Graph-Theoretic Measure for Active iGAs: Interaction Sizing and Parallel Evaluation Ensemble

Xavier Llorà[1,2], Noriko Imafuji Yasui[1], and David E. Goldberg[1]

[1]Illinois Genetic Algorithms Laboratory, Dept. of Industrial and Enterprise Systems Eng.
University of Illinois at Urbana-Champaign, Urbana IL 61801

[2]Data Intensive Technologies and Applications, National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign, Urbana IL 61801

{xllora,niyasui,deg}@uiuc.edu

ABSTRACT

Since their inception, active interactive genetic algorithms have successfully combat user evaluation fatigue induced by repetitive evaluation. Their success originates on building models of the user preferences based on partial-order graphs to create a numeric synthetic fitness. Active interactive genetic algorithms can easily reduce up to seven times the number of evaluations required from the user by optimizing such a synthetic fitness. However, despite basic understanding of the underlying mechanisms there is still a lack of principled understanding of what properties make a partial ordering graph a successful model of user preferences. Also, there has been little research conducted about how to integrate together the contributions of different users to successfully capitalize on parallelized evaluation schemes. This paper addresses both issues describing: (1) what properties make a partial-order graph successful and accurate, and (2) how partial-order graphs obtained from different users can be merged meaningfully.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*

General Terms

Algorithms, design, theory, experimentation

Keywords

Active interactive genetic algorithms, graph theory, graph density, modeling user preferences, partial-order graph, graph ensemble.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

1. INTRODUCTION

Traditional interactive genetic algorithms (iGA) [3] require the user to provide a large number of evaluations to achieve good quality solutions. Due to the repetitiveness of the tasks, interactive sessions may last for hours usually inducing fatigue and frustration. Long durations make harder for the user to maintain the evaluation criteria during the whole evolutionary process. A key contribution of active interactive genetic algorithms (aiGAs) [17] is their capacity to build models of user preferences to generate *educated guesses*—promising solutions. Presenting *educated guesses* for user evaluation helps reducing the interaction fatigue—the user can obtain high-quality solutions faster—and the frustration—generating high-quality solutions allows aiGA to avoid the repetitive display of poor solutions that may discourage the user. Such an approach has shown that user evaluations can be greatly reduced—3 to 7 times.

A key element to the success of the aiGA paradigm, as shown in [17], is its ability to reconstruct a global ordering out of the partial-order graph built using the user evaluations. However, there is little knowledge about what make a partial-order graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ a good candidate to generate an accurate enough global ordering, or ranking, of the solutions contained in V. Moreover, when facing the design of aiGA there are three variables that need to be taken into account: the length of the solutions (ℓ), the growth ratio of nodes ($\Delta(\mathcal{V})$), and the growth ratio of edges ($\Delta(\mathcal{E})$). Usually, given a certain problem ℓ is fixed, but we still need to decide how often we need to generate a new node—solution—or show a new comparison to the user. In this paper we present a first analytic study based on graph-theoretic measures to separate accurate partial-order graphs (the ones that lead to an accurate ranking which will generate accurate solutions) from the inaccurate ones. As a result, we have identified a lower bound on the density of the partial-order graph. If the partial-order graph density is below that boundary, aiGAs will not be able to provide high quality solutions—as was empirically shown in [16].

Another daunting challenge for any interactive genetic algorithm is the integration of the results obtained from different runs. Moreover, those results may also come from different users and, thus, it is likely that results may differ—or even contradict. aiGAs can help mitigate this problem by as-

sembling multiple partial-order graphs together. Contradictory evaluations may be introduced by *user mistakes*, *perceptually indistinguishable solutions*, *contradictory targets*, and *criteria shifts*. In this paper we also focuses on laying the basis for assembling contradictions produced by user mistakes. We show that such mistakes can be probabilistically modeled using binomial distributions and, thus, simple contradiction resolution policies can be put in place to maximize the quality of the resulting ensemble.

The rest of this paper is organized as follows. Section 2 presents relevant background about active interactive genetic algorithms. Then, section 3 presents a graph-theoretic analysis of the partial-order graphs. It also shows a lower bound for the partial-order graph density that needs to be satisfied to be able to compute an accurate synthetic fitness—the basis for generating new solutions. Section 4 also presents a first approach to assembling partial-order graphs obtained from different interactive sessions. Finally, section 5 presents the conclusions for the work presented in this paper.

2. ACTIVE INTERACTIVE GENETIC ALGORITHMS

Dawkin's Blind Watchmaker program [7] and the Faceprints system developed at New Mexico State University [3] are two early examples of iGAs. For example, in Faceprints, the system replaces the role of a human sketch artist in evolving the faces of criminal suspects from witness recollection. Faces are encoded as binary strings where subcodes represent different facial features (nose type, mouth type, hair type, etc.). Each full chromosome maps to a face and the population of chromosome is presented to the human critic who is asked to determine how close the face resembles that of the criminal. This subjective ten-point scale is used to drive the evolution of subsequent generations of faces, and in a relatively short time, the GA arrives at a reasonable facsimile of the correct face.

The use of interactive genetic algorithms allow the fusion of human and computer efforts for problem solving [21]. However, putting the evaluation process into the hands of a user sets up a different scenario when compared to normal optimization. Takagi [21] presented a review of research efforts related to the iGAs challenges[1]. These main research areas highlighted included: (1) discrete fitness value input methods, (2) prediction of fitness values, (3) interfaces for dynamic tasks, (4) acceleration of iGAs convergence, (5) combination of evolutionary and non-evolutionary computation, (6) active intervention, and (7) theoretical research. A detail description of these topics is beyond the scope of this paper and detailed descriptions can be found elsewhere [21, 17].

Unlike in traditional evolutionary algorithms with objective fitness measures, one of the daunting challenges of iGAs is effective methods of combating user fatigue. Even for moderately-sized problems, iGAs may require a few hundred to a few thousand fitness evaluations, which is highly improbable—oftentimes even impossible—for users to evaluate. This places a premium on a variety of *efficiency-enhancement techniques* [11], particularly *evaluation relaxation* [19]. In evaluation relaxation schemes, the computationally costly, but accurate function evaluation is replaced by a cheap, but less accurate surrogate function. A serious stumbling block in developing surrogate fitness functions in iGAs is the absence of computable fitness function. Additionally, the user evaluation is relative and user preference might change over time. Hence, existing evaluation-relaxation methods fall short and cannot effectively model user fitness function.

[1]The author generalize his research under the umbrella of interactive evolutionary algorithms (iEAs)

Table 1: Algorithmic description of the aiGA model proposed by Llorà et al. (2005).

1.	Create an empty directed graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$.
2.	Create 2^h random initial solutions ($\mathcal{V}$ set).
3.	Create the hierarchical tournament set $\mathcal{T}$ using the available solutions in $\mathcal{V}$.
4.	Present the tournaments in $\mathcal{T}$ to the user and update the partialordering in $\mathcal{E}$.
5.	Estimate $\hat{r}(v)$ for each $v \in \mathcal{V}$.
6.	Train the surrogate ε-SVM surrogate synthetic fitness based on $\mathcal{G}$ and $\hat{r}(v)$.
7.	Optimize the ε-SVM synthetic fitness using the cGA.
8.	Create a $\mathcal{S}'$ set with 2^{h-1} new different solutions, where $\mathcal{V} \bigcap \mathcal{V}' = \emptyset$, sampling out of the probabilistic model evolved by cGA.
9.	Create hierarchical tournament set $\mathcal{T}'$ with $2^h - 1$ tournaments using 2^{h-1} solutions in $\mathcal{S}$ and 2^{h-1} solutions in $\mathcal{V}'$.
10.	$\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{V}'$
11.	$\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}'$
12.	Go to 4 while not converged.

2.1 Key Components

Active interactive genetic algorithms (aiGAs) [17] proposed to model user preferences in order to create a relaxation scheme. aiGAs propose a method which consists of three major components:

1. *Partial ordering of solutions:* The qualitative decisions made by the user about relative solution quality is used to generate partial ordering of solutions, or partial-order graph.

2. *Induced global order:* The concepts of non-domination and domination counts from multi-objective evolutionary algorithms [10] to induce a complete order of the solutions in the population based on their partial ordering.

3. *Surrogate fitness function via support vector machines:* The induced order is used to assign ranks to the solutions and use them in a support vector machine (SVM) to create a surrogate—or synthetic—fitness function that effectively models user fitness.

The key element of an aiGA is its synthetic fitness function. The minimal scenario for collecting meaningful domain-independent information from the user is provided by a binary tournament scheme ($s = 2$) [12]. User evaluations introduce a partial order among the solutions presented so far. This partial order can be expressed by using a partial-order graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$—as suggested in [17]. A vertex in $\mathcal{V}$ represents solutions presented to the user, whereas the edges in $\mathcal{E}$ represent the partial-order evaluations provided by the user. Given two solutions $\{s_1, s_2\} \in \mathcal{V}$ the user may provide three possible outcomes: (1) $s_1 > s_2$, (2) $s_1 < s_2$, and

(3) $s_1 = s_2$—or *equal/don't know/don't care*. Such a graph $\mathcal{G}$ can be transformed into a normalized graph $\mathcal{G}'$ containing only *greater than* relations. A global ordering measure may be computed using a heuristic based on two dominance measures, δ and ϕ, inspired by multiobjective optimization [5, 8]. Let $\delta(v)$ be the number of different nodes present on the paths departing from vertex v. Analogously, $\phi(v)$ is defined as the number of different nodes present on the paths arriving to v. The estimated fitness of a given solution v may be computed as $\hat{f}(v) = \delta(v) - \phi(v)$. Intuitively, the more solutions a vertex v dominates, the greater the fitness. Otherwise, the more solutions dominate a solution v, the smaller the fitness. The final global estimated ranking $\hat{r}(v)$ is obtained sorting the vertex $v \in \mathcal{V}$ by $\hat{f}(v)$. This global estimate $\hat{f}(v)$ is used to train a ε-SVM for creating the synthetic surrogate fitness [17]. By optimizing such a synthetic fitness we can obtain *educated guesses*–candidate solutions–about the user preferences. The optimization step is conducted by using the compact genetic algorithm (cGA) [14] obtaining a probabilistic model of the user preferences.

2.2 Interesting properties

Llorà et al. [17] pointed out that partial-order graphs have an interesting property: given a normalized partial-order graph $\mathcal{G}'$, if a vertex k appears more than once in a path between u and v, then a cycle exists, hence, it represents an inconsistency in the user evaluations. Thus, due to the *greater than* relations (contained in $\mathcal{E}$), the consistency of the user evaluations can be identified. This property is the basis of the consistency metric [15]. In order to compute such a measure the authors need two components: *i*) cycle detection capabilities for a given graph $\mathcal{G}'$ at time t ($\mathcal{G}'^t$), and *ii*) an heuristic to quantify how much inconsistency the detected cycle is introducing. A detailed explanation of this property can be found elsewhere [15].

The estimated fitness is a key component of aiGA. It assigns a numeric estimate to each solution presented to the user and, hence, it is a key component to the probabilistic model building component mentioned below. It also allows measuring the error of the synthetic fitness surrogate when compared to the original estimated values. Such error measuring acts as a sanity check of the surrogate, since an erroneous surrogate will lead to unreliable models of the user preferences.

The original aiGA proposed by Llorà et al. [17] used the cGA [14] to obtain a probabilistic model of the user preferences by optimizing the surrogate fitness function constructed. This process is repeated every generation—see Table 1. Hence, during interaction between the user and the aiGA several probabilistic models are evolved. The original aiGA discards the previous model when a new one is obtained. However, the sequence of models provides interesting insides about the evolution of the user preferences along the run. For instance, sustained criteria on the evaluation process by the user will be reflected by slowly changing sequences of models. However, drastic changes on the user criteria may show up as sequences of dispare model sequences.

2.3 Problematic Incremental Usage

The original aiGA model proposed by Llorà et al. [17] assumes a generational approach to the interactive process—as show in Table 1. Later on, Llorà et al. [16] proposed an incremental approach to the collection of user evaluations. They maintained the steady-state approach of the original aiGA. At each evaluation step, a tournament is performed between two previously evaluated solutions that were not compared directly, or between a previoulsy evaluated solution and a solution sampled from the probabilistic model of the user preferences.

Such a tournament may result in three possible outcomes: *greater than*, *lesser than*, and *equal to*. However, the authors also introduced a key difference that become problematic. When two solutions were evaluated t*equal*, the original aiGA required a normalization step to transform the graph into another one that only includes strict comparison relations. However, that step was eliminated reducing the amount of comparisons stored by the partial ordering graph. That is, if two solutions were evaluated as *equal*, no edge between the solution vertex was introduced, which lead as we will show later to graphs not dense enough to provide an accurate reconstruction of the global order—as presented in section 2.1.

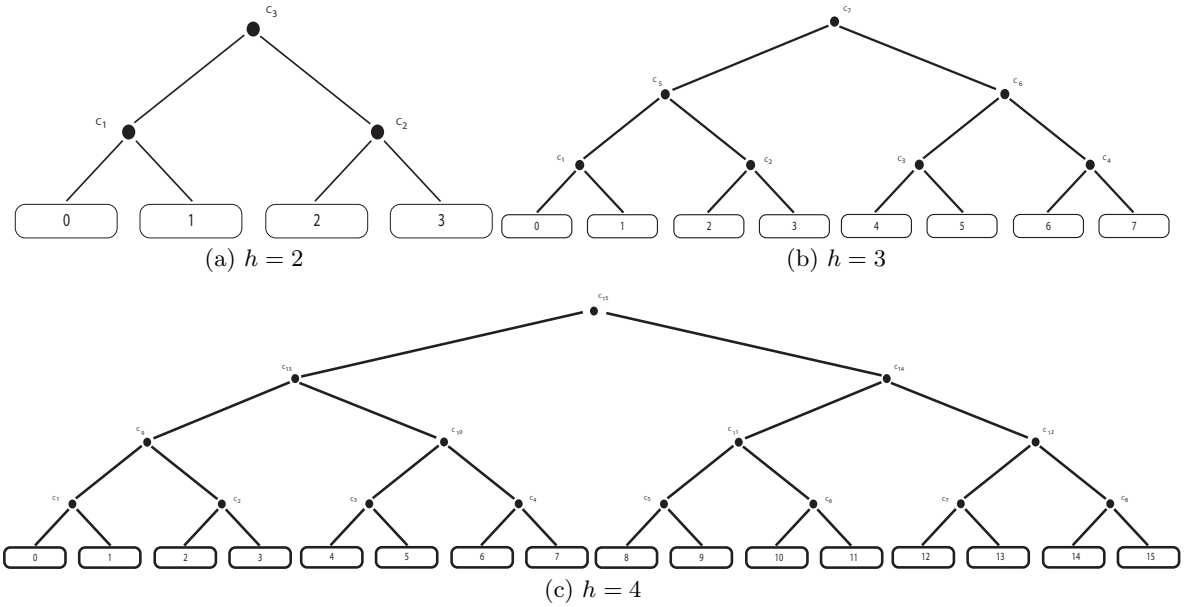
Results using this approach were worst and more unstable that the ones using the original steady-state scheme [16]. Next section presents a graph-theoretic analysis helps explains why the incremental approach was condemned to fail whereas the original steady-state approach was able to reconstruct, accurately enough, the global order required to build the synthetic fitness.

3. INTERACTION SIZING IN AIGAS

A key element to the success of the aiGA, as shown in [16], is its ability to reconstruct a global ordering out of the partial-order graph built using the user evaluations. However, there is little knowledge about what makes a partial-order graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ a good candidate to generate an accurate enough global ordering, or ranking, of the solutions contained in $\mathcal{V}$. Moreover, when facing the design of aiGA there are three variables that need to be taken into account: the length of the solutions (ℓ), the growth ratio of nodes ($\Delta(\mathcal{V})$), and the growth ratio of edges ($\Delta(\mathcal{E})$). Usually, given a certain problem ℓ is fixed, but we then need to decide how often we need to generate a new node or show a new comparison to the user.

To illustrate the interaction-sizing dilemma, we will review two basic opposite strategies: (1) to generate a new vertex (solution) every time we require a comparison from the user, and (2) only generate a new vertex (solution) when there is no comparison left between the one already present in $\mathcal{G}$. From the user perspective, the first one will always provide new elements to compare alleviating the burden of repetitiveness problem, but would provide a poor global ranking. On the other hand, the second one will increase the perception of repetitiveness and fatigue, but a completely connected partial-order graph provide a perfect estimate to reconstruct the global ranking of the solutions.

These strategies lead to graph with very different structural properties. Whereas the first one will have a large number of unconnected components (each of them form by 2 vertexes), the second one will eventually be a fully connected graph—structurally similar to the proposed incremental method in [17]. It is important to mention here, that there is a lower bound on the number of initial vertexes



(a) $h = 2$ (b) $h = 3$

(c) $h = 4$

Figure 1: Hierarchical tournament selection scheme used by aiGA. The parameter h defines the height of the tournament tree. The first 2^{h-1} solutions are drawn at random without replacement from the set of already available solutions. Solution $2^{h-1}+1$ is the best ranked solution so far, Finally, the rest of the solutions are promising solutions obtained after optimizing the synthetic fitness built on the previous iteration.

required[2]. However, satisfying that lower bound does still not guaranty that resulting graph $\mathcal{G}$ will be useful, since that is determined by the interactive policy selected.

For the above-mentioned reasons, we focused on identifying a graph-theoretic measure [13, 4] that could clarify what is a good graph structure for the purposes of computing the global ranking. Since there seams to be a tradeoff between the number of nodes ($|\mathcal{V}|$) and the number of edges ($|\mathcal{E}|$), the graph *density* provides a proper unified measure across different graphs.

Definition 1. *The graph density for a directed graph is defined as:*

$$\rho(\mathcal{G}) = \frac{|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)} \tag{1}$$

Whereas the density for a non-directed graph is defined as:

$$\rho(\mathcal{G}) = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)} \tag{2}$$

Following the theoretical framework setup in [17], we also used a *perfect user*, as defined below, to conduct the analysis of the graph structure required for the aiGA. This assumption will then be revisited in the next section.

Definition 2. *Given two solutions u, v, a perfect user Υ is a function defined as:*

$$\Upsilon(u,v) = \begin{cases} < v, u > & \text{if } f(u) < f(v) \\ < u, v > & \text{if } f(u) > f(v) \\ \lambda & \text{if } f(u) = f(v) \end{cases} \tag{3}$$

[2]As shown elsewhere [6, 20, 17], the number of initial nodes is proportional to the number of training example required to properly train the ϵ-SVM regressor used to construct the synthetics fitness. A detailed explanation is beyond of this paper and explained in detail elsewhere[17].

where f is the ideal target function, and λ stands for the null edge.

The rest of this section assumes that f is the OneMax function [10] following the original aiGA facet-wise analysis [17]. Also, assuming the usage of a *perfect user*, the density definition presented in Definition 1 for a partial-order graph can be reformulated as:

Definition 3. *The graph density for a partial-order graph given a perfect user Υ is defined as:*

$$\rho(\mathcal{G}_{\mp}) = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)} \tag{4}$$

When using a *perfect user* Υ, Definition 3 presents the same density form of equation 2. This is the result of the Υ behavior. Given two nodes u and v the will only return one of the two possible edges $\{< u, v >, < v, u >\}$.

The density of the graph generated by the hierarchical tournament method used by the original aiGA—see Figure 1—requires computing the number of vertxes and edges after each round i of tournaments. The number of nodes nn in a partial-order graph at a given tournament round i of height h can be computed as:

$$nn(i,h) = 2^h + i \cdot \left(2^{h-1} - 1\right) \tag{5}$$

Whereas the number of edges ne at a given tournament round i of height h can be computed as:

$$nn(i,h) = (i+1)(2^h - 1) \tag{6}$$

Hence, the density of the partial-order graph at a given tournament round i of height h can be expressed as:

$$\rho(G^{i,h}) = \frac{2 \cdot ne(i,h)}{nn(i,h)(nn(i,h) - 1)} \tag{7}$$

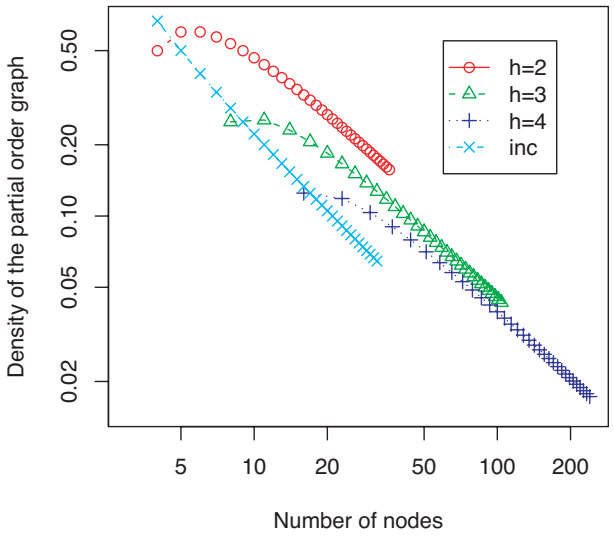
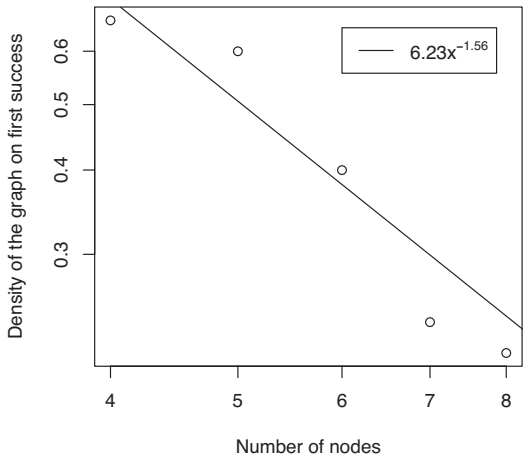


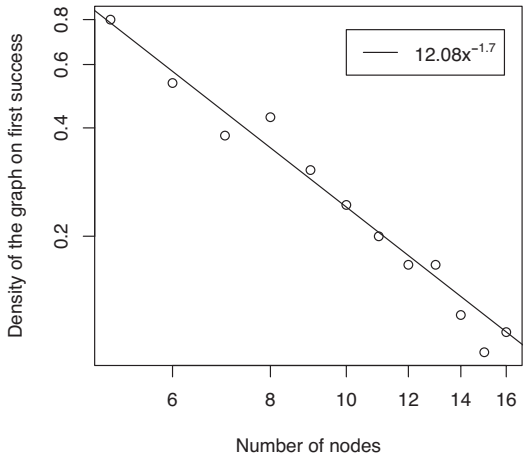
Figure 2: Density of the partial-order graph create by a hierarchical tournament scheme given the height h and the tournament round i. The figure plots the densities for $h = \{2, 3, 4\}$ and $i \in [0, 32]$. The figure also highlights the quick drop of density of the incremental approach.

Figure 2 presents the density across different heights and iterations for the hierarchical tournament in the original aiGA. Using the same density measure, we revised the incremental policy proposed in [16]. Figure 2 shows a clear change in slope and density drop for the incremental approach [16]. It also does not allow to scale based on the problem size ℓ, violating the requirements of the ϵ-SVM regressor, which may lead to inaccurate regressions. Using the density measure we were able to explain why the incremental tournament approach proposed in [16] did not work. However, this does not answer why the hierarchical tournament scheme provides accurate global rankings. In order to be able to answer such a question, we prepared a controlled experiment in order to characterize the boundary separating accurate and inaccurate partial-order graphs.

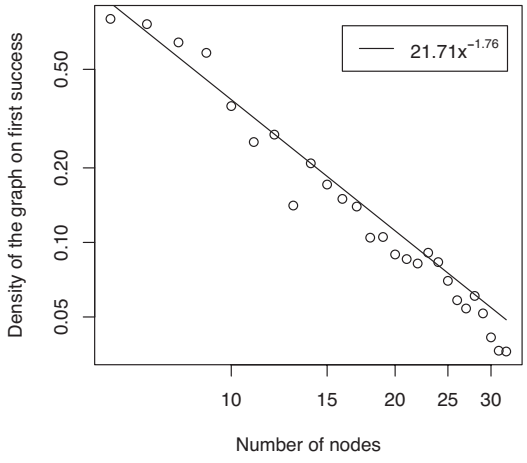
We can label a partial-order graph as an accurate one if and only if, after optimizing the synthetic fitness produced, the best solution obtained has correctly fixed at least $m-1$ building blocks [11]. In the particular case of solving One-Max, the best solution generated has at least all but one bit set to 1, otherwise, we will label such a partial-order graph as inaccurate. To identify such a transition, we proceed as follows. Given a certain problem length ℓ, we generated 75 random graphs for each of the possible number of nodes and edges. That is, for $\ell = 2$ we would generate random graph containing $\{1, 2, 3, 4\}$ nodes, and for each number of nodes we generated random graphs for all the possible number of edges $1 \rightarrow \{\}, 2 \rightarrow \{1\}, 3 \rightarrow \{1, 2, 3\}, 4 \rightarrow \{1, 2, 3, 4, 5, 6\}$. Hence, for $\ell = 2$ we generated $75 \cdot (1 + 1 + 3 + 6) = 825$ random graphs. For each of them we computed their density ρ and the probability of success (proportion of accurate partial-order graphs). Figure 3 shows that for a given number of nodes there is a required density to achieve an accurate partial-order graph—all the 75 graphs for that density were accurate. After achieving an accurate graph for the



(a) $\ell = 3$



(b) $\ell = 4$



(c) $\ell = 5$

Figure 3: Figures a, b, and c presents the density required to hit the optima using a randomly generated graph with the given number of nodes. Each point represents the average density of 75 successful independent trials—all random graphs induced a solution after the synthetic fitness optimization which had correctly fixed at least $m-1$ building blocks.

first time given a certain number of nodes, increasing the density just keeps generating accurate graphs.

Figure 3 presents the required density lower bound required to ensemble accurate partial-order graphs based on the user evaluations. Figure 2 and 3 need to be carefully compared. There is only one case that is directly comparable, the case where $\ell = 4$—Figure 3(b)—and we use a hierarchical tournament of height $h = 2$—worst case scenario. The densities of the hierarchical tournament scheme of the aiGA [17] and the boundary obtained align nicely, following the same power law—$\rho(G_{HTS}) \approx \rho(G_{rnd})$. The tail between $\ell = 4$ and $h = 2$ eventually cross each other the balance, but on an interactive GA session that would happen far after the user has finished the experiment exhausted. If we increase the height by 1, then the produced partial-order graph densities are higher than required which guarantees the creation of an accurate partial-order graph. Hence, the hierarchical tournament [17] unexpectedly turned out to be closer to the density boundary required to produce accurate partial-order graphs, which also explains why the incremental approach [16] did not work—it did not satisfy the lower density boundary.

4. PARALLEL EVALUATION ENSEMBLE IN AIGAS

Another daunting challenge for any interactive genetic algorithm is how to integrate the results obtained from different runs. Moreover, those results may be generated by different users and, thus, it is likely that results may differ. Traditional approaches try to combine (for instance averaging) the solutions obtained from the different interactive sessions. Researchers [9] have mostly focused on combining the end products of the interactive evolution. However, aiGAs beside providing a handful of good candidate solutions at the end of the run, they also provides the partial-order graph. Such information can be of great value. As shown in [1], partial-order graphs can help eliminate user contradictions by guided reevaluation of cycles—a sign of user contradictory evaluation criteria.

Thus, our approach to generate a final answer after n interactive session will involve assembling the n partial-order graphs to create a final synthetic fitness function. The promising candidates resulting from the synthetic fitness function will form the answers to the overall problem. The goal is to embed all knowledge gathered by those partial-order graphs looking for a robust methodology—combining final solutions has been shown to produce average solution of poor quality [9]. Figure 4 illustrates how such an ensemble may be constructed.

The process of building the ensemble $\hat{\mathcal{G}}$ from a set of partial-order graphs $\{\mathcal{G}_0, \mathcal{G}_1, \ldots \mathcal{G}_n\}$ has to consider different sources of possible contradictions. In another words, what could be the different causes of having edges such as $< u, v >$ and $< v, u >$ competing to enter the final ensemble $\hat{\mathcal{G}}$. A set of possible, but not all, causes is listed below.

- **User mistakes:** During the interactive process the user just got distracted and provided the wrong tournament evaluation adding a contradictory evaluation.
- **Perceptually indistinguishable:** The interface of the interactive process is not refined enough. Hence, given two solutions u and v, the user is not able to perceptually distinguish the difference between them. This situation increases the risk of providing contradictory evaluations.
- **Contradictory targets:** When dealing with aesthetic evaluations, the objective target that each user may have in mind can clearly differ. A simple analogy would be a multimodal problem where two different solutions provide the same fitness. In such situations, blindly containing the partial-order graph will lead to mediocre average solutions.
- **Criteria shifts:** The interactive nature of the process can also shift the user criteria. Such criteria shifts tend to appear in long lasting interactive sessions [2]. Users can change their desired target based on the interactive process itself. That is, users may find a new point of interest that suddenly became much more interesting than the original one, producing the criteria shift.

Addressing all these possible sources of user evaluation contradictions is beyond the reach of this paper, and will require further research. However, in this section we focused on laying the ground for the first source of possible evaluation contradictions: *user mistakes*. As mentioned earlier, during the interactive aiGA process the user just got distracted and provided the wrong tournament evaluation adding a contradictory evaluation. Such kind of mistakes can be easily modeled as a binomial distribution. Let p_{er} be the probability of edge reversal due to a user mistake. Thus, for each pair of adjacent solutions (vertexs) u and v, we can model the probability of edge reversal ($e =< u, v >$ and $e' =< v, u >$ the reversed edge) as a binomial distribution $B_e(n, p)$, where n is the number of graph G_i that contain e or e', and p the probability of user mistake—usually p is small thanks to the users efforts ($p < 0.05$). Thus, the expected number of reversed edges can be written as np. Since p is a small number, in order to maximize the probability of choosing between e and e' the right edge, a simple strategy would be to pick the one with larger support—usually know as majority rule [18, 22].

5. CONCLUSION

This paper has presented a first analytic study based on graph-theoretic measures to separate accurate partial-order graphs (the ones that provide an accurate ranking of accurate solutions) from the inaccurate ones. As a result, we have identified an existent lower bound based on the density of the partial-order graphs. If the partial-order graph lower density is below that bound, the aiGA will not provide high quality solutions. The existence of this lower bound explain previous failed attempts conducted by researchers to replace the original hierarchical steady-state tournament evaluation aiGAs with incremental versions.To create an accurate global ranking using a partial-order graph, such a graph requires a density above the lower bound identified.

We have also addressed the problem of creating partial-order graph ensembles based on graph provided by previous aiGA sessions. We have focused on how we can model user mistakes providing a strategy to build accurate partial-order ensembles. Such mistakes can be probabilistically modeled as binomial distributions and, thus, simple contradiction resolution policies can be put in place to maximize the quality of the resulting ensemble based on the majority rule.

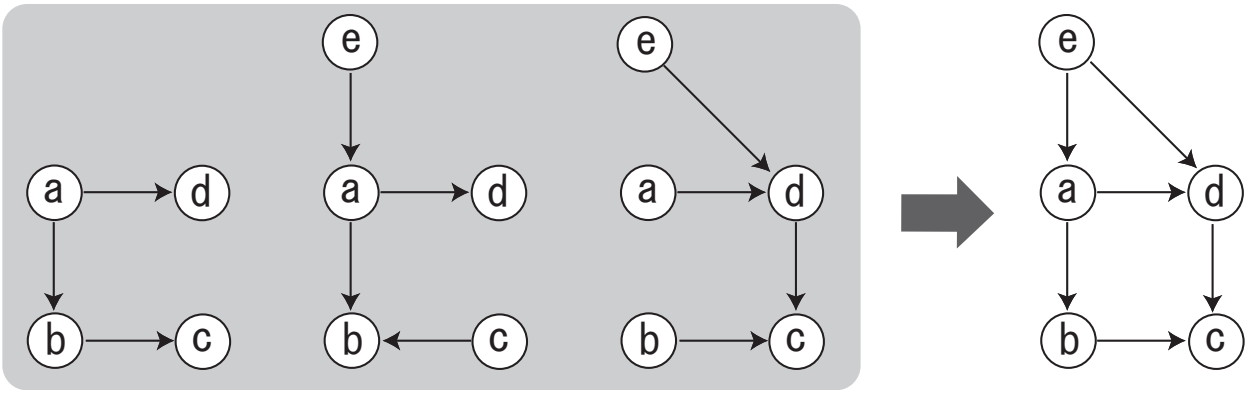


Figure 4: Creating a new partial-order graph by combining the graphs obtained in interactive sessions using aiGA. Conflicts may arise because of contradictory evaluations—see edge between b and c. Those conflicts must undergo a conflict resolution policy. User mistakes can be modeled using binomial distributions. Under those assumptions, the optimal resolution of each edge inconstancy is to choose the edge with maximum support—majority rule. The figure shows how three partial-order graphs are combined into a final ensemble, after resolving the contradiction edge between b and c.

6. ACKNOWLEDGMENTS

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096 and under grant ISS-02-09199 at the National Center for Supercomputing Applications, UIUC. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

7. REFERENCES

[1] Alías, F., Llorà, X., Formiga, L., Sastry, K., and D.E., G. Efficient interactive weight tuning for TTS synthesis: reducing user fatigue by improving user consistency. In *Proceedings of ICASSP* (Toulouse, France, May 2006), vol. I, pp. 865–868.

[2] Alm, C. O., and Llorà, X. Evolving emotional prosody. In *Proceedings of the Ninth International Conference on Spoken Language Processing (INTERSPEECH 2006)* (2006), p. paper 1741.

[3] Caldwell, C., and Johnston, V. S. Tracking a criminal suspect through face-space with a genetic algorithm. In *Proceedings of the Fourth International Conference on Genetic Algorithms* (1991), Morgan Kaufmann, pp. 416–421.

[4] Clark, J., and Holton, D. A. *A First Look at Graph Theory*. World Scientific, 1991.

[5] Coello-Coello, C. A. An updated survey of GA-Based Multiobjective Optimization Techniques. Technical report lania-rd-09-08, Laboratorio Nacional de Informática Avanzada (LANIA), Xalapa, Veracruz, México, December, 1998.

[6] Cristianini, N., and Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge Press, 2000.

[7] Dawkins, R. *The blind watchmaker*. New York: W. W. Norton, 1986.

[8] Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, 2000.

[9] Formiga, L., and Alías, F. Extracting user Preferences by GTM for aiGA Weight Tuning in Unit Selection Text-to-Speech Synthesis. *Lecture Notes in Computer Science*, 4507 (June 2007), 654–651.

[10] Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning*. Adison-Wesley, 1989.

[11] Goldberg, D. E. *The design of innovation: Lessons from and for competent genetic algorithms*. Kluwe Academic Publisher, 2002.

[12] Goldberg, D. E., Korb, B., and Deb, K. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems 3*, 5 (1989), 493–530.

[13] Harary, F. *Graph Theory*. Addison-Wesley Publishing Company, 1971.

[14] Harik, G., Lobo, F., and Goldberg, D. E. The compact genetic algorithm. *Proceedings of the IEEE International Conference on Evolutionary Computation* (1998), 523–528. (Also IlliGAL Report No. 97006).

[15] Llorà, X., Goldberg, D. E., Ohsawa, Y., Matsumura, N., Washida, Y., Tamura, H., Yoshikawa, M., Welge, M., Auvil, L., Searsmith, D., Ohnishi, K., and Chao, C.-J. Innovation and Creativity Support via Chance Discovery, Genetic Algorithms, and Data Mining.

Journal of New Mathematics and Natural Computation 2, 1 (2006), 85–100.

[16] Llorà, X., Sastry, K., Alías, F., Goldberg, D. E., and Welge, M. Analyzing active interactive genetic algorithms using visual analytics. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation* (New York, NY, USA, 2006), ACM, pp. 1417–1418.

[17] Llorà, X., Sastry, K., Goldberg, D. E., Gupta, A., and Lakshmi, L. Combating user fatigue in iGAs: Partial ordering, support vector machines, and synthetic fitness. In *Proceedings of Genetic and Evolutionary Computation Conference 2005 (GECCO'05)* (Washington DC, USA, June 2005), ACM, pp. 1363–1370.

[18] Mitchell, T. M. *Machine Learning*. McGraw Hill, 1997.

[19] Sastry, K. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master's thesis, University of Illinois at Urbana-Champaign, General Engineering Department, Urbana, IL, 2001. (Also IlliGAL Report No. 2002004).

[20] Shawe-Taylor, J., and Cristianini, N. *Kernel Methods for Patter Analysis*. Cambridge Press, 2004.

[21] Takagi, H. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. In *Proceedings of the IEEE* (September 2001), vol. 89, pp. 1275–1296.

[22] Witten, I. H., and Frank, E. *Data Mining: practical machine learning tools and techniques with java implementations*. Morgan Kaufmann, 2000.