Generating Dietary Patterns for Type II Diabetes Mellitus Out-Patients Using Constraint Logic Programming

ZYRA H. BRAVO, Ateneo de Davao University
MA. ANNE THERESE M. MUSONG, Ateneo de Davao University

Maintaining a diet is an important measure in reducing the risk of developing Type 2 Diabetes Mellitus patients. The goal of this study is to test the effectiveness of Constraint Logic Programming as an approach in solving the diet problem for type 2 Diabetes Mellitus out-patients, and to generate a suitable diet considering a patient's religion, allergies, activities and food preferences as the constraints. It also helps the patients in planning a combination of food and regulating their food consumption based on the calculated Total Caloric Requirements (TCR) by providing a generated list of food.

General Terms: Constraint Logic Programming, Type II Diabetes Mellitus, ECLiPSe-Java

Additional Key Words and Phrases: Diet optimization, dietary pattern, diet problem

1. INTRODUCTION

1.1 Background of the Study

One of the leading causes of death according to statistics is the Diabetes Mellitus. The said disease has two types, one of which is more common than the other and which also accounts a minimum of 75 percent of cases, and that is the type 2 (World Health Organization, 2008). Diabetes Mellitus is a condition where blood glucose levels are poorly controlled (Hunter, 2011). In the type 2, either the body does not produce enough insulin or ignore insulin, which is necessary for the body to be able to use glucose (Medline Plus, 2012). Thus, it requires regular monitoring and treatment. One of the treatments that can keep blood sugar levels close to normal and minimize the risk of developing complications includes self-care measures and sometimes medications (McCulloch, 2012), and risk reduction of the development of type 2 diabetes can be achieved by maintaining a healthy weight. Furthermore, studies also show that overweight people are at risk of the said disease (Hunter, 2011), in which food balancing is significant (Diabetes Meal Plans and a Healthy Diet, n.d.).

The main goal of a diabetic diet is to help maintain blood glucose levels. Eating a diabetic diet can also help in controlling blood glucose levels, decreasing the incidence of high blood sugar. Attainment and maintenance of ideal weight is significant to control diabetes which is a major advantage of a diabetic diet. According to Dr. Tim Harlan, research suggests low-carbohydrate diets may actually increase one's risk of developing diabetes. If a person already has the said disease, consuming a less restrictive food plan that includes moderate amounts of carbohydrates, proteins and fats can help lose weight. Having to control blood glucose levels requires the person to avoid consuming foods which may be a disadvantage to some individuals (Banar, 2011).

There are several approaches in generating a diet. Some of these are the Linear Programming and the Case-Based Reasoning approaches. The Linear Programming approach is the most commonly used method in generating a diet. It is an optimization of an outcome based on some set of the constraints (Wolfram MathWorld, 2013). It uses complex mathematical models as underlying structure for the food selection (Brien, Darmon, Ferguson and Erhdhart, n.d.). But the complexity of the approach uses a trial and error method of solving the problem which is difficult to find an optimal solution to a problem (Tutors On Net, 2009-2011).

The Case-Based Reasoning Approach is another way of solving the diet problem. In the study of Khan and Hoffman (2002) the method of generating a diet is to be based on the previous cases that are related to the current case and the solution is to be stored for future use. It incrementally improves by the direct expert user-system interaction. But the disadvantages of the approach are that it requires huge amount storage for the cases and it is difficult to acquire similar cases.

1.2 Problem Statement

This study wants to test the effectiveness of CLP as an approach in generating a diet for diabetes patients. Specifically, the study wants to generate a suitable diet for patients with Type II Diabetes Mellitus and consider allergies, religion and food preferences of the patient as constraints in the generation of the diet. Furthermore, the study aims to answer the following questions:

- How to generate a diet for diabetes out-patients? What are the parameters that affect the generation of the dietary pattern?
- How can CLP system generate an effective diet?
- What are the mechanisms of CLP that is important in study?
- How is CLP approach better compare to other approaches in the generation of a diet?

1.3 Research Objectives

The general objective of the study is:

- To develop a system using CLP as an approach in the diet generation.

The following are the specific objectives of the study:

- To identify constraints pertaining to the condition of an out- patient with Type II Diabetes Mellitus.
- To discover more about the characteristics of a CLP approach.
- To maximize the capability of CLP approach.
- To develop a helpful application in a using CLP approach.

1.4 Significance of the Study

Dietary pattern for Type II Diabetes Mellitus out-patients is important since food is one of the measures that should be controlled to prevent the development of the disease. Through this study, the patient will be able to find a suitable diet based from his/her given inputs into the system. It also gives the patient an option to include specific food but it strongly aims in producing a diet that satisfies the calorie requirement which is subdivided into the three important nutrients found in each food.

Furthermore, this study may be used as a foundation in solving similar diet optimization problems in the sense that it does not make use of a mathematical approach, but instead Constraint Logic Programming, which also, as the proponents observed, is not commonly used.

1.5 Scope and Limitations

The study covers the generation of the diet for the Diabetes out-patient. The Constraint Logic Programming approach will be used in the implementation of the study.

The study includes the most common type of Diabetes Mellitus which is the Type II. Hence, it excludes the generation of diet for other types of the Diabetes Mellitus and also for other types of diseases. The study does not include Diabetes out-patients with complications of other diseases. The study limits the generation of the diet for out-patient’s maintenance and not of the prevention of having the disease.

Moreover, the study does not intend to develop a quick running time of the system.

2. REVIEW OF RELATED LITERATURE

2.1 Diet Problem

The diet problem is one of the first optimization problems studied back in 1930's and 40's. Its goal is to find the cheapest combination of foods that will satisfy all daily nutritional requirements of a person. One of the early researches to study this problem was George Stigler who made an educated guess of the optimal solution to linear program using a heuristic method. Stigler's model was then solved using the simplex method by Jack Laderman.

The diet problem was verbally formulated by defining first its objectives like it has to minimize the cost of all the ingredients of the menu and the amount of nutritional requirements. The constraints were then added by defining the specific minimum and maximum values of a certain food that a person must consume. Constraints can be defined as input parameters. The decision variables were defined then in order to create a mathematical version of the model, wherein the costs were already specified. In the mathematical formulation, a set of foods and a set of nutrients were defined, and the amount of food and nutrients were assumed (NEOS, 2012).

2.2 Diet on Type II Diabetes Mellitus

Of the two types of Diabetes, the type 2 is much more common than the other type, accounting for at least 75 percent of cases. Cases in obese children and young adults are becoming more common for this type which implies that it is strongly related to being overweight (World Health Organization, n.d.). It is said that in the early stages of type 2 diabetes, diet is one measure that can often be enough to control and even reverse insulin resistance. Furthermore, diet and nutrition are widely believed to play an important role in the development of type 2 diabetes mellitus (Dam, Hu & Liu, 2001).

The American Diabetes Association (n.d.) has a so called diabetes meal plan which they defined as a guide for the diabetes patients in determining how much and what kinds of food they can choose to eat at meals and snack times. According to them, right meal plan helps in improving blood glucose and keeping weight on track, which is significant in the development of type 2 diabetes (Salzwedel, 2012).

In the study of Wheeler and Wheeler (n.d) for generating a diet for Diabetes patients they made use of Food Selection Algorithm that will combine the preferred food of the patient. They also implemented the randomization for the production of the food menu which may vary from day to day. The patient is allowed to eat a certain food given that the system will limit the food intake of the patient. The amount of each food that the patient can take is being calculated by the integer programming algorithm which aims to satisfy the diet prescription of the patient. In summary of their study, the system performs a repetitive complicated task which is to generate a diet plan that satisfies the patient's food preferences and at the same time satisfying the diet prescription for the patient.

2.3 Existing Approaches on Diet Calculation

This section shows the different approaches in calculating dietary patterns used in previous studies.

2.3.1 Linear Programming

Linear programming is one technique in solving the diet problem. It is defined as a tool to optimize a linear function of a set of decision variables while respecting multiple linear constraints.

Brien, Darmon, Ferguson and Erhdhart (n.d.) used linear programming in analyzing and optimizing children's diet during a complementary feeding program. In their study, they stated that linear programming provides an optimal solution if a diet is possible. It also specifies the types of foods that should be presented in balancing the diet. Linear programming is partly assigned to the complexity of mathematically modeling the underlying structure of food selection practices. However, even with this approach, diet optimization is still difficult due to the uncertainties with regards to the nutritional requirements. Furthermore, the quantity of food that the child can reasonably consume must be defined as a parameter which makes it difficult to estimate.

In the case study of Czyzyk and Wisniewski in 1996, they showed how the verbal formulation of the diet problem is converted into a mathematical problem or a linear program. The diet problem formulation showed a more mathematical formulation and the AMPL modeling language formulation as the solution that makes use of the World Wide Web as their medium.

Simplex algorithm, a popular algorithm for linear programming, was used in optimizing diet for overweight cardiovascular patients. The model was developed for determining the minimum objective function value, which is the minimum price, and content of meals in accordance with Dietary Reference Intake for set of nutrients. In the model, the constraints were ensured nutrients adequacy and minimum price, and can be used for designing diet for any other patients with restricted diet. The constraints also ensured diet palatability of meals, by which linear programming becomes a disadvantage in a way that it is unable to recognize organoleptic properties of food. Furthermore, the model sometimes offers unusual combinations of food in a meal. On the other hand, it was slightly modified and was used in preparing the menu for a specific number of diabetic patients and results were very satisfactory. However, simplex algorithm is not sensitive to organoleptic properties in which additional constraints should be added in the model for redirecting algorithm to acceptable solution (Pirički, Magdič, Perl & Žihlavski, n.d.).

In the study of Chien and Liu in 2004, they used the Mixed Integer Linear Programming approach (MILP), which is another type of linear programming. They wanted to find the optimal meal combinations of food items by evaluating meal desirability through assessing the associated food attributes. By constructing a multi-objective MILP model for a time scheduling of meals, they were able to come up with an optimal solution of a 5-day interval meal design under different weights. Using MILP in solving the problem, twelve different constraints are considered for designing a preferred menu.

2.3.2 Trial and Error

The diet problem can be solved by using the trial and error approach. The traditional trial and error approach, however, is iterative through which different food combinations based on informed guesses are repeatedly tried. Multiple backward steps and repeated diet designs are required to get a solution that may or may not be optimal. An adequate low-cost diet is also not provided. The trial and error method is time consuming and is also prone to error and creates an extra cause of improbability by manually solving a complex mathematical problem (Briend, Darmon, Ferguson & Erhardt, n.d).

2.3.3 Case-Based Reasoning Using Ripple-Down Rule

Khan and Hoffman (2002) based their approach on Case-Based Reasoning (CBR) in solving the diet problem. They develop this approach where the system can incrementally improve by direct expert user-system interaction; the expert can modify the system-generated diet and integrate it to the knowledge base (KB). The study produced a high level of accuracy in generating a diet of a person. They have concluded that the Ripple-down Rule (RDR) is a good knowledge acquisition approach in making a CDR system compared to the traditional approach.

2.3.4 Bayesian Networks

In the study made by Huang, Lu, Yang, Fu and Wang (n.d.) they made use of an integration platform built within the system to integrate all necessary tools to monitor the person's diet. One of these tools is the Activities of Daily Living (ADL) Inference Engine which is being implemented using the Bayesian Networks that access the inferred ADLs to be used in producing a list of diet suggestions. They gathered information of the person's daily nutrient requirement using the integrated tools. They concluded that the system is yet to be developed and is open for potential future enhancement to provide a more accurate and helpful information.

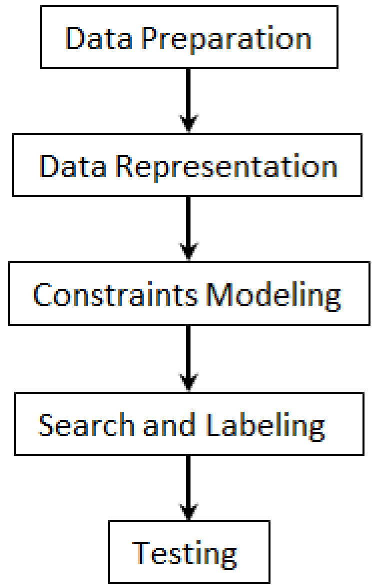
2.4 Constraint Logic Programming and Its Applications

The Constraint Logic Programming was introduced in 1987 by Jaffar and Lassez. This approach is commonly used to solve problems such as planning, scheduling and resource allocation. This approach uses the Constraint mechanism where the relations among variables are declared as the constraints and Logic Programming uses logic to declaratively state problems and deduction to solve them (Csontȯ and Paralič, n.d.).

A problem in discovering frequent pattern is solved using CLP. The problem concerns hidden patterns in large databases. It is one major problem in the areas of data mining. Furthermore, frequent pattern discovery is to determine how often a candidate pattern occurs. They compared the CLP and LP in terms of speed and memory usage in solving the problem and the results successfully showed that constraint-based implementation is slightly more efficient than the conventional logic programming (Kerdprasop & Kerdprasop, 2011).

In the study of Csontȯ and Paralič (n.d.) they used Constraint Logic Programming as an approach to solve the scheduling problem. The CLP approach combined the two different approaches namely the Constraint Solving that uses constraint mechanism where relations among variables are declared as constraints and Logic Programming that uses logic to declaratively state problems and deduction to solve them. Their proposed CLP model for the solution of the scheduling system worked well and it is able to generate all the possible desired outputs.

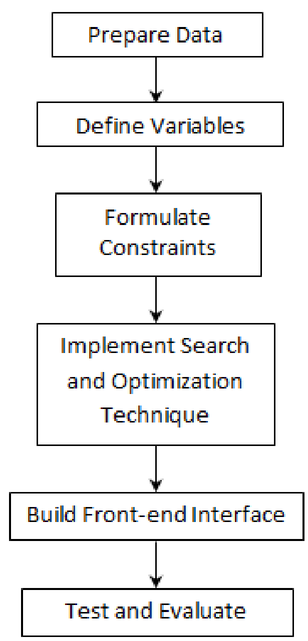
2.5 Theoretical Framework



Based on the related works, solving problems using the CLP approach was done first by preparing the data needed in the study. The next step is the data representation wherein the variables are created or represented. Constraints were then modeled which precedes the next step, that is the implementation of Search and Labeling. Once the implementation was done, the program was tested.

3. PROJECT DESIGN AND METHODOLOGY

3. 1 Conceptual Framework and Methodology



3.2 Methodology

1. Prepare Data
   - Gather all important data that will be used in the calculation of the diet.

2. Define Variables
   - Generate the variables that are to be used in the generation of the diet.
3. Formulate Constraints
   - Create constraints that will limit the possible output.
4. Implement Search and Optimization Technique
   - Implement a search technique that will give results according to the current case problem and filter the searched output which satisfies the constraints, and the optimization technique that will choose the best optimal result of the case problem.
5. Build Front-end Interface
   - Display the result of the implemented method of the calculation of the desired diet.
6. Test and Evaluate
   - Try and assess several cases to test the efficiency of the system.

4. THEORETICAL BACKGROUND

4.1 Constraint Logic Programming

Constraint Logic Programming (CLP) is a newly developed programming language that resulted from the combination of the two different programming paradigms, the Logic Programming and Constraint Solving. CLP inherit the declarative nature of Logic Programming and the efficiency of constraint handling of Constraint Solving. The combination of the two yields a very favorable result in terms with its development time and efficiency of the results (Frühwirth, et. al., 1993).

The use of constraints in CLP makes the approach declarative and direct and resulted in easy construction of simple but efficient programs. Since it is easy to construct a system in CLP, programmers can possibly experiment and modify the current system to find the best and fastest system (Rossi, n.d.).

4.1.1 Constraints

A constraint is a declarative expression that represents a relation among a subset of an object of a problem, where each object has its set of possible values, and must be satisfied (Rossi, n.d.). A constraint is evaluated in some fixed domain.

A decision procedure called the constraint solver is used to tell whether a constraint or a set of constraints is satisfiable or not. It is considered to be complete if it is able to decide the satisfiability of any set of constraints of the computation domain. The constraint solver should not start solving from scratch. It also needs to be incremental to achieve efficiency (Früwirth, et. al, 1993).

4.1.2 Constraint Solving

Constraint solving includes a diverse modeling frameworks and efficient solving tools for real-life problems that can be described through a set of variables and constraints. Constraint solving methods have been successfully applied to many application domains. Solving a problem with constraints means finding a way to assign values to all its variables such that all constraints are satisfied (Gavanelli & Rossi, 2010).

4.1.3 Logic Programming

Logic Programming (LP) is based from the idea that first order logic can be used for computing. It separates the logic from control. The logic part of the system is concerned in the correctness of the statements of the problem while the control is concerned in the efficiency of generating the solution. The separation of the two makes it easier to modify the existing problems.

But the disadvantage of LP is that it is not an efficient approach in solving complex types of problem because it uses the generate and test paradigm where the solutions are generated non-deterministically in choosing solution and then tested deterministically (Simonis, 2008).

5. RESULTS AND DISCUSSION

5.1 Gathering Patient's Data

The patient's information is the vital component in the study. Through the information given, the system will be able to produce results that are suitable according to the patient's condition. The process of gathering the patient's information is through giving out questionnaires to be filled out by random diabetic patients without the complications of other diseases. Items on the questionnaire include mainly the height, weight, gender and activities. The food preferences, allergies and religion of the patient are also considered. Each of the items in the questionnaire is important, especially of the height, weight, gender and activities, in the calculation of the Total Caloric Requirement of the patient in which the generation of the diet will be based mainly.

Inevitably, in the process of gathering the data there were information which turned out to be similar or closely identical to the other. To address or to lessen the testing of the similar data, one patient's data was chosen to represent all of its identical data. Furthermore, to minimize the discarding of a patient's data, initial interview was done first before giving out questionnaire to verify whether the patient's information has similar information to other patient.

5.2 Data

5.2.1 Patient's Data

The following are the data that were gathered from diabetes patients. We classified it into two: the first classification of data is for the calculation of the Daily Caloric Need of a particular patient and the second classification of data is for the constraining of a particular food to be included in the diet of the patient. The process and the formulas that were used to produce the Total Caloric Requirement of the patient are based from the data acquired during an interview to a dietician of the San Pedro Hospital and the manual provided.

The first classification of data is to be used in the calculation of the Ideal Body Weight (IBW) and Daily Caloric Need of a patient. First thing to do is to use the Tannhauser's formula where the height will be subtracted by 100, which is a constant value in the equation, and the result will be multiplied to 10% to obtain the IBW of the patient.

Tannhauser's Formula:

IBW (kg) = [Ht (cm) – 100] - [([Ht (cm) – 100]) *10%]

Second, calculate the TCR of the patient using the Daily Total Caloric Requirement formula. The calculated IBW will be multiplied to the corresponding kilo calories (kcal) of a specific activity level of the patient, which is obtained from the table below.

Calories for Activity Level		
Activity Level	**Male**	**Female**
Bed rest but mobile (Hospital patients)	30	27.5

Sedentary (mostly sitting, ex. Secretary, clerk, typist, administrator, cashier, bank teller)	35	30
Light (teacher, nurse, student, laboratory technician, housewife with maids, tailor, physician)	40	35
Moderate (housewife without maid, vendor, mechanic, jeepney driver, carpenter, painter)	45	40
Heavy (lumber man, farmer, laborer, cargador, coal miner, fisherman, heavy equipment operator, swimmer)	50	

Daily Total Caloric Requirement

TCR = Ideal Body Weight (IBW) x Activity Level

= (kg) (kcal/kg IBW)

Then, the height and the weight are to be used to get the BMI which is obtained by using the formula shown below:

$$\text{BMI} = \text{weight (kg)} / (\text{height (m)})^2.$$

BMI is used to classify whether the patient is obese, overweight or underweight. If the patient is overweight or obese, then the calculated TCR of the patient will be subtracted by 500-1000kcal else, it will be added by 500-1000kcal to the TCR (Rolfes, Pinna & Whitney, 2012). The range of the calories to be added or subtracted to the TCR depends on the dietician. For this study, 500kcal is used for all classifications as suggested by an expert.

Underweight	Add 500-1000 kcal	Below 18.5
Normal		18.5 to 24.9
Overweight	Subtract 500-1000 kcal	25 to 29.9
Obese	Subtract 500-1000 kcal	Above 29.9

And lastly, the calculated TCR will be subdivided into 60% for the Carbohydrates, 20% for the Protein and 20% for the Fat. And after the 60% is being obtained from the TCR for the carbohydrates, it will be divided by 4kcal/g to produce a result that is in a unit of grams, same goes with protein and fat, with 4kcal/g and 9kcal/g respectively.

Carbohydrates = [TCR * 60%] / 4 kcal/g
Protein = [TCR * 20%] / 4 kcal/g
Fat = [TCR * 20%] / 9 kcal/g

For the other classification of the patient's data, the following are going to be used to constrain the certain foods to be included in the diet. The religion or the sect where the patient belongs to is going to be considered for other religions restrict members to eat a particular food.

Referring to the data gathered from the patients, the researchers compiled a table where it listed the religious groups they belong to and listed also the corresponding food restrictions.

Religion	**Food Restrictions**

I. Islam	• Pork or pork products • Carnivorous animals (Butlersguild Food Restriction, n.d)
II. Christians	
a) Roman Catholic	• No food restrictions (A Dictionary of Patients' Spiritual & Cultural Values for Health Care Professionals, 2009)
b) Jehovah's Witness	• Blood (A Dictionary of Patients' Spiritual & Cultural Values for Health Care Professionals, 2009)
c) Baptist	• No food restriction (A Dictionary of Patients' Spiritual & Cultural Values for Health Care Professionals, 2009)
d) AnaBaptist	• No food restriction (A Dictionary of Patients' Spiritual & Cultural Values for Health Care Professionals, 2009)
e) Mormons	• Coffee • Tea • Alcohol • Large amounts of meat (CNN Eatocracy,2010)
f) Seventh Day Adventist	• Alcohol • Tea • Coffee (A Dictionary of Patients' Spiritual & Cultural Values for Health Care Professionals, 2009)
g) Iglesia ni Kristo	• Blood (iglesiaexposed.i8.com, n.d)

And lastly for the food allergies, the researchers categorized it into four namely the seafood, poultry, fruits and dairy products. The food items listed in the table are the ones that are only present in the Exchange List.

Categories	Food
I. Seafood	• Fish • Crab • Lobster • Shrimp • Shellfish
II. Poultry	• Chicken • Duck • Egg
III. Fruits	• Pineapple
IV. Dairy	• Milk • Cheese

5.2.2 Composition of Food Exchanges

In a food exchange list, foods are divided into seven groups. Each list contains the same amount of carbohydrate, protein, fat and calories for every exchange. There are certain amounts of nutrients for each specific measurement of a certain food group. The table below shows the list of food groups and the corresponding nutrient and caloric values taken from the Food Exchange Lists for Meal Planning (1994) manual of the Food and Nutrition Research Institute:

List	Food	Carbohydrate (g)	Protein (g)	Fat (g)	Energy (kcal)
I. A.	Vegetable A (1 exchange)	-	-	-	-
I. A.	Vegetable A (2 exchanges)	3	1	-	16
I. B.	Vegetable B	3	1	-	16
II.	Fruit	10	-	-	40
III.	Milk				
	Whole	12	8	10	170
	Low fat	12	8	5	125
	Skimmed	12	8	-	80
IV.	Rice	23	2	-	100
V.	Meat				
	Low fat	-	8	1	41
	Medium fat	-	8	6	86
	High fat	-	8	10	122
VI.	Fat	-	-	5	45
VII.	Sugar	5	-	-	20

However, the list shown above does not specify an exclusive meal plan for diabetics but can also be used for normal and therapeutic diets. So the Illustrated Food Choices for Diabetes (Balderamos & Que, n.d.) manual that San Pedro Hospital uses will also be used as another basis for the food groups intended for the diabetics, which does not include whole and low fat milk, and high fat meat and fish, although both manuals contain the same nutrient contents. The table as follows is used throughout the study:

List	Food	Carbohydrate (g)	Protein (g)	Fat (g)	Energy (kcal)
I. A.	Vegetable A (1 exchange)	-	-	-	-
I. A.	Vegetable A (2 exchanges)	3	1	-	16
I. B.	Vegetable B	3	1	-	16
II.	Fruit	10	-	-	40
III.	Milk				
	Skimmed	12	8	-	80
IV.	Rice	23	2	-	100
V.	Meat				
	Low fat	-	8	1	41
	Medium fat	-	8	6	86
VI.	Fat	-	-	5	45
VII.	Sugar	5	-	-	20

Each value shown correspond to one (1) exchange for a specific food group except for the Vegetable A subgroup which is negligible if it only has 1 exchange.

To get the patient's allowable exchanges for a day, all foods containing carbohydrates with the exception of rice—vegetables, fruit, milk and sugar—will be listed first. For vegetables A and B, it is desirable to allow 2-3 exchanges per day. For the fruit, 3-4 exchanges are allowable. The allowable amount and type of milk depends upon the patient's needs, food habits and other economic considerations. Based on the Food Exchange Lists for Meal Planning manual, 5-9 exchanges of sugar per day is allowed. However, according to a dietician, the usual practice is to allow only 5 exchanges for a diabetic patient which is just enough for palatability of meals since more than the said amount is already considered as high for a diabetic.

To determine the number of rice exchanges, the carbohydrates from vegetables, fruit, milk and sugar must be added. This sum will be subtracted from the prescribed carbohydrate in grams. The difference will then be divided by 23, which is furnished by 1 rice exchange. The nearest whole quotient is the number of rice exchange allowed.

To determine the allowable meat and fish exchanges, the protein furnished by the food groups already listed must be added. This sum will then be subtracted from the prescribed protein in grams. The difference will be divided by 8, furnished by 1 meat and fish exchange, and must be divided again by 2 for the low and medium fat distribution. The nearest whole quotient is the number of meat and fish exchange.

To get the fat exchange, the fat from low fat and medium fat meat and fish must be added and will then be subtracted from the prescribed fat in grams. The difference will be divided by 5 since one fat exchange contains 5 grams of fat.

The resulting exchanges will be distributed into breakfast, lunch, dinner, AM snack, PM snack and bedtime snack.

5.2.3Food Exchange List

Specific food combination is not fixed per meal as it depends upon the availability of food and the patient's data and preference. The Food Exchange List is the tool used in careful meal planning and diet instruction. It is also used in estimating carbohydrate, protein and fat of a meal. In the so called list, food items with similar nutrient contents are grouped together. Through the Food Exchange List, an item in a meal plan can be exchanged with another in the same food group. Throughout this study, the Food Exchange List in the Illustrated Food Choices for Diabetes (Balderamos & Que, n.d.) and Food Exchange Lists for Meal Planning (Food and Nutrition Research Institute, 1994) manuals were used. The following tables show the Food Exchange List:

Table 5.2.3.1: Vegetable A Exchange

1 exchange = 16 calories, 3 grams carbohydrates, 1 gram protein	
Vegetable A	**Measurement**
Uncooked Chinese Cabbage	25 grams
Uncooked Ampalaya Fruit	25 grams
Uncooked Cabbage	25 grams
Uncooked Cucumber	25 grams

Uncooked Lettuce	25 grams
Uncooked Petsay	25 grams
Uncooked Raddish	25 grams
Uncooked Tomato	25 grams
Cooked Chinese Cabbage	35 grams
Cooked Baguio beans	25 grams
Cooked Cabbage	35 grams
Cooked Cauliflower	25 grams
Cooked Alugbati	25 grams
Cooked Chayote Fruit	25 grams
Cooked Eggplant	35 grams
Cooked Okra	35 grams
Cooked Patola	35 grams
Cooked Papaya Green	35 grams
Cooked Petsay	35 grams
Cooked Tomato	35 grams
Cooked Upo	35 grams

Table 5.2.3.2: Vegetable B Exchange

1 exchange = 16 calories, 3 grams carbohydrates, 1 gram protein	
Vegetable B	**Measurement**
Ampalaya Guisado	45 grams
Chopsuey	45 grams
Sauteed Carrots	45 grams

Sauteed Squash	45 grams
Uncooked Carrots	40 grams
Cooked Squash	45 grams

Table 5.2.3.3: Fruit Exchange

1 exchange = 40 calories, 10 grams carbohydrates	
Fruit	**Measurement**
Mangosteen	65 grams
Marang	35 grams
Melon	200 grams
Grapes	55 grams
Santol	75 grams
Rambutan	50 grams
Singkamas	80 grams
Watermelon	140 grams
Fruit Cocktail	40 grams
Pineapple Crushed	60 grams
Dried Raisins	15 grams
Pineapple Unsweetened Fruit Juice	80 grams
Pineapple Sliced	30 grams
Banana Lakatan	40 grams
Banana Latundan	40 grams
Mango Green	65 grams
Mango Ripe	60 grams

Papaya Ripe	85 grams
Lanzones	70 grams
Pineapple	75 grams
Durian	30 grams
Guava	80 grams
Jackfruit	40 grams

Table 5.2.3.4: Milk Exchange

1 exchange = 80 calories, 12 grams carbohydrates, 8 grams protein	
Milk	**Measurement**
Soya Milk	250 grams

Table 5.2.3.5: Rice Exchange

1 exchange = 100 calories, 23 grams carbohydrates, 2 grams protein	
Rice	**Measurement**
Ampao	25 grams
Bibingka	45 grams
Biko	40 grams
Cassava Cake	45 grams
Kutsinta	60 grams
Puto Bumbong	45 grams
Sapin-sapin	75 grams
Suman Kamoteng Kahoy	45 grams
Pan de Sal	40 grams
Whole Wheat Bread	45 grams
Mamon Tostado	30 grams

Hopia Hapon	35 grams
Ensaymada	35 grams
Corn Boiled	65 grams
Corn Flakes	35 grams
Golden Corn Canned	145 grams
Cooked Noodles	75 grams
Sweet Potato	80 grams
Cassava	85 grams
Breakfast Cereals	25 grams
Cornstarch	25 grams
All Purpose Flour	25 grams
Cooked Sago	120 grams

Table 5.2.3.5: Low Fat Meat and Fish Exchange

1 exchange = 41 calories, 8 grams protein, 1 gram fat	
Low Fat Meat and Fish	**Measurement**
Aligue Alimango	15 grams
Aligue Alimasag	50 grams
Shrimps	25 grams
Squid	25 grams
Dried Fish Daing	20 grams
Lean Tocino	45 grams
Lean Meat Beef	30 grams
Lean Meat Carabeef	30 grams

Lean Meat Pork	30 grams
Liver Pork	35 grams

Table 5.2.3.6: Medium Fat Meat and Fish Exchange

1 exchange = 86 calories, 8 grams protein, 6 grams fat	
Medium Fat Meat and Fish	**Measurement**
Pork Leg	30 grams
Cheddar Cheese	35 grams
Chicken Wings	25 grams
Processed Sardines Tomato Sauce	45 grams
Processed Tuna Sardines	50 grams
Corned Beef	40 grams

Table 5.2.3.7: Fat Exchange

1 exchange = 45 calories, 5 grams fat	
Fat	**Measurement**
Peanut Butter	10 grams
Bacon	10 grams
Butter	5 grams
Cream Cheese	15 grams
Margarine	5 grams
Mayonnaise	5 grams
Sandwich Spread	15 grams
Sitsaron	10 grams
Whipping Cream	15 grams

Table 5.2.3.8: Sugar Exchange

1 exchange = 20 calories, 5 grams carbohydrates	
Sugar	**Measurement**
Banana Chip	5 grams
Bukayo	5 grams
Caramel	5 grams
Chewing Gum	5 grams
Condensed Milk	10 grams
Hard Candy	5 grams
Marshmallow	5 grams
Pastillas Durian	5 grams
Pastillas Langka	5 grams
Pastillas Gatas	5 grams
Sampaloc Candy	5 grams
Sugars	5 grams
Taho with Syrup and Sago	40 grams
Toffee Candy	5 grams
Yema	5 grams
Halo Halo	410 grams
Ice Candy	100 grams
Ice Drop	100 grams
Pulvoron	10 grams

These data are showed to emphasize the food items to be exchanged that are based only on the abovementioned manuals since it may vary from other food exchange lists. Also as mentioned it will be used as the reference list in the study particularly in the output generation.

5.2.4 Constraints

An optimal diet can only be produced when the following hard constraints will be satisfied:

1. Food allergies of the patient must be considered and must be excluded from the diet.
2. Based from the religious group that a patient is a member of, the food restriction enlisted in the table must be avoided.
3. It should be followed that there must be no repetition of food in a day.
4. If the patient prefers to have rice in his/her diet, the number of exchanges that a food named rice must only be within the range of 2 to 6 per day.
5. The food preference of the patient must be included in the diet.

5.2.5 Price

The prices are based on the present prices in market specifically in the Gaisano Mall Supermarket. Every food has a corresponding cost which is calculated from the given measurement. From the market price, the prices used in the input files are the equivalent values of the measurement.

To illustrate,

Given : Pineapple crushed costs P37.80 per 489g in the market
P37.80 / 489g = P0.08/g
P0.08 * 60 = P4.8 for the Pineapple crushed of 60g

5.3 Prepare Data

In this phase the data were prepared and formed it into Eclipse terms. Each Eclipse term is named after each Exchange List namely the vegetable_A, vegetable_B, fruit, milk, rice, low_fat_meat_and_fish, medium_fat_meat_and_fish, fat and sugar. Each term contains the arguments Foodname, Serving, Cost, MIN and MAX. And each list of terms is stored to the following eclipse files: vegetable_A.ecl, vegetable_B.ecl, fruit.ecl, milk.ecl, rice.ecl, low_fat_meat_and_fish.ecl, medium_fat_meat_and_fish.ecl, fat.ecl and sugar.ecl.

5.3.1 Argument

- Foodname refers to the first argument of each term which was taken from the different Exchange List.
- Serving refers to the amount of a specific food per serving that must be in the diet.
- Cost refers to the amount in Philippine peso of a specific food.
- MIN refers to a minimum value or amount of exchange that a specific food that must be included in the diet.
- MAX refers to a maximum value or amount of exchange that a specific food that must be included in the diet.

The following are sample terms of each Exchange List:

```
vegetable_A("uncooked chinese cabbage", 25, 1.34,1,1).
fruit("grapes", 55, 14.85,1,1).
low_fat_meat_and_fish("aligue alimango", 15, 7.5,1,1).
```

5.4 Create Variables

Every meal plan consists of specific food items that are required for the patient's daily consumption. These are stored in a list called `Required`. Each item in the list is compared with the terms from the `TermList`, which is broken down into heads and tails `[H1|T1]`, using the predicate `check_required`. As discussed in the previous phase, each term in the `TermList` has the arguments Min and Max which will determine the minimum and maximum number of exchanges of a certain food item. If the food item is found in the `TermList`, its corresponding variable in the `VarList` which is defined as `[H2|T2]`, will take the values `Min` and `Max` as its domains.

```
H2 #:: Min..Max
```

Otherwise, if the food item from the `TermList`is not found in the list Required then `H2` will be set to either 0 or 1.

```
H2 #:: 0..1
```

The setting of domains is performed in the predicate `set_domain` that is found in the module domain. The `set_domain` has the arguments `TermList`, `Required` and `VarList`.

```
set_domain(TermList, Required, VarList)
```

5.5 Formulate Constraints

For every identified constraint there is a corresponding Eclipse predicate which is to be evaluated and to be satisfied. The following are the predicates that have been constructed:

```
constrain_meals(TermList,VarList,Exclusion)
build_list(TermList,VarList,Required,[],ReqList)
constrain_preferences(TermList,VarList,Exchanges,Preferences,Required)
```

For the first and second constraints, it calls the predicate `constrain_meal` for the list of exclusion, which contains the allergies and the food that is restricted according to his/her religion, to be evaluated and satisfied. The said list of exclusion is created through the inputs made by the patient into the system. It then sets the domain using the following statement

```
H2 #:: 0..0.
```

The said constraints are the following:

1. Food allergies of the patient must be considered and must be excluded from the diet.
2. Based from the religious group that a patient is a member of, the food restriction enlisted in the table must be avoided.

Next, for the third, fourth and fifth identified constraints, a call to the predicate `constrain_preferences` is made that is to be executed and the following constraints will be satisfied. The goal of the main predicate is to constrain the food choices based from the preferences or input of the patient or the user. The first thing that the main goal does is to call the predicate `build_list` which builds lists namely the `PrefList` and `ReqList` by comparing the elements from the lists `Preferences` and `Required` to each element of each Exchange. Then, the two lists will be combined together and the elements from each list will be stored in the variable named `List` using the predicate `append`.The length of each list will be stored in the variables `Pref` and `Req` using the predicate `length`. Next is to obtain the maximum bound of each list and stores it to `PrefUp` and `ReqUp`. This is done through a call to the predicate `get_up_bound` for each list. The obtained values will then be evaluated to produce the `NPref`. It is done by multiplying the value of the variable `Pref` to the value of the variable `PrefUp` and same is done to produce the `NReq` using the variables `Req` and `ReqUp`. The variable named `List` will then be evaluated that is done through the following call

```
List = [].
```

It checks whether it is empty or not. If it is empty then the predicate `constrain_preferences` will be called again for the remaining List of Exchanges to be evaluated and if it is not then it will check whether the sum of `NPref` and `NReq` is greater than or equal to `H2`, which is the allowed number of exchange in each List of Exchanges, that is evaluated through the following call

```
(NPref + NReq) >= H2.
```

If it is greater than or equal to `H2` then a call to `set_sum` is made if it is not then the new variable `NewHeadExch` is made that contains the value of the sum of `NPref` and `NReq` which will be the second argument to the call of the predicate `set_sum`. Lastly, a call to predicate `constrain_preferences` is made for the remaining List of Exchanges.

The constraints that are being evaluated by this process are the following:

1. It should be followed that there must be no repetition of food in a day.
2. If the patient prefers to have rice in his/her diet, the number of exchanges that a food named rice must only be within the range of 2 to 6 per day.
3. The food preference of the patient must be included in the diet.

5.6 Search and Optimization Technique

To ensure that the diet that will be produced is an optimal one, search and optimization techniques are implemented.

5.6.1 Search

The searching technique that was used is done through a call to the predicate `search_lists`. The argument in the predicate consists of the list of domains of each list of exchanges. The domain that has a value equal to 1, which means that is allowed or preferred to be included in the diet, will be one of the possible selections that the predicate will randomly select from. After it has selected all the food items in each list of Exchanges it will then be the result and will be sent to the front end to be displayed as the output or the diet for the patient.

5.6.2 Optimization

The optimization technique that was used is made through a call to `get_cost` predicate. It checks whether the corresponding value from the `Varlist` is greater than 0. If it is, it gets the third argument from the terms which is the cost of a specific food and add it to the subtotal. After checking the domains of each term it combines all the calculated subtotal to produce the Cost of the meal of the day.

5.7 Build Front-End Interface

5.7.1 The Java-ECLiPSe Interface

In order to use the libraries or classes of the ECLiPSe in Java, the *com.parctechnologies.eclipse* package must be imported. It must be present in every Java source file that uses the Java-ECLiPSe Interface. Java interacts with ECLiPSe using the *EclipseConnection* and its subinterface *EclipseEngine*, which contains the methods. There are a number of classes which implements these interfaces. The instance of *EmbeddedEclipse* was used in the study. The initialization is done by creating and configuring an *EclipseEngineOptions* object. After the Java program has finished interacting with ECLiPSe, it is terminated by invoking the destroy method on the *EmbeddedEclipse* object.

5.7.2 ECLiPSe Goals Execution in Java

To execute the query from Java the `rpc` (Remote Predicate Call) method is invoked by passing it as an object which implements the *CompoundTerm* interface. In this case, the query becomes the goal. The goal is constructed using the *CompoundTermImpl*. Its parameters are the functor of the term followed by the objects. If one of the parameters is a variable, it is represented as *null* (Novello, et al., 2013).

The first call to `rpc` changes the directory of which the ECLiPSe files are found. It is again called to compile the main file named `generate`. The main query is passed as a *CompoundTerm* in the third `rpc` call. The first parameter of the term `generate` is a *Collection* of the nine input files. It is followed by another *Collection* which contains the number of exchanges of the nine food groups. The next parameter is a variable called `VarList` which is denoted as *null*. The succeeding three parameters are collections named `Required`, `Exclusion` and `Preferences`. The last one is a *Double* named `Cost` which gives the total cost of the generated set of food items.

The variable `VarList` is returned as a *Collection* and is converted to an Object array as `varlistArr`. Since it is a collection of lists, it is deconstructed such that each list in the collection is stored in a separate variable as a *String*. The variable `Cost` is returned as a list and then converted into a *Double*.

5.7.3 The Java User Interface

The graphical user interface is shown below. It has three panels: the first two contain the user input which must be completed before the results are generated, and the third panel contains the results that are generated based from the input given by the user.

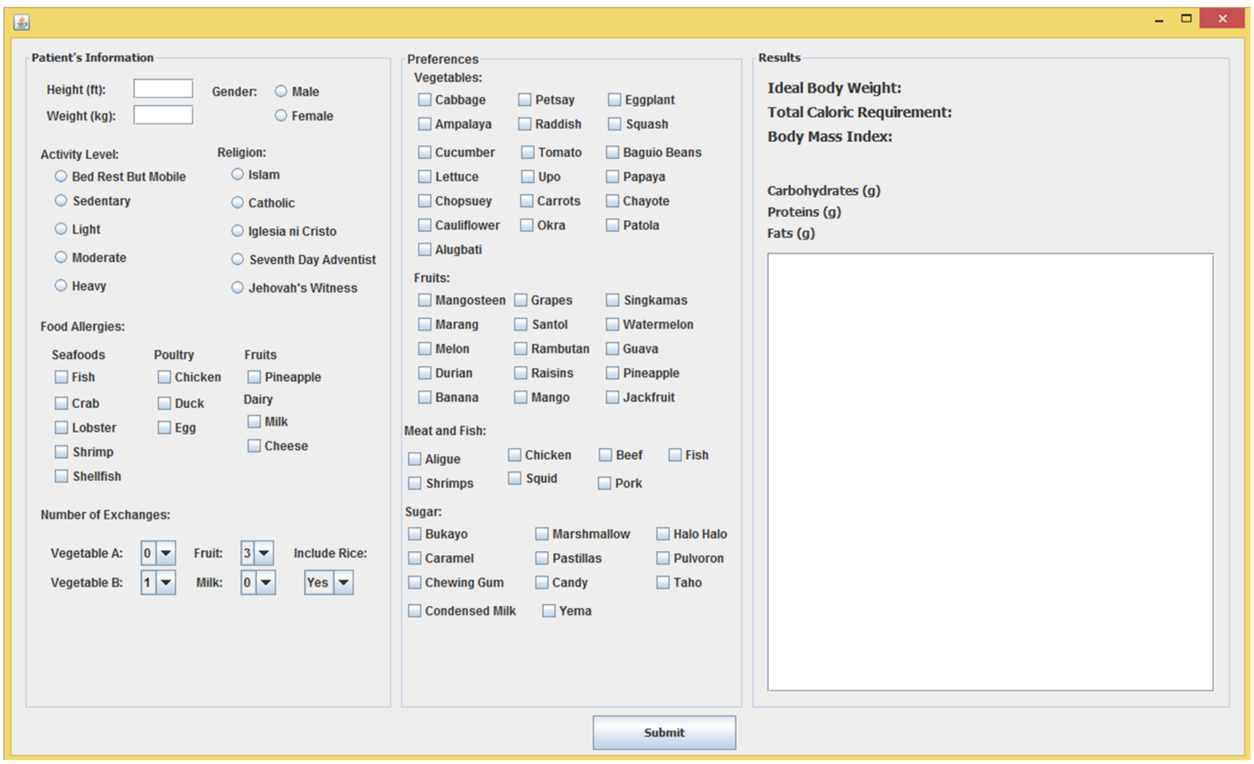


Figure 5.7.3:1. Java User Interface

The patient's information such as the height, weight, activity level and the preferred number of exchanges for the vegetable A and B, fruit and milk, and the inclusion of rice in the diet, and the hard constraints such as the religion and food allergies, are taken from the first panel. The remaining hard constraint is listed in the second panel, namely the food preference. When the input data is completed, the Submit button can be clicked.

The IBW, TCR and BMI of the patient are evaluated first before the query is sent to the ECLiPSe engine. These, along with the carbohydrates, proteins and fats in grams, and BMI classification, are displayed in the third panel. After the query is run, the Exclusion, Preference, Required, Exchange lists, Cost, and the generated list of food items are displayed in the text area.

5.8 Test and Evaluate

Below are the sample test cases with the results generated by the system.

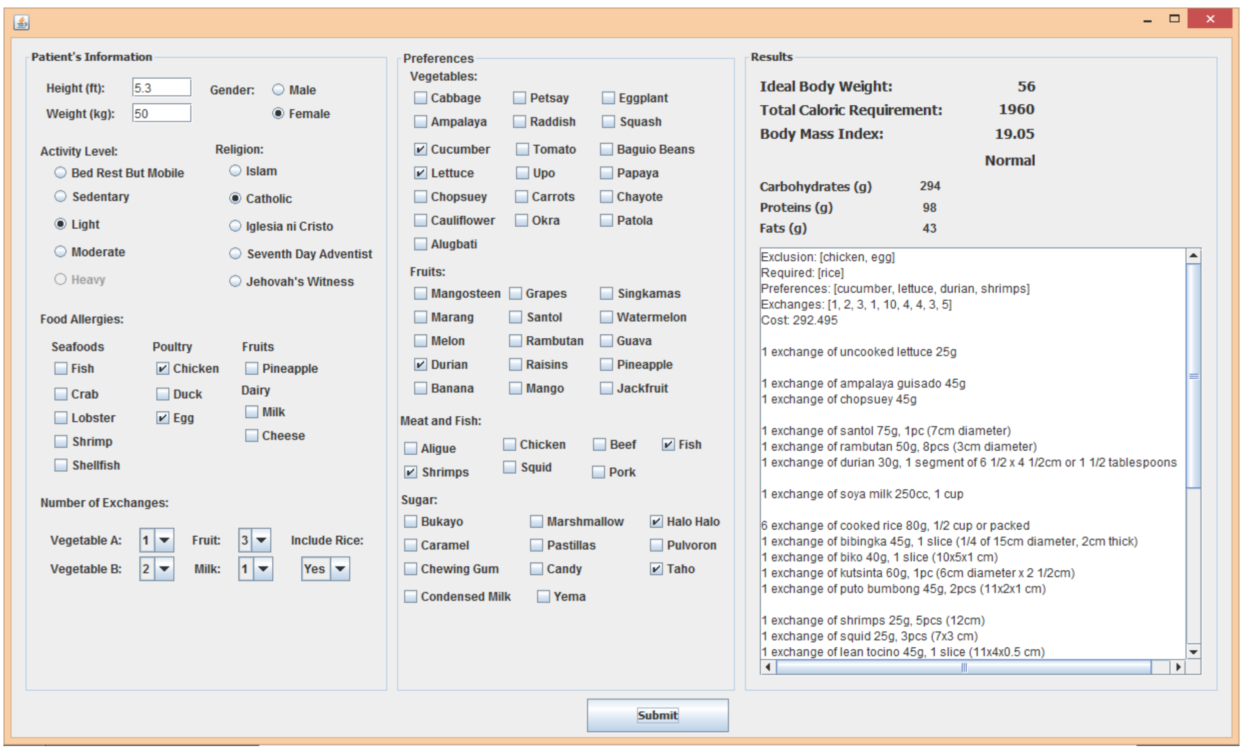


Figure 5.8:1. Test Case I

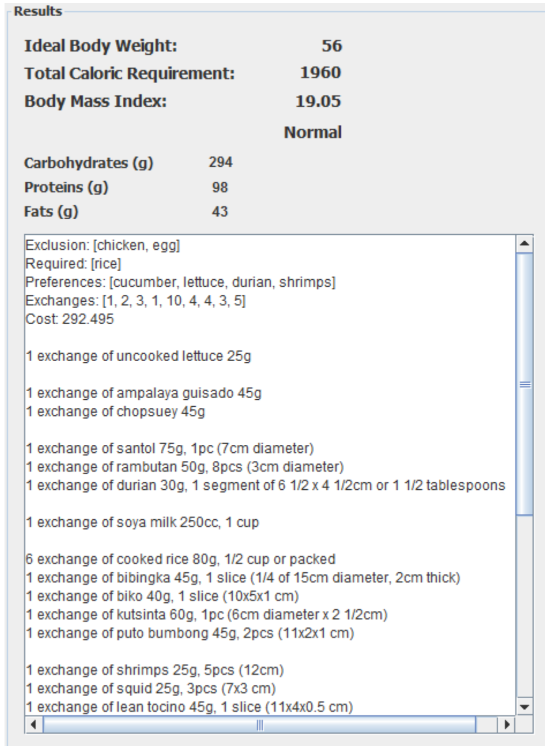


Figure 5.8:2. Test Case I Results Panel (1/2)

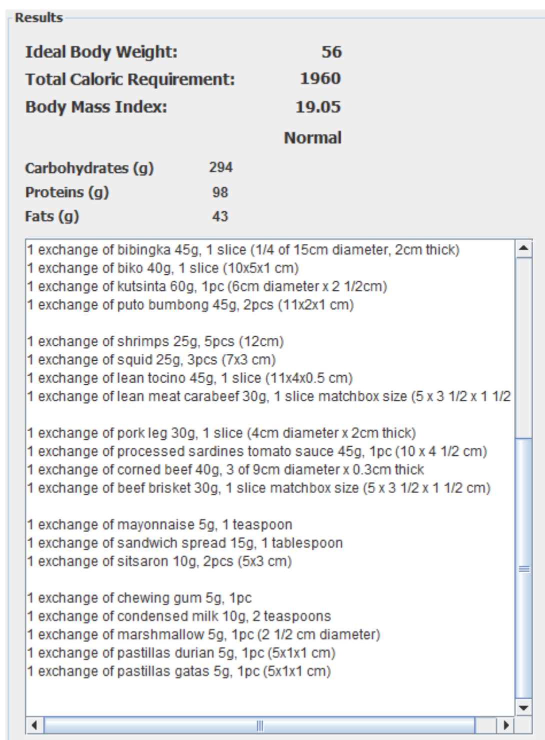


Figure 5.8:3. Test Case I Results Panel (2/2)

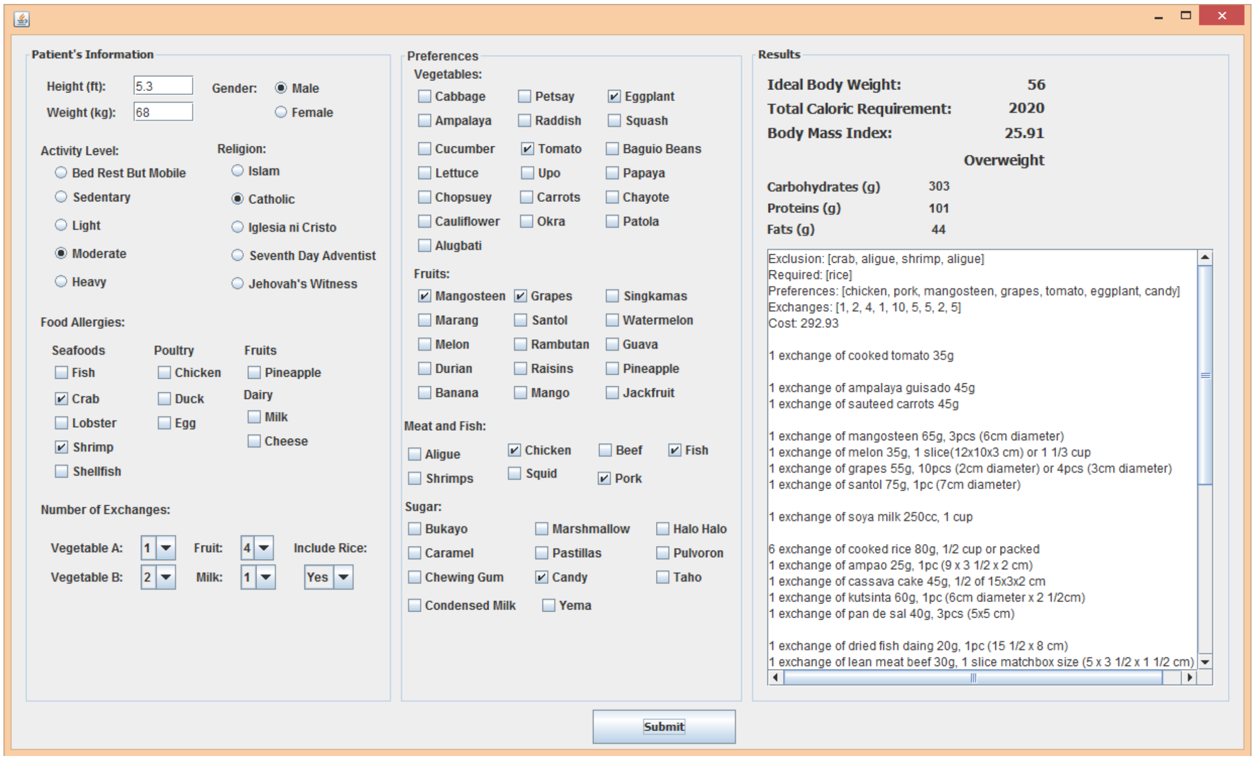


Figure 5.8:4. Test Case II

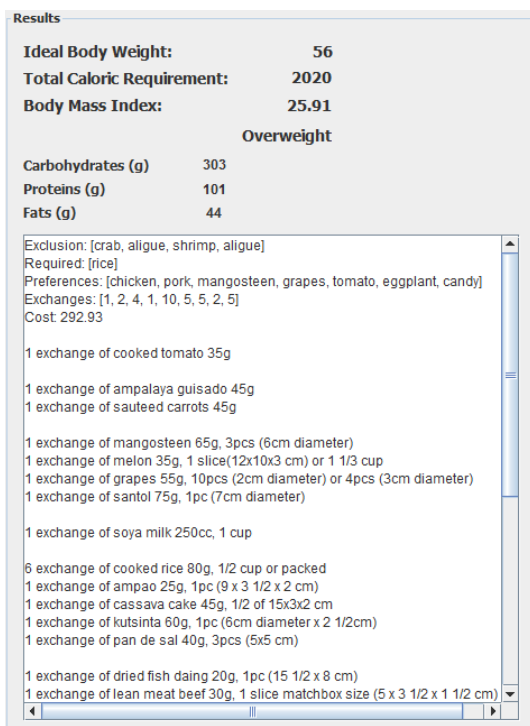


Figure 5.8:5. Test Case II Results Panel (1/2)

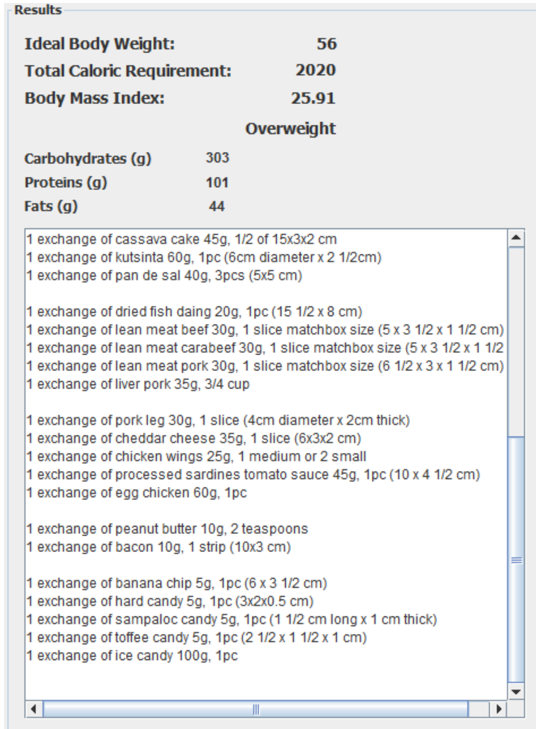


Figure 5.8:6. Test Case II Results Panel (2/2)

There are some problems that arise in testing the system. Possible conditions are the following:

1. If the user selects a lot of food items in the preference list that it exceeds the calculated TCR.
2. If the number of food items selected in the preference list excessively exceeds the number of exchanges of a particular food group.

Unfortunately, the system does not prompt the user any error messages when the problem arises. However, it does not display anything in the text area, and displays a *null* in the standard output as shown in the figures below.

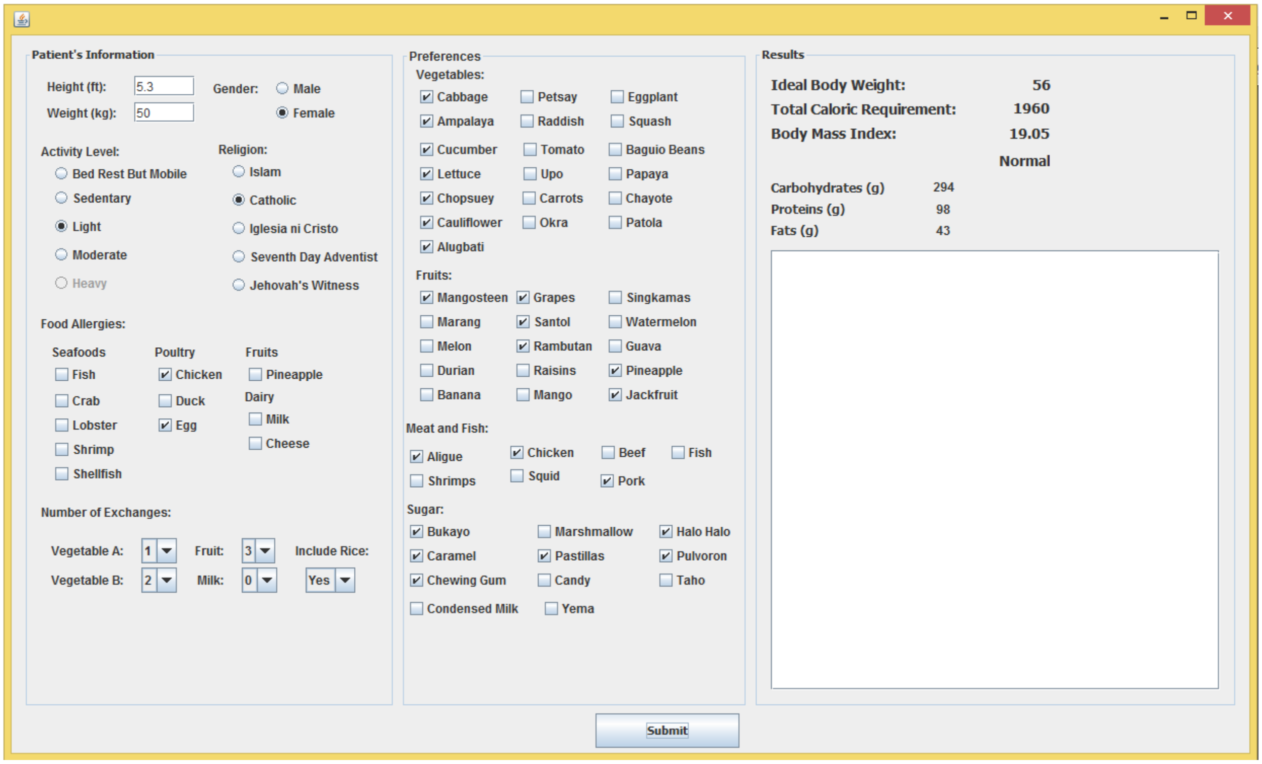


Figure 5.8:7. No Solution Output in Java

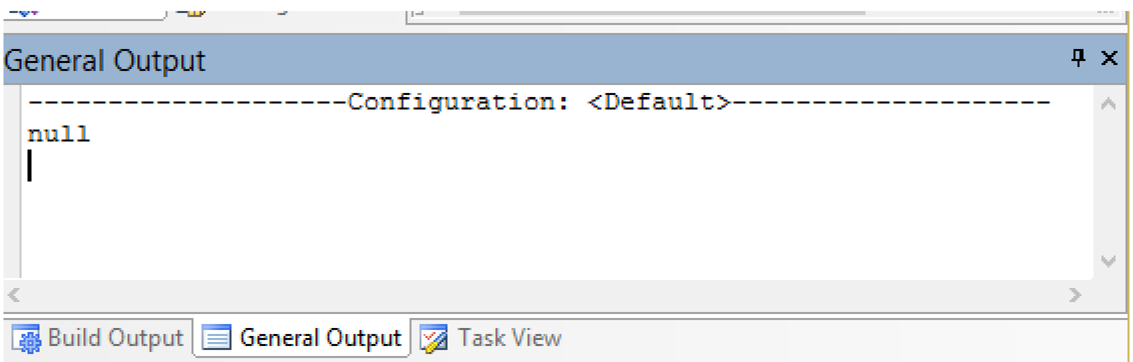


Figure 5.8:8. General Output in Java

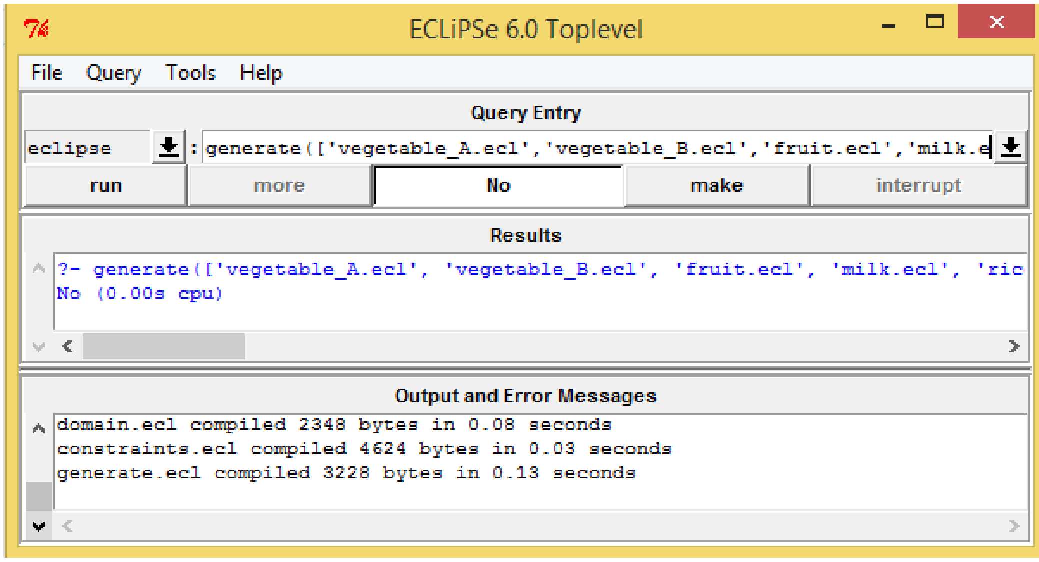


Figure 5.8:9. No Solution Output in ECLiPSe

The evaluation of the system was done by comparing the system's generated results and an expert's manual computation and diet prescription. Two test cases were given to the expert and the results are as follows:

Case 1

Dietary Rx 1900kcal-285g CHO-95g CHON-40gFat

Food Item	Ex	CHO	CHON	Fat	Kcal
Veg A	1	-	-	-	-
Veg B	2	6	2	-	32
Fruit	3	30	-	-	120
Milk	1	12	8	10	170
Rice	9	207	18	-	900
Meat:					
LF	4	-	32	4	164
MF	4	-	32	24	344
HF	-	-	-	-	-
Fat	1	-	-	5	45
Sugar	5	25	-	-	100
Total		280	92	43	1875
		(-5)	(-3)	(+3)	(-25)

JASON O. MOLINA, RND, MAEM
PRC Lic. No. 0012027
22 February 22, 2014

Test Case 1

Height: 5'3
Weight: 50kgs
Gender: Female
Activity Level: Student-Light
Religion: Roman Catholic
Food Allergies: Chicken, Egg

Number of Exchanges:
Veg. A: 1
Veg. B: 2
Fruit: 3
Milk: 1
Rice: Yes

Preferences:
5.3A. Vegetables
- Cucumber
- Lettuce

B. Fruits
- Durian

C. Meat and Fish
- Shrimps
- Fish

D. Sugar
-Halo-halo
-Taho

5x12=60+3=63 inches x 2.54= 160.02 cm-100= 60 kg – 10%= **DBW is 54kg**

%DBW= $\frac{\text{Actual weight}}{\text{Desirable body weight}}$ x 100
= 50/54 x 100%
= **92% Client has Normal Nutritional Status**

If BMI is used:
BMI= $\frac{\text{wt (kg)}}{\text{Ht (m}^2\text{)}}$
= 50/2.5064
= **19.94 client is considered Normal**

TCR= 54 x 35
= **1890 or 1900 kcal per day**

JASON O. MOLINA, RND, MAEM
PRC Lic. No. 0012027
22 February 22, 2014

Vegetable A: 1 exchange
1 cup raw Cucumber or lettuce or any substitute (**negligible kcal**)
Vegetable B: exchanges (**32 kcal**)
1 cup cooked or raw vegetables (e.g. Carrots or any substitute belonging to Group B)
- Vegetable B (Not specified in the preferences)
Fruit: (**120 kcal**)
1 slice mango (or any substitute)
1 pc banana (or any substitute)
1 segment durian (or any substitute)
Milk: (**170 kcal**)
1 cup of whole milk (fresh milk) or if powdered milk is used, use 4 tablespoon of powder added with water to make 1 cup
Rice: (900 kcal)
4 ½ cup rice (or substitute)
Low Fat Meat: (**164 kcal**)
4 matchbox-size fish (or substitute)
Medium Fat Meat: (**344 kcal**)
4 matchbox-size medium fat meat (or substitute)
Fat: (**45 kcal**)
1 tsp oil (used for frying/sautéing)
Sugar: (**100 kcal**)
5 tsp sugar (or its equivalent to Halo-halo or Taho)

Case 3

Dietary Rx 1600 kcal-240g CHO- 80g CHON- 35g Fat

Food Item	Ex	CHO	CHON	Fat	Kcal
Veg A	1	-	-	-	-
Veg B	2	6	2	-	32
Fruit	4	40	-	-	160
Milk	1	12	8	10	170
Rice	8	184	16	-	800
Meat:					
LF	4	-	32	4	164
MF	3	-	24	18	258
HF	-	-	-	-	-
Fat	1	-	-	5	45
Sugar	-	-	-	-	-
Total		242	82	37	1629
		(+2)	(+2)	(+2)	(+29)

JASON O. MOLINA, RND, MAEM
PRC Lic. No. 0012027
22 February 22, 2014

Note:

- **May exclude any exchange for fat if preferred. In case, it is still within the allowance.**
- **May include 1 exchange of sugar if there is no fat exchange,**

est Case 3 Height: 5'3 Weight: 68kgs Gender: Male Activity Level: Farmer-Moderate Religion: Roman Catholic Food Allergies: Crabs, Shrimps Number of Exchanges: Veg. A: 1 Veg. B: 2 Fruit: 4 Milk: 1 Rice: Yes Preferences: A. Vegetables - Tomato - Okra - Eggplant - Papaya B. Fruits - Mangosteen - Durian - Grapes - Mango C. Meat and Fish - Pork - Fish - Chicken D. Sugar - Candy	DBW= 54 kg(same as case No. 1) %DBW= Actual weight x 100 Desirable body weight = 68/54 x 100% **= 126% Client is Obese** If BMI is used: BMI= wt (kg) Ht (m^2) = 68/2.5064 **= 27.13 Client is Overweight** TCR= 54 x 35 **= 1900 kcal-300 kcal=1600 kcal (Reducing Diet)** JASON O. MOLINA, RND, MAEM PRC Lic. No. 0012027 22 February 22, 2014 Vegetable A: 1 exchange **(negligible kcal)** **½ cup cooked okra/eggplant/tomato or any substitute)** Vegetable B: exchanges **(32 kcal)** 1 cup cooked or raw vegetables (e.g. Carrots or any substitute belonging to Group B) - Vegetable B (Not specified in the preferences) Fruit: **(160 kcal)** 1 slice mango (or any substitute) 1 pc mangosteen (or any substitute) 1 segment durian (or any substitute) 7 pcs grapes (please check number of pieces of grapes per exchange; also the size whether small or big) Milk: **(170 kcal)** 1 cup of whole milk (fresh milk) or if powdered milk is used, use 4 tablespoon of powder added with water to make 1 cup Rice: **(800 kcal)** 4 cups rice (or substitute) Low Fat Meat: **(164 kcal)** 4 matchbox-size lean pork/fish or chicken Medium Fat Meat: **(258 kcal)** 3 matchbox-size medium fat meat (or substitute) Fat: **(45 kcal) or may not allow fat depending on preference (still within the requirement)** 1 tsp oil (used for frying/sautéing) Sugar: none (can be given 1 exchange or 1 pc of candy unless there is no fat exchange)

As the study aims to generate a diet that satisfies the abovementioned constraints, regardless of the exact number of exchanges the system generated and the dietician computed, the food restrictions and preferences were mainly observed in the comparison of the generated output and the manual diet prescription of the expert. The food allergies identified by a specific patient were avoided and the food preferences were included in the diet.

6. CONCLUSIONS AND RECOMMENDATION

6.1 Conclusions

This study generated an optimal diet for Type II Diabetes Mellitus patients using Constraint Logic Programming. The constraints identified were the patient's food preferences, allergies and restrictions based on religion. The cooked rice requirement in a daily meal plan that is identified by the user is also part of the constraints. Repetition of a specific food was also restricted. CLP limits the foods in the diet by satisfying the required carbohydrates, proteins and fat of the patient through the number of exchanges provided and computed by the patient and the system, respectively. This type of approach is of great help in solving optimization problems since it is capable of handling constraints.

6.2 Recommendation

The proponents recommend the following:

- We recommend that instead of just displaying the list of food items, it should be equally distributed into meals—Breakfast, Lunch, Dinner, AM Snacks, PM Snacks and Bedtime Snacks.
- Instead of providing only the raw foods, it should already be displayed as dishes.

REFERENCES

Balderamos, S., Que, M. (n.d.) *Food exchange list in the illustrated food choices for diabetes*

Briend, A., Ferguson, E., & Erhardt, J. (2003, January). Retrieved February 9, 2013, from pubmed.gov: http://www.ncbi.nlm.nih.gov/pubmed/12499991

Chien, C.-F., & Liu, C.-F. (2004). Retrieved February 9, 2013, from http://www.orstw.org.tw/Conference2004/B/102.pdf Diabetes Meal Plans and a Healthy Diet. (n.d.). Retrieved February 9, 2013, from American Diabetes Association: http://www.diabetes.org/food-and-fitness/food/planning-meals/diabetes-meal-plans-and-a-healthy-diet.html

Diabetes meal plans and a healthy diet. (n.d.). Retrieved February 9, 2013, from American Diabetes Association: http://www.diabetes.org/food-and-fitness/food/planning-meals/diabetes-meal-plans-and-a-healthy-diet.html

Food exchange lists for meal planning. (1994). Food and Nutrition Research Institute

Fruhwrith, T., Herold, A., Kuchenhoff, V., Le Provost, T., Lim, P., Monfroy, E., & Wallace, M. (1993, February). citeseerx.ist.psu.edu. Retrieved January 23, 2013, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.5117

Fruwrith, T. (1993, February). Retrieved January 23, 2013, from citeseerx.ist.psu.edu. Retrieved January 23, 2013, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.5117

Gavanelli, M., & Rossi, F. (n.d.). Retrieved February 9, 2013, from http://www.math.unipd.it/~frossi/gulp.pdf Hu, F., Dam, R., & Liu, S. (2001, July). Retrieved February 9, 2013, from Springer Link: http://link.springer.com/article/10.1007%2Fs001250100547?LI=true#page-2

Huang, Y., Lu, C., Yang, T., Fu, L., & Wang, C. (2010, June 22-24). Retrieved February 9, 2013, from Springer Link: http://link.springer.com/chapter/10.1007%2F978-3-642-13778-5_10s

Hunter, F. (2011, April). Retrieved February 9, 2013, from BBC: http://www.bbc.co.uk/health/treatments/healthy_living/nutrition/dietary_diabetes.shtml

Kerdprasop, N., & Kerdprasop, K. (2011). Retrieved February 9, 2013, from http://www.naun.org/multimedia/NAUN/m3as/17-218.pdf

Khan, A. S., & Hoffman, A. (2003). Retrieved February 9, 2013, from http://www.cse.unsw.edu.au/~cs9416/06s1/lectures/AIMedicine03.pdf

Novello, S., Schimpf, J., Shen, K., Singer, J. (2013). *ECLiPSe embedding and interfacing manual.*

Noyes, J., & Weisstein, E. (2013, February 8). Linear Programming. Retrieved February 9, 2013, from MathWorld - Wolfram Web Resources: http://mathworld.wolfram.com/LinearProgramming.html

Pricki, A., Magdic, D., Perl, M., & Zhilavski, I. (n.d.). Retrieved February 9, 2013, from http://bib.irb.hr/datoteka/370457.CEFood_2008_Diet_optimization_for_overweight_cardiovascular_patients.pdf

Rossi, F. (n.d.). Constraint Logic Programming. Retrieved February 8, 2013, from www.math.unipd.it/~frossi/survey-cyprus.ps.gz

Salzwedel, I. (2012, November 12). Retrieved February 9, 2013, from Health24: http://www.health24.com/dietnfood/Diabetic_diet/15-4650,46991.asp

Simonis, F. (2008, February 27). Retrieved February 7, 2013, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.177.256

Received March 2014; revised March 2014; accepted March 2014