

# Towards Creative Design Using Collaborative Interactive Genetic Algorithms

Juan C. Quiroz, Sushil J. Louis, Amit Banerjee, and Sergiu M. Dascalu

**Abstract**—We present a computational model of creative design based on collaborative interactive genetic algorithms. We test our model on floorplanning. We guide the evolution of floorplans based on subjective and objective criteria. The subjective criteria consists of designers picking the floorplan they like the best from a population of floorplans, and the objective criteria consists of coded architectural guidelines. We support collaboration by allowing individual designers to view each others' designs during the evolutionary process and the sharing of designs via case injection. This methodology supports team design, and reflects the view of creativity that collaboration accounts for much of our intelligence and creativity. We present a description of the model and a comparative study of floorplans created individually versus collaboratively. Our results show that floorplans created collaboratively were considered to be more “revolutionary” and “original” than those created individually.

## I. INTRODUCTION

Design is a fundamental, purposeful, pervasive and ubiquitous activity and can be defined as the process of creating new structures characterized by new parameters, aimed at satisfying predefined technical requirements. It consists of several phases, which differ in details such as the depth of design, kind of input data, design strategy, procedures, methodology and results [19]. Usually the first stage of any design process is the preliminary or the conceptual design phase, followed by detailed design, evaluation and iterative redesign [3]. Computers have been used extensively for all these stages of design except the creative conceptual design phase. We are interested in supporting the creative conceptual design phase by not only saving and disseminating the initial ideas of designers, but also by providing the support for initial design ideas to serve as the seeds on which new designs are founded. Interactive genetic algorithms (IGAs) have been proposed as user guided innovation pumps [13]. We propose a computational model of creative design based on collaborative interactive genetic algorithms to support creativity in design. Our model allows designers to guide evolutionary computation to breed new design ideas quickly, while supporting a team collaborative aspect, consisting of the sharing of ideas among designers.

Interactive genetic algorithms (IGAs) differ from GAs in that the objective fitness evaluation is replaced with user

evaluation, thus allowing for the user to guide an explorative evolutionary process when there is no better fitness measure than the one in the human mind [22]. With subjective evaluation IGAs can allow a designer to explore the space of possible designs, even if the requirements include aesthetic and subjective criteria. Furthermore, it has been argued that much of our creativity and intelligence comes from our interaction with peers [5], [8]. We follow along this line of thought by supporting collaboration amongst a team of designers by allowing them to view each others' designs during individual evolutionary sessions with an IGA. Sharing among the designers is supported via case injection.

In this paper we present an extension of the work by Banerjee et al. in [2], where the collaborative IGA computational model of creativity was first introduced. Banerjee et al. only describe pretests in their work, and do not conduct a thorough evaluation of the model. The authors of the model described create and choose a representative set of floorplan designs that were then evaluated by 10 computer science graduate students. The students were asked to evaluate the floorplans in terms of “practicality” and “originality”. The pretest results showed that floorplans designed collaboratively were ranked higher in “originality” than those created individually. In this paper we conduct further evaluation of the computational model of creative design. A group of 20 participants, broken into five groups of four, took part in a user study. Participants in the groups created floorplans individually and collaboratively, and then evaluated the created floorplans using a criteria subset from the Creative Product Semantic Scale [4], [1]. Our hypothesis is that the support of collaboration through the exchange of design solutions between participating peers will result in designs that are more creative than designs created individually.

We use Gero's definition of creative designing, which states that the addition of variables during the design process has the potential, but does not guarantee, to produce creative content [10]. The work presented by Banerjee et al. in [2], and in this paper, does not expand the search space by adding variables. Instead, assuming that a large design solution space is being explored, we ask whether collaboration amongst peers is sufficient to allow for the potential to produce creative content in designing. Our results show that floorplans created collaboratively were considered to be more “revolutionary” and more “original” than those created individually.

The rest of the paper is organized as follows. First we present related work on computational models of creative design, IGAs, and computer supported collaborative work.

Juan C. Quiroz, Sushil J. Louis, and Sergiu M. Dascalu are with the Evolutionary Computing Systems Lab, Department of Computer Science & Engineering University of Nevada, Reno Reno, NV 89557, USA. Email: {quiroz, sushil, dascalus} @ cse.unr.edu.

Amit Banerjee is with the School of Science, Engineering and Technology The Pennsylvania State University at Harrisburg Middletown, PA 17057, USA. Email: aub25@psu.edu.

Second, we describe the computational model of creative design in detail, along with the framework we implemented to test the model. Third, we discuss details on the floorplanning case study, including representation and specialized genetic operators used. Next we describe the experimental user study and discuss the results obtained. We conclude with directions for future work.

## II. RELATED WORK

The work presented in this paper encompasses the use of several technologies and various active areas of research, including computational models of creative design, interactive evolutionary computation, and computer supported collaborative work.

### A. Computational Models of Creative Design

Computational models of creative design have been researched since the early 90s due to the popularity and capabilities that artificial intelligence (AI) brought on. The potential for a computational model of design to produce creative content relies on expanding the solution search space by adding one or more variables [10]. A creative design process does not guarantee the production of creative solutions; rather it provides the potential for created solutions to be creative. It is the role of the designer to initially determine the creative value of the produced designs.

The design literature presents various types of design activities [9], [10]. Routine design occurs when all variables associated with a design are known a priori before exploration of design solutions. Non-routine design breaks down into two types of design: innovative and creative design. In innovative design the boundaries to design variables are allowed to go beyond practical and known “good practice” bounds. In creative design we allow the addition of one or more variables, thus expanding the design search space, allowing for potential creative solutions to be discovered. Models of creative design which have been presented by the research design community manipulate the search space through the use of various techniques, including combination, analogies, transformation, emergence, and first principles [10]. Our collaborative IGA computational model of design is the first to use IGAs to guide the exploration of subjective search spaces, along with collaboration to allow designers to expand their search spaces through case injection.

### B. Interactive Genetic Algorithms

The design process consists of an exploration through the design solution space. Evolutionary computation techniques, specifically genetic algorithms (GAs) have been used and proven in various problem domains to be effective search techniques [12]. We are trying to provide computer support for the conceptual design phase, which tends to be one of the most creative phases in the design process. Thus, we face the challenge of providing support for subjective evaluation of alternative conceptual designs. When there is no better fitness measure to a solution other than the one in the human mind, then we use interactive genetic algorithms (IGAs) [22]. IGAs

replace the fitness evaluation of individuals in the population with user evaluation. Due to the nature of IGAs, they have been used for a variety of applications which incorporate creative human input, including editorial design, industrial design, image processing, database retrieval, graphic art and computer graphics animation, control and robotics, among others [22]. In our work, the use of IGAs during conceptual design allows designers to evaluate subjective criteria and to incorporate aesthetic preference into the explorative evolutionary process.

### C. Computer Supported Collaborative Work

Computer supported collaborative work (CSCW) addresses the support and enhancement of collaboration of a group of people mediated through computers. Some examples of current research on CSCW are wikis, chatting, instant messaging, e-mail, VOIP, videoconferencing, shared desktop applications, and control version systems [7], [20]. CSCW research also has concentrated on blocks to communication and creativity that arise from conflicts between individuals [18].

The computational model of creative design based on collaborative interactive genetic algorithms supports collaboration at two levels: (1) designers can view each others’ solutions; and (2) designers can choose to inject design solutions from their peers into their IGAs. The interaction between the designer and the IGA, while allowing for exploration, results in a convergence from a random sample of the solution search space to design solutions which reflect the designer’s preference. Collaboration has the benefit of allowing each of the designers to be exposed to the diverse paths taken by each of the designer’s peers. Even if the designer chooses not to inject any solutions from peers, the designer can still view the solutions belonging to peers, thus allowing the designer to compare and contrast his/her respective solutions and reflect on design alternatives and on the reasoning behind the design solutions created by peers. With injection, designers can bias their evolutionary exploration, by allowing combination to take place between the designer’s own and the injected solutions. More importantly, whenever a designer chooses to inject a solution into his/her population, the resulting bias will eventually be viewed by the designer’s peers, resulting in a chain reaction.

Our work most closely resembles Picbreeder, a system used to evolve pictures collaboratively online [21]. Picbreeder supports collaboration by allowing users to branch from images created by other users. Users can start evolution from randomly generated images, or from an image generated by someone else in the Picbreeder community. Our work differs from Picbreeder in that we support collaboration in real time. Users of our system participate in individual evolutionary sessions concurrently, with users able to see each others’ designs as evolution progresses. Picbreeder is a special type of case injection, where users select the individual from which to start evolution, referred to in their research as “branching” [16]. In our model users work in a group setting, with users able to inject at any time any individual they find interesting from any of their peers.

### III. COMPUTATIONAL MODEL OF CREATIVE DESIGN

The computational model of creative design based on collaborative IGAs is shown in figure 1. The figure illustrates three users collaborating with each other, with each of the peers denoted by the dotted boxes. Each user interacts with a GA by acting as the subjective evaluation. As shown, the evaluation is not purely subjective, instead the evaluation of design solutions consists of the multi-objective optimization of the subjective and objective criteria. We use Pareto optimality to maximize these criteria [6]. The arrows between the GAs of each of the peers represent the communication that takes place between the peers. If a user likes a design solution from one of his/her peers, then the user has the option to inject that solution into his/her population, thus introducing a search bias.

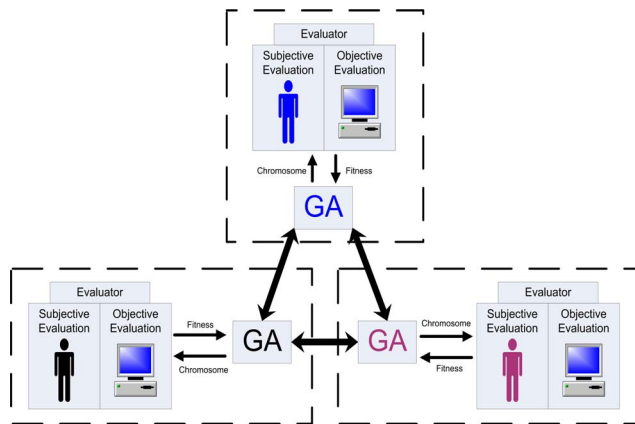


Fig. 1. Computational Model of Creative Design

#### A. Design Space Exploration

To determine whether the model has the potential to produce creative content we have to look at design space exploration. According to Geros definition of creativity, creativity has the potential to occur when a designer purposely shifts the focus of the search space [11]. We illustrate this on the left part of figure 2. Specifically, the ability to “perform goal-oriented shifts of the focus of the search activity” is crucial to a design process, done either implicitly by the designer’s understanding of the problem changing over time, or explicitly through adding variables to the search space. The right part of figure 2 shows a similar phenomenon, but how it occurs in our computational model. The white blobs show the starting search spaces of the individual designers. Through collaboration the designers are able to shift focus of their initial search spaces as a consensus, ending in the state space  $S_N$ . The implementation presented in this paper does not include the explicit addition of variables to expand the search space.

#### B. IGAP: Interactive Genetic Algorithm Peer-to-Peer

IGAP is the framework we implemented to test the computational model of creative design. Figure 3 shows the steps involved in IGAP. Each peer acts as an independent node,

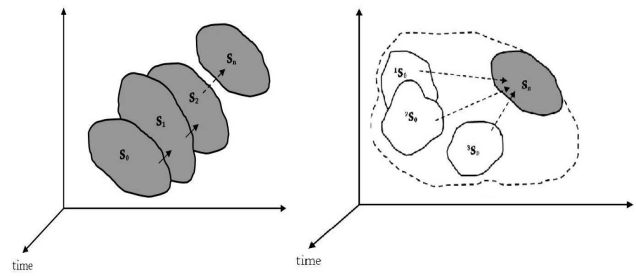


Fig. 2. Design Space Exploration

running as a server which handles incoming requests from peers. On a request, the peer node sends a subset of its best genomes to the requesting peer. The requesting peer, assuming collaboration with more than one peer, constructs a genome pool from all the genomes received from all of its peers. From the genome pool the requesting peer node then selects a random subset to display on the screen of the designer.

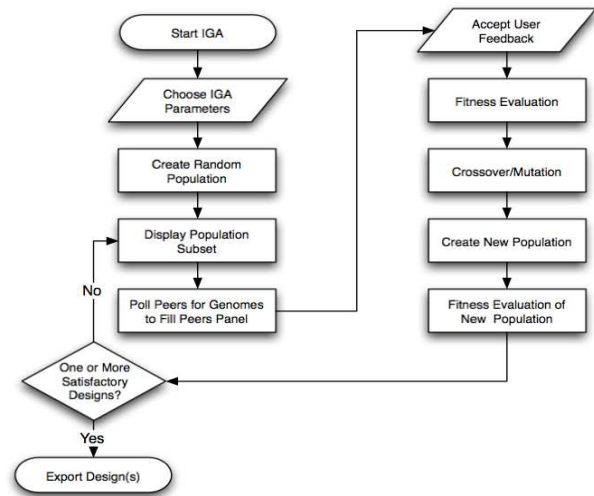


Fig. 3. IGAP Process

#### C. Multi-objective Optimization

We use the non-dominated sorted multi-objective genetic algorithm (NSGA-II) to evolve floorplans [6]. The NSGA-II creates fronts of non-dominated individuals, where within a front none of the individuals are any worse than any other individual across all optimization criteria and all individuals within a front are said to have the same rank. We select parents by using the crowded distance tournament operator. We pick two individuals to participate in the tournament, and we select the individual with the higher rank to be part of the mating pool. In case the two individuals have the same rank, and consequently belong to the same front, then the crowded distance of both individuals is computed, and we select the individual with the highest crowded distance to be part of the mating pool. This translates to the individual being in

a less crowded region of the front and hence, the crowded distance selection favors the most diverse individuals within a front.

We use NSGA-II with a two-criterion multi-objective function: objective fitness and subjective fitness. Previously, we had used a five-criterion multi-objective function, where two of the five criteria measured objective guidelines, while the remaining three criteria measured subjective preferences. However, studies have shown that the performance of the NSGA-II degrades when using more than three criteria during multi-objective optimization [23].

#### D. A Special Case of Case-Injected Genetic Algorithms

A case-injected genetic algorithm (CIGAR) works differently than a typical GA. A GA randomly initializes its starting population so that it can proceed from an unbiased sample of the search space. The methodology behind CIGARs is that it makes less sense to start a problem solving search attempt from scratch when previous search attempts (on similar problems) may have yielded useful information about the search space [16]. Instead, periodically injecting a GA's population with relevant solutions or partial solutions to similar previously solved problems can provide information (a search bias) that reduces the time taken to find a quality solution. This approach borrows ideas from case-based reasoning (CBR) in which old problem and solution information, stored as cases in a case-base, helps solve a new problem [15]. The collaborative IGA computational model is a special case of CIGARs, where the designer during the interactive evolutionary session determines when and how many individuals to inject into the population, instead of being done in an algorithmic fashion [16]. Furthermore, in CIGARs the side effect occurs in one direction, with the individuals injected from the case base affecting the performance of the running GA. In our model, when a designer chooses to inject a solution from one of his/her peers, the introduced bias will not only become apparent in the designer's own population, but the other peers will also be able to view this change as well, since designers can always see a subset of each others' designs.

#### E. Collaborative Methodology

Collaborative evolution is implemented with a peer to peer network. We treat each user participating in evolution as a node, handling incoming requests from other nodes (peers) and requesting information from peers. By using a peer to peer network, control is decentralized and each node is free to choose who to connect to and if necessary who to exclude from its set of peers.

The interface during an individual evolutionary session is shown in figure 4. During collaborative evolution (figure 5), a subset of peer-evolved designs is displayed to the right of the user's population. We limit the number of peer individuals to nine, organized in a 3x3 grid, similar to how we present the user's own population, in order to be consistent. For more than one peer, we cannot display all the individuals belonging to the subset of each peer, since we only display

nine. We do make sure that the user selected best individuals from each peer are displayed on the peers subset. We save the user selected best from generation to generation, and we always make it part of the subset displayed the next time the IGA requires user input. We select the rest of the individuals that make up the peers subset by taking random individuals from a collective pool consisting of all individuals taken from peers. By selecting a random subset, we believe that over many generations, all of the participants will get approximately the same amount of their designs displayed on the screens of collaborators.

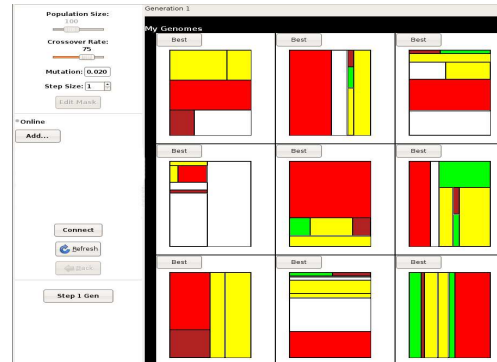


Fig. 4. Screen shot of individual floorplan designing.

The benefit of viewing the best individuals from peers is limited, unless the user is able to take promising individuals from peers and mold them to their liking. We support this by allowing the user to inject individuals from the subset of peers into the user's own population. The user can also select a best individual from the subset of individuals from peers, in which case the user selected best is automatically injected into the population, and used for fitness interpolation. We require the user to select a best individual, but it does not have to be from the user's own population - the user selected best can come from peers.

The injected individuals replace the bottom 10% of the population as done in [16]. If the number of injected individuals is less than 10% of the population, then we insert numerous copies of the injected individuals, until the total sum of the injected individuals is 10%. In CIGARs typically a case base is kept of solutions to previously solved problems, and based on problem similarity, individuals similar to the best individuals in the current population are periodically injected, replacing the worst individuals [16]. In our algorithm, the designer plays the role of determining how many, when, and which individuals to inject at any step during the collaborative evolutionary process. If the injected individuals make a positive contribution to the overall population, then they will continue to reproduce and live on, while injected individuals which do not improve the population performance will eventually die off. Hence, the user is not penalized for injecting subpar individuals.



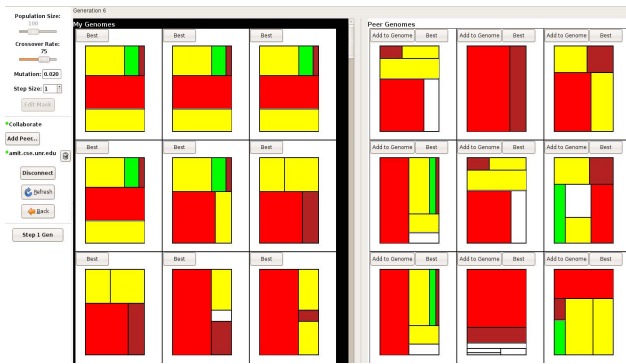


Fig. 5. Screen shot of collaborative floorplan designing.

### F. Fitness Biasing

We use fitness biasing to ensure that injected individuals survive long enough to leave a mark on the host population. We use the concept of bloodline to do fitness biasing. Injected individuals are considered to be full blood, while those individuals already in the population are treated as individuals with no blood. The bloodline consists of a number between 0 (no blood) and 1 (full blood). When a full-blooded individual crosses over with a no-blooded individual, then the offspring will inherit a bloodline value equal to a weighted sum of the bloodline of the parents, where the weight values depend on the percentage of the genetic material inherited from each parent. This is shown in equation 1, where  $p_1$  is the percent of genetic material inherited from the first parent,  $p_{1_{blood}}$  is the bloodline value of the first parent,  $p_2$  is the percent of genetic material inherited from the second parent ( $p_2 = 1 - p_1$ ), and  $p_{2_{blood}}$  is the bloodline value of the second parent. The product of an individual's fitness value and bloodline (or 1 minus the bloodline if minimizing fitness values) is the bias added to the individual's fitness.

$$child\ blood = p_1 * p_{1_{blood}} + p_2 * p_{2_{blood}} \quad (1)$$

## IV. CASE STUDY: FLOORPLANNING

We use floorplanning as the case study to test the model of creative design. The rooms in the floorplans are color coded as red (living area), yellow (bedrooms), green (eating areas - kitchen and/or dining rooms), firebrick (bathrooms), and white (empty spaces).

### A. Floorplan Representation

For evolving floorplans we have used a binary tree representation, coded as a nested list. At every node of the tree, the parameters specify how the rectangular panel at that level is subdivided (either left/right or top/bottom) and the percentage of panel area at that level contained in either the left or the top subdivision. Figure 6 shows how the rectangular panel is subdivided into rooms and spaces. A room is represented by the array  $[0, 1]$  and a space by  $[0, 0]$ . An arbitrary array  $[0, 0.75]$  represents division in top/bottom configuration with top sub-panel containing 75%

of the parent panel. Another list  $[1, 0.80]$  represents division in left/right configuration with left sub-panel containing 80% of the parent panel. Even though the representation is quite intuitive, it only allows us to represent rectangular shapes. To represent more complicated (possibly organic) shapes, a more complex representation is needed. The plans are decoded according to the guidelines in [17] - depending on the number of rooms (the number of  $[0, 1]$  arrays in the encoding) and their relative sizes, the guidelines have explicit instructions pertaining to room labels (living, bed, kitchen etc.). For example if a particular plan has two rooms, the bigger room is labeled as the Living-Bed-Kitchen (studio configuration) and the smaller room is labeled as the Restroom. For plans with more than three rooms or more, the bedrooms are separated from the living room.

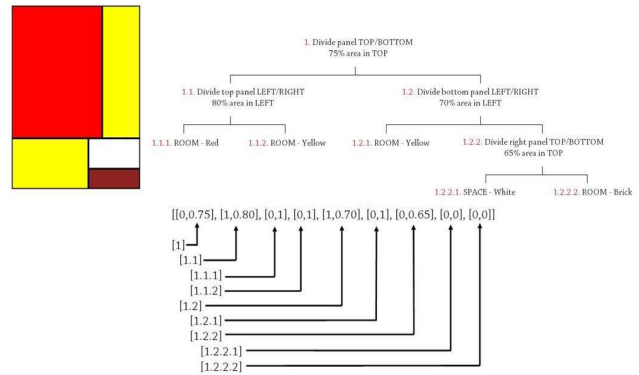


Fig. 6. The binary tree representation of a floorplan.

The binary tree representation for floorplans necessitates the need for a specialized tree-crossover operator. The nested list is parsed as a binary tree and two such parent trees are crossed at randomly chosen nodes, such that entire sub-trees following those nodes are swapped. The tree representation is used in genetic programming [14] and hence, our crossover operator maps to the crossover operator used in genetic programming. Depending on the probability of mutation, the mutation operator works on the two parameters of the nodes (or leaves) differently. It performs a binary swap on the first parameter thereby changing the subdivision configuration. Depending on the value of the second parameter, the operator either performs a binary swap (if the value is either 0 or 1), thereby changing a room to a space and vice versa, or if the second parameter is a real number between 0 and 1, the operator replaces it by another random real number in the same interval, thereby altering the dimensions of the room (or the space).

### B. Fitness Evaluation

1) *Objective Evaluation*: Plans are compared room-wise to ascertain if they meet the minimum dimension and area criteria in [17]. The guidelines for a two-room single-storey house plan call for the bigger room to be at least  $300ft^2$  and have a minimum dimension of  $20'10"$ . The number of rules that a particular plan needs to adhere to increases

with the number of rooms. The objective measure assigned to a two-room plan that satisfies the minimum area and minimum dimension requirement is  $[0.0, 0.0]$ . If the area of the bigger room of a certain other plan is  $area$  such that  $area < 300ft^2$ , and if the minimum dimension of the bigger room is  $minlen$  such that  $minlen < 20'10''$ , then the objective minimization measure is given by equation 2.

$$obj = \left[ \frac{300 - area}{300}, \frac{20'10'' - minlen}{20'10''} \right] \quad (2)$$

2) *Subjective Evaluation*: We also compare the plans to the user-selected best on three criteria. The three criteria are: (1) number of rooms, (2) room adjacencies, and (3) total built area of the plan. The first criteria compares the number of rooms in the user-selected best plan to the particular plan in question. If the user-selected best plan has  $b$  rooms, a plan with  $c$  number of rooms is assigned a measure of  $|b - c|$ . If  $1 < c < 8$ , then it is assigned a measure of:

$$1 + \max(b - 2, 8 - b) \quad (3)$$

In order to compare plans for similarity (or dissimilarity) in room adjacencies, we compare adjacency similarity in certain pairs of rooms, such as living-bedroom adjacency, restroom-bedroom adjacency, kitchen-dining area adjacency, and restroom-kitchen adjacency. This information is stored in a four-bit string (1 for adjacency and 0 for no-adjacency). The Euclidean distance between the bit string for the user-selected best plan and the bit string for the particular plan provides the second subjective measure. For plans that do not have separate kitchens or dining areas or bedrooms, the default adjacency measure is always unity. The third criteria measures the similarity between the plan in question and user-selected best plan in terms of total built area, i.e. area occupied by all rooms. If the user-selected best plan has a total built area of  $barea$ , and the plan in question has a total built area of  $carea$ , then the penalty associated with the third criteria is given by,  $(barea - carea)/barea$  if  $carea < barea$ . This is also treated as a minimum subjective requirement, hence there is no default penalty (plans with more built area than the user-selected best get a penalty measure of 0.0). These three measures, added together together and normalized, constitute the subjective penalty function of NSGA-II.

## V. EXPERIMENTAL SETUP

We had 20 participants in our study, eight females and 12 males. Out of the 20 participants, 11 were from engineering and math, two were undeclared, and seven were from social sciences. Participants were assigned to groups of four based on schedule availability. We picked groups of four, so that using a 3x3 display grid, allowed for three floorplans from each peer to be displayed on the screen of every other participant.

Participants first were allowed to get familiar with the IGA. They were instructed in how to guide the process, both

individually and collaboratively. The participants were told the set of requirements which they would have to follow after the tutorial. This was done so that participants could develop an intuition and a sense for how the system worked. The participants were not told that they were using an evolutionary system, they were simply told that after selecting the best floorplan, the screen would refresh, displaying a new set of floorplans would be similar to what they previously selected as the best.

The set of floorplan requirements given to the participants were: (1) Create a floorplan for a 2 bedroom, 1 bathroom apartment, (2) the bathrooms should be close to the bedrooms, and (3) the bathrooms should be far from the kitchen and dining room areas. During the tutorial phase participants were given the requirements, so that they could practice guiding the IGA to floorplans that meet the given requirements. We also meant for the tutorial to remove any bias with regards to unfamiliarity with the system and with IGAs. Participants were allowed to run for as many generations as they wished. Once the participants had found a floorplan that met all requirements and that they also liked, then they would make a final selection of the best floorplan, and quit the program.

After the tutorial session, the participants were instructed to create a floorplan individually that met all of the requirements. Following this, the participants created a floorplan with collaboration. During the collaborative run, the participants were allowed to inject as many designs from their peers as they wished, but their final floorplan selection had to come from their own collection of floorplans.

In each group, every participant picked one final floorplan from the individual run as the best and one final floorplan from the collaborative run as the best. Each participant in the group then graded the two best floorplans selected from each of his/her peers, so that each participant evaluated six floorplans - three collaborative and three individual floorplans. The floorplans were evaluated by the participants using the following criteria: (1) appealing - unappealing, (2) average - revolutionary, (3) commonplace - original, (4) conventional - unconventional, (5) dull - exciting, (6) fresh - routine, (7) novel - predictable, (8) unique - ordinary (9) usual - unusual, and (10) meets all requirements - does not meet requirements.

Each of these criteria was scored using a seven-point Likert scale. The criteria were derived from a subset of the Creative Product Semantic Scale [4], [1]. The presentation of the criteria was randomized for each of the floorplans, to make sure the participants were alert and to make sure they read the criteria before providing a score. The participants were given as much time as necessary to complete the evaluation.

## VI. RESULTS

Assuming the exploration of a large design solution space, we ask whether collaboration amongst peers is sufficient to allow for the potential to produce creative content in designing without explicitly expanding the design solution space by adding one or more variables. Our hypothesis was that

collaboration would be sufficient to produce creative content, and that designs evolved collaboratively would consistently rank higher in the evaluation criteria than those created individually.

The compiled evaluation results for all groups are shown in table I. The table shows all evaluation criteria in the first column. The second column, “Desired Value”, specifies the desired range in the seven-point Likert scale that would support our hypothesis. For the first criterion, “Appealing - Unappealing”, a value of 1 would represent that a given floorplan was “appealing” while a value of 7 would represent that a given floorplan was “unappealing”. For the second criterion, “Average - Revolutionary”, a value of 1 would represent that a given floorplan was “average” while a value of 7 would represent that a given floorplan was “revolutionary”, and so on.

The third and fourth columns in table I show individual and collaborative averages. The individual average provides the average score received by each floorplan evolved individually by each of the 20 participants. The collaborative average provides the average score received by each floorplan evolved collaboratively by each of the 20 participants. The fifth and sixth columns show the corresponding standard deviations. The last column shows the corresponding p-value for each criterion.

By looking at the p-value, we can say that the floorplans created collaboratively ranked slightly higher in the “average-revolutionary” and the “commonplace-original” criteria. For the other criteria, the differences in the averages are not statistically significant. However, even though the floorplans created collaboratively were considered to be more “revolutionary” and “original” than those created individually, the average scores are near the median of the seven-point Likert. Ideally, we would have liked to see these values farther apart, with the values for “revolutionary” and “original” being closer to 7 rather than to 4. Finally, we can see that the participants, either individually or with collaboration, were able to effectively bias the floorplan designs subjectively to designs which met most of the requirements, as is shown by the low average scores obtained in the requirements criterion.

From the results obtained we can deduce that perhaps collaboration was not sufficient to make a clear distinction in the creative value between the individual and collaborative floorplans. There are some issues that came up during evaluation that can shed some light on these results. The first is that the participants were not told explicitly which of the contrasting adjectives in the evaluation criteria were positive and which were negative. For example, some participants expressed that while some of the floorplans were “unappealing”, because they would not have liked to live in such an apartment, they found the floorplan “appealing” because of its innovative, and at times, bizarre room layouts. Hence, there was some ambiguity in how to evaluate the resulting floorplan designs. Another issue is the applicability of some of the criteria to floorplan design. For a student, a floorplan for an apartment might not be something that

would instill a feeling of “exciting”, even if the floorplan had a creative layout. A domain expert, such as an architect, might have a more refined appreciation of the quality of the designs, which might have yielded radically different results. Finally, the participants had one try at creating a floorplan individually and one try at creating a floorplan collaboratively. Asking the participants to evolve more than one floorplan individually and collaboratively might have also yielded a more significant difference in the results.

From our observations and feedback from the participants, we found that (while not explicitly shown by the numbers) the participants found evolution of the floorplan designs during the collaborative session to be easier. During individual evolution, the floorplans tended to converge, as expected, to floorplans which were high fitness but which differed slightly in terms of room dimensions and room layout. On the other hand, during collaborative evolution, the participants were exposed to diverse high fitness individuals which belonged to the peers. Some of the participants also used the ability to inject numerous design solutions from peers as a mechanism to manage diversity in their own populations. We gave the participants no limit to the number of generations (number of picks) before they had to pick their final floorplan selection. While the system was designed to support exploration, many participants looked through the entire population of floorplans to find a design which met the requirements, and stopped after one to two generations, instead of taking advantage of the evolutionary process in order to try and breed new interesting floorplan designs. Other participants restricted themselves to picking from the subset of the best nine floorplans, and picked as many as 10 generations, until they were certain that their population had converged. We found that many participants concentrated in finding a floorplan that met all requirements and subsequently stopped, even if the floorplan looked uninhabitable. We believe this might be due to the system’s lack of affordance, which might have encouraged users to continue exploring, and to lack of motivation, since participants had no real incentive to continue exploring and find a better looking floorplan. In average all participants evolved the floorplan designs in less than 10 minutes.

## VII. FUTURE WORK

The current case study was chosen in part because it had a convenient digital representation, thus it was well suited for our initial tests. However, we intend to apply our computational model to other case studies, which will require different and more complex representations, in order to further test the validity of the model.

In the work presented, we had the group members evaluate each other’s designs. We are interested in further evaluation of the resulting designs by domain experts. In the case of floorplanning, a group of architects could evaluate the resulting designs.

Finally, we will implement the addition of design variables to the computational model of design, so that the system conforms to the definition of creative design followed by

TABLE I  
INDIVIDUAL VERSUS COLLABORATIVE SCORING RESULTS

Evaluation Criterion	Desired Value	Ind. Avg.	Coll. Avg.	Ind. $\sigma$	Coll. $\sigma$	P-value
Appealing - Unappealing	Low	4.08	4.39	1.29	1.29	0.458
Average - Revolutionary	High	3.76	4.34	0.92	0.91	0.052
Commonplace - Original	High	3.97	4.68	1.02	1.01	0.034
Conventional - Unconventional	High	4.03	4.41	1.11	1.41	0.353
Dull - Exciting	High	3.65	3.93	0.76	0.72	0.234
Fresh - Routine	Low	3.82	3.68	0.96	1.14	0.690
Novel - Predictable	Low	3.55	3.40	0.88	0.94	0.604
Unique - Ordinary	Low	3.49	3.11	1.12	1.10	0.280
Usual - Unusual	High	4.21	4.51	1.14	1.28	0.439
Meets All Req. - Does Not Meet Req.	Low	2.63	2.83	1.37	1.54	0.666

computational models described in the literature [10]. This can be implemented in IGAP by allowing individual designers to start their corresponding evolutionary processes with a subset of all design variables being evolved, with the rest fixed. For example, assuming two individuals are collaborating, and four variables are being evolved for design exploration, then each of the users could start having two variables fixed and two being evolved. In this scenario, case injection through collaboration would expand the initial search space, from exploring designs with two evolving variables to exploring the space of designs with four evolving variables.

### VIII. CONCLUSION

We have presented a computational model of creative design based on collaborative interactive genetic algorithms. We showed the potential of the model to produce creative content by analyzing design space exploration. Our results showed that those floorplans created collaboratively scored slightly higher in terms of “revolutionary” and “unconventional” criteria than floorplans created individually. We expected collaboration in our computational model to provide enough potential to produce creative designs. However, from our results we can conclude that in the majority of the criteria, floorplans created individually scored similarly to those created collaboratively. Thus, there is a need to combine collaboration in our model with an explicit expansion of the design solution space by adding one or more variables.

### IX. ACKNOWLEDGMENTS

We thank the study participants for their time. This work was supported in part by contract number N00014-0301-0104 from the Office of Naval Research and the National Science Foundation under Grant no. 0447416.

### REFERENCES

[1] J. Andrews and D. C. Smith. In search of the marketing imagination: Factors affecting the creativity of marketing programs for mature products. *Journal of Marketing Research*, 33:174–187, May 1996.

[2] A. Banerjee, J. C. Quiroz, and S. J. Louis. *A Model of Creative Design Using Collaborative Interactive Genetic Algorithms*, pages 397–416. Springer, 2008.

[3] P. J. Bentley and J. P. Wakefield. Conceptual evolutionary design by genetic algorithms. *Engineering Design and Automation Journal*, 3:119–131, 1997.

[4] S. Besemer and K. O’Quin. Analyzing creative products: Refinement and test of a judging instrument. *Journal of Creative Behavior*, 20:115–126, 1986.

[5] M. Csikszentmihalyi. *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Perennial, 4 tra edition, June 1997.

[6] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[7] U. Farooq, J. M. Carroll, and C. H. Ganoe. Supporting creativity with awareness in distributed collaboration. pages 31–40, Sanibel Island, Florida, USA, 2007. ACM.

[8] G. Fischer. Distances and diversity: sources for social creativity. pages 128–136, London, United Kingdom, 2005. ACM.

[9] J. S. Gero. Computational models of innovative and creative design processes. *Technological Forecasting and Social Change*, 64:183–196, June 2000.

[10] J. S. Gero. Computational models of creative designing based on situated cognition. pages 3–10, Loughborough, UK, 2002. ACM.

[11] J. S. Gero and T. Schnier. Evolving representations of design cases and their use in creative design. *Preprints Computational Models of Creative Design, Key Centre of Design Computing, University of Sydney*, pages 343–368, 1995.

[12] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.

[13] D. E. Goldberg. *The Design of Innovation*. Springer, 1 edition, June 2002.

[14] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Dec. 1992.

[15] D. B. Leake. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press, Aug. 1996.

[16] S. Louis and C. Miles. Playing to learn: case-injected genetic algorithms for learning to play computer games. *Evolutionary Computation, IEEE Transactions on*, 9:669–681, 2005.

[17] E. Neufert, P. Neufert, B. Baiche, and N. Walliman. *Architects’ Data*. Wiley-Blackwell, 3 edition, Aug. 2002.

[18] R. Ocker. Influences on creativity in asynchronous virtual teams: a qualitative analysis of experimental teams. *Professional Communication, IEEE Transactions on*, 48:22–39, 2005.

[19] G. Renner. Genetic algorithms in cad. *Computer-Aided Design*, 35:707–708, July 2003.

[20] J. W. Sarmiento and G. Stahl. Group creativity in virtual math teams: interactional mechanisms for referencing, remembering and bridging. pages 37–44, Washington, DC, USA, 2007. ACM.

[21] J. Secretan and N. Beato. Picbreeder: evolving pictures collaboratively online. pages 1759–1768, Florence, Italy, 2008. ACM.

[22] H. Takagi. Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89:1275–1296, 2001.

[23] S. Tiwari, P. Koch, G. Fadel, and K. Deb. Amga: an archive-based micro genetic algorithm for multi-objective optimization. pages 729–736, Atlanta, GA, USA, 2008. ACM.