Emojinating Co-Creativity: Integrating Self-Evaluation and Context-Adaptation

João M. Cunha, Pedro Martins, Nuno Lourenço and Penousal Machado
CISUC, Department of Informatics Engineering
University of Coimbra
{jmacunha,pjmm,naml,machado}@dei.uc.pt

Abstract

Co-creative systems are useful in fostering creativity, often leading to unexpected results. Despite this, the relation between user and system is complex. The level of autonomy given to the system directly influences its potential for creative behaviour and degree of contribution to the cooperation with the user. In this paper, we present our efforts to instil a more creative behaviour to an existing visual blending system – Emojinating. In order to do so, we integrate two functionalities: self-evaluation and context-adaptation.

Introduction

Upon the development of creative systems for the visual domain, one of the biggest issues concerns the dependency on human perception – there is no optimal solution as quality depends on the preferences of the user. One approach that has been seen as suitable for such open-ended problems is Interactive Evolutionary Computation (IEC) (Parmee, Abraham, and Machwe, 2008), in which the evolutionary process relies on human evaluation.

On the other hand, when using an Evolutionary Algorithm (EA), researchers are faced with many challenges concerning the configuration and parameterisation, e.g. which operators should be used. One possible way to tackle this challenge consists in using a trial-and-error approach, where the practitioner experiments with several configurations and then select one that achieves reasonably good results. The need to remove this trial-and-error process led to the emergence of adaptive and self-adaptive algorithms. One of the first EAs to introduce this concept was Evolutionary Strategies (ES). In concrete, ES used a mechanism that adapted the rate with which operators were applied. Over the years many mechanisms have been proposed to adapt all other components of the EA (Kramer, 2010).

The combination of user interaction and system self-adaptation provides an adequate setup for a co-creative relation between human and computer. Different types of collaboration are accepted in such co-creative systems (e.g. partnership or assistantship), which vary in terms of complexity of the relation between human and computer, but also on the level of autonomy given to each of them. Instilling a self-adaptative behaviour to the system may increase its contribution in the co-creative relationship with the user.

Figure 1: Blends for *peace accord*, *car factory*, *security house*, *market depression*, *health risk* and *airline bureaucracy*.

In this paper, we build upon a system (*Emojinating*) that uses visual blending of emoji to produce visual representations of concepts. Cunha et al. (2019) presented an interactive evolutionary version of *Emojinating*, which allowed the user to interact with the system and evolve solutions that fit his/her preferences. However, the system could be said to be more close to a creativity support tool than to a co-creative system, in the way that it mostly responded to user requests. Our goal is to focus on the creative features of the system, leading to an improvement in the co-creative relation.

Our main contributions are: (i) the addition of an automatic evaluator to the evolutionary process, capable of self-evaluating the solutions and adapting to user preferences, and (ii) the introduction of context-adaptation methods. We describe the development of these methods and provide a general analysis of the results obtained.

Background The Emojinating system has three main components: (i) the *Concept Extender* (CE), which uses ConceptNet (Speer and Havasi, 2012) to retrieve related concepts to a given one; (ii) the *Emoji Searcher* (ES), which uses EmojiNet (Wijeratne et al., 2017) to retrieve existing emoji that are semantically related to a given word; and (iii) the *Emoji Blender* (EB), which takes two input emoji (Twemoji 2.3[1]) and produces visual blends.

In general, the system receives a concept from the user that is mapped to two emoji (e.g. emoji A and emoji B), which are then combined through a process of visual blending – emoji B is considered as the base for the blending and emoji A as the replacement. Three different types of operation can be used in the visual blending process: juxtaposition (JUX) – two emoji are put side by side or one over the other (e.g. *peace accord* in Fig 1); replacement (REP) – emoji A replaces part of emoji B (e.g. *health risk* in Fig 1); fusion (FUS) – two emoji are merged together by exchanging parts

[1] github.com/twitter/twemoji, retr. 2020

Proceedings of the 11th International
Conference on Computational Creativity (ICCC'20)
ISBN: 978-989-54160-2-8

(e.g. *airline bureaucracy*). Three types of operation are used: part exchange (part of emoji B is replaced by part of emoji A), part addition (part of emoji A is added) and part removal (part of emoji B is removed).

The initial version of the system (Cunha, Martins, and Machado, 2018) had a deterministic nature. Cunha et al. (2019) used an interactive evolutionary approach to improve the search space exploration, combining a standard Evolutionary Algorithm with a method inspired by Estimation of Distribution Algorithms (EDA). On a macro level, the system uses the EDA-inspired method to direct the search to areas that match the user preference by stimulating weights assigned to emoji and concepts, based on user fitness assignment. On a micro level, the system is able to focus the evolution on certain individuals by allowing the user to select individuals to be mutated. The user is able to conduct two different actions: select individuals, which increases their fitness, affecting the weight system and producing offspring through mutation; or store them in the archive to avoid losing them in the evolutionary process. For more detail, we refer the reader to Cunha et al. (2019).

Approach

Over the last few years, several works have also addressed the evaluation of creativity in co-creative approaches (Jordanous, 2017; Karimi et al., 2018). Two aspects are often considered as requirements in a co-creative system: synchronous collaboration (Davis et al., 2015) and a proactive contribution from both the user and the AI agent (Yannakakis, Liapis, and Alexopoulos, 2014). This means that both agents engage in the interaction and actively contribute to the creative task. Moreover, not only is it required that each agent expresses its own creative ideas but also that it perceives other agents' contributions (Karimi et al., 2018).

The version of the system proposed by Cunha et al. (2019), despite being able to evolve solutions that match the user taste, has a somehow passive behaviour, as the actions of the system are mostly directly triggered by the user. In this paper, our goal is to enhance the creative behaviour of the system, increasing its autonomy and improving the co-operative character of its interaction with the user. In this way, we intend to instil into the system the capability of adequately responding to user actions, thus improving the co-creative relation.

Briefly describing, the previous version of the system could be said to have two agents: an evaluator (user) and solution generator (system). The version described in this paper introduces a second evaluator (system) that is able to select solutions based on its own idea of quality and storing them in its archive. In addition, we improved the solution generator, increasing its ability to adapt to the context. In this section, we describe the changes conducted.

Representation

In the previous versions of the system, only juxtaposition and replacement blend types were implemented. In this version, we improved the blending process by changing the representation used in order to include fusion.

The emoji of Twitter's Twemoji dataset are composed of layers. We consider the blend as the phenotype of an individual. Each individual is encoded using a genotype of two chromosomes, which codify the combination between the two emoji parents. The emoji used in the blend are stored in the first chromosome. The second chromosome is composed of an undefined number of genes, each codifying an exchange between the two emoji. Each gene corresponds to a set of two numbers that refers to emoji A (A) and to emoji B (B), and define how the exchange is conducted ([a,b]). Three different situations may occur: (i) to codify the exchange of layer we use numbers in the $0 - 1$ interval, which correspond to relative position of the layer in the layer array (the number of layers is not the same among emoji); (ii) to codify using the whole emoji instead of the layer, we assign a value of -1; (iii) we use -2 when nothing is to be used of the corresponding emoji. As such, the following cases occur: [-1 , -2] adds A (juxtaposition); [-1 , $\geq$0] replaces part of B with A (replacement); [$\geq$0 , $\geq$0] replaces part of B with part of A (fusion); [$\geq$0 , -2] adds part of A (fusion); and [-2 , $\geq$0] removal of part of B (fusion).

Variation Operators

The system presents the user with a population of 20 individuals (blends) and in each generation the user selects the ones to go through a process of producing offspring. Two different operators exist: crossover and mutation. The produced offspring individuals from both operators are added to an offspring pool. A maximum percentage of the new population (50%) is reserved for the offspring, which are randomly selected from the pool. The remaining percentage corresponds to individuals generated from scratch.

Crossover Operator A crossover occurs when the user selects at least two individuals. Initially, the system only conducted crossover with individuals that shared at least one emoji. Afterwards, we realised this approach severely reduced the possible offspring. As such, we decided that blends with no shared emoji could also be combined.

In order to conduct the crossover, groups of two are randomly made with the emoji selected by the user. Two types of crossover can occur: if the number of exchange genes (second chromosome) is equal to one, one of the emoji of each parent individual is exchanged with the other individual; if the number of exchange genes in both emoji is above one, it conducts a gene crossover. A gene crossover consists in exchanging genes between individuals, using a one-point crossover. The resulting offspring individuals are added to the offspring pool.

Mutation Operators In the previous version of the system (Cunha et al., 2019) only three types of mutation existed (replacement emoji, replaced layer and blend type mutation). With the implementation of the new representation, the types of mutation increased to ones presented in Table 1.

Adaptation

Two types of adaptation can be said to exist: to the user and to situations within the system (context). The former has been addressed by Cunha et al. (2019). One of our goals is to focus on the latter, allowing the system to adapt to the

Table 1: Probability of each mutation type based on the type of blend (JUX, REP OR FUS) of the individual being mutated. RP stands for Replacement Part.*Implemented but not used.

mutation type	JUX	REP	FUS
replacement emoji is changed	40	30	5
base emoji is changed	40	10	0
*blend type changes to juxtaposition**	0	0	0
replaced layer is changed	20	35	25
RP changes from whole to a layer	10	5	0
RP is changed by selecting a new layer	0	0	15
RP changes from a layer to whole	0	0	5

population at the moment, as different stages in the run may require different behaviour from the system. Two different means of context-adaptation were implemented: adaptive blending process (individual generation) and adaptive variation operators (mutation).

The adaptive blending process consists in changing the likelihood of a given type of blend occurring, according to the state of the population. This is used in the generation of new individuals from scratch. The types of blend have different variation potential (juxtaposition has the lowest potential and fusion has the highest). Due to this, our approach is that blend types with higher variation potential should occur more frequently when there are fewer different emoji used in the blends of the population. As such, we assign the probability of each blend type based on the number of different emoji (N_E): for N_E ≥ 20 (higher), JUX $= 10\%$ and FUS $= 20\%$; for N_E ≤ 8 (lower), JUX $= 2\%$ and FUS $= 50\%$. For $8 <$ N_E < 20 the following equation is used to calculate probabilities:

$$\text{LOWER\_VAL} + (\text{UPPER\_VAL} - \text{LOWER\_VAL}) \times \frac{(\text{N\_E} - 8)}{12}$$

where LOWER_VAL and UPPER_VAL are the probability values of the blend type used in the lowest and highest bounds of N_E (e.g. in JUX 2% and 10%, respectively). The value for replacement is always REP $= 100 -$ JUX $-$ FUS.

Regarding mutation adaptation, our initial approach was similar to the adaptive blending process: we tried to assign the same value to each operator and change it according to the state of the population. Later we concluded that due to the characteristics of the problem, this approach would not lead to good results – each type of blend has its own particularities and, therefore, has different mutation requirements. For example, in juxtaposition mutating the replaced emoji is simple as the whole emoji is used, whereas in fusion it is more complex as the layer-based exchanges are relative to the array of layers of each emoji, which varies in number of elements – mutating the replaced emoji in fusion would result in something entirely different. As such, mutation adaptation consists in changing the occurrence probability of each mutation operator according to the blend type of the individual being mutated. We established values for each mutation, depending on the type of blend of the parent (see Table 1). The emoji mutations are independent of the rest. If juxtaposition occurs, none of the rest occurs. If no juxtaposition occurs, any of the other mutation types can occur.

Self-evaluation and selection

In order to give some autonomy to the system, we decided to bring another agent to the evolutionary process. This agent is an automatic evaluator that has two possible actions: evaluate individuals according to its preferences and store individuals in its own archive. The user can see the archive and is able to retrieve individuals from it but only the automatic evaluator is able to add individuals.

Quality Assessment: Criteria Defining criteria for quality assessment of blends is not an easy task. First because quality is dependent on visual attributes but also on conceptual ones (e.g. does the user perceive the concept). Moreover, as they depend on user understanding and perception it makes this an open-ended evolution problem. In this paper, we chose to focus on the first type of criteria (visual attributes). We considered two aspects that are related to the quality of an icon: complexity (the simpler the better) and legibility (should be perceivable in smaller sizes). Also, given that we are conducting visual blending, we need to also consider the degree of change in comparison to the parents. With this in mind, we defined the following criteria:

1. overall complexity: $\frac{1}{\#\text{BLEND\_LAYERS}}$;
2. area exchanged: $\sum_{i=1}^{\#LE(b)} a(l_i)$;
3. relation between added area and added layers: $\frac{\sum_{i=1}^{\#LA(b)} a(l_i)}{\#\text{ADDED\_LAYERS}}$;
4. difference in complexity: $(\#\text{ADDED\_LAYERS} - \#\text{REMOVED\_LAYERS})$.

$\#LE$ is the number of layers exchanged (added+removed) in the blend (b), $\#LA$ the number of added layers and function a calculates the area of a layer l.

Quality Assessment: Fitness Calculation The goal of the automatic evaluator is to be able to assess solutions based on its own idea of quality. In this sense, there are two options: having an evaluator that tries to get similar solutions to the user, in order to present good alternative solutions; or get solutions that are distinct from what the user is selecting. In this paper, we chose to focus on the first approach.

The system's idea of good solutions is therefore dependent on user choices. This is achieved by making the system analyse the blends in the user archive – which are assumed as being good – and afterwards change its idea of a good solution to match these user-selected blends. In the beginning, the system starts with default values (all equal to 1). As soon as the user stores individuals in the archive, the system evaluates them and changes its fitness goal, based on their characteristics. To obtain the goal, the system calculates the average of each criterion for the individuals stored in the user archive, which results in an average blend profile. This profile is then set as the new goal and used for selecting individuals that the system finds interesting. As such, the system goal changes over time, according to user preferences. In order to calculate the fitness of an individual, the system uses a Euclidean distance between the individual and the average profile, which assesses how far away the system is from the goal and is updated at the end of each generation.

Individual selection As already mentioned, the automatic evaluator has its own archive. At the end of each generation, and after calculating the new goal, the system performs an analysis of the population to check for good individuals. The evaluator's archive capacity was set to 5 to avoid storing too many individuals. Also to avoid collecting too many individuals, the evaluator only stores one blend per emoji combination. This way, the system tries to improve the fitness of individuals for each emoji combination. In each generation, the evaluator selects the best blend in the population, checks whether it already has a blend for the emoji combination and proceeds as follows: (1) if there is already a blend for the emoji combination and the fitness of the stored one is lower than the current population best, it replaces the individual in the archive; (2) if it does not have any blend for the emoji combination and the archive has free space, the evaluator stores the blend; (3) if it does not have any blend for the emoji combination but there is no space in the archive, the evaluator checks if the blend to store has higher fitness than the worst individual in its archive and, if so, replaces it. In each generation, the evaluator discards individuals when the difference of their fitness to the best individual in the population is >2 (empirically obtained), which often happens when the fitness goal drastically changes.

Discussion and Future Work

As this paper presents work in progress, the results of the system are, at this point, qualitatively evaluated and discussed by the authors.

In general, the system is able to learn from the user behaviour, which is observed in the storing of similar blends in its archive (e.g. if the user selects blends with a large exchanged area, the system tends to replace the blends in its archive to match the user preference). Moreover, the system archive is useful to highlight blends that the user may have missed as the system only selects blends that were previously shown to the user. However, there are some improvements that need to be made, e.g. the evaluator should avoid blends that are exactly the same to any stored by the user. Regarding the evaluator behaviour, we explored a strategy of searching for blends similar to the user but another possible approach is to try to go in directions that are different from the user's, in order to increase the variety of results. Moreover, a possible future direction is to allow the automatic evaluator to select blends to be reproduced, generating offspring from its own stored individuals.

The values used in this paper were empirically obtained through experimentation and adjustments. However, due to the high number of parameters we consider that further tuning is required. One example is the probability of fusion, which depends on the number of different emoji in the population. Its probability of occurrence was set to a high value (50%) for low emoji number, as it is the type of blend that leads to the highest variety of results – theoretically, this would be suitable in situations in which few emoji exist. However, this does not work when put in practice as it makes it harder to identify both parent emoji (e.g. *airline bureaucracy* in Fig. 1), which worsens the user perception from the first generation. A possible solution may be to also consider the number of the current generation.

Concerning fitness, there are also some issues that need to be addressed. First, using an average of individuals' properties as a goal does not work well for every case – an individual located between two good individuals is not always a good individual. Even more, as we are dealing with pictographs, in which the perception of the concept is more important than some of the considered visual features (e.g. area changed). Further studies are required to better validate the implemented approach. In future work, we will conduct testing with users to understand how the co-creative functionalities help in obtaining better results. A video of the system being used can be seen at `https://rebrand.ly/iccc20short`.

Acknowledgements This work is partially funded by the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, and under the grant SFRH/BD/120905/2016.

References

Cunha, J. M.; Lourenço, N.; Correia, J.; Martins, P.; and Machado, P. 2019. Emojinating: Evolving emoji blends. In *Proceedings of EvoMusArt 2019*. Springer.

Cunha, J. M.; Martins, P.; and Machado, P. 2018. How shell and horn make a unicorn: Experimenting with visual blending in emoji. In *Proc. of the 9th ICCC*.

Davis, N.; Hsiao, C.-P.; Popova, Y.; and Magerko, B. 2015. An enactive model of creativity for computational collaboration and co-creation. In *Creativity in the Digital Age*. Springer. 109–133.

Jordanous, A. 2017. Co-creativity and perceptions of computational agents in co-creativity. In *Proc. of the Eighth Int. Conf. on Computational Creativity, Atlanta, US. ACC*.

Karimi, P.; Grace, K.; Maher, M. L.; and Davis, N. 2018. Evaluating creativity in computational co-creative systems. In *Proc. of the 9th ICCC*.

Kramer, O. 2010. Evolutionary self-adaptation: a survey of operators and strategy parameters. *Evolutionary Intelligence* 3.

Parmee, I. C.; Abraham, J. A.; and Machwe, A. 2008. User-centric evolutionary computing: Melding human and machine capability to satisfy multiple criteria. In *Multiobjective Problem Solving from Nature*. Springer. 263–283.

Speer, R., and Havasi, C. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, 3679–3686.

Wijeratne, S.; Balasuriya, L.; Sheth, A.; and Doran, D. 2017. Emojinet: An open service and api for emoji sense discovery. In *Proc. of ICWSM-17*.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proc. of the 9th Conf. on the Foundations of Digital Games*.