- [5] W. Sverdlík, R. G. Reynolds and E. Zannoni., "HYBAL: A Self-Tuning Algorithm for Concept Learning in Highly Autonomous Systems.," 1992.
- [6] H. A. Al-Shehri, *Evolution-based decision tree optimization using cultural algorithms*. Thesis, Detroit, MI: Wayne State University, 1997.
- [7] R. G. Reynolds and D. Ostrowski, "Knowledge-Based Software Testing Agent using Evolutionary Learning with Cultural Algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 1999.
- [8] M. Z. Ali, N. H. Awad, P. N. Suganthan, R. M. Duwairi and R. G. Reynolds, "A novel hybrid Cultural Algorithms framework with trajectory-based search for global numerical optimization," *Information Sciences*, vol. 334, no. C, pp. 219-249, 2016.
- [9] X. Che, M. Z. Ali and R. G. Reynolds, "Robust evolution optimization at the edge of chaos: Commercialization of culture algorithms," in *IEEE Congress on Evolutionary Computation*, Barcelona, Spain, 2010.
- [10] R. G. Reynolds, B. Peng and X. Che, "Knowledge Swarms Generating Emergent social structure in Dynamic Enviroments," in *Proceedings of the agent Conference on generative social processes, Models , and Mechanims.*, 2005.
- [11] B. Peng, "Knowledge Swarms in Cultural Algorithms for Dynamic Environments, Ph.D. Thesis," Wayne State University, Detroit,MI, 2005.
- [12] R. G. Reynolds and M. Ali, "Embedding a social fabric component into cultural algorithms toolkit for an enhanced knowledge-driven engineering optimization," *International Journal of Intelligent Computing and Cybernetics (IJICC)*, vol. 1, no. 4, pp. 356-378, 2008.
- [13] R. G. Reynolds and M. Ali, "The social fabric approach as an approach to knowledge integration in Cultural Algorithms," in *IEEE Congress on Evolutionary Computation*, 2009.
- [14] X. Che, M. Z. Ali and R. G. Reynolds, "Weaving the Social Fabric: The Past, Present, and Future of Optimization Problem Solving with Cultural Algorithms," in *AAAI Fall Symposium: Complex Adaptive Systems*, Arlington, Virginia, 2010.
- [15] R. G. Reynolds and L. Kinnaird-Heether, "Optimization problem solving with auctions in Cultural Algorithms," *Memetic Computing*, vol. 5, pp. 83-94, 2013.
- [16] R. G. Reynolds and L. Kinnaird-Heether, "Problem solving using social networks in Cultural Algorithms with auctions," in *IEEE Congress on Evolutionary Computation*, San Sebastian, Spain, 2017.
- [17] Y. A. Gawasmeh and R. Reynolds, "A Computational Basis for the Presence of Sub-Cultures in Cultural Algorithms," in *IEEE Symposium on Swarm Intelligence*, Orlando, FL , 2014.
- [18] J. Schell, *The Art of Game Design: A Book of Lenses*, Boston: Elsevier/Morgan Kaufmann, 2008.
- [19] A Test Problem Generator for Nonstationary Environments," in *Evolutionary Computation, CEC*, pp. 25-31, Washington, DC, 1999.