DOI: http://dx.doi.org/10.7551/978-0-262-32621-6-ch023

Collective Design of Robot Locomotion

Mark Wagy and Josh Bongard

University of Vermont, Burlington, VT 05401
mwagy@uvm.edu

Abstract

It has been shown that the collective action of non-experts can compete favorably with an individual expert or an optimization method on a given problem. However, the best method for organizing collective problem solving is still an open question. Using the domain of robotics, we examine whether cooperative search for design strategies is superior to individual search. We use a web-based robot simulation to determine whether groups of human users can leverage design intuition from others to focus search on relevant parts of a complex design space. We show that individuals that work cooperatively with the aid of a simple optimization algorithm are better able to improve the design of robot locomotion than if they were to work individually with the aid of the optimization algorithm. This result suggests that groups of designers may more effectively work in tandem with optimization algorithms than individuals working in isolation.

Introduction

There is a long history of humans interacting with computer systems to design software agents. In one of the most famous examples of interactive design of software agents, human users interacted with Dawkins' *Blind Watchmaker* to generate "biomorphs": increasingly lifelike beings that resulted from a process of interactive selection (Dawkins (1996)). However, there is little basis for groups of humans interacting to design software agents collaboratively.

We know that positive feedback within a collective can spread through the group and result in wider perceptual abilities than that of the individual (Couzin (2007)). These types of positive feedback or autocatalytic phenomena may result in behavior that outperforms what any single agent in a population could achieve. Anecdotally, we know that teamwork in an organization can result in creative approaches to problem solving and that the intelligence of a group cannot be predicted by the average intelligence of its individual members (Woolley et al. (2010)). So might a group working with an optimization algorithm collectively arrive at better performing designs than an individual would on their own? With this question in mind, we use a web-based platform for robot and behavior optimization to determine whether groups of human users can leverage design intuition from others to focus search on promising regions of a complex design space.

In order for users to be able to easily communicate designs, we have adoped a means of communicating through a language of graphical symbols, which in some sense acts as a "visual programming language". Visual programming structures can result in better "closeness of mapping" than textual representations (Green and Petre (1996)). That is, visual languages map programming abstractions more directly to the domain that they are modeling. We exploited the intrinsic pattern recognition capabilities of humans in the experiments reported here by developing a visual language for robot configuration. Users 'wire' subsets of a robot's joints together using colored lines, which constrains those subsets to move in phase with one another. An underlying evolutionary optimization algorithm then optimizes the movements of the joints to produce locomotion. Subsequent visitors to the site can then ascertain promising areas of the search space by visually inspecting the wiring patterns created by others and contribute their own computer's time to those regions, or create new designs of their own.

Three main literatures informed the design of our experiment: interactive evolutionary algorithms, visual design languages, and collaborative problem solving.

Interactive Evolutionary Algorithms

A group of non-experts, even children, have been found capable of training robots to achieve complex movement. Lund et al. (1998) employed child subjects in a programming-free, interactive evolutionary robotics experiment to train artificial neural networks to produce interesting robot behaviors such as obstacle avoidance and line and wall following. Chernova et al. (2011) used multiplayer online games featuring simulated robot agents to train a Case-Based Reasoning system. The robots then interacted with humans in a physical environment similar to the web-based simulation in which they were trained. However, the focus of most crowdsourcing robotics studies has been on using individual human users to train robots.

Similarly, in studies involving interactive evolutionary robotics, problem solving primarily involved a single individual dedicated to each training instance (Lund and Miglino (1998), Lund et al. (1998), Dozier (2001)). There was a one-to-one relationship between the human interacting with a robot control algorithm, rather than a group of individuals collectively working toward developing better robot control. Typically the individual directly evaluated fitness of computer-generated designs. Takagi (1998) gives a broad overview of how interactive evolutionary computation has been used for engineering and creative domains, from the design of hearing aids to evaluation of 3D CG images. The designs tended to be evaluated by a single human interacting individually with an evolutionary computing process. Celis et al. (2013) demonstrated the use of interactive evolutionary algorithms in robot control to overcome local optima through demonstration. Also, Bongard and Hornby (2013) used human users to train a user model to avoid local optima in an interactive evolutionary algorithm for the control of robot movement. That work built on the user modeling approach to interactive evolutionary robotics pioneered in (Akrour et al. (2011)). Recent work on exploring the possibility of tapping into user creativity using interactive evolutionary algorithms can be seen in Kowaliw et al. (2012) and García-Valdez et al. (2013).

Visual Design Languages

A substantial challenge in collaborative design is effectively communicating complex concepts between members of a group. In order to allow members of a group to build upon prior knowledge and thus improve past designs, key concepts in the problem domain must be communicated through some form of *working memory*. Larkin (1989) suggested that the use of a diagram is a means for storing problem states in a working memory. With this idea in mind, we developed a simple visual language of symbols for robot joint configuration. Users working together use these symbols as a blackboard system (Nii (1986)) to "exchange" ideas and improve the robot design.

Simplification of robot control by defining domain-specific languages is common in the robotics literature. For example, Peterson et al. (1999) define a declarative language for controlling robot behavior. Their framework is also focused on defining robust robot control rather than performing a single task such as fast locomotion. Cox and Smedley (1998), Cox et al. (1998) and Pfeiffer Jr (1998) all successfully employed visual programming languages for robot control. However, in the present work we are using symbols as a means of indirectly constraining the evolution of high-level robot behavior, rather than programming specific responses of the robot to its environment. It is not the intent of our study to be able to train a robot to exhibit complex behavior with our symbolic cues. Rather, the symbols that represent different classes of robot movement act as a means by which to communicate possible unexplored parts of the evolutionary search space to others. Though in both cases, the importance of the direct mapping of visual indicators to the movement of robots is noted as an important benefit of visual over textual representations of a complex problem such as robot control.

Collaborative Problem Solving

Much of crowdsourcing work is focused on taking a large problem and dividing it into independent "microtasks" to be implemented by independent web-based workers. However, one of the most visible platforms for crowdsourcing is a collaborative one. The crowdsourced online encyclopedia Wikipedia [1] is one of the best examples of the success of collaborative human computing. Collaborative, networked design tasks have also shown promise in past literature. Kan et al. (2001) deployed a virtual reality framework in which the users of the system directly work with each other on product design. This contrasts with the present work in which individuals in the experimental group only indirectly collaborate with each other through the sharing of past designs asynchronously, but without the intent to directly work with each other. Users implicitly work towards a common goal, without directly working together (Doan et al. (2011)), similar to the approach described in (Von Ahn and Dabbish (2004)). Collaborative crowdsourcing of design tasks with an evolutionary computing backbone has also been explored in past work. Yu and Nickerson (2011) crowdsourced selection and recombination in an evolutionary process of designing alarm clocks. They shared drawings of alarm clocks among Mechanical Turk workers to iteratively vet more creative and practical designs through a process of drawing combination. They show a significant improvement in the practicality and creativity of designs at the third generation versus the initial designs. The *Picbreeder* system developed by Secretan et al. (2011) employed a web-based collaborative system to create realistic images using the NEAT algorithm (Stanley and Miikkulainen (2004)). In this paper we investigate whether such collaborative dynamics can be brought to bear on robot design.

Methodology

We deployed a web-based platform (see Figure 1 for a screenshot) in which users "design" different robots, and then an evolutionary optimization algorithm evolves behaviors for them. Users were partitioned into control and

[1] www.wikipedia.org

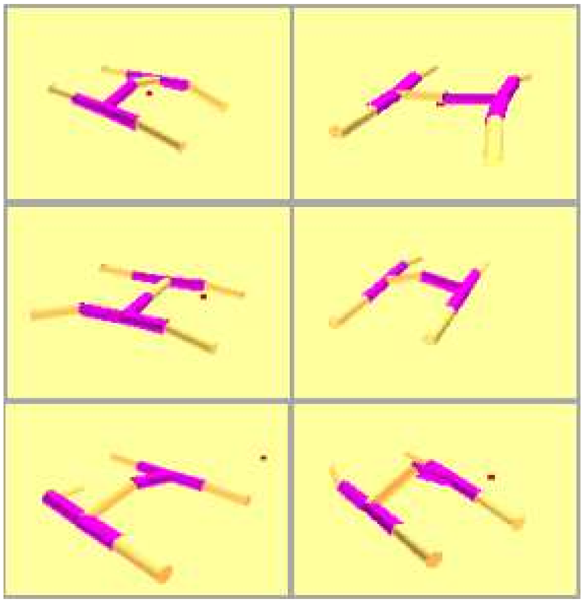

Figure 2: Screenshots the of robot.

experimental groups: those in the control group could not see robots designed by others, while those in the experimental group could. Each user's objective was to create robots that were most amenable to behavior optimization by the evolutionary algorithm. The specific behavior we investigated was legged locomotion.

Robot Form

The robot's form was fixed but users could constrain different subsets of joints to move in phase with one another. This process in essence reduces the dimensionality of the search space for the evolutionary algorithm, as it only has to optimize movements for each joint subset, rather than all joints independently. We henceforth refer to each such user-generated set of joint constraints as a robot design.

The robot's form consisted of ten cylindrical segments, connected with nine hinge joints. The robot had four legs, each with both a "shoulder" joint and an "elbow" joint. Each of the four legs were connected to a "spine" (see Figure 2 for screenshots of robot). The hinge joints could sweep through a range of $[-45°, 45°]$. The "shoulder" joints rotated the legs through the transverse plane. The "elbow" and "spinal" joints rotated through the sagittal plane.

The robot did not contain any sensors: the evolutionary algorithm produced open-loop controllers for the robots. The robot joints were actuated using a displacement-control sinusoidal signal. This sinusoidal actuation resulted in several possible gaits, depending on the phase value of the sinusoid that drove each joint. The evolutionary algorithm described below could alter the relative phase between joints, but not the frequency or amplitude. The frequency was set such that the robot typically exhibited 40 rotations of each joint per evaluation. The amplitude allowed each joint to reach just about to its maximum and minimum range. However, this range could be constrained by the momentum of the robot and its friction with the ground plane.

User Interaction

Using a graphical interface representing a top-view of the robot form, users drew lines that linked hinge joints at the robot legs and spine into a group. Each grouping of joints was assigned a distinct color, based on the user-drawn lines that visually linked the nodes representing leg or spine joints (e.g. see *Design Panel* in lower right corner of Figure 1). When users drew links between joints, they were forcing those joints to have the same phase offset of the sinusoidal control as all others in the joint group - i.e. the phase value in the sinusoidal displacement control signal was the same for all joints in a grouping. This visually-oriented, symbolic display of joint connections allowed for easy communication of intuition about movement of the robot. For example, a user might notice that connecting diagonal joints results in trotting. They may realize that connecting the joints of the front legs together, and connecting the joints of the back legs together produces a two-beat version of cantering. The lines connecting joint group configurations acted as a symbol language that was designed to support the inherent pattern recognition abilities of the humans interacting with the underlying machine learning algorithm.

The robot movement was simulated with the *JBullet*[2] physics simulation engine, a Java port of the *Bullet Physics Library*[3]. The user could design their joint grouping in an interactive drawing panel and then press the *Run* button to see a "heads up" or graphical display of robot movement simulated in the *simulation panel*.

The time allowed for each simulation was fixed at 20 seconds. Since there was some level of non-determinism in the simulation, we added *background runs* to increase the sample size for each of the grouping/phase configurations.

Each time a user clicked the "GO" button to launch a "heads up" run (an actual visual simulation of the robot movement), four other background runs with the same joint grouping design and the same associated phase offset values were run, which the user did not see. Fitness was set to the mean distance produced by these five runs. More background runs would have been preferable, but we were

[2] http://jbullet.advel.cz

[3] http://bulletphysics.org

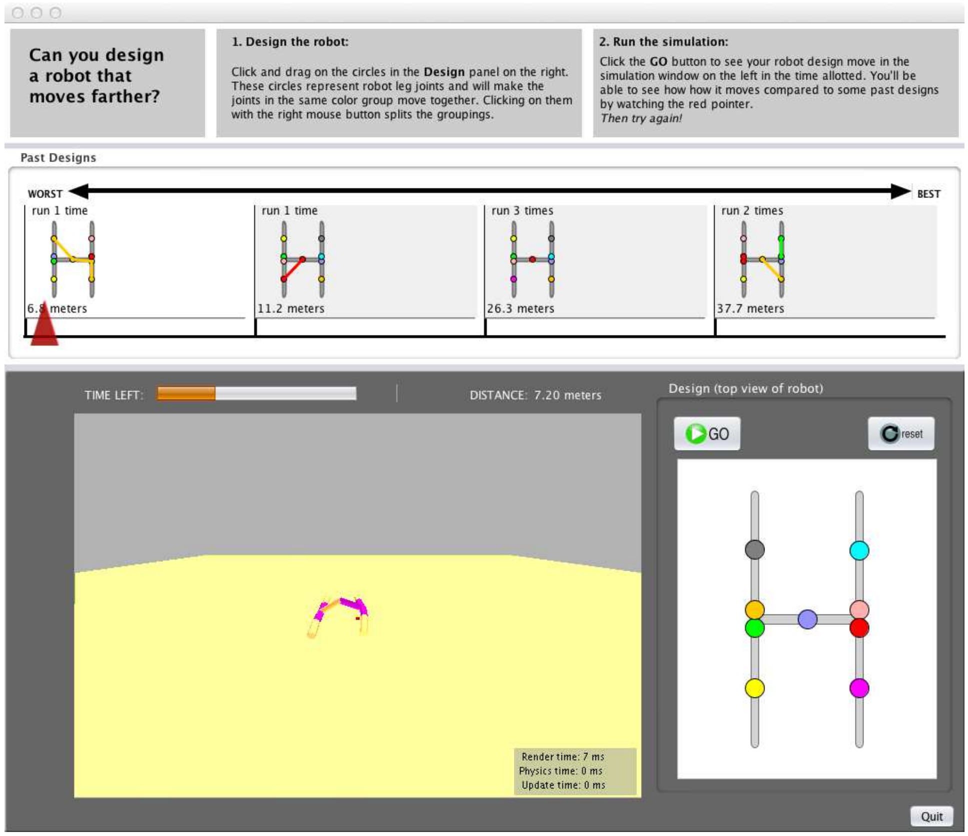

Figure 1: Screenshot of web-based design tool. A description of how to use the tool is on the top with an arrow indicating that the ordering of designs is from best to worst. The red arrow below the designs indicates in real-time how the current robot compares to the historical designs displayed. The *GO* button runs a robot simulation and the *reset* button loads a random design. The lower left panel is where the visualization of the robot movement is displayed.

constrained by the need to maintain usability of the tool. Running multiple physics simulation processes at once required enough computing resources that more than four background runs might have degraded the user experience.

Collaborative and Partitioned Groups

When a user opened the application, they were randomly assigned to either a control or experimental group. In the experimental group, users were allowed to see a subset of at most ten past designs created by other users in the experimental group. The ten designs were chosen randomly. If they returned to the site, they would see a different set of ten random designs. Users in the control group were limited to seeing only their own past designs in the history panel. If they ran more than ten designs, ten of their past designs were randomly shown. Users assigned to the control group would remain in the control group even if they left the site and then returned later, and similarly for the experimental group. Since users were anonymous, this group assignment was enforced using their IP address. As such, if a user were to use the platform from two different computers they could have been exposed to another group. The designs in the history panel were ranked from "worst" (left) to "best" (right) (by best average distance that was achieved for that particular design). The average distance moved by the robot with a given joint grouping was reported under that design. The number of times that particular design had been evaluated was reported above it. Users could use the designs in the history panel as inspiration for how best to improve on the designs and could contribute more runs to a particular design if they wished. They also had the option of generating a random design by clicking the Reset button and altering it into a new design of their making.

Evolutionary Algorithm

The initial values used for each joint group's phase-offset were randomly selected from a uniform distribution between zero and 2π radians. If a given design for a joint grouping was reused, a mutated version of the best past phase values was used in this new simulation. Mutation was implemented by picking a new value for each grouping with a probability of 0.1. Thus, the optimization used a hill-climber algorithm in which fitness (mean distance) was the basis for drawing from the population of eligible individuals (those that matched a given joint grouping). This best individual was then mutated. The distance that the robot was able to achieve with a given joint grouping and phase offset combination was interpreted as the fitness for that set of phase offsets. When multiple designs were created, the system became a parallel hill climber. Each design corresponds to an individual hill climber, and when a user creates a design that was previously created by another user (or re-created a design she has already created herself), another iteration of that hill climber was executed. If a user created a new design, a new hill climber with an initially random set of phase offsets was assigned to that design. The search space was thus a combination of both the combinatorial design space of joint group configurations and the real-valued space of phase-angle values to assign to each of those joint groupings.

User Incentivization

In order to incentivize users to participate in this experiment - and maintain their continued interest in using it - a sliding pointer under the *history panel* was incorporated into the user interface. As the robot moved, its real-time distance was indicated by the sliding pointer relative to the visible historical designs in the *history panel*. This allowed the user to visualize how well their design performed relative to a subset of random past designs, either created by themselves (the control group) or their own and others' designs (the experimental group). A secondary function of the performance slider was to gamify (Deterding et al. (2011)) the platform. We hoped that the user would be motivated to either create new designs that outperformed past designs or provide additional computation to previous designs to improve on them. In short, the user was imbued with the sense that they were in direct competition with a subset of historical designs.

Results

A total of 57 anonymous volunteer users participated in the study via the web. The platform was advertised on online message board systems to attract participants. Participants were not financially rewarded for their participation. Of the users that volunteered, 31 were assigned to the control group and 26 were assigned to the experimental group. The control group ran the simulation a total of 529 times and the experimental group ran the simulation a total of 581 times (see Table 1).

A random sampling of designs created by users in both the control and experimental groups can be seen in Figure 5. We aggregated the maximum distance values each user was able to achieve. These maximum values were collected per group: the control (n=31) and experimental (n=26). The distributions of the control and experimental group were tested for rejection of the null hypothesis. The null hypothesis in this case was that the events come from the same distribution - that there is no difference between users who worked collectively versus those that worked individually. Using the t-test to reject the null hypothesis under the assumption of normality, we found significant cause to reject the null hypothesis at the 0.05 alpha level (p-value of 0.0404). Testing for the rejection of the null hypothesis without the assumption of normality also gave significant cause to reject the null hypothesis at the $\alpha = 0.05$ significance level, using the non-parametric Wilcoxon Rank Sum test (p-value of 0.0415).

	Control	Experimental
Number of users	31	26
Number of runs	567	581
Number of designs	76	78

Table 1: User statistics

Boxplots of the two distributions can be seen in Figure 3.

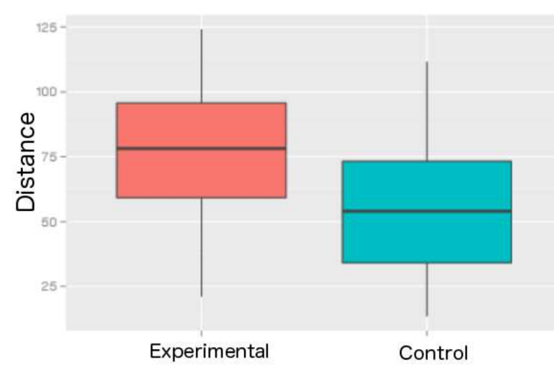

Figure 3: Distance distribution and p-values for significance between experimental and control groups. Significance tests: t-test p-value=0.0406, Wilcoxon Rank Sum p-value=0.0418.

Discussion

We have shown that the control and experimental groups are significantly different at the α level of 0.05. As such, we can say with some degree of confidence that the control and experimental groups are not drawn from the same distribution. That is, the maximum distances achieved by robots designed and optimized by those working collectively was significantly different than the maximum distances of robots designed and optimized by those working individually. Furthermore, the largest distances achieved by the group working collectively were higher than those achieved by the group of individuals working alone. The experiment thus shows that working in a cooperative setting improved the overall search for better robot designs under these experimental settings.

These results could be a consequence of the fact that the experimental group was exposed to more total designs as a whole. When users of the experimental group first opened the platform, they were exposed to designs from past users (with the exception of the first user). However, exposure to past designs by other users does not necessarily imply that the experimental group has an advantage. The hill climber assigned to a given design may have been able to perform more iterations in the experimental group compared to the hill climber assigned to the same design in the control group, because multiple users in the experimental group may have contributed computational effort to that hill climber. However, this would only advantage the experimental group if this extra effort was directed toward an eventually successful design. Groupthink may have led users in the experimental group to contribute effort to one initially promising design, yet caused them to neglect a design that was less successful at the outset but may have yielded a superior controller in the long run.

It is also possible that the designs to which the experimental group were exposed informed their design decisions, which led to improved designs. Isolating the cause for improved designs in the experimental group will be pursued in future work.

Our experiment involved two embedded search spaces: 1) the combinatorial assignment of joints to phase-locked groups, and 2) the assigning of phase offsets to each group. The users directly searched the first, combinatorial space and the optimization algorithm searched the second, real-valued space. The interaction between the two search spaces may contributed to a stronger separation between the control and experimental groups. Additionally, the simulation of robot movement was not entirely deterministic; although there was a clear separation of mean distances achieved by different configurations. A plot of distributions of one hundred runs for several random configurations can be seen in 4. There is a clear separation of distributions between "good" (large distance traveled) versus "bad" (small distance traveled) configurations as to inform the user's intuition. And this non-determinism and large search space was present in both the control and experimental population so each group operated close to the same conditions.

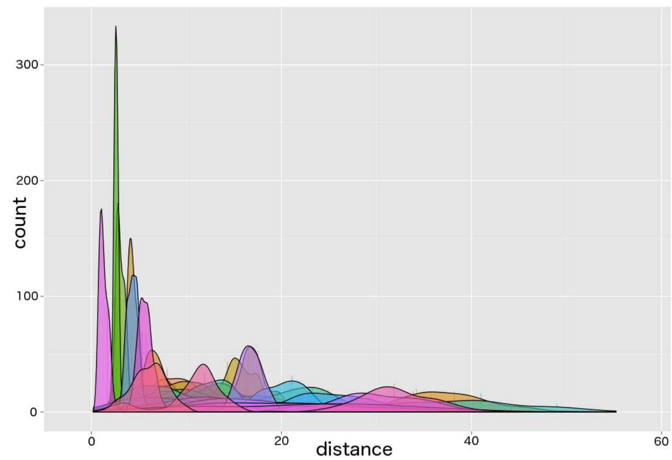

Figure 4: Distance distributions of robot movement for several joint group configurations. Each configuration was run 100 times.

As is common in web-based usage, a small minority of designs attract the majority of computation (Wilkinson (2008)). We can see in Figures 6 and 7 that more users focused their attempts on those designs that were able to move farther. That is, more users ran designs that were able to achieve higher distances in the experimental group; and more design evaluations were dedicated to higher fitness designs. Instead of focusing on designs that exhibited comical or low fitness behavior, users elected to add computing effort to promising, high fitness designs. Although there is not enough data to make definite conclusions about the possibility of "leap-frogging" (the tendency for users to draw inspiration from designs by other users to focus their efforts on promising designs) taking place in the experimental group, the data seem to suggest the possibility of positive feedback contributing to better performance when users worked collectively.

Using human subjects in evaluating fitness and developing new designs presents a number of challenges. When developing the platform, some effort was devoted to gamifying the system. The distance indicator was used to inspire the user to try and "beat" previous designs. Even though written instructions and the distance indicator were used to motivate the user to attempt to evaluate designs based on their ability to cause the virtual robot to move as far as possible, it is possible that users were either unmotivated by the preconceived goal of the simulation or had contrarian intentions to design robots that move as little or as comically as possible.

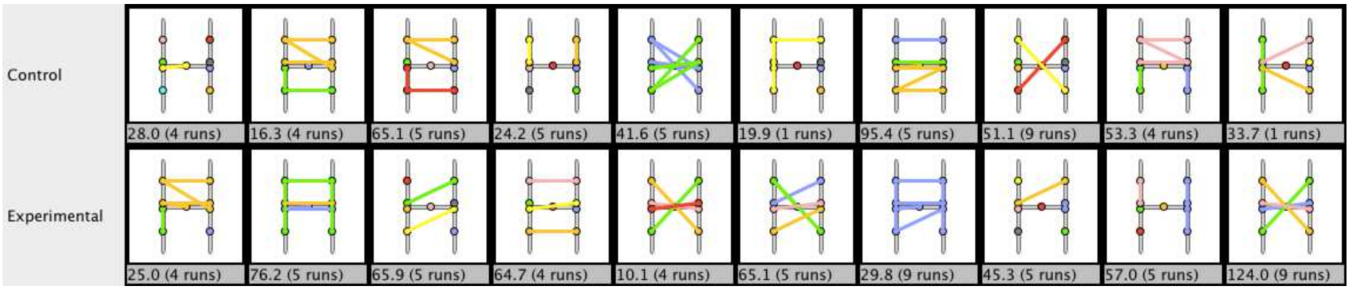

Figure 5: A random sample of designs created by users from the control and experimental groups, shown with best distance achieved and number of times a user evaluated that design (a “run”).

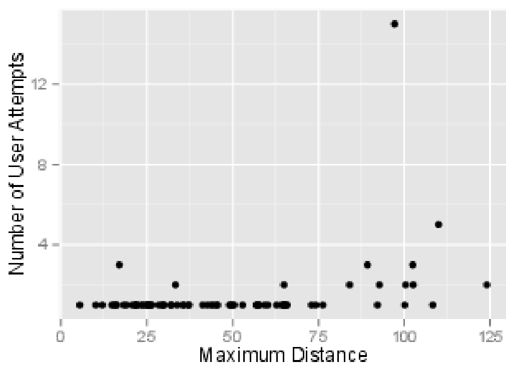

Figure 6: Each point corresponds to a design in the experimental group. The plot shows the number of users that attempted a unique design vs the best distance that that design achieved. Note that the design that received the most attention at 15 attempts was the “default” design that appeared when the application was launched in the user’s web browser.

Conclusion and Future Work

We showed that in a hierarchical search space of robot designs in which human users searched the combinatorial level of the space and an optimization algorithm searched the real-valued level, a group that collectively searched the space was able to outperform a group of individuals that worked alone. The distributions of best robot distance achieved per person in each group were significantly different at an α level of 0.05 under both normal and non-parametric assumptions.

Using our framework, the user was shown the best distance that a given robot configuration moved as well as the number of runs contributed to each design. Novelty of a design might also be a useful message to communicate. Users might be incentivized to explore unknown regions of the search space if they were told that they came up with a design that no one else has tried. Additionally, the amount of symmetry in a given design or an indication of the spread of distances realized, such as standard deviation could be communicated. Social networks may be another means of incentivizing search: indicating which designs the user’s friends have already explored could be built into the framework.

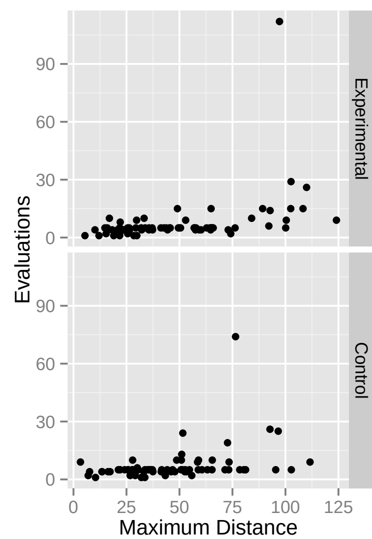

Figure 7: Number of evaluations for each design vs the maximum distance that design achieved. As is the case in Figure 6, the design that received the most evaluations was the “default” design that appeared when the application was launched in the user’s web browser.

Another aspect of this work was the use of a symbolic language to easily communicate simple concepts of robot movement between human users. The language was so specific that it would only work with one form of robot. Future work on generalizing this language to quickly communicate complex ideas between collaborators may warrant further investigation.

Acknowledgments

This work was supported by the National Science Foundation (NSF) under project DGE-1144388. This work was also supported by the NSF under grant PECASE-0953837, and by the Defense Advanced Research Projects Agency (DARPA) under grants W911NF-11-1-0076 and FA8650-11-1-7155.

References

Akrour, R., Schoenauer, M., and Sebag, M. (2011). Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 12–27. Springer.

Bongard, J. C. and Hornby, G. S. (2013). Combining fitness-based search and user modeling in evolutionary robotics. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 159–166. ACM.

Celis, S., Hornby, G. S., and Bongard, J. (2013). Avoiding local optima with user demonstrations and low-level control. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 3403–3410. IEEE.

Chernova, S., DePalma, N., Morant, E., and Breazeal, C. (2011). Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *RO-MAN, 2011 IEEE*, pages 21–26. IEEE.

Couzin, I. (2007). Collective minds. *Nature*, 445(7129):715–715.

Cox, P. T., Risley, C. C., and Smedley, T. J. (1998). Toward concrete representation in visual languages for robot control. *Journal of Visual Languages & Computing*, 9(2):211–239.

Cox, P. T. and Smedley, T. J. (1998). Visual programming for robot control. In *Visual Languages, 1998. Proceedings. 1998 IEEE Symposium on*, pages 217–224. IEEE.

Dawkins, R. (1996). *The blind watchmaker: Why the evidence of evolution reveals a universe without design*. WW Norton & Company.

Deterding, S., Sicart, M., Nacke, L., O'Hara, K., and Dixon, D. (2011). Gamification. using game-design elements in non-gaming contexts. In *PART 2———Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 2425–2428. ACM.

Doan, A., Ramakrishnan, R., and Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96.

Dozier, G. (2001). Evolving robot behavior via interactive evolutionary computation: From real-world to simulation. In *Proceedings of the 2001 ACM symposium on Applied computing*, pages 340–344. ACM.

García-Valdez, M., Trujillo, L., de Vega, F. F., Guervós, J. J. M., and Olague, G. (2013). Evospace-interactive: a framework to develop distributed collaborative-interactive evolutionary algorithms for artistic design. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 121–132. Springer.

Green, T. R. G. and Petre, M. (1996). Usability analysis of visual programming environments: a cognitive dimensions framework. *Journal of Visual Languages & Computing*, 7(2):131–174.

Kan, H., Duffy, V. G., and Su, C.-J. (2001). An internet virtual reality collaborative environment for effective product design. *Computers in Industry*, 45(2):197–213.

Kowaliw, T., Dorin, A., and McCormack, J. (2012). Promoting creative design in interactive evolutionary computation. *IEEE transactions on evolutionary computation*, 16(4):523.

Larkin, J. H. (1989). Display-based problem solving. *Complex information processing: The impact of Herbert A. Simon*, pages 319–341.

Lund, H. H. and Miglino, O. (1998). Evolving and breeding robots. In *Evolutionary Robotics*, pages 192–210. Springer.

Lund, H. H., Miglino, O., Pagliarini, L., Billard, A., and Ijspeert, A. (1998). Evolutionary robotics-a children's game. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 154–158. IEEE.

Nii, H. P. (1986). The blackboard model of problem solving and the evolution of blackboard architectures. *AI magazine*, 7(2):38.

Peterson, J., Hager, G. D., and Hudak, P. (1999). A language for declarative robotic programming. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1144–1151. IEEE.

Pfeiffer Jr, J. J. (1998). Altaira: A rule-based visual language for small mobile robots. *Journal of Visual Languages and Computing*, 9(2):127–150.

Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., and Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403.

Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *J. Artif. Intell. Res.(JAIR)*, 21:63–100.

Takagi, H. (1998). Interactive evolutionary computation: System optimization based on human subjective evaluation. In *IEEE Int. Conf. on Intelligent Engineering Systems (INES98)*, pages 17–19.

Von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM.

Wilkinson, D. M. (2008). Strong regularities in online peer production. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 302–309. ACM.

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., and Malone, T. W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688.

Yu, L. and Nickerson, J. (2011). Generating creative ideas through crowds: An experimental study of combination. In *Thirty Second International Conference on Information Systems*.