What is Interactive Evolutionary Computation?

Ambre HUTIER[1[24278912]] and Matthias POMMERAY[2[24293342]]

[1] **University of Limerick**, **Limerick**, V94 T9PX, Ireland
[2] **University of Limerick**, **Limerick**, V94 T9PX, Ireland
24278912@studentmail.ul.ie
24293342@studentmail.ul.ie

Abstract: Interactive Evolutionary Computation (IEC) merges traditional Evolutionary Computation with subjective human evaluation, eliminating the quantitative fitness function used in standard methods. By leveraging principles like selection, mutation, and recombination, IEC applies user feedback to guide evolutionary processes, making it particularly suitable for creative and subjective tasks. A notable application is Multi-Objective Interactive Genetic Algorithms (MO-IGA), which balance conflicting objectives by integrating qualitative human input with quantitative criteria, as seen in product design scenarios.

IEC has proven invaluable in domains such as art, music, and user interface design, where subjective human preferences are central. For instance, humanoid robots can learn aesthetic dances through human-aided evolutionary algorithms, while personalized music compositions are generated by analysing indirect physiological feedback. These applications demonstrate IEC's capability to adapt technology to human creativity and emotions.

However, challenges persist, particularly user fatigue during extensive evaluations, which can impact feedback quality and efficiency. Techniques like indirect feedback collection (e.g., eye-tracking) and collaborative user inputs aim to address these issues. Additionally, advancements like large language models enhance IEC's potential by improving user interaction and understanding subjective feedback. Emerging applications include customizing video game NPC behaviours and adaptive medical devices, showcasing IEC's growing versatility. Despite its challenges, IEC represents a transformative approach for integrating human intuition with computational efficiency across diverse fields.

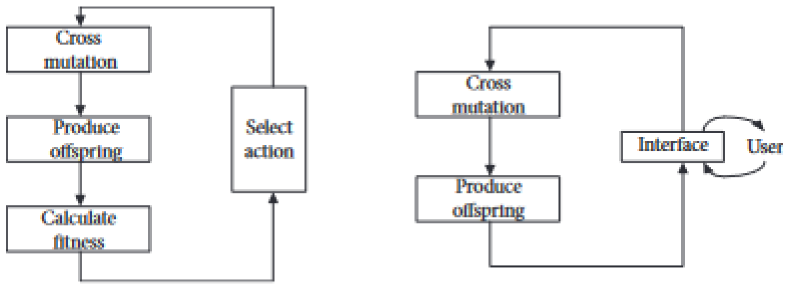
Keywords: Interactive Evolutionary Computation, Evolutionary Computation, Human Users, Muti-objective IGA.

1 Principles of Interactive Evolutionary Computation

1.1 Definition

Interactive Evolutionary Computation, also known as IEC, is a branch of Evolutionary Computation, also known as EC.
Like EC, IEC is inspired by the principles of natural evolution and genetics and simulates evolutionary processes such as selection, mutation, recombination or survival of the fittest. IEC differs, however, in that it combines this evolutionary calculation with subjective human evaluation and selection. In other words, there is no quantitative calculation of the fitness function and selection operation in IEC.

Fig. 1. Evolution strategy of GA and IEC taken from [2]



1.2 Multi-objective IGA

Let's take the example of multi-objective IGA, also known as MO-IGA, to understand how interactive EC works. My explanations are based on research by Alexan-dra Melike Brintrup, Hideyuki Takagi and Jeremy Ramsden [1].
A MO-IGA is a version of the standard IGA designed to solve problems with multiple, sometimes conflicting, objectives. This is where it differs from conventional IGA, which focuses on a single objective such as aesthetics or ease of optimisation. MO-IGA aims to balance the trade-offs between multiple objectives, while incorporating subjective user feedback to guide the evolution process. A simple example of the use of MO-IGA is in the context of product design. In this case, not only aesthetics but also cost and ergonomics have to be considered.
Here are the main steps of the algorithm with examples:
Step 1- Initialise the population P_0 with N number of random solutions.
Step 2- Display the offspring individuals, in this case the solutions of P_0, to the user and ask them for an evaluation, for instance on the aesthetic of each offspring, which can only be evaluated qualitatively.
Step 3- Calculate the fitness for what can be done quantitatively, for example the cost of the offspring.
Step 4- Sort and rank the population P_0 for non-domination.

The idea is to group solutions into different Pareto fronts based on their dominance relationship. They can then be ranked according to their degree of non-dominance, starting with the least dominated group [4].

A solution A dominates a solution B if:

- A is never worse than B in any objective
- A is better than B in at least one objective

Solutions are classified by their level of non-domination, starting with the most non-dominated (Pareto-optimal) solutions.

Let's take an example:

Table 1: Solutions example

Solution	Objective 1	Objective 2
A	5	4
B	3	4
C	4	5
D	6	3

In this example:

- A dominates B
- There is no domination between A and C
- There is no domination between A and D
- C dominates B
- There is no domination between B and D
- There is no domination between C and D

We then group the solutions that are not dominated in the Front Pareto F_1. In this example:

$$\mathrm{F}_1 = \{A, C, D\}$$

The second Front Pareto F_2 contains the solutions that are only dominated by F_1 , here:

$$\mathrm{F}_2 = \{B\}$$

Step 5- Calculate the crowding distance of each solution

To do this, we use the Front Pareto groups we created earlier.

For each Front Pareto group and each objective, sort the solutions in ascending order. Let's call d_i the crowding distance of a solution i. If i is an extreme solution (in the example above, for example, with the smallest value for instance), then

$$d_i = \infty$$

else

$$d_i = \sum_{k=0}^{M} \frac{f_{k,i+1} - f_{k,i-1}}{f_{k,max} - f_{k,min}}$$

With M the number of objectives, $f_{k,i+1}$ and $f_{k,i-1}$the values of the neighbours of i in the objective k, $f_{k,max}$ and $f_{k,min}$ the maximum and minimum values in the objective k.

Using the example above, for $\mathrm{F}_1 = \{A, C, D\}$:

- Sorting for objective 1: C,A,D

- Sorting for objective 2: D,A,C

Because C and D are respectively the smallest value for the objective 1 and 2, we have:

$$d_C = d_D = \infty$$

And for the solution A:

$$d_A = \frac{f_{obj1,D} - f_{obj1,C}}{f_{obj1,D} - f_{obj1,C}} + \frac{f_{obj2,C} - f_{obj2,D}}{f_{obj2,C} - f_{obj2,D}} = 2$$

For $F_2 = \{B\}$, $d_B = \infty$

Step 6- Select the solutions with the least crowded distance in P_0

Step 7- Perform cross-over and mutation operators to create offsprings O_0 of size N

Step 8- Generation count i= 0

Step 9- Combine P_i and O_i to get R_i of size 2N

Step 10- Rank and calculate crowding distance of the population R_i for non-domination

Step 11- Sort R_i according to non-domination rank

Step 12- Fill new population P_i with the sorted population R_i while $P_i < N$

Step 13- Sort the remaining population R_i by crowding distance

Step 14- Include the most widely spread solution of R_i to P_i

Step 15- Perform cross-over and mutation operators to P_i to produce offsprings O_{i+1} of size N

Step 16- Display the offspring individuals, here the solutions of P_i, to the user and ask them for an evaluation

Step 17- Calculate the fitness for what can be done quantitatively, for example the cost of the offspring.

Step 18- If the user asks for termination or if the number of generations is reached, end the program. Otherwise, $i = i + 1$, go to **Step 9**.

2 Applications

2.1 Art and Design

Fields such as art and design typically don't work well with the world of algorithms and computation. In fact, they represent one of the most difficult parts of the human intelligence to reproduce as they involve creative choices that cannot be easily explained and are often done in a highly subjective and instinctive manner. Interactive Evolutionary Computation can be used in these exact situations, where human input is necessary to understand what makes a piece of art attractive or vice versa. IEC will, after multiple interactions with the user, get a better understanding of a general aesthetical ideal.

This is especially true with dancing. A human will dance instinctively to the music he hears without knowing why these movements are the proper ones for this type of music.

The idea of making a humanoid robot that "learns some associations between music and movements through an internal mechanism of evaluation, and by means of a genetic algorithm"[5] was proposed by Adriano Manfré, Agnese Augello, Giovanni Pilato, Filippo Vella and Ignazio Infantino. For the idea of making a robot dance aesthetically to work, music is first categorized thanks to objective data such as speed (Beats Per Minutes) and intensity. According to this, a dance is generated using a mix of both long-term memory (saved passed actions and movements for evolutionary learning) and the assessment process (internal and external fitness types). The internal one represents an opinion based on given models, and the external one represents the verdict of an audience. Using this method, the humanoid robot learns by saving the most appreciated dances. Without this external assessment process, the robot wouldn't be encouraged to be innovative and would struggle to improve this fast.

Music generation could also be done using the same techniques. As Yan Pei states in its research: "IEC provides an implementation pathway to create music compositions and sound synthesis that are more human-like."[6]. Music can be judged in an indirect way thanks to natural body reactions such as the changing heart rate. This method could be used to create user-tailored music recommendations in the context of a music-streaming service. Everyone could get a different experience thanks to preferences given implicitly (skip a song means disliking it and on the contrary, listening to it until the end means liking it). These represent emotion-driven preferences that are impossible to replicate without IEC.

2.2 Other Professional Use Cases

Everything that requires (or is based on) user feedback is a good candidate for using IEC. For example, user interfaces[7] can be upgraded by showing different types to the users and let them decide which they instinctively prefer. The same came be applied for recruitment[7]: teaching the algorithm to judge applicant files automatically. Finally, in the medical[7] field, hearing aids can be programmatically adapted for each patient.

3 Challenges

3.1 The user fatigue challenge

In 2019, two Japanese scientists, Makoto Fukumoto and Yoshiko Hanada [3], investigate the efficiency of IEC and how asking humans for multiple ratings could affect potential biases in the solution and feedback delay.

To do this, they used a system of Interactive Differential Evolution, also known as IDE, which generates sign sounds. The aim of this algorithm was to design "a 'bright' sign sound suited to each user's feeling."[3].

The experiment was then divided into two study groups. Each group had to make thirty evaluations, giving a score between 1 and 7 for each new solution.
However, the first group had to complete all the ratings in one go, or one day, while the second group had to spread the ratings over two days. The researchers then tried to record both the scores and the time taken for each person in each group so that they could be compared. The scores recovered were those of the 0th, 14th and final iteration.

Surprisingly, the scientists found the same trends in both groups, both in terms of the algorithm's search space and the reduction in the number of candidate solution modifications. Similarly, the fitness function increased almost identically in both groups. However, as expected, there was a difference in the evaluation time between the first and second groups, with the second group being faster.

3.2 Impact of the user fatigue

This study simply shows that the main problem with IEC is user fatigue. This fatigue can easily occur, especially with large populations or large numbers of generations, as was the case in the study above. It then affects the user's feedback delay by reducing the speed with which the user will evaluate and provide feedback.

Finally, all this has an impact on the evaluation capacity of IECs. In fact, we can't evaluate the same number of solutions with a computer, which can evaluate thousands of them, and with a human, who usually evaluates between 10 and 20 generations. It is for this reason that new algorithms are attempting to combine qualitative and quantitative fitness, reducing the number of evaluations by the user while allowing a large search space [1].

These challenges give rise to other problems such as the notion of user bias or scalability.

Another issue is the choice of interface to enable the user to judge a solution. Users will not give the same feedback if, in order to evaluate a sound, they are shown musical notes on the one hand and the sound on the other.

4 Evolution

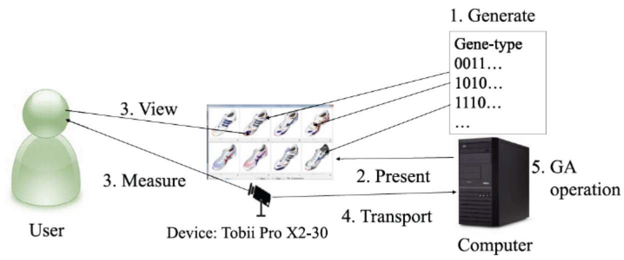
4.1 Integrating Other Technologies

Large language models (short, LLM) such as *GPT-4* (Open AI) appeared to the public quite recently. They are a step forward to create more human-like behaviours as they make understanding and generating text possible. This can be used alongside with interactive evolutionary computation to upgrade certain processes. One of them is the understanding of the human feedback[8]. Indeed, it could be used to let the user freely express their opinion on the given material to assess, instead of relying on binary input fields or scales.

The interaction would be more natural and help the user give more details and emotions. The LLM will optimize the given feedback and extract the key improvements to be made.
LLMs could also be used not only to understand the user input, but to generate natural questions as well, making them less neutral and more human-like. Users are better motivated when the interaction feels less generic.

Another technology that could be integrated into the process could be the use of indirect feedback methods, such as eye tracking. It was indeed researched by Hiroshi Takenouchi and Masataka Tokumaru in 2018[9]. It is used in the context of overcoming user fatigue by reducing manual input requests while avoiding the use of additional devices for the user to wear. The presented "E-IEC" method shows the user successive panels with eight candidate solutions each. The appreciation for one candidate solution is measured in the number of gazes on it in relation to the others presented.

Fig. 2. "Outline of the E-IEC system" taken from [9]



4.2 New User Interactions

As stated previously, one of the main challenges of IEC is the user fatigue. Overcoming the issue could be done thanks to diverse methods. Indirect feedback (using eye tracking for example) is one solution, as the user does not have to manually enter its feedback, and it is less prone to change with time. Another solution would be user collaboration: users could work together in assessing content which would make them more efficient longer.

IEC can also be used to create new user experiences. It was used by Patrikk D. Sørensen, Jeppeh M. Olsen and Sebastian Risi to make an NPC move in the *Super Mario Bros.* video game. In fact, users could choose the movements they preferred for Mario to perform. These inputs were used in the character's controlling neural network that lead to diverse movements performed programmatically. Users thought the process was enjoyable and engaging[10]. This approach could be extended to other games, allowing players to design NPC behaviours in a creative and intuitive way.

References

1. Brintrup, A.M., Takagi, H., Ramsden, J.: Evaluation of Sequential, Multi-objective, and Parallel Interactive Genetic Algorithms for Multi-objective Floor Plan Optimisation. In: 9th International Proceedings on Genetic Algorithms, pp. 1–2. Cranfield University, UK (2007).
2. Xu, Qi, Hubin Liu, Yulong Liu, and Shan Wu. "Innovative Design of Intangible Cultural Heritage Elements in Fashion Design Based on Interactive Evolutionary Computation." Mathematical Problems in Engineering. Volume 2021, Article ID 9913161, 11 pages. https://doi.org/10.1155/2021/9913161.
3. Fukumoto, Makoto, and Yoshiko Hanada. "Investigation of the Efficiency of Continuous Evaluation-Based Interactive Evolutionary Computation for Composing Melody." *IEEJ Transactions on Electrical and Electronic Engineering* 15 (2020): 235–241. https://doi.org/10.1002/tee.23050.
4. Deb K., Agrawal S., Pratap A., and Meyarivan T.: A fast elitist non dominated sorting genetic algorithm for multi objective optimization: NSGA-2, Proc. Parallel Problem Solving from Nature (PPSN2000), Paris, France (2000) 858-862
5. Manfré, Adriano, Agnese Augello, Giovanni Pilato, Filippo Vella, and Ignazio Infantino. 'Exploiting Interactive Genetic Algorithms for Creative Humanoid Dancing'. *Biologically Inspired Cognitive Architectures* 17 (1 July 2016): 12–21. https://doi.org/10.1016/j.bica.2016.07.004.
6. Pei, Yan. "A Comprehensive and Brief Survey on Interactive Evolutionary Computation in Sound and Music Composition for Algorithmic Auditory and Acoustic Design with Human-in-the-Loop". In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 1990-96. Lisbon Portugal: ACM, 2023. https://doi.org/10.1145/3583133.3596301.
7. Wang, Yanan, and Yan Pei. « A Comprehensive Survey on Interactive Evolutionary Computation in the First Two Decades of the 21st Century ». *Applied Soft Computing* 164 (octobre 2024): 111950. https://doi.org/10.1016/j.asoc.2024.111950.
8. Wu, Xingyu, Sheng-hao Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. 'Evolutionary Computation in the Era of Large Language Model: Survey and Roadmap'. arXiv, 29 May 2024. https://doi.org/10.48550/arXiv.2401.10034.
9. Takenouchi, Hiroshi, and Masataka Tokumaru. 'Interactive Evolutionary Computation System with User Gaze Information'. *International Journal of Affective Engineering* 18, no. 3 (2019): 109–16. https://doi.org/10.5057/ijae.IJAE-D-18-00026.
10. Sorensen, Patrikk D., Jeppeh M. Olsen, and Sebastian Risi. 'Breeding a Diversity of Super Mario Behaviors through Interactive Evolution'. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–7. Santorini, Greece: IEEE, 2016. https://doi.org/10.1109/CIG.2016.7860436.