IMAT5314-1920-519 Project

Terms of Reference

James Izzard - P17177670

17/11/19

Rev	Date	Description
0001	17/11/19	Initially created.

Contents

---

1 Proposed Title

An Application of Computational Intelligence to Meal Optimisation

2 Background

Diet has a powerful influence on many physiological and psychological characteristics of people. Buying food often represents a significant portion of an individual's living costs. Therefore, many people attempt to optimise their diet against goals and within constraints to manipulate physiological, psychological and financial components.

Goals may include targeting particular macronutrient, micronutrient and energy content, and minimising the financial cost of ingredients. Since most ingredient combinations do not form viable meals, any optimisation is constrained to viable recipes and their respective tolerance to variation of ingredient ratios. Consequently, designing optimal recipes is a complex optimisation problem, and a specific case of the *knapsack problem* [2].

Human nutrition specialists typically charge a fee to apply experience, knowledge and heuristic methods to solve the problem. Computational intelligence techniques have been applied to derive satisfactory solutions to many incarnations of the knapsack problem. Most people dislike eating the same meals, so may frequently require new solutions. Additionally, diet optimisation goals may change as a function of body weight, training intensity, proximity to competition and other factors. Therefore, obtaining different solutions from a nutritionist can become expensive. Alternatively, people may attempt to create meal plans, which is time-consuming. The individual must consider the opportunity cost of this time.

The author proposes to apply computational intelligence techniques such as evolutionary algorithms, to derive solutions to the dietary optimisation problem. While the precise nature of the optimisation algorithm is currently undetermined, it is perhaps reasonable to outline the architecture as follows:

1. *Recipe Pre-Selection* - This may be a search heuristic or a brute force search, to determine which recipes in the database provide the most scope to meet the user's requirements.
2. *Recipe Manipulation* - This may be an evolutionary algorithm which attempts to tune the ingredient ratios within the pre-selected recipes, within specified tolerances, to approach the user's goals.
3. *Final Recipe Selection* - The best candidate from the population is selected. This stage may incorporate ingredient cost discrimination.
4. *Recipe Scaling* - All ingredient quantities within the candidate recipe are scaled by a constant factor to meet the energy content the user has specified for the meal.

The software would process the user's specification list, addressing each meal on each day and compiling results. A mechanism will prevent the software from consecutively repeating similar meals from similar specifications.

3 Proposed Solution Specifications

A brief review of the literature suggests that current dietary optimisation systems fall into two categories, which must be distinguished. The first category aims to inform the user of the diet they should be following, based on a description of their physical characteristics or medical requirements. The author will refer to this category as *dietary recommendation* for the remainder of the work. The second category aims to inform the user of the foods they should be eating to adhere to a particular diet. The work proposed here falls into the second category. The author will refer to this as *meal recommendation* for the remainder of the work.

The author proposes to create a diet planning application using the Python language. The user shall control the software via configuration files and terminal commands to maintain focus on the optimisation task. The software shall be capable of designing weekly meal plans, accommodating different dietary goals for each day. The configuration files will describe these goals. There shall be a non-deterministic factor to promote meal variation. If possible, the non-determinism should be controllable by tuning discrimination parameters. The efficacy of balancing optimisation against other properties probably relies on the software locating solutions sufficiently close to the actual global optima.

3.1 Software Inputs

The software shall accept macronutrient, micronutrient and caloric quantities describing the user's diet, on each day of the week. The user shall be allowed to omit specifications for macronutrient and micronutrient quantities. In these situations, the software shall fall back on EFSA dietary reference values [1]. The software shall accept a cost-sensitivity parameter, allowing the user to balance meal variation against the financial cost of ingredients. The software will operate on a database of recipes, to be populated gradually by the author throughout the project (see Section 9). Each recipe will specify a list of ingredients, ingredient ratios and their respective tolerable variations. The variation tolerance will describe the allowable variation of the ingredient mass against the total mass of the meal before the meal becomes impracticable. The macronutrient and micronutrient properties of each ingredient will be stored in a separate database, thus preventing information duplication across recipes including the same ingredient.

3.2 Software Outputs

The program shall output a text file presenting a summary of the optimised meals for each day of the week, a weekly ingredient shopping list, and estimated ingredient costs.

4 Academic Objectives

Primarily, the author hopes to develop the ability to apply computational intelligence techniques to real-world optimisation problems. The opportunity to practice architecting a non-trivial software solution is of particular value. Additionally, the author wishes to exercise the capability of applying research methods to current problems.

Furthermore, the process of developing and documenting solutions to a relatively complicated problem will provide opportunities to develop secondary skills required to support academic activity. These skills include academic writing skills, proficiency with the LaTeXtypesetting system and the Python programming language, and practice using numerous supporting software packages.

5 Background Research Objectives

The author proposes to conduct a review of the existing literature, directly related to the application of computational intelligence to dietary optimisation. Initial research suggests the literature approaches dietary optimisation from various directions and with numerous distinct motivations.

Conducting a review of the literature provides an opportunity to develop an understanding of the strengths and shortcomings of these approaches regarding their particular application to dietary optimisation. Moreover, since problems in optimisation typically belong to abstract classes, the work will provide an opportunity to develop an understanding of the effectiveness of different techniques on these classes.

In addition to the technicalities of dietary optimisation, the author wishes to examine the motivations and implications behind developing a practical solution to dietary optimisation. Specifically, assuming a practical solution is feasible:

- Where would the target markets lie?
- What would be the criteria for practicality?
- How would ingredient and recipe data be submitted?
- How could a single system meet the requirements of different markets, such as medical and athletic?
- How could the system be deployed?

6 Research Questions

The research will primarily concentrate on the technical aspects of developing a meal optimisation algorithm. Fundamentally, the author hopes to identify a suitable computational intelligence strategy. From here, additional evaluative questions will arise, regarding the relative strengths and limitations of candidate algorithm designs. These evaluative questions will, in turn, require answers to explanatory research questions, for example: 'Why can algorithm A often locate more optimal meals than algorithm B?'.

Collaterally to the technical aspects of development, the author has an interest in the opportunities for the commercial deployment of a meal optimisation program. As such, the work aims to address the comparative research question around the differences, from a user's perspective, between a qualified human nutritionist, and a software solution capable of emulating a human nutritionist. Furthermore, the work will begin to identify the most practical ways a user can describe their nutrition goals to a software system. As mentioned previously, at this stage, the system will not incorporate

a GUI component, but will still be capable of investigating quantitative and practical methods for the user to describe nutritional goals, and extending the recipe and ingredient database.

7 Resources Required

No non-standard resources are required.

8 Risk Assessment

Several risks could conceivably be associated with the finished software. These could lead a user to attempt to follow an inappropriate dietary recommendation. These scenarios include logical faults within the algorithm, incorrect goals and incorrect data in the ingredient database. The author can mitigate these risks by informing potential users that, at present, the software is for research purposes only, and the user should not implement meal recommendations without prior consultation with a qualified nutritionist. Furthermore, the program will be designed to insert a cautionary statement into the results output page.

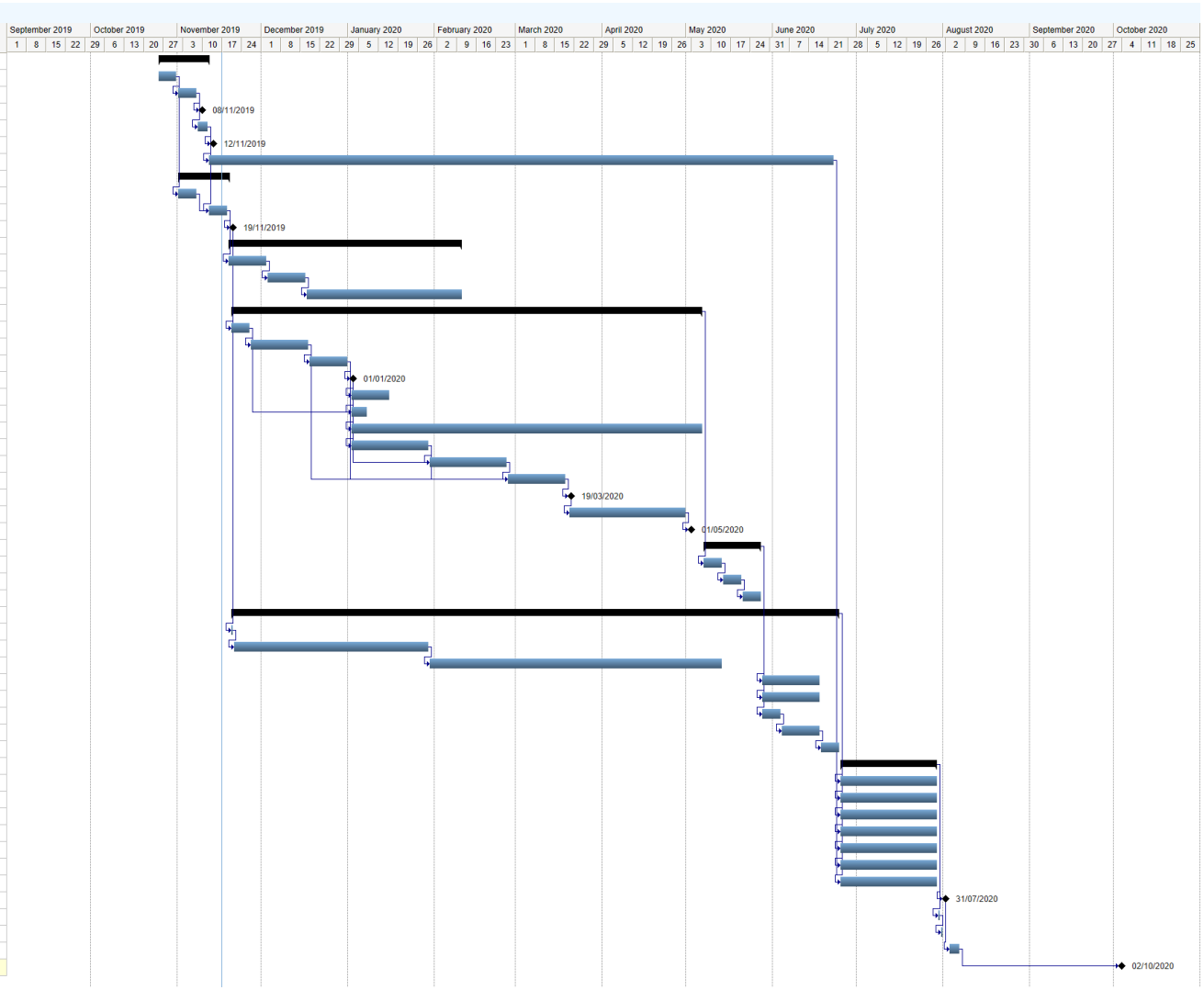
Additionally, risks to the project include the accidental loss of data due to information technology failure. The project data divides roughly into two categories: research documentation and code. To protect documentation from accidental loss, the author will take regular weekly backups, and utilise the version control and backup capabilities in Google Drive. To protect the code from accidental loss, the author will take regular weekly backups and utilise the cloud-based code repository, GitHub.

Poor time management could lead to progress falling behind, and the completed work being unavailable at the submission deadline. The author will mitigate this risk by creating and following a high-level progress plan (see Section 9). Additionally, the author will maintain an appropriate level of communication with the supervising lecturer. Specifically, this communication will include reporting progress and making work available, as appropriate.

9 Time Plan

Project Name **MSc Intelligent Systems & Robotics - Dissertation**

	Name	Duration	Start	Finish	Predecessors	Resources
1	**Project Selection**	**13days**	**25/10/2019**	**12/11/2019**		
2	Shortlist Candidate Project Ideas	5days	25/10/2019	31/10/2019		
3	Preliminary Literature Survey	1wk	01/11/2019	07/11/2019	2	
4	Discuss Potential Project With Supervisor	1day	08/11/2019	08/11/2019	3	
5	Draft Research Questions	2days	08/11/2019	11/11/2019	3	
6	Confirm Project Supervisor	1day	12/11/2019	12/11/2019	5	
7	Refine Research Questions	32wks	12/11/2019	22/06/2020	5	
8	**Terms of Reference**	**13days**	**01/11/2019**	**19/11/2019**		
9	Draft Terms of Reference	1wk	01/11/2019	07/11/2019	2	
10	Refine Terms of Reference	1wk	12/11/2019	18/11/2019	9,5	
11	Submit Terms of Reference	1day	19/11/2019	19/11/2019	10	
12	**Literature Survey**	**60days**	**19/11/2019**	**10/02/2020**		
13	Detailed Literature Selection	2wks	19/11/2019	02/12/2019	10	
14	Primary Readthrough of Selected Literature	2wks	03/12/2019	16/12/2019	13	
15	Secondary Notetaking on Selected Literature	8wks	17/12/2019	10/02/2020	14	
16	**Research and Development**	**121days**	**20/11/2019**	**06/05/2020**		
17	Prototype Recipe/Ingredient Databases	1wk	20/11/2019	26/11/2019	11	
18	Prototype Data Classes/Unit Tests	3wks	27/11/2019	17/12/2019	17	
19	Test Data Class and Database Function	2wks	18/12/2019	31/12/2019	18	
20	Finalise Recipe/Ingredient Database Structure	1day	01/01/2020	01/01/2020	19	
21	Choose/Design Recipe Pre-Selection Method	2wks	02/01/2020	15/01/2020	20	
22	Create Architectural Sketch	4days	02/01/2020	07/01/2020	20,18,19,17	
23	Populate Recipe/Ingredient Database	18wks	02/01/2020	06/05/2020	20	
24	Prototype Meal Optimiser Classes/Unit Tests	4wks	02/01/2020	29/01/2020	20	
25	Prototype Meal Optimiser Functions	4wks	30/01/2020	26/02/2020	24,20	
26	Integrate Prototype System	3wks	27/02/2020	18/03/2020	18,19,24,25	
27	Shakedown of Prototype Software	1day	19/03/2020	19/03/2020	26	
28	Test/Iterate/Refactor	6wks	20/03/2020	30/04/2020	27	
29	Complete Software Development	1day	01/05/2020	01/05/2020	28	
30	**Testing**	**15days**	**07/05/2020**	**27/05/2020**		
31	Design Functional Tests	1wk	07/05/2020	13/05/2020	16	
32	Functional Test Implementation	1wk	14/05/2020	20/05/2020	31	
33	Test Data Collection	1wk	21/05/2020	27/05/2020	32	
34	**Drafting Writeup**	**156days**	**20/11/2019**	**24/06/2020**		
35	Create Section Template	1day	20/11/2019	20/11/2019	11	
36	Draft Section - Introduction	10wks	21/11/2019	29/01/2020	35	
37	Draft Section - Literature Survey	15wks	30/01/2020	13/05/2020	36	
38	Draft Section - Optimisation Process	3wks	28/05/2020	17/06/2020	30	
39	Draft Section - Implementation	3wks	28/05/2020	17/06/2020	30	
40	Draft Section - Testing	1wk	28/05/2020	03/06/2020	30	
41	Draft Section - Discussion	2wks	04/06/2020	17/06/2020	40	
42	Draft Section - Conclusion	1wk	18/06/2020	24/06/2020	41	
43	**Refining Writeup**	**25days**	**25/06/2020**	**29/07/2020**		
44	Refine Section - Introduction	5wks	25/06/2020	29/07/2020	34,7	
45	Refine Section - Literature Survey	5wks	25/06/2020	29/07/2020	34,7	
46	Refine Section - Optimisation Process	5wks	25/06/2020	29/07/2020	34,7	
47	Refine Section - Implementation	5wks	25/06/2020	29/07/2020	34,7	
48	Refine Section - Testing	5wks	25/06/2020	29/07/2020	34,7	
49	Refine Section - Discussion	5wks	25/06/2020	29/07/2020	34,7	
50	Refine Section - Conclusion	5wks	25/06/2020	29/07/2020	34,7	
51	Finalise TEX Document	2days	30/07/2020	31/07/2020	43	
52	Draft Section - Abstract	1day	30/07/2020	30/07/2020	43	
53	Refine Section - Abstract	1day	31/07/2020	31/07/2020	52	
54	Proofread Writeup	4days	03/08/2020	06/08/2020	51	
55	Submit	1day	02/10/2020	02/10/2020	54	



PNG Generated On: 17/11/2019, 09:04:05

References

[1] European Food Safety Authority. Dietary reference values. https://www.efsa.europa.eu/en/topics/topic/dietary-reference-values, 2019. [Online; accessed 11-11-19].

[2] M. Shahpasand and S. A. H. Golpayegani. Knapsack problem. *Emerging Trends in ICT Security*, 2014. [Online: accessed 17-11-19].