The Impact of Subcultures in Cultural Algorithm Problem Solving

Robert G. Reynolds
Computer Science Department
Wayne State University
Detroit, MI USA
Reynolds@wayne.edu

Yousof A. Gawasmeh
Computer Science Department
Wayne State University
Detroit, MI USA
ygawasmeh@wayne.edu

Areej Salaymeh
Computer Science Department
Wayne State University
Detroit, MI USA
Areej@wayne.edu

***Abstract*— Cultural Algorithms are computational models of social evolution based upon principles of Cultural Evolution. A Cultural Algorithm composed of a Belief Space consisting of a network of active and passive knowledge sources and a Population Space of agents. The agents are connected via a social fabric over which information used in agent problem solving is passed. The knowledge sources in the Belief Space compete with each other in order to influence the decision making of agents in the Population Space. Likewise, the problem solving experiences of agents in the Population Space are sent back to the Belief Space and used to update the knowledge sources there. It is a dual inheritance system in which both the Population and Belief spaces evolve in parallel.**

In this paper we compare three different social fabrics (homogeneous, heterogeneous and Sub-Cultures) over a wide range of problem complexities. The performances of these three different evolutionary approaches are compared relative to a variety of benchmark landscapes of varying entropy, from static to chaotic. We show that as the number of independent processes (layers) that are involved in the production of a landscape increases, the more advantageous subcultures are in directing the population to a solution. Such landscapes are often characteristic of deep learning problems in which patterns are generated by the interaction of many simple interactions. While sub-cultured approaches can emerge in a given problem, they do not have to. It is shown that for single layer generators for a landscape or image, sub-cultures do not effectively emerge since they are not needed to solve such problems.

I. INTRODUCTION

Cultural Algorithms are computational models of complex cultural systems. A complex system can be defined as a system made up of a group of heterogeneous agents who interact and adapt with their environment and each other [18]. The separate behaviors of the heterogeneous agents when aggregated together can cause higher-level behaviors to emerge from the group as a whole which work to solve problems facing the group. In summary, complex systems are composed of different layers of interaction and in each layer there may be different problems to be solved. A solution to one problem at a level can be integrated with the solutions to problems solved at other levels.

The social fabric term was introduced into the Cultural Algorithm by Reynolds and Ali [1] in order to build connections among the individuals in the Population Space of a Cultural Algorithm. These connections formed networks that allow the knowledge to be transferred from an individual (source) to a connected individual (destination). The social fabric for a Population Space can be constructed from homogeneous networks, heterogeneous networks, or both. They termed it the social fabric since the connections could be torn or broken if not kept up.

In order to test the impact of social fabrics on Cultural Algorithm problem solving an optimization problem landscape generator developed by Morrison and De Jong was employed [7]. The landscape generator produced landscapes of varying degrees of entropy by combining surface produced by individual generator processes. A static landscape was produced by a single generator, a periodic landscape by the convolution of two separate generators, while a chaotic landscape was produced by the convolution of a large number of individual generators. Deep learning problems are often associated with problems of this type.

Previously, Ali [1] and Che [13] used static homogeneous social fabrics in Cultural Algorithms. Both Ali and Che [13] showed that some of the homogeneous topologies operated better with specific complexity class problems than others. Also, [15], the performance of Cultural Algorithms had been examined with using both homogeneous and heterogeneous social fabrics over the complete range of Langton's' optimization problem complexities [5]. It was concluded that heterogeneous topologies worked well at solving chaotic class problems, but their relative performance decreased with problems of lower complexity or fixed complexity. In contrast, homogeneous topologies work well at solving low or fixed class problems, but their performance decreased as the complexity of the problem increases.

Based upon these results it was suggested that a Cultural System will benefit by the use of multiple topologies when faced with a diverse set of problem complexities. As a result Kobti [8]

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.261

suggested an approach with multiple population spaces, each with its own homogeneous topology and an associated Belief Space.

One of the manifestations of cultural systems, particularly in recent times, has been the emergence of sub-cultures. Each subculture is characterized by a topology and a set of concepts and information that are passed through the topology. In this work, a single Population Space with multiple topologies is employed. Each topology is viewed as a subculture. The Belief Space keeps information about the performance of each subculture as a combination of a topology and a knowledge source. The topologies compete with each other to find patterns in a problem landscape. The more generators that combined to produce a landscape, the more underlying patterns there are for each topology to undercover. Such problems that require many learning layers such as pattern recognition problems in economic systems are suitable problems for this approachThis sub-cultured social fabric is compared with the performance of systems that use the homogeneous and heterogeneous social fabrics.

In the next section, the Cultural Algorithm framework is described. In Section 3, the homogeneous topologies are described with a little more detail about how they operate among the Population Space. In Section 4, the heterogeneous topologies and their influence on the performance of Cultural Algorithms are presented. The Sub-Cultured model will be given in detail in Section 5. The experimental framework will be given in detail in section 6. The results of the experiments and analysis will be given in section 7 and the conclusions presented in section 8.

II. THE CULTURAL ALGORITHM FRAMEWORK

The Cultural Algorithm (CA) is an evolutionary computational model derived from conceptual models of the Cultural Evolutionary process [9] [10]. The Population Space, Belief Space, and Communication Protocol between the Population and the Belief Space are the three major components of this model, as shown in Fig. 1. The Cultural Algorithms have been used to solve problems in natural languages [19], multi agent systems [3] [4] [11] [14] [18], and game programming [6] [12] [16].

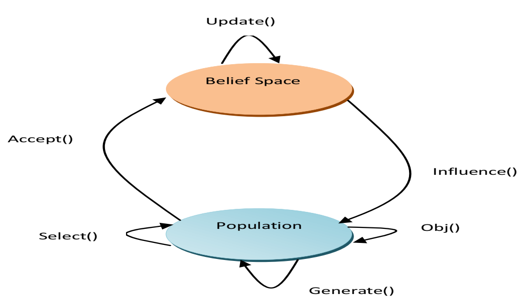


Fig. 1. Cultural Algorithms Framework.

The Belief Space is a network of knowledge sources that represent different categories of knowledge. The five classes of knowledge currently implemented are: Situational, Normative, Domain, Temporal, and Topographic.The Situational knowledge keeps track of the individuals who exhibit the best behavior in the population. The Normative knowledge provides standards for individual behaviors to ensure that the individual is still within the scope of the system. The Domain knowledge contains the knowledge about the problem domain that can be used to predict relationships between problem variables. Temporal knowledge is useful in dynamic environment to store all changes in the problem landscape, such as the direction of the population behaviors either towards convergence or divergence. Topographical Knowledge gives a distribution of landscape values over feasible and infeasible regions of the search space.

The basic program of the Cultural Algorithm is shown in Fig. 2, where P (t) and B (t) represents the Population Space and the Belief Space at time t, respectively. Obj() and accept() functions needs to be executed at the end of each iteration. Obj() is the performance function, which is used to evaluate the performance of each individual in the Population Space. Then, a group of individuals will be chosen to update the Knowledge Sources in the Belief Space using the Acceptance Function, accept(). The Acceptance Function selects the best elements to update, but in some cases the Acceptance Function selects from different performance categories in order to explore all results of the population. Updating the Belief Space occurs via the update () function. In the Belief Space, there are many kinds of Knowledge Sources. Some sources are updated by the Update Function directly and some of them indirectly by the interaction with other updated sources. Next, the influence () function transmits the updated knowledge from the Knowledge Sources in the Belief Space to the individuals that reside in the Population Space to direct their performance. Transferring the experiences between the Belief Space and the Population Space is called the Communication Protocol.

```
Begin
        t=0;
        Initiate B(t) and P(t)
        Repeat
                Evaluate P(t) using Obj()
                Update B(t) using Accept()
                Generate P(t) using influence()
                t=t+1
                Select P(t) from P(t-1)
        Until (termination condition achieved)
End
```

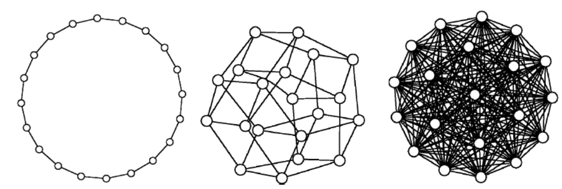
Fig. 2. Outlined Program of Cultural Algorithm.

The Cultural Algorithms repeatedly produce a new generation, and update the Belief Space until the termination condition is satisfied. The Cultural Algorithm is flexible since it can look for small scale and large scale solutions and run in static and dynamic environments [1]. The termination conditions can reflect the number of generations used, or the convergence to a desired solution.

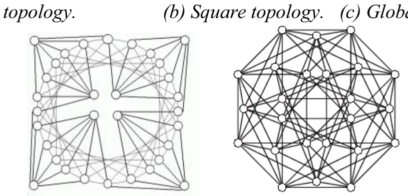
III. Homogeneous Topologies

In the earliest Cultural Algorithms individuals acted independently and were influenced by a single Knowledge Source [17]. Ali [1] introduced three homogeneous topologies and Che [13] introduced three more into the Cultural Algorithm to allow the Population Space's individuals to communicate– making six possible homogeneous topologies: lbest (ring), gbest (global), square, octagon, hexagon, and sixteen sided. Fig. 3 displays some of these homogeneous topologies.

Homogeneous topologies worked well in solving problem landscapes of low entropy, but performance degraded as the complexity of the problem increased. Here, complexity reflects the number of generating layers used to produce the landscape. Fig. 4 shows the relationship between the problem class and the average number of generations to find the solution using a homogeneous octagon topology. The value of A (discussed in detail in section five) relates to the problem entropy such that 1.01, 3.35, and 3.99 respectively represent fixed, periodic and chaotic problem classes [9]. μ is the average number of generations used to find the solution. As can be seen in the figure, as the entropy increases, more steps are needed to find the solution.



(a) Ring topology. *(b) Square topology.* *(c) Global topology.*



(d) Hexagon Topology. *(e) Octagon Topology.*

Fig. 3. The Homogeneous Topologies Used in the Previous Models.

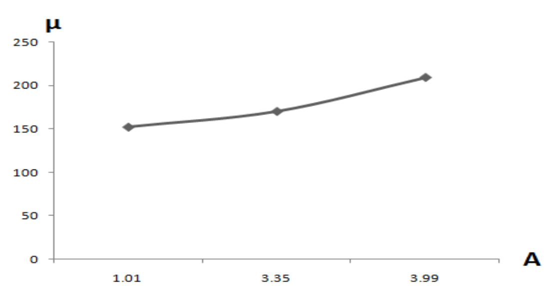


Fig. 4. Homogeneous Octagon Topology Used in Solving Three Different Problem Classes.

IV. Heterogeneous Topologies

The heterogeneous topology set consists of the six homogenous topologies [19] mentioned earlier, working together in the Population Space. An individual can change the topology hat they are using in each generation. Two wheels were used to configure the system in each generation. One is used to select the Knowledge Source and the other for the topologies. The topology wheel is spun first for each individual to establish their connections with others. Next, the Knowledge Source wheel is spun to select a Knowledge source to direct the individual A graphical description of the algorithm is given in Figure 5.

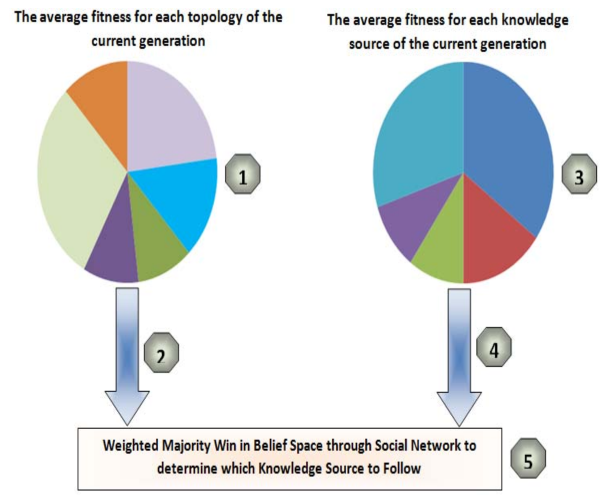


Fig. 5. Heterogeneous Model.

. Step one constructs the selection wheel for the topologies based upon their previous performance. The area under the wheel for each topology is its normalized average performance in the previous generation. In step 2 this wheel is used to select a topology for each individual in the population.. Next in step 3 the Knowledge Source wheel is updated based upon the previous performance of the Knowledge Sources. This wheel is then used in step four to generate the direct influence for each individual in the population and collect the direct influence Knowledge Sources for its neighbors. In step five, a weighted majority win conflict resolution rule is used to determine the winning Knowledge Source for each individual if the Knowledge Sources of its neighbors are different from its own.. The weight of each Knowledge source is based on the average fitness of the Knowledge source in directing individuals in the previous generation. The individual will then use the Knowledge Source with the highest score to influence it in the subsequent generation. The individuals are then modified and evaluated in the performance environment. The results are used to update the selection wheels and the process starts again for the next generation.

The Heterogeneous topology increases the performance of solving the problems by increasing the complexity of problem classes. Chaos landscapes are best exploited by heterogeneous networks, whereas static landscapes are best exploited by single homogeneous topologies. Heterogeneous topologies produce a social organization that is better able to exploit landscape produced by multiple generators than a single homogeneous topology.Keeping the Knowledge Sources and topology wheels separate from one another hides the relationship between the Knowledge Sources and the topologies used from the influence function selection process. Since Sub-Cultures can each contain a distinct organization topology along with a selective interest in knowledge sources relevant to their activities, it was of interest to see how this new information effects the performance.

V. SUB-CULTURED HETEROGENEOUS TOPOLOGIES

In the previous model the selection of the Topology and Knowledge Source was done independently of each other. However, subcultures can be characterized by their use of a social fabric topology that supports the Knowledge Sources used In the subculture model here the system learns to configure a subculture by identifying Knowledge Source Topology combinations that work well together.

This is done through the use of X by Y matries where X is the number of possible topological building blocks, and Y is the number of supporting Knowledge Sources. Here, the matrices will be of dimension 6 by 5. For each of the 6 topologies, information about its performance when paired with one of the five knowledge sources can be collected.

Thus, each cell (c[x, y]) in row x and column y contains a real value which represents the average fitness of the individuals who use topology x and Knowledge Source y and can be calculated by the following equations:

$$m = \sum_{i=1}^{n} \sum_{T_i = \mathrm{x} \text{ and } K_i = \mathrm{y}} 1 \qquad (1)$$

$$c[x, y] = \left(\sum_{i=1}^{n} \sum_{T_i = \mathrm{x} \text{ and } K_i = \mathrm{y}} f_i\right)/m \qquad (2)$$

N is the number of individuals; m is the number of individuals who use topology x and Knowledge Source y; T_i is the topology that connects individual i to other individuals; K_i is the Knowledge Source selected by individual i; and f_i is the fitness value of individual i. Equation (1) determines how many individuals are using topology x and following knowledge source y. Equation (2) calculates the average of fitness values of the individuals that use topology x and knowledge source y.

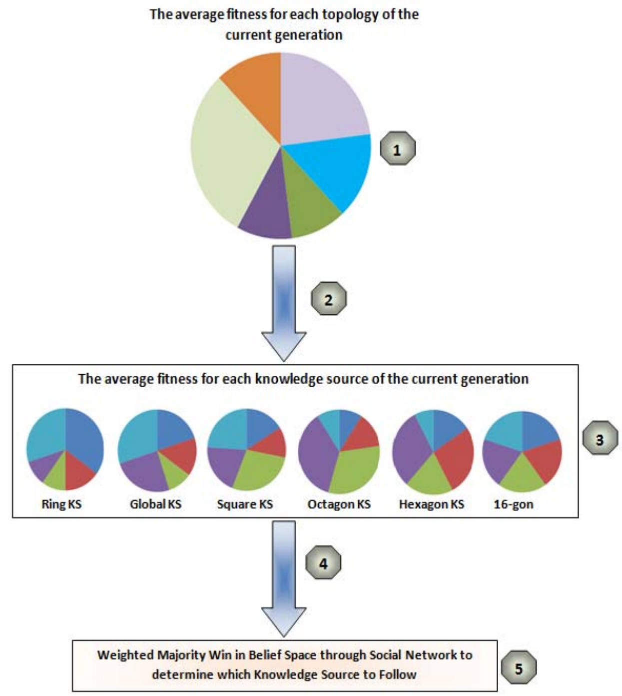


Fig. 6. Sub-Cultured Heterogeneous Model.

The Sub-cultured algorithm is shown graohhically in Figure 6 above. The Knowledge Source wheel will be divided into n wheels, where n is the number of topologies. Each wheel gives the individuals' average fitness of the Knowledge Sources for a specific topology. After the individual selects the topology, they will get the direct Knowledge Source by spinning the Knowledge Sources wheel related to its selected topology. Fig. 6 demonstrates how the process has been changed for the new Sub-Cultured heterogeneous model. Step one reflects the selection of one of the fixed topologies for use in a generation based upon the previous performance of each topology using a roulette wheel approach. The area under the wheel for each topology is its normalized average performance in the previous generation. The selected topology is then used to condition the selection one of the Knowledge Source wheels. Then, for each topology there is a separate wheel, shown in step three, which is used to select the Knowledge Source that is used to influence each individual based upon the past performance of the Knowledge Sources for the selected topology. Thus, there are six separate Knowledge Source wheels, one for each topology that is used for a generation. The wheel is used in step four in order to generate the direct influence for each individual in the population and collect the direct influence Knowledge Sources for its neighbors. In step five, the weighted majority win conflict resolution rules are used to determine the winning Knowledge Source for each individual as in the Heterogeneous model. The individuals are then modified and evaluated. The results are used to update the selection wheels and the process starts again for the next generation.

It is important to realize that although sub-cultures can emerge in this model, they do not have to. It may turn out that there is no significant performance dependence between the Knowledge Source and the Topologies. In that case the evolved matrices will not differ from those produced by the heterogeneous approach. The goal in the experiments is to identify those problem classes in which sub-suclutred configurations are likely to emerge.

VI. EXPERIMENTAL FRAMEWORK

The Cones World Generator [7] will be used to produce evolutionary problems in arbitrary complexities. Multi-dimensional landscape can be generated by scattering cones in different heights, slopes, and locations over the landscape. Therefore, the Cones World generator follows two stages to generate an evolutionary landscape: A static step, in which cones are placed on a landscape. Then, cones are combined to produce a continuous functional landscape using a max function. If two cones overlap, the maximum of the two will be taken. The second stage, then iteratively go through the landscape and dynamically adjust the cone parameters will be adjusted. The base landscape is given by the equation

$$f(<x_1, x_2, ..., x_n>) = \max_{j=2,k}(H_j - R_j * \sqrt{\sum_{i=2}^{n}(x_i - C_{j,i})^2}) \quad (3)$$

Where k is the number of cones, n is the dimensionality (2D), H_j is height of cone j, R_j is slope of cone j, and $C_{j,i}$ is the coordinate of cone j in dimension i. The values for each cone (H_j, R_j, and $C_{j,i}$) are randomly generated based on user specified ranges. The second step is to specify the dynamics, which gives the ability of deterministically changing the values of Cones' parameters like the slope, height, and location. The changes of Cone's parameters will be managed by the logistic function [7]:

$$Y_i = A * Y_{i-1} * (1 - Y_{i-1}) \quad (4)$$

Where, A is a constant with a range between 1.0 and 4.0, and Y_i is the value at iteration i. A bifurcation map of this function is provided in Fig. 7. This Figure shows relationship between the given values of A, and the logistic function results Y. A can be a single value (a small step-size change in the Cone's parameter), a double value (two step-size changes in the Cone's parameter), and an interval of values (many step-size changes in the Cone's parameters). It can be applied numerous time to produce a deeply nested performance landscape. That landscape can be viewed as an image form which distinctive patterns can be abstracted. As such, subcultures are viewed here as vehicles for deep social learning.

Morrison and Jong attached a logistic function with a specified A value to each parameter in the cones world. They gave the Cone's World four parameters: height, slope, x-axis location, and y-axis location. Three values of A were selected in order to test the behavior of the Cultural system in three different problem situations. The red, blue and green lines in Fig. 7 represent the chosen values of A. The values for A = (1.4, 1.8, 2.2 and 2.6) represent the one step-landscape generation and simulates the Fixed class problems. The periodic class problems can be simulated using a two-step generation process, using A values of 3.1, 3.2, 3.3 and 3.4 for A were chosen. Finally, chaotic class problems are produced by many generators due to the bifurcations, so A = (3.6, 3.7, 3.8 and 3.9) are a good choice for that. Fixed, periodic, and chaotic class problems are described and proposed by Langton [5].

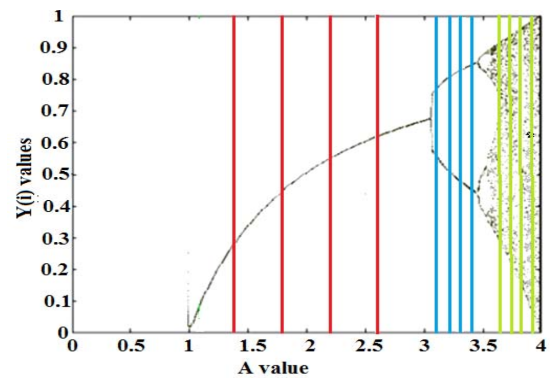


Fig. 7. *Logistic Function Y with Characteristic A Values.*

TABLE I. The performance Comparison of the Social Fabric TopologIES for POPULATION SIZE 50.

	Entropy	Homogeneous		Heterogeneous		Sub-Cultures	
	α	**μ**	**σ**	**μ**	**σ**	**μ**	**σ**
Fixed	**1.4**	92.38	37.24	170.20	60.86	147.36	76.61
	1.8	112.10	29.41	210.72	102.96	206.16	132.10
	2.2	133.48	21.92	241.88	102.42	209.48	64.86
	2.6	140.92	90.90	205.14	198.78	211.90	157.36
Periodic	**3.1**	229.08	170.41	248.90	154.21	231.76	125.01
	3.2	375.28	379.88	324.78	257.11	230.72	108.98
	3.3	329.68	239.43	271.70	160.49	178.72	57.55
	3.4	325.60	212.87	236.38	121.49	206.58	101.39
Chaotic	**3.6**	369.40	423.24	288.84	314.08	195.90	202.33
	3.7	468.44	359.44	373.68	290.78	264.88	133.98
	3.8	518.82	306.92	326.00	93.35	260.16	98.29
	3.9	482.06	386.87	358.46	198.20	287.32	213.89

In order to test how these problem categories are reflected by differences in the corresponding social organization of the Cultural systems needed to solve them, an example set of four landscapes from each of the three basic classes fixed (A=1.4, 1.8, 2.2 and 2.6), periodic (A=3.1, 3.2, 3.3 and 3.4) and chaotic (A=3.6, 3.7, 3.8 and 3.9) will be generated. The task here will be to measure the Sub-Cultured Heterogeneous Performance and compare it with the Homogeneous and Heterogeneous topologies' performance in solving problems from different complexity classes. The Square topology was selected to represent the Homogeneous social fabric category because it outperformed the other homogeneous topologies in previous experiments.

Each complexity (1.4, 1.8, 2.2, 2.6, 3.1, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8 and 3.9) has five randomly generated example Landscapes. A combination of ten independent runs for each landscape (from the generated example Landscapes) was tested on each of the proposed Homogeneous, Heterogeneous, and Sub-Cultured topologies. Fifty runs will be conducted as a total for each of the topologies on each of the complexities. Three population sizes were used to add more variability to our experiments. The population are 50, 75 and 100.

TABLE II. THE PERFORMANCE COMPARISON OF THE SOCIAL FABRIC TOPOLOGIES FOR POPULATION SIZE 75.

	Entropy	Homogeneous		Heterogeneous		Sub-Cultures	
	α	μ	σ	μ	σ	μ	σ
Fixed	1.4	88.64	54.50	100.34	41.59	117.18	65.70
	1.8	116.52	70.92	198.24	158.44	165.74	92.24
	2.2	134.70	44.58	175.94	84.79	151.64	70.71
	2.6	109.10	71.32	190.26	239.41	127.12	92.21
Periodic	3.1	143.24	108.76	204.50	147.15	124.66	76.70
	3.2	279.82	283.60	186.10	128.20	156.14	89.76
	3.3	225.10	126.25	181.02	93.61	161.66	63.29
	3.4	224.86	134.49	171.44	84.02	143.98	77.58
Chaotic	3.6	249.86	266.87	177.60	165.32	142.86	148.46
	3.7	286.44	240.85	224.18	195.38	191.84	138.56
	3.8	332.14	190.71	258.14	222.91	207.86	67.01
	3.9	329.46	311.18	274.62	218.82	230.00	150.70

VII. RESULTS AND ANALYSIS

In this section, the use of Sub-Cultured heterogeneous topologies will be compared with the homogeneous and heterogeneous topologies for the social fabrics. Their effect on the Knowledge Sources within the three complexity categories will also be tested (fixed, periodic, and chaotic categories).

Tables I, II, and III summarize the experimental results. Each row represents the entropy of the generated problem landscape. The homogeneous, heterogeneous and Sub-Cultures columns represent the individuals' interaction approach that was used to weave the social fabric. The Entropy column gives the complexity used in each of in the runs. The values in Columns (3 and 4), (5 and 6) and (7 and 8) give the average and standard deviation of the number of generations needed to solve a problem in the 50 runs using homogeneous, heterogeneous, and Sub-Cultured topologies, respectively.

For Static problems, the heterogeneous and Sub-Cultured versions are in general outperformed by the homogeneous topologies, as shown in the darkened cells of Tables I, II, and III under the homogeneous column (μ).

TABLE III. THE PERFORMANCE COMPARISON OF THE SOCIAL FABRIC TOPOLOGIES FOR POPULATION SIZE 100.

	Entropy	Homogeneous		Heterogeneous		Sub-Cultures	
	α	μ	σ	μ	σ	μ	σ
Fixed	1.4	68.50	45.40	91.86	68.78	88.9	90.11
	1.8	83.32	45.98	135.98	218.49	78.36	25.07
	2.2	117.56	64.80	128.58	44.78	97.96	43.31
	2.6	106.82	128.94	191.7	265.00	93.82	118.70
Periodic	3.1	114.82	88.75	196.82	106.93	124.34	101.88
	3.2	217.48	212.11	157.92	120.56	116.22	70.53
	3.3	182.20	105.84	162.08	59.03	123.84	80.19
	3.4	157.32	84.55	115.92	67.17	121.76	67.18
Chaotic	3.6	160.14	151.08	115.08	105.21	87.72	62.09
	3.7	349.74	199.34	251.16	213.43	169.3	144.08
	3.8	353.28	250.50	230.16	128.75	196.18	165.80
	3.9	331.94	264.44	248.64	186.93	220.76	161.91

The presence of just a single generator function did not require the additional complexity of either a heterogeneous topology or the emergence of Sub-Cultures. Their presence in an environment produced by just one generator process does not contribute to the effectiveness of the optimization process for those landscapes.

For the periodic class, (problems with A= 3.2, 3.3 and 3.4) as shown in Tables I, II, and III, the Sub-Cultured Heterogeneous configurations outperformed all of the other topologies in terms of the average number of steps needed to solve the problem.

TABLE IV. STATISTICAL T-TESTS (TWO TAIL TEST, ALPHA = 0.05) BETWEEN SUB-CULTURES AND HOMOGENEOUS FOR POPULATION SIZE 50.

	A-Value	Calculated t-value	Heterogeneous vs Homogeneous	Calculated t-value	Sub-cultures vs Heterogeneous	Calculated t-value	Sub-cultures vs Homogeneous
Fixed	1.4	7.71	Reject	1.65	Accept	4.56	Reject
	1.8	6.51	Reject	0.19	Accept	4.91	Reject
	2.2	7.31	Reject	1.88	Accept	7.84	Reject
	2.6	2.07	Reject	0.188	Accept	2.76	Reject
Periodic	3.1	0.60	Accept	0.61	Accept	0.08	Accept
	3.2	0.77	Accept	2.38	Reject	2.58	Reject
	3.3	1.42	Accept	3.85	Reject	4.33	Reject
	3.4	2.57	Reject	1.33	Accept	3.56	Reject

Chaotic	3.6	1.08	Accept	1.75	Accept	2.61	Reject
	3.7	1.44	Accept	2.40	Reject	3.75	Reject
	3.8	4.25	Reject	3.43	Reject	5.67	Reject
	3.9	2.01	Reject	1.72	Accept	3.11	Reject

For periodic class problems with entropy 3.1, Sub-Cultures outperformed the heterogeneous topologies and were very close to the homogeneous topologies' average number of generations. In fact, the Sub-Cultures were more effective than the homogeneous topologies in solving problems with entropy 3.1 for population size 75 as shown in Fig. 9, but were not for population sizes 50 and 100 as shown in Fig. 10 and Fig. 11. Overall, Sub-Cultures were the most effective topology in solving problems from the periodic class once a certain level of complexity was reached, A=3.2, and were gradually transitioning into a dominant position prior to that point as illustrated at A=3.1.

For chaotic class problems, when the number of generators becomes large, the utility of Sub-Cultures is effectively increased. The presence of Sub-Cultures in this chaotic environment does impact the number of generations needed to solve a problem successfully. The last 4 rows of Tables I, II and III show that Sub-Cultures were the most effective approach in solving chaotic class problems in terms of the number of generations. It may be that the presence of Sub-Cultures allows the system to focus on a solution once it is in the ballpark of the sub-cultures and distribute it to other subcultures for use.

The average number of generations needed to solve problems with entropies from fixed class problems (1.4, 1.8, 2.2 and 2.6), periodic class problems (3.1, 3.2, 3.3 and 3.4), and chaotic class problems (3.6, 3.7, 3.8 and 3.9) for homogeneous, heterogeneous and Sub-Cultures topologies are given in Fig. 8 through 10. The results suggest that when the number of layers of generators (complexity) increases, multiple networks are needed to focus on parts of the landscape instead of exploring the landscape as a single unit. Homogeneous topologies explore the landscape as a coherent unit but Sub-Cultured topology explores the landscape with small groups of individuals that form Sub-Cultures over parts of the landscape. Therefore, Sub-Cultures topology can solve problems efficiently when more layers of complexity added to the landscape and are not likely to emerge for shallow layered problems or images.

Several preliminary observations can be drawn from the three figures: (1). There is a level of complexity beyond which the heterogeneous and Sub-Cultures topologies outperform the homogeneous topologies for the population sizes 50, 75 and 100, (2).

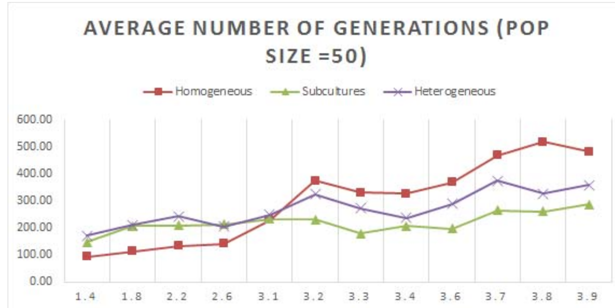


Fig. 8. *The Average Number of Generations Comparison between Homogeneous, Heterogeneous and Sub-cultures for Population Size 50.*

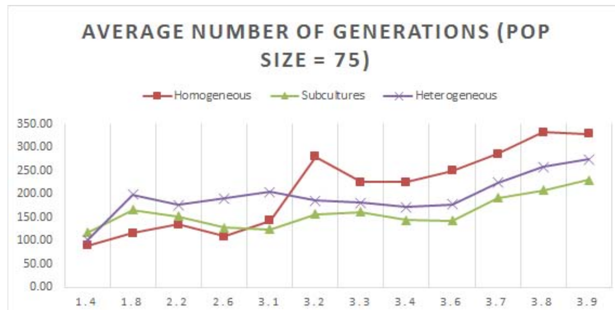


Fig. 9. *The Average Number of Generations Comprison between Homogeneous, Heterogeneous and Sub-cultures for Population Size 75.*

Sub-Cultures outperformed or equaled the heterogeneous approach for every given entropy except the simplest, (3). Sub-Cultures and heterogeneous topologies outperformed the homogeneous topology in chaotic problems class, (4). The number of generations taken to solve the problem with Sub-Cultures is less dependent on population size and landscape complexity than for the other two approaches, and 5). There appears to be a singularity, or region of complexity, where the best heterogeneous approach and the homogeneous approach perform at the same level. The former dominates after that and the latter strategy before that.

In Table IV the statistical significance of the performances between the two heterogeneous configurations and the best performing homogeneous topology in terms of the number of generations used for successfully solved problems is explored. As can be seen, for the fixed class the square homogeneous topology uses significantly more generations per solved problem than either heterogeneous or Sub-cultured approaches. The latter two configurations do not exhibit any statistical difference in terms of the number of generations used between them.

For the periodic class, Sub-Cultures use significantly fewer generations than either the heterogeneous or homogeneous topologies. There is a statistically significant difference in the number of generations used by the latter two approaches for entropy 3.4. It is clear that the presence of the Sub-Cultural structure in the heterogeneous topology is the key to its improved performance there.

In the chaotic situation with many generators, heterogeneity in the topologies becomes more important since both heterogeneous and Sub-cultured categories outperformed the homogeneous one. There was no observed statistical difference in performance between the two latters classes which suggests that Sub-Cultures do not have a significant impact on the number of generation used to solve a problem successfully. However, the presence of Sub-Cultures tends to reduce the standard deviation of the number of generations used to get a solution, even though it was not tested for significance here.

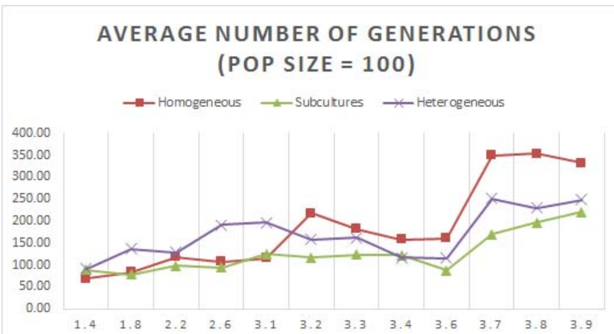


Fig. 10. *The Average Number of Generations Comprison between Homogeneous, Heterogeneous and Sub-cultures for Population Size 100.*

Predictability can be measured by standard deviations. In this section, the number of generations' standard deviations for each of the topologies (homogeneous, heterogeneous and Sub-Cultures) on different levels; (1) entropy values level as shown in Tables I, II and III; and (2) topologies (homogeneous, heterogeneous and Sub-Cultures) level is collected in Table V.

For entropies 1.4, 1.8, 2.2 and 2.6 and population sizes of 50 and 75 found in Tables I and II, homogeneous topologies were the most predictable by showing the lowest standard deviation in terms of the number of generations needed to solve fixed class problems. In the periodic and chaotic classes, problems with population sizes 50 and 75, Sub-Cultured topologies were the most predictable with the lowest standard deviation for entropies 3.1, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8 and 3.9 that are shown in Tables I and II. In the runs with population sizes of 100, homogeneous topologies were the most predictable for entropies 1.4 and 3.1, but the Sub-Cultured topologies were the most predictable for the rest of generator values. Heterogeneous topologies were more predictable than homogeneous topologies for entropies 3.2, 3.3, 3.4, 3.6, 3.7, 3.8 and 3.9 as shown in Tables I, II and III.

Again the value 3.1 is a critical entropy point in terms of social configurations and the use of Sub-cultured approach. The following conclusion can be extracted from Tables I through IV: (1). Homogeneous topologies were more predictable than heterogeneous topologies for entropies less than or equal to 3.1; (2). Sub-Cultured topologies were the most predictable for entropies greater or equal to 3.1 for population sizes 50 and 75. For population size 100, Sub-Cultured topologies were the most predictable for entropies greater than the critical entropy point 3.1; and (3) homogeneous topologies were the most predictable for entropies that were either less than 3.1 of population sizes 50 and 75, or less than or equal 3.1 of population size 100.

Table V shows the standard deviation of all runs for the homogeneous, heterogeneous, and Sub-Cultured topologies using the population sizes 50, 75 and 100. As shown in the table, heterogeneous topologies were more predictable than the homogenous topologies for all runs over all of the entropies that were used in the experiments. The Sub-Cultured topologies possessed the lowest overall standard deviation of all of the topologies tested. As such, it served to tie individuals closer together even as the problems that they were solving became more complex.

TABLE V. THE STANDARD DEVIATION COMPARISON OF THE HOMOGENEOUS, HETEROGENEOUS AND SUB-CULTURES SOCIAL FABRIC TOPOLOGIES FOR ALL RUNS.

Standard Deviation (All Runs)			
Population size	Homogeneous	heterogeneous	Sub-cultures
50	280.62	183.87	124.42
75	186.12	150.82	95.68
100	172.74	144.43	103.45

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, a Sub-Cultured heterogeneous topology was compared with other Homogeneous and hHeterogeneous topologies over twelve entropy values (1.4, 1.8, 2.2, 2.6, 3.1, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8 and 3.9). The results of the statistical tests conducted suggest in computational terms two basic ways why Sub-Cultures can emerge successfully in certain social situations. First, with multiple pattern generators in the search space, the Sub-Cultures can be useful in sorting out the different patterns produced by each generator, assuming of course that there are not too many. Second, the presence of Sub-Cultures tends to reduce the overall variability in the problem solving process along with dampening down the impact of increased problem complexity.

Homogeneous topologies were more efficient and predictable in problems class produced by a single generator layer with the tested entropies 1.4, 1.8, 2.2 and 2.6. Entropy value 3.1 was a critical point (or singularity) for the topologies to start defining their trends in solving different problem categories. Sub-cultures showed the highest predictability and efficiency in periodic and chaotic problems classes as was shown previously for entropies 3.2, 3.3, 3.4, 3.6, 3.7, 3.8 and 3.9.

The results suggest the Sub-cultures emerge as the number of generators producing the image or performance landscape increases. One advantage to reduce the variability in the performance within a population since diversity is partitioned into different subcultural groups.

In future work the goal will be to map real world problems into these complexity classes to that one can predict what social networks will be most effective. Possible set of problem categories can be image recognition from the Engineering realm, and the recognition Economic patterns from the social realm.

References

[1] M. Ali, "Using Cultural Algorithms to Solve Optimization Problems with a Social Fabric Approach," Ph.D., Computer Science Wayne State University, Detroit, MI, 2008.

[2] Y. Gawasmeh and R. Reynolds"A Computational Basis for the Presence of Sub Cultures in Cultural Algorithms" In Proceedings of the 2014 IEEE Symposium Series on Computational Intelligence (*IEEE SSCI 2014*), Orlando, FL, 2014.

[3] Z. Kobti, R. G. Reynolds, and T. Kohler, "A multi-agent simulation using cultural algorithms: the effect of culture on the resilience of social systems," in Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, Australia, 2003, pp. 1988-1995.

[4] Z. Kobti, R. G. Reynolds, and T. Kohler, "The effect of kinship cooperation learning strategy and culture on the resilience of social systems in the village multi-agent simulation," in Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, 2004, pp. 1743-1750.

[5] C. Langton, "Life at the Edge of Chaos," Artificial life II, pp. 41–91, 1992.

[6] D. Loiacono, J. Togelius, P. L. Lanzi, L. Kinnaird-heether, S. M. Lucas, M. Simmerson, D. Perez, R. G. Reynolds, and Y. Saez, "The WCCI 2008 Simulated Car Racing Competition," in Proceedings of the 2008 IEEE Symposium on Computational Intelligence and Games, Perth, Australia, 2008, pp. 119-126 .

[7] R. W. Morrison and K. A. De Jong, " A test problem generator for non-stationary environments " in Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, 1999, pp. 25-31.

[8] M. R. Raeesi and Z. Kobti, " Heterogeneous Multi-Population Cultural Algorithm " in IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013, pp. 292-299.

[9] R. G. Reynolds, "On Modeling the Evolution of Hunter-Gatherer Decision-Making Systems," Geographical Analysis, vol. 10, pp. 31- 46, 1978.

[10] R. G. Reynolds, "An Introduction to Cultural Algorithms," in Proceedings of the Third Annual conference on Evolutionary Programming, San Diego, CA, 1994, pp. 131-139.

[11] R. G. Reynolds, "Predicting the Past Using Multi-Agent Models of Cultural Evolution," presented at the Midwest Sociological Society, Chicago, IL, 2003.

[12] R. G. Reynolds, M. Ali, and R. S. Alomari, "Optimization Problem Solving using Predator/Prey Games and Cultural Algorithms," in Proceedings of the 2006 IEEE Symposium on Computational Intelligence and Games, Reno, NV, 2006, pp. 119-125.

[13] R. G. Reynolds, X. Che, and M. Ali, "Weaving the social fabric: The past, present and future of optimization problem solving with cultural algorithms," International Journal of Intelligent Computing and Cybernetics, vol. 3, pp. 561-592, 2010.

[14] R. G. Reynolds and C. Chung, "The Use of Cultural Algorithms to Support Self-Adaptation in Evolutionary Programming," in Proceedings of Adaptive Distributive Parallel Computing Symposium, Dayton, OH, 1996, pp. 260-271.

[15] R. G. Reynolds and Y. A. Gawasmeh, "Evolving Heterogeneous Social Fabrics for the Solution of Real valued Optimization Problems Using Cultural Algorithms," in proceedings of 2012 IEEE World congress on Computational Intelligence, Brisbane, Australia, 2012, pp. 1-8.

[16] R. G. Reynolds, J. O'Shea, X. Che, Y. Gawasmeh, G. Meadows, and F. Fotouhi, "The AGILE Design of Reality Game AI," in Multi-Agent Applications with Evolutionary Computation and Biologically Inspired Technologies: Intelligent Techniques for Ubiquity and Optimization, ed, 2011, pp. 193-207.

[17] R. G. Reynolds and B. Peng, "Knowledge Learning and Social Swarms in Cultural Systems," The Journal of Mathematical Sociology, vol. 29, pp. 115-132, 2005/04/01 2005.

[18] R. G. Reynolds, R. Whallon, M. Z. Ali, and B. M. Zadegan, "Agent-Based Modeling of Early Cultural Evolution," in Proceedings of 2006 IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada 2006, pp. 1135-1142.

[19] J. M. Stefan, R. G. Reynolds, F. Fotouhi, A. Aristar, L. Shiyong, and D. Ming, "Evolution based approaches to the preservation of endangered natural languages," in Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, Australia, 2003, pp. 1980-1987.