BrainCrafter: An Investigation Into Human-based Neural Network Engineering

Jan Piskur, Peter Greve, Julian Togelius, and Sebastian Risi
japi@itu.dk, pgr@itu.dk, julian@togelius.com, sebr@itu.dk

***Abstract*—This paper presents the online application *BrainCrafter*, in which users can manually build artificial neural networks (ANNs) to control a robot in a maze environment. Users can either start to construct networks from scratch or elaborate on networks created by other users. In particular, BrainCrafter was designed to study how good we as humans are at building ANNs for control problems and if collaborating with other users can facilitate this process. The results in this paper show that (1) some users were in fact able to successfully construct ANNs that solve the navigation tasks, (2) collaboration between users presented difficulties and (3) the human-developed ANNs that managed to solve the task had certain regularities, suggesting that humans can use some of their intuition and spatial understanding in the design of ANNs. Most importantly, the initial results in this paper can serve as a starting point for investigating how to best combine human and machine design capabilities to create more complex artificial brains.**

I. Introduction

The idea behind evolutionary robotics is simple: you let a robot learn how to behave by itself through an advanced form of trial and error. You just have to define what the robot should do, not how, and encode this into a fitness function that rewards good behavior and punishes bad. Neuroevolution – the creation of artificial neural networks (ANNs) through evolutionary algorithms – will take care of the rest.

At least in theory. While neuroevolution has shown promising results for solving a variety of difficult control tasks [1]–[6], these are typically tasks that require exquisite fine-tuning but not a large and varied behavior repertoire to solve. Neuroevolution is rarely used to learn complex behaviors or to solve complex tasks. And the neural networks that are evolved are nowhere near as complex as those of a human or even a fly or snail. It could legitimately be argued that neuroevolution, and by extension evolutionary robotics, has so far failed to "scale up".

The malady behind the lack of scaling up has been diagnosed in various ways and various remedies have been proposed. One suggestion is that the root of the problem is the abundance of deceptive fitness landscapes, where the fitness gradient does not lead to a global optimum from most places. In problems with such fitness landscapes, algorithms tend to get stuck in local optima [7]. *Novelty search* [8] was designed specifically as a remedy to this problem. Novelty search avoids the fitness function altogether, instead introducing a novelty metric that only encourages the search for novel behaviors.

Another diagnosis is that the types of fitness functions we can encode do often not allow for sufficiently complex behaviors or networks to emerge. This limitation could be mitigated by including human input in the construction of the neural network. In many cases a human can see a novel and potentially useful evolved behavior before this behavior leads to improved fitness [9]; conversely, a human might be able to see what is wrong with a particular evolved solution, and suggest some way in which the behavior pattern could be improved. Human input could take at least two different forms: the human guiding evolution through evaluating candidate behaviors, i.e. acting like a fitness function, and the human explicitly designing the network or part of it. Using humans as the fitness function is called *interactive evolutionary computation* (IEC). IEC has been studied for more than a decade and used for a number of different domains [10]. IEC can also be combined with novelty search into something called *novelty-assisted interactive evolutionary computation* (NA-IEC) and has proven even more useful [9].

Another approach to combining human input and computational evolution is *mixed-initiative co-creation* [11]. In this paradigm, humans and computers can both take initiative, and change the artifact that is being created. Several mixed-initiative applications to design computer game levels have been developed, showing the power of combining human and machine creativity [12]–[16]. In addition to being a creative partner in its own right, the computer can act as intermediary and facilitator for human-to-human collaboration. Such collaboration, particularly in the form of "crowdsourcing", has shown remarkable results in applications such as Foldit [17], [18] where players were able to solve an important problem relating to an HIV enzyme within three weeks [19].

So far, these approaches have not been applied to creating neural networks, despite the apparent limits of purely evolutionary approaches. Recently, however, Risi et al. [20] suggested how mixed-initiative and crowdsourcing techniques could be combined in an effort to produce ANNs with a high degree of complexity. This would involve humans collaborating with each other and evolution to create neural networks. However, ANNs are often regarded as "black boxes" [21], [22], not accessible to human understanding; trained or evolved networks are frequently opaque and incomprehensible to even the inventors of the algorithms that created them. This issue raises the question if it is possible for humans to *manually* design them. A step in this direction is to investigate how humans construct ANNs and also, if and how collaborations

Jan Piskur, Peter Greve, and Sebastian Risi are with the Center for Computer Games Research, IT University of Copenhagen, 2300 Copenhagen, Denmark. Julian Togelius is with New York University, 2 Metrotech Center, Brooklyn, 11201 New York.

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

between the human creators could be utilized in this domain.

Several attempts have been made to create specialized languages to allow humans to simply create and edit robot behavior; some of these are "graphical" programming languages that let users program behavior through connecting blocks of functionality. A popular example is the language that is use for programming the *Lego Mindstorms* robotic toy. Such languages typically operate at a higher level of abstraction than a neural network, on the presumption that a higher level of abstraction is easier to work with for human thinking. However, neural networks are by far the most common choice for controller representation in evolutionary robotics [1], [4]. This has several reasons, most importantly a desire to allow the evolutionary process maximum freedom and minimal bias in creating a controller (historically, the similarity to models of biological brains has also been a reason for this choice). This makes it important to investigate how well humans can interact with this particular controller representation, as opposed to languages designed for humans.

This paper presents some first steps towards understanding if humans can solve this challenging design problem collaboratively through intuition and spatial reasoning. To achieve this objective the online application *BrainCrafter* (http://braincrafter.dk) was created. In BrainCrafter users can build their own neural networks to control a robot in a maze navigation task. The program allows users to build ANNs from scratch or on top of existing networks made by other users. The effects of ANN modifications on the behavior of the robot can be observed in real-time, allowing the user to gain some understanding of the relationship between neural structure and resulting behavior.

As the results in this paper show, some users were in fact able to successfully construct ANNs that solve the navigation tasks. Additionally, it was found that collaborations between users were challenging, likely due to the difficulties of communicating intentions behind the design of an ANN between them. Interestingly, the user-created ANNs shared certain regularities in their connectivity patterns. For example, connections of the rangefinders on opposite sides (left and right) often had inverted connection weights to the left/right output. While these preliminary result suggest that humans can use some of their intuition and spatial understanding in the design of ANNs, to what extend this ability generalizes to more complex tasks is an important open question.

In the future, the initial results in this paper should lay the foundation for a mixed-initiative network engineering approach that can benefit from the different skill sets of a human and a computational creator. For example, while this paper showed that humans can use some of their spatial understanding in the design of network topologies, a computational method is likely more effective at fine-tuning specific synaptic weights. This combined approach could allow us to solve challenging problems in neuroevolution that are too difficult to be solved by humans or machines on their own.

II. Background

This section briefly reviews neural networks, which are constructed by human users in the experiments reported in this paper. Additionally, it gives background information on the popular neuroevolution method NEAT, novelty search and presents previous work on human-in-the-loop approaches.

Artificial neural networks (ANNs) are computational models inspired by biological neural networks, and widely used as function approximators in various areas of machine learning and control. They are organized as networks of "neurons" or units that receive inputs, and propagate the sum of their inputs to other neurons. While ANNs used for supervised learning are often trained by the backpropagation algorithm [23], it is also possible to train them using evolution; this is particularly effective when training ANNs for control. A network is typically made up of an input layer, output layer, and hidden layers. The input layer receives signals from the sensors, which are then propagated through the hidden layers to the output layer. For each neuron j in the input, hidden, and output layer its activation is calculated by:

$$I_j = \sum_i w_{ij} O_i + \theta_j. \tag{1}$$

Here w_{ij} is the weight between neuron j and neuron i, O_i is the output from neuron i, and θ_j is the bias. The weighted input from each incoming connection is summed and the bias is added. An activation function (typically a sigmoid function) is then applied to calculate the final output value of the neuron:

$$S(I_j) = \frac{1}{1 + e^{-I_j}}. \tag{2}$$

A. Neuroevolution of Augmenting Topologies (NEAT)

The process of manually designing ANNs in this paper is compared to previously published results on applying neuroevolution to control a robot in a maze navigation task [9]. Woolley and Stanley use a method called *neuroevolution of augmenting topologies* (NEAT; [24]), which starts with a population of simple neural networks and then adds complexity over generations by adding new nodes and connections through mutations. By evolving networks in this way, the topology of the network does not need to be known a priori; NEAT searches through increasingly complex networks to find a suitable level of complexity. Because it starts simply and gradually adds complexity, it often tends to find a solution network close to the minimal necessary size [24].

B. Novelty Search

A method that recently has shown promise in avoiding deception in a variety of different domains (including the maze domain in this paper) is novelty search, which is based on the radical idea of ignoring the objective [7]. The idea is to identify novelty as a proxy for stepping stones. That is, instead of searching for a final objective, the learning method is rewarded for finding any behavior whose functionality is significantly different from what has been discovered before. Thus, instead of an objective function, search employs a novelty metric. That way, no attempt is made to measure overall progress. In effect, such a process gradually accumulates novel behaviors.

C. Human-in-the-Loop Approaches

Collaborative games like *Foldit* demonstrate some of the power of crowdsourcing the human brain's natural abilities for certain tasks that involve e.g. pattern matching or spatial reasoning; these type of tasks are hard to solve with computational approaches. Foldit is a collaborative online game in which the goal of the user is to fold proteins into their most compact three-dimensional structure [25]. Predicting the correct protein structure is computationally very expensive because of the high degrees of freedom. However, using their pattern matching abilities together with the ability to collaborate, allowed Foldit users to configure the structure of a particular enzyme, which was an unsolved goal for the last 15 years.

An evolutionary human-in-the-loop approach is *interactive evolutionary computation* (IEC) [10]. The main idea behind IEC is that the user is performing the selection, replacing the traditionally employed fitness functions. IEC methods can also allow users to create content collaboratively. One example of such a system is *Picbreeder* [26], in which users can elaborate on two-dimensional pictures evolved by other users through a web-based system. However, in traditional IEC applications the role of the user is often reduced to solely judging the created artifacts and only "nudging" evolution by deciding between a discrete choice of candidates. In other words, only the computer creates content (e.g. images, ANNs, etc.) and the role of the human is to guide evolution to content they prefer.

The core idea of mixing human knowledge with computational procedures has been taken in several directions. *Human-based genetic algorithms* [27] expands on interactive evolutionary computation by allowing the human to take part in all parts of the algorithm, not only evaluating candidate solutions but also selection and recombination. Similarly, in *hyperinteractive evolutionary computation* [28] human users choose when and where to apply computational operators.

Woolley and Stanley [9] combined IEC with novelty search [7], demonstrating that the approaches complement each other and together address some of the challenges that each method struggles with by itself (e.g. novelty search can get lost in large search spaces, interactive evolution is limited by user fatigue). *Novelty-assisted interactive evolutionary computation* (NA-IEC) combines human intuition with novelty search to help discover agent behaviors for a deceptive maze navigation task and was able to find solutions in fewer steps and faster than novelty search alone.

More recently, Bongard and colleagues showed how the model of a human user can be complementary to the traditionally employed fitness-based search [29] and how utilizing the preferences of multiple users can accelerate this process [30]. In a later study Wagy and Bongard also demonstrated how a group of human users can successfully leverage design intuition from each other in the interactive creation of robot morphologies [31].

III. The BrainCrafter Approach

This paper describes first steps towards an ultimately mixed-initiative approach, in which humans should be able to construct ANNs that solve difficult control tasks in collaboration with each other and with evolutionary algorithms. As a part of developing such an approach it is useful to first determine how good humans are at building such ANNs for robot control problems *without* the help of artificial evolution, which is the focus of this paper. Additionally, an important question in this context is if collaborating with other users can prove useful in the construction of ANNs. Insights from this experiment should also provide useful clues about the strengths of human ANN design and most importantly, non-intuitive aspects of the design process we tend to struggle with (i.e. aspects which would benefit most from the assistance of a computational creator).

The BrainCrafter system presented in this paper is an online application that allows users to build ANNs for a maze-traversing robot by adding neurons and connections in a drag and drop like fashion. While building ANNs the user can observe the resulting simulated robot behaviors in real-time, proving insights into the effects of different network modifications. BrainCrafter also allows users to collaborate by building on high-scoring solutions created by other people.

A. Development

BrainCrafter's graphical ANN editing and simulation environment is based on the Unity game engine. The game is integrated to a web site created with Laraval, a PHP based model-view-controller framework, through the Unity Web Player plugin. JSON encodes the neural networks in Unity, which are transmitted to the web site layer and saved in a database. The ANN family trees are visualized with the javascript library D3.js.

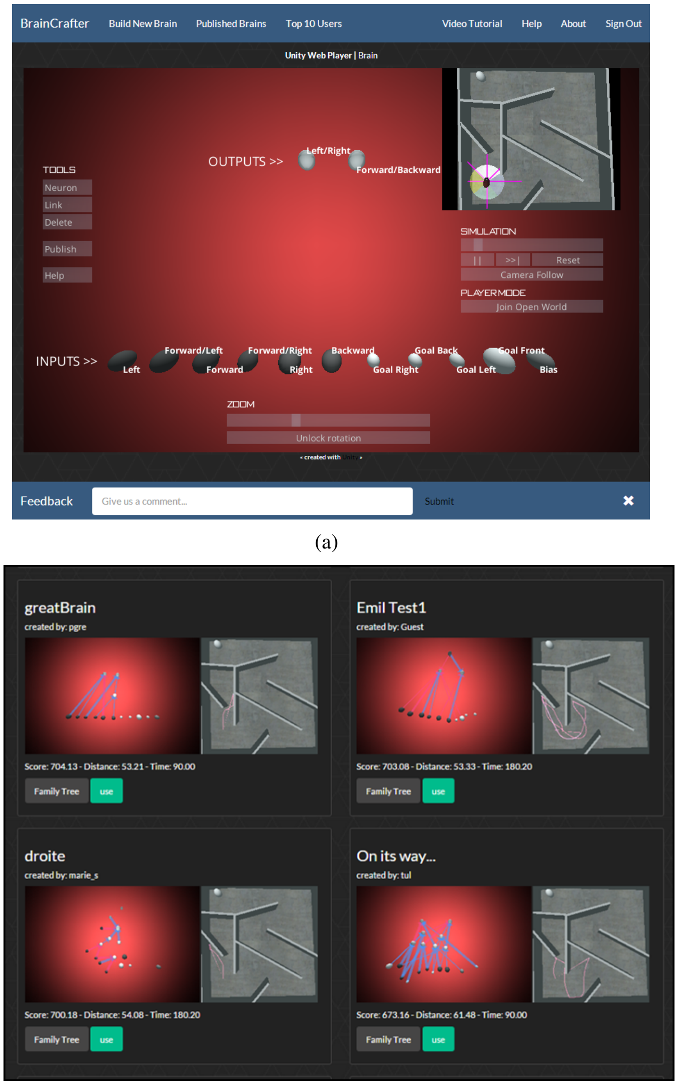
B. Application and User Interface

The main interface (Figure 1a) lets the user connect and add neurons in a drag&drop like fashion. The user can also browse the networks created by others and elaborate on them (Figure 1b). A number of visual features were implemented with the aim to give the user a rudimentary understanding of how the neural network functions. When the input neurons receive input from the sensors and the activation is propagated through the network, the neurons will increase in size relative to the strength of the incoming activation. While constructing the network the user can observe the behavior of the simulated robot in real-time, adjust the speed of the simulation, pause it or reset the position of the robot.

An essential functionality in BrainCrafter is the possibility for users to publish the networks they have created, making them available for other users to elaborate on. The purpose of the publishing mechanism is two-fold. First, it makes the network visible to other users thereby allowing them to collaborate. Second, publishing involves benchmarking the network (i.e. how fast can it solve the given task), thereby gamifying the experience and encouraging competition among the users.

IV. Navigation Task Experiment

In this paper, human neural network engineering abilities are tested in a deceptive maze navigation domain. One such deceptive maze, aptly called the *hard maze* (Figure 2a), was introduced by Lehman and Stanley to demonstrate the power of novelty search [8]. That way, human ANN engineering abilities can be compared to prior results on novelty search and fitness-based search in the same domain. The goal of the robot is to navigate from the start to the end location in the maze in the given amount of time. The hard maze is



(a)

greatBrain
created by: pgre
Score: 704.13 - Distance: 53.21 - Time: 90.00

Emil Test1
created by: Guest
Score: 703.08 - Distance: 53.33 - Time: 180.20

droite
created by: marie_s
Score: 700.18 - Distance: 54.08 - Time: 180.20

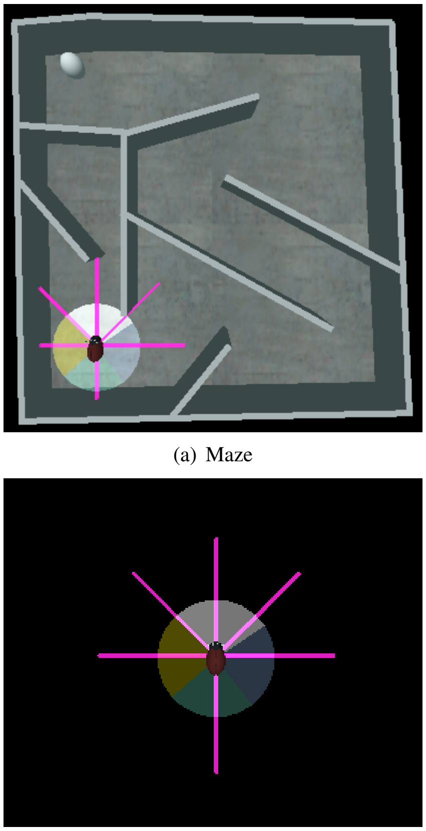
On its way...
created by: tul
Score: 673.16 - Distance: 61.48 - Time: 90.00

(b)

Figure 1. **BrainCrafter User Interface.** (a) BrainCrafter (http://braincrafter.dk) is designed to allow users to easily construct ANNs, while observing the behavior of the simulated robot. The different sensor inputs that the user can connect to are shown at the bottom while the two network outputs are shown at the top. (b) Users can publish their networks, compete for high-scoring solutions and elaborate on the networks of others.

intentionally designed with cul-de-sacs that create local optima in the fitness landscape. These local optima make the deceptive maze a challenging problem for evolutionary algorithms with traditional objective-based performance metrics. The question in this paper is then whether the task is similarly deceptive for a human trying to construct a maze-navigation solution without help from computational approaches.

The robot has two sensor types, rangefinder and pie-slice sensors (Figure 2b). An additional bias input provides a constant activation of 1.0. The pie-slice sensors act as a compass and are activated when a line from the robot to the goal falls within a pie-slice. To give the user visual feedback during the ANN construction, the pie-slice sensors and rangefinders will light up when the user selects the corresponding input neurons.



(a) Maze

(b) Sensors

Figure 2. **Maze Map and Sensor Setup.** (b) The goal of the agent is to is to reach the goal point in the top/left corner. (b) The agent is equipped with six rangefinder sensors that indicate the distance to walls and four pie-slice sensors that act as a compass towards the goal.

Following Lehman and Stanley [8], the robot's two effectors (left/right and forward/backward) result in forces that respectively turn and propel the robot. The forward/backward output moves the robot forward if the activation value is higher than 0.5, and backwards otherwise. The left/right output that rotates the robot around its own axis works accordingly.

V. Results

Data was collected over a period of approximately one month. During that period 48 users signed up and of those, eight users created one or more brains. The total number of brains published was 25. Most users already had some background knowledge about neural networks, while 12.5% had no prior ANN knowledge. Because of Unity's non-deterministic collision detection, each ANN was tested five times to determine its general goal seeking abilities. Four of the 25 networks solved the hard maze in all five trials, while eight of them solved the hard maze in at least one out of the five trails.

Five out of 25 ANNs are collaboratively built, which means they are branched from other users' controllers. Four out of eight active users published a solution that solves the hard maze (Figure 3). Three out of eight active users collaborated on top of existing ANNs. From these three collaborative solutions, one was branched from an existing solution and one from a

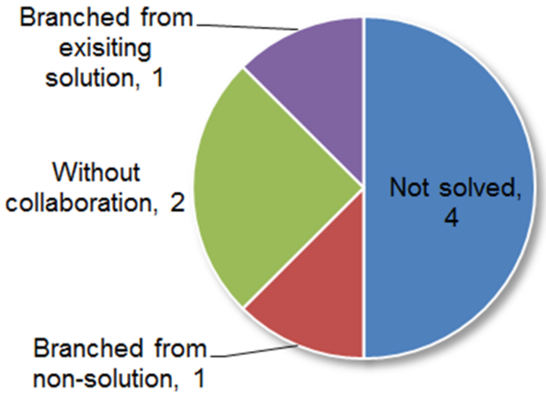


Figure 3. **Human-based Network Engineering Results.** From eight users, four published a brain that solved the hard maze. From these four, two solved it without collaborating with other users, one user solved it by branching from a non-solution network, and one user branched from a network that could already solve the task. The main result is that — while it is not easy – it is indeed possible for a human to construct an ANN solution to the hard maze.

non-solution network. Two of the users created a solution to the hard maze without branching from other users networks (i.e. they discovered solutions independently).

To determine if it is likely that users solved the task by pure chance, 200 randomly generated ANNs were also tested on the maze navigation task. Boundaries for the number of hidden nodes and connections were set following Lehman and Stanley [8] and Woolley and Stanley [9]. The maximum number of randomly added hidden nodes was six and the maximum number of connections was 42. From the 200 randomly generated networks, only one was able to solve the hard maze, suggesting that the users are not just randomly adding neurons and connections but instead follow a more principled approach in the design of these networks. While the fact that a randomly-generated network can solve the hard maze might suggest that the task is not very difficult, prior research shows that it is a rather challenging domain for traditional fitness-based approaches. In fact, Lehman and Stanley [8] reported that a fitness-based approach is only able to solve the hard maze in four out of 30 evolutionary runs.

Figure 4 shows an example of a user successfully elaborating on the ANN built by another user. In the first step the user adjusted the weights from the rangefinder inputs to the left/right output to both have a value of 1.0. This small adjustment changes the balance between turning left/right and going forward, which in effect enabled the robot to break through the small gap in the bottom middle of the maze. While this modification improved the behavior of the robot it still got stuck when facing a wall directly with no walls on either side. Thus the user added two new connections from the forward/left and forward/right rangefinder inputs to the left/right output, each with a weight of 0.5. The resulting ANN was then able to control the robot to reach the goal. This results suggest that, even though not many users choose to build on the design of others, collaboration is possible in principle and can lead to solutions to the maze navigation task.

Procedure	Successfulness	Evals. / Resets	Hidden nodes
Fitness-based NEAT	4 out of 30 runs	-	-
Waypoint directed (non-deceptive)	30 out of 30 runs	26,954 (sd=18,464)	3.5 (sd=2.0)
Novelty search	30 out of 30 runs	33,320 (sd=20,949)	3.3 (sd=1.8)
NA-IEC	30 out of 30 runs	7,481 (sd=6,610)	0.5 (sd=1.01)
BrainCrafter	3 out of 8 users	50 (sd=17.52)	3 (sd=3)

Table I. **Computational Approaches and Human Engineering.** This table shows previous results of different approaches on the hard maze domain [9]. The IEC population size was 12, while the Novelty Search and fitness-based search population sizes were 250, with each run limited to 250,000 total evaluations. For comparison, the number of resets of BrainCrafter and the number of users that discovered non-trivial solutions to the task are also shown.

A. Human-based Network Engineering vs. Evolutionary Approaches

How does human-based network engineering fair when compared to computational approaches? While it is difficult to compare the BrainCrafter approach directly with evolutionary search methods (and not the main point of the paper), it still allows us to gain an idea of the complexity of the investigated domain. Additionally, how different approaches compare can lead to insights into the difficulties of other deceptive domains and the potential in leveraging human intuition in general.

Prior results in the hard maze domain by Woolley and Stanley [9], extending work by Lehman and Stanley [8], are shown in Table I. The main result of their work is that fitness-based approaches are deceived by cul-de-sacs in the hard maze, while novelty search is successful in avoiding the deception in this domain and leveraging human intuition through an interactive evolutionary approach can complement a novelty-based approach.

In BrainCrafter the number of resets (i.e. restarting the simulation with the robot back at the starting position) serves as an indication of the number of times the network has been evaluated by the user. The number of resets for the non-trivial solutions (i.e. networks either branched from non-solutions or build without collaboration) by the three users in BrainCrafter are substantially lower than the number of evolutionary evaluations. However, it is important to note that this paper does not suggest that human ANN engineering alone is a viable alternative when compared to computational approaches. Instead, the results suggest that users can at least use some of their insights and ability to discover promising stepping stones in this maze navigation domain even at the low-neuron-level. Thus now it is possible to compare how users would fair at a higher level of abstraction, which is an important future research direction.

An interesting question is if the solutions created by human users are comparable to the ANNs created through artificial evolution in terms of network complexity. As shown in Table I, the number of hidden nodes for pure novelty search are 3.3 (sd = 1.8) and for the novelty-assisted search 0.5 (sd = 1.01). In BrainCrafter, a user found a solution network with zero hidden nodes, while the other two solutions have three and six hidden nodes. However, due to the lack of human-engineered solutions, no general statistical claims about the differences in network complexities can be made at this point.

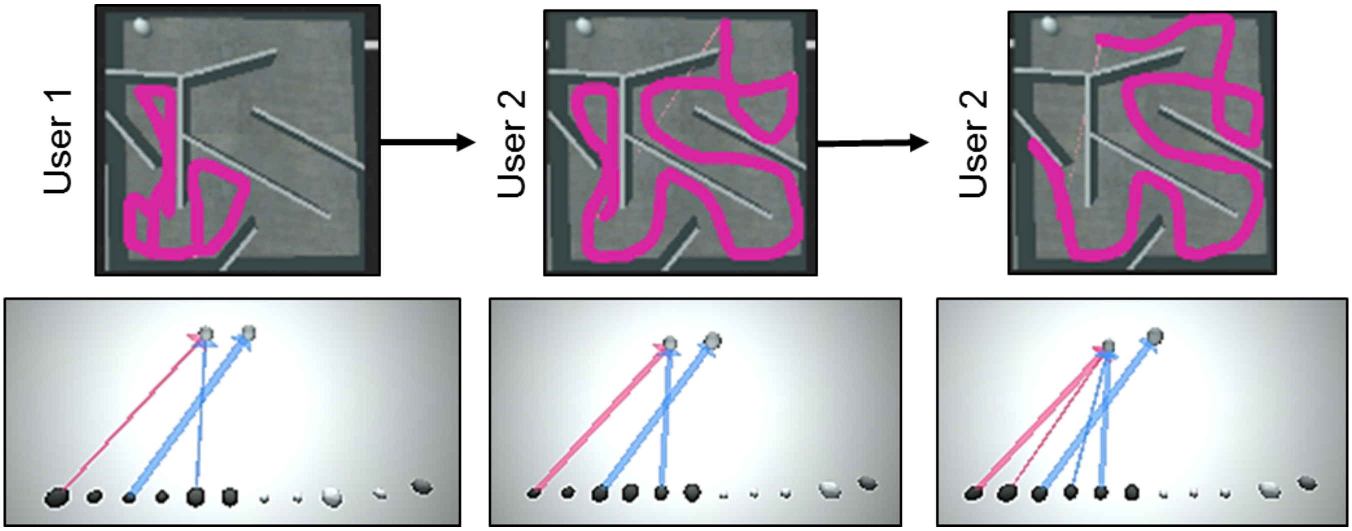
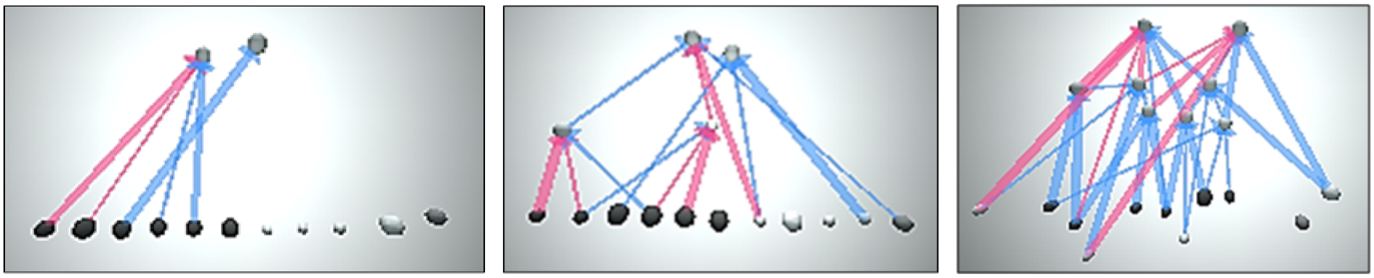


Figure 4. **Collaboration Example.** This figure shows how a user successfully elaborated on the non-solution design of another user. The path of the robot is shown at the top, while the corresponding networks are shown at the bottom. Excitatory connections are blue while inhibitory connections are shown in red.



(a) (b) (c)

Figure 5. **Example Solution Networks.** The discovered solution ANNs shown varying degrees of complexity, from a network with five connections and zero hidden nodes (a) to a network with 31 connections and six hidden nodes (c).

B. Constructed Solution Examples

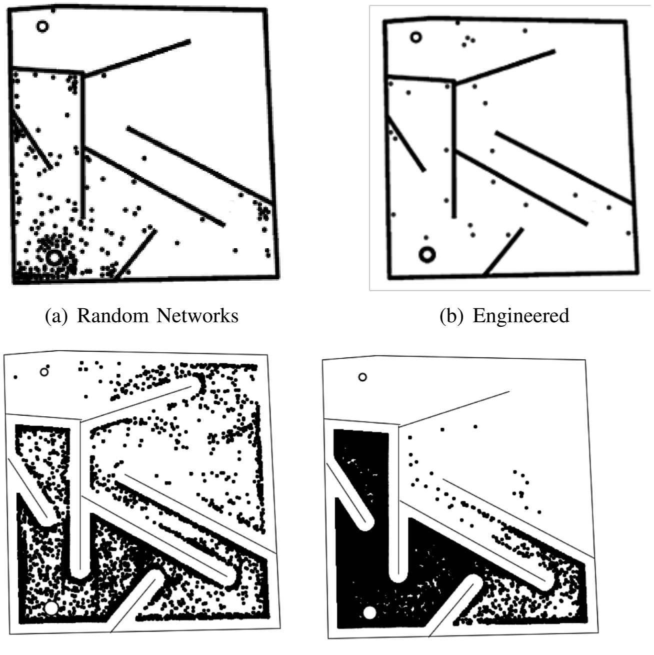
Figure 5 depicts two example network solutions that show a varying degree of complexity. The network with the most complex topology has 31 connections and 6 hidden nodes. In contrast to this, the smallest found solution has only five connections and zero hidden nodes. In general, while it is challenging to draw any definite conclusions from the limited amount of constructed solutions, some networks had indeed common regularities that indicate that human users can at least use some of their intuition in the design process. For example, a common theme is the connection of the rangefinders on opposite sides (left and right) with inverted connection weights to the left/right output (e.g. positive connection from the right rangefinder and negative connections from the leftmost rangefinder; Figure 5a). Another recurring theme was that a group of forward sensors fed positively into the forward/backward output and a group of backward sensors fed negatively into the same output (e.g. Figure 5b).

C. Typical Behaviors

Figure 6 depicts the end points of individuals from randomly generated ANNs, ANNs created by human users, novelty search and fitness-based search. While the limited amount of data points for BrainCrafter makes a comparison difficult, some general tendencies can still be observed. The user-guided search in BrainCrafter has some similarities to novelty search, avoiding the deceptive cul-de-sac areas. Additionally, both novelty search and BrainCrafter networks show more evenly distributed endpoints compared to a fitness-based approach. Also interestingly, the endpoints of the user constructed ANNs are very different from those of randomly created brains, which tend to get stuck in the starting area.

VI. Discussion and Future Work

A total of three non-trivial solutions to the hard maze were found in the one month testing period. Some of the solutions suggest that human insights on how to solve a maze can help in constructing these networks. For example, a well-known strategy to solve mazes is to just employ wall following until the maze exit is reached; knowing this strategy seems to help players in designing their networks. Additionally, a common connection theme was to connect rangefinders on opposite sides (left and right) with inverted weights to the outputs. This simple connection scheme results in a robot controller that turns away when it is too close to a wall on either side. Furthermore, while the fact that the resets in BrainCrafter are substantially lower than the evaluations in evolutionary setups does not indicate that users are better at constructing these network, it does however suggest that humans are able to use



(a) Random Networks

(b) Engineered

(c) Novelty Search [8]

(d) Fitess-based Search [8]

Figure 6. **Robot End Locations.** This figure shows the final positions of the robot after the end of the evaluation. The density of points shows how the different approaches behave in the deceptive maze domain. The endpoints of 200 randomly generated ANNs are shown in (a). (b) shows the endpoints of the 25 human engineered network. (c) shows typical runs with novelty-based NEAT, and typical run with fitness-based NEAT are shown in (d). The last two images are taken from [8].

some of their insights and intuition in these kind of domains. Interestingly, humans also seem to solve the task in a way more similar to novelty search instead of an objective-based performance measure (Figure 6).

Only a few of the already existing solutions were improved upon by other users. This may be due to a general lack of user activity, but perhaps also because people may find it difficult to elaborate on other users' ANNs. The inner workings of ANNs are in general difficult to understand and considered as a kind of "black box" [21], [22]. Thus, without any mechanism to convey the idea how a user-constructed ANN in BrainCrafter works, the thoughts that went into its design process (which might be necessary to elaborate on it), can get lost between users. Finding a way to efficiently communicate user intentions in the construction of these networks will be an important future research direction.

While a few users were able to build solution networks to the maze navigation task, a purely human-based construction approach will likely fail for more complex problems. Also not surprisingly, many users reported that it was rather challenging to construct ANNs from scratch, even with prior experience in neural networks. In future revisions of BrainCrafter we will investigate ways of letting users edit the networks at a somewhat higher level of abstraction. This might be in the form of manipulating and inserting complete modules, or specifying certain network constraints (e.g. symmetry). It might also take the form of the user specifying objectives for very short evolutionary runs, so as to further evolve the network in a particular direction; the effects of these might be limited to small circumscribed part of the network. Ultimately we aim to combine human design ingenuity with computational approaches in the most effective and complementary way possible [20]. The insights from the initial experiments reported in this paper should provide a useful starting point for such an endeavor.

It should be noted that most study participants had some knowledge of ANNs, and generally good knowledge of computers including programming experience. It would be interesting to know how (and how well) people without knowledge of neural networks and with lesser computing experience would solve the task. It is probable, but not certain, that they would solve it even less easily and in a different manner.

One might question the fairness of comparing the performance of algorithms based on the number of times the complete task was attempted. When it comes to human users, they might be visualizing some of the effects of the changes they are effecting by doing a "partial simulation" in their heads, and therefore only test solutions they are rather certain of or where their uncertainty about what the network will do is particularly high. However, this sort of difference between an evolutionary algorithm and a human is exactly what we want to capture and characterize in the future. Further, we cannot think of a more fair evaluation metric than the number of fitness evaluations performed; this is the standard metric used in evolutionary computation research.

The initial experiments reported in this paper purposefully let the user edit ANNs at the low-neuron-level instead of a potentially easier higher-level description. The reasons for this are three-fold. While editing ANNs at a higher-level is likely easier for the user, wrong assumptions in how users use their intuition in creating ANNs could lead to sub-optimal high-level editing tools. Additionally, while it does not initially seem reasonable to use human resources at such a low level, humans have shown remarkable abilities in complicated design task, when allowed to effectively collaborate. Therefore their ability at the low-level should be tested before moving to higher level tools. Third, working at the low-level could be used as an educational tool, allowing students to get familiar with some basic notions of the network operators.

VII. Conclusion

This paper presented BrainCrafter, an online tool that allows users to build ANNs for a maze navigation task in a collaborative fashion. The framework enables users to create networks in a drag and drop manner, while observing the resulting robot behavior in real time. The main result is that it is indeed possible to construct an ANN to solve a maze navigation task without the help of computational approaches. Interestingly, the solutions hint at our human ability to incorporate some intuition in the neural network construction. The initial exploration in this paper is a step towards determining the best way to combine human and machine design capacities when it comes to designing artificial brains.

References

[1] S. Nolfi and D. Floreano, *Evolutionary robotics.* Cambridge, MA: MIT Press, 2000.

[2] D. Floreano, T. Kato, D. Marocco, and E. Sauser, “Coevolution of active vision and feature selection,” *Biological Cybernetics*, vol. 90, pp. 218–228, 2004.

[3] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, “Real-time neuroevolution in the nero video game,” *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 6, pp. 653–668, 2005.

[4] D. Floreano, P. Dürr, and C. Mattiussi, “Neuroevolution: from architectures to learning,” *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008.

[5] F. Gomez, J. Schmidhuber, and R. Miikkulainen, “Efficient non-linear control through neuroevolution,” in *Machine Learning: ECML 2006*. Springer, 2006, pp. 654–662.

[6] J. Lehman and R. Miikkulainen, “Neuroevolution,” *Scholarpedia*, vol. 8, no. 6, p. 30977, 2013, revision no. 133684.

[7] J. Lehman and K. O. Stanley, “Novelty search and the problem with objectives,” in *Genetic Programming Theory and Practice IX*. Springer, 2011, pp. 37–56.

[8] ——, “Abandoning objectives: Evolution through the search for novelty alone,” *Evolutionary computation*, vol. 19, no. 2, pp. 189–223, 2011.

[9] B. G. Woolley and K. O. Stanley, “Exploring promising stepping stones by combining novelty search with interactive evolution,” *arXiv preprint arXiv:1207.6682*, 2012.

[10] H. Takagi, “Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation,” *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[11] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014.

[12] A. Liapis, G. N. Yannakakis, and J. Togelius, “Sentient sketchbook: Computer-aided game level authoring,” in *Proceedings of ACM Conference on Foundations of Digital Games*, 2013.

[13] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive planning and constraint solving for mixed-initiative level design,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 201–215, 2011.

[14] N. Shaker, M. Shaker, and J. Togelius, “Evolving playable content for cut the rope through a simulation-based approach.” in *Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2013.

[15] ——, “Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.” in *Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2013.

[16] J. Zhang, R. Taarnby, A. Liapis, and S. Risi, “DrawCompileEvolve: Sparking interactive evolutionary art with human creations,” in *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt)*, 2015.

[17] J. Keeley, “Protein-folding game taps power of worldwide audience to solve difficult puzzles,” http://www.eurekalert.org/pub_releases/2010-08/hhmi-pgt080310.php, August 4, 2010, [EurekAlert!].

[18] J. Marshall, “Victory for crowdsourced biomolecule design,” http://www.nature.com/news/victory-for-crowdsourced-biomolecule-design-1.9872#/b1, January 22, 2012, [Nature magazine].

[19] M. J. Coren and F. Company, “Foldit gamers solve riddle of HIV enzyme within 3 weeks,” http://www.scientificamerican.com/article/foldit-gamers-solve-riddle/, September 20, 2011, [Scientific American].

[20] S. Risi, J. Zhang, R. Taarnby, P. Greve, J. Piskur, A. Liapis, and J. Togelius, “The case for a mixed-initiative collaborative neuroevolution approach,” *WebAL-1: A Workshop at the 14th International Conference on the Synthesis and Simulation of Living Systems (ALIFE-14)*, 2014.

[21] J. M. Benitez, J. L. Castro, and I. Requena, “Are artificial neural networks black boxes?” *IEEE Transactions on neural networks*, vol. 8, no. 5, 1997.

[22] R. Setiono, W. K. Leow, and J. Y. Thong, “Opening the neural network black box: an algorithm for extracting rules from function approximating artificial neural networks,” in *Proceedings of the twenty first international conference on Information systems*. ICIS, 2000, pp. 176–186.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Cognitive modeling*, vol. 5, 1988.

[24] K. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[25] S. Cooper, A. Treuille, J. Barbero, A. Leaver-Fay, K. Tuite, F. Khatib, A. C. Snyder, M. Beenen, D. Salesin, D. Baker *et al.*, “The challenge of designing scientific discovery games,” in *Proceedings of the Fifth international Conference on the Foundations of Digital Games*. ACM, 2010, pp. 40–47.

[26] J. Secretan, N. Beato, D. B. D’Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley, “Picbreeder: A case study in collaborative evolutionary exploration of design space,” *Evolutionary Computation*, vol. 19, no. 3, pp. 373–403, 2011.

[27] A. Kosorukoff, “Human based genetic algorithm,” in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 5. IEEE, 2001, pp. 3464–3469.

[28] B. J. Bush and H. Sayama, “Hyperinteractive evolutionary computation,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 3, pp. 424–433, 2011.

[29] J. C. Bongard and G. S. Hornby, “Combining fitness-based search and user modeling in evolutionary robotics,” in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 159–166.

[30] A. Bernatskiy, G. S. Hornby, and J. C. Bongard, “Improving robot behavior optimization by combining user preferences,” in *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, vol. 14, pp. 726–733.

[31] M. Wagy and J. Bognard, “Collective design of robot locomotion,” in *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, vol. 14, pp. 138–145.