RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

MASTER THESIS

EvoChef! Teach me how to cook..Machine Generated Recipes Using Evolutionary Algorithm

Author:
Nargis Tahara
Matrikel-Nr.: 2954968

Supervisor:
Dr. Hajira JABEEN

A thesis submitted in fulfillment of the requirements for the degree of Master in Computer Science

in the

Smart Data Analytics Research Group
Universität Bonn

November 14, 2018

Declaration of Authorship

I, Nargis Tahara, certify that the work presented in this master thesis is the outcome of my own research. I have not used any sources other than those listed as references in the bibliography.

Signature:

Date:

Abstract

Different regions across the world use different combination of ingredients and spices and different cooking methods for the same ingredients to create a recipe. We present EvoChef, that illustrates the potential of an evolutionary algorithm to produce unique recipes by combining these individual cooking techniques. We represent each recipe in a property graph that contains ingredients, spices and cooking instructions. The starting solutions are the combination of randomly taken recipes from various internet resources that are evolved using single-point crossover operator. The rating of a recipe obtained through crowdsourcing is regarded as fitness function to evaluate the quality of a recipe. Based on this fitness function, algorithm choose the recipes from current population and recombine them to produce new recipes until we achieve the targeted quality. Our results reveal that the evolutionary algorithm mostly produces valid and novel recipes of a high quality and only a small percentage of unpleasant recipes. To the best of our knowledge, EvoChef is the first semi-automated and open source recipe generator that can output valid and novel recipes. We also conducted a blind-comparison of the original recipes with the EvoChef recipes, and the EvoChef recipes were rated to be more innovative.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Hajira Jabeen for her continuous support throughout the process of researching and writing my thesis. She always guided me to the correct direction whenever I needed it. I am very thankful to Prof. Dr. Jens Lehmann for allowing me to pursue the master thesis in the Smart Data Analytics (SDA) research group that is a part of Institute for Computer Science at the University of Bonn. Finally, I would like to give thanks to my parents and to my friends for providing me with constant support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Individual cultures develop preferences for ingredients, spices, cooking methods, and ways of combining these to build a recipe [1]. The concept of "good food" is a cultural construction. Different cultures value different aspects of food, such as texture and sweetness. Moreover, cultures have different perceptions of what a healthy diet is, which also has an impact on their cuisine. Variations in cooking techniques and ingredients reflect environmental, economic, and cultural traditions and trends. Gradually, the way people perceive certain flavors becomes programmed based on how they typically consume them. Cultural cuisines in addition to differing on the dominant ingredients (e.g., cheddar cheese, curry spices, or chilis), also reflect different opinions on what is used with what. A study [2] on flavor compounds has indicated that Western European and North American recipes generally combine ingredients that share flavor compounds. By contrast, Southern European and East Asian cuisines do not usually combine ingredients that share flavor compounds.

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, unlimited number of patterns are missed that can be formed by mixing together cross-cultural cooking techniques, spices and ingredients. Because individuals often have a specific idea of what “good food” is, they often do not discover new patterns.

Our thesis proposes an approach based on evolutionary algorithm to automate the process of discovering unusual recipes by combining recipes from different cultures and improving them in accordance with feedback of users. Evolutionary computation has proven its artistry in music compositions [3], robot navigation [4], Software Engineering, Distributed Computing and Machine Learning [5]. The creativity of evolutionary algorithms and other machine-learning algorithms in the cooking domain is limited because computers are unable to test the results without human help. The texture and the edibility of the resulting food also need to be considered, and various factors can produce unfavorable results, such as food that is not properly cooked and is therefore inedible, overcooked food cooked that tastes unpleasant, or recipes that are unappetizing due to uncontrolled quantities of spices and ingredients.

We encoded the individual recipes using property graph provided by GraphX[1] and divided them into two parts, one of which is the main process, which consists of the main ingredient and the method of cooking it, while the other part consists of the side ingredients and the instructions for cooking them. The evolutionary algorithm selects recipes as parents from the current population using the method of probabilistic selection, where recipes with high ratings have a high probability of being selected. Each two parents can produce a maximum of two children through a single fixed-point crossover, where the point that separates the main process of the recipe from the side process is the crossover point. Each child recipe is checked for

[1]https://spark.apache.org/graphx/

validity using a compatibility function. Invalid offspring are dropped immediately after being generated and valid or best children based on users' ratings are selected to be part of the next generation. This technique leads to the generation of better recipes after each iteration. Experiment results reveal that our algorithm produces surprising recipes with an extremely low percentage of low quality recipes.

The rest of the thesis is organized as follows. Chapter 2 gives an overview of the techniques we used. Chapter 3 describes related work. In Chapters 4 and 5, we discuss the methodology and results. Chapter 6 wraps up the discussion, and presents a few interesting future perspectives of this work.

Chapter 2

Technical Details

2.1 Evolutionary Algorithm

An evolutionary algorithm [6, 7] is a stochastic method that produces best quality solutions with low time complexity. An evolutionary algorithm requires less problem-specific knowledge compared to other methods and only needs the fitness function to be optimized for a given problem.

The evolutionary algorithm starts its working on a set of individuals/chromosomes that are evolved under the specific selection and reproduction rules to generate better individuals over time. Every individual is assigned a fitness value, which defines how good the individual is. Individuals are chosen based on their fitness value, and crossover and mutation operations are performed to produce offspring. In general, a evolutionary algorithm operates in the following steps:

1. Encoding
2. Selection
3. Reproduction

2.1.1 Encoding

Choosing the correct way to represent an individual is an important step in the evolutionary algorithm. The mode of representation depends on the type of problem; however, the chosen data structure must be suitable for crossover and mutation operations. The following are some commonly used encoding methods:

1. Binary Encoding

Binary encoding is the simplest encoding technique, in which an individual is represented by a fixed length string with 0 or 1 at each index. Crossover and mutation operations can be performed easily, as an index can be changed by merely flipping a bit. For a given hypothesis, 1 or 0 represents the presence or absence of a property. A hypothesis with 2^n possible solutions can be represented with a string of length n. Below are examples of chromosomes in binary encoding:

Chromosome 1: 10010
Chromosome 2: 11001

2. Permutation Encoding

In permutation encoding, each chromosome is represented by a permutation of numbers. This representation is useful for problems such as task ordering. For example,

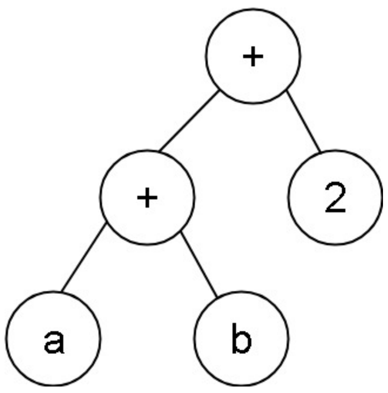


FIGURE 2.1: Tree Encoding

given a set of numbers (1,2,3,4), following are the example chromosomes:

Chromosome 1: 1342
Chromosome 2: 2413

In this type of representation, none of the numbers should appear more than once. This rule must be adhered to during mutation and crossover.

3. Value Encoding

Value encoding is used for complicated problems where it is not possible to represent chromosomes with binary strings. Each chromosome is represented by a string that contains simple strings, real numbers or complex objects. Some examples of chromosomes for value encoding are given below:

Chromosome 1: 1.34 3.89 4.55 6.98
Chromosome 2: (right), (left), (back), (left)

Chromosome 1 contains real numbers, and Chromosome 2 consists of directions. Value encoding is useful for determining weights in neural network architecture.

4. Tree Encoding

Tree encoding produces a tree of objects that are connected to each other in a specific order. Tree encoding is useful for genetic programming. Crossover and mutation can be performed by switching the tree branches and by modifying the values of nodes. A chromosome to represent an equation "(a+b) + 2" encodes to the tree shown in FIGURE 2.1.

2.1.2 Selection and Reproduction

After chromosomes are represented in a data structure, some of them are selected as parents based on their fitness values and their ability to produce children to form a new generation. This procedure is repeated until a fixed number of generations is created or until a population with the required properties is produced. A general evolutionary algorithm is shown in FIGURE 2.2.

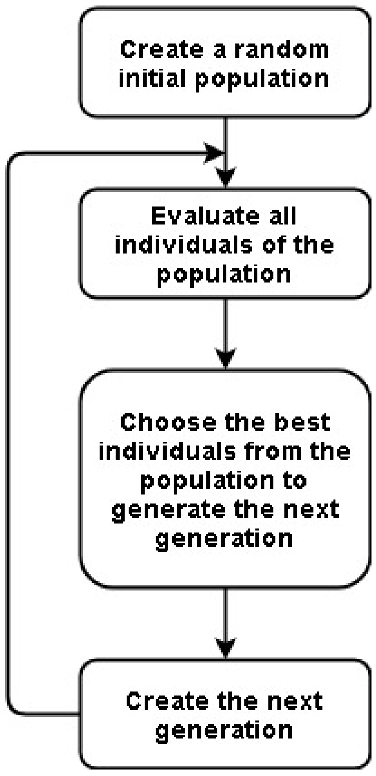

Create a random initial population
Evaluate all individuals of the population
Choose the best individuals from the population to generate the next generation
Create the next generation

FIGURE 2.2: Evolutionary Algorithm

Selection

Selection is the stage of an evolutionary algorithm in which individuals are selected from the population for later breeding. The selection method can vary according to the task to be achieved and depends on the nature of the problem and the number of individuals in the population. In general, individuals with high fitness values have a high probability of being selected for reproduction, and individuals with low fitness values have less chance of being selected.

There are several key selection techniques: The roulette wheel selection method selects individuals with the probability directly proportion to their fitness. In this method, the fitness level is used to associate the probability of selection to each individual. It can be imagined as a Roulette wheel in a casino. The large portion of the wheel is assigned to the individual with high fitness, so, they have a high chance of selection when the wheel is rotated. Tournament selection is another procedure of selection that is more flexible than roulette wheel selection. In this method, a fixed number of random individuals are chosen from the population and the individual with higher fitness is selected for the crossover. The size of the tournament is adjustable. If the size of the tournament is smaller, the week individuals get more chance of selection; as in a large poll, there is a high possibility that it contains a strong individual that will be selected for the crossover. For the tournament size 1, it is like a random selection and all individuals get the equal chance of selection regardless of the fitness. Another selection method is Truncation selection which takes a proportion of best individuals and truncates the rest.

Other than the above-mentioned methods, a custom selection method can also be used that might produce a better result depending upon the nature of the problem.

Reproduction

Reproduction is the process of producing new individuals, and it consists of crossover and mutation. Crossover intermixes the genetic properties of parents to produce an offspring. Further modification in the produced offspring is made to create diversity; this process is called mutation. There are several methods for crossover. Some of them are the following:

Single-point crossover - A random point is selected as a crossover point for the parents. The child takes one part from one parent and second part from another parent. The following is an example of a single-point crossover of two-bit strings:

11101011+11011**111** = 11101111

Two-point crossover - Two-point crossover is the same as performing single point crossover two times. Two random crossover points are selected, and the right part of these points is taken from one parent and the left part from another parent to create a child. An example of two points crossover is given below:

000010**11** + 11**0111**11 = 00011111

Uniform crossover - Each bit in the child is either the bit from the first parent on same index or from the second parent; as shown in the example below:

1**00**010**11** + **11**0**101**01 = 10010111

Arithmetic crossover - For an arithmetic crossover, an arithmetic operation, such as AND or OR, is performed between the parents to produce a child. The example below shows an AND operation performed on parents to create a child:

10001111 + 11011111 = 10001111 (AND)

Mutation

The purpose of mutation is to maintain diversity between generations of the algorithm. The mutation can be performed in several ways, e.g., by inverting a randomly selected bit in a binary string. If the chromosomes consist of real numbers, the order of two selected numbers can be changed to alter the chromosome or a positive or negative value can be added to a random number. In the case of symbols, these can be altered, e.g., "+" can be changed to "-" in an equation.

Crossover Without Mutation

Reproduction without using mutation is used when the population already has high degree of randomness. The algorithm repeatedly produces better individuals by recombining the properties of previous individuals in the population without altering any of them after the crossover. As the crossover is performed on individuals with a high value of fitness, each individual of the next generation is better than those of the previous one.

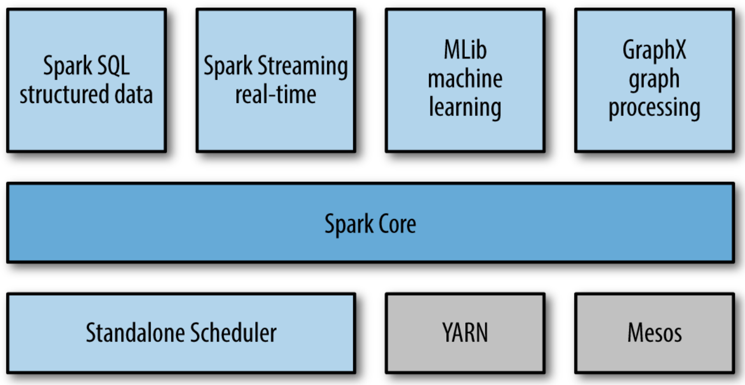


FIGURE 2.3: Components of Spark[3]

Mutation Without Crossover

An evolutionary algorithm without a crossover is also called an evolutionary strategy. Without a crossover, an algorithm acts like a randomized local search. It alters individuals in parallel and selects the best N individual for the next generation. Mutation without crossover is not suitable for most research problems as it does not use the best individuals in the current population and uses a guess and check technique instead.

2.2 Spark

Apache Spark[1,2] [8] is a powerful open-source data analytics engine for large-scale SQL, data streaming, machine learning and graph processing. It is an in-memory data processing framework that enables real-time, batch, and advanced analytics on the Apache Hadoop platform. Spark is written in Scala but provides high-level APIs in Scala, Java, R, and Python. Spark is intended to enhance the Hadoop stack and can read and write data from an existing HDFS and other storage systems. Unlike two-stage disk-based MapReduce computation engine of Hadoop, Spark's multi-stage in-memory primitives allow most computations to run in memory. This means that data can be loaded into a cluster's memory and can be queried repeatedly, which gives better performance for certain applications, such as iterative algorithms or interactive analytics. Spark runs programs up to 100 times faster than Hadoop MapReduce in memory and 10 times faster than on disk.

All the components of Spark including Spark SQL, Spark Streaming, MLibs and GraphX are build on top of the Spark Core as shown in FIGURE 2.3. Spark can run on top of Hadoop YARN, Apache Mesos, standalone or in the cloud (Amazon EC2 or IBM Bluemix). All the components of Spark are explained below:

Spark Core

Spark core is the base of the overall project and it is responsible for scheduling, distributed task dispatching, and basic Spark I/O functionalities. All other applications

[1]https://spark.apache.org
[2]https://databricks.com/spark/about
[3]`https://www.oreilly.com/library/view/learning-spark/9781449359034/ch01.html`

interact with spark core using a language interfaces such as Java, Python, Scala, or R.

Spark SQL

Spark SQL is a component added to Spark Core that provides powerful integration of SQL queries with machine learning. It can process structured and semi-structured data. It also provides an engine for Hive to run unmodified Hadoop Hive queries to run up to 100 times faster on existing deployments.

Spark Streaming

Spark streaming is responsible for high-throughput, scalable, fault-tolerant stream processing of live data. It uses fast scheduling ability of Spark Core to perform streaming analytics.

Spark MLlib

Apache Spark is equipped with a rich library known as MLlib, which contains a wide range of high-quality machine-learning algorithms, including classification, clustering and collaboration filtering, as well as a number of lower-level primitives. Apache Spark MLlib is one of the best choices for a data scientist, due to its in-memory data processing capability, which dramatically improves the performance of iterative algorithms.

Spark GraphX

GraphX[4] is a component of Spark that is used for graphs and graph-parallel computation. GraphX extends the Spark resilient distributed datasets (RDDs) by including property graphs, which are directed multi-graphs where vertices and edges have properties attached to them. These properties are user-defined objects. In order to attach different types of properties to the vertices of the same graph, inheritance is used where all object types extend from a single parent object. Each vertex is identified by its unique 64-bit long identifier called a vertex id. Correspondingly, edges are identified by their source and destination vertices. GraphX does not force any restrictive order on the vertex identifiers.

To simplify the graph analytics tasks, GraphX provides many graph algorithms[5] and builders[6]. To support graph computation, GraphX includes a set of underlying operators including reverse, connectedComponents[7], and aggregateMessages[8]. Following is the description of some of the useful operators provided by GraphX:

1. `def reverse: Graph[VD, ED]`

   The reverse operator reverses the direction of all edges and returns a new graph.

[4]https://spark.apache.org/docs/latest/graphx-programming-guide.html
[5]https://spark.apache.org/docs/latest/graphx-programming-guide.html#graph_algorithms
[6]https://spark.apache.org/docs/latest/graphx-programming-guide.html#graph_builders
[7]https://spark.apache.org/docs/latest/graphx-programming-guide.html#connected-components
[8]https://spark.apache.org/docs/latest/graphx-programming-guide.html#aggregateMessages

2. `def groupEdges(merge: (ED, ED) => ED): Graph[VD, ED]`

   The groupEdges operator merges parallel edges in the multigraph.

3.
   ```
   def joinVertices[U](table: RDD[(VertexId, U)])
   (mapFunc: (VertexId, VD, U) => VD)
   : Graph[VD, ED]
   ```

   The joinVertices joins the vertices with the input RDD and apply the user defined map function to the joined vertices. It returns a new graph with the new vertex properties resulted in joining and map operations. Vertices without a matching value in the RDD keep their actual value.

4.
   ```
   def collectNeighbors(edgeDirection: EdgeDirection)
   : VertexRDD[Array[(VertexId, VD)]]
   ```

   The collectNeighbors operator collects neighboring vertices and their attributes at each vertex.

5.
   ```
   def aggregateMessages[Msg: ClassTag](
         sendMsg: EdgeContext[VD, ED, Msg] => Unit,
         mergeMsg: (Msg, Msg) => Msg,
         tripletFields: TripletFields = TripletFields.All)
       : VertexRDD[A]
   ```

   The aggregateMessages operator executes a user defined sendMsg function to every edge triplet and afterward uses the mergeMsg function to aggregate those messages at their goal vertex.

6. `def connectedComponents(): Graph[VertexId, ED]`

   The connected components algorithm assigns every connected component of the graph with the ID of its lowest-numbered vertex.

2.3 Scala

Scala[9] is a general-purpose programming language that merges functional and object-oriented programming in one concise, high-level language. It is object-oriented, but also provides functional programming features such as lazy evaluation, currying, immutability, and higher order functions. Static types of scala help avoid bugs in complicated applications, and its JVM and JavaScript runtimes empower users to build high-performance systems. It also provides access to a vast ecosystem of libraries. Scala is designed to run on a JVM platform, which means direct access to Java libraries and other feature-rich APIs. Scala's complex features result in more effective coding and increased performance.

[9]https://www.scala-lang.org

Chapter 3

Related Work

Most of the research in the cooking domain has focused on developing techniques for producing successful ingredient combinations. Some of the researchers have considered complete recipes, but have failed to provide exact quantities of ingredients. Many recipes produced by these techniques contain invalid cooking instructions and are not easy to follow. In this chapter, we review the significant contributions to research in this field.

3.1 IBM Chef Watson

A team of IBM researchers has built a program called Chef Watson[1] that uses mathematics, chemistry, and vast quantities of data to discover new recipes [9]. According to the IBM, Chef Watson is part of IBM's mission to develop cognitive computing applications that can assist individuals in discovering new ideas [10]. Chef Watson has analyzed 10,000 Bon Appetit[2] recipes and looked for statistical correlations between ingredients that tended to appear together in recipes. Chef Watson inspects the patterns in existing recipes and combines them with the science behind food pairings to offer unexpected combinations.

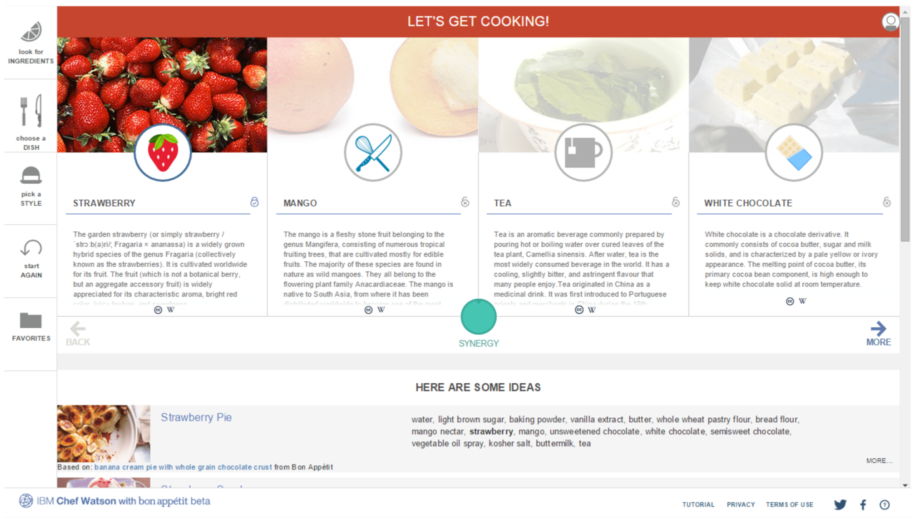
IBM also provides a web application that allows users to search by ingredients, a particular dish, or a style of cuisine to access the recipes. The web interface of this web application is shown in FIGURE 3.1. The algorithm generates millions of new recipe ideas to match the user's requirements and chooses some of them to show to the user. The new recipes are produced by changing the ingredients of existing recipes with the user's preferred ingredients. The user can follow exact recipes or tweak them using alternative ingredients suggested by the application. From the vast number of recipes that the application generates, selections are made based on novelty and quality. According to IBM researchers, a creative idea ought to be novel and of high quality. IBM evaluates the novelty of a recipe using a mathematical tool known as Bayesian Surprisal to measure how much it differs from existing recipes. Bayesian Surprisal compares existing beliefs about food with new beliefs after the introduction of a newly created recipe; the more noteworthy the change in beliefs, the greater the value of the novelty.

The quality of a recipe is measured based on the smell of dish. To measure the smell, the algorithm examines all the discrete flavor molecules in a recipe and finds their chemical properties. These chemical properties are then compared to that of 70 other odor molecules to estimate how good a specific molecule smells. They mix the smells of distinct molecules together and compute the final 'pleasantness' of smell

[1]http://www.ibmchefwatson.com

[2]https://www.bonappetit.com

[3]`http://www.dataversity.net/ibm-and-bon-appetit-serve-up-chef-watson-for-all/`



The garden strawberry (or simply strawberry / ˈstrɔːb(ə)ri/; Fragaria × ananassa) is a widely grown hybrid species of the genus Fragaria (collectively known as the strawberries). It is cultivated worldwide for its fruit. The fruit (which is not a botanical berry, but an aggregate accessory fruit) is widely appreciated for its characteristic aroma, bright red color, juicy texture, and sweetness.

The mango is a fleshy stone fruit belonging to the genus Mangifera, consisting of numerous tropical fruiting trees, that are cultivated mostly for edible fruits. The majority of these species are found in nature as wild mangoes. They all belong to the flowering plant family Anacardiaceae. The mango is native to South Asia, from where it has been distributed worldwide to become one of the most

Tea is an aromatic beverage commonly prepared by pouring hot or boiling water over cured leaves of the tea plant, Camellia sinensis. After water, tea is the most widely consumed beverage in the world. It has a cooling, slightly bitter, and astringent flavour that many people enjoy.Tea originated in China as a medicinal drink. It was first introduced to Portuguese priests and merchants in China during the 16th

White chocolate is a chocolate derivative. It commonly consists of cocoa butter, sugar and milk solids, and is characterized by a pale yellow or ivory appearance. The melting point of cocoa butter, its primary cocoa bean component, is high enough to keep white chocolate solid at room temperature.

water, light brown sugar, baking powder, vanilla extract, butter, whole wheat pastry flour, bread flour, mango nectar, **strawberry**, mango, unsweetened chocolate, white chocolate, semisweet chocolate, vegetable oil spray, kosher salt, buttermilk, tea

Based on: banana cream pie with whole grain chocolate crust from Bon Appétit

FIGURE 3.1: IBM Chef Watson's Web Interface[3]

of each dish. Finally, the algorithm generates a list of recipes ranked according to three categories: novelty, pleasantness of smell, and flavor pairings.

While IBM's Chef Watson has generated many successful recipes, including ”Portuguese Lobster Roll” [11] and ”Caymanian Plantain Dessert” [12], it has also created many unappetizing recipes, including ”beef burrito accented with chocolate and edamame” [13] and ”Chicken Breast Taco” [14]. Although much effort has been put into IBM's research, Chef Watson has failings. Some of these are as follows:

1. Chef Watson does not consider the texture of the final dish.
2. If the system discovers that okra works better than the onion in a recipe, it replaces onion with okra but does not replace the cooking method for the onion with the cooking method for okra. This poses a problem because onions can be added raw to some recipes without preprocessing steps, whereas okra cannot be used in that way. Some replacement ingredients may take more time to fry or bake than the original ingredient, which can result in many recipes suggested by IBM Chef Watson having invalid cooking instructions.
3. As the system focuses mostly on novelty, it tries to generate unseen recipes, which are often complex and may contain ingredients that are difficult to find. A user of IBM's web application reported: "I've played around with the Chef Watson app but sometimes it leads to hilarious results. Calls for wasabi powder (never used), shelled green peas (21/2 cup shelled green peas) cut into 3/4" pieces. Then placed on a barbecue." [4]
4. IBM clearly states that Watson is a tool to aid chefs more than anything else. Chefs can discover novel recipes and new ingredient combinations with the help of the system, but they cannot completely rely on the instructions it gives.

[4]https://www.reddit.com/r/IAmA/comments/3id842/we_are_the_ibm_chef_watson_team_along_with_our/#bottom-comments

Example#01	Example#02
Ingredients	Ingredients
• 1 cup skillet	• Med okra
Instructions:	• Lot sugar
1. Chop: skillet	Instructions:
2. Serve:	1. Boil: sugar okra sugar

TABLE 3.1: Recipes from Initial Version of EMMA

3.2 Cover:Cheese (EMMA)

"Cover:Cheese"[5] is a project developed by the Hampshire College Mad Science Club[6] that uses a genetic algorithm to produce recipes. The club has built a computerized program named Evolutionary Meal Management Algorithm (EMMA) that can automatically produce new recipes and enhance their quality by considering the feedback on their taste.

As the evaluation of new recipes is a crucial part of the genetic algorithm and the algorithm can generate new recipes far faster than they actually can be prepared or tested, the evaluation of recipes is conducted through crowdsourcing. To calculate the rating of a recipe, they publish the recipes on their website and allow people to rate them between 1 and 10, 10 being the highest rating. Users can rate a recipe either by prediction (without cooking the dish), or they can give an actual rating (after cooking).

The earlier version of EMMA was not so well developed to detect and differentiate between edible and non-edible recipes. TABLE 3.1 shows some example of recipes from the initial version of EMMA.

Even the current version of EMMA is not able to suggest high-quality recipes. The top-rated recipes listed on their official website have no clear instructions. Take the example of the recipes shown in TABLE 3.2.

EMMA appears to be a failed meal management system at this level. A professional cook might get an idea for a new recipe from the ingredients and proposed instruction, but a learner cook might follow the recipes exactly thinking that any deviation from instructions would produce poor results. Currently, the EMMA website has approximately 130 recipes rated from 0.03 to 7.0 out of 10 by random visitors. Only 38 recipes have earned ratings of more than 5. The two main flaws in EMMA are as follows:

Instructions are Not Easy to Understand

The cooking instructions are not easy to understand for either professional or the learner cooks. According to the EMMA team, recipes can sometimes appear strange and unclear, because they must be in a format that their algorithm can understand. According to them, efforts have been made to make the recipes easy to read, but the

[5]https://covercheese.appspot.com
[6]https://www.hampshire.edu/

Example#01	Example#02
Ingredients	Ingredients
• Small angel_food	• lot chocolate sauce
• Small eggplant	• med rice pudding
Instructions:	• lot cinnamon
1. Slice: angel_food	Instructions:
2. eggplant angel_food	1. warm:sauce pudding
	2. mix:
	3. combine:cinnamon

TABLE 3.2: Recipes from the Latest Version of EMMA

process still does not work perfectly. For the purpose of cooking, a user at his own need to comprehend the available recipes and use the given instructions and ingredients for the sake of idea only.

Exact Quantities are Not Given

Recipes generated by the program list ingredients but do not specify exact quantities. They explained in FAQ section of their website[7] that it is extremely difficult to get a computer to figure out the correct quantity of something, because such a calculation entails both an understanding of what the ingredient is and how it fits with the rest of the ingredients (i.e. one cup of sugar might be appropriate for a pie but not for soup).

3.3 Recipe Recommendations using Ingredient Networks

A recipe recommendation algorithm developed by Teng et al. [15] recommends a recipe based on its rating. They developed a technique to automate rating of recipes based on the ingredients used in the recipe. They used allrecipe.com[8] website as the main data source, and collected 46,337 recipes. Using these recipes, the team constructed two kinds of networks to represent the correlation between ingredients. The complement network shown in FIGURE 3.2 represents the co-occurrence of the ingredients and is composed of two large groups: savory and sweet. The co-occurrence of ingredients across recipes is measured to extract the knowledge of users about combining ingredients. Complementary ingredients incline to occur together far more frequently than would be expected by chance.

The substitute network shown in FIGURE 3.3, obtained from user-generated recommendations for modifications, is made up of multiple communities of functionally equivalent ingredients and captures users' preference for healthier versions of recipes. This weighted, directed network captures users' knowledge of ingredient

[7]https://covercheese.appspot.com
[8]https://www.allrecipes.com

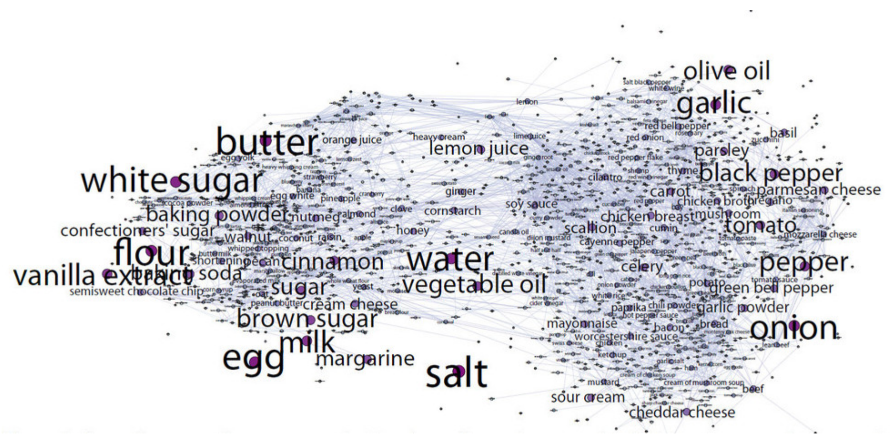


FIGURE 3.2: Ingredient Complement Network [15]

replacement, with the network consisting of ingredients as nodes. These ingredient networks are used along with nutritional information to predict recipe ratings, although the ingredient networks contribute the majority of the prediction power.

The algorithm uses ingredient networks as the main feature sets to predict recipe ratings and compares them against features that encode nutritional information and other basic features such as preparation, cooking methods and cooking times. It employs stochastic gradient boosting trees [16], a discriminative machine-learning method, to predict the ratings of recipes. The developers have found that if the recipe has at least two complementary ingredients, it marginally improves its prospects but having clashing or unrelated ingredients does not appear to do harm.

3.4 Flavor Network and the Principles of Food Pairing

A flavor network has been developed by Yong-Yeon Ahn et al. [2] that represents the flavor compounds shared by culinary ingredients. To explore the influence of flavor compounds on ingredient combinations, they used a network-based approach. Food chemists identified the flavor compounds contained in most culinary ingredients and then each ingredient was linked to an average of 51 flavor compounds. They built a bipartite network that contains two different kinds of nodes: (i) 381 ingredients used in recipes all over the world; and (ii) 1,021 flavor compounds that contribute to the flavor of these ingredients. The flavor network is a projection of this bipartite network in which two nodes are linked if have at least one flavor compound in common in them. The weights of the links between nodes (ingredients) represent the number of shared flavor compounds between the ingredients represented by the nodes, which makes the flavor network a weighted network.

To identify the relation between ingredient combinations and flavor network, they used 56,498 recipes from two American repositories (epicurious.com and allrecipes.com) and one Korean repository (menupan.com). Finally, they found that the ingredient pairs sharing more compounds are more likely to be used in particular cuisines.

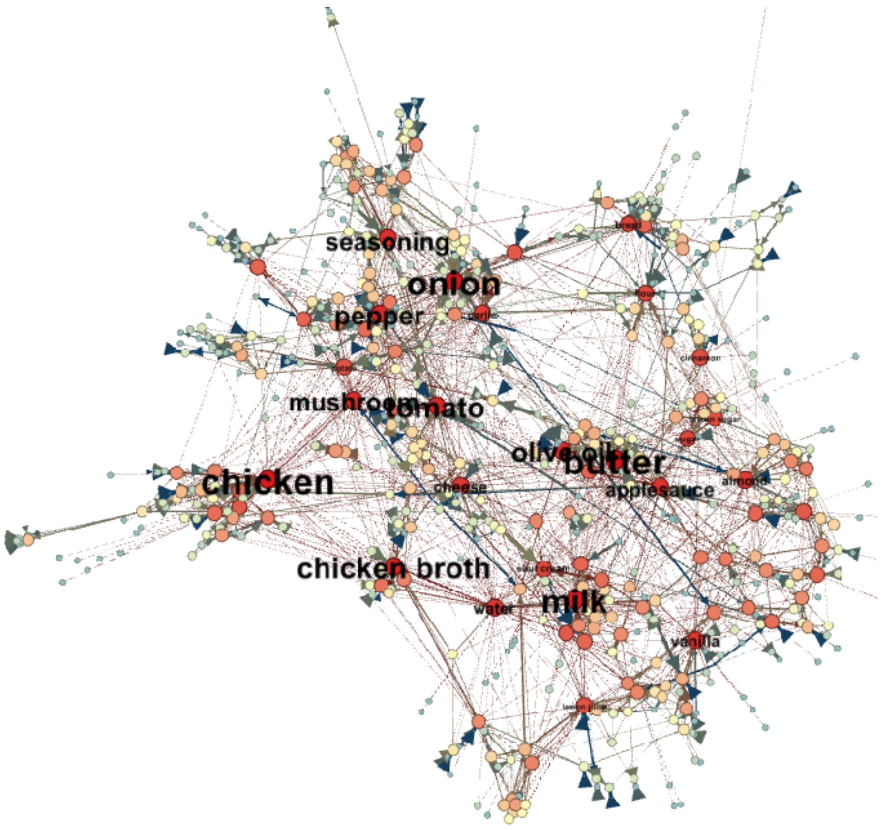


FIGURE 3.3: Ingredient Substitute Network [15]

3.5 Computational Creativity in the Culinary Arts

Erol et al. [17] has developed a technique for discovering new ingredient combinations for salad dishes. They used the methods introduced by Yong-Yeon Ahn et al. [2] and Teng et al. [15] to discover new ingredient combinations and selected the ones with the best ratings. The system was designed in two steps. First, they built a statistical model to rank recipes. Then, they applied different search algorithms to explore salad recipes and discovered novel ingredient combinations. Like IBM, they focus on novelty. Their approach ignores the corresponding quantities of ingredients, which is a key determinant of a recipe's success. Some of the ingredient combinations proposed by EROL are following:

Combination 1:

"Paprika, poppy seed, sesame seed, spinach, strawberry, vegetable oil, white sugar, white wine" [17]

Combination 2:

"Cherry, chive, granny smith apple, mushroom, onion powder, pine nut, salsa, salt" [17]

Combination 3:

"Bok choy, feta cheese, green onion, mango, radish, red onion, rotini pasta" [17]

Chapter 4

Methodology

Sections 4.1 and 4.2 in this methodology chapter describe how the properties of the different components of the recipes are represented and the relations between them. In section 4.3, the details of data collection and cleaning are given. Section 4.4 explains the evolutionary algorithm, and section 4.6 describes a test run of the algorithm on our data.

4.1 Components of a recipe

A recipe consists of a main ingredient, spices, one or more side ingredients, and cooking instructions. Along with the main cooking steps, a recipe might have preprocessing and post-processing steps. Preprocessing steps are performed on ingredients before they are cooked, and post-processing mixes ingredients together after the actual cooking steps are performed on them. The TABLE 4.1 explains these components in more detail.

4.2 Solution Encoding

A recipe contains a set of ingredients and spices and instructions for cooking. We used the property graph provided by GraphX to encode recipes, where vertices in the graph represent components of the recipes (i.e., the ingredients, the spices and the cooking instructions) and the edges define relations between them (e.g., "has spice" or "goes-in").

The cooking process can be divided into two sub-processes, i.e., the main process and the side process. The main process contains the preprocessing and cooking of the main ingredient before it is mixed with other ingredients and spices. The main process also contains those side ingredients and spices that go directly with the main ingredient, e.g., when a recipe calls for potatoes to be boiled with a pinch of salt, if potato is the main ingredient, the salt will also be the part of the main process, although it is a spice. The side process consists of the preprocessing and cooking of the side ingredients that are not part of the main process. The post-processing steps of a recipe are also considered part of the side process in our graph representation. In FIGURE 4.1, an example recipe is given with one main ingredient, two side ingredients and one spice. The green area on the graph shows the main process that includes one main ingredient, and one spice and the further step of cooking. The rest of the graph represents the side process. Each ingredient will have a block in the graph that takes its own way of cooking and its own spices with it. TABLE 4.2 and 4.3 describe the properties of vertices (components of the recipe in our case) and edges (relation between the components of a recipe) respectively.

Component	Description
Main ingredient	The main ingredient is a major ingredient of a recipe, for example, in all types of rice dishes; rice is the main ingredient.
Side ingredient	Side ingredients are all the ingredients used in the recipe other than the main ingredient.
Spice	The spice is a type of side ingredient used to add flavor to food.
Step	Steps compose the cooking instruction.
Pre-processing step	A step required before actual cooking step. For example, to fry onions, we first need to peel and cut them into chunks.
Post-processing step	The post-processing steps mix the individual ingredients together after the actual cooking steps.

TABLE 4.1: Components of a recipe

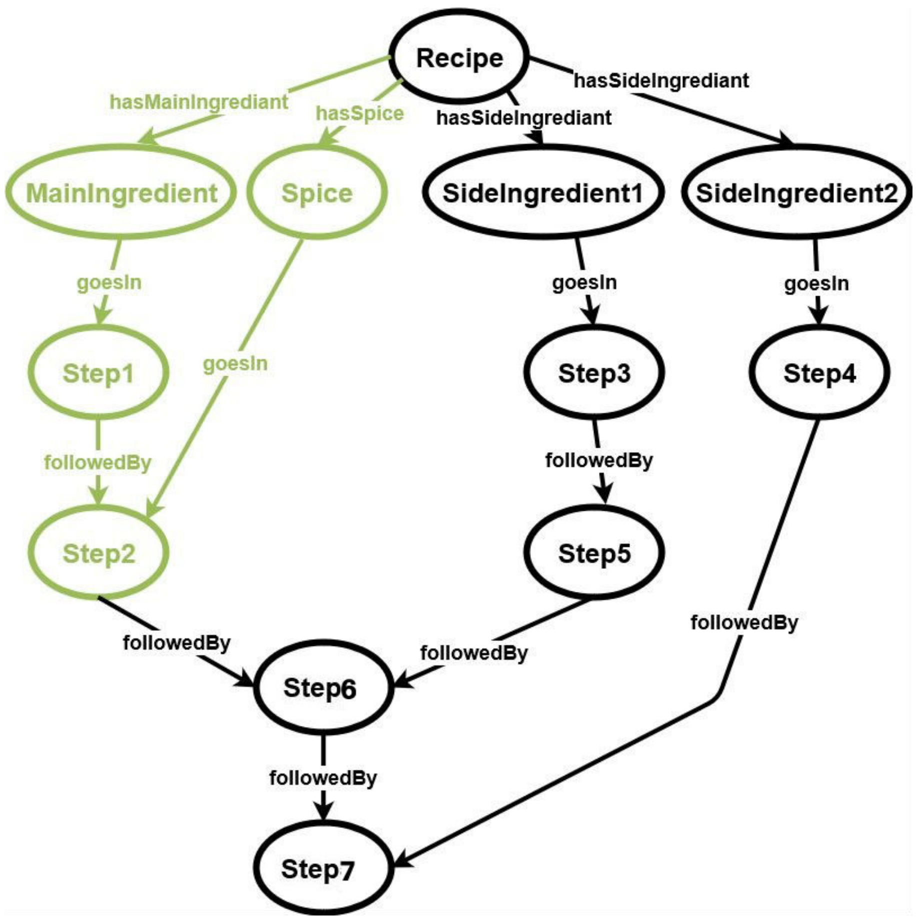


FIGURE 4.1: Graph Representation of an Arbitrary Recipe

	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 in finished. In other words, it is the level to which the main ingredient is cooked before it is going to be added to 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, step can also be a post-step or last-step
usedIn	Whether the step is the part of main process or side process

TABLE 4.2: Vertex 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: The 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

TABLE 4.3: Edges/Relations properties

4.3 Data Collection and cleaning

The recipes were collected from multiple cooking websites, including Yummly.com[1], Allrecipes.com[2], Recipes-plus.com[3], Geniuskitchen.com[4], Simplyrecipes.com[5], Omnivorescookbook.com[6] and Greenevi.com[7]. Data consists of Southern, American, Italian, Spanish, Hungarian and Chinese recipes.

These recipes consist of cooking instructions and a list of ingredients with quantity. Sometime pre-processing steps are also listed with ingredients instead of the instruction part. Following steps were performed on each recipe to make it ready to be used for GraphX:

1. Multiple quantity options were converted to one fixed 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 with instruction part. 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 as a separate step in the instruction part.

[1]https://www.yummly.com
[2]https://www.allrecipes.com
[3]http://recipes-plus.com
[4]https://www.geniuskitchen.com
[5]https://www.simplyrecipes.com
[6]https://omnivorescookbook.com
[7]http://greenevi.com

4. Each step was assigned a property usedIn (whether the step is part of the main process or side process).
5. The recipe node was assigned properties that are recipe name, the total time required to cook the recipe, number of servings, category of the recipe (Main dish/Side dish), cooked type of main ingredient after the main process (raw, half cooked, cooked).

The example below illustrates the above steps briefly:

Recipe: Roasted potato spinach and parmesan salad [18]
Ready in: 55 mins, Serves: 6

Ingredients:

- 2lbs yukon gold potatoes, cut into 3/4 inch wedges
- 2 ½ tablespoons olive oil
- ¾ teaspoon dried rosemary, crumbled
- salt and freshly ground black pepper
- ½ cup light mayonnaise
- 1 teaspoon lemon zest, grated
- 1 ½ tablespoons water
- 1 garlic clove, crushed
- 6 cups Baby Spinach
- ½ cup parmesan cheese, shaved

Instructions:

1. Preheat oven to 450 degrees F.
2. On a large baking sheet, toss together potatoes, oil, rosemary, and 3/4 teaspoon salt.
3. Roast until potatoes are tender and golden, tossing once or twice, about 40 minutes; cool.
4. In a large salad bowl, whisk together mayonnaise, lemon zest, lemon juice, water and garlic.
5. Add potatoes, spinach, and Parmesan cheese; toss.
6. Season with freshly ground black pepper.

The recipe above will be converted to the following format:

Recipe (recipeName – cookTime – servings – category – cookedType – mainIngredient)

```
ROASTED POTATO SPINACH AND PARMESAN SALAD-00:55-6-Side Dish-cooked
```

Ingredients (ingredientName-quantity-measurementUnit-ingredientType-usedIn)

```
yukon gold potatoes-2-lbs-main-main
olive oil-2.5-tablespoons-side-main
dried rosemary-0.75-teaspoon-side-main
salt-0-null-spice-main
black pepper-0-null-spice-side
light mayonnaise-0.5-cup-side-side
lemon zest-1-teaspoon-side-side
lemon juice-0-null-side-side
water-1.5-tablespoons-side-side
garlic clove-1-unit-side-side
Baby Spinach-6-cups-side-side
parmesan cheese-0.5-cup-side-side
```

Steps (description-stepNo-usedIn)

```
Cut potatoes into 3/4 inch wedges.-1-main
Crumble dried rosemary.-2-main
Grate lemon zest.-3-side
Crush garlic clove.-4-side
Shave parmesan cheese.-5-side
Preheat oven to 450 degrees F. On a large baking sheet, toss together potatoes,
oil, rosemary, and 3/4 teaspoon salt.-6-main
Roast until potatoes are tender and golden, tossing once or twice, about
40 minutes; cool.-7-main
In a large salad bowl, whisk together mayonnaise, lemon zest, lemon juice,
water and garlic.-8-side
Add potatoes, spinach, and Parmesan cheese; toss.-9-side
Season with freshly ground black pepper.-10-side
```

As shown in the steps part of the above recipe, pre-processing steps that came with ingredients were added as steps instead of ingredients. The first five steps including “Cut potatoes into 3/4 inch wedges”, “Crumble dried rosemary”, “Grate lemon zest”, “Crush garlic clove” and “Shave parmesan cheese” were not the part of the cooking instruction in the originally collected recipe.

Side ingredients and spices that are used with the main ingredient are assigned the value "main" to their "usedIn" property, in above example, olive oil and dried rosemary are side ingredients and the value of their "usedIn" property is "main" as they are part of the main process. In the same way, salt is a spice and it is also the part of the main process as it is added to potatoes while baking. If the quantity and measurement unit for a spice or ingredient is not given in the ingredient list of a recipe; a “0” and “null” are assigned as a replacement for quantity and measurement unit respectively.

4.4 Evolutionary algorithm to evolve recipes

To improve the validity and taste of recipes over time, we used an evolutionary algorithm. In this section we explain the initial population, the evaluation of recipes, the selection process and the creation of new generation. The algorithm is stopped

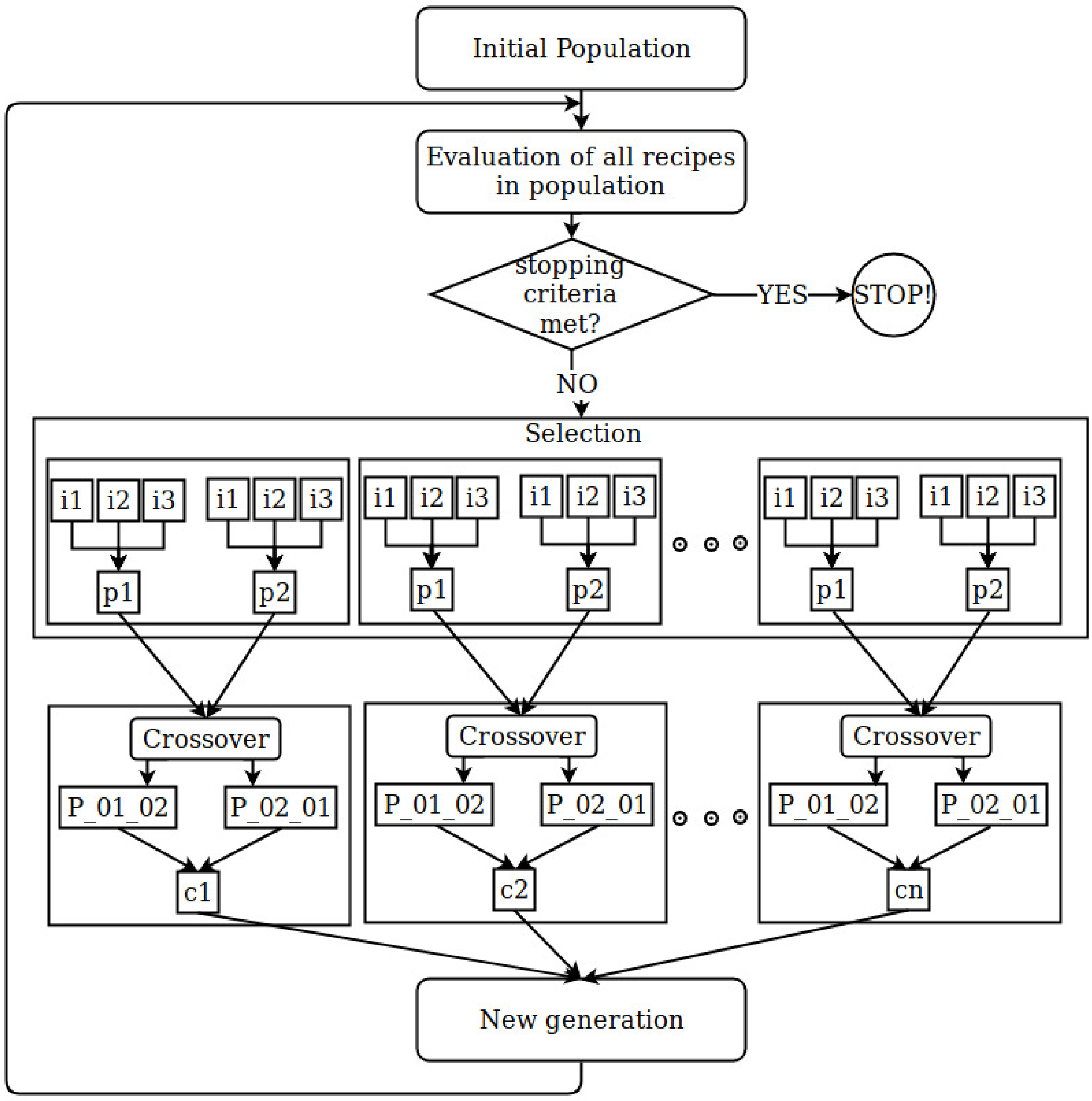


FIGURE 4.2: Evolutionary Algorithm

when more than 85% of the recipes have a favorable rating (more than 4 out of 5). FIGURE 4.2 describes this process in detail.

4.4.1 Initial Population

The initial population is the first group of recipes, which are generated by combining recipes written by humans. Our data contains recipes from well-known websites. The recipes are written and tested by people and most are highly rated recipes as most of the users share successful recipes. Our purpose is to improve the recipes that are generated by combining these recipes. To generate the initial population, random pairs were chosen from the database, and a fixed initial population was generated by performing a crossover on each pair of recipes.

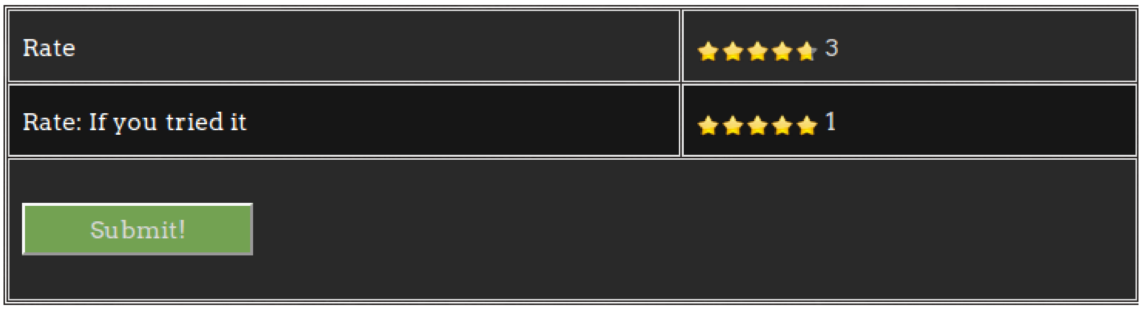


FIGURE 4.3: Recipe Rating Box

4.4.2 Evaluation of recipes

As a true assessment of the quality of the recipes could only be carried out by humans, to get the ratings of the recipes, we listed them on a dedicated website[8] and placed a rating box at the end of each recipe as shown in FIGURE 4.3.

Users could rate recipes by awarding them between 1 and 5 stars (where 1 was the minimum and 5 the maximum rating). There were two options for rating. For the first one, a person could just read the recipe, assesses its quality, and rate it. The second option was for users who had cooked and tasted the recipe before rating it. The first type of rating was used as the fitness function for the evolutionary algorithm, as we did not receive enough ratings of the second type.

4.4.3 Selection

For the selection of parents to produce the next generation of individuals, we used tournament selection[9] with a tournament size of three. Tournament selection runs a number of tournaments between randomly selected individuals from the current population, and the individuals with the best fitness are selected as parents. The number of randomly selected individuals in each tournament is equivalent to the tournament size. To produce a child, we ran two tournaments, each time selecting the best individual out of three randomly selected individuals. Then, the two selected parents were crossed to produce a child. This process was repeated until we generated a fixed number of individuals in the next generation. For example, for eight individuals in each generation, the process of selecting parents and producing a child was repeated eight times.

The pseudo code in FIGURE 4.4 depicts tournament selection used in our selection method: As shown in the selection part of FIGURE 4.2, the individuals i1, i2, i3 are randomly selected from the population and p1 and p2 are two best individuals selected as parents after running the tournament two times.

4.4.4 Offspring Generation

After the selection of two recipes as parents, their crossover is taken in order to generate new off-springs. The FIGURE 4.5 explains the process of generation of children after the selection of parents.

In the first step of the FIGURE 4.5, two recipes Recipe#01 and Recipe#02 selected as parents are shown. After selecting a pair of recipes, it is necessary to check if it

[8]http://www.machinegeneratedrecipes.de
[9]https://en.wikipedia.org/wiki/Tournament_selection

```
1   Run tournament1: select 3 recipes from the population at random and
    choose 1 of them with best rating
2   Run tournament2: select 3 recipes again from the population at random
    and choose 1 of them with best rating
3   Take the recipes selected in step 1 & step 2 as parent recipes to produce
    an offspring
4   Repeat step 1, 2 and 3 until we have enough parents to produce individuals
    of next generation
```

FIGURE 4.4: Tournament Selection

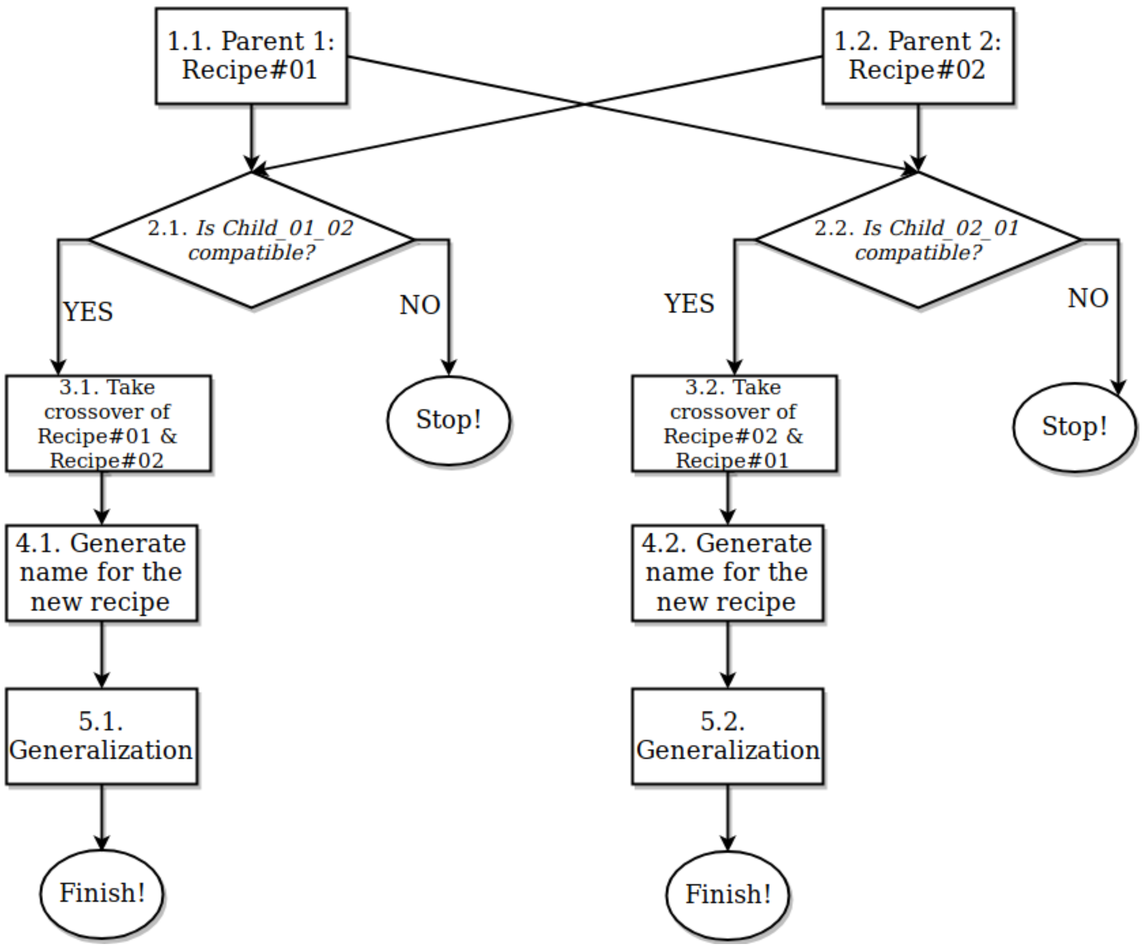

1.1. Parent 1: Recipe#01
1.2. Parent 2: Recipe#02
2.1. Is Child_01_02 compatible?
2.2. Is Child_02_01 compatible?
YES
NO
YES
NO
3.1. Take crossover of Recipe#01 & Recipe#02
Stop!
3.2. Take crossover of Recipe#02 & Recipe#01
Stop!
4.1. Generate name for the new recipe
4.2. Generate name for the new recipe
5.1. Generalization
5.2. Generalization
Finish!
Finish!

FIGURE 4.5: Offspring Generation

is appropriate to combine these two recipes; whether the combination will generate a valid or invalid recipe. If the compatibility function returns NO, the process of generation of the new recipe is stopped.

In step 2.1 and 2.2 of the FIGURE 4.5, the compatibility of Child_01_02 and Child_02_01 is checked. These two children represent two different recipes as the main process of the new recipe is taken from the one of the parent recipes and the side process is taken from another parent. So, in case of Child_01_02 the new recipe has the main process from Recipe#01 and side process from Recipe#02; and this recipe is different from Child_02_01 where the newly generated recipe has the main process from Recipe#02 and side process from Recipe#01.

After the compatibility check, the crossover of the recipes is taken using fixed one-point crossover where the crossover point separated main process and side process of a recipe. For a pair of recipes, there is a possibility that Child_01_02 is compatible, and Child_02_01 is not compatible and vice versa. So, the process of offspring generation can return two new recipes or one new recipe based on the compatibility function that is explained below.

Compatibility function:

To check compatibility, the level to which the main ingredient of each recipe was cooked determined. To keep track of the status of the main ingredient, a property named **cookedType_mainIngredient** is assigned to the recipe that represents the state of the main ingredient after the main process of the recipe is finished. This property can have values "raw", "cooked" or "half cooked". The parameter **cookedType_mainIngredient** of the two parent recipes is compared in the following way to check compatibility. A Child_01_02 is compatible if:

1. The main ingredient in both recipes is cooked to the same level, i.e., (raw, raw), (cooked, cooked) or (half cooked, half cooked) where the 1st parameter is for Recipe#01 and the second for Recipe#02. For example, if the main ingredient in both recipes is raw, when the main ingredient of Recipe#01 is combined with Recipe#02, it is raw and needs to be cooked to generate a valid recipe. But the main ingredient of Recipe#02 is also raw after the main process is finished, which indicates that at a later point the main ingredient will be cooked. Therefore, the final recipe is valid. As the original recipes are valid recipes, all the ingredients are cooked in the finished recipe. This means that recipes with the same cookedType_mainingredient are always valid.

2. The **cookedType_mainingredient** in Recipe#01 is "half cooked" and in Recipe#02 is "raw". As Recipe#02 is a valid recipe and its main ingredient is raw after the main process is finished, the main ingredient of Recipe#02 will be cooked at some point in the side process of Recipe#02. Therefore, the final generated recipe will be valid. It might be the case that the main ingredient will be overcooked, but the recipe is still valid and has a high probability of tasting good.

3. If the main ingredient of Recipe#01 is "cooked" and that of Recipe#02 is either "raw" or "half cooked," the recipe is still valid as a cooked ingredient can be cooked again.

4. In case that the cooked type of the main ingredient of Recipe#01 is "half cooked" and Recipe#02 is "cooked", the combination of these recipes is still valid. As the main ingredient is being used in the side process of Recipe#02, it will be

cooked during the side process and the newly generated recipe will still be easy to follow and will produce an edible dish.

In cases where rule 4 was applied, the generated recipe were generally rated highly. However, there is a chance that the main ingredient will not be cooked well during the side process, and the recipe may not taste good. Below are two examples recipes generated based on rule 4, one of them is a high rated recipe, while the other one received bad rating from the users. TABLE 4.4 shows two parent recipes, where Recipe#01 has the main ingredient that is half cooked after the main process and Recipe#02 with the main ingredient that is cooked. Unexpectedly, the side process of Recipe#02 not cooked the main ingredient and the generated a low quality recipe. The main process of parent recipes is represented by green color and black color shows side process of recipes. Here is the to-do part of child recipe generated in result of cross-over of Recipe#01 and Recipe#02 (Child_01_02) from TABLE 4.4.

1. Precook the potatoes in water (or in the microwave) until almost, but not quite, done. Drain thoroughly.
2. Crush the garlic clove.
3. Chop parsley.
4. In a bowl mix the mayonnaise, chopped parsley, crushed garlic, and paprika.
5. Add the potatoes to the garlic mayonnaise and mix well.
6. Leave in the fridge for 30 mins before serving at room temperature.

As it is clear from the instruction part that if the recipe is followed exactly as it is stated, potatoes will not be fully cooked in the resultant dish. As they are not quite done when being mixed into the garlic mayonnaise mixture in step 5.

But the recipes passing through compatibility function using rule 4 can also generate a high-quality food as our next example will prove. TABLE 4.5 shows the combination of Recipe#01 with another parent recipe Recipe#02, where Recipe#01 is with the main ingredient that is half cooked and Recipe#02 with the main ingredient that is fully cooked. The recipe generated by the combination of these two recipes got high rating and generated an edible dish. Recipe generated in result of cross-over of Recipe#01 and Recipe#02 (Child_01_02) from the TABLE 4.5 is shown below:

1. Precook the potatoes in water (or in the microwave) until almost, but not quite, done. Drain thoroughly.
2. Chop cilantro.
3. Mix cayenne pepper, cumin powder, five spice powder, chicken stock and the rest of salt in a small bowl.
4. Add butter in a non-sticky skillet and heat with medium heat. When butter melted and stop bubbling, add potatoes and spice mixture, stir fry till potatoes are evenly seasoned and browned all sides.
5. Garnish with cilantro and serve immediately.

Recipe#01 [19]	Recipe#02 [20]
1. Coarsely chop onion.	1. Crush the garlic clove.
2. Peel and mince garlic.	2. Chop parsley.
3. Coarsely chop tomatoes.	3. Peel the potatoes and boil in salted water for approx. 20 mins check that the potatoes are done then drain and leave to cool.
4. Precook the potatoes in water (or in the microwave) until almost, but not quite, done. Drain thoroughly.	4. In a bowl mix the mayonnaise, chopped parsley, crushed garlic, and paprika.
5. While the potatoes are cooking, saute the fenugreek seed in the oil on medium heat until light brown, being careful not to burn them.	5. Once the potatoes have cooled, chop them into small chunks (2x2cm).
6. Add the onion and continue cooking for five minutes. Add the ginger and garlic and cook another five minutes. Add the spices and saute briefly to release their flavors. Add the tomato, the dried whole peppers, and a little water. Simmer until the flavors meld together, about 30 minutes.	6. Add the potatoes to the garlic mayonnaise and mix well.
7. Gently add the potatoes, stir, and reduce heat. Cook until potatoes are tender, adding water if the sauce gets too dry. If the sauce is too runny, simply crush one of the potatoes to thicken it.	7. Leave in the fridge for 30 mins before serving at room temperature.

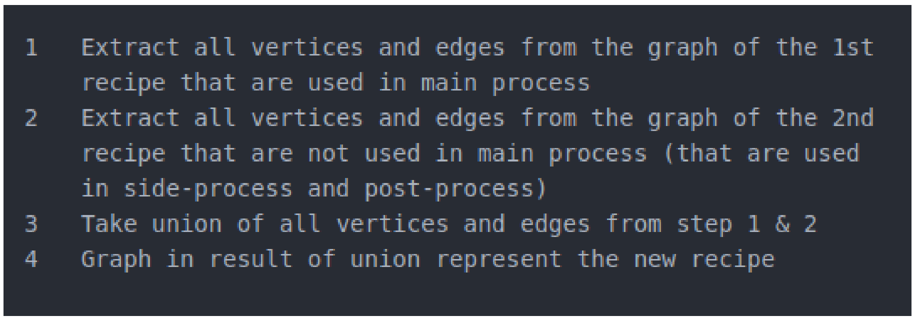
TABLE 4.4: Example1 - Recipes with **cookedType_mainIngredient** property value "half cooked" (Recipe#01) and "cooked" (Recipe#02)

Recipe#01 [19]	Recipe#02
1. Coarsely chop onion.	1. Chop cilantro.
2. Peel and mince garlic.	2. Add potato in a medium pot with a teaspoon of salt. Add water to cover potato and boil about 20 minutes, till you can poke through potato with a chopstick.
3. Coarsely chop tomatoes.	3. Drain potato and let it chill a bit. Cut into 1.5 cm squares.
4. Precook the potatoes in water (or in the microwave) until almost, but not quite, done. Drain thoroughly.	4. Mix cayenne pepper, cumin powder, five spice powder, chicken stock and the rest of salt in a small bowl.
5. While the potatoes are cooking, saute the fenugreek seed in the oil on medium heat until light brown, being careful not to burn them.	5. Add butter in a non-sticky skillet and heat with medium heat. When butter melted and stop bubbling, add potatoes and spice mixture, stir fry till potatoes are evenly seasoned and browned all sides.
6. Add the onion and continue cooking for five minutes. Add the ginger and garlic and cook another five minutes. Add the spices and saute briefly to release their flavors. Add the tomato, the dried whole peppers, and a little water. Simmer until the flavors meld together, about 30 minutes.	6. Garnish with cilantro and serve immediately.
7. Gently add the potatoes, stir, and reduce heat. Cook until potatoes are tender, adding water if the sauce gets too dry. If the sauce is too runny, simply crush one of the potatoes to thicken it.	

TABLE 4.5: Example2 - Recipes with **cookedType_mainIngredient** property value "half cooked" (Recipe#01) and "cooked" (Recipe#02)

Id	Recipe#01/Cook Property	Recipe#02/Cook Property	Pair_01_02/Compatible	Pair_02_01/Compatible
1	Raw	Raw	Yes	Yes
2	Raw	Cooked	No	Yes
3	Raw	Half Cooked	Yes	Yes
4	Half Cooked	Raw	Yes	Yes
5	Half Cooked	Cooked	Yes	Yes
6	Half Cooked	Half Cooked	Yes	Yes
7	Cooked	Raw	Yes	No
8	Cooked	Cooked	Yes	Yes
9	Cooked	Half Cooked	Yes	Yes

TABLE 4.6: Compatibility Possibilities



```
Extract all vertices and edges from the graph of the 1st
recipe that are used in main process
Extract all vertices and edges from the graph of the 2nd
recipe that are not used in main process (that are used
in side-process and post-process)
Take union of all vertices and edges from step 1 & 2
Graph in result of union represent the new recipe
```

FIGURE 4.6: Crossover Process

Above combinations generate a valid recipe, but there can be the case when the final recipe is invalid, and the process of recipe generation will be stopped as shown in FIGURE 4.5.
Child_01_02 is incompatible if the main ingredient of Recipe#01 is “raw” and that of Recipe#02 is “cooked”. In that case, there is a high probability that the side process of Recipe_02 will not cook the main ingredient of Recipe_01, and in the final dish, the main ingredient is likely to be uncooked and inedible.
The TABLE 4.6 shows all the possible combinations of the compatibility function.

Crossover

As shown in the FIGURE 4.2, cross-over is performed after the selection process. Two recipes named p1 and p2 are chosen as parent for each cross-over. Each pair of recipes generates a new pair of child recipes if they both are compatible with each other. As our recipes are represented in the form of a graph, we used the algorithm shown in FIGURE 4.6 to produce a new graph that represents a child recipe. Crossover is performed on the two parent recipes and intermixes the properties of these recipes to produce new recipes that are called child recipes. One child takes the main process from the first recipe and side process from the second recipe. The second child is composed of the main process from the second recipe and side process of the first parent recipe.

TABLE 4.7 shows two parent recipes that we want to evolve in order to generate new recipes. Recipe#01 have one main ingredient and four side ingredients. The main process of the recipe is represented in green colour while text with black colour

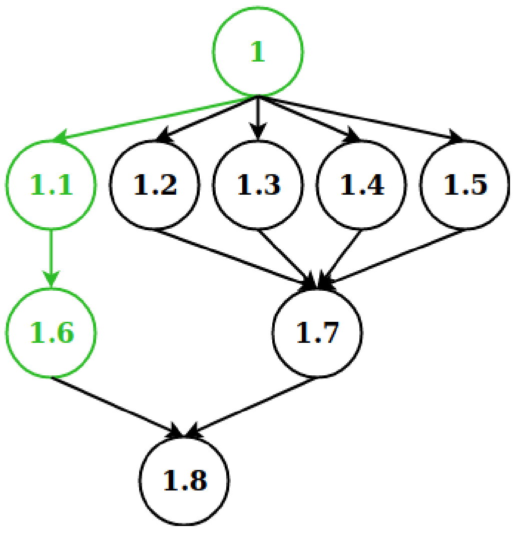


FIGURE 4.7: Graph Representation of First Parent (Recipe#01)

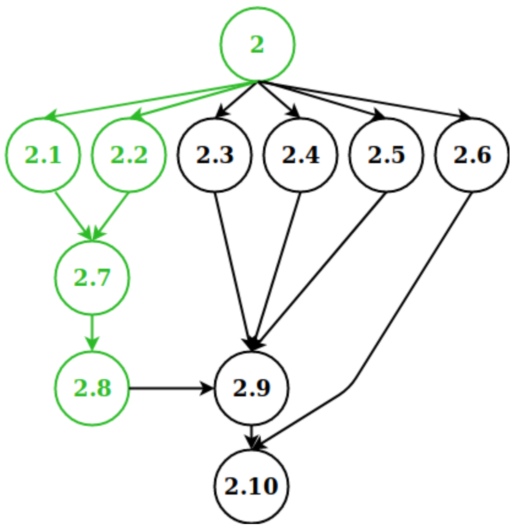


FIGURE 4.8: Graph Representation of Second Parent (Recipe#02)

shows side process of the recipe. In the first recipe, there is only one step that is part of the main process. FIGURE 4.7 shows graph representation of the first recipe. For the understandability, properties of the nodes were omitted from the graph and listed in TABLE 4.8. In the graph, ingredients and spices are directly attached to the root node that represent overall recipe. In the second layer of this graph, green node represents "sweet potatoes" that is the main ingredient and the black nodes are for brown sugar, water, butter and salt that are side ingredients. Further layers show the cooking steps of these ingredients, node with id 1.6 is for the first step of recipe that is part of main process, other two nodes are for steps 2 and 3. The edge between nodes and steps show that which ingredients are the part of the step. In the same way, Recipe#02 is represented by FIGURE 4.8 and node properties are listed in TABLE 4.9.

Now we have two parents recipes encoded in the data structure that can be used for the crossover. To generate a new pair of recipes, we will check compatibility for each recipe in pair, and if pass the compatibility check, a new recipe is produced. TABLE 4.10 shows two child recipes in result of recombination of two parent recipes shown in TABLE 4.7. As **cookedType_mainIngredient** property of both of the recipes is **cooked**, so both of the children produced by these recipes are valid according to the validity conditions shown in TABLE 4.6. Child_01_02 consists of one main ingredient as of Recipe#01 and four side ingredients as of Recipe#2. The steps are interchanged in the same way. FIGURE 4.9 shows the graph representation of Child_01_02, and the TABLE 4.11 explains the property of nodes of the graph. To produce second child Child_02_01, main process is taken from the second recipe, so, its main process consists of two ingredients and two steps as the main process of second recipe and side process of the second child is as of side process of Recipe#01. FIGURE 4.10 is graph representation of second child Child_02_01 and properties of nodes are listed in TABLE 4.12.

Recipe#01	Recipe#02
Glazed Sweet Potatoes with Brown Sugar [21] Cook Time: 00:30, Servings: 4 **Ingredients:** • brown sugar, 0.5 cup • water, 0.33 cup • butter, 1 tablespoon • salt, 0.25 teaspoon • sweet potatoe, 3.0 units **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 [22]** Cook Time = 00:30, Servings = 6 **Ingredients:** • red potatoes, 2.5 lbs • garlic cloves, 3.0 units • butter, 2 tablespoons • milk, 0.5 cup • salt,1 teaspoon • parmesan cheese, 0.25 cup **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. Mash with the butter, milk, and salt 4. Stir in the parmesan cheese.

TABLE 4.7: Parent Recipes

Node	Type	Properties
1	Recipe[a]	Glazed Sweet Potatoes with Brown Sugar - 00:30 - 4 - Side Dish - Cooked
1.1	Ingredient[b]	sweet potatoes - 3 - units - main - main
1.2	Ingredient	brown sugar - 0.5 - cup - side - side
1.3	Ingredient	water - 0.33 - cup - side - side
1.4	Ingredient	butter - 1 - tablespoon - side - side
1.5	Ingredient	salt - 0.25 - teaspoon - side - side
1.6	Step[c]	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. - 1 - main
1.7	Step	In a heavy skillet, combine brown sugar, water, butter, and salt. Simmer over low heat for 5 minutes. - 2 - side
1.8	Step	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. - 3 - side

[a] Recipe(recipeName - cookTime - servings - category - cookedType-mainIngredient)
[b] Ingredients(ingredientName - quantity - measurementUnit - ingredientType - usedIn)
[c] Steps (description - stepNo - usedIn)

TABLE 4.8: Node Properties of Recipe#01

Node	Type	Properties
2	Recipe[a]	Mashed Red Potatoes With Garlic And Parmesan - 00:30 - 6 - Side Dish - Cooked
2.1	Ingredient[b]	red potatoes - 2.5 - lbs - main - main
2.2	Ingredient	garlic cloves - 3.0 - units - side - main
2.3	Ingredient	butter - 2 - tablespoons - side - side
2.4	Ingredient	milk - 0.5 - cup - side - side
2.5	Ingredient	salt - 1 - tablespoon - side - side
2.6	Ingredient	parmesan cheese - 0.25 - cup - side - side
2.7	Step[c]	Put potatoes and garlic in lg pan. Cover with water. Bring to a boil. - 1 - main
2.8	Step	Reduce heat and simmer for 25 minutes, until potatoes are tender. Drain well. - 2 - main
2.9	Step	Mash with the butter, milk, and salt - 3 - side
2.10	Step	Stir in the parmesan cheese. - 4 - side

[a] Recipe(recipeName - cookTime - servings - category - cookedType-mainIngredient)
[b] Ingredients(ingredientName - quantity - measurementUnit - ingredientType - usedIn)
[c] Steps (description - stepNo - usedIn)

TABLE 4.9: Node Properties of Recipe#02

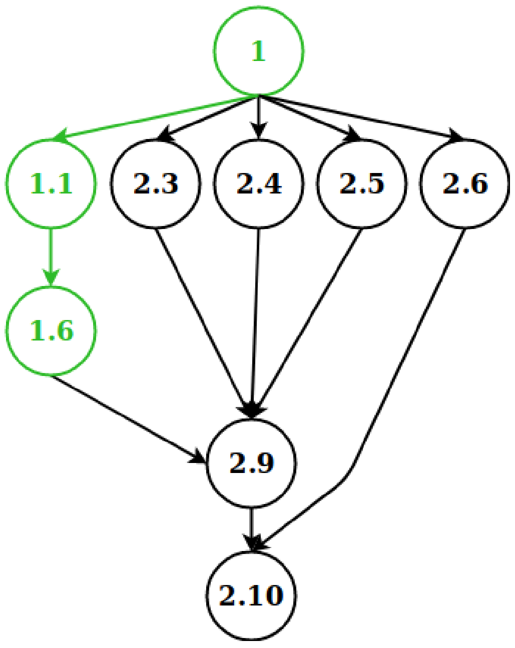


FIGURE 4.9: Graph Representation of First Child (Child_01_02)

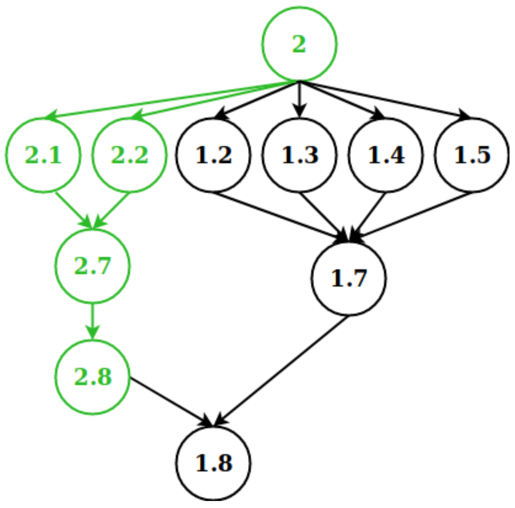


FIGURE 4.10: Graph Representation of Second Child (Child_02_01)

Child_01_02	Child_02_01
Glazed Sweet Potatoes with parmesan cheese Cook Time: 00:30, Servings: 4 **Ingredients:** • butter, 2 tablespoons • milk, 0.5 cup • salt, 1 tablespoon • parmesan cheese, 0.25 cup • sweet potatoes, 3.0 units **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** Cook Time: 00:30, Servings: 6 **Ingredients:** • red potatoes, 2.5 lbs • garlic cloves, 3.0 units • brown sugar, 0.5 cup • water, 0.33 cups • butter, 1 tablespoon • salt, 0.25 unit **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.

TABLE 4.10: Child Recipes

Node	Type	Properties
1	Recipe[a]	Glazed Sweet Potatoes with Brown Sugar - 00:30 - 4 - Side Dish - Cooked
1.1	Ingredient[b]	sweet potatoes - 3 - units - main - main
2.3	Ingredient	butter - 2 - tablespoons - side - side
2.4	Ingredient	milk - 0.5 - cup - side - side
2.5	Ingredient	salt - 1 - tablespoon - side - side
2.6	Ingredient	parmesan cheese - 0.25 - cup - side - side
1.6	Step[c]	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. - 1 - main
2.9	Step	Mash with the butter, milk, and salt - 3 - side
2.10	Step	Stir in the parmesan cheese. - 4 - side

[a] Recipe(recipeName - cookTime - servings - category - cookedType-mainIngredient)
[b] Ingredients(ingredientName - quantity - measurementUnit - ingredientType - usedIn)
[c] Steps (description - stepNo - usedIn)

TABLE 4.11: Node properties of Child_01_02

Generation of Recipe Name:

A newly generated recipe inherits the name from its first parent(from which it inherits its mains process). If the name of the first parent recipe contains the names of side ingredients, these are replaced with the side ingredients of second parent recipes(as the child recipe inherits side ingredients from the second parent). According to that, for Child_01_02, the name is taken from Recipe#01 and if it contains the name of a side ingredient from Recipe#01, this is replaced with a side ingredient of Recipe#02. Thus, the name that is generated is compatible with the content of the new recipe.

If the name taken from the first parent does not contain the name of side ingredients, two side ingredients from the second recipe are added at the end of the recipe name with the propositions "with" or "and." If the second recipe has only one side ingredient, only one side ingredient is added to the name. An example is shown below:

```
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
```

LISTING 4.1: Child name generation

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 these are

Node	Type	Properties
2	Recipe[a]	Mashed Red Potatoes With Garlic And Parmesan - 00:30 - 6 - Side Dish - Cooked
2.1	Ingredient[b]	red potatoes - 2.5 - lbs - main - main
2.2	Ingredient	garlic cloves - 3.0 - units - side - main
1.2	Ingredient	brown sugar - 0.5 - cup - side - side
1.3	Ingredient	water - 0.33 - cup - side - side
1.4	Ingredient	butter - 1 - tablespoon - side - side
1.5	Ingredient	salt - 0.25 - teaspoon - side - side
2.7	Step[c]	Put potatoes and garlic in lg pan. Cover with water. Bring to a boil. - 1 - main
2.8	Step	Reduce heat and simmer for 25 minutes, until potatoes are tender. Drain well. - 2 - main
1.7	Step	In a heavy skillet, combine brown sugar, water, butter, and salt. Simmer over low heat for 5 minutes. - 2 - side
1.8	Step	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. - 3 - side

[a] Recipe(recipeName - cookTime - servings - category - cookedType-mainIngredient)
[b] Ingredients(ingredientName - quantity - measurementUnit - ingredientType - usedIn)
[c] Steps (description - stepNo - usedIn)

TABLE 4.12: Node properties of Child_02_01

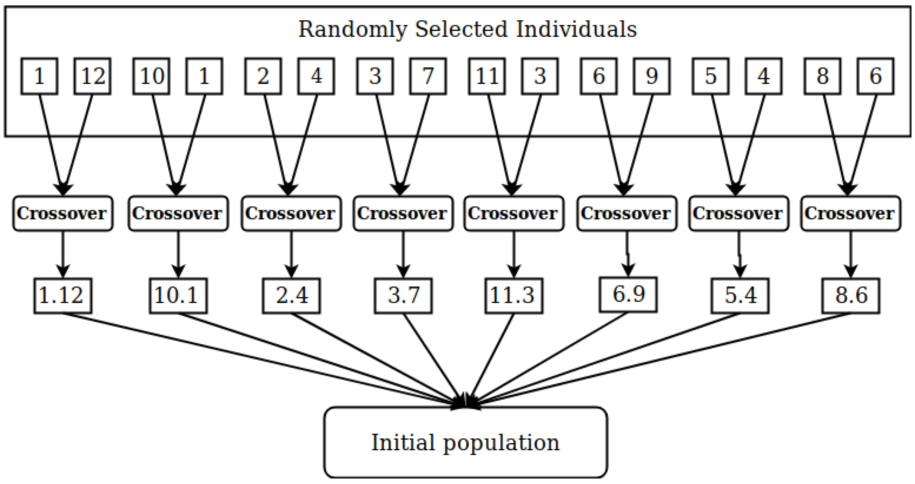


FIGURE 4.11: Initial Population

added into other recipes that may call potatoes by another name, to avoid confusion all types of potatoes are called simply "potatoes" in the instructions (but not in the ingredient list). This is because it is completely acceptable to refer "sweet potatoes" in the ingredient list as "potatoes" in cooking instructions. As shown in TABLE 4.10, in step 4 of Child_02_01, "sweet potatoes" is replaced with the "potatoes".

4.4.5 New generation

From the two child recipes generated from the crossover process, we choose the best one instead of including both in new generation. So, the next generation consists of better recipes than those of the previous generation. If crossover generates only one child, it is added to the next generation without evaluation and comparison to any other recipe.

4.5 Experimental Results

In this section, an example of the evolutionary algorithm explained in section 4.4 is presented. Our database consists of twelve recipes that can be found here[10]. The population size used in this experiment is eight.

4.5.1 Initial population

The initial population was generated by randomly picking 8 pairs of recipes from original recipes and intermixing these pairs to generate 8 children recipes. FIGURE 4.11 shows the process of generation of the initial population. Each recipe was assigned a unique ID that is a number between 1 and 12 to uniquely identify it. Two random numbers were generated to pick a pair of recipes from the database. To create eight individuals, the following eight pairs were generated:

(1,12),(10,1),(2,4),(3,7),(11,3),(6,9),(5,4),(8,6)

[10]http://www.machinegeneratedrecipes.de/original-recipes

Id	Recipe	Rating
1	Deviled garlic clove potato salad[a]	4.6
2	Sautéed potato with egg and green onions[b]	5
3	Glazed sweet potatoes with flour[c]	4.2
4	Emilys famous hash browns with onion and cumin[d]	3.9
5	Hungarian potato[e]	4.2
6	Roasted potato spinach and parmesan salad with garlic cloves and shredded mozzarella cheese[f]	3.9
7	Mashed red potatoes with garlic and parmesan with flour and vegetable stock[g]	4.2
8	New potato salad with light mayonnaise and dill recipe[h]	3.9

[a] `www.machinegeneratedrecipes.de/deviled-garlic-clove-potato-salad`
[b] `www.machinegeneratedrecipes.de/sauteed-potato-with-egg-and-green-onions`
[c] `www.machinegeneratedrecipes.de/glazed-sweet-potatoes-with-flour`
[d] `www.machinegeneratedrecipes.de/emilys-famous-hash-browns-with-onion-and-cumin`
[e] `www.machinegeneratedrecipes.de/hungarian-potato`
[f] `www.machinegeneratedrecipes.de/roasted-potato-spinach-and-parmesan-salad-with-garlic-cloves-and-shredded-mozzarella-cheese`
[g] `www.machinegeneratedrecipes.de/mashed-red-potatoes-with-garlic-and-parmesan-with-flour-and-vegetable-stock`
[h] `www.machinegeneratedrecipes.de/new-potato-salad-with-light-mayonnaise-and-dill-recipe`

TABLE 4.13: List of recipes in initial population

After taking the crossover of these pairs, a set of population consisting of 8 individuals was produced. TABLE 4.13 represents the list of recipes in the initial population. According to the process explained in FIGURE 4.2, after the creation of a new population, it is important to know if the algorithm should be stopped or further iterations should be performed. As we can see from the TABLE 4.13, five out of eight recipes have the good rating (greater than 4 out of 5) that are 62% of the total recipes, while we want to achieve 85% good recipes in current population before the termination of the algorithm. Next step is to create a new generation from the current one to improve the quality of the recipes.

4.5.2 First Generation

For the next iteration of the algorithm, individuals from initial population were selected using tournament selection. FIGURE 4.12 explains the process of creation of 1st generation from initial population. We picked three individuals randomly from initial population and checked their ratings. The individual with best rating out of these was selected as a parent. For the selection of two parents, the tournament was performed two times. Following pairs were selected from initial population for the further recombination:

(4,3),(8,6),(2,7),(6,5),(5,8),(1,1),(3,2),(7,4)

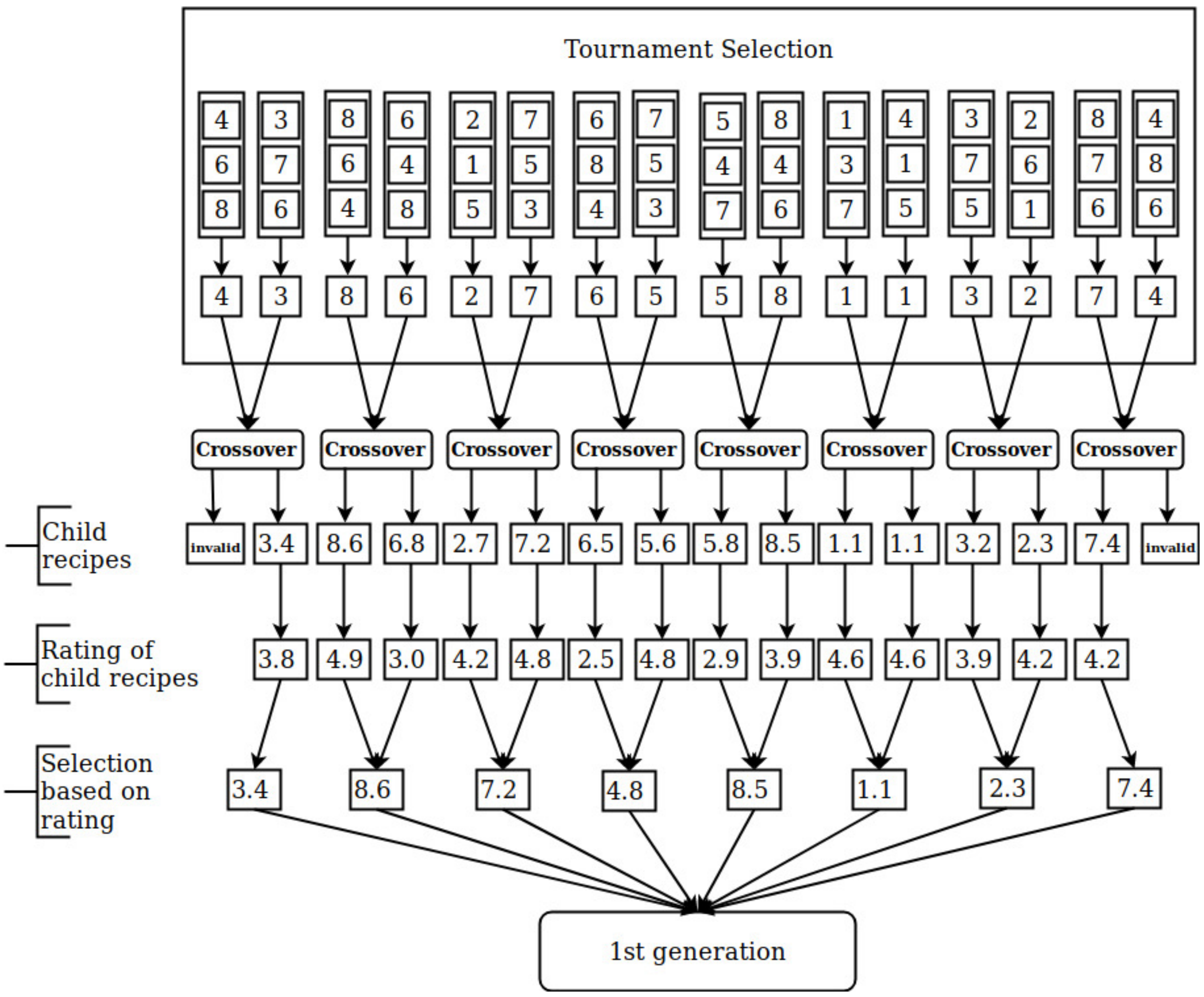


FIGURE 4.12: First Generation

After selecting the eight pairs of recipes, the crossover of these pairs was taken. In result of crossover, each pair generating one or two new recipes. All the possible combinations generated from selected pairs are following:

(4.3, 3.4) (8.6, 6.8) (2.7, 7.2) (6.5, 5.6) (5.8, 8.5) (1.1,1.1) (3.2, 2.3) (7.4, 4.7)

Now we checked for compatibility to skip invalid children and then checked rating of each child to select the best one. All the children created in 1st generation had following ratings:

(invalid,3.8) (4.9,3) (4.2,4.8) (2.5,4.8) (2.9,3.9) (4.6,4.6) (3.9,4.2) (4.2, invalid)

The first (4,3) and last (7,4) pairs generated one invalid child each. The third last pair contains same recipe as two parents that result into the generation of exactly same children as the parent recipe. After getting the ratings of each newly generated recipe, the recipe with the highest rating was selected out of these pairs to be the part of next generation. Invalid recipes were skipped, and the duplicate recipes were also not taken, instead, the other recipe with a low rating in that pair was taken. Here are the finally selected recipes in this iteration:

3.4, 8.6, 7.2, 4.8, 8.5, 1.1, 2.3, 7.4

The TABLE 4.14 shows our first population with the recipe names and ratings. The total 14 new recipes were created from the recombination of selected recipes from initial population. And rating of selected recipes was better than the overall rating of all the children. FIGURE 4.13 and 4.14 show the ratings of all and selected recipes respectively. In this iteration we were able to improve the ratings of our population that raised to 75% of good recipes from 62% in the initial population as shown by the green area in the graph shown in FIGURE 4.14.

4.5.3 2nd Generation

For the next iteration, the process of selection and recombination was repeated on 1st population. FIGURE 4.15 explains the creation of 2nd generation from recipes in 1st generation.

Following pairs were selected from 1st population for the further recombination: (3,2),(7,4),(3,6),(5,7),(2,8),(4,3),(7,6),(6,4)

All possible combinations:
(3.2, 2.3) (7.4, 4.7) (3.6, 6.3) (5.7, 7.5) (2.8, 8.2) (4.3, 3.4) (7.6, 6.7) (6.4, 4.6)

Respective ratings:
(4.9,4.5) (4.8,3.1) (4.8, no rating) (4.5,3.9) (4.2, 4.9) (4.9,4.9) (3.2,4) (3.9,3.9)

Selected children recipes:
3.2, 7.4, 3.6, 5.7, 8.2, 4.3, 6.7, 6.4

TABLE 4.15 shows our second population with the recipe names and ratings.
As the population size remains fixed in the evolutionary algorithm, we generated

Id	Recipe	Rating
1	Glazed sweet potatoes with onion[a]	3.8
2	New potato salad with garlic cloves and dill recipe[b]	4.9
3	Mashed red potatoes with garlic and parmesan with egg and green onions[c]	4.8
4	Hungarian potato garlic cloves[d]	4.8
5	New potato salad with and dill recipe[e]	3.9
6	Deviled garlic clove potato salad[f]	4.6
7	Sauteed potato with flour and vegetable stock[g]	4.2
8	Mashed red potatoes with garlic and parmesan with onion and cumin[h]	4.2

[a] `http://www.machinegeneratedrecipes.de/glazed-sweet-potatoes-with-onion`
[b] `http://www.machinegeneratedrecipes.de/new-potato-salad-with-garlic-cloves-and-dill-recipe`
[c] `http://www.machinegeneratedrecipes.de/mashed-red-potatoes-with-garlic-and-parmesan-with-egg-and-green-onions`
[d] `http://www.machinegeneratedrecipes.de/hungarian-potato-garlic-cloves`
[e] `http://www.machinegeneratedrecipes.de/new-potato-salad-with-and-dill-recipe`
[f] `http://www.machinegeneratedrecipes.de/deviled-garlic-clove-potato-salad`
[g] `http://www.machinegeneratedrecipes.de/sauteed-potato-with-flour-and-vegetable-stock`
[h] `http://www.machinegeneratedrecipes.de/mashed-red-potatoes-with-garlic-and-parmesan-with-onion-and-cumin`

TABLE 4.14: List of recipes in 1st population

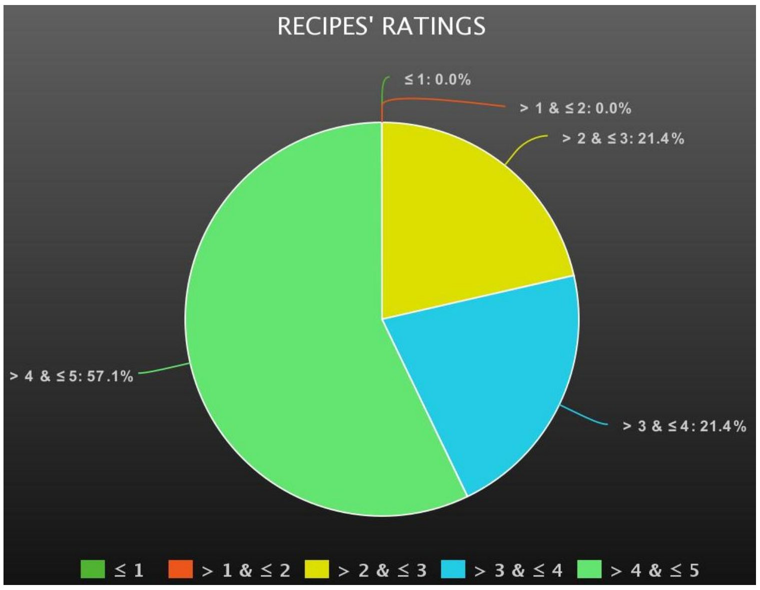


FIGURE 4.13: Recipe Ratings of all Recipes generated in the Fist Iteration

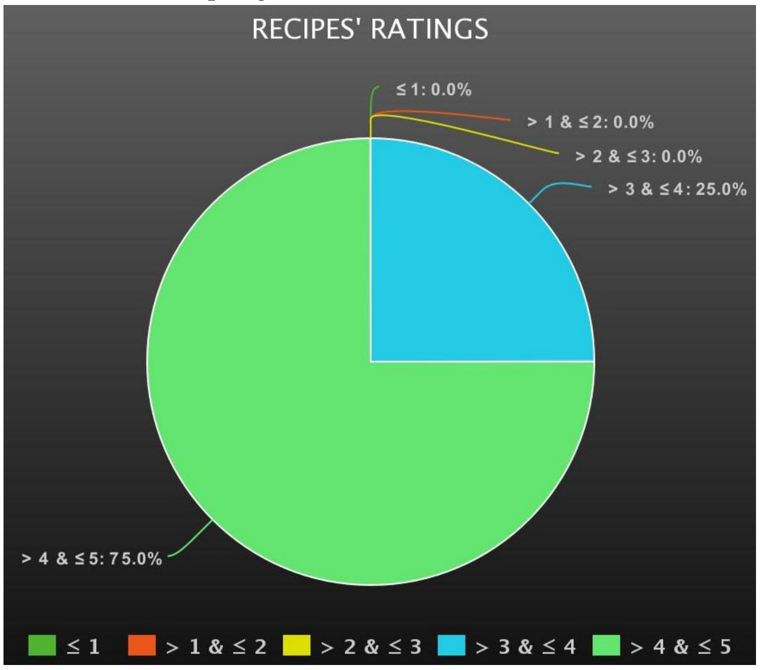


FIGURE 4.14: Recipe Ratings of the Selected Recipes in First Generation

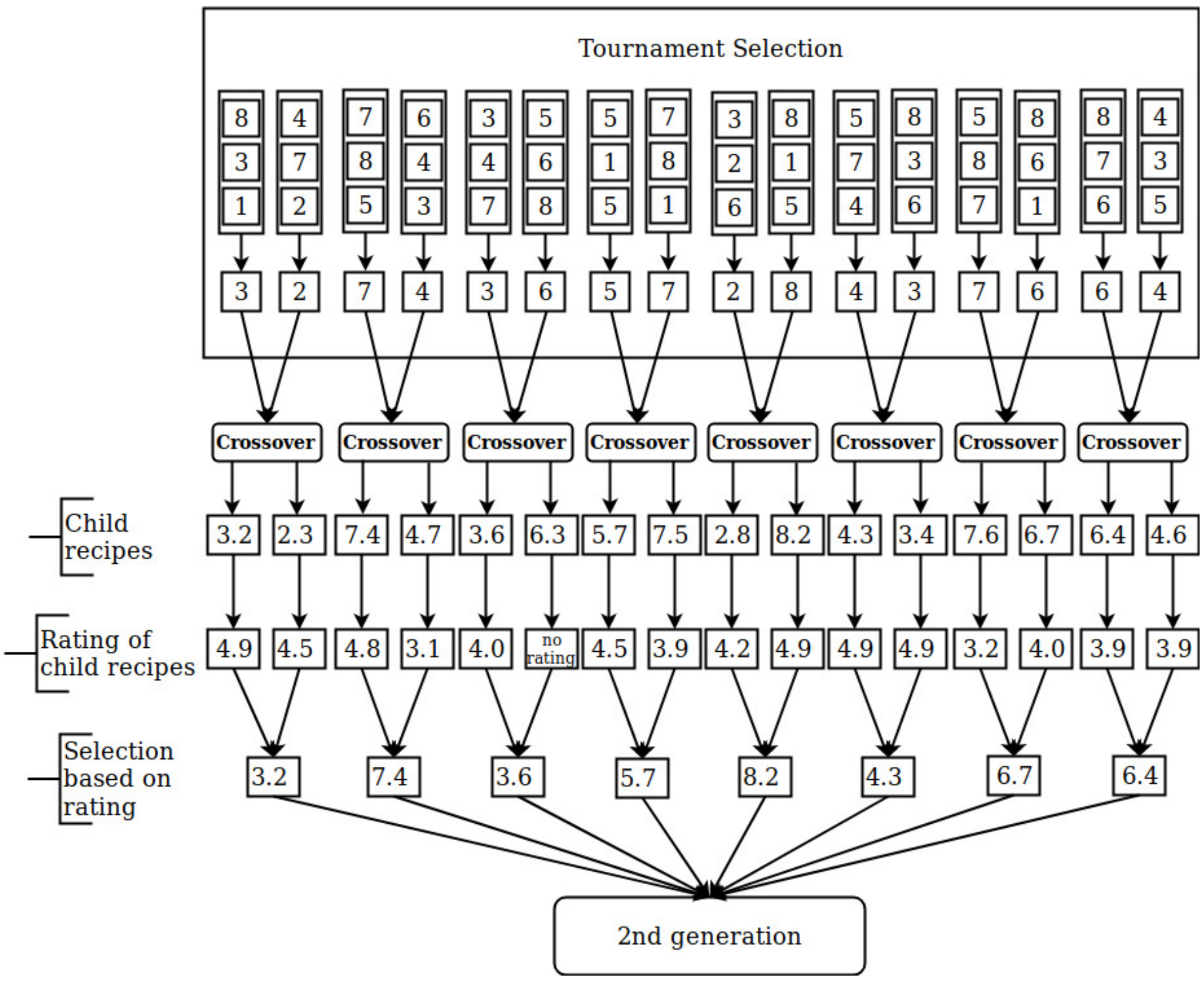


FIGURE 4.15: Second Generation

Id	Recipe	Rating
1	Mashed red potatoes with garlic and parmesan with garlic cloves and shredded mozzarella cheese[a]	4.9
2	Sauteed potato with garlic cloves and shredded mozzarella cheese[b]	4.2
3	Mashed red potatoes with garlic and parmesan with garlic clove and fresh parsley[c]	4.8
4	New potato salad with flour and dill recipe[d]	4.5
5	Mashed red potatoes with garlic and parmesan with garlic cloves and shredded mozzarella cheese[e]	4.9
6	Hungarian potato egg[f]	4.9
7	Deviled flour potato salad[g]	4
8	Deviled garlic cloves potato salad[h]	3.9

[a] `http://www.machinegeneratedrecipes.de/mashed-red-potatoes-with-garlic-and-parmesan-with-garlic-cloves-and-shredded-mozzarella-cheese`
[b] `http://www.machinegeneratedrecipes.de/sauteed-potato-with-garlic-cloves-and-shredded-mozzarella-cheese`
[c] `http://www.machinegeneratedrecipes.de/mashed-red-potatoes-with-garlic-and-parmesan-with-garlic-clove-and-fresh-parsley`
[d] `http://www.machinegeneratedrecipes.de/new-potato-salad-with-flour-and-dill-recipe`
[e] `http://www.machinegeneratedrecipes.de/mashed-red-potatoes-with-garlic-and-parmesan-with-garlic-cloves-and-shredded-mozzarella-cheese`
[f] `http://www.machinegeneratedrecipes.de/hungarian-potato-egg`
[g] `http://www.machinegeneratedrecipes.de/deviled-flour-potato-salad`
[h] `http://www.machinegeneratedrecipes.de/deviled-garlic-cloves-potato-salad`

TABLE 4.15: List of recipes in 2nd population

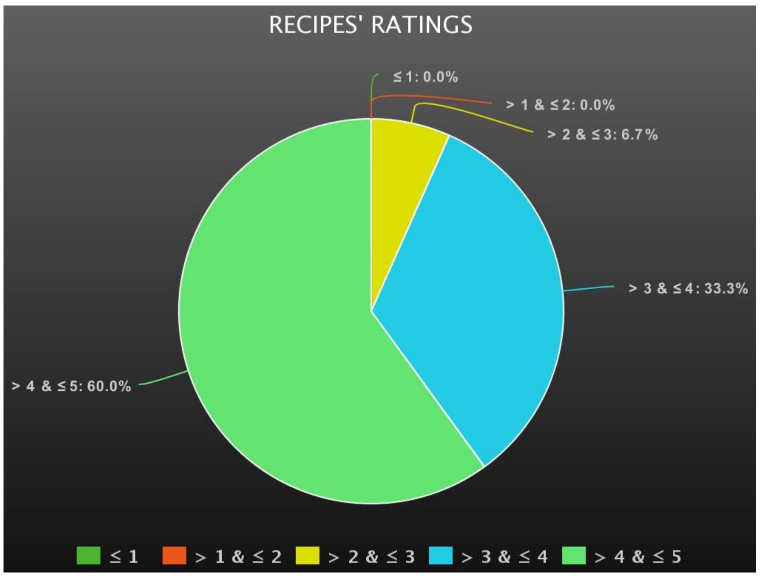


FIGURE 4.16: Recipe Ratings of all Recipes generated in the Second Iteration

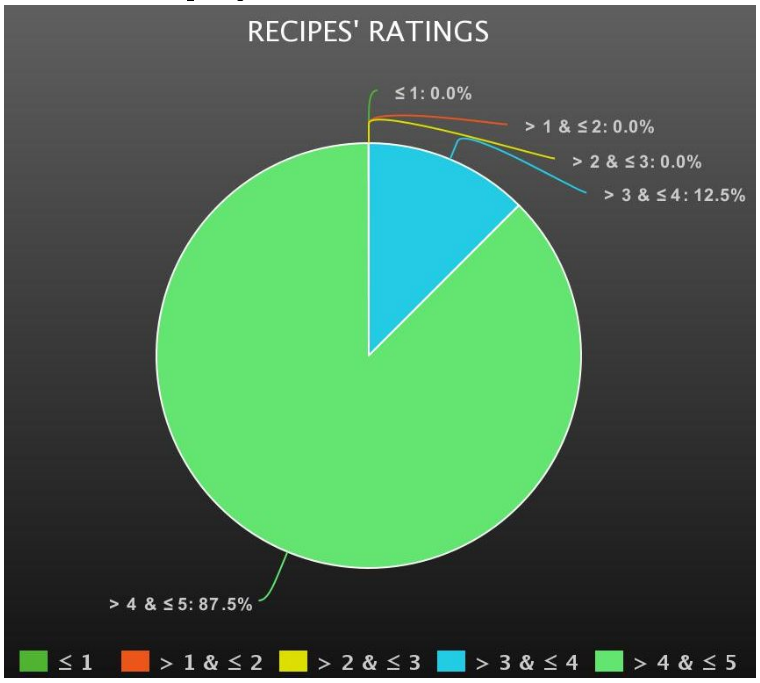


FIGURE 4.17: Recipe Ratings of the Selected Recipes in Second Generation

16 new recipes from the recombination of selected recipes from first population. FIGURE 4.16 and 4.17 show the ratings of all and selected recipes respectively. In this iteration we were able to improve the ratings of our population from 75% good recipes to 87.5% good recipes as shown by green area in the graph in FIGURE 4.17. After the second generation, we terminated the algorithm as we achieved our target of more than 85% good recipes in a population.

Chapter 5

Results

5.1 Improvement over iterations

In the experiment explained in section 4.5, we observed a continued improvement in the quality of the recipes after each iteration of the algorithm where the quality is determined by ratings of recipes assigned by users. FIGURE 5.1 shows the ratings of recipes in each generation.

In the initial population, we had 62% recipes with the good rating (greater than 4 out of 5), which increased to 75% in first generation. In the second generation, we had 87.5% good recipes and we achieved our target of getting 85% high rated recipes in the final population.

5.2 Original VS New Recipes

To compare the novelty in our newly generated recipes, we conducted a blind comparison of recipes[1]. We listed the original recipe with a new generated recipe from the original one, and users were asked to select a recipe that has more novelty. Three of twelve comparisons rated the original recipe as more novel, while eight of them rated new recipes more novel based on users vote. One of the recipes received equal votes for the original and new recipe. TABLE 5.1 shows the novelty scores received by each recipe, where Recipe#01 is original and Recipe#02 is the new recipe.

5.3 Overall rating

For the purpose of taking ratings of users in advance, we generated all possible combinations of recipes involved in the experiment(12 recipes) that resulted in 123 new recipes. The results were taken and total number of votes and final rating of the recipe was recorded that can be seen in Appendix A in more detail. The overall rating of these recipes revealed surprising results. Only four out of 123 recipes received the rating less than or equal to 2, from which 2 recipes got rating 1 and 2 got rating between 1 and 2. 19 recipes had the rating between 2 and 3. Rest of the recipes received good ratings, from which 48 had rating more between 3 and 4, while 52 recipes had the rating more than 4. TABLE 5.2 shows all these ratings that are further shown in graph form in FIGURE 5.2.
The recipes shown below are taken from each section of the graph shown in FIGURE 5.2.

[1]http://www.machinegeneratedrecipes.de/original-vs-new-recipes

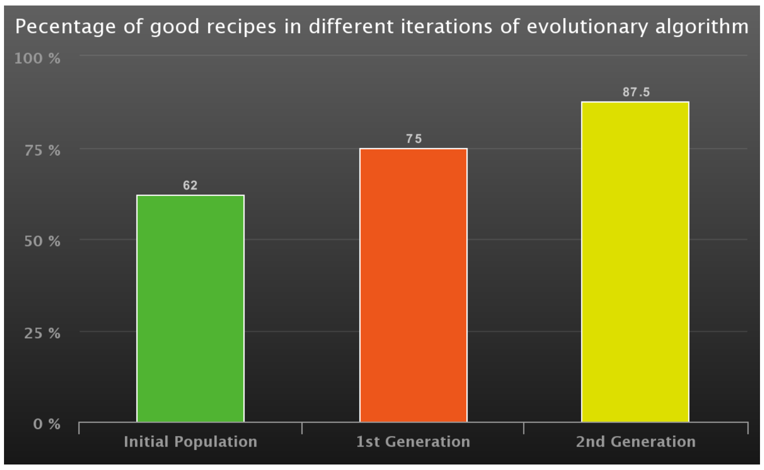


FIGURE 5.1: Improvement during Iterations

Id	Recipe#01	Recipe#02	Both are equal	I do not know
1	40.00%	50.00%	10.00%	0.00%
2	63.00%	25.00%	13.00%	0.00%
3	20.00%	80.00%	0.00%	0.00%
4	20.00%	80.00%	0.00%	0.00%
5	67.00%	33.00%	0.00%	0.00%
6	25.00%	75.00%	0.00%	0.00%
7	50.00%	50.00%	0.00%	0.00%
8	20.00%	80.00%	0.00%	0.00%
9	33.00%	67.00%	0.00%	0.00%
10	25.00%	75.00%	0.00%	0.00%
11	0.00%	75.00%	0.00%	25.00%
12	80.00%	20.00%	0.00%	0.00%

TABLE 5.1: Votes for the Novelty of Recipes

Ratings	Quantity
≤ 1	2
> 1 & ≤ 2	2
> 2 & ≤ 3	19
> 3 & ≤ 4	48
> 4 & ≤ 5	52

TABLE 5.2: Recipe Ratings of all New Recipes

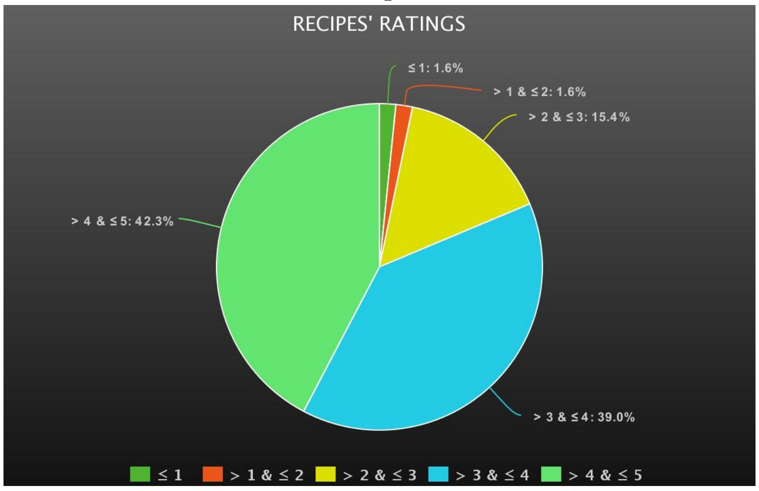


FIGURE 5.2: Recipe Ratings of all New Recipes

1- Recipe with rating between 4 and 5:

Baked potatoes with garlic cloves, mushroom and carrot cream[2]
Parent recipes: Baked potatoes with leek, mushroom and carrot cream [23], Baked Italian Potatoes [24]

Cook Time = 00:90, Servings = 4

Ingredients:

- baking potatoes, 6.0 large
- oil, 4.0 tbsp
- olive oil, 3.0 teaspoons
- baby portabella mushrooms, 0.2 cup
- scallions, 2.0 tablespoons
- garlic cloves, 3.0 unit
- dried oregano, 0.5 teaspoon
- tomato sauce, 0.5 cup
- shredded mozzarella cheese, 0.33 cup
- grated parmesan cheese
- fresh parsley, 0.25 teaspoon

[2]http://www.machinegeneratedrecipes.de/baked-potatoes-with-garlic-cloves-mushroom-and-carrot-cream

Instructions:

1. Preheat oven to 400°F. Place potatoes on a baking sheet and cut a cross in the top of each. Season then drizzle with oil and bake for 1 hour. Remove potatoes from oven and set aside to cool slightly.
2. Slice mushrooms.
3. Chop scallions.
4. Chop parsley.
5. Heat oil in small sauce pan. Add mushrooms, scallions, garlic, and oregano. Cook over medium to high heat for about 3 minutes.
6. Add tomato sauce to mushroom mixture. Cook, stirring frequently, for another 3 minutes.
7. Spoon tomato sauce over each of the potatoes and top with mozzarella cheese. Add a little Parmesan cheese if you'd like.
8. Place on baking sheet and bake at 450 degrees Fahrenheit for 5 to 8 minutes (until cheese is melted). Garnish with fresh parsley.

2 - Recipe with rating between 3 and 4:

Glazed sweet potatoes with flour[3]
Parent recipes: Baked potatoes with leek, mushroom and carrot cream [23], Glazed Sweet Potatoes with Brown Sugar [25]

Cook Time = 00:30, Servings = 4

Ingredients:

- sweet potatoe, 3.0 unit
- butter, 3.0 tbsp
- leek, 1.0 unit
- carrots, 3.0 unit
- mushrooms, 0.33 lb
- flour, 3.0 tbsp
- vegetable stock, 1.7 cup
- heavy cream, 0.75 cup
- Gouda cheese, 2.5 oz

Instructions:

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. Slice the leek in half lengthwise and cut into strips.
3. Peel and cube carrots.
4. Slice mushrooms in half
5. Grate cheese

[3]http://www.machinegeneratedrecipes.de/glazed-sweet-potatoes-with-flour

6. Melt 1 tbsp of butter in a saucepan, add leeks, carrots and mushrooms and sauté until soft. Remove from heat. Melt remaining 2 tbsp butter, add flour and cook briefly. Add stock and cream, bring to a boil and simmer for about 5 mins. Season to taste then add vegetables.
7. Open up potatoes and fill with sauce. Sprinkle with cheese and bake for 12 to 15 mins.

3 - Recipe with rating between 2 and 3:

Roasted potato spinach and parmesan salad with grated parmesan cheese and butter[4]
Parent recipes: Roasted Potato Spinach and Parmesan Salad [26], Mashed Red Potatoes With Garlic and Parmesan [27]

Cook Time = 00:55, Servings = 6

Ingredients:

- yukon gold potatoes, 2.0 lbs
- olive oil, 2.5 tablespoons
- dried rosemary, 0.75 teaspoon
- salt
- butter, 2.0 tablespoons
- milk, 0.5 cup
- salt, 1.0 teaspoon
- grated parmesan cheese, 0.25 cup

Instructions:

1. Cut potatoes into 3/4 inch wedges.
2. Crumble dried rosemary.
3. Preheat oven to 450 degrees F. On a large baking sheet, toss together potatoes, oil, rosemary, and 3/4 teaspoon salt.
4. Roast until potatoes are tender and golden, tossing once or twice, about 40 minutes; cool.
5. Mash with the butter, milk, and salt.
6. Stir in the parmesan cheese.

[4]http://www.machinegeneratedrecipes.de/roasted-potato-spinach-and-parmesan-salad-with-grated-parmesan-cheese-and-butter

4 - Recipe with rating between 1 and 2:

Baked italian potatoes with light mayonnaise and garlic clove[5]
Parent recipes: Baked Italian Potatoes [24], Roasted Potato Spinach and Parmesan Salad [26]

Cook Time = 00:35, Servings = 2

Ingredients:

- potatoes, 2.0 unit
- black pepper
- light mayonnaise, 0.5 cup
- lemon zest, 1.0 teaspoon
- lemon juice
- water, 1.5 tablespoons
- garlic clove, 1.0 unit
- Baby Spinach, 6.0 cups
- parmesan cheese, 0.5 cup

Instructions:

1. Preheat oven to 450 degrees. Bake potatoes in oven or microwave.
2. Split and fluff the baked potatoes.
3. Grate lemon zest.
4. Crush garlic clove.
5. Shave parmesan cheese.
6. In a large salad bowl, whisk together mayonnaise, lemon zest, lemon juice, water and garlic.
7. Add potatoes, spinach, and Parmesan cheese; toss.
8. Season with freshly ground black pepper.

5 - Recipe with rating 1:

Tibetan potato curry with egg and green onions[6]
Parent recipes: Tibetan Potato Curry [28], Deviled Egg Potato Salad [29]

Cook Time = 00:55, Servings = 6

Ingredients:

- potatoes, 6.0 cups
- egg, 5.0 unit
- mayonnaise, 0.75 cup
- plain greek yogurt, 0.5 cup
- yellow mustard, 2.0 tablespoon
- cayenne pepper
- green onions, 4.0 unit
- paprika

[5]http://www.machinegeneratedrecipes.de/baked-italian-potatoes-with-light-mayonnaise-and-garlic-clove
[6]http://www.machinegeneratedrecipes.de/tibetan-potato-curry-with-egg-and-green-onions

Instructions:

1. Precook the potatoes in water (or in the microwave) until almost, but not quite, done. Drain thoroughly.

2. At the same time, place eggs in another pot, and cover with an inch of water. Bring to a boil uncovered, and boil for three minutes. Cover the pot, and turn off heat completely for 10 minutes. Pour off the hot water, and soak the eggs in cold water until cool enough to peel.

3. Peel your cooked eggs, and cut them in half. Drop the yolks into the bowl your finished potato salad will go into.

4. Slice the green onions.

5. Add the mayonnaise, yogurt, mustard, salt, and pepper to the bowl with the egg yolks. Mash together with a fork until smooth. Stir in sliced green onions.

6. Chop egg whites, and add them to the mayonnaise mixture along with the cooked potatoes. Stir to coat. Sprinkle with paprika if desired. Serve immediately, or store in the refrigerator until ready to enjoy.

Chapter 6

Conclusion and Future work

6.1 Future Work

The proposed method produces complete and precise recipes. Currently, we have tested this technique only on potato recipes, but this can be extended to all kinds of recipes. To date, we have ignored certain relevant features, including flavor compound information (which can be helpful for decisions on ingredient pairing), nutritional information and the texture of the child recipes. Extending our approach to include this information is an avenue for future work. The following are some extensions that can be integrated into the current approach to make it more comprehensive.

6.1.1 N-point Crossover

In the crossover process, we are applying one-points crossover with a fixed point to generate a new recipe from two parent recipes. This fixed point is the point that separates the main process of a recipe from its side process. New offspring consists of the main process of one parent and the side process of another parent recipe. In that way, if we have N recipes in a generation, the maximum number of offspring recipes that can be produced are:

$$N * (N - 1)$$

Where N is the size of the population. The diversity of the population is also limited as in the child recipe, there is a fixed connected part of one chromosome (main process of one parent recipe) and a fixed consecutive part of the other (side process of second parent recipe). Application of multi-parent approach is also not possible with this structure as the new offspring can only have two parts including the main process and side process which can be taken from the maximum two parents. To extend this approach, we can divide the graph of the recipe into multiple parts where there are different recognizable sections for each side ingredient as well, instead of making all the side ingredients as the part of single side process. The FIGURE 6.1 shows an example of such structure which represent an italian breakfast recipe[1]. The green part in graph represents all the ingredients and spices which are part of main process, while each ingredient in the side process can be identified by it own block represented by purple (for the side ingredient "Hot Italian Sausages") and Blue (for the side ingredient "Egg") colours.

The structure shown in FIGURE 6.1 gives an opportunity to apply N-point crossover where N is 3 in this case. In general, N = (S + 2), where S represents the number of

[1]https://www.yummly.com/recipe/Italian-Sausage-Breakfast-Hash-1803020

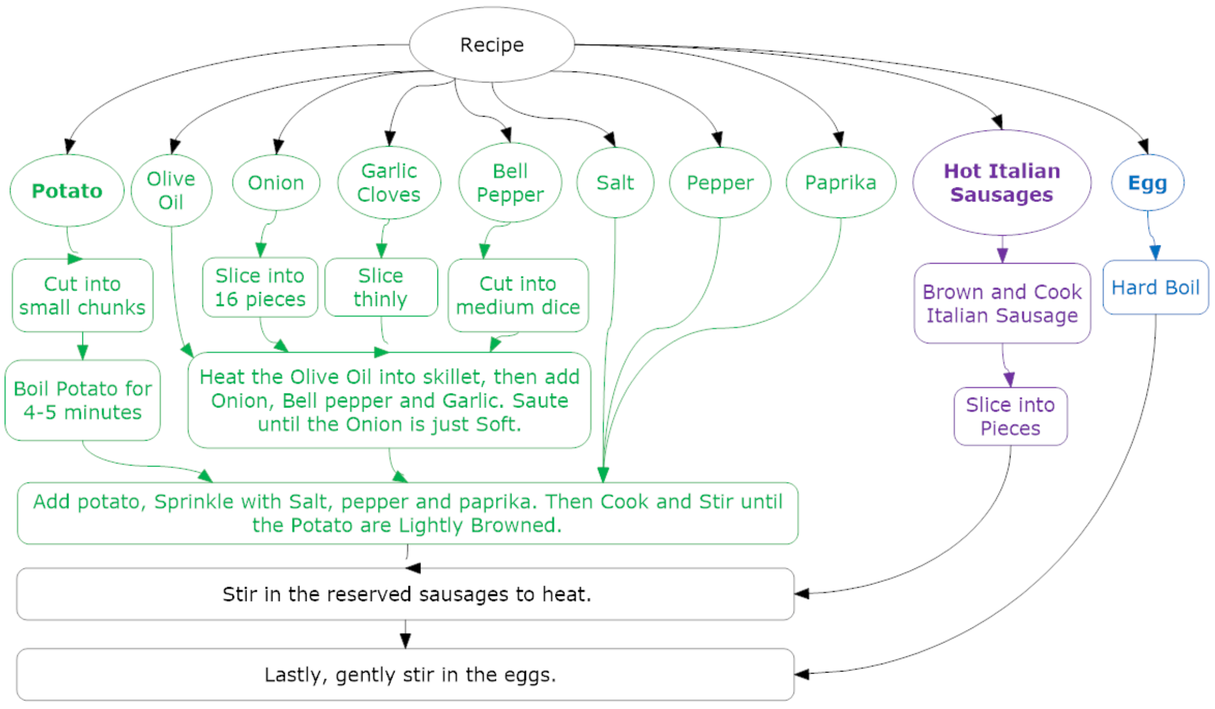

Recipe
Potato
Olive Oil
Onion
Garlic Cloves
Bell Pepper
Salt
Pepper
Paprika
Hot Italian Sausages
Egg
Cut into small chunks
Slice into 16 pieces
Slice thinly
Cut into medium dice
Brown and Cook Italian Sausage
Hard Boil
Boil Potato for 4-5 minutes
Heat the Olive Oil into skillet, then add Onion, Bell pepper and Garlic. Saute until the Onion is just Soft.
Slice into Pieces
Add potato, Sprinkle with Salt, pepper and paprika. Then Cook and Stir until the Potato are Lightly Browned.
Stir in the reserved sausages to heat.
Lastly, gently stir in the eggs.

FIGURE 6.1: Italian Sausage Breakfast Hash

side ingredients. Multi-parents approach can also be applied to this structure where children can take one or more blocks of ingredients from one parent recipe.

6.1.2 Adding mutation

The mutation is performed in an evolutionary process to alternate the newly generated offspring to make it different from its parent recipes. Mutation is an exploration of the search space to increase diversity of the process and is an important part of a genetic algorithm. It is performed after the inheritance (crossover) process as shown in FIGURE 6.2. The part of the mutation is missing in our evolutionary algorithm. It can be added in the following ways given the current structure of the recipes.

Ingredient Switching

After performing the crossover, one side ingredient can be replaced with a side ingredient of any randomly chosen recipe of the same generation. This alters the properties of children recipes, so they are different from their parent recipes. To switch an ingredient, we have to switch the entire block of the ingredient in the graph, which may also include the preprocessing and cooking instructions for that specific ingredient. FIGURE 6.3 represents a child recipe (named Recipe#01) in result of performing crossover on two parent recipes. Other than the main process, the recipe consists of one side ingredient represented by blue part of the graph. To apply mutation, we have chosen another recipe that we named Recipe#02 as shown in FIGURE 6.4. Recipe#02 consists of the main process and two side ingredients represented by blue and purple blocks. To replace one side ingredient of Recipe#01 with one side ingredient of Recipe#02, we replaced the blue block of Recipe#01 with the blue block of

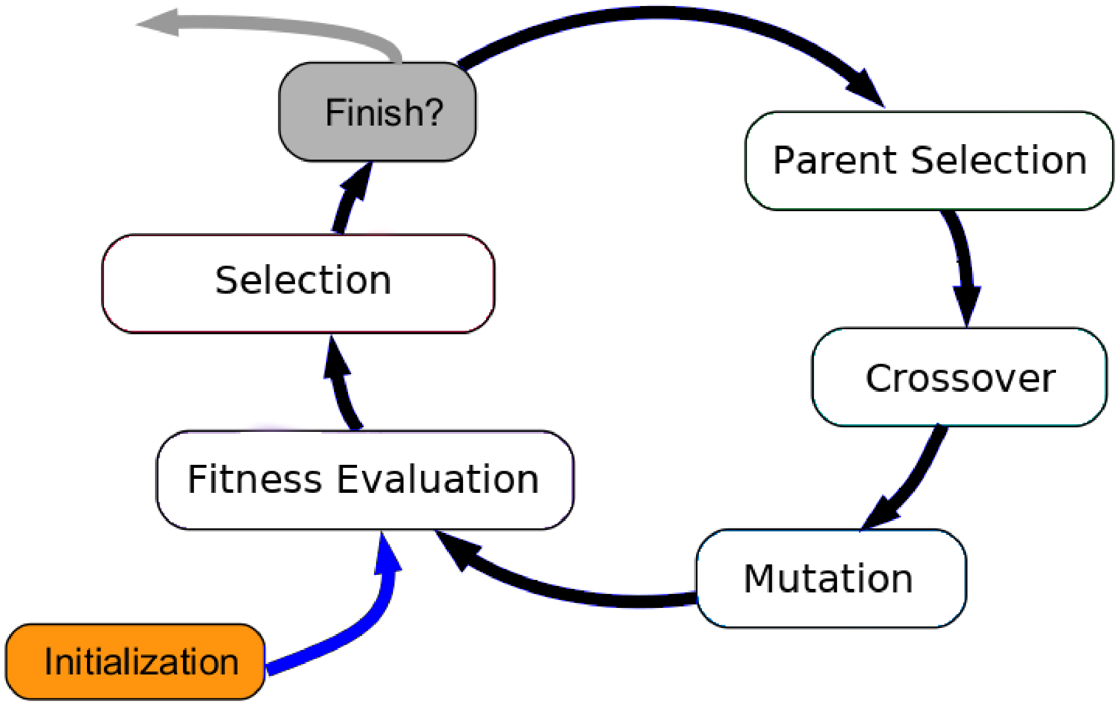


FIGURE 6.2: Genetic Algorithm Cycle

Recipe#02. The resultant recipe is shown in FIGURE 6.5 that consists of properties from its parents and some random properties.

Ingredient Substitution (without switching cooking instructions)

Teng et al. [15] has constructed an ingredient substitution network that can be used to replace an ingredient with another suitable ingredient without affecting the consistency of the recipe. One such ingredient substitute table has been compiled by Allrecipes.com[2]. After producing a new child, one or more of its ingredients can be replaced with the ingredients suggested by ingredient substitution network or an ingredient substitution table. In this case, we do not need to change the spices and cooking method of the ingredient as the substituted ingredient will most probably be well suited in the same cooking environment. TABLE 6.1 shows an example of ingredient substitution table that shows the one or more alternative of an ingredient. In the child recipe, one or more ingredients can be replaced to perform mutation. TABLE 6.2 presents an example of such replacement where onion is replaced by the leek.

6.1.3 Ingredient pairing

The ingredient pairing can be improved by using the flavour network. Currently, we are not considering if the combination of ingredients follows the rules of flavour network or not. This work is already done by Erol et al. [17] on the salad recipe.

[2]www.allrecipes.com

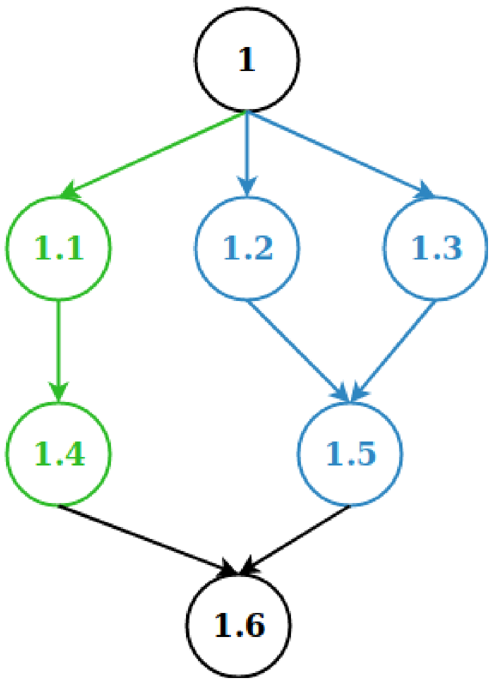


FIGURE 6.3: Graph Representation of Recipe#01 - the child recipe after crossover

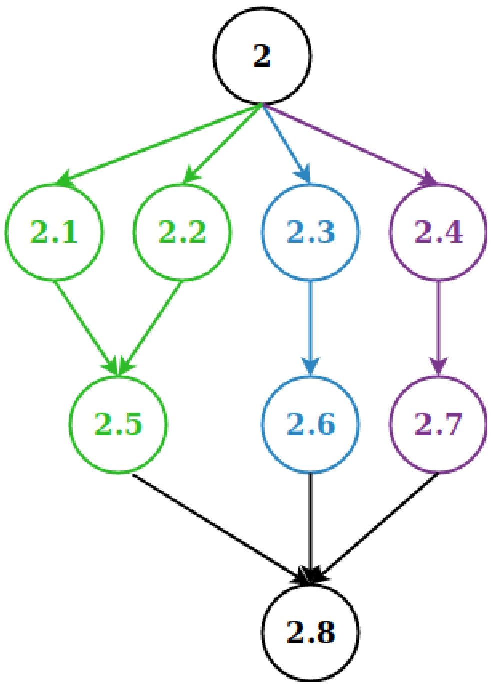


FIGURE 6.4: Graph Representation of Recipe#02 - a randomly chosen recipe from current population

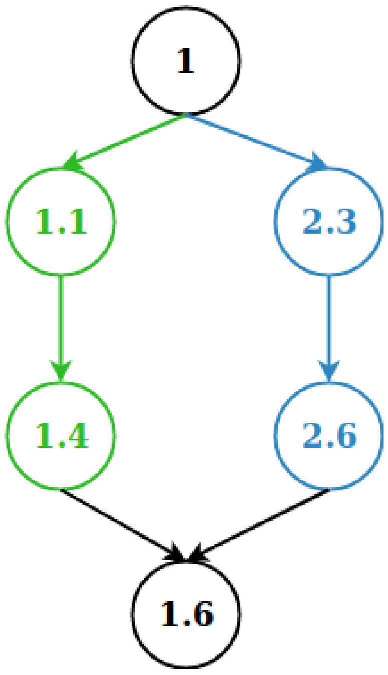


FIGURE 6.5: Resultant Recipe after Mutation

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 6.1: Sample ingredient substitute table

Their approach can be generalized to use on all type recipes and the current compatibility function can be extended with the restriction related to suitable ingredient combination.

6.1.4 Automatic recipe ranking

The recipe ranking technique developed by Teng et al. [15] can be used to evaluate the child recipes produced by the crossover process. This will automate the entire evolutionary process. As an evolutionary algorithm produces new recipes very quickly, but evaluating them is time-consuming. If the mutation is also the part of evolutionary algorithm, a little alteration in the recipe places a heavy burden of evaluation. If the ranking process could be made automatic with a high accuracy, this technique can produce recipes quicker and with more diversity.

6.2 Conclusion

Our work investigates the possibility of using an evolutionary algorithm in the cooking domain to develop a system that produces valid and novel recipes. The graph encoding of the recipes provided a powerful representation of the recipes to undergo the crossover operation. The compatibility function used for the validity check reduced the possibility of generation of unpleasant recipes, and we improved our set of recipes with each iteration. A test run of the algorithm produced a total of 123 recipes, from which more than 80% achieved good ratings (more than 3 out of 5) and 42% were rated excellent (more than 4 out of 5). Only 7% of the child recipes were rejected by the compatibility function and had to be dropped.

The results produced by EvoChef are complete and precise. While we have constrained ourselves to focus only potato recipes. This approach can be extended by

Original recipe [28, 29]	Recipe after mutation [28, 29]
Ingredients:	**Ingredients:**
.	.
.	.
~~onion~~	**leek**
.	.
.	.
Instructions:	**Instructions:**
1. Peel potatoes and Cut into 1 inches cubes.	1. Peel potatoes and Cut into 1 inches cubes.
2. Place potatoes in a pot, and cover with an inch of water. Add about a teaspoon of salt to the water. Bring to a boil, and cook at a low boil for 10 to 15 minutes or until your desired tenderness. Drain, and set cooked potatoes aside to cool.	2. Place potatoes in a pot, and cover with an inch of water. Add about a teaspoon of salt to the water. Bring to a boil, and cook at a low boil for 10 to 15 minutes or until your desired tenderness. Drain, and set cooked potatoes aside to cool.
3. Coarsely chop ~~onion~~.	3. Coarsely chop **leek**.
4. Peel and mince garlic.	4. Peel and mince garlic.
5. Coarsely chop tomatoes.	5. Coarsely chop tomatoes.
6. While the potatoes are cooking, saute the fenugreek seed in the oil on medium heat until light brown, being careful not to burn them.	6. While the potatoes are cooking, saute the fenugreek seed in the oil on medium heat until light brown, being careful not to burn them.
7. Add the ~~onion~~and continue cooking for five minutes. Add the ginger and garlic and cook another five minutes. Add the spices and saute briefly to release their flavors. Add the tomato, the dried whole peppers, and a little water. Simmer until the flavors meld together, about 30 minutes.	7. Add the **leek** and continue cooking for five minutes. Add the ginger and garlic and cook another five minutes. Add the spices and saute briefly to release their flavors. Add the tomato, the dried whole peppers, and a little water. Simmer until the flavors meld together, about 30 minutes.
8. Gently add the potatoes, stir, and reduce heat. Cook until potatoes are tender, adding water if the sauce gets too dry. If the sauce is too runny, simply crush one of the potatoes to thicken it.	8. Gently add the potatoes, stir, and reduce heat. Cook until potatoes are tender, adding water if the sauce gets too dry. If the sauce is too runny, simply crush one of the potatoes to thicken it.

TABLE 6.2: Ingredient substitution example

taking into account the ingredient pairing algorithm for ingredient combination and can be generalized for all kinds of recipes to develop a comprehensive system of automatic recipe generation.

Appendix A

Detailed result of recipe evaluation

Id	Recipe	Rating(number of votes)
1	Spanish tapas potatoes in garlic pasta	4.1(6)
2	Hungarian potato garlic clove	3.9(4)
3	Sauteed potato with garlic clove and fresh parsley	4.9(4)
4	Baked italian potatoes with garlic clove and fresh parsley	4.1(6)
5	New potato salad with garlic clove and dill recipe	4.1(3)
6	Tibetan potato curry with garlic clove and fresh parsley	3.1(4)
7	Roasted potato spinach and parmesan salad with garlic clove and fresh parsley	4.1(6)
8	Mashed red potatoes with garlic and parmesan with garlic clove and fresh-parsley	4.9(3)
9	Baked potatoes with garlic clove mushroom and carrot cream	2.9(5)
10	Glazed sweet potatoes with garlic clove	4.5(2)
11	Deviled garlic clove potato salad	4.6(5)
12	Spanish tapas potatoes in garlic chicken stock	4.8(3)
13	Hungarian potato chicken stock	4.5(2)
14	Sauteed potato with-pasta and pickles	4.8(4)
15	Baked italian potatoes with pasta and pickles	4(2)
16	New potato salad with pasta and dill recipe	3.5(4)
17	Tibetan potato curry with pasta and pickles	3.7(3)
18	Roasted potato spinach and parmesan salad with pasta and pickles 2	5(2)
19	Mashed red potatoes with garlic and parmesan with pasta and pickles	4.1(3)
20	Baked potatoes with pasta and carrot cream	3.8(3)
21	Glazed sweet potatoes with pasta	3.9(4)
22	Deviled pasta potato salad	3(2)
23	Spanish tapas potatoes in garlic garlic cloves	4.5(2)
24	Hungarian potato garlic cloves	4.8(3)
25	Sauteed potato with garlic cloves and shredded mozzarella cheese	4.8(3)
26	Baked Italian potatoes with chicken stock and butter	4(2)
27	New potato salad with chicken stock and dill recipe	4.5(5)
28	Tibetan potato curry with chicken stock and butter	4(4)

29	Roasted potato spinach and parmesan salad with chicken stock and butter	4.8(3)
30	Mashed red potatoes with garlic and parmesan with chicken stock and butter	4.1(3)
31	Baked potatoes with chicken stock and carrot cream	3.1(6)
32	Glazed sweet potatoes with chicken stock	4.1(4)
33	Deviled chicken stock potato salad	4.1(4)
34	Spanish tapas potatoes in garlic minced fresh dill	4(4)
35	Hungarian potato minced fresh dill	3(3)
36	Sauteed potato with minced fresh dill and apple cider vinegar	3.9(3)
37	Baked Italian potatoes with minced fresh dill and apple cider vinegar	3.1(9)
38	New potato salad with garlic cloves and dill recipe	4.9(5)
39	Tibetan potato curry with garlic cloves and shredded mozzarella cheese	4.2(5)
40	Roasted potato spinach and parmesan salad with garlic cloves and shredded mozzarella cheese	3.9(3)
41	Mashed red potatoes with garlic and parmesan with garlic cloves and shredded mozzarella cheese	4.9(4)
42	Baked potatoes with garlic cloves mushroom and carrot cream	5(4)
43	Glazed sweet potatoes with garlic cloves	4(6)
44	Deviled garlic cloves potato salad	3.9(6)
45	Spanish tapas potatoes in garlic onion	5(4)
46	Hungarian potato onion	2.5(2)
47	Sauteed potato with onion and cumin	4(3)
48	Baked italian potatoes with onion and cumin	3.9(6)
49	New potato salad with onion and dill recipe	4.1(4)
50	Tibetan potato curry with minced fresh dill and apple cider vinegar	4(7)
51	Roasted potato spinach and parmesan salad with minced fresh dill and apple cider vinegar	3.8(3)
52	Mashed red potatoes with garlic and parmesan with minced fresh dill and apple cider vinegar	3.8(7)
53	Baked potatoes with minced fresh dill mushroom and carrot cream	3.1(6)
54	Glazed sweet potatoes with minced fresh dill	4.1(3)
55	Deviled minced fresh dill potato salad	4.5(4)
56	Spanish tapas potatoes in garlic light mayonnaise	3(2)
57	Hungarian potato light mayonnaise	2.9(6)
58	Sauteed potato with light mayonnaise and garlic clove	3(2)
59	Baked italian potatoes with light mayonnaise and garlic clove	2(3)
60	New potato salad with light mayonnaise and dill recipe	3.9(9)
61	Tibetan potato curry with light mayonnaise and garlic clove	3(7)
62	Roasted potato spinach and parmesan salad with onion and cumin	3.1(6)
63	Mashed red potatoes with garlic and parmesan with onion and cumin	4.1(4)

64	Baked potatoes with onion mushroom and carrot cream	4(5)
65	Glazed sweet potatoes with onion	3.9(3)
66	Deviled onion potato salad	4.1(3)
67	Spanish tapas potatoes in garlic grated parmesan cheese	3.9(5)
68	Hungarian potato grated parmesan cheese	4.9(5)
69	Sauteed potato with grated parmesan cheese and butter	3.9(4)
70	Baked italian potatoes with grated parmesan cheese and butter	3.5(6)
71	New potato salad with grated parmesan cheese and dill recipe	3.5(2)
72	Tibetan potato curry with grated parmesan cheese and butter	3.1(3)
73	Roasted potato spinach and parmesan salad with grated parmesan cheese and butter	3(3)
74	Mashed red potatoes with garlic and parmesan with light mayonnaise and garlic clove	4.1(3)
75	Baked potatoes with light mayonnaise mushroom and carrot cream	3(3)
76	Glazed sweet potatoes with light mayonnaise	3(3)
77	Deviled light mayonnaise potato salad	4.5(2)
78	Spanish tapas potatoes in garlic flour	3(2)
79	Hungarian potato flour	3.1(4)
80	Sauteed potato with flour and vegetable stock	4.1(3)
81	Baked italian potatoes with flour and vegetable stock	4.5(6)
82	New potato salad with flour and dill recipe	4.5(8)
83	Tibetan potato curry with flour and vegetable stock	2.9(3)
84	Roasted potato spinach and parmesan salad with flour and vegetable stock	3.9(3)
85	Mashed red potatoes with garlic and parmesan with flour and vegetable stock	4.1(3)
86	Baked potatoes with grated parmesan cheese mushroom and carrot cream	3.9(9)
87	Glazed sweet potatoes with grated parmesan cheese	4.2(5)
88	Deviled grated parmesan cheese potato salad	4.1(8)
89	Spanish tapas potatoes in garlic	4(4)
90	Hungarian potato	4.1(9)
91	Sauteed potato	3.9(5)
92	Baked italian potatoes	1(2)
93	New potato salad with and dill recipe	3.9(4)
94	Tibetan potato curry	3.9(5)
95	Roasted potato spinach and parmesan salad	2.5(2)
96	Mashed red potatoes with garlic and parmesan	2.5(2)
97	Baked potatoes with and cream	3(5)
98	Glazed sweet potatoes with flour	4(2)
99	Deviled flour potato salad	4(4)
100	Spanish tapas potatoes in garlic brown sugar	4(3)
101	Hungarian potato brown sugar	3(3)
102	Sauteed potato with brown sugar	4.9(3)
103	Baked italian potatoes with brown sugar	3.1(4)
104	New potato salad with brown sugar and dill recipe	2(2)
105	Tibetan potato curry with brown sugar	4(1)

106	Roasted potato spinach and parmesan salad with brown sugar	4(2)
107	Mashed red potatoes with garlic and parmesan with brown sugar	4.5(2)
108	Baked potatoes with brown sugar and cream	2.1(4)
109	Glazed sweet potatoes	4.1(3)
110	Deviled potato salad	4(3)
111	Spanish tapas potatoes in garlic egg	3.8(3)
112	Hungarian potato egg	4.9(4)
113	Sauteed potato with egg and green onions	5(1)
114	Baked italian potatoes with egg and green onions	4.1(3)
115	New potato salad with egg and dill recipe	4.5(2)
116	Tibetan potato curry with egg and green onions	1(1)
117	Roasted potato spinach and parmesan salad with egg and green onions	5(2)
118	Mashed red potatoes with garlic and parmesan with egg and green onions	4.9(3)
119	Baked potatoes with egg mushroom and carrot cream	3.5(2)
120	Emilys famous hash browns with onion and cumin	3.9(5)
121	Glazed sweet potatoes with egg	4.1(5)
122	Roasted potato spinach and parmesan salad with pasta and pickles	4(1)

TABLE A.1: User-generated Rating of all Recipes

Bibliography

[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] Yong-Yeol Ahn, Sebastian E. Ahnert, James P. Bagrow, and Albert-László Barabási. Flavor network and the principles of food pairing. In *Scientific reports*, 2011.

[3] D Matic. A genetic algorithm for composing music. Yugoslav J. Operat. Res., 20: 157-177. DOI: 10.2298/YJOR1001157M, 2010.

[4] T. W. Manikas, K. Ashenayi, and R. L. Wainwright. Genetic algorithms for autonomous robot navigation. *IEEE Instrumentation Measurement Magazine*, 10(6):26–31, December 2007.

[5] Samriti . Applications of genetic algorithm in software engineering, distributed computing and machine learning. *International Journal of Computer Applications Information Technology*, 9, 01 2017.

[6] Xinjie Yu. *Introduction to Evolutionary Algorithms*. Industrial Engineering and Management Systems. 9. 1-1. 10.1109/ICCIE.2010.5668407, 2010.

[7] Introduction to Evolutionary Algorithms. `https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac`. [Online; accessed 19-august-2018].

[8] Overview of Apache Spark. `https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-overview.html`. [Online; accessed 20-august-2018].

[9] 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].

[10] Richard Brandt. Chef Watson has arrived and is ready to help you cook. `https://www.ibm.com/blogs/watson/2016/01/chef-watson-has-arrived-and-is-ready-to-help-you-cook/`, 2016. [Online; accessed 01-march-2018].

[11] Yahoo Food. Portuguese Lobster Roll Recipe from 'Cognitive Cooking with Chef Watson'. `https://www.yahoo.com/lifestyle/portuguese-lobster-roll-recipe-from-cognitive-116381787146.html?guccounter=1`, 2015.

[12] Terrence O'Brien. Portuguese Lobster Roll Recipe from 'Cognitive Cooking with Chef Watson'. `https://www.engadget.com/2015/06/12/cooking-with-watson-caymanian-plantain-dessert/`, 2015.

[13] Mark Wilson. IBM's Watson Designed The Worst Burrito I've Ever Had. https://www.fastcompany.com/3045147/ibms-watson-designed-the-worst-burrito-ive-ever-had, 2015.

[14] Guia Marie Del Prado. This is the weirdest recipe IBM's supercomputer chef has created. https://www.businessinsider.com/the-weirdest-recipe-ibms-supercomputer-chef-has-created-2015-8?IR=T, 2015.

[15] Chun-Yuen Teng, Yu-Ru Lin, and Lada A. Adamic. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 298–307, New York, NY, USA, 2012. ACM.

[16] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Statist.*, 28(2):337–407, 04 2000.

[17] Erol Cromwell, Jonah Galeota-Sprung, and Raghuram Ramanujan. Computational creativity in the culinary arts, 2015.

[18] threeovens. Roasted Potato Spinach and Parmesan Salad. http://www.geniuskitchen.com/recipe/roasted-potato-spinach-and-parmesan-salad-433702#activity-feed. [Online; accessed february-2018].

[19] Elmotoo. Tibetan Potato Curry. http://www.geniuskitchen.com/recipe/tibetan-potato-curry-137219. [Online; accessed february-2018].

[20] Jamilahs_Kitchen. Spanish Tapas Potatoes in Garlic Mayonnaise. https://www.geniuskitchen.com/recipe/spanish-tapas-potatoes-in-garlic-mayonnaise-369442. [Online; accessed february-2018].

[21] 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, 2018. [Online; accessed february-2018].

[22] 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].

[23] Recipes+. Baked Potatoes with Leek, Mushroom and Carrot Cream. http://recipes-plus.com/recipe/baked-potatoes-leek-mushroom-carrot-cream-19713?utm_campaign=yummly&utm_medium=yummly&utm_source=yummly.

[24] Dominick and Amanda. Baked Italian Potatoes. https://www.geniuskitchen.com/recipe/baked-italian-potatoes-290218.

[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.

[26] threeovens. Roasted Potato Spinach And Parmesan Salad Recipe. http://www.geniuskitchen.com/recipe/roasted-potato-spinach-and-parmesan-salad-433702#activity-feed.

[27] MizzNezz. Mashed Red Potatoes With Garlic and Parmesan. http://www.geniuskitchen.com/recipe/mashed-red-potatoes-with-garlic-and-parmesan-34382#activity-feed.

[28] Elmotoo. Tibetan Potato Curry. `http://www.geniuskitchen.com/recipe/tibetan-potato-curry-137219`.

[29] The Weary Chef. Deviled Egg Potato Salad. `https://www.yummly.com/recipe/Deviled-Egg-Potato-Salad-1709427`.