EvoChef: Show me What to Cook! Artificial Evolution of Culinary Arts

Hajira Jabeen[1], Nargis Tahara[1], and Jens Lehmann[1],[2]

[1] Informatik III, University of Bonn, Germany
jabeen@cs.uni-bonn.de nargistahara@gmail.com
[2] Fraunhofer IAIS, Germany
{firstname.lastname}@iais.fraunhofer.de

Abstract. Computational Intelligence (CI) has proven its artistry in creation of music, graphics, and drawings. EvoChef demonstrates the creativity of CI in artificial evolution of culinary arts. EvoChef takes input from well-rated recipes of different cuisines and evolves new recipes by recombining the instructions, spices, and ingredients. Each recipe is represented as a property graph containing ingredients, their status, spices, and cooking instructions. These recipes are evolved using recombination and mutation operators. The expert opinion (user ratings) has been used as the fitness function for the evolved recipes. It was observed that the overall fitness of the recipes improved with the number of generations and almost all the resulting recipes were found to be conceptually correct. We also conducted a blind-comparison of the original recipes with the EvoChef recipes and the EvoChef was rated to be more innovative. To the best of our knowledge, EvoChef is the first semi-automated, open source, and valid recipe generator that creates easy to follow, and novel recipes.

Keywords: Recipe · evolutionary algorithms · culinary art · genetic algorithm.

1 Introduction

Culinary art presents itself as an attractive example of a complex art that combines ingredients and spices, blending them into amazing flavours by applying a variety of cooking methods like baking, grilling or frying. It is considered an intricate combination of art and science that handles flavour, texture, nutrients, aroma, and results in a recipe that is edible, healthy and presentable, simultaneously. Over the years, human experts (chefs) have proudly produced wonderful recipes that use novel ingredients, methods, and spices resulting in culinary masterpieces.

Individual cultures have unique preferences for the use of ingredients, spices, cooking methods and their combinations to build a recipe [1]. Different cultures value diverse outlooks on food, such as texture, taste or healthiness, which impacts their cuisine.

Most of the recipes available on the internet share region or culture-specific cuisines that follow the concept of good food of the person who introduces the recipe. Therefore, we miss the unlimited number of possible patterns that can be formed by mixing different cooking techniques, ingredients and spices from different cultures. This innovation remains limited owing to the unlimited number of possible combinations of the ingredients and spices, and availability of these ingredients in different parts of the world.

The role of intelligence in arts or creativity of machines has been a subject of discussion since long. Computational Intelligence (CI) is successfully playing a prominent role in many applications of creative arts. This includes generation of new music [2], compositions [3], art [4] and graphics [5]. A few approaches (e.g. IBM Chef-watson [6,7], Edamam [8], Covercheese [9]) have tried to automatically create or evolve recipes but they have remained limited.

In this paper, we aim to bridge this research-gap by using CI to create, intermix, evolve and optimize the recipes taken from different cuisines. We hypothesize-and-test, "Can Computational Intelligence make it possible to intermix the regional foods from different cultures to create novel works-of-art that are similar yet different from the original recipes?".

We propose "EvoChef", that gathers recipes from multiple cuisines and evolves them to create new recipes with the objective of feasibility and novelty of the resulting recipes. The main contributions of the paper are as follows:

1. An open source software to create and evolve novel recipes
2. Encoding of recipes as trees that can be evolved using genetic algorithm
3. A first proof of concept that combines different regional cosines
4. The resulting recipes:
   (a) Are understandable
   (b) Can be cooked and eaten
   (c) Use common ingredients
5. A system that can be easily extended to evolve novel recipes

To the best of our knowledge, Evochef is the first open source[3], robust and generic culinary artist that produces new recipes that are novel, valid, edible and comprehensive enough to be followed and cooked.

This paper is organized as follows. In section 2, we discuss previous research related to recipe generation. In sections 3 and 4, we present the working of EvoChef and results. Section 5 wraps up the discussion, results, and presents future perspectives of this work.

1.1 Preliminaries

Genetic Algorithms Genetic Algorithms (GA) [10] are heuristics built upon Darwin's idea of natural selection and survival of the fittest for evolution. GAs mimic evolving a population of random solutions to an optimization problem. These solutions, having abstract representations (called chromosomes or the

[3] https://github.com/SmartDataAnalytics/EvoChef

genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes), are evolved for a certain number of generations by exploiting the information from the fitness of all individuals and exploring the search space using recombination operators. The chromosome or genotype of the solution must be carefully designed to represent a viable possible solution, that can undergo recombination operators (mutation or crossover) and has ability to explore the search space for the optimal solution. We briefly cover a few important concepts of GA below:
1. **Initial Population**: The initial population is created randomly from the solution space .
2. **Fitness**: This is the objective function to optimize. It determines the quality of an individual and helps in estimating the probability of being selected as a parent.
3. **Selection**: The process used to select the two (relatively fit) parents for recombination. The selection mechanism mostly takes the fitness of individuals into account.
4. **Crossover**: This operator combines chromosomes of two parent solutions to create two new offspring solutions
5. **Mutation**: This operator injects new traits into a chromosome by randomly updating a portion of the selected chromosome.
6. **Termination**: The criteria to stop the evolution process e.g. the number of generations(iterations), or desired (average/maximum) fitness.
The pseudo-code of GA is detailed in Algorithm 1.

Algorithm 1: Genetic Algorithms

```
Result: Best solution
Optimization problem
START
Generate the initial population
Compute fitness of the population
while termination condition do
    Selection
    Crossover
    Mutation
    Compute Fitness
STOP
```

We have chosen to work with GA in this work, as it has been proven efficient [11] in searching for complex search spaces while maintaining the population diversity.

GraphX GraphX [12,13] is a graph processing library of Apache Spark [14] for graph parallel computation. GraphX supports graph abstraction of a directed graph with properties attached to its vertices and edges. Each vertex in the graph is identified by a unique identifier called VertexId. The edges are represented by the identifiers of their source and destination vertex. GraphX provides a generic interface to work with properts graphs. We have used property graph to represent recipes as trees. The nodes in this graph represent steps and properties represent the type, names, quantities or status. The fact that this additional information can be a part of the node has made easier to perform recombination operators. We detail this information in the later sections when we discuss the recombination operators.

2 Related Work

The use of CI in culinary arts has remained limited. This is partially due to the complexity and multimodality inherent in recipe generation. There have been a few attempts to develop computer-generated culinary arts but they have remained far from being optimal. One of the most prominent efforts was by IBM [15], named IBM Chef-Watson [6] presented in 2014. IBM built a system that produced novel recipes by introducing new ingredients in existing recipes from bon-appetit [**?**], and other resources. Their constrained system provided an interface, where the user could select the dish, the cuisine type, and the ingredients. Based on these choices, the system showed new recipes to the users. The quality of the recipe was determined based on its novelty and aroma. The novelty was measured as the deviation from the common recipes, and the aroma was measured by evaluating the chemical properties of flavour molecules used in the recipe [16]. Despite the imagined level of effort involved in IBM-research, the Chef-Watson was unable to produce valid recipes; The instructions were incomplete or incomprehensible and the quantities of the ingredients were missing. The recipes were taken from bon-appetite, and therefore, were mainly western cuisine recipes. It focused novelty using chemical combinations of the ingredients and therefore, often resulted in hard to find ingredients. Overall IBM Chef Watson lacked a few key elements for regular cooking. The technical details of the working of Chef Watson were not formally published. One of the user reviews states the combination of ingredients suggested by Watson was so unfamiliar and there were many mistakes in the recipes[4]. This project seems to be canned now, and the web page[5] is no longer available.

Evolutionary Meal Management Algorithm (EMMA) [9] is another attempt to generate the recipes using machine learning algorithms. EMMA can automatically generate recipes for new food items and improve them based on users feedback about their taste. In the early versions of EMMA, the system was unable to detect edible ingredients and produced inedible recipes e.g. *"1 cup skillet, chop skillet and serve"*. The later generations, look a little better but they are very basic and often miss clear instructions or quantities. Moreover, the work

[4] https://daljiblog.wordpress.com/2016/11/30/how-chef-watson-calculates-deliciousness/
[5] http://www.ibmchefwatson.com

seems experimental and there are no published papers about the underlying approach.

Erol et al. [17] have developed an approach to discover novel ingredient combinations for the salads. The system is designed in two steps: First, a statistical model is constructed to rank recipes using a deterministic classifier that, given two recipes, predicts which one is better. Second, they experimented with various search algorithms to explore the salad recipe space and discover novel ingredient combinations. Like Chef-Watson; their focus is also on novelty. In the end, the algorithm only suggests ingredients and ignores any instructions or the quantities of ingredients, which is one of the crucial elements of any recipe.

Most of the above-mentioned approaches do not always present valid recipes to their users and focus mostly on novelty by providing good food pairing. They do not emphasise easy to follow recipes, easy ingredients, or regional cuisines. On the contrary, our work attempts to cover novelty, cultures, usability, and comprehensiveness of resulting recipes.

Table 1. Components of a recipe

Component	Description
Main ingredient	The main ingredient is a major ingredient of the recipe. e.g. in all types of rice dishes, rice is the main ingredient.
Side ingredient	Side ingredients are the remaining ingredients of the recipe minus the main ingredient.
Spices	A spice is a type of side ingredient used to add flavour to the ingredients in the recipe.
Steps	These are the cooking instructions.
Pre-processing	A step required before actual cooking. For example, to fry onions, we first need to peel and cut them into chunks.
Post-processing	The post-processing steps mix the individual ingredients together and do some processing e.g. bake, fry or saute.

3 Proposed Methodology

3.1 Preliminaries

Components of a recipe We have represented the recipes such that each recipe contains a main ingredient, some side ingredients, spices, and instructions for cooking that are named as steps. A recipe can also have some pre-processing or post-processing steps. Pre-processing steps are performed on ingredients before mixing and post-processing is performed after mixing the ingredients. Table 1 explains the recipe structure in more detail.

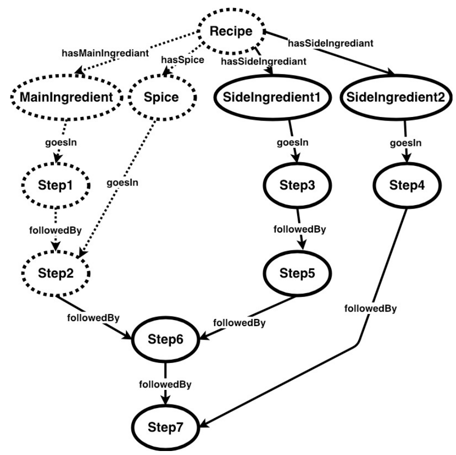

Fig. 1. Graph representation of an arbitrary recipe.

Data Collection and Cleaning In order to achieve diversity in our initial population, we have collected our recipes from different cooking websites including Yummly.com [18], Allrecipes.com [19], Recipes-plus.com [20], Geniuskitchen.com [21], Simplyrecipes.com [22], Omnivorescookbook.com [23] and Greenevi.com [24]. Our recipe data consists of Southern, American, Italian, Spanish, Hungarian and Chinese recipes.

Due to the fact that the recipes are written by different users of these popular web pages, they do not possess a regular format or consistent structure. e.g. the pre-processing steps for the ingredients are either written in the instructions or mentioned besides the ingredient name. Therefore, we have cleaned the data to make it feasible with our proposed recipe structure. The following steps are performed on each recipe to make it ready to be ingested by EvoChef:

1. Multiple quantity options were converted to a single quantity. For example, if there is "3-4 potatoes" in the ingredient list, it is converted to either "3 potatoes" or "4 potatoes".
2. Properties of ingredients were extracted and assigned to each ingredient accordingly. Properties include ingredient name, quantity, measurement unit, ingredient type (main ingredient/side ingredient) and usedIn (whether the ingredient is used in the main-process or side-process).
3. If pre-processing steps are along with ingredients, they were separated and added to the instructions. For example, if "4 potatoes (cut into small chunks)" is in the ingredient list. We have removed the "cut into small chunks" from the ingredient list and added this as another step in the instructions.
4. Each step in the instruction is assigned the property usedIn (whether the step is part of the main process or side process).
5. The root node, recipe, has the properties: recipeName, the totalTime required to cook the recipe, numberOfServings, category of the recipe (Main dish/Side dish) and the state of the main ingredient after the main process has been applied (e.g. raw/half cooked/cooked).

Considering a fully automated, NLP based, approach beyond the scope of this work, we have adopted a semi-automated approach to curate and clean the recipe data.

Table 2. Edges/Relations properties

Properties	Description
hasSideIngredient	Domain: Recipe Range: Ingredient Description: The recipe has a side ingredient
hasMainIngredient	Domain: Recipe Range: Ingredient Description: The recipe has a main ingredient
hasSpice	Domain: Recipe Range: Ingredient Description: Recipe has a spice
followedBy	Domain: step, post-step Range: step, post-step, last-step Description: A step is followed by another step
goesIn	Domain: main Ingredient, side ingredient, spice Range: Step Description: The ingredient or spice goes in (added to) a step

3.2 Evolution of the Recipes

This section describes the overall evolution process and representation decision taken in this work.

Solution Encoding A recipe contains a set of ingredients, spices and cooking steps. We have used the property graph of GraphX to encode recipes, where vertices in the graph represent important components of the recipes (e.g. ingredients, spices and method) that make up the textual representation of a recipe, and edges define relations (e.g. hasSpice, goesIn etc.) between them. This representation corresponds to the genotype encoding to ensure credible evolution. The cooking process of a recipe can be divided into two sub-processes; a main process and a side process. The main process contains the cooking steps of the main ingredient that are required prior to mixing with the rest of the ingredients. The side process represents the pre-processing and cooking steps of the side ingredients that are later to be mixed with the main process. In Fig. 1, an example recipe is shown with one main ingredient, two side ingredients and one spice. The nodes and edges represented with the dotted line below the root show the main process. The right hand side of the root with solid line represents the side ingredients and the process. Table 3 and 2 describe the properties of vertices and edges of the graph respectively.

Initial Population The initial population is generated by randomly intermixing (crossover) the recipes taken from the recipe pages. This was done to avoid the first generation to have the best fitness, as we selected the best rated recipes.

Fitness Evaluation The challenging part of our work is the evaluation of the fitness of a recipe. Human experts can best judge a recipe by either tasting or even examining the description of the recipe. We listed the recipes generated by our system on a dedicated website[6] and gathered user ratings (friends, colleagues, and research partners) from a range of users as 1-5 stars. This fitness-rating is used as the fitness function by EvoChef.

Selection We have used tournament selection as the parent selection mechanism. To produce a child, we run the tournament two times for each parent. Each parent goes through a compatibility check before performing the recombination operator. The compatibility check tests if "cookedType_mainIngredient" property of both parent recipes is cooked, to ensure that the children produced are valid and we are not mixing cooked ingredients with uncooked ones, this might result in an uncooked ingredient.

Crossover We have used the fixed point crossover operator in EvoChef. The main and side processes of the parents are swapped to create two children with different valid recipes. Table 4 shows two parent recipes.
Parent recipe#01 has one main ingredient and four side ingredients. The main process of the recipe is represented in italic text while rest of the text shows the side process of the recipe. Fig. 2 shows the graph representation of the first recipe.The edge between nodes and steps show the ingredients of the step. The Parent recipe#02 is represented as a graph in Fig. 3. Now we have two parents recipes (phenotypes) encoded as graphs (genotypes). We can apply crossover and mutation on these graphs to generate new off springs. Table 5 shows the children phenotypes after recombination of the parent recipes from Table 4. Fig. 4 shows the genotypes of child_01_02. Fig. 5 is graph representation of second child child_02_01. Out of the two children, the fittest, is returned to be added to the following generation.

Generation of recipe names The child recipe name is generated from its parents. If there is any side ingredient name from one parent recipe, it is replaced with the name of the side ingredients from the second parent. Following this, in child_01_02, recipe name from recipe#01 will be taken and if there is the name of any side ingredient from recipe#01 in the name of recipe; it will be replaced with the side ingredient of recipe#02, as in new recipe main ingredient of recipes#01 and side ingredients if recipe#02 are being used.

```
Name of recipe#01: Tibetan potato curry
Name of recipe#02: Glazed sweet potatoes with brown sugar
Name of child_01_02: Tibetan potato curry with brown sugar
Name of child_02_01: Glazed potatoes
```

Listing 1.1. Child name generation

[6] http://www.machinegeneratedrecipes.de

Table 3. Vertex Properties

	Recipe properties
Property	Description
recipeName	Name of the recipe.
cookTime	Time required to cook recipe.
servings	To how many people the recipe serves.
category	Category of the recipe. For example, main dish, side dish, dessert etc.
cookedType_mainIngredient	Whether the main ingredient is raw, half cooked or cooked after the main process is finished. In other words, it is the condition of the main ingredient before it is going to be added with other side ingredients.
	Ingredient properties
Property	Description
ingredientName	Name of the ingredient.
quantity	Quantity of the ingredient.
measurementUnit	Measurement Unit of the ingredient, for example, kg, cups, tablespoon etc.
ingredientType	Main ingredient, side ingredient or spice.
usedIn	Whether the ingredient is the part of main process or side process.
	Steps properties
Property	Description
description	Cooking description of the step, The step can also be a post-step or last-step.
usedIn	Whether the step is a part of the main process or side process.

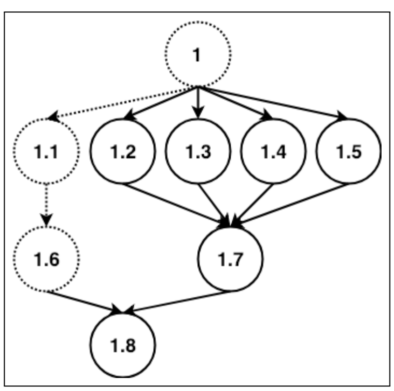

Fig. 2. Graph representation of parent recipe#01

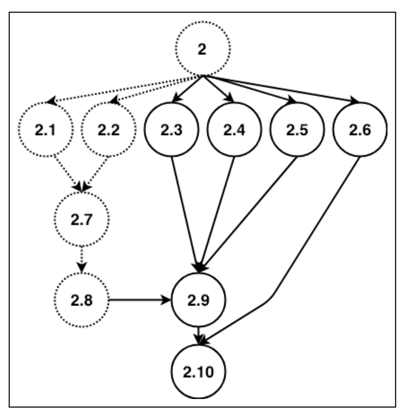

Fig. 3. Graph representation of parent recipe#02

Table 4. Parent recipes

Parent recipe#01	Parent recipe#02
Glazed Sweet Potatoes with Brown Sugar [25] **Ingredients:** brown sugar, water, butter, salt, *sweet potatoes* **Instruction:** 1. *Peel the sweet potatoes and cut them into 0.5 inch to 1-inch thick slices. Place the sweet potato slices in a saucepan and cover with water. Bring to a boil and cook for about 12 minutes, or until just tender.* 2. In a heavy skillet, combine brown sugar, water, butter, and salt. Simmer over low heat for 5 minutes. 3. Add the sliced sweet potatoes to the brown sugar mixture. Simmer for 10 minutes, or until well glazed, turning frequently to keep them from scorching.	**Mashed Red Potatoes With Garlic And Parmesan [26]** **Ingredients:** *red potatoes, garlic cloves,* butter, milk, salt, parmesan cheese **Instruction:** 1. *Put potatoes and garlic in a large pan. Cover with water. Bring to a boil.* *2. Reduce heat and simmer for 25 minutes, until potatoes are tender. Drain well.* 3. Mash with the butter, milk, and salt 4. Stir in the parmesan cheese.

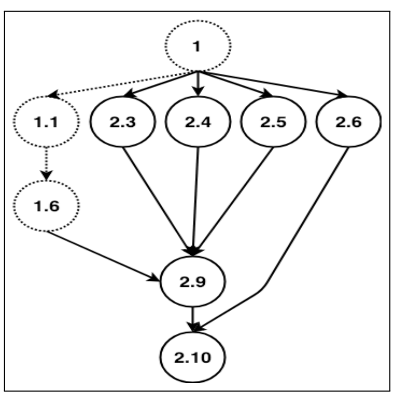

Fig. 4. Graph representation of Child_01_02

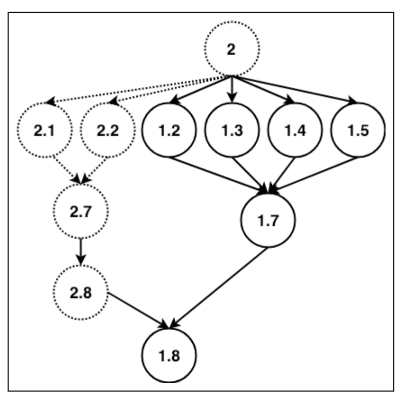

Fig. 5. Graph representation of Child_02_01

Table 5. Child Recipes

Child recipe_01_02	Child recipe_02_01
Glazed Sweet Potatoes with parmesan cheese **Ingredients:** butter, milk, salt, parmesan cheese, *sweet potatoes* **Instruction:** 1. *Peel the sweet potatoes and cut them into 0.5 inch to 1-inch thick slices. Place the sweet potato slices in a saucepan and cover with water. Bring to a boil and cook for about 12 minutes, or until just tender.* 2. Mash with the butter, milk, and salt. 3. Stir in the parmesan cheese.	**Mashed Red Potatoes With brown sugar And butter** **Ingredients:** *red potatoes, garlic cloves,* brown sugar, water, butter, salt **Instruction:** 1. *Put potatoes and garlic in lg pan. Cover with water. Bring to a boil.* *2. Reduce heat and simmer for 25 minutes, until potatoes are tender. Drain well.* 3. In a heavy skillet, combine brown sugar, water, butter, and salt. Simmer over low heat for 5 minutes. 4. Add the sliced potatoes to the brown sugar mixture. Simmer for 10 minutes, or until well glazed, turning frequently to keep them from scorching.

These new names remain compatible with the content of the new recipe. If there are no side ingredients in the first parent, the side ingredients from the second recipe are added at the end of the recipe name with propositions "with" or "and". An example is shown in Listing 1.1.

Name generalization At the end of the process of offspring generation, the name of the main ingredient in the cooking instructions of the recipe is generalized to avoid confusion in the new recipe. For example, in our case, different kinds of potatoes are used in different recipes including Red Potatoes, Russet Potatoes, sweet potatoes, etc. As they are added in another recipe that might be referring to the potatoes with another name. To avoid confusion, all type of potatoes is replaced with "potatoes" in the cooking instructions. As shown in Table 5, in step 4 of Child_02_01, sweet potato is replaced with the potato.

Mutation In order to introduce diversity through mutation, we have developed an ingredient replacement table shown in Table 6. This is a simplified version of the network provided by food-network [27]. An ingredient of the selected recipe is picked randomly and replaced by its substitution from the table. This type of mutation does not make any changes in the cooking method itself, but is still able to introduce novelty through replacing the ingredients. Table 7 represents an example where butter in the recipe is replaced with margarine in the given recipe.

Table 6. Sample ingredient substitute table

Ingredient	Substitute
Arrowroot starch	flour, cornstarch
Baking mix	pancake mix, Biscuit Mixture
Beer	nonalcoholic beer, chicken broth
Bread crumbs	cracker crumbs, matzo meal, ground oats
Butter	margarine, shortening, vegetable oil, lard
Buttermilk	yogurt
Cheddar cheese	shredded Colby cheddar, shredded Monterey Jack cheese
Chervil	chopped fresh parsley
Chicken base	chicken broth, chicken stock
Cocoa	unsweetened chocolate
Cottage cheese	farmer's cheese, ricotta cheese
Egg	silken tofu pureed, mayonnaise
Evaporated milk	light cream
Garlic	garlic powder, granulated garlic
Honey	corn syrup, light treacle syrup
Lemon juice	vinegar, white wine, lime juice
Onion	green onions, shallots, leek

Table 7. Mutation by using ingredient substitution

Original recipe	Recipe after mutation
Glazed Sweet Potatoes with parmesan cheese **Ingredients:** ~~*butter*~~, milk, salt, parmesan cheese, sweet potatoes **Instruction:** 1. Peel the sweet potatoes and cut them into 0.5 inch to 1-inch thick slices. Place the sweet potato slices in a saucepan and cover with water. Bring to a boil and cook for about 12 minutes, or until just tender. 2. Mash with the ~~*butter*~~, milk, and salt. 3. Stir in the parmesan cheese.	**Glazed Sweet Potatoes with parmesan cheese** **Ingredients:** *margarine*, milk, salt, parmesan cheese, sweet potatoes **Instruction:** 1. Peel the sweet potatoes and cut them into 0.5 inch to 1-inch thick slices. Place the sweet potato slices in a saucepan and cover with water. Bring to a boil and cook for about 12 minutes, or until just tender. 2. Mash with the *margarine*, milk, and salt. 3. Stir in the parmesan cheese.

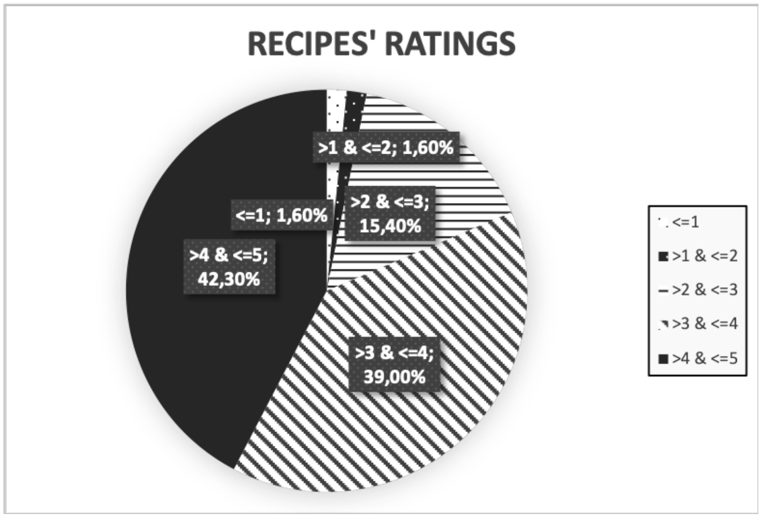

Fig. 6. Recipe ratings in the initial generation

4 Evaluation

Due to limited available data and dependence of our fitness function on expert input, our experiments have mainly remained limited, but they have yielded promising results nonetheless. The initial population size is eight. with 0.9 crossover rate and 0.1 mutation rate. We evolve the GA until 85% of the population has acheived the fitness rating above 4, with 5 being the highest and 0 being the lowest fitness-rating.

4.1 Generated Recipes

While we have presented some child recipes in the earlier sections. All the generated recipes are listed on our webpage[7]. We have gathered fitness-ratings and comparison scores for our recipes from the users. These users were invited over social media channels, friends, colleagues and research partners.

4.2 Recipe Ratings (fitness)

This section overviews the fitness-ratings of the evolving recipes. Fig. 6 shows the overall fitness of the initial population. The fitness of the first three generations is shown in Fig. 7 and it can be observed that the average fitness of the population successfully improved with the number of generations.

Evochef produced a total of 123 recipes from all the possible combinations of recipes in the initial population, from which more than 80% recipes have

[7] www.machinegeneratedrecipes.de

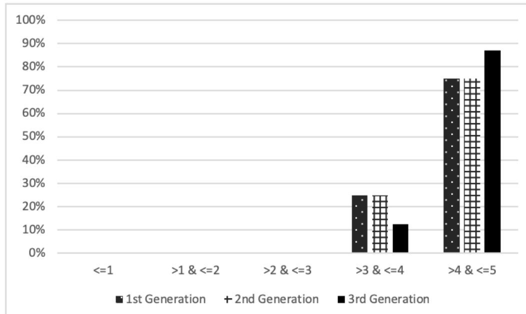

Fig. 7. Child population ratings in 1st, 2nd and 3rd generation

good rating (more than 3 out of 5) and 42% recipes have an excellent rating (more than 4 out of 5). 7% of the produced child were rejected by compatibility function hence dropped.

4.3 Novelty

To compare the novelty in our newly generated recipes, we made a blind comparison of recipes. We listed a randomly selected original recipe with an EvoChef recipe, and users were asked to select a recipe with more novelty. Three of 12 comparisons rated original recipe as more novel while 8 of the new recipes received more votes as a novel recipe. One of the recipes received equal votes for original and new recipes.

Table 8. Novelty of EvoChef Recipes

Original	EvoChef	Both are equal	I do not know
36.9%	59.2%	1.90%	2.00%

4.4 Comparison with parent recipes

All the recipes selected as the input to EvoChef had the rating of '5' on their parent webpages. In order to have a fair comparison, we copied the same recipes on our page and also gathered ratings from the same users. At the end of our experiments, we found that the average ratings of parent recipes is 4 and the average rating of EvoChef recipes in the last generation is 4.5.

4.5 Comparison with other recipe generators

IBM Chef-Watson is no longer active. However, we found some reviews [28] about the recipes that "Calls for wasabi powder (never used), **shelled green**

peas (212 cup shelled green peas) **cut into 3/4" pieces**. Then placed on a barbecue.". Cover cheese [9] has recipes like "Ingredients med okra, lot sugar and Instructions: **boil: sugar okra sugar**" Or "Ingredients: Small angel_food, Small eggplant and Instructions: 1. Slice:angel_food, 2. eggplant angel_food" Both of these examples do not create a valid recipe. Erol et al [17], only targets salads and their recipes look like **"Cherry, chive, granny smith apple, mushroom, onion powder, pine nut, salsa, salt"** Their salad recipes have no instructions or the quantities of the ingredients.

5 Conclusion

Our work has investigated the possibility of using evolutionary algorithms in culinary arts to develop a system that produces valid and novel recipes. The encoding of the recipes as graphs has provided an optimal representation of recipes to undergo genetic operators like crossover and mutation. The fact that our initial recipes are highly rated (5) and are taken from popular webpages. They mostly use easy-to-find ingredients. This important feature is also reflected in the recipes generated by Evochef.

The results produced by EvoChef are complete and precise. While we have constrained ourselves to focus only potato recipes. This approach can be extended to different kinds of recipes by automatically extracting recipe-data using natural language processing approaches.

Our limited data and the bottleneck of human feedback for the evaluation forced us to stop the evolution process earlier. Machine learning approaches can be employed to predict the ratings of the child recipes automatically. Currently, we ignore some relevant features like flavor information for ingredient-pairing, nutritional information, or the texture of the recipe. Extending the approach with this information could yield healthier food. In conclusion, our preliminary work has produced interesting results in the this under-represented area or evolving culinary arts, and there is a range of possibilities to extend this work.

6 Acknowledgement

This work is partly supported by the EU Horizon2020 projects BigDataOcean (GA no. 732310), LAMBDA (GA no. 809965) and Boost4.0 (GA no. 780732).

References

1. K. Kim and C. Chung. Tell me what you eat, and i will tell you where you come from: A data science approach for global recipe data on the web. *IEEE Access*, 4:8199–8211, 2016.
2. Zaccagnino R. De Prisco R. An evolutionary music composer algorithm for bass harmonization. *In Workshops on Applications of Evolutionary Computation 2009 Apr 15 (pp. 567-572). Springer, Berlin, Heidelberg.*

3. Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. Affective evolutionary music composition with metacompose. *Genetic Programming and Evolvable Machines*, pages 1–33, 6 2017.
4. Joanna Misztal and Bipin Indurkhya. A computational approach to re-interpretation: Generation of emphatic poems inspired by internet blogs, 2014.
5. Matthew Lewis. Evolutionary visual art and design. In *The art of artificial evolution*, page 337. Springer, 2008.
6. IBM Chef Watson. http://www.ibmchefwatson.com.
7. Florian Pinel. Whats cooking with chef watson? an interview with lav varshney and james briscione. *IEEE Pervasive Computing*, 14(4):5862, 2015.
8. EDAMAM. https://www.edamam.com/.
9. Cover:Cheese. https://covercheese.appspot.com/.
10. Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1996.
11. Christine M Anderson-Cook. Practical genetic algorithms, 2005.
12. Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, and Ion Stoica. Graphx: A resilient distributed graph system on spark. In *First International Workshop on Graph Data Management Experiences and Systems*, GRADES '13, pages 2:1–2:6, New York, NY, USA, 2013. ACM.
13. GraphX Programming Guide. https://spark.apache.org/docs/latest/graphx-programming-guide.html#connectedcomponents.
14. Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, 2016.
15. IBM. https://www.ibm.com.
16. Aatish Bhatia. A New Kind of Food Science: How IBM Is Using Big Data to Invent Creative Recipes. https://www.wired.com/2013/11/a-new-kind-of-food-science/, 2013. [Online; accessed 01-march-2018].
17. Erol Cromwell, Jonah Galeota-Sprung, and Raghuram Ramanujan. Computational creativity in the culinary arts, 2015.
18. Yummly. https://www.yummly.com/.
19. Allrecipes. https://www.allrecipes.com.
20. RecipesPlus. http://recipes-plus.com/.
21. Genius Kitchen - Recipes, Food Ideas And Videos. https://www.geniuskitchen.com.
22. Simply Recipes. https://www.simplyrecipes.com/.
23. Omnivore's Cookbook. https://omnivorescookbook.com/.
24. Green Evi. http://greenevi.com.
25. Diana Rattray. Glazed Sweet Potatoes with Brown Sugar. https://www.thespruceeats.com/glazed-sweet-potatoes-with-brown-sugar-3061580?utm_campaign=yummly&utm_medium=yummly&utm_source=yummly. [Online; accessed february-2018].
26. MizzNezz. Mashed Red Potatoes With Garlic and Parmesan. http://www.geniuskitchen.com/recipe/mashed-red-potatoes-with-garlic-and-parmesan-34382#activity-feed. [Online; accessed february-2018].
27. Common Ingredient Substitutions (Infographic). http://dish.allrecipes.com/common-ingredient-substitutions/.
28. IBM-ICE. https://www.reddit.com/r/IAmA/comments/3id842/we_are_the_ibm_chef_watson_team_along_with_our/.