

A feasibility study of combining expert system technology and linear programming techniques in dietetics

A van der Merwe
10100059

Dissertation submitted in partial fulfillment of the requirements for the degree *Magister Scientiae* in *Computer Science* at the Potchefstroom Campus of the North-West University

Supervisor: Prof HA Krüger
Co-supervisor: Prof T Steyn

September 2014

Abstract

Linear programming is widely used to solve various complex problems with many variables, subject to multiple constraints. Expert systems are created to provide expertise on complex problems through the application of inference procedures and advanced expert knowledge on facts relevant to the problem. The diet problem is well-known for its contribution to the development of linear programming. Over the years many variations and facets of the diet problem have been solved by means of linear programming techniques and expert systems respectively. In this study the feasibility of combining expert system technology and linear programming techniques to solve a diet problem topical to South Africa, is examined. A computer application is created that incorporates goal programming- and multi-objective linear programming models as the inference engine of an expert system. The program is successfully applied to test cases obtained through knowledge acquisition. The system delivers an eating-plan for an individual that conforms to the nutritional requirements of a healthy diet, includes the personal food preferences of that individual, and includes the food items that result in the lowest total cost. It further allows prioritization of the food preference and least cost factors through the use of weights. Based on the results, recommendations and contributions to the linear programming and expert system fields are presented.

Keywords: Expert system, goal programming, multi-objective linear programming, rule-based system, knowledge engineering, diet problem.

Uittreksel

Lineêre programmering word algemeen gebruik om verskeie komplekse probleme op te los, met baie veranderlikes, en onderhewig aan verskeie beperkings. Deskundige stelsels word geskep om kundigheid oor komplekse probleme te voorsien, deur die toepassing van inferensie prosedures en gevorderde deskundige kennis op feite relevant tot die probleem. Die dieetprobleem is bekend vir sy bydrae tot die ontwikkeling van lineêre programmering. Oor die jare heen is baie variasies en fasette van die dieetprobleem opgelos deur die toepassing van lineêre programmeringstegnieke en deskundige stelsels onderskeidelik. In hierdie studie word die lewensvatbaarheid van die kombinerings van deskundige stelsel tegnologie en lineêre programmeringstegnieke om 'n dieetprobleem aktueel tot Suid-Afrika op te los, ondersoek. 'n Rekenaar toepassing word geskep waarin doelwitprogrammering en multidoelwit lineêre programmering as die inferensie enjin van 'n deskundige stelsel geïnkorporeer word. Die program word suksesvol toegepas op toetsgevalle verkry deur kennis insameling. Die stelsel lewer 'n eetplan vir 'n individu wat voldoen aan die voedingsbehoefte van 'n gesonde dieet, die persoonlike kos voorkeure van die individu, en ook die voedsel-items insluit wat lei tot die laagste totale koste. Verder maak dit voorsiening vir prioritisering van kos voorkeure en minste koste faktore deur die gebruik van gewigte. Aanbevelings en bydraes tot die lineêre programmering- en deskundige stelsel velde word aangebied, gebaseer op die resultate.

Sleutelwoorde: Deskundige stelsel, doelwitprogrammering, multidoelwit lineêre programmering, reël-gebaseerde stelsel, kennisingenieurswese, dieetprobleem.

Acknowledgements

I want to start by thanking my Lord and Saviour without whom completion of this study would not have been possible.

I would also like to thank my supervisor, Prof Hennie Krüger and co-supervisor, Prof Tjaart Steyn for their support and patience with an 'older' student. Your mentorship means a lot to me.

To my husband and girls: "This one's for you!" Thank you for understanding, helping and standing in for an ever-absent mother these past few months. I'm very proud of all of you and hope "my book" will make the suffering seem sweeter.

Thank you mom, for having a look at my "delicious English". I hope I have not strayed too far from all your teachings!

Finally, I would like to thank the rest of my family, friends and colleagues for their support and advice through all of the trials and tribulations which just never seemed to stop throughout this study.

Conference contributions

Excerpts from the current study have been presented as follows:

Decision modelling by means of expert systems in a diet application

A. van der Merwe*, H. A. Krüger, T. Steyn

(Presented at the Telkom CoE Seminar, Potchefstroom, South Africa, 22 May 2012)

Applying expert system technology in dietetics

A. van der Merwe*, H. A. Krüger, T. Steyn

(Presented at the 41st ORSSA Annual Conference, Muldersdrift, South Africa, 16-19
September 2012)

Combining expert system technology and mathematical programming techniques in dietetics

A. van der Merwe*, H. A. Krüger, T. Steyn

(Presented at the 26th EURO Conference, Rome, Italy, 1-4 July 2013)

Solving the diet problem using expert system technology and mathematical programming techniques

A. van der Merwe*, H. A. Krüger, T. Steyn

(Presented at the 42nd ORSSA Annual Conference, Stellenbosch, South Africa, 15-
18 September 2013)

Table of Contents

List of Figures	ix
List of Tables	xi
Chapter 1 – Introduction and problem statement.....	1
1.1. Introduction	1
1.2. Problem statement	2
1.3. Research goals	3
1.4. Research methodology	4
1.4.1. Interpretivism	4
1.4.2. Critical social theory	4
1.4.3. Positivism	5
1.5. Chapter outline.....	6
1.6. Chapter summary.....	7
Chapter 2 - Mathematical programming techniques	8
2.1. Introduction	8
2.2. Linear programming	8
2.2.1. Formulating a linear programming model.....	9
2.2.2. Solving a linear programming model	10
2.2.2.1. Graphical solution method	11
2.2.2.2. The simplex method.....	13
2.2.3. Special cases	17
2.2.4. Sensitivity analysis	18
2.3. Integer linear programming	19
2.3.1. Logical conditions.....	20
2.3.2. Solving an ILP by means of the graphical representation method.....	21
2.3.3. The branch-and-bound method for ILP	23
2.4. Goal programming	25
2.4.1. The weights method.....	27
2.4.2. The preemptive method	27
2.5. Multi-objective programming	28
2.6. Chapter summary.....	34

Chapter 3 – Expert systems principles and development	36
3.1. Introduction	36
3.2. Expert systems explained	36
3.3. Knowledge representation techniques	38
3.4. The components of an expert system	40
3.4.1. The inference engine.....	40
3.4.2. The working memory.....	42
3.4.3. The knowledge base	42
3.4.4. The user interface	43
3.5. Knowledge engineering.....	43
3.5.1. Preliminary planning.....	47
3.5.1.1. Selecting the appropriate paradigm	47
3.5.1.2. Payoff.....	47
3.5.1.3. Tools	47
3.5.1.4. Cost.....	48
3.5.2. Expert system development life cycle	49
3.5.2.1. Planning.....	51
3.5.2.2. Knowledge definition.....	52
3.5.2.3. Knowledge design.....	53
3.5.2.4. Code and checkout	53
3.5.2.5. Knowledge verification	53
3.5.2.6. System evaluation.....	54
3.6. Chapter summary.....	54
 Chapter 4 - The diet problem and South Africa	 55
4.1. Introduction	55
4.2. History of the diet problem	55
4.3. Recent developments in linear programming diet applications	56
4.4. Computerized solutions to the diet problem	59
4.4.1. The implementation of linear programming techniques to solve diet related problems	59
4.4.2. Expert systems employed to solve diet and health-related problems.....	62
4.5. Health in South Africa	67
4.5.1. Obesity	69
4.5.2. Dieting and abnormal weight control behaviour.....	71

4.6. Chapter summary.....	72
Chapter 5 – Knowledge engineering: Planning and definition.....	73
5.1. Introduction	73
5.2. Development paradigm	73
5.3. Planning	74
5.3.1. Task phasing	74
5.3.2. Feasibility assessment	74
5.3.3. Preliminary functional layout.....	75
5.3.4. High-level requirements.....	76
5.4. Knowledge definition	76
5.4.1. Knowledge source identification and selection	76
5.4.2. Knowledge acquisition.....	78
5.4.2.1. Interviews conducted	78
5.4.2.2. Observations	80
5.4.2.3. Researching documents	80
5.4.3. Knowledge analysis.....	81
5.4.3.1. How to compile a diet.....	81
5.4.3.2. Observations feedback	86
5.4.3.3. Documents analysis	86
5.4.4. Knowledge extraction	88
5.4.4.1. Functional layout.....	88
5.4.4.2. The phases of the expert system	89
5.5. Chapter summary.....	89
Chapter 6 – Knowledge engineering: System development and evaluation	91
6.1. Introduction	91
6.2. Knowledge design.....	91
6.2.1. Knowledge classification	92
6.2.2. Detailed design.....	97
6.2.2.1. Goal programming model for number of exchanges from each list ...	98
6.2.2.2. Multi-objective linear programming model for least cost and highest preference diet.....	101
6.3. Code and checkout	103
6.4. Testing and evaluation of the system.....	109

6.4.1. System application	110
6.4.2. Application of the expert system to the consultation sessions with the dietician.....	115
6.5. Chapter summary.....	118
Chapter 7 – Summary and results	120
7.1. Introduction	120
7.2. Development of the expert system.....	120
7.3. System performance	122
7.4. Contribution to linear programming- and expert system research.....	122
7.5. Concluding remarks	125
7.6. Chapter summary.....	125
Chapter 8 – Conclusions	127
8.1. Introduction	127
8.2. Objectives of the study.....	127
8.3. Problems experienced	130
8.4. Possibilities for further research.....	131
8.5. Chapter summary.....	131
Appendix A - Letter to the parents.....	132
Appendix B - Excerpt of the transcriptions of the consultation sessions summary	133
Appendix C - Diet for subject 6 as received from the dietician	134
Appendix D - Food list and price database.....	142
Appendix E - VBA (Visual Basic for Applications) code.....	146
Appendix F - Eating-plans generated by the expert system	147
References.....	148

List of Figures

Chapter 2 - Mathematical programming techniques

Figure 2-1: Graphical representation of the objective function and constraints in the example model.....	11
Figure 2-2: Isoprofit line solution of the example linear program.....	12
Figure 2-3: The corner point solution method.....	13
Figure 2-4: Determining the pivot element	15
Figure 2-5: Graphical solution for the ILP example model.....	22
Figure 2-6: The branch-and-bound method applied to the example problem.....	24
Figure 2-7: Trade-off between two conflicting objectives.....	29

Chapter 3 – Expert systems principles and development

Figure 3-1: Problem domain vs. knowledge domain	37
Figure 3-2: Example semantic net.....	39
Figure 3-3: The mechanism of an expert system	43
Figure 3-4: The waterfall model.....	49
Figure 3-5: The spiral model for developing expert systems (Giarratano and Riley, 2005)	50
Figure 3-6: Stages in the linear model in expert system development (Giarratano and Riley, 2005)	51

Chapter 6 – Knowledge engineering: System development and evaluation

Figure 6-1: Control structure of the expert system	93
Figure 6-2: Decision tree representing the iterative reasoning process of the dietician	99
Figure 6-3: A typical rule that compares the cost of two items with each other	103
Figure 6-4: The start-up menu of the system	104
Figure 6-5: Automatic daily energy requirement calculation object	104
Figure 6-6: Output of the goal programming model that calculates the number of exchanges from each group	106
Figure 6-7: Results of the goal programming model	107
Figure 6-8: Setting the user preferences for fruit	108
Figure 6-9: User form to set the user's cost vs. food preference priority	108

Figure 6-10: Example output of a one-day eating-plan generated by the expert system..... 109

Figure 6-11: The effect of the changes in preference and cost weights on the prices of the resulting eating-plans for a 6164 kJ daily energy requirement 113

Figure 6-12: The effect of the changes in preference and cost weights on the preference satisfaction of the resulting eating-plans for a 6164 kJ daily energy requirement..... 114

List of Tables

Chapter 2 - Mathematical programming techniques

Table 2-1: The first simplex tableau	14
Table 2-2: The second simplex tableau.....	16
Table 2-3: The third simplex tableau	17
Table 2-4: Expected demand increase.....	31
Table 2-5: The amount and types of coal produced at the mines.....	31
Table 2-6: Disadvantages to producing coal	31
Table 3-1: A student frame.....	39
Table 3-2: A class frame	39
Table 5-1: The phases to be performed within each task.....	74
Table 5-2: Boundaries for the diet problem to be solved by the expert system	75
Table 5-3: Ratings of sources of knowledge for the expert system	77
Table 5-4: How to determine the activity factor from the activity level	82
Table 5-5: How to calculate the additional daily energy requirements of a growing girl	82
Table 5-6: Recommended macronutrient intakes for girls between the ages of 14 and 18 years	84
Table 5-7: The exchange lists from which foods are chosen to add to a diet.....	85
Table 6-1: Food list for the <i>Fruit</i> exchange group.....	95
Table 6-2: Price list as compiled for the <i>Fruit</i> list.....	97
Table 6-3: Data structures for the elements in the expert system	98
Table 6-4: Dietary guidelines for food group selections	100
Table 6-5: Similar food items to be constrained so that only one is chosen	102
Table 6-6: Setting the food preferences of subject 6.....	111
Table 6-7: Information needed for required daily energy calculation.....	111
Table 6-8: Statistics for the 6164 kJ eating-plan generated by the expert system .	112
Table 6-9: Statistics for the 5700 kJ eating-plan generated by the expert system .	114
Table 6-10: Comparison of the food items in the eating-plans	117
Table 6-11: The number of food matches, group matches and exceptions between the eating-plans.....	118
Table B-1: Summary of the transcriptions of the consultation sessions	133

Chapter 1 – Introduction and problem statement

1.1. Introduction

The occurrence of obesity and eating disorders among teenagers in South Africa is increasing (Kimani-Murage, 2013; Reddy, Resnicow, James, Funani, Kambarran, Omaidien, Masuka, Sewpaul, Vaughan and Mbewu, 2012; Monyeki, Neetens, Moss and Twisk, 2012; Puoane, Bradley and Hughes, 2002) and although many reasons for the change can be argued, finding a solution to the problem is equally as important.

Many computer systems have been developed to solve different varieties of the well-known diet problem by means of linear programming techniques. The diet problem in general reads as follows: Determine a diet that

- is optimised in terms of an objective (usually least cost);
- provides all the nutrients necessary to sustain a healthy body.

However most systems are usually designed for specific target groups and take into consideration certain factors that are unique to the chosen environment.

Research has shown that due to changes on cultural, social and political levels, certain health issues like obesity and other weight problems are on the increase in South Africa (Stupar, Eide, Bourne, Hendricks, Iverson and Wandel, 2012). In the study conducted among schoolgirls between the ages of 14 and 16 years as well as school staff-members, it was concluded that changes need to be made in the nutrition policies for developing countries like South Africa to prevent health-conditions caused by obesity.

Among five research-based proposals presented for preventing obesity and other related disorders among adolescents, Neumark-Sztainer (2009) recommended that girls should be encouraged to:

- Eat healthily rather than follow diets;
- Adopt a positive body image;

- Have meals with their families;
- Take part in some form of physical activity; and
- Involve families of overweight teenagers in addressing these problems.

The last and most important recommendation was that weight-neglect should be addressed with overweight teenagers as well as with their families. Providing information or educating teenagers on healthy eating and living could help to take the emphasis away from weight issues and rather initiate behavioural change.

The ideal solution would be to provide teenage girls with full access to a qualified dietician who could teach them how and what to eat to sustain a healthy body. However due to financial constraints most teenage girls in South Africa will never be able to afford the services of such an expert. An expert system is proposed which incorporates the necessary linear programming techniques to solve the diet problem and facilitate sharing of the required knowledge to provide teenagers with the means to create their own individual eating-plans.

The purpose of this chapter is to guide the reader through the research study by presenting the problem statement, the goals of the study and the methodology employed. A layout of the study which explains the purpose of each chapter is also presented.

1.2. Problem statement

An expert system can be described as a computer program which utilizes expert knowledge and inference procedures to solve problems that require significant human expertise (Giarratano and Riley, 2005). Dieticians possess advanced expertise in the field of dietetics which they use to evaluate people and provide them with among other things, eating-plans. Such an eating-plan is generated subject to numerous considerations and restraints.

The classic diet problem gives a method for determining an eating-plan for an individual, subject to certain constraints (Darby-Dowman and Wilson, 2002; Dantzig, 1990). The constraints may, among others, include personal food preferences, minimum nutritional requirements and possible allergies.

In order to bridge the gap that exists between an average South African teenage girl and the advanced knowledge a dietician can provide to alter her eating habits, a

combination between an expert system and the linear programming models used to solve the diet problem, is proposed.

Therefore, it is necessary to determine the feasibility of combining an expert system which interacts with a user to gather input data, with linear programming techniques which could use the input data to produce a unique solution for every individual that uses the system.

1.3. Research goals

The objective of this study is to determine the feasibility of combining the heuristic inference capabilities of an expert system with the optimizing advantages that linear programming techniques can provide. It is proposed that an expert system is created that performs the process that a dietician (the expert in this instance) follows to generate a healthy eating-plan for a female teenager in South Africa. The expert system will be responsible for generating the constraints relevant to the problem, based on input-information obtained from a user. The constraints will then be applied to the linear programming model to solve the diet problem and provide the user with an individualised eating-plan.

Secondary objectives to be met include:

- Providing an overview of the mathematical programming techniques referred to and applied in this study;
- Providing an overview of expert system concepts, principles, and development;
- Performing an exploratory investigation into the use of linear programming techniques and expert systems in solving health-related problems and the presence of such problems in South Africa;
- Performing the initial stages of knowledge engineering to create a theoretical design of an expert system for solving the relevant diet problem;
- Performing the development and evaluation stages of knowledge engineering; and
- Interpreting the results, stating the possible contributions resulting from the system and making appropriate recommendations.

1.4. Research methodology

The purpose of this section is to differentiate between the different research paradigms by briefly explaining the properties of each, to identify the paradigm according to which this study was conducted.

Research is conducted according to a distinct thought pattern or concept which is referred to as a research paradigm. In the scientific community, research is done according to the *positivistic paradigm*, *interpretivistic paradigm* or *critical social theory*.

1.4.1. Interpretivism

As the name implies, the purpose of doing research in the interpretivistic paradigm is to find and examine different interpretations or hidden meanings on subjects or conditions while still remaining objective (De Villiers, 2005).

Interpretivistic research is characterised by the existence of multiple realities or rather different impressions on a matter (Oates, 2006). People who share a common cultural background usually share similar beliefs which are transmitted dynamically by means of language, meanings and similar backgrounds. Researchers using the interpretivistic paradigm to conduct research usually have their own interpretations and beliefs on certain subjects but should remain objective for the duration of their study by self-reflecting on the reason for their research as well as the methods used. Personal opinions or beliefs on the research subject should not influence the work because such a study is aimed at researching people in their natural environment. The perspectives of the participants in the research must be accepted without prejudice.

Research done in the interpretivistic paradigm is done mainly to formulate a theory rather than to prove a hypothesis (Oates, 2006).

1.4.2. Critical social theory

Critical social theory is strongly entrenched in history and is based on the work of Karl Marx (Taylor, 2007). The focus of critical social theory is to identify and eliminate areas of concern like conflict, power dominations, and contradictions within an environment (Oates, 2006). Critical research therefore tries to transform social realities by eradicating sources of alienation and domination (Orlikowski and

Baroudi, 1991) and resembles interpretivistic research in the sense that the reasons for some phenomena are explained, but critical research also tries to improve some social situation. Critical research tries to understand different perspectives on a subject and also change a situation for the better.

One major property of critical research is that the researcher often believes that progress in the development of technology and only some people having access to that new technology can provide them with, an unfair advantage (Bilton, 2011). Due to the emancipatory nature of critical research, the researcher rejects the belief that people need to adapt to technological advances because it can create unfair power strongholds.

1.4.3. Positivism

The positivistic paradigm, also called the scientific method, is used when behaviour is studied in a way similar to natural scientists studying behaviour in the natural world. The positivistic researcher makes two main assumptions in his research: the world is not random but ordered, and the world can be studied objectively (Oates, 2006).

Positivism is generally associated with empirical and logical knowledge or, knowledge as represented in natural sciences and logic or mathematics (Hughes and Sharrock, 1997). Researchers are said to emulate mainly three sets of logic namely rules of formal logic, rules of experimental- and quasi-experimental design, and rules of hypothetical deductive logic.

Positivistic research is characterised by formulating, empirical testing and verification of a hypothesis and is often based on mathematical modelling or simulation of a real-world situation (Hughes and Sharrock, 1997). A mathematical model of a problem or situation aims to generalise the problem by formulating it mathematically in order to solve it (Romm, 1991).

The data collection methods generally used in positivistic research includes surveys, experiments, numerical methods and econometrics, and/or the design and creation of some artefact.

Surveys are used to study large groups of people and can appear in the form of questionnaires, interviews, observations and documentation (Oates, 2006).

Experiments are conducted to determine cause-and-effect relationships wherein the experimenter changes nothing other than one factor in an environment to determine the difference or change in the overall result (Romm, 2010). *Numerical methods and econometrics* involves the use of numerical models to simulate scientific occurrences in economics (Tobin, 2006). *Design and create* describes the process during which the researcher studies or observes a series of events, formulates a hypothesis and develops and tests the hypothesis by creating an artefact for that purpose (Oates, 2006; Romm, 1991).

In Information Science, an information system known as an *artefact* is designed and created and used to define the problem, solve it, and test the hypothesis. The types of artefacts that can be created include *constructs* which represent the vocabulary used in data flow diagrams in Computer Science, *models* that combine constructs to develop systems that simulate real world situations, *methods* which are the processes followed to develop the software artefact, and an *instantiation* which is a fully functioning software program wherein the model and methods are implemented to successfully prove or disprove the hypothesis (Oates, 2006).

In this study, an artefact was designed and created according to the positivistic paradigm in the form of an expert system.

1.5. Chapter outline

This section aims to guide the reader through the chapters that follow by briefly stating the purpose of each chapter.

Chapter 2 presents basic mathematical programming concepts and definitions and explains the formulation of linear programming models. Solution methods referred to in this study are also discussed, supported by some basic examples.

The purpose of chapter 3 is to present the reader with a general understanding of expert systems as subfield of Artificial Intelligence and to provide some background into the application fields of expert systems. A discussion on expert system development life cycles is followed by a layout of the knowledge engineering process used to create the system in this study.

Chapter 4 provides background on the application field of this study, which is dietetics. The chapter presents the history of the diet problem and discusses some

linear programming and expert system solutions related to the problem by highlighting previous research in the fields. Health concerns relevant to South Africa are also revealed to reinforce the rationale behind the study.

Chapter 5 discusses the first two stages of the knowledge engineering process which involves the initial planning and knowledge accumulation stages. The chapter presents the theoretical design of the expert system that was created.

Chapter 6 presents a continuation of the knowledge engineering process by discussing the practical design stages involved in expert system development. The results from testing and evaluating the system are also presented.

Chapter 7 starts with a summarizing review of the study, presents possible contributions to linear programming- and expert system research and the chapter finishes with related concluding remarks.

The last chapter demonstrates how the research goals of this study were achieved and concludes by presenting some recommendations and considerations for future work.

1.6. Chapter summary

Chapter 1 served as a basic introduction to the research study by explaining the research problem and presenting the objectives. A brief overview of research methodologies was presented to highlight the paradigm according to which this study was conducted. The structure of the study and a brief layout of each of the chapters were also presented.

Chapter 2 - Mathematical programming techniques

2.1. Introduction

A number of mathematical programming methods were used to solve different phases of the problem in this study which necessitates a chapter that outlines some of the basic principles and models referred to in this report.

Although the diet problem specific to this study and the subsequent solution is discussed in chapter 5, this chapter provides insight into mathematical programming techniques and -methods used in this study.

The chapter starts by explaining the concept of a linear programming model and some of the definitions relevant in the formulation process. The graphical solution method is used to explain the meaning of the feasible region and the simplex method is discussed as a way to solve a linear programming model by means of an example. Special cases that can arise and cause problems when solving a model are also briefly mentioned, followed by a brief section on sensitivity analysis and terms.

Integer linear programming, goal programming and multi-objective programming which form an integral part of this study are briefly explained and some examples are also shown.

2.2. Linear programming

A lot of management decisions involve the determination of the best way to achieve certain objectives of the company, subject to certain restrictions of the company's operating environment. Problems like these can typically be solved using *linear programming* which, according to Taylor (2013) can be defined as:

"...a model that consists of linear relationships representing a firm's decision(s), given an objective and resource constraints."

Linear programming gets its name from the fact that the functional relationships in the mathematical model are *linear*, and the model is solved by a series of predetermined steps, or a *program*.

Taylor (2013) provides the following three steps for applying linear programming to a specific problem:

1. Identify the problem as being solvable by linear programming.
2. Formulate the unstructured problem in a mathematical model.
3. Solve the model using recognized mathematical techniques.

Identifying whether a problem can be solved using linear programming can be done by applying Taylor's definition and establishing whether linear relationships exist between factors that determine the goal and if reaching that goal is constrained by environmental conditions.

Formulating and solving the model will be discussed in more detail in the next sections.

2.2.1. Formulating a linear programming model

A linear program consists of some basic components. The *objective function* is a function that describes a specific goal for example maximum revenue, in terms of linear relationships between the factors that determine the total value of that goal, e.g. the price of certain items. The objective function is either minimised or maximised, depending on the needs and goals of an individual or company.

Decision variables are mathematical symbols that represent the factors that contribute to the goal in the objective function, for instance the items that are to be sold.

Model constraints are linear relationships of the decision variables that represent restrictions placed on the individual or company by the environment.

Coefficients are numeric values in the objective function and the constraints, and define the magnitude of contribution of the various variables in the model.

Furthermore, the following basic assumptions are made regarding the model (Render, Stair and Hanna, 2009):

- The values used in the objective function and the constraints are known with *certainty*, and will not change for the duration of the solution process.

- *Proportionality* exists between the objective and the constraints, which means that the contribution to the objective function of every decision variable is assumed constant and therefore independent of the variable level (McCarl, 2007).
- *Additivity* is assumed which means that the total of the factors determining the objective equals the sum of all of the individual factors.
- *Divisibility* is assumed, meaning that the solution is not restricted to being an integer.
- *Non-negativity* of all of the decision variables is assumed.

A linear program can be written in the general form with the objective function shown in (2-1), and the constraints in (2-2) and (2-3).

$$\text{Minimise / maximise} \quad \sum_{j=1}^n c_j x_j \quad (2-1)$$

$$\text{subject to} \quad \sum_{j=1}^n a_{ij} x_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_i \quad (i = 1, \dots, m), \quad (2-2)$$

$$x_j \geq 0 \quad (j = 1, \dots, n), \quad (2-3)$$

where x_j represents the decision variables,

c_j denotes the contribution of x_j to the objective function and is called an *objective coefficient*,

a_{ij} is called a *technical coefficient* and depicts the contribution of one unit of x_j in the i^{th} constraint,

b_i is the upper- or lower limit imposed by the i^{th} constraint,

for n decision variables and m constraints.

The formulation process involves the identification of the discussed components and writing the mathematical expressions for the objective function and the constraints.

2.2.2. Solving a linear programming model

This section aims to briefly describe the graphical- and simplex solution methods for linear programs. Consider the following example problem from Render *et al.* (2009) which will be used for the purpose of illustrating these solution methods:

$$\text{Maximise} \quad 7x_1 + 5x_2, \quad (2-4)$$

$$\text{subject to} \quad 4x_1 + 3x_2 \leq 240 \quad (2-5)$$

$$2x_1 + 1x_2 \leq 100 \quad (2-6)$$

$$x_1, x_2 \geq 0 \quad (2-7)$$

2.2.2.1. Graphical solution method

The graphical method is limited to be used only in very trivial linear programming models that have only two decision variables because the method entails the graphical representation of the constraints in a 2-dimensional graph.

Figure 2-1 represents the 2-dimensional graphical plot of the example model described by (2-4) to (2-7).

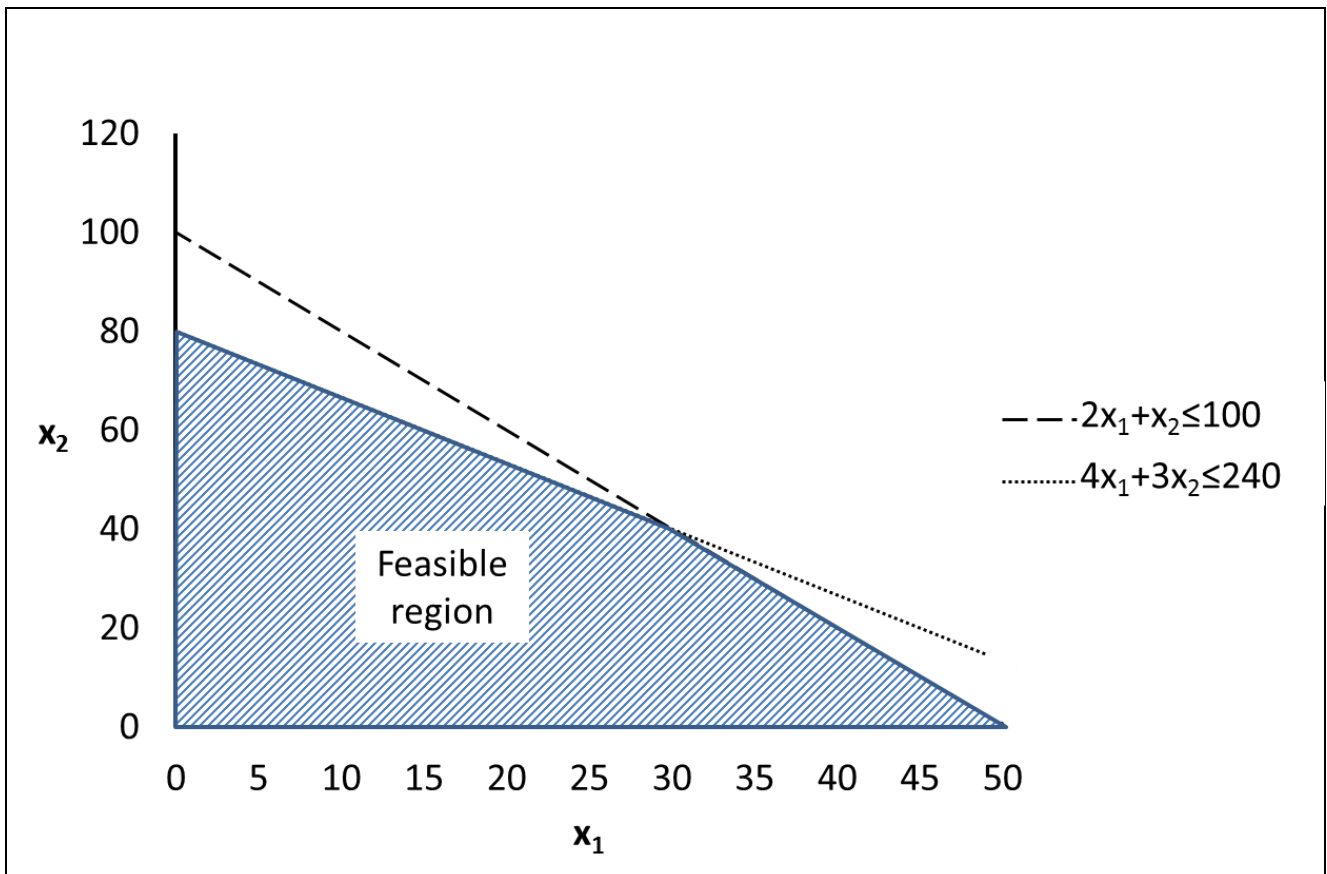


Figure 2-1: Graphical representation of the objective function and constraints in the example model

The shaded area in the picture represents the permissible region where a feasible solution can be found. The corners of the feasible region in the 2-dimensional example are called *extreme points*.

Finding the optimal solution can be done in one of two ways:

1. The *isoprofit line method*. This method involves plotting a line with the same slope as that of the objective function on the graph and moving the line upwards in a “maximising” direction. The last extreme point of the feasible region that the line touches before exiting the region is where the maximum value is found. In a special case, the slope of the objective function is equal to that of one of the constraints which means that all points on that edge of the feasible region will represent the same maximum solution. The isoprofit line method is illustrated in Figure 2-2.

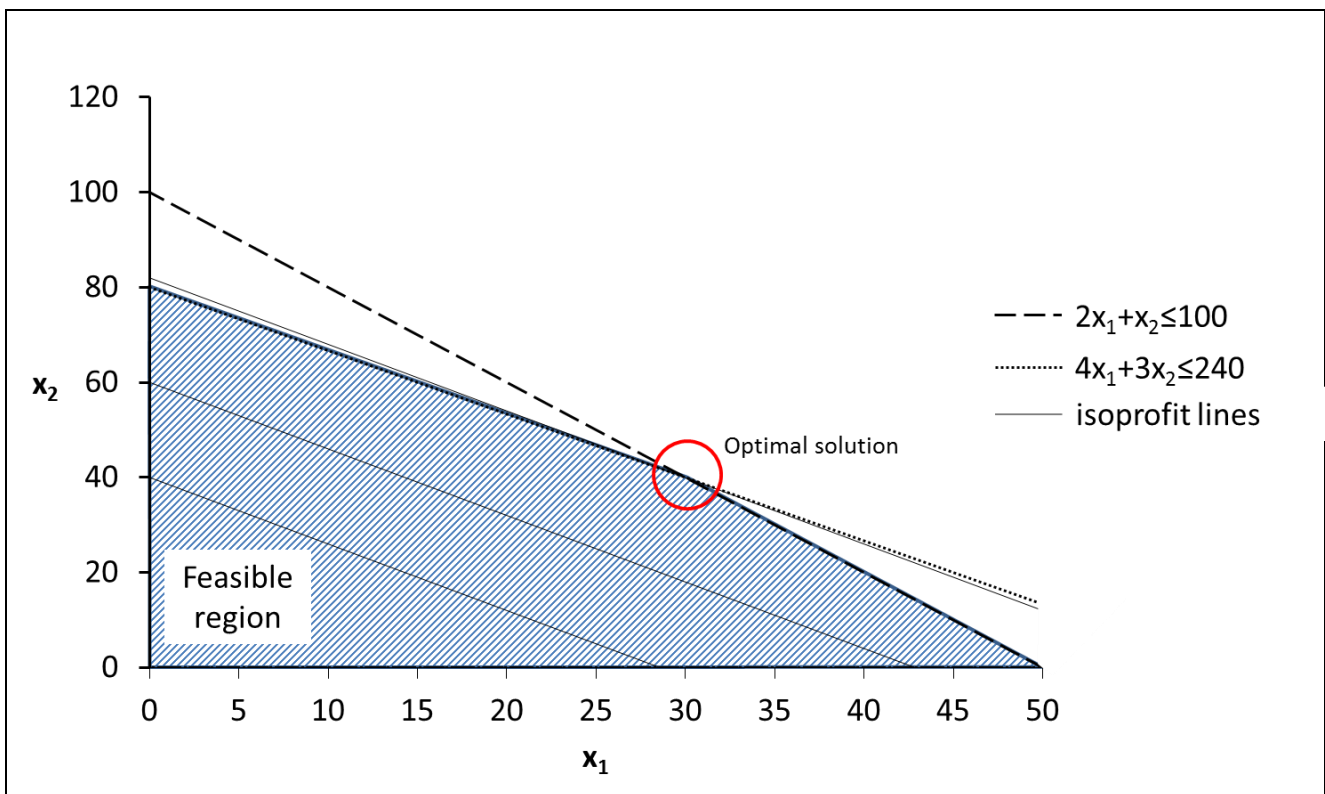


Figure 2-2: Isoprofit line solution of the example linear program

2. The *corner (extreme) point solution method*. As the name says, this method involves inspecting the objective function value at each of the extreme points of the feasible region by substituting the values for the decision variables into the objective function. The extreme point that produces the maximum value is chosen as the optimal solution. This is illustrated in Figure 2-3.

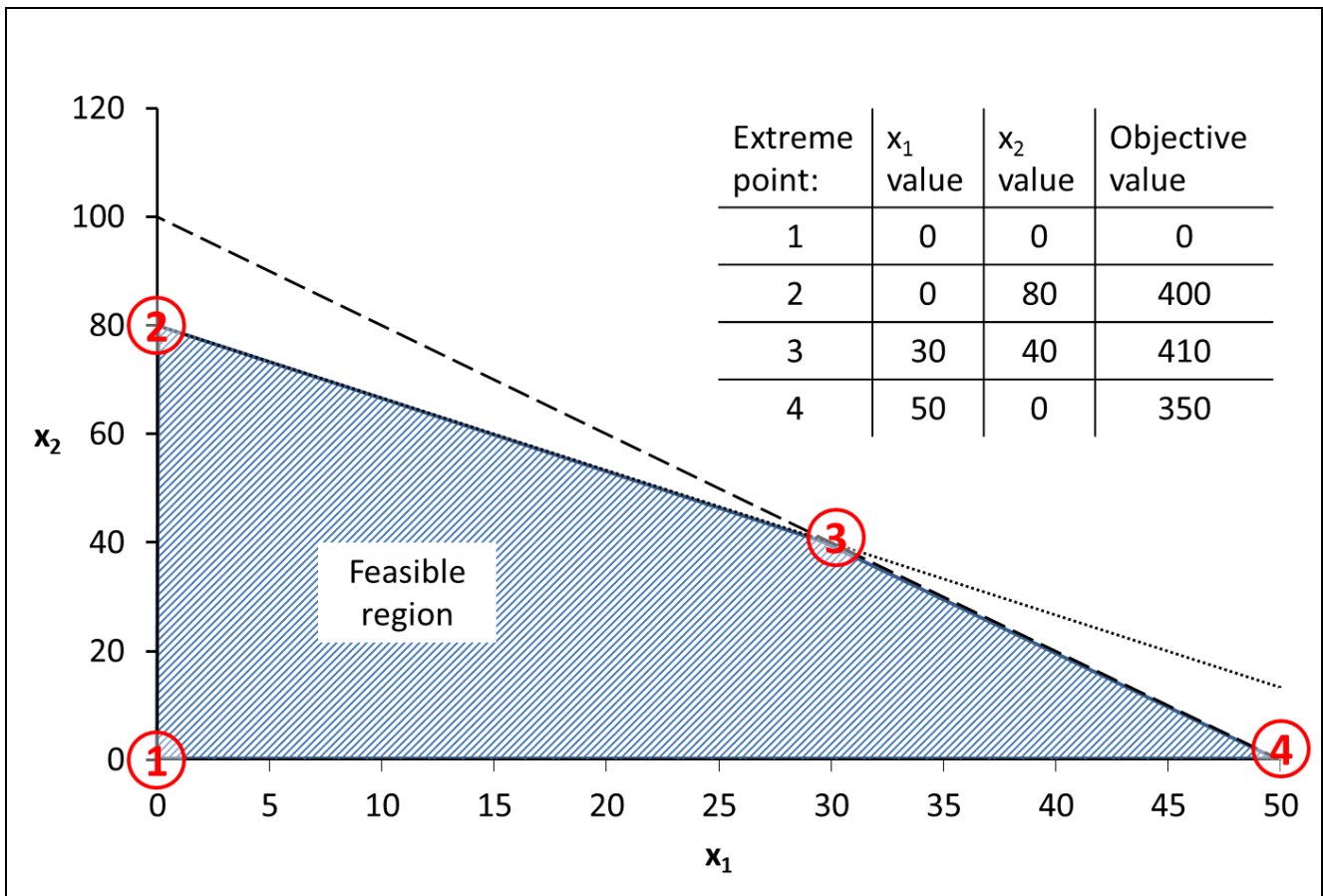


Figure 2-3: The corner point solution method

The figure shows that the optimum objective value of 410 is at extreme point 3, which echoes the answer of the isoprofit line method.

The next section discusses the simplex method which is used to solve linear programs with two or more decision variables.

2.2.2.2. The simplex method

In general, linear programming models have a very large number of decision variables which means that representing the feasible region of such problems on a 2-dimensional graph is not possible. However, the optimal solution will still be found at one of the extreme points of the multi-dimensional, multi-extreme point feasible region.

The simplex method uses basic algebraic concepts to iteratively and systematically evaluate the extreme points until an optimal solution is found (Rolf, 2013). It offers a more efficient way to reach an optimal solution by limiting the search for the optimal extreme point(s) to a reasonably small number of corners.

A linear programming problem is typically presented in the form of a set of tableaux (matrices) on which row-operations are performed to determine the optimal solution. The basic function of the simplex method therefore, is to use an existing feasible solution for the problem and improve upon that solution until the optimal is found.

Performing the simplex method can sometimes create situations where finding a feasible solution poses some challenges. Problems like these are discussed in section 2.2.3. The mathematical and theoretical motivation of the simplex method will not be discussed in detail, however for the purpose of demonstrating the operation of the simplex method, consider the following basic example problem.

$$\text{Maximise} \quad z = 3x_1 + x_2 + 2x_3 \quad (2-8)$$

$$\text{subject to} \quad x_1 + x_2 + x_3 \leq 15 \quad (2-9)$$

$$3x_1 + 2x_2 - 2x_3 \leq 18 \quad (2-10)$$

$$x_1, x_2, x_3 \geq 0 \quad (2-11)$$

For each of the constraints in (2-9) and (2-10) the value of the left-hand side equals at most the value of the right-hand side (RHS), with a possible slack value between the values. The constraints in the model are converted to equations by introducing *slack variables* s_1 and s_2 that represent the slack values of the constraints. This would result in the following set of equations:

$$x_1 + x_2 + x_3 + s_1 = 15 \quad (2-12)$$

$$3x_1 + 2x_2 - 2x_3 + s_2 = 18 \quad (2-13)$$

and the objective function is written as:

$$-3x_1 - x_2 - 2x_3 - 0s_1 - 0s_2 + z = 0 \quad (2-14)$$

This set of equations is written in a matrix and presented in tableau form as follows:

x_1	x_2	x_3	s_1	s_2	z	RHS	Basis
1	1	1	1	0	0	15	s_1
3	2	-2	0	1	0	18	s_2
-3	-1	-2	0	0	1	0	

Table 2-1: The first simplex tableau

Step 1:

An initial basic feasible solution corresponding with an extreme point is found by assigning the values $s_1 = 15$ and $s_2 = 18$ to the slack variables and setting the decision variables $x_1, x_2, x_3 = 0$. This results in an initial goal function value of $z = 0$. In the tableau the columns under s_1 and s_2 form unit vectors and s_1 and s_2 are therefore called *basic variables*. The basic variables constitute a *basis* and consist of the variables solved, whereas $x_1, x_2,$ and x_3 are *non-basic variables* with values of zero.

Step 2:

The current tableau is manipulated by means of a so-called *pivot process* so that an improved goal function value (currently $z = 0$) can be found. For this purpose the *pivot element* (p_e) is found for the current tableau. This is the element that lies at the intersection of the *pivot column* and the *pivot row*:

- The pivot column is the column with the lowest negative value in the last row of the tableau and identifies the variable to enter the basis on the next iteration;
- The pivot row is found by dividing each right-hand side value (except that in the last row) with the corresponding entries in the pivot column and selecting the smallest non-negative ratio. This identifies the variable leaving the basis on the next iteration.

This process is illustrated in Figure 2-4.

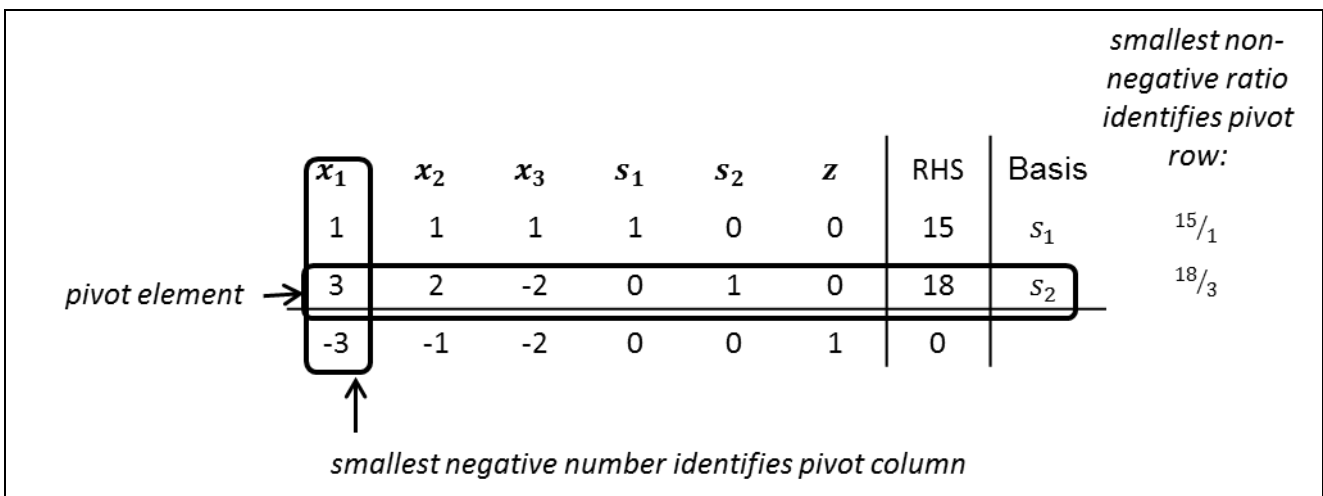


Figure 2-4: Determining the pivot element

Step 3:

To improve on the goal function value and set up the new tableau, the pivot element is utilised to perform basic row operations on the current tableau. In the example, this is accomplished by performing the following row operations:

$$Row1' \leftarrow Row1 - \frac{p_e}{p_{1c}} Row2 \tag{2-15}$$

$$Row2' \leftarrow \frac{Row2}{p_e} \tag{2-16}$$

$$Row3' \leftarrow Row3 + \frac{p_e}{p_{3c}} Row2 \tag{2-17}$$

where p_e is the pivot element,

p_{1c} is the value in $Row1$ (non-pivot row) in the pivot column,

p_{3c} is the value in $Row3$ (last row) in the pivot column.

In the new tableau the element corresponding with the current pivot column will be a unit vector with the 1 corresponding with the (current) pivot element. These operations yield the second simplex tableau which is shown in Table 2-2.

x_1	x_2	x_3	s_1	s_2	z			Basis
0	$\frac{1}{3}$	$\frac{5}{3}$	1	$-\frac{1}{3}$	0	9		s_1
1	$\frac{2}{3}$	$-\frac{2}{3}$	0	$\frac{1}{3}$	0	6		x_1
0	1	-4	0	1	1	18		

Table 2-2: The second simplex tableau

The basis consists of x_1 and s_1 and the *non-basic variables* $x_2, x_3,$ and s_2 are equal to zero. The solution provided by this tableau is $x_1 = 6, x_2 = 0, x_3 = 0, s_1 = 9, s_2 = 0$ and goal function value $z = 18$ which is an improvement on the initial $z = 0$.

Step 4:

If there are any negative coefficients in the last row of the tableau, it may be possible to improve the goal function value. The solution in the second tableau may not be optimal because the negative entry -4 appears in the last row.

Step 5:

To improve on the goal function value, steps 2 and 3 are therefore repeated. This yields the third tableau shown in Table 2-3.

x_1	x_2	x_3	s_1	s_2	z			Basis
0	$\frac{1}{5}$	1	$\frac{3}{5}$	$-\frac{1}{5}$	0	$\frac{27}{5}$		x_3
1	$\frac{4}{5}$	0	$\frac{2}{5}$	$\frac{1}{5}$	0	$\frac{48}{5}$		x_1
0	$\frac{6}{5}$	0	$\frac{12}{5}$	$\frac{1}{5}$	1	40		

Table 2-3: The third simplex tableau

The basic feasible solution provided by this tableau is $x_1 = \frac{48}{5}$, $x_2 = 0$, $x_3 = \frac{27}{5}$, $s_1 = 0$, $s_2 = 0$, $z = 40$. There are no negative entries in the last row indicating that $z = 40$ is the optimal solution.

Using the simplex method to solve a minimization problem generally involves maximising the negative of the given goal function. For more detail and comprehensive examples, refer to Haeussler, Paul and Wood (2008). Greater-than-or-equal-to and equality constraints are discussed in section 2.2.3.

2.2.3. Special cases

There are some special cases which can arise and complicate the process when solving a linear program with one of the discussed methods (Render *et al.*, 2009).

- *Greater-than-or-equal-to constraints.* To handle a greater-than-or-equal-to type constraint, a surplus variable is introduced in a similar way than that of a slack variable and is subtracted to deliver an equality constraint.
- *Equality constraints.* An initial equality constraint is handled by introducing an artificial variable which is used solely for generating an initial basic feasible solution. The values of these variables are driven to zero during the course of the solution process. For more detail on using surplus and artificial variables, refer to Render *et al.* (2009) and Haeussler *et al.* (2008).
- *Infeasibility.* If two or more constraints conflict with one another, the model has no feasible solution and the problem should be redefined and formulated properly. When working with the simplex method, an infeasible solution can

be identified when a tableau has no negative values in the last row, but one or more artificial variables are still in the basis.

- *Unboundedness.* Sometimes a linear program does not have a finite solution, for example when a profit maximization problem has an infinitely large profit because of some missing constraints. A problem like this is called *unbounded* and is found in the simplex method when an entering variable exists for a next tableau but no leaving variable can be identified due to non-positive pivot column values.
- *Alternate optimal solutions.* Consider a problem being solved with the graphical solution method, with a constraint that is parallel to the objective function: all the points on the intersection between the constraint and the objective function will be optimal solutions. Typically in a large company the advantage this will provide is that management has some flexibility in deciding which solution to select. When using the simplex method, alternate optimal solutions can be identified when, in the final tableau, a value exists in the last row that is equal to 0 and is not a basic variable.
- *Degeneracy.* In the case of the simplex method, this develops when the number of constraints passing through one extreme point exceeds the number of decision variables in the problem. Degeneracy can be identified when the RHS/pivot column ratio calculations are made and there is a tie for the smallest value and the pivot element cannot be determined uniquely, or the RHS value equals zero. It could lead to a state known as *cycling*, which means that the simplex algorithm cycles between the same non-optimal solutions indefinitely.

2.2.4. Sensitivity analysis

Solutions for linear programs are calculated under the assumption that the conditions are not to change for the duration of the solution process. In reality however, due to a variety of business-related reasons many changes can occur that will influence the final solution.

Sensitivity analysis is done to examine the effect of changes in the contribution rates of the decision variables in the objective function, the coefficients in the constraints (technical coefficients) and the available resources (right-hand sides of the

constraints). Sensitivity analysis allows managers in companies to experiment with the values of the input parameters to the problem.

Changes in the values of the objective coefficients will cause the slope of the objective function to change which will not necessarily change the optimal solution. Sensitivity analysis provides the *objective coefficient ranges* which give the range of change for each objective coefficient over which no change in the optimal solution will occur.

Changes in the technical coefficients and RHS values will have no effect on the value of the objective function but will affect the shape of the feasible region and could therefore influence the value of the optimal solution.

Sensitivity analysis can be used to evaluate the variation in the value of the objective function when, for example, small changes occur in the resource quantities, objective coefficients and/or technical coefficients. The rate of change for each of the resources is called the *shadow price*.

More details on sensitivity analysis concepts and calculations can be found in Rolf (2013), Render *et al.* (2009), and Moore and Weatherford (2001).

2.3. Integer linear programming

In linear programming models the decision variables are permitted to take on fractional values in the optimal solution. In some applications however, the decision variables are required to be integer valued (Taylor, 2013).

Consider the example of an airline company that wants to know how many Boeing 737s and Boeing 777s it needs to buy to maximise its profit: the company cannot order a total of 2.5 airplanes. Such a problem can be solved using an *integer linear program* (ILP) which differs from a linear program in that one or more of the decision variables, will be required to take on an integer value in the optimal solution.

There are three common types of integer programming problems:

In an *all-integer linear program* all of the decision variables are required to be integers. The standard form for such a program is shown in (2-18) to equation (2-20).

$$\text{Minimise / maximise} \quad \sum_{j=1}^n c_j x_j \quad (2-18)$$

$$\text{subject to} \quad \sum_{j=1}^n a_{ij} x_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_i \quad (i = 1, \dots, m) \quad (2-19)$$

$$x_j \geq 0 \text{ and integer} \quad (j = 1, \dots, n) \quad (2-20)$$

where the variables and coefficients are as described in (2-1) to (2-3).

In a *mixed integer linear program* (MILP), only some of the variables are required to be integer.

In a *binary (or 0-1) integer linear program*, the integer variables are further restricted to take on only the values 0 or 1. These types of constraints are often used to represent true or false decisions. The standard form for a binary integer linear program is the same as in (2-17) to (2-19) but the constraint in (2-20) changes to (2-21).

$$x_j \text{ binary} \quad (j = 1, \dots, n). \quad (2-21)$$

2.3.1. Logical conditions

One of the most important uses of binary integer linear programs is to impose constraints that represent logical conditions which take on true or false values (Williams, 2013). Some very specific problems exist in which binary variables are utilised for particular purposes.

In a problem where not more than a specific number of variables should be chosen, binary values are introduced. Suppose that from 3 binary variables x_1, x_2 , and x_3 , at most only one should be chosen. The constraint can be created like this:

$$x_1 + x_2 + x_3 \leq 1. \quad (2-22)$$

However, binary variables can also be used to force a dependent relationship upon two or more decisions. Suppose x_1 may only be chosen if x_2 was chosen, then the following constraint can be created:

$$x_1 \leq x_2. \quad (2-23)$$

Another use of binary variables is to restrict the size of a lot or batch of items that are allowed in the solution. Suppose for example that if x_1 is chosen, its value must be

between an upper and a lower bound $l \leq x_1 \leq u$. However this cannot be used as a constraint in a linear program. The introduction of a binary variable y_1 makes it possible to incorporate the specification into a linear program with the following constraints:

$$x_1 \geq ly_1 \text{ and } x_1 \leq uy_1, \text{ with } y_1 \text{ binary.} \quad (2-24)$$

Some models require that only a certain number of the constraints in the model be satisfied. Let the set of m constraints on a model's n -amount of non-binary variables be $g_i(x_1, \dots, x_n) \leq b_i, i = 1, \dots, m$. Each g_i is a given constraint function on the n decision variables x_i . Introduce m additional binary decision variables y_1, \dots, y_m to the model, and choose M as a very large number so that for every $i, g_i(x_i, \dots, x_n) \leq M$ for every x satisfying any set of k inequalities taken from the above m . Then the next $m + 1$ constraints are expressed as:

$$\sum_{i=1}^m y_i = k \quad (2-25)$$

$$g_i(x_i, \dots, x_n) \leq b_i y_i + M(1 - y_i) \quad (i = 1, \dots, m) \quad (2-26)$$

$$y_i \text{ binary} \quad (i = 1, \dots, m). \quad (2-27)$$

The constraint in (2-25) forces a total of k of the additional decision variables y_i to have the value 1. The very large choice of the number M makes the remaining constraints redundant so that they do not affect the optimal solution.

Solving an integer linear program differs from the method for a standard linear program. The next section discusses some solution methods.

2.3.2. Solving an ILP by means of the graphical representation method

Moore and Weatherford (2001) explains the steps to optimizing an ILP by means of the graphical representation method. It starts by following the same approach as described in section 2.2.2.1 to determine the feasible area. The graphical representation of the model entails the identification of all the integer intersections inside the feasible area.

Consider the following example problem for illustration purposes:

$$\text{Maximise} \quad 7x_1 + 5x_2 \quad (2-28)$$

$$\text{subject to} \quad 4x_1 + 3x_2 \leq 19 \quad (2-29)$$

$$2x_1 + 1x_2 \leq 8 \quad (2-30)$$

$$x_1, x_2 \geq 0 \text{ and integer} \quad (2-31)$$

Consider the reasoning of the isoprofit line and the corner-point methods. Figure 2-5 shows that although the intersection of the constraints (2-29) and (2-30) would yield a higher objective value of 32.5 at point A, this would entail a value of 2.5 for x_1 and 3 for x_2 which would mean a violation of the integer constraint (2-31). The optimal objective value of 32 with $x_1 = 1$ and $x_2 = 5$ is found at point B with all the constraints satisfied.

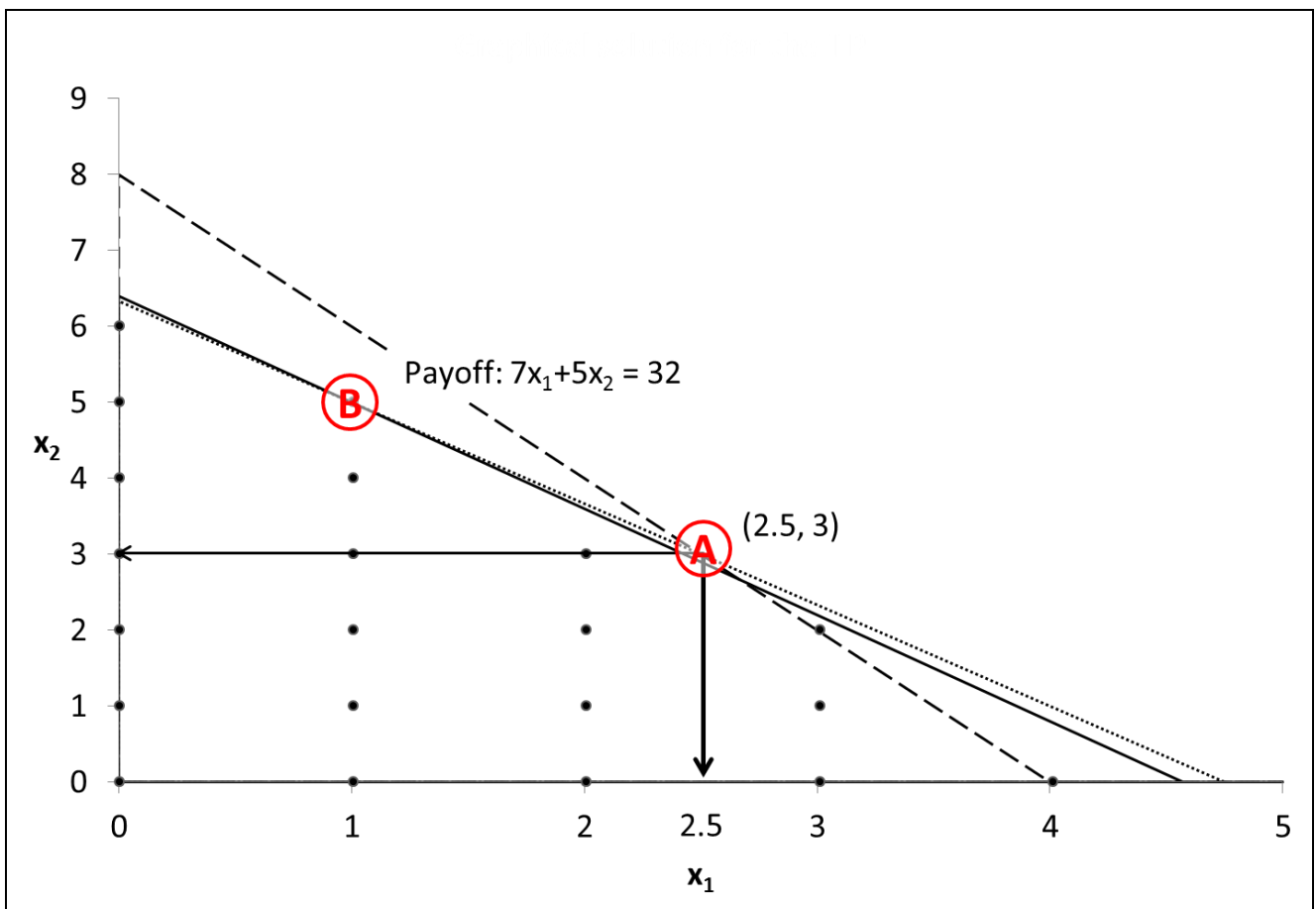


Figure 2-5: Graphical solution for the ILP example model

The graphical representation method can be used to successfully solve simple ILP models with 2 decision variables. However in general many problems exist that are more complex and have thousands of decision variables with integer constraints that

need to be solved with alternative methods. One such method is the branch-and-bound method which will be discussed in the next paragraph.

2.3.3. The branch-and-bound method for ILP

The branch-and-bound method is the most commonly used algorithm for solving ILPs. Branching-and-bounding performs a shrewdly designed search of some related linear programming problems, comparing them with each other to find the optimal integer solution (Vanderbrei, 2008).

The idea of the branch-and-bound method is to subdivide the feasible region of a problem and calculate upper and lower bounds on the optimal objective value of the resulting subproblems or *branches* (Pfeuffer, Stigmayr and Klamroth, 2012). A subproblem is solved if the upper and lower bounds meet, otherwise the problem should be subdivided into further subproblems. If the calculated bounds indicate that an optimal solution does not exist in that subproblem, the branch may be *pruned*. Upon completion of the algorithm, the optimal solution among all those calculated in the process is the optimal solution for the overall problem.

Finding a solution for a maximization problem by means of the branch-and-bound method involves the following steps (Vanderbrei, 2008; Eiselt and Sandblom, 2000):

1. Find an initial solution for the relaxed ILP, in which the integrality constraints are discounted, by means of linear programming and if the solution also satisfies the integer constraints the problem has been solved. Otherwise, the initial objective function value is considered as the initial upper bound for the ILP.
2. Find a feasible solution that meets the integrality constraints. This can be done for each decision variable by considering the integer value smaller than the decision variables in order to obtain a lower bound for the problem.
3. Select any non-integer variable from step 1 that is required to be integer and branch the problem into two sub-problems by selecting the integer values immediately above and immediately below the chosen non-integer value and defining them as constraints.
4. Solve the new problems to create new nodes at the end of these branches.
5. Inspect the new branches:
 - a. Terminate a branch if it gives an infeasible solution;

- b. If the solution provided by a branch is feasible but not an integer solution, proceed to step 6;
 - c. If a feasible integer solution is found, calculate the value of the objective function. If it equals the upper bound, an optimal solution has been found. If it is less than the upper bound and higher than the lower bound, set it as the new lower bound. If the objective value is lower than the lower bound, terminate the branch.
6. Calculate the values for the objective function at both of the newest branches and set the upper bound equal to the maximum. If this upper bound equals the lower bound an optimal solution has been found and the investigation can stop, or else start the next iteration from step 3 onwards.

The example from section 2.3 will be used to illustrate the application of the steps outlined above, while the iterations are shown graphically in Figure 2.6.

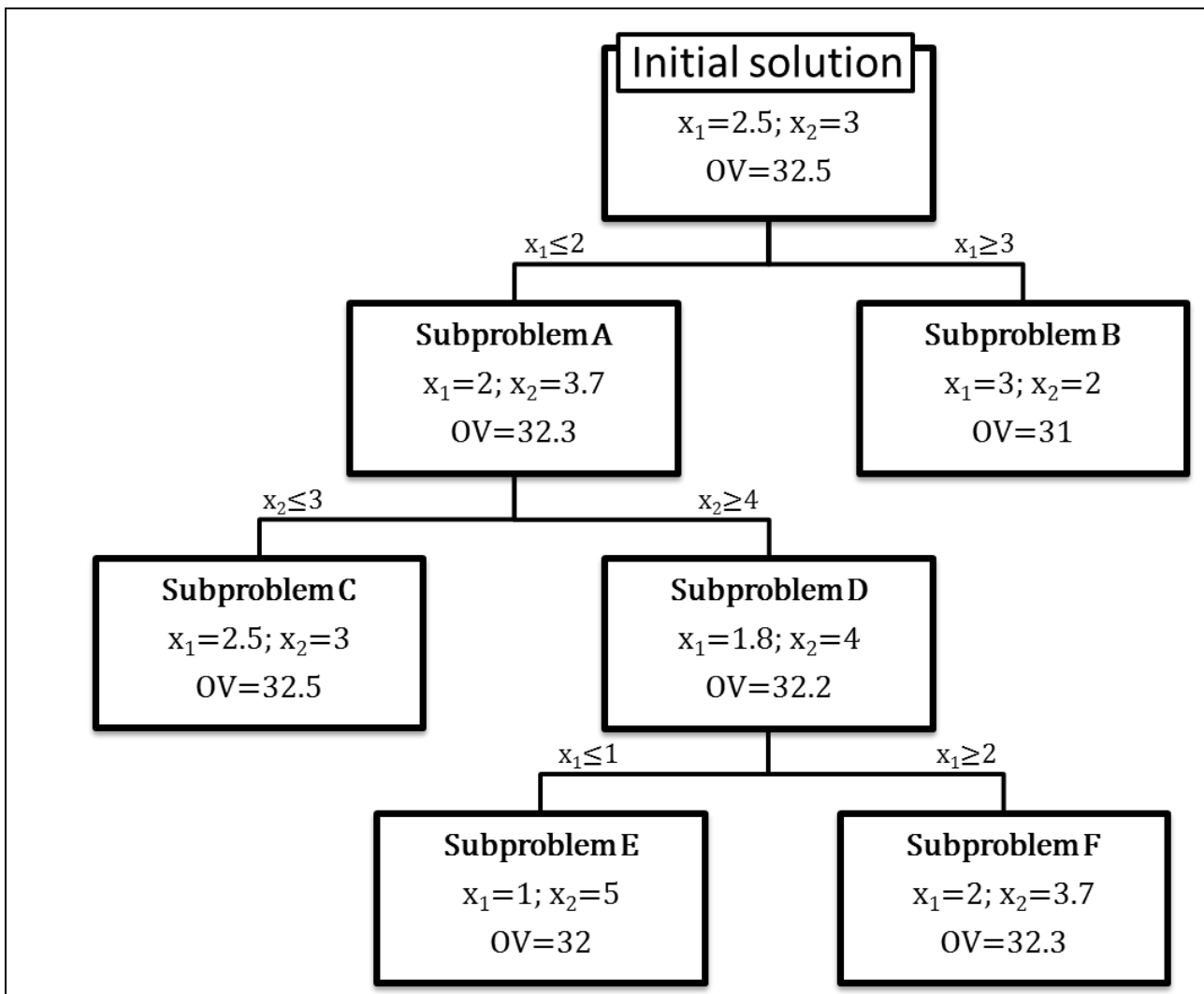


Figure 2-6: The branch-and-bound method applied to the example problem

The initial feasible non-integer solution of $x_1 = 2.5$ and $x_2 = 3$ with an objective value of 32.5 completes step 1 of the branch-and-bound method for this example. This solution is used to set 32.5 as the initial upper bound, and rounding down of the variables provides the lower bound of 29, with $x_1 = 2$ and $x_2 = 3$. The first two subproblems, A and B, are created by introducing additional constraints on variable x_1 as this is the only non-integer variable in the initial solution.

The solution resulting from subproblem B is feasible and the objective value of 31 does not equal the upper bound but it is higher than the lower bound. The value of 31 is therefore set as the new lower bound according to step 5(c). Subproblem B cannot be branched further because no non-integer decision variables exist in the solution.

The decision variables in subproblem A have non-integer values and the problem has a feasible non-integer solution with an objective value of 32.3. The new upper bound is therefore set to this value (step 5(c)) and this branch is expanded further.

Steps 1 to 6 are repeated iteratively up to subproblem E, which yields a feasible solution that satisfies all of the constraints with an objective value of 32.

When working with a minimisation problem, the roles of the upper and lower bounds are reversed. A more detailed discussion of this method can be found in Taylor (2013).

2.4. Goal programming

A goal programming model typically has a set of goals that need to be reached where the main aim of the program is to meet these goals as closely as possible (Jones and Mehrdad, 2010). In situations where many goals exist that need to be met, it is most probable that some or all of the goals conflict with each other. Consider for example, a case where a company wishes to reduce its overhead costs but also wants to reach a larger target population by improving its market exposure. Cutting overhead costs relates to saving money while broadening market exposure will mean spending more money. Problems like these can be solved using goal programming techniques which aim to find a solution that is as close as possible to both of the goals in the case of the example scenario.

Goal programming entails the use of linear programming techniques on a model in which multiple goals need to be optimized with the exception that optimal targets or allowable levels for the goals are known (Render *et al.*, 2009). In the example problem, the manager will typically provide a target for the amount to be cut from the overheads as well as a maximum target the company is willing to spend on marketing. The goals are then converted to constraints by adding certain variables to the model and optimising the variables.

Sometimes constraints exist that must not be violated, for example when time is an issue. Such constraints are called hard or *system constraints* whereas constraints that may be violated are called soft or *goal constraints* and the model will attempt to generate a solution that is as close as possible to the goal constraints.

Deviation variables are introduced into the model for this purpose. They do not have any effect on the goals but are the only variables in the objective function. In English the problem would read: “Minimize the deviation from the goals.” In mathematical terms, the model is formulated as follows:

$$\text{Minimise} \quad \sum_{i=1}^m (d_i^- + d_i^+) \quad (2-32)$$

$$\text{subject to} \quad \sum_{j=1}^n (a_{ij}x_j) + d_i^- - d_i^+ = b_i \quad (i = 1, \dots, m) \quad (2-33)$$

$$d_i^-, d_i^+ \geq 0 \quad (i = 1, \dots, m) \quad (2-34)$$

$$x_j \geq 0 \quad (j = 1, \dots, n) \quad (2-35)$$

where x_j represents the decision variables,

m is the number of goal constraints,

n is the number of decision variables,

d_i^- represents the underachievement of goal i ,

d_i^+ denotes the overachievement of goal i ,

a_{ij} depicts the use of the items in the i^{th} goal constraint by one unit of x_j ,

b_i is the upper- or lower limit imposed by the i^{th} goal constraint.

With the variables d_i^- and d_i^+ in the objective function, the linear program will attempt to minimise both. In a system constraint, the values of d_i^- and d_i^+ would be zero so

that there is no deviation from the required level or target. The objective is therefore to minimize the total sum of all deviations.

Goal interval constraints are used in goal programming to restrict the goal to a range of numbers or an interval, rather than a single number. The constraint set in (2-33) can also be written as the following:

$$\sum_{j=1}^n (a_{ij}x_j) + d_i^- \geq b_i^- \quad (i = 1, \dots, m) \quad (2-36)$$

$$\sum_{j=1}^n (a_{ij}x_j) - d_i^+ \leq b_i^+ \quad (i = 1, \dots, m). \quad (2-37)$$

where b_i^- is the lower limit imposed by the i^{th} goal constraint and b_i^+ is the upper limit imposed by the i^{th} goal constraint.

Emrouznejad and Ho (2012) names two algorithms that are used for solving a goal programming model that include the *weights method* and the *preemptive method* which is discussed briefly in the next two sections.

2.4.1. The weights method

This method sees the introduction of positive weights to the goals of the problem which represent the importance of the goals. A typical weighted goal problem is shown in (2-38).

$$\text{Minimise} \quad w_1d_1^- + d_1^+ + w_2d_2^- + d_2^+ \quad (2-38)$$

Weight values of $w_1 = 1$ and $w_2 = 2$ would mean that d_2^- carries a higher priority than d_1^- . The multiple goals are effectively converted into one single objective function and solved using linear programming techniques.

2.4.2. The preemptive method

The preemptive method represents the importance of the goals according to priority levels P_1, P_2, \dots, P_n where $P_1 > P_2 > \dots > P_n$.

For instance if d_3^+ has a higher priority than d_2^- which in turn is more important than d_1^- , then the algorithm would assign the priorities and define the objective function as follows:

$$\text{Minimize} \quad P_3 d_1^- + P_2 d_2^- + P_1 d_3^+ \quad (2-39)$$

The method acts proactively by solving one goal at a time in order of priority. Once the highest prioritized goal has been satisfied, the next one is considered. The solution process stops when a solution from a lower-prioritized goal degrades a solution found for a higher-prioritized goal.

A goal programming model can be solved graphically (section 2.2.2.1) or by utilising the simplex method as discussed in section 2.2.2.2. For comprehensive examples and solutions refer to Moore and Weatherford (2001).

2.5. Multi-objective programming

As the name implies, *multiple objective linear programming* (MOLP) problems are linear programming problems requiring multiple objective functions to be optimised (Gass, 2003) with the difference that no targets are known before the problem is solved. Part of the solution process is to determine the targets for the different objectives and then find the optimal solution (Moore and Weatherford, 2001).

Models like these are required in cases where more than one objective, usually conflicting with others, need to be optimized. The problem with conflicting objectives in a linear program is that when one of the objectives performs better, it is usually at the expense of another. For example when a manager wants to minimize cost, it could be at the expense of quality. The ideal is to find the optimal compromise or *trade-off* between the conflicting objectives.

Assume for example that a girl has a limited budget and would like to keep the total cost of her diet to a minimum (objective 1). Unfortunately she has very expensive taste in food and wants to follow a healthy eating-plan containing most of her favourite foods (objective 2).

Consider the illustration presented in Figure 2-7. Each point on the curve in the graph corresponds to a possible level of cost and the number of favourite food items in the diet. As the number of food items (objective 2) increase, the total cost (objective 1) of the diet also increases. Obviously, it is preferable to keep the cost as low as possible and the number of food items as high as possible.

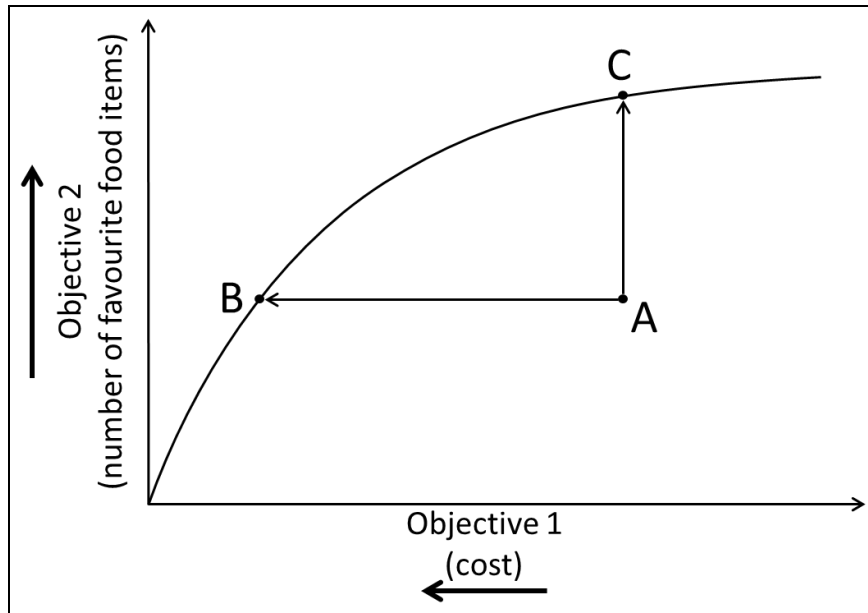


Figure 2-7: Trade-off between two conflicting objectives

Consider point A which represents a combination of cost and number of food items that she definitely would not want. Point B however, represents a diet that contains the same number of her favourite food choices but at a lower cost and is preferable to point A. Similarly, point C represents a diet that contains more of her favourite food items, but at the same cost than at point A. Reaching higher levels of objective 2 will incur higher levels of objective 1. Points B and C are said to *dominate* point A because point B provides a better trade-off in terms of cost and point C in terms of favourite food items.

The trade-off between the two objectives is called *Pareto efficiency* which is a state of allocation of resources where it is impossible to make one resource better off without making at least one other worse off. Finding a solution that makes the state of one resource better without affecting any other resource is called *Pareto improvement*. Points B and C, and all solutions along the curve between these points are said to *dominate* point A. Pareto dominance is defined mathematically by Niu, Ong and Nee, (2013) as:

“For a minimization problem, a feasible solution x_1 is said to Pareto dominate over another feasible solution x_2 , denoted as $x_1 > x_2$, if and only if

$$\forall j \in \{1, 2, \dots, n\}, f_j(x_1) \leq f_j(x_2)$$

$$\exists k \in \{1, 2, \dots, n\}, f_k(x_1) < f_k(x_2).”$$

The decision maker should decide which level of trade-off between the two objectives is the better in each specific case.

To solve a MOLP, the problem can be viewed as a special kind of goal programming model. For each of the objectives, target values are first determined by optimising them separately after which the model is solved using the MINIMAX objective.

The MINIMAX objective is a helpful tool to use with multiple objectives and involves the creation of an additional MINIMAX variable Q denoting the deviation from the optimal target for each of the objectives (Ragsdale, 2007) so that the problem can be stated as:

$$\text{Minimise} \quad Q \quad (2-40)$$

$$\text{subject to} \quad d_i^-, d_i^+ \leq Q, \quad (i = 1, \dots, m) \quad (2-41)$$

$$\sum_{j=1}^n (a_{ij}x_j) + d_i^- - d_i^+ = b_i \quad (i = 1, \dots, m) \quad (2-42)$$

$$d_i^-, d_i^+ \geq 0 \quad (i = 1, \dots, m) \quad (2-43)$$

$$x_j \geq 0 \quad (j = 1, \dots, n) \quad (2-44)$$

where x_j represents the decision variables,

m is the number of goal constraints,

n is the number of decision variables,

d_i^- represents the overachievement of goal i ,

d_i^+ denotes the underachievement of goal i ,

a_{ij} depicts the use of the items in the i^{th} goal constraint by one unit of x_j ,

b_i is the upper- or lower limit imposed by the i^{th} goal constraint.

However, the target values for the objectives still need to be calculated. For this purpose, each of the objectives is optimised individually and the optimum solutions are set as the targets for the respective goals. The percentage deviation for each of the objectives is calculated and used to determine the value of Q . The constraint in (2-41) is then reformulated as:

$$w_i \left(\frac{a_i - t_i}{t_i} \right) \leq Q, \quad (i = 1, \dots, m) \quad (2-45)$$

All the values of the deviation variables must be less than or equal to the variable Q which means that the maximum deviation from all the objectives are minimized. Q will therefore, always be equal to the greatest deviation. The MINIMAX objective is explained further by illustrating an example taken from Ragsdale (2007).

Consider the following illustrative example:

The owner of a mining company has to schedule extra shifts of workers at the two mines that her company operates due to an expected increase in the demand for coal in the area. The expected demand increase per month is shown in Table 2-4.

High-grade coal	Medium-grade coal	Low-grade coal
48 ton	28 ton	100 ton

Table 2-4: Expected demand increase

The costs involved in the scheduling of an extra shift at the two mines are \$40,000 and \$32,000 respectively per month and only one shift can be added each month at each mine. The amount of coal that can be produced in a month's time is summarized in Table 2-5.

Type of coal	Mine 1	Mine 2
High-grade	12 tons	4 tons
Medium-grade	4 tons	4 tons
Low-grade	10 tons	20 tons

Table 2-5: The amount and types of coal produced at the mines

Two disadvantages to producing coal include: the methods applied to extract coal from the mines produce a certain amount of toxic water that poisons the groundwater aquifers; and a certain average of life-threatening accidents occurs per shift each month at both of the mines, see Table 2-6.

	Mine 1	Mine 2
Toxic water	800 gallons	1250 gallons
Accidents	0.2	0.45

Table 2-6: Disadvantages to producing coal

The problem management faces is to determine the number of months to schedule an extra shift at each of the mines so that the expected demand increase can be met. The decision variables are defined as:

x_1 = number of months of extra shifts at mine 1,

x_2 = number of months of extra shifts at mine 2.

The three objectives in this problem are to minimise the extra costs involved in extra shifts, the toxic water resulting from the production process, and the number of life-threatening accidents. The objectives are defined in (2-46) to (2-48) and the constraints in (2-49) to (2-52).

$$\text{Minimise} \quad 40x_1 + 32x_2 \quad (2-46)$$

$$\text{minimise} \quad 800x_1 + 1250x_2 \quad (2-47)$$

$$\text{minimise} \quad 0.20x_1 + 0.45x_2 \quad (2-48)$$

$$\text{subject to} \quad 12x_1 + 4x_2 \geq 48 \quad (2-49)$$

$$4x_1 + 4x_2 \geq 28 \quad (2-50)$$

$$10x_1 + 20x_2 \geq 100 \quad (2-51)$$

$$x_1, x_2 \geq 0. \quad (2-52)$$

If the objectives had target values, the model could be solved in a way similar to a goal programming model. Therefore, targets are determined for the objectives by minimising each one individually and using the solution as target for each respective goal.

Computing the targets for the example problem yields $t_1 = 244$ as the target for the total production cost, $t_2 = 6950$ as the target for the total gallons of toxic water, and $t_3 = 2.0$ as the target number of life-threatening accidents.

Consider the total production. Knowing that the actual total production cost can never exceed the optimum value of 244, the percentage deviation from this goal can be computed as:

$$\frac{(\text{actual value} - \text{target value})}{\text{target value}} = \frac{(40x_1 + 32x_2) - 244}{244}. \quad (2-53)$$

Similarly, the percentage deviation from the toxic water and life-threatening accidents respectively are:

$$\frac{(\text{actual value} - \text{target value})}{\text{target value}} = \frac{(800x_1 + 1250x_2) - 6950}{6950} \text{ and} \quad (2-54)$$

$$\frac{(\text{actual value} - \text{target value})}{\text{target value}} = \frac{(0.20x_1 + 0.45x_2) - 2}{2}. \quad (2-55)$$

The percentage deviations calculated in (2-53) to (2-55) are linear functions of the two decision variables, therefore a single objective function can be created by combining the three equations and weighting them according to importance. The resulting objective is shown in (2-56).

$$\begin{aligned} \text{Minimise } w_1 \left(\frac{(40x_1 + 32x_2) - 244}{244} \right) + w_2 \left(\frac{(800x_1 + 1250x_2) - 6950}{6950} \right) \\ + w_3 \left(\frac{(0.20x_1 + 0.45x_2) - 2}{2} \right). \end{aligned} \quad (2-56)$$

The weights $w_1, w_2,$ and w_3 can be used to control the amount by which any one objective influences the solution. Assume for example that management decides that the total production cost is twice as important as the other two objectives. The weight corresponding to the cost objective $w_1,$ can then be set to half the value of the other two.

The resulting model can be solved by means of goal programming methods or by using the MINIMAX objective which aims to MINImise the MAXimum deviation from any of the goals or, in the example problem:

$$\begin{aligned} \text{Minimise the maximum of } w_1 \left(\frac{(40x_1 + 32x_2) - 244}{244} \right), \\ w_2 \left(\frac{(800x_1 + 1250x_2) - 6950}{6950} \right) \text{ and} \\ w_3 \left(\frac{(0.20x_1 + 0.45x_2) - 2}{2} \right). \end{aligned} \quad (2-57)$$

The MINIMAX variable Q is introduced as the objective so that:

$$\text{Minimise} \quad Q \quad (2-58)$$

$$\text{subject to} \quad w_1 \left(\frac{(40x_1 + 32x_2) - 244}{244} \right) \leq Q, \quad (2-59)$$

$$w_2 \left(\frac{(800x_1 + 1250x_2) - 6950}{6950} \right) \leq Q, \quad (2-60)$$

$$w_3 \left(\frac{(0.20x_1 + 0.45x_2) - 2}{2} \right) \leq Q. \quad (2-61)$$

The model is subject to the constraints set out in (2-49) to (2-52), with all the weights positive constants. Solving the model is done in the same way a linear program is solved. For detailed solution procedures refer to Ragsdale (2007).

Multiple-objective optimization is summarized in the following 9 steps (Ragsdale, 2007):

1. Identify the decision variables.
2. Identify and formulate the objectives in the standard way.
3. Identify and formulate the constraints in the standard way.
4. Determine the target values for the objectives by solving the problem for each individual objective.
5. Write the objectives as goal constraints using the targets computed in step 4 as optimal values.
6. Create a deviation function for each goal which measures the amount by which any solution fails to meet the goal.
7. Assign weights to the deviation functions created in step 6 and create constraints that require the values of the deviated functions to be less than the MINIMAX variable Q .
8. Minimise Q in the resulting problem.
9. If the decision maker is not satisfied with the solution, the weights in step 7 could be adjusted and steps 8 and 9 repeated.

2.6. Chapter summary

The aim of this chapter was to provide sufficient background to the mathematical programming techniques and concepts that will be referred to and used in this study.

The basic concept of linear programming was discussed while explaining conventional definitions and special cases. Sensitivity analysis was also briefly discussed. Some solution techniques, including the simplex method, were also shown and discussed. Integer linear programming, goal programming and multi-objective programming were outlined and explained, using some examples.

This study combines mathematical programming with expert system principles, which compels a chapter on the basic theory and principles of expert system technology. Chapter 3 serves to provide some insight into the definitions and concepts common to expert systems and also offers a layout of the steps necessary to develop an expert system.

Chapter 3 – Expert systems principles and development

3.1. Introduction

Computer technology is widely used to aid in solving complex problems in a variety of different fields. Programs are generally developed to follow some sequential set of instructions or procedures and execute them accordingly. Some problems however, cannot be solved using conventional algorithms because additional information is needed to formulate and solve them. Solutions to such types of problems are not always provided in the form of definite answers but rather as conclusions or recommendations based on the given input data.

Systems like those discussed are generally termed *intelligent systems* and are classified as a subfield of Artificial Intelligence (AI) (Liao, 2005). The earliest and still popular definition of AI is to make computers think like people (Russel and Norvig, 2010).

An expert system is created in this study in conjunction with other programming techniques and solution methods. For that purpose it is important to explore the basic principles and development techniques of expert systems.

This chapter defines expert systems concepts and principles, and explains how the mechanics of an expert system relates to human thought-processes. It further discusses the components essential to an expert system and presents the steps involved during the development of such a system.

3.2. Expert systems explained

Giarratano and Riley (2005) provides basic definitions, concepts and principles about expert systems and developing expert systems. An *expert system* is defined as an intelligent computer program that uses expert knowledge and inference procedures to solve problems that require advanced human expertise. The development process therefore involves close interaction between the system programmer and a human expert in the relevant application field.

When developing an expert system, the programmer tries to make a computer emulate a human expert in the way he makes decisions. Therefore, *cognitive science* or, the study of how humans process information (Thagard, 2012), plays an important role when creating the system. A programmer typically has to make the computer understand the reasoning process of the expert and develop a system so that the computer follows a similar thinking process. For this purpose, the specific domain in which the reasoning is to take place should be clearly defined.

An *expert* can be defined as a person who possesses a high level of skill in a specific field that has been obtained through experience, education, or both. The knowledge that a human expert has is unlike general problem-solving methods, specific to a certain field like medicine, finance, science, or engineering and is called the *problem domain* (Giarratano and Riley, 2005). Within that field, somebody who specialises typically has the knowledge and experience to solve very specific problems. The knowledge such an expert possesses, is called the *knowledge domain*. The relationship between the problem domain and the knowledge domain is shown in Figure 3-1.

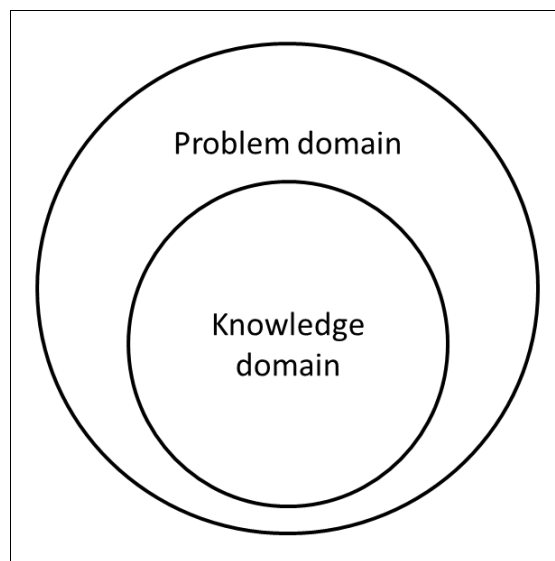


Figure 3-1: Problem domain vs. knowledge domain

The knowledge domain, e.g. paediatrics, is encompassed by the problem domain, e.g. the medical field, and an expert usually does not have enough knowledge to solve problems outside of his knowledge field. An expert system that emulates a human expert must therefore, retain expert knowledge within the knowledge domain

of the specific field and have complex thought and reasoning abilities, similar to that of a human expert.

3.3. Knowledge representation techniques

Much of human understanding and reasoning can be expressed by basic IF-THEN rules (Liao, 2005) which are called *production rules*. Production rules are among those knowledge representation methods that are used in building expert systems (Wang, Bai, Cai and Yan, 2010) and consist of the individual rules that constitute an expert system. A *rule* is a small, modular collection of knowledge which is called a *chunk* (Boose, 1986). A production rule comprises of an antecedent which is also called the conditional part or left-hand side, and the consequent also referred to as the right-hand side which can represent an action to be taken or a conclusion to be made (Lindqvist and Porras, 1999). A rule is activated when the conditions in the antecedent are satisfied. For example, a human would reason that “IF I am hungry, THEN I eat”, the person will only eat if the condition holds.

It is believed that human memory is organised in chunks that exist independently but are linked through possible relationships. Emulating that in a computer would mean the earlier example can be expanded to include another chunk: “IF I am hungry AND I have food, THEN I eat.” Production rules will be used to represent knowledge in this study.

Among other knowledge representation methods are also semantic nets, frames, and logic.

A *semantic net* is a graphical representation technique for propositional information in Artificial Intelligence. Figure 3-2 shows an example semantic net taken from Giarratano and Riley (2005). It consists of nodes representing objects and arcs that represent links or edges. The figure shows relationships between members of a family and can for example be used to infer that David is the grandfather of John.

A *frame* structure consists of slots that denote the item’s attributes, and fillers which are the values for those attributes. A frame can be used to represent an object or a class of objects (Fikes and Kehler, 1985), for instance one frame could represent a student (see Table 3-1) while another frame could denote a class of students (Table 3-2).

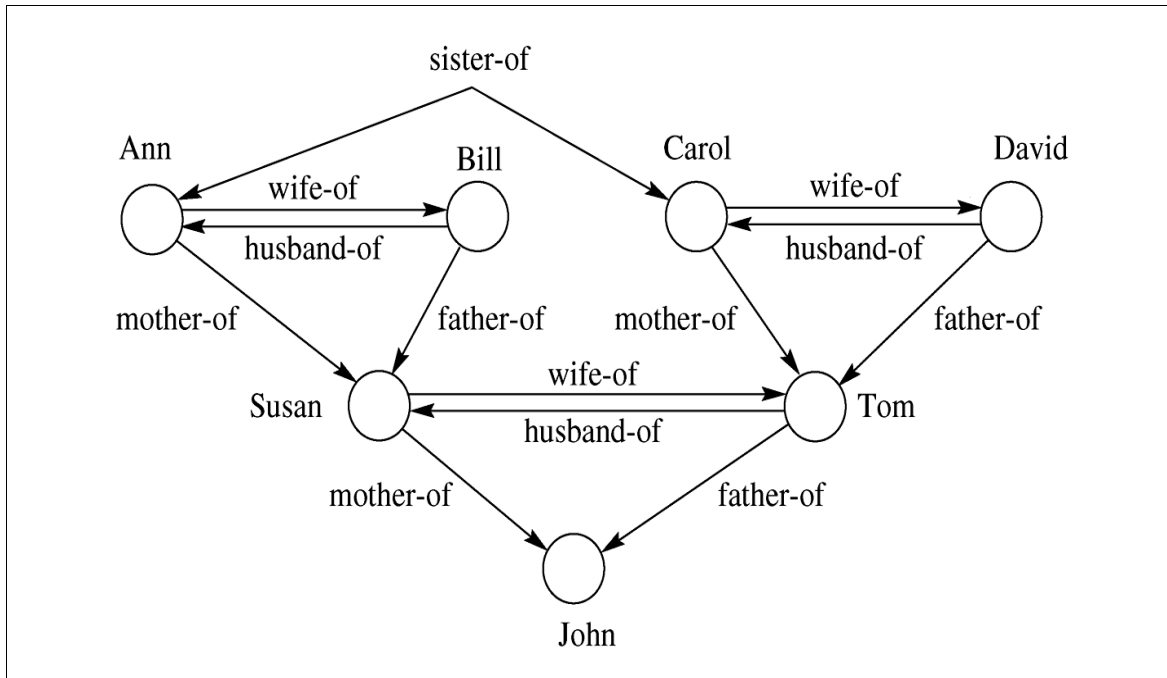


Figure 3-2: Example semantic net

Slot	Filler
Name	Mary
Gender	Female
Length	168cm

Table 3-1: A student frame

Slot	Filler
Class grade	10
Major subjects	English, Mathematics
Members	Mary, John, Daisy

Table 3-2: A class frame

Logic is defined as the study of rules of exact reasoning. A programmer wants to formally construct arguments in such a way that conclusions are accepted as valid (Huth and Ryan, 2004). The *sylogism* is one of the earliest forms of logic and consists of two premises or assertions which if true, infer one conclusion. For instance, assume one premise reads “all girls are sweet” while another reads “Mary is a girl”. The conclusion “Mary is sweet” is inferred through logic. Other forms include propositional logic and first-order predicate logic which will not be discussed in this study.

The next section describes how the function of an expert system correlates to human reasoning and provides more information on the different components encompassed in such a system.

3.4. The components of an expert system

The function of an expert system is to emulate an expert by utilising similar reasoning techniques. The system developer creates several different components for this purpose.

The four main components in an expert system are interlinked to work together when deriving conclusions and include the *inference engine* (section 3.4.1), the *working memory* (section 3.4.2), the *knowledge base* (section 3.4.3), and the *user interface* (section 3.4.4) (Liao, 2005; Giarratano and Riley, 2005). Other optional components include an *explanation facility* which rationalises the reasoning of the system to the user, an *agenda* which represents a list of rules of which the conditions are met when the user enters information into the system, and a *knowledge acquisition facility* which is useful in a self-learning system and allows a user to enter new knowledge automatically.

3.4.1. The inference engine

The reasoning process of an expert system is called *inference* (Ruis-Mezcua, Garcia-Crespo, Lopez-Cuadrado and Gonzalez-Carrasco, 2011) and is done in the same way a human expert makes decisions. Considering the facts received, the system comes to a possible logical conclusion based on prior knowledge. Inference is done by the inference engine which can be described as the brain of the system and is generally performed by means of a problem-solving strategy like *forward chaining* or *backward chaining* strategies.

Reasoning according to the forward chaining method entails considering the known facts and generating a conclusion or solution resulting from those facts. Each fact is evaluated against conditional expressions in the system after which conclusions or new derived facts are established (Lindqvist and Porras, 1999). Forward chaining is the inference method that will be used to develop the expert system in this study.

Backward chaining is reasoning in reverse, i.e. a hypothesis exists that needs to be proven by collecting supporting evidence from the known facts (Lindqvist and Porras, 1999).

Other methods include *means-end analysis*, *backtracking*, and *constraint handling* which will not be discussed in this dissertation.

The inference engine works by iteratively performing three steps (Grzegorz, Bobek, Ligeza and Kaczor, 2011):

- it searches for all of the rules that can be activated by the data or facts received from the user by checking the conditions;
- it tests the relevant rules against the prior knowledge in the system; and
- it determines the candidate rules for execution.

Typically in a production system, many rules can occur that match different subsets of data or conditions and is called a *conflict set*.

To avoid the time-consuming process per iteration of considering all of the different possible rules in a conflict set, an expert system can be designed according to the *Rete algorithm*. The Rete algorithm is a pattern-matching procedure designed to implement a rule-based production system more efficiently (Graciani, Guitiérrez-Naranjo, Péres-Hurtado, Riscos-Núñez and Romero-Jiménez, 2013). The algorithm effectively creates and stores a directed acyclic graph of the rules with the purpose of avoiding checking all of the rules every time new information is added by the user.

When developing a Rete production system a programmer must attempt to create combinations of rules in which the same pattern or condition appears on the left-hand side of a rule and to store some information on the history of execution of the rules. Instead of repeating the three steps for the entire production system on every consecutive cycle, the algorithm should keep record of which rules had previously been activated and look only for changes in pattern-matches. The structure of the Rete algorithm provides the advantage that although the system holds a huge amount of information, it searches through the rules in a very short response time as opposed to standard sequential computer programs that check every rule.

3.4.2. The working memory

The working memory can be described as the dynamic memory that contains the task specific data for the problem currently being solved (Chen, Hsu, Liu and Yang, 2012). The task specific data describes the responses to the prompts that the system presents to the current user and usually changes for every new user (Giarratano and Riley, 2005).

Working memory correlates to human short-term memory which actively but temporarily holds multiple pieces of transitory information in the mind. In similar fashion, the working memory of an expert system holds the information read into the system temporarily while it is in use. Depending on the application field of an expert system, it can also be developed in such a way that some of the dynamic data (like options chosen most by users) is stored as part of the knowledge base or removed (like options rarely chosen).

3.4.3. The knowledge base

The knowledge base is an information repository (Gagne, Andersen and Norford, 2011) that contains:

- the available knowledge that is considered true prior to the development of the system;
- necessary procedures; and
- the production rules used by the inference engine to make conclusions.

A knowledge base is more than a database in the sense that it includes the knowledge relevant to the problem domain translated into rules and strategies that can change depending on the prevalent problem scenario. The domain specific knowledge is acquired through a process referred to as *knowledge engineering* which will be discussed in section 3.5.

The knowledge contained in the knowledge base may consist of expertise gained from experience and knowledge that is available from books or other trusted sources like journals or human experts. Knowledge is stored in a form that is readable by computers so that it can be used in automated *deductive reasoning*. This type of reasoning performs inference by linking conditions with actions or premises with conclusions.

3.4.4. The user interface

The user interface is the platform where communication between the user and the computer occurs. The user interface should have a user-friendly design that is easy to understand and to use by a user with any level of computer competency.

When designing the user interface, a developer should consider the format in which information is to be received and entered from the user. The system must be able to translate the information so that it can be understood and used by the rest of the system (Anonymous, 2008). The user interface is also responsible for relaying the final results or conclusions obtained by the system, to the user.

3.5. Knowledge engineering

The four main components discussed in section 3.4 work in conjunction to perform the basic function of an expert system, as shown in Figure 3-3.

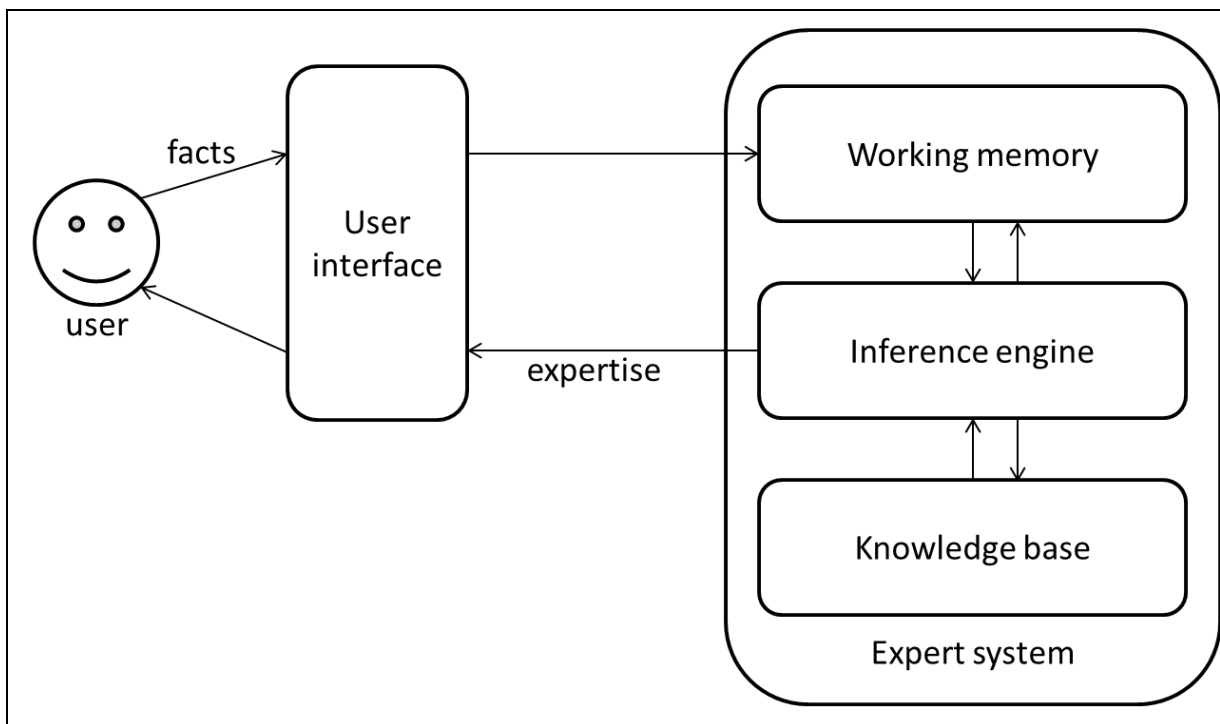


Figure 3-3: The mechanism of an expert system

Knowledge engineering is the process during which a *knowledge engineer* acquires knowledge from a human expert or other source and codes it into the main components of the system (Wagner, Chung, Najdawi, 2003). The knowledge in the system can be represented in different ways like the *knowledge-based approach*

used in this study, where IF-THEN rules express the knowledge of experts, and objects are used to pattern-match against facts.

The development of an expert system is done in several different stages and is considered part of the Knowledge Management (KM) field in Information Technology (IT), which deals with all the knowledge assets available to an organisation. Major companies usually have their own procedures and guidelines to follow regarding the development and maintenance concerning knowledge, especially when intelligent systems can greatly reduce human expenses. Giarratano and Riley (2005) presents a software engineering methodology for creating a quality expert system.

Before building an expert system, the knowledge engineer has to determine the need for such a program by answering the following questions:

1. *Can the problem be solved by means of conventional programming?* As discussed in section 3.4.1, the expertise or conclusions provided by an expert system depends on the input given by the user. The system does not follow a sequential set of instructions for finding a solution to any problem but uses user input to dynamically determine the solution path. If a standard algorithm solution can be developed for the problem at hand, an expert system is unnecessary.
2. *Is the knowledge domain of the problem well-bounded?* The boundaries of an expert system should be very well-defined so that the capabilities of the system are known beforehand. While developing the system, the knowledge engineer can easily add knowledge from related domains to the knowledge base. Although it is believed that the system might be more complete and thorough, this trend expands the complexity and the consequent coordination of the expertise over multiple domains tremendously. Many domains also imply many experts, and experts more often than not, have conflicting opinions when their domains intersect with those of others.
3. *Is there a need for an expert system in the domain and setting?* Creating an expert system when there are no users who are interested in using it is a pointless waste of time. In some settings, experts are readily (and cheaply) available and then nobody would want to consult a computer for answers that a human expert can provide.

4. *Is there at least one human expert who is willing to assist the knowledge engineer?* An expert system is by definition supposed to provide expertise that is usually delivered by a human expert. Part of the knowledge engineering process involves the acquisition of expert knowledge from a human expert. Systems have been developed that utilise knowledge gained from other sources like journals and books but they are based on the interpretation of the knowledge engineer without personal verification by a human expert. The credibility of a system increases when it is created in collaboration with a human expert who can verify that it delivers good quality expertise.
5. *Can the expert explain the knowledge so that the knowledge engineer understands it and can convert it into a system?* Good communication between the knowledge engineer and the expert is needed to determine both if the engineer understands the knowledge correctly and if the expert concurs with the symbolic representation of the knowledge in the computer program.
6. *Is the knowledge used for problem-solving mainly heuristic and uncertain?* Expert systems are ideal when the knowledge of the expert has primarily been gained through experience by means of trial-and-error. Such knowledge can then only be gathered from a person who has made difficult choices and has learnt from mistakes and successes.

Expert systems as problem-solving tools provide the following advantages:

Increased availability: Whereas human experts tend to provide expertise only during certain office-hours, a web-published expert system is available all hours of the day, seven-days-a-week (depending on the trustworthiness of the service provider). An example of such an expert system is the SETH system which was created for the purpose of identifying what drug a person has been intoxicated with (Droy, Darmoni, Massari, Blanc, Moritz and Leroy, 1993).

Reduced cost: The cost of obtaining expertise using an expert system is much lower than that of seeing a human expert who charges by the hour for hard-earned knowledge and experience.

Reduced danger: Expert systems can be used in environments which present danger to humans like earthquakes or floods (Ness, 1997).

Permanence: Unlike human experts who retire, quit or even die, expert systems are available as long as the developer deems it necessary.

Multiple expertise: The knowledge of many experts can be used to gain expertise. Although using many experts when developing an expert system can provide some problems regarding difference in opinions, a developer can expand the knowledge base of the system with knowledge from other credible sources such as published works like books and journal articles.

Increased availability: Expert systems can increase the confidence that the correct decision has been made by providing a second opinion or even a third, when human experts do not agree on some topic. An example of such application area is the medical field where an expert system might aid a person in establishing a second opinion on a diagnosis (Masić, Ridanović and Pandza, 1995).

Explanation: The confidence level of decisions made by expert systems is high because the systems keep track of how the resulting conclusions had been reached and can explain the inference process to the user. Applications such as tutoring programs (Clancey, 1988) provide explanation facilities because the purpose of such programs is to teach.

Fast response: In some applications, fast or real-time responses may be required to determine the next action, for example in an emergency situation where a human expert is not available to perform the necessary actions during power problems (Chen-Chin and Tomsovic, 1986).

Steady, unemotional and complete response at all times: An expert system does not feel emotion like maybe a human expert might when somebody with a personal connection to the expert is involved and might influence the final decision.

Intelligent tutor: An expert system can play the role of an intelligent tutor in a simulation-like capacity, teaching and explaining the steps to reaching a certain conclusion.

Intelligent database: Expert systems can be utilized in data mining, like the ADAM (Audit Data Analysis and Mining) system which is an expert system utilising data

mining for intrusion detection on computer networks (Barbara, Couto, Jajodia, Popyack and Wu, 2001).

Once it has been established that an expert system is the best option for providing the solution to a problem the preliminary planning can start.

3.5.1. Preliminary planning

When developing an expert system, like any other computer program, there are a number of considerations to be taken into account before committing a large number of resources, time and people to a specific project.

3.5.1.1. Selecting the appropriate paradigm

Among the paradigms available for creating an expert system is:

- The *knowledge paradigm* which is used for the representation of large quantities of declarative- or procedural knowledge;
- The *progressive achievement paradigm* which handles unstructured requests of the user; and
- The *intelligent agent paradigm* which is used for data mining and formal knowledge discovery.

The paradigm appropriate to the problem should be selected, based on the type of knowledge which will be used in the system as well as the type of expertise to be given.

3.5.1.2. Payoff

It is important to know what 'return on investment' a system would provide. This is not necessarily measured in monetary units, but also in terms of the advantages such a system would provide, as discussed in section 3.5.

3.5.1.3. Tools

Although there are many expert system development tools available, it is important to use the ones that are appropriate for the problem to be solved. Expert systems, being non-procedural can be created with the use of a variety of available *software tools*. Among those are any object-oriented programming language like Java, Visual Basic, C++, or C#; logic-based languages like PROLOG which has a built-in

backward chaining inference capability; and rule-based tools like OPS5 and CLIPS which is:

“...a multi-paradigm programming language that provides support for rule-based, object-oriented, and procedural programming” (Giarratano and Riley, 2005).

The language to be used for the expert system should be a tool that the developer knows well and should also be appropriate for the specific problem at hand. The basic structure of an expert system and its main difference from procedural algorithms is the interrelationship between rules: if the condition of one rule is true, it triggers another rule and so forth.

During the process of gathering information and knowledge for the planned system, the developer will most probably conduct an interview with an expert in the knowledge domain. It is important to create written *transcripts* of the expert interview for post-referencing and double-checking during later development.

Another tool that can be used is the *Protégé* program (Gennari, Musen, Fergerson, Grosso, Crubézy, Eriksson, Noy and Tu, 2003) which is an ontology editor and knowledge base and is available free of charge from Stanford University. It provides an integrated development environment for use with CLIPS and is ideal for very large systems with thousands of rules.

3.5.1.4. Cost

The cost of building an expert system depends largely on people, resources and the available development time. The total cost of a system is hard to determine because maintenance costs are sometimes impossible to calculate before actually creating the system. Availability of the expert used during the knowledge engineering process, during and after implementation of the system, also has to be considered. Contracting a new expert after a system had been produced, might prove extremely costly because the opinions of different experts could vary considerably on some issues. The ideal would be to create a program that could operate as a standalone system, which could teach itself by means of some criteria based on the history of user inputs.

3.5.2. Expert system development life cycle

The system life cycle, or process model, spans the entire lifetime of the program or software produced, from the concept phase through the implementation phase up until retirement. The process model is in actual fact a *metamethodology* because it determines what stage should be performed next and how long the stage should last. Process models for use in conventional programming include the following:

- The *waterfall model* (Dixit and Saxena, 2011) is presented in Figure 3-4. When using this model, the developer goes through the planning-, designing-, coding-, implementing-, and maintaining phases while revisiting the previous phase from any other phase is always an option.

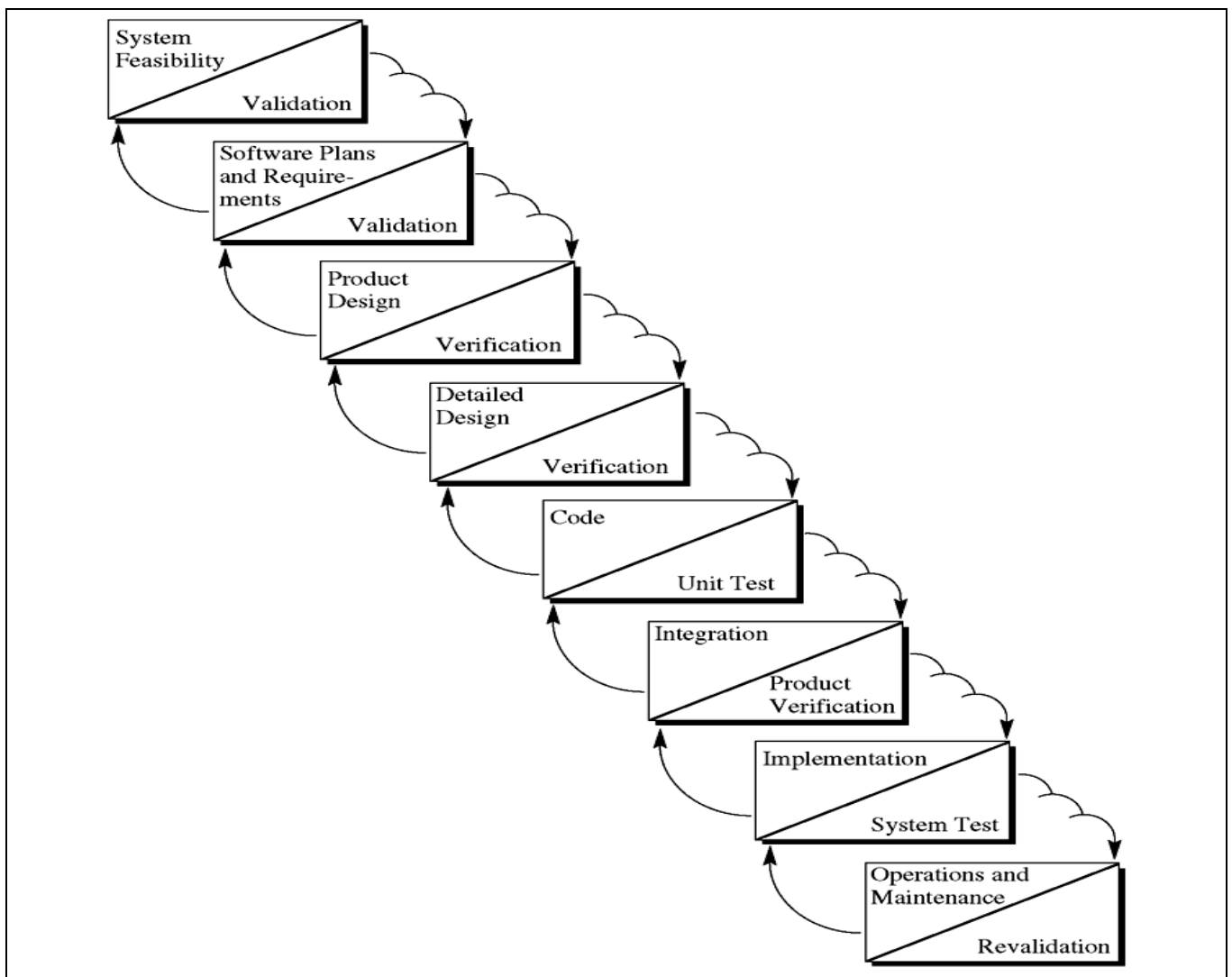


Figure 3-4: The waterfall model

- In the *code-and-fix model* (Giarratano and Riley, 2005), the developer writes some software solution, implements it, and fixes errors that may occur during execution. For obvious reasons, this is the oldest and least popular model.
- The *incremental model* (Dixit and Saxena, 2011) is used when software is produced in increments of functional capability. By the time every section of software is provided, it has already been verified and validated.

Process models used to develop expert systems differ from those for conventional programming and among others, include the *spiral model*, and the *linear model*.

- The *spiral model* (Larman and Basili, 2003) is an adaptation of the incremental model, and is shown in Figure 3-5. During each new circuit of the spiral the same tasks are performed after another functional capability is added to the system.

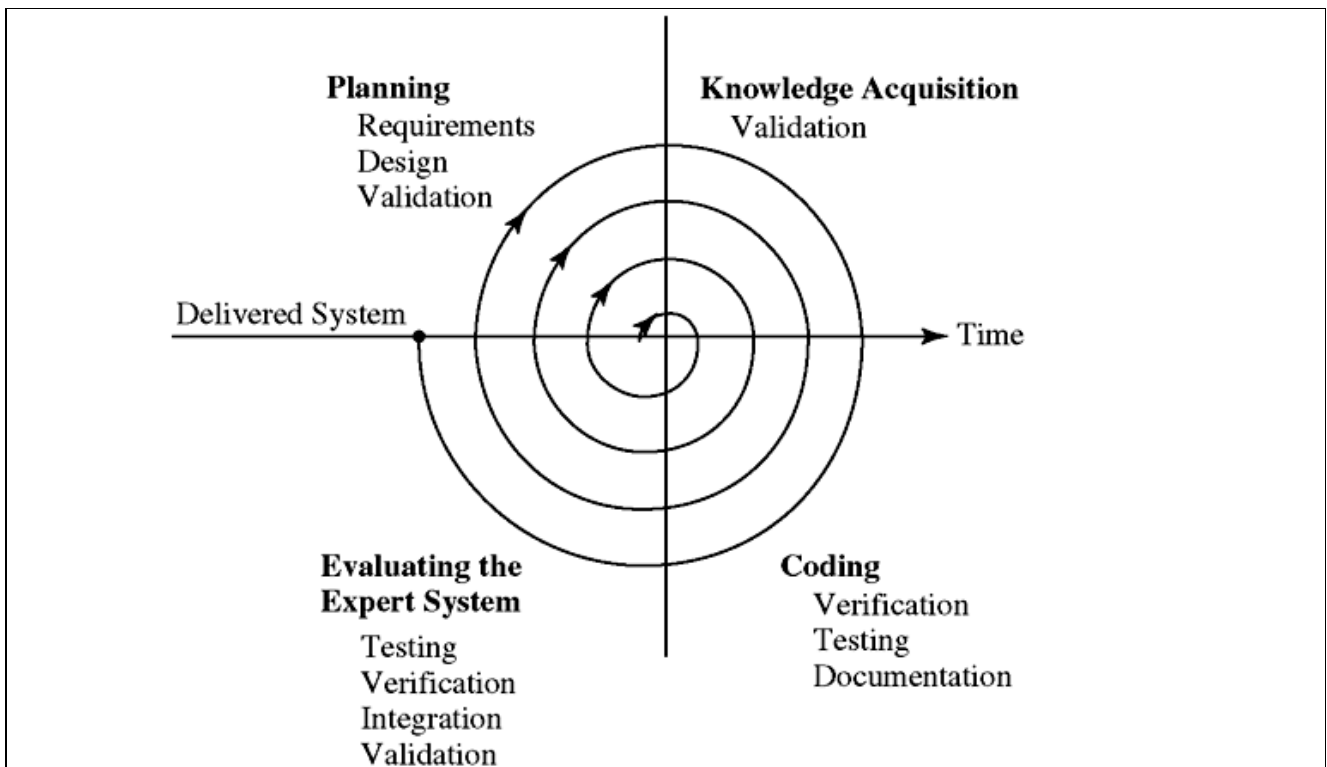


Figure 3-5: The spiral model for developing expert systems (Giarratano and Riley, 2005)

- The *linear model* (Giarratano and Riley, 2005) has been successfully used for development in many expert system projects. The model is shown in Figure 3-6. This model includes every stage, from the planning up to the evaluation stage. Verification and validation proceeds parallel with all stages in the

model. Every stage in the model consists of tasks that may or may not be necessary depending on the type of application field. For example, in a purely search-type system, not all of the tasks in each stage would be necessary and the model is therefore considered a development guideline.

The linear model will be used for development of the expert system in this study. Each of the stages of the linear model consists of various tasks that could be performed depending on the specific application being developed.

The stages and tasks are outlined and explained in the next sections whereas the implementation and completion of the tasks will be discussed in more detail in chapters 5 and 6 which will elaborate on knowledge engineering in detail.

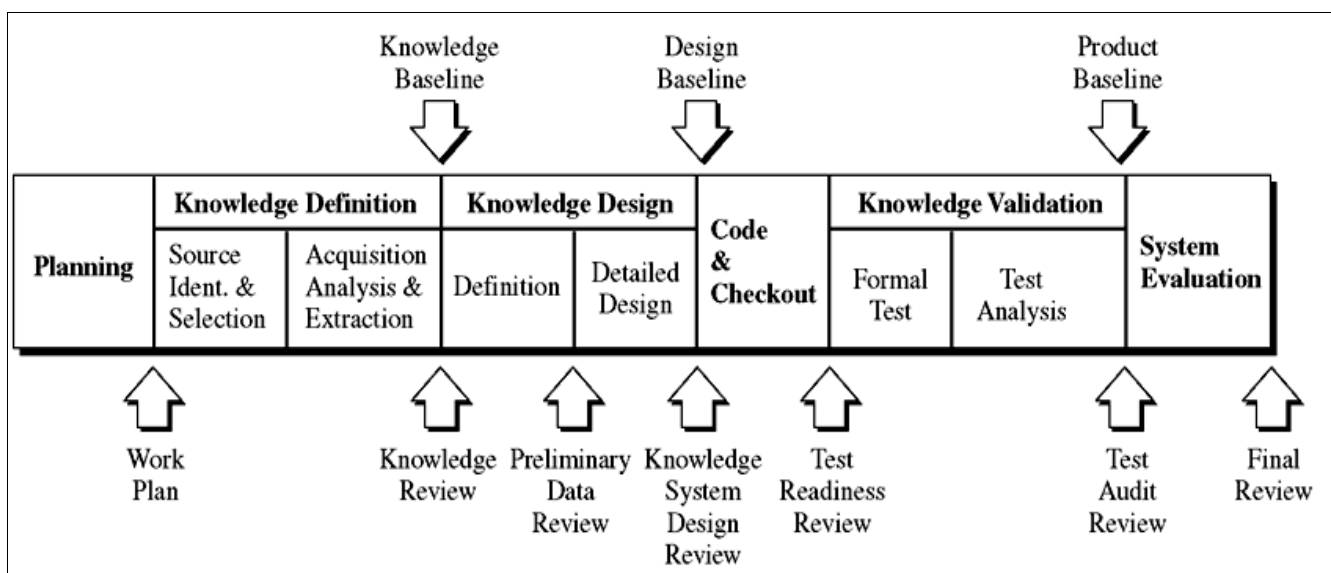


Figure 3-6: Stages in the linear model in expert system development (Giarratano and Riley, 2005)

3.5.2.1. Planning

In the planning stage, the developer creates a proper workplan, consisting of a set of documents that will be used to guide and assess the development process. The tasks to be done during the planning stage include the *feasibility assessment*, *resource management*, *task phasing*, *schedules*, *preliminary functional layout*, and the *high-level requirements*.

In the feasibility assessment task, the developer must determine if it is sensible to build the system as an expert system rather than a conventional program. When this has been established the requirements for the system in terms of people, time,

money and technology are determined and acquired during the resource management task.

Task phasing involves the definition of the tasks and establishing their order in the stages. The starting and delivery dates of the tasks and stages are specified in the schedule task.

The purpose of the system is specified in the preliminary functional layout task by stipulating and defining what the system should accomplish in terms of high-level functions. In the final task in the planning stage, the functions of the system are described in high-level terms.

3.5.2.2. Knowledge definition

The purpose of the knowledge definition stage is to define the knowledge requirements of the expert system. The two main tasks of this stage are the *knowledge source identification and selection* and *knowledge acquisition, analysis and extraction*.

Knowledge source identification and selection is subdivided into smaller tasks but the basic actions to be taken include determining the sources for the knowledge base, prioritizing sources in terms of importance to development, listing the sources in terms of availability and selecting of sources based on importance and availability.

During knowledge acquisition, analysis and extraction, the developer should specify the methods that will be used to acquire the knowledge from the expert or source. The main purpose of this set of tasks is to produce and verify the information for the knowledge base required by the system. Once the baseline has been created, the knowledge should have been verified and must be correct. The knowledge specific to the domain must then be identified, classified and organised so that it can be easily verified later. The functional capabilities of the system should also be specified in more detail and on a more technical level. The developer should distinguish between the general phases that the system will execute so that a basic execution-flow is established. Then the system should be described from the point of view of the user so that the usefulness can be determined. A validation standard should be created that defines exactly what the system is supposed to do and the baseline knowledge of the system should be structured.

3.5.2.3. Knowledge design

The knowledge design stage produces a detailed design for the expert system in the *knowledge classification* and *detailed design* tasks.

Knowledge classification involves defining how the knowledge is to be represented (i.e. rules, frames, etc.), how the system will be initiated, how related groups of rules are to be controlled, and the confluence of the structures. The internal structure for facts should be determined and a preliminary or draft user interface designed. An initial test plan can also be created that contains test data and test drivers so that the system could be tested.

In the detailed design task, the design structure, implementation strategy, detailed user interface, design specifications and report, and the detailed test plan are finalised. The result of this task is the baselined design document which will be used for coding.

3.5.2.4. Code and checkout

During the code and checkout stage, the concrete implementation of the program code starts.

The *coding* task entails the implementation of the programming code of the system. During *testing*, the implemented code is tested using test data, analysis and procedures. The *source code* is produced with explanatory comments and documents. The *user manual* which assists users and experts alike in using the system and *installation/operations guide* which contains information on the correct installation of the program, is created. Finally, the *system description documentation* is created which explains the functionality, limitations and problems of the system.

3.5.2.5. Knowledge verification

Knowledge verification is done to determine the correctness, completeness and consistency of the system and consists of two main tasks. *Formal test procedures* are implemented so that the test results can be documented. *Test analyses* are done on the test results in search of incorrect-, incomplete-, and/or inconsistent answers. A document is then created with recommendations and conclusions based on these findings.

3.5.2.6. System evaluation

The system evaluation stage is necessary to summarize the recommendations so that improvements and corrections can be made. *Results evaluation* is performed by summarizing the results from the testing and verification stages. *Recommendations* are made that could effect changes in the system after which the system is *validated* in terms of user requirements and needs. The *interim or final report* is published if the system is complete, otherwise an interim report should be created that describes the increased functionality of the latest update of the system.

3.6. Chapter summary

The purpose of this chapter was to explain the basic concepts, components and the phases and tasks of the design principles of expert system technology. Definitions were provided and the inference engine, working memory, knowledge base, and user interface which form the main components of an expert system were explained and discussed. Some of the process models for development were briefly described, and the linear model for expert system development was discussed in more detail

Some of these principles were employed for the system developed in this study and will be referred to when the development of the actual system is explained in chapter 5.

Chapter 4 will sketch the need for an expert system in dietetics, particularly in South Africa. Some existing computer software solutions for the diet problem are also briefly outlined.

Chapter 4 - The diet problem and South Africa

4.1. Introduction

The primary objective of this study is to determine the feasibility of combining expert system principles with the optimizing advantages of linear programming techniques. The application field for this study is dietetics with the specific knowledge domain of adolescents in South Africa.

For the purpose of providing sufficient background on the diet problem and how linear programming as well as expert systems have been used to provide solutions for this problem, this chapter presents an introductory overview of the origin of the diet problem in section 4.2 and some recent developments in linear programming techniques employed to solve the diet problem in section 4.3.

Some linear programming and expert system examples that have been developed to solve specific diet- and health-related issues are discussed in section 4.4.

Section 4.5 provides some background on how the diet problem relates to South Africa.

4.2. History of the diet problem

The diet problem originated in the mid-twentieth century when a scientist by the name of George Stigler (1945) presented a method for developing a minimum cost diet for an “active economist”. Stigler’s process involved the compilation of a list of foods that could be included in an adequate diet and the systematic reduction of food selections. The reduction process was done by eliminating items by comparing the nutritive values of each with the other foods in the list. Stigler concluded that the model lacked practicality because the resulting diet was not always palatable, especially for people from different cultures.

The original model had the capacity to provide quite impractical results when small changes were made to some of the influential factors like palatability and nutritive values. A student of Stigler’s work, George Dantzig (Dantzig, 1990; Dantzig, 1963), started during 1947 to apply the model to the real-life situation of creating a weight-loss diet for himself. Unfortunately, the resulting diet always left him feeling hungry. He consequently changed the objective of the diet to maximise the sensation of

“feeling full”. The result was a diet that contained some conceivable items and others in totally ridiculous amounts, for example 500 gallons (1892.7ℓ) of vinegar per day. For obvious reasons, he introduced the use of constraints to restrict the number of times one item could appear in the diet.

Dantzig was the creator of the well-known simplex method (Nash, 2000; Dantzig, 1963) which was discussed in section 2.2.2.2 in chapter 2. Solutions provided by the simplex method were considered much more realistic and practical at the time. Dantzig later modified the model to also limit food items with high glucose index values from being repeated too many times in the resulting eating-plan.

The diet problem generally refers to the optimisation of an objective function that is defined in terms of the contribution of food items to the goal. For example, a least cost diet entails the minimising of the total cost of a diet, where the objective function is defined in terms of the cost contribution of each food item considered. The formal definition was discussed in chapter 2.

Over time, new developments in linear programming have been used frequently to solve various adaptations of the diet problem as well as in many other application-fields. Some of the developments related to the diet problem and health-care are discussed in the next section.

4.3. Recent developments in linear programming diet applications

Since the origin of the diet problem and the consequent simplex method, researchers have applied the method to many variations of the original diet problem. Section 4.3 aims to provide a development timeline of the use of linear programming techniques in dietetics by means of some examples.

The World Health Organisation frequently selects and distributes diets for Third World Countries in an attempt to reduce global hunger. However, these diets sometimes result in an oversupply of some of the micronutrients that the human body needs and are therefore not optimal (Anderson and Earle, 1983). A method was developed that uses linear programming and goal programming techniques to minimize the shortfall and excess of the micronutrients provided by the selected diets. Goal programming (section 2.4, chapter 2) is an extension to linear programming and is used when a problem has two or more objective functions that

conflict with each other. Within the diet problem application, the objective function (least cost) is replaced by a set of objectives representing the deviation in nutrients from the levels required for a healthy body. Goal programming was discussed in more detail in chapter 2.

For the purpose of eliminating the repetition of food selections within a diet, mixed-integer linear programming was employed in 1993 (Sklan and Dariel, 1993). The motivation behind the work was to limit the number of times a certain food item should appear in a diet. The problem was solved using an adaptation of the simplex method (see chapter 2) and applying additional integer and bound restrictions. The method proved to be a repetitive, iterative method that created different solutions for the problem and determining which of the solutions was the optimal one. The work provides a typical example of the branch-and-bound method (chapter 2) in practice. The nutrient requirement for this solution was used as per the Recommended Daily Allowances put forth by the Food and Nutrition Board in 1989. This process was very time consuming but could handle 800 equations with about 4000 variables.

Garille and Gass (2001) discussed the use, significance, and evolution of the diet problem since the work of Stigler (1945). Using linear programming, diets have been developed for specific cases involving certain regions of the world, different objectives and combinations of objectives. Diets have been developed for people who need to restrict the intake of certain food-stuffs like fat and carbohydrates. Stigler's original solution to the diet problem was modified to include updated nutritional data from 2001, as well as added constraints that restricted the intake of food items believed to have harmful effects on people with high cholesterol levels. The change in dietary habits for the purpose of treating certain health-related issues is discussed in chapter 5 as part of knowledge engineering.

In 2004, work was done to set food based dietary guidelines using linear programming techniques (Ferguson, Darmon, Briend and Premachandra, 2004). A four-stage solution was created and implemented in Malawi. The solution entailed the use of a goal programming model to optimise the energy contributed to the diet by certain foods as well as considering the food products most consumed in the region. The resulting dietary guidelines were then implemented and evaluated for minimal nutrient adequacy. The process was expanded and used to develop seasonal dietary guidelines, taking into consideration food availability. The study

successfully accomplished the development of a nutritionally balanced diet for a specific region that included food types frequently consumed in that area.

Hamzah, Shuib, Tumin, Shamsudin and Yusof (2011) developed a two stage linear program and used it to successfully create a nutritionally balanced diet at the lowest cost for Malaysian households. The first stage entailed the use of linear programs to first develop a healthy diet for each of the separate family members by taking into consideration their different ages. The second stage of the system optimised the individual diets acquired in stage 1 into one eating-plan for the entire household. Instead of creating a new eating-plan, the system evaluated the resulting food lists consisting of typical Malaysian food items for a family by comparing it to existing menus.

Although the diet problem originated after questions were raised regarding eating habits of humans, a lot of linear programming work has also been done in the field of developing balanced diets for animals. Although such studies were applied in different fields, they also contributed to developments in linear programming.

Moraes, Wilen, Robinson and Fadel (2012) used linear programming to prepare diet mixes for dairy cattle when environmental policies are in place. Their work entailed the evaluation of nitrogen and mineral excretions by the cattle as well as methane emissions, with the aim to minimise these emissions so that the restricting policies in place could be satisfied. A tax constraint on the gas emissions was added to the linear programming model. Although people are not taxed for gas emissions, a tax constraint can be applied to a human diet for categories like allergies. If an individual is allergic to a certain food type, it should be excluded from the diet. Allergies and the application thereof are discussed further in in chapters 5 and 6.

Another least cost application of the diet problem in an animal setting was performed by Aldeseit, Majdalawi and Ata (2012), specific to Jordan in Western Asia. The model used in this study was designed to create least cost fodder ration combinations with the objective of fattening calves as much as possible. Their work entailed having to consider the feeding of calves at different ages, the fodder available in the country as well as the subsequent evaluation process of the actual growth in each age group. A linear programming model was constructed and then executed repeatedly for the different stages of the ages of the cattle.

Linear programming techniques have also been used to determine the effect of applying a cost constraint to existing diets that are considered nutritionally adequate (Darmin, Ferguson and Briend, 2006). The work showed that the energy provided by fresh foodstuffs proved more expensive because as soon as the cost constraint was applied to the model, more energy was obtained from preserved foods than from fresh. However, the study was conducted in France indicating that the prices of food items used were relevant only in France. If one was to apply the existing system to analyse diets in other parts of the world, the food database would have to be updated.

Iterative systems are rather time consuming but when working with an enormous amount of variables like one variable for each food type in one objective function, iterative systems seem plausible. Some systems have been developed with success in France to optimise nutrient density rather than cost, while also considering individuals' food choices (Maillot, Vieux, Amiot and Darmon, 2010).

The next section discusses concrete implementations of specific mathematical programming systems relating to the diet problem.

4.4. Computerized solutions to the diet problem

The nature of this study involves the consideration of both linear programming techniques and expert systems. Therefore, applications of both of these solution methods in problems related to diets or nutrition, are reviewed.

4.4.1. The implementation of linear programming techniques to solve diet related problems

The advances in microcomputer technology triggered the development of many user-friendly software solutions for almost any existing problem, the diet problem included. Bassham, Fletcher and Stanton (1984) developed a computer program in the form of a software package with the goal of assisting dieticians working in hospitals to analyse the diets of hospital patients. The program was implemented in the United Kingdom (UK) and was reported to have reduced the calculation time associated with creating a diet considerably, and improved the quality of service the dieticians could provide. However, the authors concluded that the program had difficulty in handling the large amounts of data available in the UK and that there was

a lack of appreciation among dieticians regarding the usefulness of computer technology in their professional field. Regardless, many software applications utilising mathematical programming techniques to solve different variations of the diet problem have been emerging ever since.

In 1991 a computer system was developed that used linear programming techniques to modify an existing diet so that it conforms to the nutritional constraints for a certain user (Soden and Fletcher, 1992). The program would ask the user to set up a list of food items that are usually consumed in one day. Linear programming techniques were used to alter the quantities of the listed foods so that the resulting diet provided the required daily nutrients. The user then had the option to exclude or add certain food items to the diet after which the program again modified the quantities to conform to the nutritional constraints. This program was concluded to be a useful tool to use in teaching a dietician how to make changes in food choices and how to alter the quantities of those foods.

In 2004, Cadenas, Pelta, Pelta and Verdegay published a paper on a computer system named SACRA which was designed to assist in developing a more accurate diet for livestock in Argentina (Cadenas *et al.*, 2004). The authors argued that it was quite impossible to accurately measure the amount of food each animal in a herd ate on any given day. The amount of available feed-mix was also uncertain, as the farmers assisting in the study could not give a definite weight value of what they had stored. Instead of developing a diet for the animals based on exact daily nutritional requirements, the program allowed for constraint violations utilizing fuzzy linear programming for the nutrient requirements as well as the available feed-mix. The system was said to have provided a high level of satisfaction to the users because of the slack it allows on the uncertain variables.

In Istanbul, scientists have developed a computer system that requires input from a user regarding food preferences and then minimises the cost of a diet that includes most of the chosen preferences (Kahraman and Seven, 2005). The developed system used an iterative process to obtain multiple solutions and evaluated them by doing a comparison between each new solution and the last best one according to the branch-and-bound method.

Mino and Kobayashi (2009) developed a system that uses a linear programming model to create a customized diet for an individual in Japan. In their model they considered typical nutrient intake guidelines taken from an Internet source and grouped them according to protein, carbohydrates, lipids (fat) and salt. During pre-processing, they organised the menu items into groups that contained meat, fish and vegetables, and rice or noodles as the main dish. The objective of the model was to create a recipe for one meal of the day that would assist a typical user in losing weight. The system required the user to enter the food elements usually consumed, as well as the types of events (like parties, lunch, sports, etc.) in which the user typically participates. The system incorporated all of the required information gathered from the user into the program and provided a recipe recommendation that was optimal in terms of the user's typical food choices, as well as an essential nutrient distribution. They concluded that their system could affect an average weight-loss of 1.0 kg in about ten days' time.

Frega, Lanfranco, De Greve, Bernardini, Geniez, Grede, Bloem and De Pee (2012) applied a linear programming tool to develop cost-effective ways to provide enriched supplementation in the Gaza province in Mozambique. The program was used to evaluate the average dietary needs in a typical Mozambican household, to determine and create a database of the food items available in specific seasons, and to create a least cost diet based on the findings. Upon evaluation of the system it was found that among the essential nutrients, iron and vitamin B2 tend to increase the overall price of the diet so that not any random family could afford it. Regardless, the system yielded a feasible diet that met all the dietary constraints and nutritional requirements.

Hornick, Krester and Nicklas. (2008) studied the possible improvements to be done on the quality of a diet by means of the MyPyramid food guidance system. The MyPyramid system was developed by the United States Department of Agriculture to suggest recommendations on the dietary nutrient intakes of males and females of various ages and activity levels. Hornick *et al.* used the system in a study to determine the feasibility of gradually modifying the existing diets of a certain focus group by implementing small changes and evaluating the quality of the resulting diets. The focus group used was American women between the ages of 31 and 50 years. They used data collected and published by the National Health and Nutritional

Examination Survey done in 1999-2000 and established an estimation on the lack of intake of certain nutrients. It was discovered that among others, American women do not consume enough fruit, vegetables and milk; and that 20-40% of all the women in the group generally skipped breakfast. The MyPyramid model was used to develop a seven-day diet that started on the first day with a menu that closely resembles the average diet of the focus group. Small changes were added to the menus over the course of the seven days with the aim of improving the quality of the resulting diet. They incorporated the gradual addition of breakfast as well as an average decrease of about 11% in total energy. Their menu was successfully implemented and they found that the quality of the new diet showed great improvements in the intake of the essential nutrients. They resolved that the system was subject to the accuracy of the information provided by the focus group and stated that those resources were limited. Their work did not consider any cost implications that the changes in the diet might have and they also concluded that the small changes made to a diet might in actual fact prove to be rather unpractical for some individuals in the focus group. Cultural influences upon eating patterns were also not considered in this study.

An interesting twist to the diet problem can be seen in the work of Mamat, Rokhayati, Noor and Mohd (2011) who used linear programming to develop a least cost diet subject to fluctuating food prices. In their model, they defined the cost of each item in terms of fuzzy numbers. A fuzzy number is formally defined as:

“...a fuzzy subset of the real line whose highest membership values are clustered around a given real number called the mean value; the membership value is monotonic on both sides of this mean value.” (Dubois and Prade, 1978)

The 2011 model considered maximum and minimum price levels for each food item and used it to successfully develop low-cost-, low-carbohydrate- and low-fat diets respectively.

4.4.2. Expert systems employed to solve diet and health-related problems

Many expert systems have been developed and created to assist in some way in solving complex problems that require human expertise to solve them. Among those

are systems that plan diets or menus as well as systems intended for use in some health application.

In 1996, four different menu planning expert systems were constructed, studied and compared with the purpose of determining the importance of common sense knowledge in the construction of a final meal plan (Sterling, Petot, Marling, Kovacic and Ernst, 1996). The resulting program, PRISM 3.0, was believed to have been developed to such an extent that the resulting meal plans generated by the system were considered reasonable. The work emphasises the importance of considering context-sensitive relationships between menu parts which the PRISM 3.0 system incorporated in the form of layered networks.

An interesting application of the expert system technology can also be seen in the design of an agent designed to detect false and misleading claims on the internet for weight-loss products in Korea (Sing and Chang, 2005). The knowledge base of the system contained a set of text-searching rules created in such a way that certain phrases could be identified, for example if a phrase contained the words “*guarantee*” and “*target weight*” the expression could be regarded as a potential false claim. The system boasted a knowledge management system which collected keywords for examining false and misleading claims and a keyword modification function which updated keywords in accordance to the changing perspectives of experts on the subject. The approach provides a classic example of a system with *self-learning* capabilities.

The Chinese Medical Diagnostic System (CMDS) is an expert system that was developed by Huang and Chen (2007) for diagnosing a number of digestive diseases. For the CMDS system, the required components included:

- The user interface which consulted with the online physicians’ information system (a system that provides information on the digestive system and related issues);
- The inference engine implemented in the Java based expert system shell, JESS, and which communicated with the explanation system as well as the knowledge descriptor;
- The database set which consisted of the general database with knowledge about digestive diseases and the knowledge base with the rules needed to perform inference; and

- The knowledge extractor which incorporated a web-based knowledge extractor and an object-oriented knowledge acquisition editor.

The system was implemented successfully and could provide expert diagnoses on a limited number of diseases and was also considered a good educational tool. The work provides a good example of an expert system consisting of non-conventional components which are essential in the specific application for which it had been created.

Expert system technology is sometimes combined with other solution techniques and methods. Arsene, Dumitrache and Mihiu (2011) proposed a prototype system which was Ontology and Bayesian Network based and used for diagnosing a restricted number of medical conditions. The layers in the framework included the knowledge layer, the uncertainty model, and the user interface. The resulting system was a Java-based application which could be installed on different platforms and devices. The phases performed by the system to make the necessary conclusions, included knowledge acquisition, uncertainty assignment, and evidence assignment and beliefs update. It was indicated that the system be used as a prototype framework for developing or expanding existing and future expert systems.

In 2003, the American Dietetic Association implemented the Nutritional Care Process and Model (NCPM) which aims to examine and improve the nutrition quality of relevant regions and countries like Taiwan (Chen *et al.*, 2012). In 2012, a rule-based expert system was constructed that would make a nutritional diagnosis, which is one of the steps in NCPM. Nutrition diagnosis is done by doing certain tests and evaluations on a person and providing a diagnosis on any nutritional problems that were identified. The system that was constructed created an account for each subject (or patient) under consideration as well as a nutritional profile by documenting laboratory test results, anthropometric measurements, physical exam findings, and subject nutrition history. Inference was performed on the profile so that conditions such as underweight, overweight, numerous levels of obesity, kwashiorkor¹, marasmus², and several other could be identified by the system. Results obtained from such an expert system could assist dietitians with the other

¹ Kwashiorkor is a severe form of malnutrition which, due to an extreme lack of protein, causes swelling of the gut.

² Marasmus is a form of severe malnutrition characterized by energy deficiency and extensive tissue and muscle wasting, and drastic loss of body fat.

steps in NCPM which included nutrition- assessment, intervention, monitoring and evaluation.

A recommender system is one that considers a multitude of sources providing information on a person or user, and recommends products the person might benefit from based on that profile (López-Nores, Blaco-Fernández and Pazos-Arias, 2010). In 2012 (López-Nores, Blanco-Fernández, Pazos-Arias and Gil-Solla, 2012), a health-aware recommender system was developed to deliver relevant information based on sources like television watching records (Pazos-Arias, López-Nores, Garcia-Duque, Diaz-Redondo, Blanco-Fernández, Ramos-Cabrer, Gil-Solla and Fernández-Vilas, 2008), web-browsing history, and also electronic health records. The types of recommendations such systems make include advertising non-prescription medication to people who might benefit from them. The health-aware recommender system uses property-based collaborative filtering which includes the capability to avoid offering products to people who are not suitable for using them. User profiles were represented by internal matrix structures which allowed storage for additional user attributes like demographic features, etc. Effectively, the data input into the system was used to create a user profile which, following a property-based filtering process, allowed the system to make certain recommendations on products the users might benefit from.

Within the field of artificial intelligence, other methods exist that provide support in making decisions based on external information. Bourguet, Thomopoulos, Mugnier and Abécassis (2013) presented an argumentative reasoning system which comprehends situations where information is unintelligible because of the difference in opinions of information sources as well as differing data acquisition techniques from those sources. The system was developed to assist in analysing food quality in a public health system in France. The aim of the system was to provide decision support by constructing and analysing arguments regarding health based on relevant input data. The work merits mentioning because of the similarities between the acquisition techniques and those in knowledge engineering for expert systems. Knowledge was obtained from different sources like peer-reviewed articles, technical and web-published reports, scientific conferences and project meetings, as well as human experts. The steps that the system was expected to perform included the input of all relevant arguments, definition of the audiences to prioritize system goals,

generation of preference relations between the arguments for each audience, definition of the defeat relations between arguments and computing decisional resolution so that recommendations could be made. The system was used for assistance in decision making by formulating the arguments needed for consideration.

Expert systems are used frequently in a wide variety of applications within the health industry. Some of the expert systems that have been developed include:

- A medical tourism expert system called the ubiquitous Context-aware Healthcare Service System (UCHS) which used micro sensors to create a health profile for users and recommend natural medicine for the users (Lo, Chen, Cheng and Kung, 2011);
- Wang and Su (2012) created an expert system that used a biomedical measurement technique and considered factors like user movement and rotation speed of the exercise equipment among others, to diagnose the health status of the user;
- The knowledge-based system CASESIAN was developed to facilitate the sharing of clinical insights and practice experience in the healthcare industry for the purpose of improving the process of issuing prescriptions (Ting, Kwok, Tsang and Lee, 2010);
- An empirical study was published in 1995 regarding the development of a drug prescription system by the name of OPADE which had been developed for improving the prescribing behaviour of physicians (Berry, Gillie and Banbury, 1995);
- Expert system technology was used in the expansion of a support vector machine (SCM) which is used in classification tasks for the diagnosis of hypertension (Su and Yang, 2008); and
- Youm, Lee, Park and Zhu (2011) developed an expert system which a person could use to do a self-check-up and monitor the health status of his/her lifestyle.

Linear programming techniques and expert systems have been used for solving a variety of different diet- and health-related problems in various environments. The purpose of the next section is to highlight how the diet problem relates to South Africa.

4.5. Health in South Africa

It seems logical that a linear programming system that aims to develop a healthy eating-plan for an individual should consider food items that are available in the specific region as well as their prices within that region. The application area of this study is dietetics in South Africa, targeting female teenagers specifically. However before food prices are examined, insight into the health issues among teenagers in South Africa needs to be discussed. The purpose of the next section is to provide background and discuss research relevant to the eating habits and related errors common among teenagers in South Africa.

It has been found that eating habits of young children are often maintained when the children grow up (Nicklas, 1995). According to research done over a five-year period on the eating habits of adolescents in urban areas in South Africa, poor eating habits are common among teenagers for various reasons (Feeley, Musenge, Pettifor and Norris, 2012). These poor eating habits are also shown to be well established by the time children reach the age of thirteen. Children are frequently exposed to energy-dense fast foods which are low in essential nutrients; subsequently the rate of fast food consumption also increases with age. Teenagers, and especially females, are increasingly skipping breakfast which is generally considered to be the most important meal of the day. Girls in particular are also shown to eat more confectionary (sweets) than boys. Among the items most frequently bought at school tuck-shops are sweets, crisps, cold drinks, fried chips and white bread. Feeley *et al.* (2012) quite alarmingly found that fast food consumption increased by 10.4% over the duration of the five year study, and also that girls eat more chocolates than boys. The “snacking” phenomenon, which entails the consumption of food items between main meals, also increased and can be harmful to a person depending on the types of snacks and the quantity consumed. The increase in fast food consumption can be ascribed to the inexpensive prices at which fast foods can be acquired as well as the comfort of not having to prepare a meal. An increase in fast food consumption effectively means a decrease in the consumption of healthy food and a lack of essential nutrients in the diet.

Fast food consumption is reported to be on the increase in Western civilizations and especially among teenagers (Jaworowska, Blackham, Davies and Stevenson, 2013). A typical fast food meal is said to contain twice the amount of energy of that of a

healthy meal but people do not necessarily eat smaller portions when consuming the former. Fast food is also more energy dense than healthy food which means that it takes the human body longer to metabolise the food consumed. Food that has not been digested by the next mealtime is usually stored as auxiliary energy in fat cells in the body. It is alarming to note that food supply high in fat, protein and calories is increasing rapidly in South Africa (Igumbor, Sanders, Puoane, Tsolekile and Schwarz, 2012). It is a common occurrence that fast foods are usually bought in conjunction with soft drinks, which is alarming because it has been shown to be directly related to obesity in children and adolescents (Steyn, Labadarios and Nel, 2011). Obesity is discussed in more detail in the next section. Schönfeldt, Gibson and Vermeulen (2010) reported that most South Africans do not have the financial means to afford a balanced diet but consume mostly staple food starches and little to no fruits and vegetables. The main reasons for this are assumed to be a lack of education as well as the state of poverty in the country. With the ever increasing inflation causing food prices to rise even further, it is expected that SA households will in future omit even more healthy food choices from their diets, as the majority of South Africans' diet show very little variety (Labadarios, Steyn and Nel, 2011).

A study on the food intake in South Africa shows that poor hygiene habits as well as insufficient education regarding nutrition are causing certain conditions like stunting and undernutrition in young children (Steyn, Labadarios, Maunder, Nel and Lombard, 2005). Stunting describes a reduced growth rate in a person usually due to malnutrition during childhood. In South Africa in particular, it has been observed that although there is a high prevalence of stunting among black children in rural areas, obesity is common among black adult women. This is related to the contradictorily incidence of undernourished children living in the same community as teenagers who frequently consume inexpensive fast food products. A stunted child is at risk of being overweight or even obese later in life which in turn could lead to other related chronic diseases like diabetes or hypertension. Chronic diseases and dietary intervention for treating these diseases are discussed in chapter 5 as part of knowledge engineering. It is widely advocated that caregivers should be sufficiently educated on the importance of propagating healthy eating habits among children and teenagers in South Africa.

4.5.1. Obesity

Dietary intervention for the sake of obesity prevention during adolescence is becoming increasingly important, especially considering the harmful effects obesity can have.

Van der Merwe (2012) relates that a 9-year-old boy weighing somewhere between 105 and 120 kg, had died due to secondary symptoms caused by obesity. Obesity describes the unbalanced state of consuming more energy than the body can spend, causing excess energy to be stored as fat in the body (Goedecke, Jennings and Lambert, 2006). A person is classified as obese when that person's body mass index (BMI) exceeds 30 kg/m². The BMI represents the relationship between a person's mass and height according to the following equation:

$$BMI = \frac{mass(kg)}{(height(m))^2} \quad (4-1)$$

The World Health Organisation (2006) uses BMI to classify a person as underweight (BMI < 18.5), overweight (25 ≤ BMI ≤ 30) or obese (BMI > 30).

According to statistics published by the World Health Organisation (WHO) the number of obesity and overweight cases in Africa has doubled from 1990 to 2010 (Rossouw, Grant and Viljoen, 2012) with a higher occurrence among children and adolescents. It maintained that the prevalence of overweight and obesity up until late adolescence was especially high for the previously disadvantaged groups in South Africa and particularly among girls.

Results from the 1998 census showed that among black South African women, 27% were classified overweight and 32% obese (Puoane *et al.*, 2005). Later work showed that in 2011, an average of 73% of all women in South Africa have a BMI higher than 25 kg/m² (Steyn *et al.*, 2011). Black women generally have the opinion that “big is beautiful” but do not always realize the health implications this perspective may have. In some cultures, girls actually believe that being overweight is an indication of wealth and happiness and even a sign that they do not have HIV or AIDS³ (Rossouw *et al.*, 2012). South Africa has the two-sided problem of overweight that

³ Human immunodeficiency virus (HIV) is a virus that causes acquired immunodeficiency syndrome (AIDS) which causes progressive immune system failure in humans (Sepkowitz, 2001)

contributes to diseases being caused by undernutrition (a diet where certain essential nutrients are lacking) and communicable diseases. The occurrence of overnutrition⁴ is predominant in South Africa but there is a higher occurrence of obesity in urban areas (Puoane, Steyn, Bradshaw, Laubscher, Fourie, Lambert and Mabananga, 2002) than in rural areas.

A study that examined the body shape satisfaction among black female students showed that the students supposedly do not feel obligated to be thin and that they view obesity as a normal state of health (Senekal, Steyn, Mashego and Nel, 2001). It was proposed that suitable intervention programs be initiated to prevent eating disorders among black female students in South Africa after a study conducted in the University of the North in the Northern Province showed 18.2% of black students to be overweight and 6.5% obese (Steyn, Senekal, Brits, Alberts, Mashego and Nel, 2000). It would suggest that either black females become obese after acquiring a tertiary education, or that those who are students are better educated regarding healthy eating habits.

A ten year longitudinal study has shown that eating disorders tend to start early in life and continue into adulthood (Neumark-Sztainer, Wall, Larson, Eisenberg and Loth, 2011), suggesting that intervention before a female reaches adulthood would be more successful than one that is implemented later in life.

The number of black schoolgirls between the ages of 15-18 that are dissatisfied with their body shape is less than the number of white or coloured girls (Caradas, Lambert and Charlton, 2001). The ideal body shape of black girls is also considerably larger than those of the other groups, meaning that by the time they reach adolescence they are already starting to change their views on what a healthy body should look like. However, the prevalence of abnormal eating attitudes among South African girls aged only 11 is about 1% (Peterson, Norris and Pettifor, 2006). Educating girls, as early as in their adolescent years, about healthy eating habits could help to prevent obesity later in life.

Some recommendations on how to prevent obesity and other related disorders include the discouragement of unhealthy dieting and to address teenagers about

⁴ A specific type of malnutrition that involves consuming an oversupply of nutrients required for a healthy body

weight maltreatment (Neumark-Sztainer, 2009). Adolescents should be educated on the effects of combining healthy eating with physical activity. In an effort to control weight, they are increasingly using unhealthy weight control behaviours like skipping meals, excessive dietary restrictions, and the use of diet pills to name only a few (Neumark-Sztainer, Wall, Story and Hannan, 2006). Obesity is said to be linked to infertility and can in some circumstances cause diabetes or even hypertension (Ogbuji, 2010). A diabetes sufferer is typically somebody with high blood sugar due to insulin deficiency. Hypertension is also called high blood pressure and it increases the strain on the heart to circulate blood through the body.

Although a person is classified obese only when the BMI ≥ 30 kg/m², it has been found that the possibility of premature death among black women from any health-related cause increases when the individual has a BMI of more than 25 kg/m² (Bogg, Rosenberg, Cozier, Wise, Coogan, Ruiz-Narvaes and Palmer, 2011).

4.5.2. Dieting and abnormal weight control behaviour

Due to media exposure, peer-pressure and other social factors, young people employ different and often unhealthy methods to try to lose weight (Neumark-Sztainer, Wall, Story and Standish, 2012). An interesting observation is that dieting or unhealthy weight control behaviour during adolescence causes an average increase in BMI over a ten-year period. On a short-term basis, dieting and abnormal weight control behaviour can also cause excessive weight-loss and eating disorders. Research was proposed to intervene and not only provide adolescents with adequate information on healthy living, but also educate them on the difference between eating healthy and dieting.

Another reason for the prevalence of eating disorders is disturbed eating attitudes (Fryer, Waller and Kroese, 1997). These disturbed eating attitudes usually materialize during adolescence and are more common among females than males. Teenagers sometimes handle stress through eating behaviour which can cause bulimia and/or anorexia both of which can lead to death if not treated. Bulimia nervosa is characterized by uncontrolled binge-eating followed by a deliberate attempt to rid oneself of the food that has been consumed. Anorexia nervosa refers to an individual's extreme food restriction usually caused by an illogical fear of gaining weight.

In South Africa many social and economic changes have taken place during the recent past. Black communities have access to food types that were not always traditionally consumed. A lower dependency on plant foods had the effect of a lower intake of essential vitamins (Walker, 1995). It was reported that gastroenteritis⁵ with protein-energy-malnutrition (PEM) is one of the main reasons for black children being admitted to hospital. Among others, this type of malnutrition have been noted to lead to coronary heart disease, an increase in dental caries, obesity and hypertension, diabetes, chronic bowel disease, and even diet-related cancers.

“Girls need health system changes that make the entire sector more responsive to their needs. Many such changes will require only marginal adjustments that are likely to be low cost.” (Temin, Levine and Stonesifer., 2010)

4.6. Chapter summary

The aim of this chapter was to provide an introductory background on the origin of the diet problem and some insight into developments in linear programming techniques as well as linear programming and expert system applications, followed by some background on the health status among teenagers in South Africa.

Chapter 5 will provide a description of the process that was used to gather information and knowledge about the methods used to develop a diet for an individual and discuss the initial steps followed during expert system creation.

⁵ Gastroenteritis is a medical condition categorized by infection in the gastrointestinal tract and is usually accompanied by vomiting and stomach cramps (Elliot, 2007).

Chapter 5 – Knowledge engineering: Planning and definition

5.1. Introduction

The previous chapters provided the background information and motivation for developing an expert system that creates a custom diet for an individual by using linear programming techniques.

The aim of this chapter is to guide the reader through the theoretical stages of the knowledge engineering process as well as the reasoning behind the phases which the system will be expected to perform.

Knowledge engineering encompasses the entire system development process, from the initial planning stages to the production of the final system. This chapter will start by briefly motivating the development paradigm according to which the expert system was created. Sections 5.3 and 5.4 will discuss the preliminary planning and knowledge accumulation stages which form the theoretical design of the system.

The practical implementation of the design into an expert system will be discussed in the knowledge design and code and checkout stages in chapter 6.

5.2. Development paradigm

The hypothesis that expert system principles can be combined successfully with linear programming techniques was evaluated in this study by designing and creating such a system within the positivistic research paradigm which was discussed in chapter 1. The paradigm within which an expert system is developed is referred to as the *knowledge paradigm*, the steps and phases of which were outlined in sections 3.5.1 and 3.5.2.

To test the hypothesis, the first four stages in the knowledge paradigm were performed to create the expert system artefact. Within the knowledge paradigm, the knowledge definition stage involves tasks which correlate to the data collection methods in the positivistic research paradigm, and the tasks in the knowledge design phase relate to data analysis.

The tasks performed within the planning and knowledge definition stages are discussed in the next sections, whereas the tasks in the knowledge design and code and checkout stages will be discussed in chapter 6.

5.3. Planning

The tasks that were performed in the planning stage include task phasing, feasibility assessment, preliminary functional layout and high-level requirements.

5.3.1. Task phasing

The phases and respective tasks that were performed to develop the expert system are shown in Table 5-1.

Stage	Task
Planning (Chapter 5)	Feasibility assessment Task phasing Preliminary functional layout High-level requirements
Knowledge definition (Chapter 5)	Knowledge source identification and selection Knowledge acquisition, analysis and extraction
Knowledge design (Chapter 6)	Knowledge classification Detailed design
Code and checkout (Chapter 6)	Coding Tests Source listings

Table 5-1: The phases to be performed within each task

5.3.2. Feasibility assessment

Feasibility assessment is done to determine whether an expert system is needed to solve a specific problem. In section 3.5, a number of questions were posed that were used as guidelines for assessing the feasibility of the expert system created in this study.

1. *Can the problem be solved by means of conventional programming?* Although the process followed to create an eating-plan for an individual consists of an

ordered number of steps, the proposed expert system will not create a solution according to an existing standard, step-by-step algorithm. Instead the system will pose certain questions to the user and use the answers obtained to determine how the solution will be found.

2. *Is the knowledge domain of the problem well-bounded?* The knowledge domain is well-bounded within the problem domain of dietetics, as shown in Table 5-2, and can therefore be solved by an expert system.

Age	Gender	Country/region
16-18 years	Female	South Africa (Northwest Province)

Table 5-2: Boundaries for the diet problem to be solved by the expert system

3. *Is there a need for an expert system in the domain and setting?* The need for an expert system that solves the diet problem for a female teenager in South Africa follows from the literature discussed in chapter 4.
4. *Is there at least one human expert who is willing to assist the knowledge engineer?* There are human experts who are willing to assist in the knowledge engineering process. Difference of opinions between the experts regarding some aspects of the diet problem confirms that the knowledge is uncertain and mainly experience-dictated.

Considering all the aspects discussed, the feasibility of creating an expert system to solve the specific diet problem was confirmed.

5.3.3. Preliminary functional layout

The purpose of the system is to provide the user with the capability to create a personalised diet that meets the required nutritional criteria and incorporates the food preferences of the user whilst minimising the total cost of the resulting diet.

The high level functions that the system should be able to perform include prompting the user for certain personal information needed to do the necessary calculations and providing feedback on the calculated eating-plan for the individual. The resulting eating-plan must conform to the nutritional requirements for a teenage girl between the ages of 16 and 18 years and also incorporate the food preferences of the user.

The functional components involved in the system are the user interface (section 3.4.4) and working memory (section 3.4.2) for obtaining information and compiling data, and the inference engine (section 3.4.1) and knowledge base (section 3.4.3) which are responsible for doing the necessary calculations and developing the resulting diet. The user interface will also be utilised to provide the final eating-plan as well as some additional information.

5.3.4. High-level requirements

The functions of the user interface will be accomplished by the use of the Visual Basic for Applications (VBA) (Albright, 2012) capability in Microsoft Excel with the basic calculations for data preparation by the working memory coded in the Visual Basic programming language.

The precompiled knowledge base and inference engine will be represented by a linear programming model that uses integer-, goal-, and multi-objective linear programming techniques incorporated into Microsoft Excel.

5.4. Knowledge definition

The aim of knowledge definition is to describe the knowledge requirements for the proposed expert system. To define the requirements for the system, the sources of the expert knowledge have to be identified, evaluated, and selected. Section 5.4.1 discusses the source identification and selection task in more detail. The knowledge acquisition, analysis and extraction tasks are discussed in the sections thereafter.

5.4.1. Knowledge source identification and selection

The sources of knowledge which were considered in this study include books, study material of undergraduate courses in dietetics at the North-West University Potchefstroom Campus, human experts, electronic and internet resources, and journal articles and publications. The sources that will be used for knowledge acquisition will be selected on the basis of the availability and importance of each, which is summarized in Table 5-3.

Knowledge source	Availability	Importance	Priority
Books	High	High	2
Study material	Subject to collaboration with human experts	High	2
Human experts	High	Very high	1
Electronic and internet resources	High	Moderate	4
Journal articles and publications	Moderate	High	3

Table 5-3: Ratings of sources of knowledge for the expert system

The reliability of human experts who are willing to help makes this source of knowledge absolutely invaluable (Giarratano and Riley, 2005) because it means that people who have practical experience in dietetics are willing to share and explain their expertise and increase the quality of the knowledge base. The human experts used in this study included three experts who teach and do research in dietetics, as well as two qualified practising dietitians. Human experts have the highest rating according to importance and availability and were selected so that they could be consulted for the creation of the knowledge base.

The study material referred to in Table 5-3 includes the study guides as well as the text books used by the human experts teaching subjects in dietetics (Nienaber-Rousseau, Venter and Wright, 2012; Whitney and Rolfes, 2008). Sections of the study guides as well as the text books were provided by the experts and will be used in conjunction with one another at a priority rating of 2.

Although journal articles and publications are considered very important and reliable, the availability of such items was not as high as that of the previously discussed sources. The majority of articles and publications that were considered discuss work in settings and environments other than South Africa which, although the information is studied, renders the priority rating a 3.

General electronic resources and internet pages are considered non-authoritative (Naufel, Briley, Harackiewicz, Marzec and Nielsen, 2010) and will only be referred to if the information used can be verified by another authoritative source, hence the priority rating of 4.

5.4.2. Knowledge acquisition

To build the knowledge base of the expert system, one needs to understand how a dietician compiles a diet for an individual. This information was gathered during the knowledge acquisition task which included the following steps:

- *Interviews* were conducted with lecturers and researchers in the dietetics field.
- *Interviews* were conducted with qualified dieticians in practice.
- Learners from Potchefstroom Gimnasium High School were asked to partake in an independent set of consultations with the practising dietician which was *observed*.
- Additional *documents* were studied to identify steps common to varying diet compilation methods.

5.4.2.1. Interviews conducted

Interviews and discussions were conducted with lecturers and researchers in dietetics at the North-West University (NWU) which provided some valuable information on the diet problem.

1. In recent times, an increase in eating disorders has been observed among teenage girls in South Africa. This trend was also confirmed by the literature discussed in chapter 4.
2. Although computer technology progresses at a daily rate, dieticians do not seem to want to benefit from the advantages it could offer them. Dieticians are still doing extremely time consuming manual calculations and using standard calculators to develop diets for individuals. Although many software applications have been created for the purpose of solving the diet problem, they are subject to the environments for which they are created. For instance, a program that develops an eating-plan for an average Malaysian household (Hamzah *et al.*, 2011) cannot be used to design a diet for a teenage girl in South Africa for many reasons. The staple food of a South African differs from that of a Malayan native, the cultures are not the same, the same food types cannot be found in both countries, etc.
3. There is concern that girls living in rural areas in South Africa do not have access to information provided by a qualified dietician which could help them establish healthy eating patterns. It has been indicated that a computer

program might reach more people in need of some vital information regarding their eating habits. Although the primary objective of this study is not to create such an artefact for the purpose of distributing it on the internet, the fact that such a system does not exist did add additional motivation for creating an expert system with a linear programming inference engine in the dietetics field.

4. A need exists for a computer system that considers certain health conditions of a person before developing the eating-plan for the individual.
5. The process that is taught to undergraduate students at NWU on how to compile an eating-plan for an individual was explained. It is important to note here that some of the personal information required from an individual for the purpose of compiling a personalised diet include age, height, weight, activity level, and gender. The compilation process is discussed in more detail in section 5.4.3.

In order to gain some more comprehensive insight into the development of a diet, interviews were also conducted with two qualified dieticians who have practical experience in developing diets for individuals.

Insights gained include:

1. Many dieticians create their own computer programs that assist in the preliminary calculations needed to develop a diet. These programs do not however, have the capabilities of randomly selecting food types from a list of options to include in a diet. When a dietician has calculated the final nutrient distribution for the formulation of the diet, a manual process is used to select and add foods to the diet while keeping some sort of count of the totals of the nutrient contributions after each addition.
2. The food choices that a dietician selects for the final menu are very much dictated by habit rather than considering what the person's likes and dislikes are.

It was agreed with the dietician that a certain number of volunteers were to be asked to partake in consultation sessions with the practising dietician. The dietician was asked to compile a healthy eating-plan for the volunteers and the dietician's question-and-answer process of gathering information for each of them were

recorded. Due to the expert system to be developed aimed at teenage girls in South Africa, the factors in this environment that will remain constant are therefore age and gender. It was decided that the volunteers would therefore be selected according to similar ages and varying activity levels.

5.4.2.2. Observations

Following e-mail correspondence with the deputy-headmaster of Potchefstroom Gimnasium high school, permission was given that all the female learners between the ages of 16 and 18 years could be invited to partake in the consultation sessions with the dietician. A preliminary list of volunteers was compiled by the deputy headmaster, and the girls were asked to note their ages and their daily activity levels.

Due to funding restrictions, a total number of six girls were selected that were all within one year of each other's age but had indicated varying activity levels on the initial name list. The purpose of the consultation sessions was explained to the volunteers and each one was given a letter to explain the study and to clarify payment terms to their parents. A copy of the letter to the parents is attached as Appendix A. Thereafter each volunteer was contacted individually for appointments with the dietician.

Within a time period of two weeks, the sessions had been completed and the resulting diets had been provided to the volunteers. A summary of the transcriptions of the consultation sessions is attached as Appendix B. Due to the slight differences in the diets received from the dietician, only one of them is attached as an example per Appendix C.

There are some differences between the diet compilation process taught to undergraduate students at NWU and the steps the practising dietician follows. The reason for the difference is mainly experience-dictated and this will be discussed in more detail in section 5.4.3 where the dietician's compilation process is also discussed.

5.4.2.3. Researching documents

For the purpose of gaining comprehensive insight into the compilation process of an eating-plan, additional documents were studied to confirm the steps that are

common to experts in academia and in practice. The common steps were identified and it was decided that the steps that proved different in respective processes would be programmed in the expert system as factors that need to be acquired from the user. The additional documents that were studied included sections of study material that was provided by the experts in academia.

The proposed system to be evaluated will aim to provide a diet at minimum cost. Therefore electronic documents will also be studied with the purpose of building a cost database for all of the food items available in this study.

Literature sources were studied to determine if and how some health conditions like headaches, fatigue, bloating and constipation could be catered for in the expert system. The findings are discussed in section 5.4.3.

5.4.3. Knowledge analysis

The knowledge analysis section identifies the knowledge that is essential to the proposed system and specifies the functional layout in terms of general phases. The diet compilation processes of the academic experts, the practicing dieticians and commonalities in additional documents are discussed.

During the interviews conducted with researchers in dietetics, the need was expressed for a diet compilation system that takes health conditions into consideration when providing an eating-plan. Studies on health conditions are also discussed here.

5.4.3.1. How to compile a diet

The steps as described in the interviews are substantiated by the study material (Nienaber-Rousseau *et al.*, 2012) and textbooks (Whitney and Rolfes, 2008; Mahan and Escott-Stump, 2004) are as follows:

Step 1: Calculate the total daily energy requirement for the individual.

The Harris-Benedict formula is used to calculate the basal metabolic rate (BMR) or resting energy expenditure (REE) which is an estimate of the energy needed to sustain the human body (Nienaber-Rousseau *et al.*, 2012). The Harris-Benedict equations differ for male and female and are shown in (5-1) and (5-2).

$$\text{For males: } BMR = (66.5 + 13.8m + 5h - 6.8a) \times 4.2 \quad (5-1)$$

$$\text{For females: } BMR = (655 + 9.6m + 1.8h - 4.7a) \times 4.2 \quad (5-2)$$

where m denotes the mass of the person in kg,
 h denotes the height of the person in cm,
 a denotes the age of the person in years.

Step 2: Multiply the BMR with the activity factor.

Multiply the value obtained by calculating the BMR for a female from (5-2) with the activity factor which is determined by the activity level of the person as explained in Table 5-4.

Activity factor	Category	Definition
1.2	Sedentary	Little to no exercise or deskwork
1.375	Low level of activity	Little exercise, 1-3 times per week
1.55	Medium level of activity	Medium exercise, 3-5 times per week
1.725	High level of activity	Strenuous exercise, 6-7 times per week
1.9	Extremely active	Extreme daily exercise

Table 5-4: How to determine the activity factor from the activity level

Step 3: Modify the energy requirements to cater for energy needed for growth.

The additional energy requirements for growth are dependent on the age and gender of the person under consideration. The study was conducted to develop a diet for girls specifically, so the information provided was to accommodate girls only. Table 5-5 shows the additional energy requirements based on the age of the girl.

Age (years)	Additional daily energy requirement
10-14	+8 kJ per kg of weight per day
15	+4 kJ per kg of weight per day
16-18	+2 kJ per kg of weight per day

Table 5-5: How to calculate the additional daily energy requirements of a growing girl

It was emphasized during the interview that it is important for a dietician to have the option to increase or decrease the daily energy requirement.

Step 4: Modify the daily energy requirement to cater for a person who needs to either lose weight or gain weight.

The dietician likes to reserve the right to modify the daily energy requirement based on information provided by the person regarding possible weight gain or weight-loss. Dieticians and other health professionals can determine whether it is necessary to modify the intake recommendations to meet the needs of the person (Whitney and Rolfes, 2008).

The choice of the amount by which the energy requirement is to be modified is determined by the dietician. For a weight increase or loss of about 0.5 kg per week, the energy requirement can be increased or decreased by 1300-3500 kJ.

Step 5: Determine the distribution of the daily energy requirement into carbohydrates, protein and fat.

The recommended intakes for individuals as tabulated in either *Krause's Food, Nutrition and Diet Therapy* (Mahan and Escott-Stump, 2004) or *Understanding Normal and Clinical Nutrition* (Whitney and Rolfes, 2008) are used to determine the distribution of the daily energy into the macronutrients carbohydrates (CHO), protein and fat.

The guidelines provided in these books are summarized for a female between the ages of 14-18 years of age in Table 5-6. Specifically, *Krause's Food, Nutrition and Diet Therapy* suggests that the protein content of the resulting diet is calculated by the estimate of 0.85g per kilogram of body weight per day and *Understanding Normal and Clinical Nutrition* uses the estimate of 0.8g per kilogram of body weight per day. The values for the CHO are read from the tables in the respective books and the data content is calculated from the amount of energy that is left of the daily requirement after subtracting that of the other two macronutrients. When the percentages of the macronutrients are known, the sugar content which should be an estimated 10-15% of the total daily energy requirement is calculated. Sugar will form part of the CHO macronutrient grouping in the final diet.

Source	CHO as a percentage of energy required	Protein as a percentage of energy required	Fat as a percentage of energy required
<i>Krause's Food, Nutrition and Diet Therapy</i>	45-65%	10-30%	25-35%
<i>Understanding Normal and Clinical Nutrition</i>	45-65%	10-35%	20-35%

Table 5-6: Recommended macronutrient intakes for girls between the ages of 14 and 18 years

It was understood from the interview that the percentage protein is calculated first, after which the CHO are calculated and then the fat. Once the percentages for the macronutrients have been calculated, the weight amounts of each to be used in the diet are calculated in grams, as follows:

$$protein(g) = \frac{percentage\ protein(\%) \times total\ daily\ energy\ requirement(kJ)}{17(kJ/g)} \quad (5-3)$$

$$CHO(g) = \frac{percentage\ CHO(\%) \times total\ daily\ energy\ requirement(kJ)}{17(kJ/g)} \quad (5-4)$$

$$fat(g) = \frac{percentage\ fat(\%) \times total\ daily\ energy\ requirement(kJ)}{38(kJ/g)} \quad (5-5)$$

where 17kJ of protein weighs 1g,
 17kJ of CHO weighs 1g,
 38kJ of fat weighs 1g.

Step 6: Create the diet plan.

Exchange lists are dietary tools that help the dietician to organize foods according to their contribution proportions of the three macronutrients (Whitney and Rolfes, 2008). The concept of an exchange list is used to determine how many food exchanges to include in the diet, from each of the food groups in the food pyramid. Say for example, the fruit exchange list contains 50 types of fruits that provide the same macronutrients and only 3 fruits are allowed per day. Then the dietician can select any 3 from the exchange list to include in the diet. The exchange lists and their respective macronutrient contributions are shown in Table 5-7.

List	Number of exchanges	CHO (g)	Protein (g)	Fat (g)	kJ
Milk					
Full cream	1	12	8	8	640
Low fat	1	12	8	5	525
Fat free	1	12	8	1	375
Fruit	1	15	0	0	255
Vegetables	1	5	2	0	120
Meat					
Lean	1	0	7	3	230
Medium	1	0	7	5	310
High in fat	1	0	7	8	420
Bread/starch	1	15	2	0	300
Fat					
Mono-unsaturated	1	0	0	5	190
Poli-unsaturated	1	0	0	5	190
Saturated	1	0	0	5	190

Table 5-7: The exchange lists from which foods are chosen to add to a diet

The *food guide pyramid* is a food guidance system which was used by the United States Department of Agriculture (USDA) to develop *MyPyramid*, a set of dietary guidelines for Americans (FNIC, 2013).

The basic guidelines set forth in the food pyramid include:

- Use fats, oils and sweets sparingly.
- Dairy products: include 2-3 servings of milk, yoghurt and cheese.
- Protein: use 2-3 servings of meat, poultry, fish, dry beans, eggs and nuts.
- Vegetables: include 3-5 servings of vegetables.
- Fruit: 2-4 servings of fruit should be taken.
- Starch: 6-11 servings of bread, cereal, rice and pasta should be included.

The recommendations are to be used as guidelines instead of rigid rules and are for the daily food intake (Haven, Burns, Britten and Davis, 2006).

The interviewee explained that the exchange selection process is an iterative process, during which the dietician selects a certain item from the list, determines the quantity of that item to be used and calculates the contribution of CHO, protein and fat resulting from the selection. For example, if a dietician chooses to add 3 exchanges of full cream milk products to the diet, it would contribute 36g CHO, 24g of protein and 24g of fat to the macronutrient intake for that day. The total CHO, protein and fat is monitored throughout the selection process to ensure that the required amounts are not exceeded. The process of selecting food items is iterative because at some point the diet will reach a point where one or more macronutrient has reached its daily level while the other(s) have not.

Step 7: Create a meal plan.

The exchanges chosen in step 6 are distributed between the different meals of the day.

Step 8: Create the menu.

The meal plan is transformed into a menu by replacing the exchanges with food items from the food group represented.

5.4.3.2. Observations feedback

The practicing dietician was asked to develop an eating-plan for each of the six volunteers. Included in the diets is the daily energy requirement for each person, the macronutrient distribution and a complete menu plan. The diets that were obtained for the volunteers will be used to evaluate the expert system outputs.

5.4.3.3. Documents analysis

Interviews with researchers in dietetics (Rankin, 2012) revealed that there is a demand for a diet planning computer system that accommodates health issues like fatigue, headaches, bloating, and constipation. Additional documents were researched to establish whether the proposed system should accommodate these conditions and these documents will be discussed briefly in the following paragraphs.

Due to the study being conducted in South Africa, a food item cost database was created by identifying two of the major supermarket franchises in the country who provide internet shopping. The supermarkets that were consulted in this regard are Pick 'n Pay (2012) and Woolworths (2012). Food items included in the system were

selected, based on the responses of the volunteers for the consultation sessions as well as results obtained in an independent project conducted to determine the food preferences of 192 high school dormitory learners (Linde and Delport, 2012). The price database that was compiled is attached as Appendix D and will be discussed in more detail in chapter 6.

The medical community is undecided whether diet can influence constipation but rather recommends that people drink lots of water and have a healthy daily fibre intake (Stewart and Schroeder, 2013).

Experiments conducted to determine the effect of dietary intervention upon chronic fatigue syndrome revealed that the safest dietary intervention would be to increase the fibre intake, have at least 5 servings of fruit and vegetables per day, reduce fat consumption and increase the fish intake to at least twice a week (Hobday, Thomas, O'Donovan, Murphy and Pinching, 2008). Although the importance of a healthy diet is emphasized (Head and Kelly, 2009) for help with fatigue, no strict dietary interventions are recommended for the treatment of fatigue.

A study on dietary influence or causes of headaches revealed that the consumption of cocktails and coffee, smoking, and a lack of physical activity have the highest influence on headache attacks (Midle-Busch, Blaschek, Borggäfe, Heinen, Straube and Von Kries, 2010). Dietary recommendations for assisting in headache relief include a lower consumption of alcoholic drinks, lower coffee intake and an increase in water intake (Torelli, Evangelista, Bini, Castellini, Lambro and Manzoni, 2009).

McKenzie, Alder, Anderson, Wills, Goddard, Gulia, Jankovich, Mutch, Reeves, Singer and Lomer (2012) presented some dietary intervention guidelines for the treatment of irritable bowel syndrome (IBS), of which bloating is a symptom. However, it was noted that a person who does not suffer from IBS should rather identify the food items that cause bloating and try to avoid them in future.

Although many studies are conducted regularly on possible dietary interventions for the purpose of treating health conditions, the authors in general agree that if a person suffers from one of the conditions mentioned, the best solution would be to consult a health-care professional. In the light of this information and the fact that the dietician was asked to provide a simple healthy eating-plan for each of the

volunteers, it was decided that conditions like fatigue, headaches, constipation and bloating will not be considered or included in the expert system.

5.4.4. Knowledge extraction

The purpose of the knowledge extraction task is to specify the functional capabilities of the system, describe the general phases that the system will execute, and specify the requirements that will be used to evaluate the system.

5.4.4.1. Functional layout

According to section 5.3.3, the first step in compiling a custom diet is to calculate the daily energy requirement for the person under consideration but conversations with the experts revealed a certain hesitancy to trust a computer program to complete this step. It was decided that the user of the system will be offered the choice of calculating the daily energy requirement manually and simply entering the value into the system or, providing the system with the values necessary to do these calculations electronically. The user will also be offered the choice of changing the calculated energy requirement if needed.

The diets calculated by the practising dietician revealed that the macronutrient distribution used for all of the volunteers was 52% CHO-, 19% protein-, and 29% fat of the energy required, due to their ages and gender being constants in this study scenario. For the purpose of uniformity the initial levels of the required macronutrient distribution will also be set to these levels. The choice will however also be offered to the user to change the macronutrient distribution.

The step during which the dietician selects the exchanges from each food group to add to the diet is the single most time consuming stage of the entire compilation process. One source even confessed that it had taken her a total of two weeks to finish the iterative process with an acceptable selection of foods. A linear programming model will be used to solve this problem, the formulation and discussion will follow in chapter 6. The introduction of linear programming techniques into the system provides the added benefit of incorporating the objective of maximising the food preferences of the user into the food selection process. Another objective that will be considered in the model is the minimisation of the total cost of

the diet by using the food price database. The food list will be presented to the person and she will be prompted to enter her level of preference for each item.

The output that the program will provide will be in the form of a 1-day eating-plan, with food items organised according to their eligibility for each of the three meals or snacks.

5.4.4.2. The phases of the expert system

The phases needed for performing the functions outlined in section 5.4.4.1 are:

- Calculate the daily energy requirement.
- Present the food list to the user and obtain a list of her preferences.
- Use the food preferences and the food price database to solve the linear program that provides a list of foods that optimises both goals.
- Provide the output to the user.

Once the knowledge definition stage has been concluded, the theoretical design of the expert system is complete and the practical implementation of the design can start. Therefore this chapter continues in chapter 6 where the practical design and implementation of the system will be discussed.

5.5. Chapter summary

Chapter 5 started by briefly explaining the relationship between the research paradigm and the development paradigm within which the system was developed. The phases of the knowledge paradigm for an expert system that overlaps with that of the positivistic paradigm were also clarified.

The tasks that were performed in the planning phase were discussed in section 5.3 and include setting up a task phasing schedule, doing a feasibility assessment for the proposed system, preparing a preliminary functional layout and defining the high-level requirements of the system.

A detailed account of the knowledge definition phase followed in section 5.4. The section started by discussing the methods used to collect the data necessary to form a clear understanding of the diet compilation process and the need for a computer system to aid in this regard. The knowledge sources were discussed and rated and the data collection was discussed. The information acquired during the collection

processes was analysed and the knowledge needed for the expert system was identified. The functional layout of the proposed system was provided and translated into the phases that the program will need to execute in section 5.4.4.2.

Chapter 6 will discuss the last two stages of the knowledge engineering process for this study, the *knowledge design* and *code and checkout* stages. These stages entail the practical development, implementation and evaluation of the expert system.

Chapter 6 – Knowledge engineering: System development and evaluation

6.1. Introduction

Chapter 5 discussed the first two phases of the knowledge engineering process of the expert system. These phases include the planning and knowledge definition phases, which involved the establishment of an overall theoretical design for the system. The purpose of chapter 6 is to discuss the development and evaluation stages of knowledge engineering by describing the practical design and implementation of the system, the evaluation process, and the results.

The chapter starts in section 6.2.1 by defining the knowledge that is used in the system in terms of the input data and the structures that are used for the necessary elements in the application program, as part of the knowledge classification task.

A description of the detailed design follows in section 6.2.2 where the linear programming models are discussed. It is also explained how the rules are incorporated into the models. Section 6.2.3 provides a detailed description on the implementation of the structures and models in the expert system which utilises the linear programming models to perform the necessary inference.

Testing and evaluation of the system is discussed in section 6.3 wherein the consultation sessions between the dietician and the volunteers were concretely applied to the expert system and the results are considered and discussed.

The chapter finishes in section 6.4 by presenting a general discussion on the development of the expert system for solving the diet problem, offering some considerations regarding the execution and evaluation of the system and some final conclusions regarding the study.

6.2. Knowledge design

In the knowledge design phase, the knowledge classification task specifies the knowledge representation structure and the general control structures for the rules of the system. The detailed design task specifies how the knowledge is organised, how the system is to be implemented and creates the user interface design.

6.2.1. Knowledge classification

Microsoft (MS) Excel combines an object oriented programming approach through spreadsheets with procedural programming via Visual Basic for Applications (VBA) and it has the added benefit of the use of Solver to solve linear programs which makes it the ideal implementation platform for this expert system.

Upon completion of the knowledge acquisition phases, the data that will be required from the user has been identified as the following:

- The values needed for calculation of the daily energy requirement;
- The daily energy requirement (in the case of the dietician choosing to calculate it manually);
- The macronutrient distribution;
- The food items in the exchange lists;
- The food preferences of the patient; and
- The cost information of the food items in the exchange lists.

The rules that will be incorporated in the inference engine via linear programming techniques include:

- The guidelines for the upper and lower limits of the number of exchanges from each group;
- The weight units of each macronutrient that were obtained by applying the guidelines for the number of exchanges;
- The food preference ratings of the user for each food item in each exchange list;
- The cost information of each food item in each list; and
- The priority of the user regarding a least cost or highest food preference diet.

The layout of the control structure of the system in Figure 6-1 will be used to discuss the data structures of the knowledge and data.

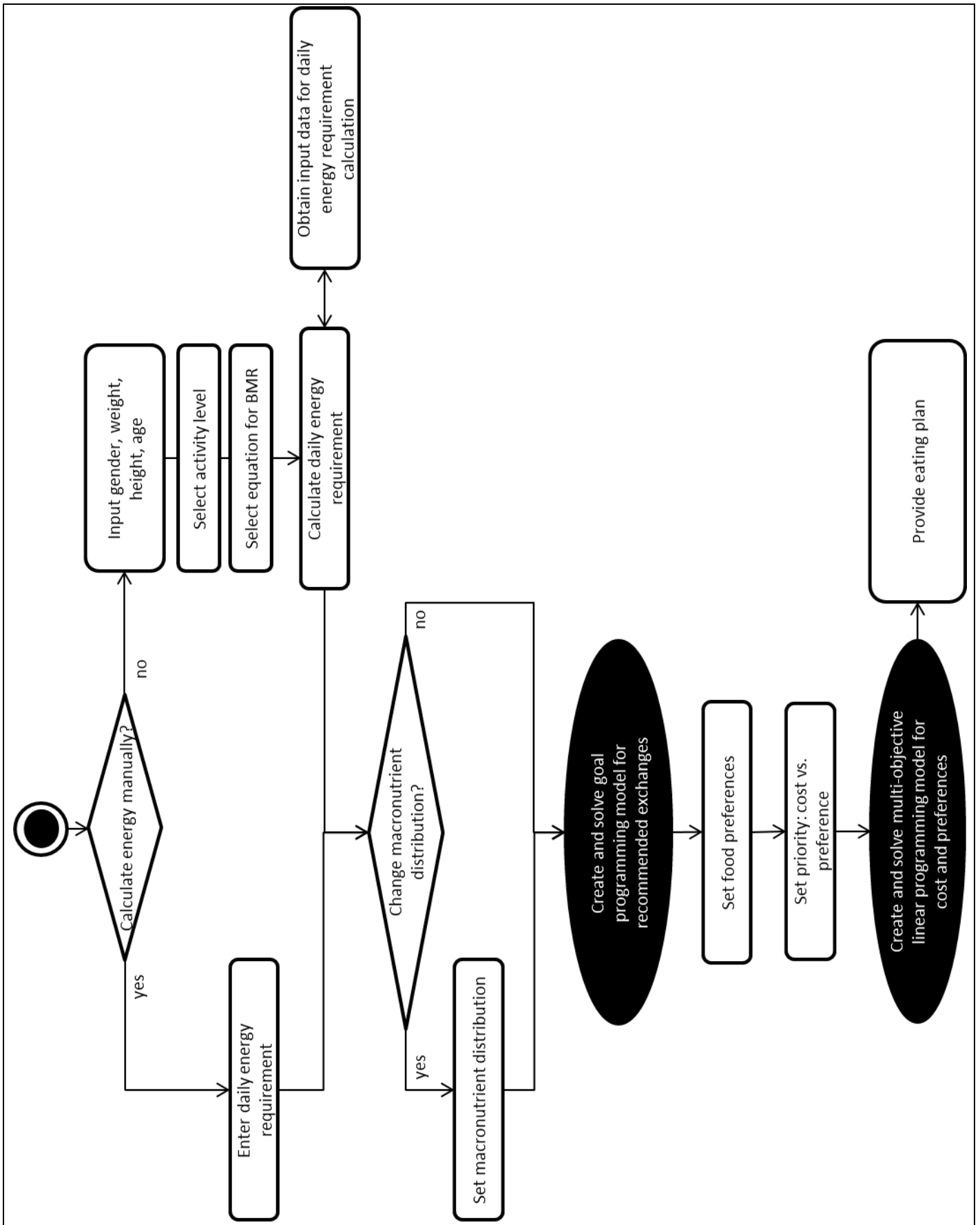


Figure 6-1: Control structure of the expert system

The initial choice of whether or not the user wants to calculate the daily energy requirement manually, is represented by a *commandbutton* structure which is a programmable control structure that will execute when a certain event occurs, for example when the button is clicked. The layout and design of the user interface is discussed in section 6.2.2.

The input values needed for energy calculation by the system are acquired by utilising the *cell* variable structures within MS Excel. A cell adjacent to the input location of each variable is used to prompt for the required value. The activity level and equation for calculating the BMR are coded as *in-cell dropdown lists*, allowing the user to select the activity level and BMR equation amongst a number of options rather than having to type the required information. The body mass index is calculated which provides a general relationship between the weight and height of the person being considered (Cole, Freeman and Preece, 1995), and is used to determine whether the patient needs to lose weight, eat healthily or gain weight. As mentioned in chapter 5, the dietician prefers to have some control over the amount of energy to deduct or add to the daily requirement. Therefore the cell representing the amount of energy that is added or deducted from the daily requirement, is represented by a *spinbutton* control which allows the user to change this quantity as needed. The energy needed for growth in terms of the age of the patient is determined automatically by means of the *VLOOKUP* function in Excel which “looks up” the value of the required variable based on the value of the age.

Although the macronutrient distribution used for teenage girls is a constant distribution of 52% carbohydrates (CHO), 19% protein and 29% fat of the energy requirement, the user is given the choice of changing the distribution by means of the *inputbox* structure in Excel. The final distribution levels are passed as parameters to the next phase of the program.

The goal programming model that is used for calculating the number of exchanges from each exchange list is discussed in section 6.2.1.1. Important to note here is that the energy requirement and macronutrient distribution is used as input to the model and the solution provided represents the exact number of exchanges that should be chosen from each of the exchange lists.

A list of frequently consumed foods have been compiled for each exchange list based on the responses from the volunteers who took part in the consultation sessions as well as the data from the independent study in 2012, as mentioned in chapter 5. As an example, items in the **Fruit** exchange group that provide similar macronutrient contributions are shown in Table 6-1.

Fruit	Serving size
Strawberry	150 g
Apple	80 g
Apple juice	90 g
Dates	15 g
Grape	75 g
Granadilla	90 g
Cherry	75 g
Guava	85 g
Orange	100 g
Litchi	70 g
Mango	70 g
Pawpaw	100 g
Pear	100 g
Peach	100 g
Grapefruit	125 g

Table 6-1: Food list for the *Fruit* exchange group

Each of the items in the **Fruit** exchange list will result in similar contributions to the total macronutrient count provided by the resulting diet, if selected by the program based on the preference entered by the user. Food lists have been compiled for each of the exchange lists presented in Table 5-7 (section 5.4.3.1), and are included in the list in Appendix D. All of the items on the food list have also been categorized according to meal suitability, for example grapefruit is categorized as *breakfast*. The reason for the categories is to enable the system to compile the final eating-plan according to categories. The user is presented with each food list and asked to enter her preference for each item in the list based on a Likert scale (Allen and Seaman, 2007). If the user is allergic to any of the items on any food list, she will be prompted

to enter a preference rating of 100 which will ensure that the item is not selected for the final diet.

A price list was compiled from the online shopping websites of two of the major supermarket chain stores available in Potchefstroom, Northwest, South Africa. The relevant stores are *Pick 'n Pay* (2012) and *Woolworths* (2012). The price list includes all the food items in all the respective food lists. The phenomenon of different serving sizes providing similar macronutrient distributions in each exchange group, presented a challenge when compiling a price list to use for determining the least cost item. For each item in the list, the price and selling pack size was determined by searching for it on the two selected supermarket websites. The price for each item was then calculated per serving size using the following equation:

$$price = MIN \left[\left(\frac{\text{serving size}}{\text{pack size } PnP} \times \text{pack price } PnP \right), \left(\frac{\text{serving size}}{\text{pack size } Ww} \times \text{pack price } Ww \right) \right] \quad (6-1)$$

where *PnP* refers to Pick 'n Pay, and

Ww refers to Woolworths.

The *MIN*-function determines the minimum of the values in brackets and assigns it to the final price. For items that are seasonal and which could not be found on the websites at the time of compilation, the final price was set at an ultimate high level of R1000 which effectively eliminates them from being selected by the program. Table 6-2 shows an excerpt from the price list where the **Fruit** items are listed.

Fruit	Serving size	Pack Price (R) Pick 'n Pay	Pack Price (R) Woolworths	Pack size Pick 'n Pay (relevant unit)	Pack size (relevant unit) Woolworths	Price / required quantity (R)
Strawberry	150 g	29.99	29.99	400	400	11.25
Apple	80 g	9.99	17.99	1000	1500	0.80
Apple juice	90 ml	9.99	15.95	500	1000	1.44
Dates	15 g	24.99	39.99	250	400	1.50

Grape	75 g	19.99	1000	500	1	3.00
Granadilla	90 g	15.99	24.99	500	500	2.88
Cherry	75 g					1000
Guava	85 g	26.99	1000	660	1	3.48
Orange	100 g	5.99	7.99	1000	1000	0.6
Litchi	70 g					1000
Mango	70 g	24.99	1000	100	1	17.49
Pawpaw	100 g	10.00	15.99	781	314	1.28
Pear	100 g	17.99	16.99	1500	1500	1.13
Peach	100 g					1000
Grapefruit	125 g	9.99	1000	492	1	2.54

Table 6-2: Price list as compiled for the *Fruit* list

The program uses a *messagebox* with an embedded *scrollbar* to allow the user to set her personal priority for the cost and food preference goals. If the user does not care about the cost, she will most likely want the diet to have the most of her personal food preferences in the diet and vice versa.

The price, preference and number of exchanges from each group are used as input for the multi-objective linear programming model which will be discussed in section 6.2.2.

Upon completion of executing the multi-objective linear programming model, the resulting one-day eating-plan is formulated and compiled in a list on a separate spreadsheet. The final eating-plan provides the user with an estimated total cost of all the items in the plan as well as an indication on the satisfaction ratio of her personal food preferences.

The data structures to be used for the discussed elements are shown in Table 6-3.

6.2.2. Detailed design

The use of linear programming techniques facilitates the accommodation of the rules for the expert system within the different models.

Data structure	Items
Cell variables	Gender Age Weight Height Activity level multiplication factor Energy required for growth Equation for BMR Daily energy requirement Macronutrient weight units Number of selections from each exchange group
Records (row with corresponding data)	Energy needed for growth Activity factor based on activity level One record for each food item
Objects	Energy calculation Data for energy calculation Goal programming model (for calculation of number of each exchange group) Food preference list (one for each exchange group) Price list Multi-objective linear programming model (for calculation of final eating-plan) Resulting eating-plan

Table 6-3: Data structures for the elements in the expert system

6.2.2.1. Goal programming model for number of exchanges from each list

One of the challenges faced by the dieticians when compiling a diet is the iterative trial-and-error process followed to choose the number of exchanges from each exchange group. The reason for this is that they start to randomly assign a number of selections for each exchange group, while keeping count of the total CHO, protein and fat contributions. When the total contribution of any of the macronutrients exceeds the tolerance of 3-4g, the exchanges are adapted by will and the macronutrient contribution recalculated. The iterative process followed by the dietician, is shown in the decision tree in Figure 6-2.

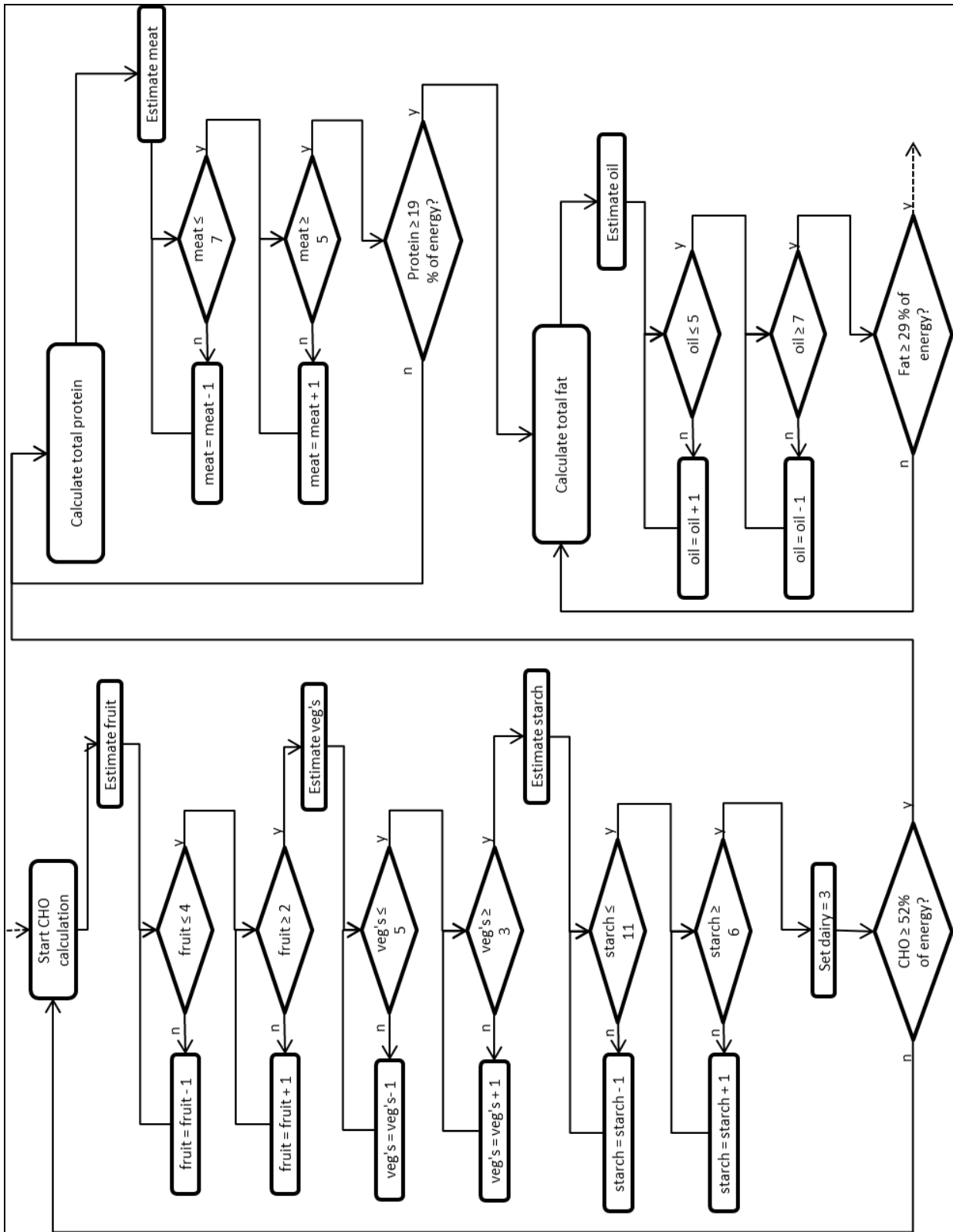


Figure 6-2: Decision tree representing the iterative reasoning process of the dietician

The reasoning of the dietician is incorporated into the system as a set of rules and implemented by means of a goal programming model. The goal programming model is used to determine the number of exchanges that should be chosen from each

exchange group so that the total contribution to the macronutrients is as close as possible to the distribution which, by default is 52% CHO, 19% protein, and 29% fat, or the distribution as modified by the user. Additional constraints in the model are those imposed by the guidelines in the food pyramid as discussed in chapter 5. These guidelines are shown in Table 6-4.

Food group	Minimum number of foods	Maximum number of foods
Fruit	2	4
Vegetables	3	5
Dairy	3	3
Starch	6	11
Meat	5	7
Fat	5	6

Table 6-4: Dietary guidelines for food group selections

The model is represented by (6-2) to (6-6).

$$\text{Minimise} \quad \sum_{i=1}^3 (u_i + v_i) \quad (6-2)$$

$$\text{subject to} \quad \sum_{j=1}^n a_{ij}x_j + u_i - v_i = M_i \quad (i = 1,2,3) \quad (6-3)$$

$$l_j \leq x_j \leq b_j \quad (j = 1, \dots, n) \quad (6-4)$$

$$u_i, v_i \geq 0 \quad (i = 1,2,3) \quad (6-5)$$

$$x_j, l_j, b_j \text{ integer } \forall j \quad (6-6)$$

where u_i represents the amount by which the solution exceeds the required value of macronutrient i ;

v_i represents the amount by which the solution falls short of the required value of macronutrient i ;

l_j represents the lower bound for exchange group x_j ;

b_j represents the upper bound for exchange group x_j ;

M_i represents the weight amount required from each macronutrient; and

a_{ij} represents the contribution of exchange group j to macronutrient i .

The upper and lower bounds for the number of exchanges are constrained to be integer values because one cannot make partial selections from a list. The final values of a_{ij} , the food preferences of the user, the price values of all the food items, and the cost and preference priority levels of the user are used as inputs to the multi-objective linear programming model that creates the eating-plan and which is discussed in the next section.

6.2.2.2. Multi-objective linear programming model for least cost and highest preference diet

The multi-objective linear programming model is utilised for each of the exchange groups. The number of foods from the particular exchange group is incorporated as a constraint that specifies how many food selections should be made from the group. The purpose of the model is to select foods based on their prices and preference ratings as entered by the user.

The objectives of the multi-objective linear programming model therefore are to simultaneously maximise the food preference and to minimise the cost of the resulting selections. The model is formulated using the MINIMAX principle as discussed in chapter 2 (section 2.5) and is shown in (6-7) to (6-11).

$$\text{Minimise} \quad Q \quad (6-7)$$

$$\text{subject to} \quad \frac{w_{pref}(\sum_{j=1}^n p_j x_j - t_{pref})}{t_{pref}} \leq Q \quad (6-8)$$

$$\frac{w_{cost}(\sum_{j=1}^n s_j x_j - t_{cost})}{t_{cost}} \leq Q \quad (6-9)$$

$$\sum_{j=1}^n x_j = N \quad (6-10)$$

$$x_j = 0 \text{ or } 1 \quad (6-11)$$

where N denotes the number of selections to be made;

$x_j = 1$ if food j is selected or otherwise $x_j = 0$;

p_j depicts the user's preference for food item x_j ;

s_j represents the cost of food item x_j ;

w_{pref} represents the weight as selected by the user for the food

preference as priority;

w_{cost} represents the weight for cost;

t_{pref} represents the optimum target for preference rating; and

t_{cost} represents the optimum target for cost.

For simplification purposes, instead of maximizing the preference rating, the user interface is designed so that the user is prompted to enter her preference rating for the food items from 1 meaning highest preference, to 5 meaning lowest preference. Therefore, minimizing the preference of food selections will result in selecting the foods that the user likes most.

Due to the presence of similar products with similar prices appearing in some of the lists, additional constraints are used to restrict the selection of similar items. Assume for instance that x_1 denotes potato and x_2 denotes mash. Constraining a selection to contain only one of these items is then formulated as follows:

$$x_1 + x_2 = 1 \quad (6-12)$$

Items that are considered similar are shown in Table 6-5.

Item	Similar item
Porridge (cooked)	SPP (raw) Rice flakes Hi-fiber All bran Puffed wheat Wheet-bix Muesli
Potato	Mash
Fresh/frozen mealie	Canned corn
Muffin	Scone Crumpet

Table 6-5: Similar food items to be constrained so that only one is chosen

Peanut butter is also a special case. If peanut butter, which resides in the protein group is chosen, then it also contributes two exchanges of fat/oil. The total exchanges to be selected from the oils group will then be adapted to accommodate this restriction.

The model is formulated in such a way that for every exchange group, the optimal preference rating target is calculated as well as the optimal cost. The optimal target for the preference is calculated by means of a standard minimisation linear program as formulated in (2-1) to (2-3) in chapter 2 (section 2.2.1).

Effectively, the model represents the rules that would determine the combination of foods that result in the best preference rating, as depicted in Figure 6-3.

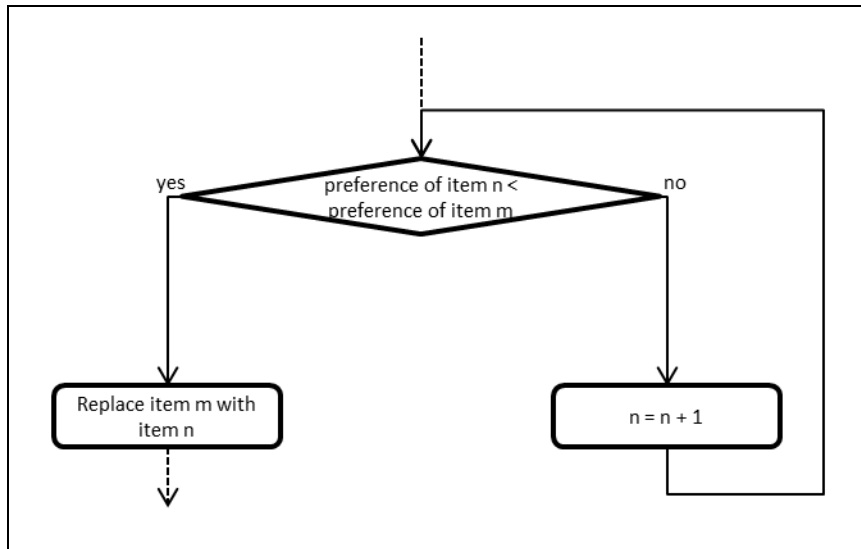


Figure 6-3: A typical rule that compares the cost of two items with each other

The target for the optimal cost is determined in a similar manner. The calculated optimal values for preference and cost are used as targets for the two respective objectives, of which the deviation Q is minimised.

The resulting eating-plan is formulated by displaying the food items that were selected by the multi-objective linear program. A preference rating is calculated when the eating-plan is compiled by calculating the margin by which the preferences of all the products in the solution eating-plan deviates from a perfect preference of 100%.

6.3. Code and checkout

As explained in section 6.2.1, the coding and implementation of the system is done by means of MS Excel. The elements listed as objects in Table 6-3 are implemented as spreadsheets. The main start-up options presented to the user are shown in Figure 6-4.

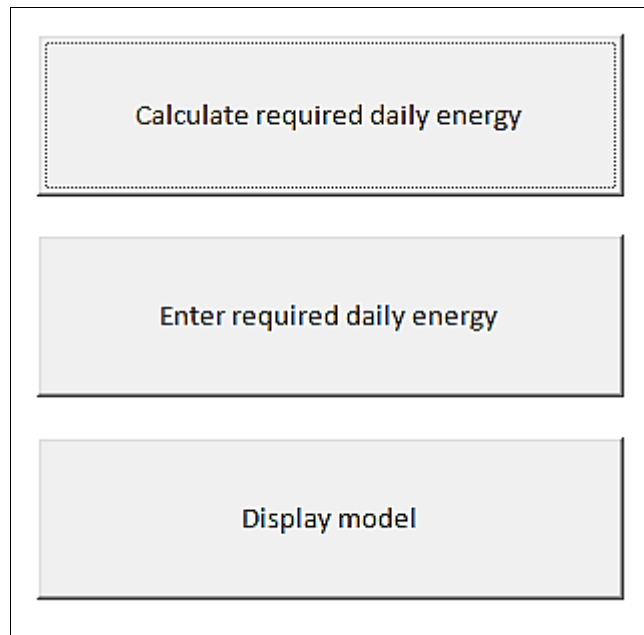


Figure 6-4: The start-up menu of the system

If the user chooses to calculate the required daily energy, the system activates the calculation object, which is shown in Figure 6-5.

Please complete the following:				
By the patient:				
Gender:	F		BMI:	
Weight:	79	kg	Actual:	27.01686
Height:	171	cm	Safe:	22
Age:	14.5	years	Ideal weight:	64.3302
Activity level:	Little to no exercise, deskwork		1.2	
Goal through diet:	Lose Weight			
By the dietician:				
Equation for BMS:	Harris-Benedict			
Calorie adjustment:	MINUS	500	kJ	
Energy needed for growth:	8 kJ/kg/d			
BMR:	6351.323664 kJ			
Total Energy:	7636.229997 kJ			
				Continue

Figure 6-5: Automatic daily energy requirement calculation object

The user interface requires the user to enter the gender, weight, height, age, and to select the activity level. The system then determines the activity factor, calculates the actual and ideal BMI and based on these values, determines whether the patient is required to lose or gain weight.

The user can select which equation to use for calculation of the required energy, as well as the number of kilojoules to add or deduct from the requirement, based on the BMI. The system uses the patient's age to automatically determine the energy required for growth.

When all the necessary calculations have been made, the required daily energy is shown (see the bottom of Figure 6-5) and the user can advance to the model that calculates the number of exchanges from each group. This is done by clicking on the *Continue* button (Figure 6-5).

Should the user choose to calculate the daily energy requirement manually, she is prompted to enter the value and the program advances automatically to the goal programming model that calculates the number of exchange group selections.

The goal programming model formulated in (6-2) to (6-6) is activated when the system passes the daily energy value to the spreadsheet object shown in Figure 6-6.

VBA code was written and implemented so that the system automatically sets up the model and constraints based on the data acquired from the user. The VBA code includes the commands to solve the model with the default macronutrient distribution when the object is activated. The relevant VBA code is attached as Appendix E.

The user can change the macronutrient distribution if required, by clicking on the *Set macronutrient distribution* button (Figure 6-6). Selecting this option will cause an additional input form to appear that prompts the user for the new macronutrient distribution. Data validation is used to ensure that the total distribution adds up to 100%.

Clicking on the *Continue* button results in the display of the goal programming solution which shows the macronutrient contribution in grams, the total energy for which the information is relevant, and the number of selections that should be made from each exchange group. This is shown in Figure 6-7.

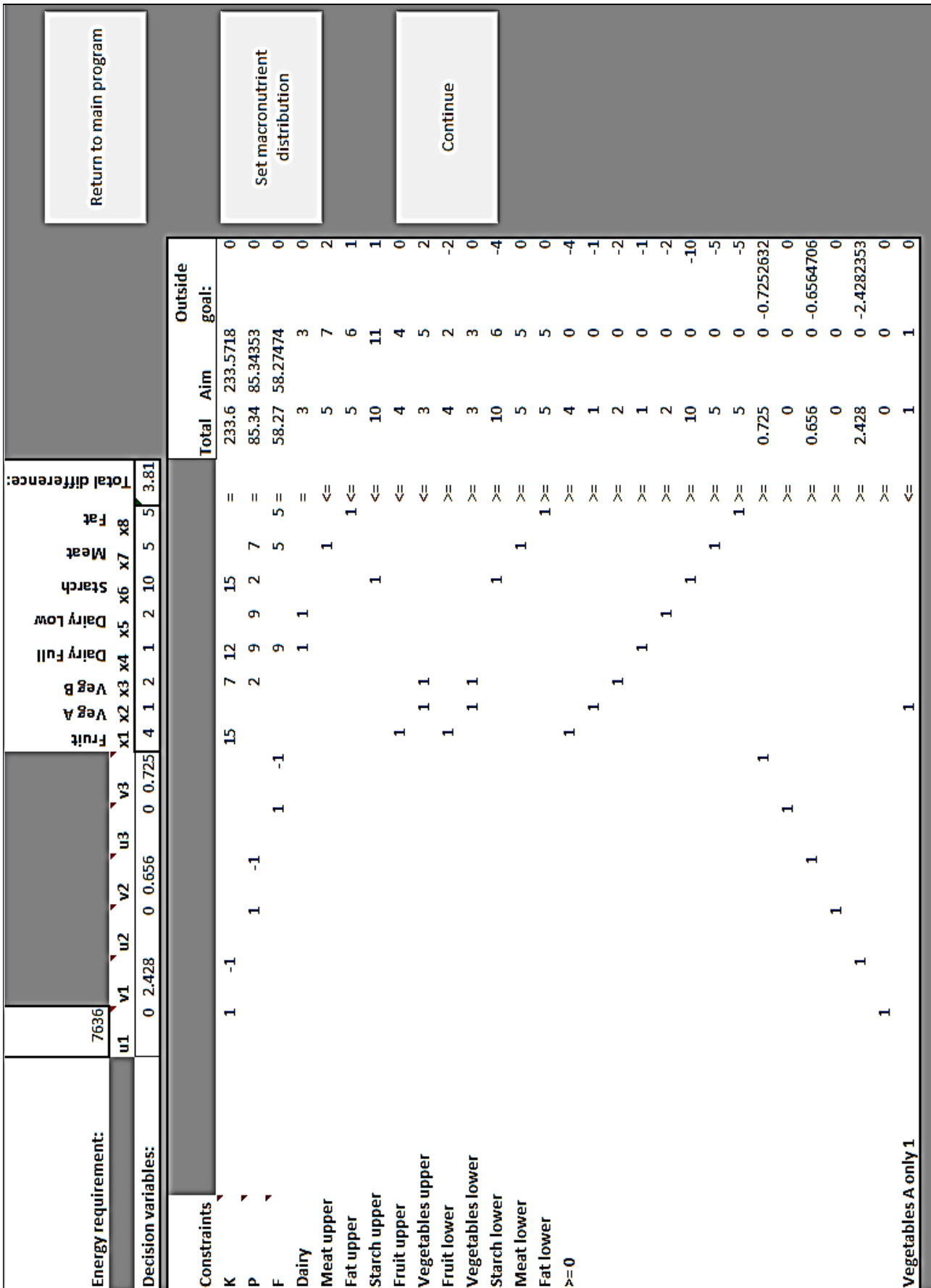


Figure 6-6: Output of the goal programming model that calculates the number of exchanges from each group

The screenshot shows a window titled "Calculated results" with a close button (X) in the top right corner. The main content is a "Results Summary" box. In the top right of this box, there is a "Nutritional information:" section with the following values: Carbohydrates (g): 233.57, Protein (g): 85.34, Fat (g) 58.27, and Total Energy (k): 7636. Below this, a section titled "Recommended exchanges from each group:" lists eight food groups, each with a quantity and a "Set Preferences" button. The groups are: Fruit: 4, Vegetable group A: 1, Vegetable group B: 2, Full cream dairy: 1, Low fat dairy: 2, Starch: 10, Meat: 5, and Fat: 5. An "OK" button is located to the right of the "Set Preferences" buttons.

Figure 6-7: Results of the goal programming model

For each of the exchange groups, the user can set her preferences by selecting the *Set Preferences* button that corresponds to that particular group. A separate spreadsheet object is used to enter the user preferences for each group. As an example, the preference input sheet for fruit is shown in Figure 6-8.

The program uses message boxes to ask the user to enter her preference rating for each item and to enter the value of 100 if she is allergic for any item in the list. This preference rating process is repeated for each exchange group, after which the program advances to the next phase of the system.

Before the eating-plan is generated, the user is asked to set her priority rating for cost or preference, by means of the user form shown in Figure 6-9.

After entering the priority, the multi-objective linear programming model is activated that makes the selections of food types from each group that conforms to all the requirements. The food preferences, cost information, and weights for the priorities of the cost vs. preferences objectives, are passed as arguments to the model.

Fruit	Preference Rating:
Strawberry	2
Apple	1
Apple juice	3
Dates	2
Grape	2
Granadilla	4
Cherry	5
Guava	3
Orange	2
Litchi	5
Mango	5
Pawpaw	1
Pear	2
Peach	1
Grapefruit	5

Figure 6-8: Setting the user preferences for fruit

Figure 6-9: User form to set the user's cost vs. food preference priority

After solving the multi-objective linear programming model, the program presents the user with the output which represents a one-day eating-plan. An example output is shown in Figure 6-10.

The resulting eating-plan is compiled and laid out according to the categories in which the food items were listed. The total CHO, protein and fat contributions of the diet are displayed as well as the total cost of the eating-plan. On the eating-plan, statistics are also displayed that show the total cost of the plan, as well as a percentage value which indicates the level at which the eating-plan satisfies the patient's food preferences. This preference is determined by counting the number of

food items for which the user indicated preference levels and calculating the percentage level of those included in the resulting eating-plan.

Personalised diet for a total of 7636 kJ		
Total carbohydrates: 233.57g	Total cost of this diet: R 24.56	
Total protein: 85.34g	Personal food preference satisfied: 65.71%	
Total fat: 58.27g		
Breakfast	Portion size	
Egg (boiled)	50 g	Protein
Weet-bix	25 g	Starch
Main meal		
Cabbage	125 ml	Vegetable
Beetroot	125 ml	Vegetable
Carrots	125 ml	Vegetable
Beef, lamb, pork, poultry	30 g	Protein
Potatoes (cooked)	100 g	Starch
Sweet potatoes (cooked)	60 g	Starch
Dried beans	25 g	Starch
Snacks		
Low fat milk powder	60 ml / 35 g	Low fat dairy
Apple	80 g	Fruit
Apple juice	90 g	Fruit
Orange	100 g	Fruit
Peanut butter	30 g + 2 fat exchange	Protein
Raisin bread	1 slice	Starch
Cream crackers	3 ea	Starch
Rusks	20 g	Starch
Pop corn	25 g	Starch
Any Time		
Full cream milk	250 ml	Full cream dairy
Low fat milk	250 ml	Low fat dairy
Pawpaw	100 g	Fruit
Cheese, cheddar, sweetmilk	30 g	Protein
Viennas	50 g / 1 medium	Protein
Cake flour	20 g	Starch
Maize flour	25 g	Starch
Canola	1 tsp	Oil
Sunflower oil	1 tsp	Oil
Mayonaise	1 tsp	Oil

Figure 6-10: Example output of a one-day eating-plan generated by the expert system

6.4. Testing and evaluation of the system

The evaluation process entails the following steps:

- Demonstrating the effect of the weights representing the priorities of the cost vs. preference goals;
- Applying the consultation sessions to the program;

- Comparing the eating-plans generated by the expert system with the diets received from the dietician.

6.4.1. System application

To ensure the anonymity of the volunteers, they will be referred to as subjects 1 through 6. For the purpose of demonstrating the effect of the cost vs. weight priority on the program, the system was used to replicate the consultation session of subject 6 because the variety of foods for which personal preferences had been indicated was the highest.

Two sets of eating-plans were generated by the program. The first set was generated using the automatic daily energy requirement calculation capability of the system in conjunction with five different pairs of cost/preference weights. The second set was generated by manually entering the exact daily energy requirement as used by the dietician.

Subject 6 indicated that she avoids olives, fat, mango, and raw onions. Her favourite food items include numerous variants from most of the available food lists. Her preferences for the purpose of applying them in the models available in the system are set as indicated in Table 6-6.

Food item	Preference
Olives Pineapple Melon Watermelon Avocado Naartjie	Not in the food list
Mango	5
Full cream milk Low fat milk Onion	4
Peas Meat and fish Cheese Eggs Cake (flour)	2

Yoghurt Tomato Strawberry Apple Apple juice Litchi Pawpaw Bread Peanut butter Porridge and cereal Pasta Rice Potato or mash Biscuits	1
All other food items	3

Table 6-6: Setting the food preferences of subject 6

The information required for the system to calculate the required daily energy is shown in Table 6-7.

Variable	Value
Gender	Female
Age	18 years
Weight	61.9 kg
Height	171 cm
Activity level	Sport 3-5 times per week
Goal	Maintain current weight

Table 6-7: Information needed for required daily energy calculation

The process was repeated for the following cost and preference priority weights:

- Weight for cost = 4, weight for preference = 0;
- Weight for cost = 3, weight for preference = 1;
- Weight for cost = 2, weight for preference = 2;
- Weight for cost = 1, weight for preference = 3;
- Weight for cost = 0, weight for preference = 4.

The system calculated the total required energy at a level of 6164 kJ and compiled the first set of eating-plans accordingly. All eating-plans produced by the system are attached as Appendix F.

When the priority for cost is high it means that probably not all preferred foods would be included in the eating-plan because precedence is given to lowering the total cost. Similarly, when the priority for preference is high it means that the total cost should increase because it has a low priority and would be disregarded during the calculation process. The statistics for the first set of eating-plans are summarized in Table 6-8.

Personalised diet for a total of 6164 kJ			
Total carbohydrates: 188.55 g			
Total protein: 68.89 g			
Total fat: 47.04 g			
Cost	Pref.	Total cost of the diet:	Food Preference satisfied:
4	0	R 28.76	79.17%
3	1	R 29.41	82.14%
2	2	R 30.45	82.81%
1	3	R 32.59	84.72%
0	4	R 53.14	88.16%

Table 6-8: Statistics for the 6164 kJ eating-plan generated by the expert system

Table 6-8 confirms the expectation that the total cost increases as the priority for food preference increases, and that the percentage of food preference satisfied decreases as the priority for lower cost increases. Therefore when the priority of food preference is at level 4 and that of total cost is zero, the cost of the resulting eating-plan is not considered when selecting foods, but those foods with the highest preferences are.

During the consultation session with the dietician, subject 6 provided some information regarding her personal food preferences in terms of likes and dislikes. However, not all of the food items on the list were actually presented to the volunteer during the session and the preference satisfaction is therefore calculated only for the items that had been mentioned. For this reason, the preference satisfaction percentage varies only by 9.02% between the least cost eating-plan and the highest preference eating-plan.

The price of the resulting eating-plan increases dramatically when the priority level of the total cost decreases. This phenomenon is due to the program completely

discounting the food prices when the cost weight is set to 0. The items with the highest preference ratings are selected without considering the effect their prices would have on the total cost of the diet. Figure 6-11 shows the effect of the weight changes in terms of cost.

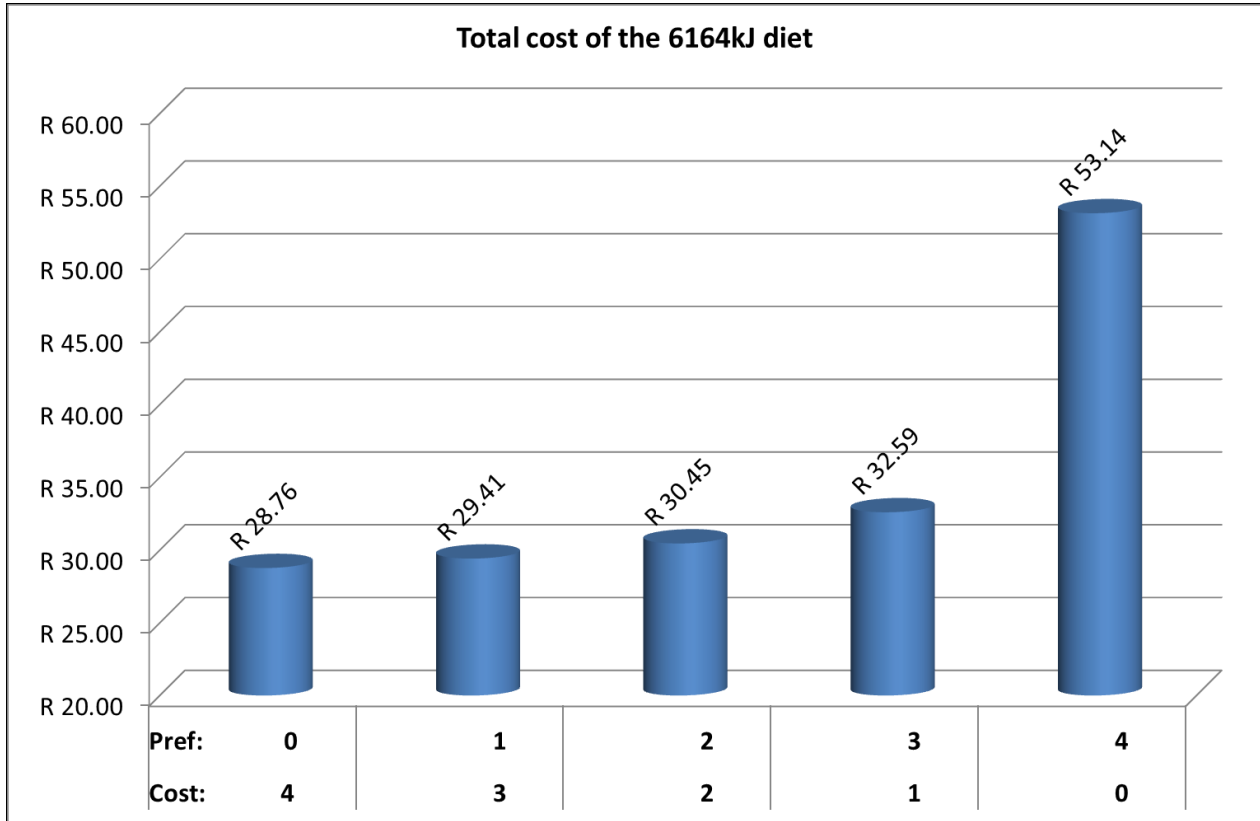


Figure 6-11: The effect of the changes in preference and cost weights on the prices of the resulting eating-plans for a 6164 kJ daily energy requirement

Figure 6-12 shows the effect of the weight changes in terms of preference satisfaction. The percentage by which the user's food preference is satisfied increases as its priority level increases and that of the cost decreases.

Upon examination of the eating-plan provided by the dietician for subject 6, it was observed that the daily energy requirement for which the plan was compiled was slightly lower than that generated by the expert system.

For better comparison purposes a second set of eating-plans was generated utilising the system for the energy value used by the dietician by using the manual energy input capability in the system. The second set of eating-plans were therefore generated with a predetermined input energy value of 5700 kJ.



Figure 6-12: The effect of the changes in preference and cost weights on the preference satisfaction of the resulting eating-plans for a 6164 kJ daily energy requirement

The statistics for the second set of eating-plans are summarized in Table 6-9. Note the increase in the price as the cost priority decreases. The low variation in the preference rating is again rationalised by the many food items in the exchange lists that have an exact preference rating of 3.

Personalised diet for a total of 5700 kJ			
Total carbohydrates: 174.35g			
Total protein: 63.71g			
Total fat: 43.5g			
Cost	Pref.	Total cost of the diet:	Food Preference satisfied:
4	0	R 27.95	79.17%
3	1	R 28.96	82.14%
2	2	R 30.03	82.81%
1	3	R 30.99	83.82%
0	4	R 52.94	88.89%

Table 6-9: Statistics for the 5700 kJ eating-plan generated by the expert system

The changes in total cost and food preference satisfied for the 5700 kJ eating-plan resembles that of the 6164 kJ.

6.4.2. Application of the expert system to the consultation sessions with the dietician

Each of the consultation sessions with subjects 1 to 5 was applied to the expert system. The manual energy calculation capability of the program was used to enter the energy level in each of the cases, so that the resulting diets could be compared.

The consultation sessions were replicated by executing the expert system and emulating each of the volunteers' replies to the relevant questions. Each of the generated eating-plans was compared to the eating-plans received from the dietician. One of the major differences between the plans is that the program generates a plan for one day, while the dietician's plan provides several options for each meal which can be applied over a longer period of time. Another difference is that the program does not allow more than one selection of the same food item, which means that if for example a person prefers full cream milk above any other full cream dairy product and it is the cheapest, it would be selected only once for the eating-plan. However, it is possible that a person would like to consume all of the full cream dairy exchanges as milk. Therefore the food items in the output eating-plans were compared to the food items in the dietician's plans according to the following criteria:

1. Food items that appear in both eating-plans;
2. Food items that originate from the same exchange group as other items that have not yet been listed in criteria 1;
3. Food items that appear in the computer generated eating-plan and not in the dietician's plan.

Due to the fact that the dietician's eating-plans provide several different options to choose from for each meal, it would serve no purpose to list the foods that appear in the dietician generated plans and not in the expert system generated plans.

The results for the comparisons are summarized in Table 6-10.

	Foods in both plans	Foods in both plans from the same exchange groups	Foods only in the expert system generated plan
Subject 1	Egg (boiled) Muesli Asparagus Beetroot Gem squash Carrots Beef/lamb/pork/ poultry/mince Potato (cooked) Apple Apple juice Orange Provitas Full cream milk Low fat milk Pawpaw Cheese (cheddar/ sweet-milk) Viennas Bread (whole wheat/brown/white) Canola oil Sunflower oil	Peanut butter Buttermilk from low fat Cake flour Maize flour Mayonnaise	Low fat milk powder
Subject 2	Egg (boiled) Porridge (cooked) Spinach Beetroot Carrots Fish (white) Potato (cooked) Sweet potatoes (cooked) Apple Orange Pear Low fat milk Pawpaw Cheese (cheddar, sweet-milk) Canola oil Sunflower oil Pop corn	Peanut butter Full cream milk Low fat milk Viennas Cake flour Maize flour Mayonnaise	Low fat milk powder Buttermilk from low fat
Subject 3	Egg (boiled) Asparagus Beetroot Gem squash Carrots Beef/lamb/pork/ poultry/mince Potato (cooked) Apple Orange Low fat milk Cheese (cheddar, sweet-milk) Canola oil	Wheet-bix Peanut butter Full cream milk Low fat milk powder Viennas Cake flour Maize flour	Buttermilk from low fat Low fat milk
Subject 4	Egg (boiled) Porridge (cooked) Cabbage Beetroot Beef/lamb/pork/ poultry/mince Potato (cooked) Apple Apple juice Orange	Low fat milk powder Full cream milk Viennas Cake flour Maize flour Peanut butter Sunflower oil Mayonnaise	Buttermilk from low fat

	Low fat milk Cheese (cheddar, sweet-milk) Bread (whole wheat/brown/white) Pop corn Canola oil		
Subject 5	All bran Asparagus Beetroot Gem squash Carrots Beef/lamb/pork/ poultry/mince Potato (cooked) Sweet potatoes (cooked) Apple Orange Pear Cottage cheese (not cream cheese) Pop corn Low fat milk Pawpaw Cheese (cheddar, sweet-milk) Viennas Canola oil	Peanut butter Cake flour Maize flour	Low fat milk powder Full cream milk Buttermilk from low fat
Subject 6	Egg (boiled) Beetroot Gem squash Carrots Beef/lamb/pork/ poultry/mince Fish (white) Rice (white/brown, cooked) Potato (cooked) Apple Apple juice Orange Full cream milk Low fat milk Tomato Pawpaw Cheese (cheddar, sweet-milk) Bread (whole wheat/brown/white) Sunflower oil Mayonnaise	Porridge (cooked) Low fat milk powder Buttermilk from low fat Cake flour Maize flour Canola oil	Peanut butter

Table 6-10: Comparison of the food items in the eating-plans

One of the concerns the dietician had prior to development of the system was that when creating an eating-plan for an individual, the dietician tends to choose foods for the eating-plan out of habit rather than considering the patient's food preferences. To determine the variety in food items that the expert system provides, the following data was obtained from the eating-plans for each subject:

- The similarity which gives a percentage of the total number of foods that match in both plans;

- The variation which gives a percentage of the foods in the dietician's plan that differs from foods in the system's plan but comes from the same exchange group; and
- The exception which provides a percentage of the number of foods that only appears in the system generated plans and not in the dietician's plan.

	Number of food matches	Number of exchange group matches	Number of exceptions in system generated plans	Total food items in each plan	% similarity between the plans	% variation provided by the program	% exceptions
Subject 1	20	5	1	26	76.92%	19.23%	3.85%
Subject 2	17	7	2	26	65.38%	26.92%	7.69%
Subject 3	12	7	2	21	57.14%	33.33%	9.52%
Subject 4	14	8	1	23	60.87%	34.78%	4.35%
Subject 5	18	3	3	24	75.00%	12.50%	12.50%
Subject 6	19	6	1	26	73.08%	23.08%	3.85%
Average:					68.07%	24.97%	6.96%

Table 6-11: The number of food matches, group matches and exceptions between the eating-plans

The data is shown in Table 6-11. For the total number of food items generated by the expert system of 146, the average similarity between the plans was 68.07% and the variation in food items provided by the program was 24.97%, with an average of 6.96% exceptions.

6.5. Chapter summary

Chapter 6 concluded the discussion of the knowledge engineering process for the expert system created in this study.

The practical design and implementation of the system were described in section 6.2 by providing a detailed description of knowledge classification and detailed design stages. The formulation and implementation of the goal programming and multi-objective linear programming models were discussed in detail.

Section 6.3 described how the system was used to generate eating-plans which were compared to the eating-plans generated by the dietician. Concluding remarks, exceptions and recommendations were presented in section 6.4.

Chapter 7 will present a general discussion on the research performed and an overview of the expert system that was created. A summary will be given on the results obtained by applying the system to the six volunteer case studies, followed by some concluding remarks.

Chapter 7 – Summary and results

7.1. Introduction

The purpose of this chapter is to present a general discussion on the research performed and to provide an overview of the expert system that was created.

A brief summary will be given on the motivation and development of the expert system in section 7.2. A short overview of the application of the system and its performance is presented in section 7.3. Points to consider regarding contributions to the linear programming and expert system research fields will be discussed in section 7.4. General concluding remarks will be made in section 7.5.

7.2. Development of the expert system

The objective of this study was to evaluate the feasibility of using expert system technology in conjunction with linear programming techniques to solve the diet problem for a typical female teenager in South Africa. The diet problem involves finding an optimal eating-plan for a certain individual that conforms to the daily energy and nutritional requirements for that person.

Numerous linear programming solutions have been developed in the past with the purpose of solving some sort of adaptation of the diet problem.

- Anderson and Earle (1983) presented a goal programming model that solves the problem of an imbalance in the amount of nutrients provided by a certain diet.
- A two-stage linear programming solution process was developed to determine a least cost nutritionally balanced diet for Malaysian households (Hamzah *et al.*, 2011).
- Lancaster (1992) reviewed the transformation of the diet problem from a manual method to the electronic program CAMP (Computer Assisted Menu Planning) in which mathematical programming techniques had been used to implement the rules and constraints of the diet problem into an executable, usable software program.

Expert systems have also been employed to solve health-related problems.

- PRISM 3.0 was developed to generate reasonable meal plans (Sterling *et al.*, 1996).
- The Chinese Medical Diagnostic System (CMDS) is an expert system that was developed to aid in diagnosing digestive system diseases (Huang and Chen, 2007).
- The Nutritional Care Process and Model (NCPM) was developed to evaluate the nutrition quality of certain regions in countries like Taiwan (Chen *et al.*, 2012).

To test the hypothesis of this study, an expert system was created which incorporated linear programming techniques to solve the diet problem. The system was created by means of a knowledge engineering process which involved consulting a number of human experts as well as various published works and developing the components essential to an expert system. The system was developed and implemented on a Microsoft Excel platform and consisted of an inference engine, knowledge base, working memory and user interface.

Due to the heuristic nature of the knowledge needed to make the necessary inferences, linear programming techniques were introduced to facilitate certain subsections of the problem. After acquiring some personal information and data from the user, the system used a goal programming model to calculate the number of foods that should be chosen from each of the major food groups so that the nutritional requirement for the user would be satisfied.

Concerns regarding food intake which were identified during knowledge engineering include:

- The total cost of an eating-plan is a major consideration which can cause teenagers to select unhealthy combinations of foods to consume; and
- A teenager would follow an eating-plan that consists of food items which she actually prefers rather than one with foods which she hates.

The system required the user to rate numerous food items from all the different food groups according to personal preference. The number of food selections from each food group as calculated by the goal programming model was used by a multi-objective linear programming model to determine the combination of specific food

items from the food groups that represented the lowest cost and/or highest user food preference combination. The user was presented with the option of setting the priority of food cost or personal food preference when generating the eating-plan. These options were catered for by using weights in the model which placed a penalty on the relevant least cost or highest preference priority. The combination of foods was presented in the form of a one day eating-plan that also provided the user with the total carbohydrate-, protein- and fat intake resulting from the eating-plan. The total cost and personal food preference satisfaction percentage was also displayed on the eating-plan.

7.3. System performance

The expert system was used to generate eating-plans for the case studies of the six volunteers who partook in the knowledge engineering process.

The case study of subject 6 was selected based on the variety of food items discussed during the consultation session, and applied to the expert system, with five different priority levels for total cost of the eating-plan and personal food preference of the user. The results confirmed that the cost of the resulting eating-plan increased when the cost priority decreased and the personal food preference satisfaction increased and vice versa.

One-day eating-plans were generated for each of the case studies using equal cost- and food preference priorities so that the food items in the expert system generated eating-plans and the dietician generated eating-plans could be compared. An average of 68.07% of food items appeared in both plans while the expert system plans provided an average variety in food items of 24.97%. The percentage of 6.96% of the items in the expert system plans did not appear in the dietician's plans.

7.4. Contribution to linear programming- and expert system research

This research study evaluated the feasibility of combining two problem-solving techniques from different research fields. The following contributions to the linear programming- and expert system fields could therefore be considered:

- The study proved that expert system technology can be used in conjunction with mathematical programming techniques which, according to literature has not been done before to solve the diet problem.
- Expert system technology was used successfully to obtain the necessary data to generate constraints unique to an individual user, for the linear programming models. The linear programming models were effectively used to represent the rules of the expert system.
- Standard applications which use linear programming to solve the diet problem either determine the combination of foods that would provide a least cost diet, or they determine a diet that consists of a user's favourite food items. The system developed in this study uses multi-objective linear programming to consider both of these objectives simultaneously.
- The developed expert system generates eating-plans that are unique to the South African environment. The food list used to generate the eating-plans is comprised of items frequently consumed by high-school learners in the Northwest province in South Africa. Although related research had been done previously, the resulting solutions provided for foods often consumed in other countries. As far as it could be determined this is the first time a system has been developed specifically for South African foods.
- An alternative method for incorporating the user food preference rating was proposed and implemented successfully. Instead of setting a maximum food preference at the highest available level, the system prompts the user to set a highest preference at a level of 1 rather than 5. This enabled the system to minimise both of the objectives, cost and preference.
- The software used for developing and executing the system can be acquired easily and the program performed its functions in an acceptable time. The system was implemented using the object structures of MS Excel and combining them with Visual Basic for Applications (VBA) programming code. The VBA code was developed to control the translation of the data gathered by the expert system user interface into the format that the solver in Excel required as well as the execution of the solver component.

- It was shown successfully how the weighted priorities for cost and preference effected changes in the total costs of eating-plans generated by the system, as well as in the percentage of food preferences that were satisfied.
- It was also proved that goal programming and multi-objective linear programming could be combined to successfully represent the inference engine of an expert system. The output of the goal programming model which performed calculations in one phase of the program was used as input data for the multi-objective linear programming model to perform further calculations and to provide the resulting solution. Similar combined models have not been used in the field of dietetics before.
- As far as could be established, utilising goal programming for the purpose of determining a combination of food exchanges that provides a macronutrient contribution that is as close as possible to a certain calculated ratio, has not been done before.
- Existing prior research include among others, related solutions that generate eating-plans either based on a calculated daily energy requirement, a manually entered energy value, or a number of food items for which user preferences are known. The system developed in this study however, presented some flexibility in terms of allowing the user to choose if the system should calculate the required daily energy value or, if prompted by experience, a dietician could simply enter the energy requirement and let the system generate the eating-plan. It further also considered the food preferences of the user when generating the eating-plan.
- The program successfully provided further flexibility by the implementation of weighted priorities which allowed the user to set the importance of having an eating-plan at the lowest cost, one that conforms to food preferences, or one that considered both of these objectives in a relationship controlled by the user.

The next section presents some general concluding remarks relating to the study.

7.5. Concluding remarks

The expert system used a goal programming- and a multi-objective linear programming model to deliver eating-plans that conformed to the nutritional requirements for a healthy diet of a specific individual.

- It is concluded that combining expert system technology with the advantages of linear programming techniques, is feasible.
- The system calculated the total cost of the generated diets by considering the serving sizes of the items on the list. However, most of the food items are sold in larger quantities than one serving size and it could dramatically influence the total cost of a diet if the user has most of the required food items in stock.
- The expert system generated eating-plans that could only include one serving of each food item at a time. If a user was to rather choose to have two or more of the same food items that are cheaper or more expensive than the other food items, the total cost of the diet could change drastically. The capability of allowing the system to cater for more than one serving per food item could be considered for future work.
- During the consultation sessions, the volunteers were asked to name their favourite and least favourite foods. Some of the girls were complacent in simply naming one or two food items which meant that the preference satisfaction percentage had been calculated for a very small number of foods. The expert system presented the user with all of the food items in the respective exchange lists and asked the user to enter a personal preference for each item. Therefore, the variation in preference percentage levels of some of the sessions when applied to the system is very small.

7.6. Chapter summary

Chapter 7 started by providing a summarized overview of the reasons for developing the system as well as a short description of the function of the system. The performance of the system was outlined in section 7.3 and some possible contributions to the linear programming and expert system research fields were discussed in section 7.4. The chapter finished by presenting some general concluding remarks in section 7.5.

Chapter 8 will present the objectives of the study and discuss how they were achieved and addressed in this dissertation, present problems experienced during the course of this study, and discuss matters arising from this study which could present opportunities for further research.

Chapter 8 – Conclusions

8.1. Introduction

Chapter 8 presents final comments and concluding remarks on this study. The objectives of the study and how they were achieved will be summarised. New opportunities for further study that arose during the course of this study will also be outlined.

8.2. Objectives of the study

Chapter 1 stated that the primary objective of this study was to determine the feasibility of combining the heuristic inference capabilities of an expert system with the optimizing advantages that linear programming provides. To accomplish this, a list of secondary research objectives was defined in order to achieve the primary objective.

They included:

- Providing an overview of the mathematical programming techniques referred to and applied in this study;
- Providing an overview of expert systems concepts, principles, and development;
- Performing an exploratory investigation into the use of linear programming techniques and expert systems in solving health-related problems and the presence of such problems in South Africa;
- Performing the initial stages of knowledge engineering to create a theoretical design of an expert system for solving the relevant diet problem;
- Performing the development and evaluation stages of knowledge engineering; and
- Interpreting the results, stating the possible contributions resulting from the system and making appropriate recommendations.

A summary of how these objectives were achieved and addressed follows below.

Providing an overview of the mathematical programming techniques referred to and applied in this study.

This goal was addressed in chapter 2 and achieved by explaining the concepts and definitions of mathematical programming, providing basic formulations and presenting solution methods by means of examples. Linear programming formulation and 2-dimensional solutions as well as special cases and sensitivity analysis were discussed in section 2.2. Integer linear programming was addressed and explained in section 2.3, goal programming in section 2.4, and multi-objective programming in section 2.5.

Providing an overview of expert systems concepts, principles, and development.

This goal was achieved and an overview was provided in chapter 3 which started by providing the basic definitions and concepts about expert systems in section 3.2. Knowledge representation techniques were briefly outlined in section 3.3, followed by a concise layout of the components of an expert system in section 3.4. The knowledge engineering process was discussed in section 3.5 by presenting system preliminary planning and stages of the expert system development life cycle.

Performing an exploratory investigation into the use of linear programming techniques and expert systems in solving health-related problems and the presence of such problems in South Africa.

The goal was achieved by way of completing a study on the use of linear programming techniques to solve variations of the diet problem (section 4.4.1) and the use of expert systems in solving health-related problems (section 4.4.2). Section 4.5 provided a summary of research done on health in South Africa and elaborated in section 4.5.1 on obesity and in section 4.5.2 on dieting and abnormal weight control behaviour in the country.

Performing the initial stages of knowledge engineering to create a theoretical design of an expert system for solving the relevant diet problem.

This goal was achieved by performing the preliminary planning (section 5.3) as well as the knowledge source identification and selection (addressed in section 5.4.1), and knowledge acquisition (section 5.4.2), analysis (section 5.4.3) and extraction tasks (section 5.4.4) of the knowledge definition stage. Chapter 5 presented a theoretical design of the expert system created in this study by discussing how the tasks were performed.

Performing the development and evaluation stages of knowledge engineering.

The development stages were achieved and addressed in chapter 6 which explained how the knowledge design stage was performed by discussing knowledge classification task in section 6.2.1, and the detailed design task in section 6.2.2. The execution of the code and checkout stage was presented in section 6.3. The evaluation stage was achieved and addressed in section 6.4 by presenting results acquired through application of the system to the example cases acquired through the knowledge acquisition task.

Interpreting the results, stating the possible contributions resulting from the system and making appropriate recommendations.

This goal was achieved and addressed in chapter 7 where the system performance was summarized in section 7.3 followed by a discussion on how the study contributes to the linear programming and expert system research fields in section 7.4.

To summarize, all objectives set forth in Chapter 1 were achieved and addressed. Based on the results and discussion presented in chapter 7, it was concluded that:

- It is feasible to combine expert system technology with linear programming techniques;
- Using input information obtained from the user, the thus created expert system performs the basic calculations to compute the daily required energy intake whilst still providing the user with the option to enter the value manually;
- The expert system successfully generates an eating-plan suitable for a female teenager in the South African environment;
- The eating-plans generated provided the user with the choice of setting a priority for the cost and food preferences included in the resulting eating-plan; and
- During the knowledge acquisition phase it was revealed that a major concern of the dietician was that selecting food types for an eating-plan is a habit-forming exercise and the plans generated by the expert system presented a variation in food types of 24.97%.

8.3. Problems experienced

The limited number of food items for which the volunteers presented preference levels in the consultation sessions meant that the resulting preference percentage was calculated for a very small number of items. The volunteers seemed slightly intimidated by the fact that the sessions were being recorded which could account for their reluctance to elaborate and offer too much information. The system provided for this by prompting the input of user preferences of all of the food items in the knowledge base at runtime.

The group of volunteers consisted of six white girls although a demographic representation of the South African population would have been preferred. During the course of the study however, it became clear that the ethnic race of the participant or end-user in the case of the expert system, does not play a role in determining the nutritional requirements for the teenage girl. The items on the food list can be expanded to include foods frequently consumed by girls other than whites. This issue is addressed in section 8.4.

Some food items on the list are only available during certain seasons of the year. For this reason, the prices of all the items could not be obtained during system development. Prices also fluctuate regularly for many different social and economic reasons. This problem will also be addressed in section 8.4.

Due to budget constraints, only a limited number of volunteers could be asked to participate in consultation sessions with the dietician. The sessions were therefore used as a means to gather knowledge on the eating-plan compilation process whereas the list of foods frequently consumed by teenage girls was obtained from an independent research project.

The standard form of the solver used to execute the linear programming models in MS Excel is an input form which halts execution to wait for user response relating to the linear programming model in memory at runtime. This was accommodated for through the use of Visual Basic for Applications (VBA) code which was used to initialise the solver, set up the model and constraints, execute, and save the solutions of the various respective models.

8.4. Possibilities for further research

The number of food items in the expert system was limited for the purpose of evaluating the feasibility of the system. Increasing the food items in each exchange group would increase the number of items presented to the user for selecting preference levels which means that a greater variety would be available for selection by the program.

The expert system generated one-day eating-plans whereas the plans received from the dietician spanned a seven-day period. One disadvantage to the seven-day plan is that a person can choose which meal to eat but there is no limitation on the number of times that meal may or may not be eaten. The expert system could be expanded by means of more linear programming models, to generate an eating-plan for several days while limiting the consumption of certain food items.

Due to many diverse factors like inflation, drought, economy, politics, etc. the prices of food items change frequently. The expert system created for the purpose of this study used price data that was obtained from electronic sources. The price data remained fixed throughout the development process and subsequent evaluations. For future work however, a food database that is dynamically linked to reputable sources of price data could be considered.

8.5. Chapter summary

Chapter 8 is the final chapter of this study. This chapter presented a summary of the initial objectives and how the objectives were achieved and addressed. In conclusion, the problems experienced and possible future research opportunities were outlined.

Appendix A - Letter to the parents



NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
POTCHEFSTROOMKAMPUS

24 Julie 2012

Geagte Ouer,

Soos reeds vroeër die jaar aan die leerlinge bekend gemaak, is ons besig met 'n navorsing studie by die NWU Rekenaarwetenskap in die toepassingsarea van dieetkunde. U dogter het aangedui dat sy bereid is om deel te neem as vrywilliger. Hiermee net 'n bietjie meer inligting.

Wat ons van elke vrywilliger vra is om een besoek af te lê by Dr. Driekie Rankin. Sy is 'n gekwalifiseerde dieetkundige wat haar eie praktyk in Potchefstroom bedryf. Die NWU onderneem om die koste van hierdie afspraak te dek. In ruil hiervoor vra ons die geleentheid om die gespreksessie op band op te neem (klank). Die doel hiervan is om brondata rondom die berekening van 'n dieet vir 'n individu in te samel, bv. ouderdom, geslag, aktiwiteitsvlak, ens. Geen persoonlike inligting sal gebruik of bekendgestel word nie. Elke vrywilliger ontvang na afloop van die afspraak haar individueel berekende dieet en is dan onder geen verdere verpligting teenoor die NWU nie. Data wat ingesamel is sal gebruik word vir die ontwikkeling van 'n rekenaar stelsel.

Voltooi asb. die skeurstrokie en stuur saam na Dr. Rankin se spreekkamer te Lupinestr. 4, Grimbeeckpark (straat wat grens aan die gholfbaan).

Dankie by voorbaat vir u hulp in hierdie verband.

Vriendelik die uwe,

Mev. Annette van der Merwe
Rekenaarwetenskap en
Inligtingstelsels
NWU
Tel: 018 299 2541
www.nwu.ac.za

Hiermee gee ek, _____, ouer/voog van _____ toestemming dat sy aan die dieet-navorsingsprojek wat deur die NWU bedryf word mag deelneem. Ek bevestig dat die NWU verantwoordelik is vir die vereffening van die rekening van een besoek aan dr. Driekie Rankin.

Handtekening

Datum

Appendix B – Excerpt of the transcriptions of the consultation sessions summary

Table B-1 presents an excerpt of the summarizing table of the consultations sessions between the dietician and the six volunteers. The “*Primary question*” column represents the questions posed by the dietician whereas the “*Related question*” column represents questions that were asked if the answer of the volunteer prompted further enquiry. “*Dietician comments*” represents the comments made by the dietician regarding a question and the “*Subject 1*” to “*Subject 6*” columns represent the responses of subjects 1 to 6 to the posed questions. Not all questions were repeated for every participant. For the full Table, refer to the enclosed disk.

Primary Questions	Related Questions	Dietician comments	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6
How old are you?			17	18	18	16	16	17
What is your reason for coming?		If you want to lose weight I will deduct kJ from total energy requirement.	Wants to lose weight and eat right. Am very active but not sure that I'm eating right.	Lose weight.	Want to maintain current weight for the matric farewell.	Lose weight.	I'm not a sports person but eat what I want, do not gain weight, and live in hostel. Am just curious. Can gain weight but not too much.	Just for the experience, I want to eat healthy.
Where do you live?				Parents	Home	Home	Hostel	Parents on the farm
Who cooks?				Self	Mother	Mom	Hostel / self	Parents
		Dietician explains the patient can expect a healthy meal plan. There are a wide variety of ways to lose weight; everybody has their own way of seeing things. Most important thing is to eat healthy. It's important to get in the habit of eating healthy: when you get older it will be nothing new.						
Do you suffer from constipation?			N	N	Y	N	Y, stress related - spastic colon especially when singing opera.	N
	How frequently?		Not relevant.	Not relevant.	Daily, when I eat bread / pap (porridge).	Not relevant.	When I'm stressed, it remains for about 6 days, I have stress epilepsy.	Not relevant.
		Very unhealthy, the colon is basically the body's garbage bin						

Table B-1: Summary of the transcriptions of the consultation sessions

Appendix C – Diet for subject 6 as received from the dietician



Dr Driekie Rankin, Geregistreerde Dieetkundiges (SA)

BSc Dieetkunde, MSc. Dieetkunde, PhD Dieetkunde

PR: 084 000 0237353

Eetplan: [REDACTED]

Tipe eetplan: Gesonde dieet vir gewigsverlies

(5700 kJ, 52% CHO, 19% Prot, 29% Vet)

Ontbyt (voor skool)

Voorbeeld 1	Voorbeeld 2	Voorbeeld 3
1 k Oatspap (Of gewone Oatspap, of Instant Oats van Heartland of Woolies) OF 1 k Mieliepap – effens afgekoel	½ k Pronutro (whole-wheat of applebake) OF ½ k Meusli	2 stukkies Lae GI/ volgraan beskuit
125 ml lae-vet melk OF 125 ml (½ k) lae-vet jogurt		
Voorbeeld 4	Voorbeeld 5	Voorbeeld 6
1 k All Bran Flakes gemeng met Hi-Fibre bran	1 k Strawberry Pops OF 1 k Otees	2 snytjies Lae GI brood (bv Sasko's seeded loaf) MET 1 tlp margarien MET Marmite, bovirol, of heuning
125 ml lae-vet melk OF 125 ml (½ k) lae-vet jogurt		
Voorbeeld 7		
(Beperk hierdie voorbeeld tot een keer per week)		
1 sny Lae GI brood MET 1 eier (gebak/ poached/ roereier) MET 1 repie maer bacon		

Peuselhappie 1 – Hierdie kan of saam met ontbyt of 'n bietjie later deur die oggend geet word

1 x enige vrug waarvan jy hou; bv: 1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels druwe, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie



Peuselhappie 2 – Deur die loop van die oggend/ Pouse

Voorbeeld 1	Voorbeeld 2
<p>1 x enige vrug waarvan jy hou; bv:</p> <p>1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels duiwe, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie</p>	<p>1 sny Lae GI saadbrood</p> <p>OF</p> <p>3 provita's OF 2 Ryevita's</p> <p>*Elke opsie met 1 tlp lite margarien MET Marmite, bovirol of slaagroente (blaarslaai, komkommer en tamatie)</p>
<p>30g stukkie lae-vet kaas (Tussers, Ricotta, Mozzarella, DairyBell InShape Gouda, reduced-fat feta) OF 1 E maaskaas OF 2 tlp lae-vet kaassmeer OF 30g stukkie biltong OF 10cm stukkie droe wors OF 1 vienna (Lean)</p>	



Middagete (Net na skool) Berei 'n maaltyd deur een voorbeeld uit elke kolom te kies:

Kies een van die volgende stysel opsies:		
<ul style="list-style-type: none"> • ½ k pasta – volgraan pasta • 1/3 k rys (Tastic wit rys, bruinrys met lensies, of basmati rys) • 1 medium aartappel – afgekoel met skil • 3 baba aartappels – afgekoel met skil • ½ k aartappel (mashed of aartappel gereg) – effens afgekoel 	<ul style="list-style-type: none"> • ½ k patats – verkieslik nog in skil/ asb nie soet patats nie • 1 medium patat • 1/3 k stywe pap – afgekoel • ½ k stampmielies – gemeng met droë bone • 1 sny brood (Volgraan/ Lae 	<ul style="list-style-type: none"> • 1 klein broodrolletjie – volgraan met saadjies (bv woollies se saadbroodrolletjies • ½ Panini • 1 Wrap • ½ k peulgroente (baked beans, botterbone, lensies, sprouts, hummus)
Kies groente soos volg: 2 x ½ k groente		
<ul style="list-style-type: none"> Blaarslaai Brusselsprout Murgpampoentjie Patty Pans Baby Marrows Sugar snaps Groenboontjies Wortels Ertjies 	<ul style="list-style-type: none"> Uie Aspersies Spinasie Komkommer Gasuie Eiervrug Pampoen Butternut Mengelslaai 	<ul style="list-style-type: none"> Brokkoli Sampioene Kool Tamatie Greenpeppers Argurkies Lemoenpampoentjies Beet Wortelslaai
Kies 4 x 30g van enige van die volgende proteïene		
<ul style="list-style-type: none"> • Maalvleis (Topside) • Lae-vet kaas* (Tussers, Ricotta, DairyBell InShape Gouda, Mozzarella, vet-vrye maaskaas of reduced fat feta kaas) 	<ul style="list-style-type: none"> • Vis (dit kan vars vis, I&J vissies, tuna, of visvingers wees) • Visvingers (1 = 30g) • Rooivleis (1-2x/week) • Wildsvleis • 1 Eier (1 eier = 30g, daarom kan jy 3x van die ander proteïene ook eet) 	<ul style="list-style-type: none"> • Volstruis fillets • Hoender fillet • Lae-vet ham • Biltong – verkieslik wildsbiltong, anders biltong waarvan die vet verwyder is
Kies een van die volgende vet opsies:		
1 tlp botter, 35g se avokadopeer, 1tlp canola olie of 10g se neute		

Of kies een van die volgende voorbeelde.....

Voorbeeld 1	Voorbeeld 2	Voorbeeld 3
½ k Stirfry groente met uitgeloopte ertjies EN ½ k groen slaai	½ k gemengde goente EN ½ k Beet (met kruie oor)	½ k groenboontjies, EN ½ k pampoen met kaneel,
3 baba aartappels – opgekook in skil OF ½ k chips – gebruik verkieslik oven-bake chips	1/3 k basmatiese rys	1 medium patat in die skil gebak met 1tlp maaskaas in
120g bees fillet	120g Vis (Gebak; nie 'gedeepe fry') – kan byvoorbeeld I&J vissies gebruik	120g vark tjoppie (vet verwyder voor gaarmaak)
1 tlp botter, 35g se avokadopeer, 1tlp canola olie of 10g se neute		

Voorbeelde vervolg op die volgende bladsy...



Voorbeelde vervolg vanaf die vorige bladsy....

Voorbeeld 4	Voorbeeld 5	Voorbeeld 6
1 k stirfry groente	Stew/ Kerrie-en-rys: Ongeveer ½ k – 1 k groente saam met die vleis gaargemaak	Tamatie, mushrooms, greenpepper
½ k pasta/ noodles	1/3 k bruinrys gemeng met lensies	1 snye brood (Lae GI)
120g hoender repies (kan vervang met steak of vark repies ook)	120g vleis gaargemaak as 'n stew/ kerrie-en-rys of 'n vleis gereg	1 eier MET 2 repies maer bacon MET 30g lae-vet kaas
1 tlp botter, 35g se avokadopeer, 1tlp canola olie of 10g se neute		
Voorbeeld 7	Voorbeeld 8	
1 k Roasted veggies	1 – 1 ½ k vleis en groente sop MET 1 sny brood OF 3 provita's	
1/3 k bruin rys gemeng met lensies		
120g Maer rooivleis (sirlion, flank steak, tender sirlion, lean mince, skaapvleis waarvan vet verwyder is) (Maak gaar soos gewoonlik, maar verwyder alle sigbare vet)		
1 tlp botter, 35g se avokadopeer, 1tlp canola olie of 10g se neute		

Peuselhappie 3 – deur die loop van die middag

Voorbeeld 1	Voorbeeld 2	Voorbeeld 3
1 x enige vrug waarvan jy hou; bv: 1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels druive, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie	3 Marie Beskuitjies OF 1 x 30g volgraan beskuit OF 3 k Popcorn (tuisgemaak, en verkieslik met wind gespring/ mikrogolf)	1 x CrunchyVite koekie OF 1 x BettaSnack koekie

Peuselhappie 4 – Laat middag (Drink hierdie net na die oefening)

250 ml lae-vet melk OF 200 ml Lae vet jogurt
--



Aandete

Berei 'n maaltyd deur een voorbeeld uit elke kolom te kies:

Kies een van die volgende stysel opsies:		
<ul style="list-style-type: none"> 1 k pasta – volgraan pasta 2/3 k rys (Tastic wit rys, bruinrys met lensies, of basmati rys) 1 groot aartappel – afgekoel met skil 6 baba aartappels – afgekoel met skil 1 k aartappel (mashed of aartappel gereg) – effens 	<ul style="list-style-type: none"> 1 k patats – verkieslik nog in skil/ asb nie soet patats nie 1 groot patat 2/3 k stywe pap – afgekoel 1 k stampmelies – gemeng met droë bone 2 snye brood (volgraan/ Lae 	<ul style="list-style-type: none"> 2 klein broodrolletjies – volgraan met saadjies (bv woollies se saadbroodrolletjies 1 Panini 1 groot Wrap ½ k peulgroente (baked beans, botterbone, lensies, sprouts, hummus)
Kies groente soos volg: 1 x ½ k groente		
Blaarslaai Brusselsprout Murgpampoentjie Patty Pans Baby Marrows Sugar snaps Groenboontjies Wortels Ertjies	Uie Aspersies Spinasie Komkommer Gasuie Eiervrug Pampoen Butternut Mengelslaai	Brokkoli Sampioene Kool Tamatie Greenpeppers Argurkies Lemoenpampoentjies Beet Wortelslaai
Kies 2 x 30g van enige van die volgende proteïene		
<ul style="list-style-type: none"> Maalvleis (Topside) Lae-vet kaas* (Tussers, Ricotta, DairyBell InShape Gouda, Mozzarella, vet-vrye maaskaas of reduced fat feta kaas) 	<ul style="list-style-type: none"> Vis (dit kan vars vis, I&J vissies, tuna, of visvingers wees) Visvingers (1 = 30g) Rooivleis (1-2x/week) Wildsvleis 1 Eier (1 eier = 30g, daarom kan jy 1x van die ander proteïene ook eet) 	<ul style="list-style-type: none"> Volstruis fillets (Hoender fillet) Lae-vet ham Biltong – verkieslik wildsbiltong, anders biltong waarvan die vet verwyder is
Kies een van die volgende vrugte opsies:		
1 x enige vrug waarvan jy hou; bv:		
1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels druiwe, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie		
Kies een van die volgende vet opsies:		
1 tlp botter, 35g se avokadopeer, 1tlp canola olie of 10g se neute		

Of kies een van die voorbeelde soos op die volgende bladsy aangedui....



Of kies een van die volgende voorbeelde:

Voorbeeld 1	Voorbeeld 2	Voorbeeld 3
1 skyfies koue vleis (30g) SAAM MET 30g lae-vet kaas (lae vet fata, Tussers, Mozarella of Ricotta)	Roereier met kaas: Gebruik 1 eier en 30g se kaas	Maak 'n tuna slaai of 'n warm tuna-pasta- gereg: 60g tuna (in water) <i>(Jy kan dit vervang met ander vis ook)</i> 1 k volgraan pasta
2 snytjies Lae GI/ volgraan brood	2 snytjies Lae GI/ volgraan brood	
½ k se slaai groente: Bv tamatie, blaarslaai, sampioene, agurkies, pynappel	½ k se slaai groente: Bv tamatie, blaarslaai, sampioene, agurkies, pynappel	½ k warm groente (Jy kan enige groente eet) OF ½ k slaai groente
1 x enige vrug waarvan jy hou; bv: 1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels druiwe, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie		
1 t Lite margarien OF 35g Avokadepeer OF 1 E Lite mayonaisse of slaaisous OF 10 olywe		
Voorbeeld 4	Voorbeeld 5	Voorbeeld 6
60g se lae vet kaas (Tussers, Ricotta, Dairy Bell InShape Gouda of Mozarella) OF 1 E maaskaas MET 30g fyn biltong	Maak 'n gevulde broodjie of Wrap: 60g maer maalvleis OF 60g se vis (tuna)	60g Feta kaas saam met die groente en CousCous (*Gebruik liefies die reduced fat feta)
6 Provita's OF 4 Rye Vita's	1 groot Pita broodjie OF 1 Wrap	1 k CousCous
½ k se slaai groente: Bv Tamatie, komkommer, blaarslaai, sampioene, agurkiesm pynappel	½ k se slaai groente: Bv Tamatie, komkommer, blaarslaai, sampioene, agurkiesm pynappel	½ k Roasted veggies (of enige ander groente waarvan jy hou)
1 t Lite margarien OF 35g Avokadepeer OF 1 E Lite mayonaisse of slaaisous OF 10 olywe		
1 x enige vrug waarvan jy hou; bv: 1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels druiwe, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie		

Voorbeelde vervolg op die volgende bladsy....



Voorbeelde vervolg vanaf die vorige bladsy....

Voorbeeld 7	Voorbeeld 9	Voorbeeld 9
60 g van enige vleis wat oor is van vorige aand	1 pakkie Cup a Soup MET 1 sny brood OF 3 provita's	1 koppie tuisgemaakte sop (kan enige sop wees) MET 1 sny brood OF 3 provita's
2/3 k rys OF 1 groot aartappel (opgekook in skil) OF 3 baba aartappels opgekook in skil OF 1 k aartappel		
½ k warm groente OF ½ k slaai		
1 t Lite margarien OF 35g Avokadepeer OF 1 E Lite mayonaisse of slaaisous OF 10 olywe		
1 x enige vrug waarvan jy hou; bv: 1 appel, 1 peer, 1 piesang, 1 lemoen, 2 klein naartjies, 17 korrels druwe, 1 perske, 3 vars appelkose, 2 litchi's, ½ k papaja blokkies, ½ k spanspek blokkies, ½ k mango blokkies, 1 ¾ k heel aarbeie		

Neem die Here Jesus Christus as jou skuilplek en God sal Sy ENGELE opdrag gee om jou te beskerm oral waar jy gaan. Op hulle hande sal hulle jou dra (Ps: 91 9-12)

Jesus Sëen

Bereken deur Cornelia King (RD,
SA) Dr. Driekie Rankin (RD, SA)



ALGEMENE RIGLYNE:

- $\frac{1}{2}$ k = 125ml, en 1 k = 250 ml
- 30g vleis = die grootte van 'n vuurhoutjie boksie
- Alle hoeveelhede dui op reeds gaar voedsel

- Onthou om sowat 2 L water per dag te drink.

- Moet asb nie meer of minder as die voorgeskrewe hoeveelheid eet nie; moet asb ook nie maaltye oorslaan nie.

- Hou asb aan met die oefening soos wat jy nou doen sodat ons die vet persentasie 'n bietjie kan afkry, maar nie van jou spier massa verloor nie.

- Laat weet my asb indien jy baie honger raak deur die dag, of as jou energie vlakke baie laag is.

- Hou asb vir my die eerste 2 weke baie streng by die eetprogram sodat ons kan sien of die gewig afkom, en of die dieet nog aangepas moet word.

Kontak my gerus indien jy enige verdere vrae het, of as iets onduidelik is.

Appendix D - Food list and price database

	Serving size	Equivalent unit sold in	Price (R) Pick n Pay	Price (R) Woolworths	Pack size Pick n Pay (relevant unit)	Pack size (relevant unit) Woolworths	Price / required quantity
Full cream Dairy	CHO = 12g Protein = 9g Fat = 9g Kilojoules = 690						
Full cream milk	250 ml	250	7.99	11.65	1000	1000	1.9975
Thick milk from full milk (condensed milk)	250 ml	153	11.99	17.95	385	385	4.764857
Yoghurt from full milk	250 ml	122.5	21.99	9.95	1000	150	2.693775
Evaporated milk, undiluted, unsweetened	125 ml	63	16.99	18.95	410	380	2.610659
Full cream milk powder	60 ml / 35 g	35	35.99	N. A. ⁶	500	1	2.5193
Low fat Dairy	CHO = 12g Protein = 9g Fat = 0g Kilojoules = 350						
Low fat milk	250 ml	250	7.99	11.65	1000	1000	1.9975
Thick milk from low fat	250 ml	126	18.99	N. A.	380	1	6.296684
Butter milk from low fat	250 ml	250	N. A.	12.45	1	500	6.225
Low fat milk powder	60 ml / 35 g	35	35.99	N. A.	500	1	2.5193
Vegetables A	CHO = 0g Protein = 0g Fat = 0g Kilojoules = 0						
Asparagus	125 ml	16.75	N. A.	37.99	1	100	6.363325
Baby marrows	125 ml	13	14.99	28.99	400	1000	0.37687
Broccoli	125 ml	11.38	15.99	17.91	300	500	0.407632
Eggplant	125 ml	10.25	13.99	N. A.	1000	1	0.143398
Green beans	125 ml	12.5	19.99	12.99	1000	350	0.249875
Cucumber	125 ml	13	8.99	9.99	301	301	0.388272

⁶ N.A. = Not available at time of list compilation

Appendix D

Cabbage	125 ml	11.12	9.99	12.99	908	908	0.122344
Parsley	125 ml	7.5	7.99	9.99	25	30	2.397
Radish	125 ml	14.5	N. A.	9.99	1	150	0.9657
Celery	125 ml	12.62	14.99	N. A.	300	1	0.630579
Mushrooms	125 ml	8.75	17.99	33.99	250	500	0.594825
Spinach	125 ml	3.75	9.99	14.99	300	200	0.124875
Tomato	125 ml	22.5	9.99	10.99	750	738	0.2997
Sourcroust	125 ml						N. A.
Vegetables B	CHO= 7g Protein= 2g Fat= 0g Kilojoules= 150						
Beetroot	125 ml	17	4.99	9.99	750	410	0.113107
Gem squash	125 ml	16.25	9.99	18.99	1000	1000	0.162338
Pumpkin	125 ml	14.5	16.99	16.99	500	500	0.49271
Carrots	125 ml	27.5	23.99	11.99	5000	1000	0.131945
Peas	125 ml	36.25	28.99	12.99	200	80	5.254438
Onion	125 ml	40	12.99	13.99	1000	1000	0.5196
Fruit	CHO = 10g Protein = 0g Fat = 0g Kilojoules = 170						
Strawberry	150 g	150	29.99	29.99	400	400	11.24625
Apple	80 g	80	9.99	17.99	1000	1500	0.7992
Apple juice	90 g	90	9.99	15.95	500	1000	1.4355
Dates	15 g	15	24.99	39.99	250	400	1.4994
Grape	75 g	75	19.99	N. A.	500	1	2.9985
Granadilla	90 g	90	15.99	24.99	500	500	2.8782
Cherry	75 g	75					N. A.
Guava	85 g	85	26.99	N. A.	660	1	3.475985
Orange	100 g	100	5.99	7.99	1000	1000	0.599
Litchi	70 g	70					N. A.
Mango	70 g	70	24.99	N. A.	100	1	17.493
Pawpaw	100 g	100	10	15.99	781	314	1.28041
Pear	100 g	100	17.99	16.99	1500	1500	1.132667

Appendix D

Peach	100 g	100					N. A.
Grapefruit	125 g	125	9.99	N. A.	492	1	2.53811
Meat (cooked) and legumes	CHO = 0g Protein = 7g Fat = 5g Kilojoules = 310						
Beef, lamb, pork, poultry, mince	30 g	30	66.99	69.99	1000	1000	2.0097
Egg (boiled)	50 g	50	8.99	9.99	300	300	1.498333
Peanut butter (25 ml)	30 g + 2 fat exchanges	30	20.99	21.95	400	400	1.57425
Cheese, cheddar, sweetmilk	30 g	30	64.99	45.95	1000	400	1.9497
Cottage cheese (not cream cheese)	30 g	30	16.99	N. A.	250	1	2.0388
Sardines (in brine)	30 g	30	9.99	12.95	120	120	2.4975
Fish (white)	30 g	30	32.99	46.36	400	380	2.47425
Viennas	50 g / medium	50	18.99	49.99	500	1000	1.899
Bread	CHO = 15g Protein = 2g Fat = 0g Kilojoules = 290						
Bread, whole wheat/brown/white	1 slice	28	4.99	10.65	400	700	0.3493
Boston	1 slice	28.35					N. A.
Raisin	1 slice	26	8.99	N. A.	400	1	0.58435
Rye	1 slice	32	N. A.	19.95	1	400	1.596
Porridge (cooked) Malt	125 ml	20.62	11.99	N. A.	350	1	0.706382
SPP (raw)	50 g = 2 ex + 1 fat						N. A.
Rice flakes	20 g	20	28.99	30.95	400	400	1.4495
Hi-fibre	20 g	20	27.79	27.95	375	375	1.482133
All bran	20 g	20	39.99	44.95	750	750	1.0664
Puffed wheat	20 g	20	18.49	N. A.	350	1	1.056571
Weet-bix	25 g	25	28	34.95	900	900	0.777778
Muesli	30 g	30	23.49	38.95	750	500	0.9396
Provitas	3 ea	12	14.99	14.95	250	250	0.7176
Cream crackers	3 ea	9.3	9.99	11.95	200	200	0.464535
Salticrax	6 ea	24	14.99	14.95	200	200	1.794

Appendix D

Pasta (cooked)	80 g	80	9.99	19.95	500	500	1.5984
Rice (white/brown, cooked)	80 g	80	5.99	22.95	500	2000	0.918
Cake flour	20 g	20	9.99	10.99	1000	1000	0.1998
Maize flour	25 g	25	6.49	18.95	1000	2500	0.16225
Potatoes (cooked)	100 g	100	18	20	2000	7000	0.285714
Mash with milk, no butter	125 g	125	18	20	2000	7000	0.357143
Sweet potatoes (cooked)	60 g	60	9.99	29.99	1000	4000	0.44985
Mealie (fresh / frozen)	75 g	75	15.99	16.99	408	408	2.939338
Creamed corn	75 g	75	8.99	11.99	410	420	1.644512
Dried beans	25 g	20	23.89	16.95	1000	500	0.4778
Scone	30 g	30	3.99	5.74	57	57	2.1
Muffin (bran)	35 g	35	4.83	4.33	113	113	1.34115
Rusks (buttermilk)	20 g	20	22.99	26.95	1000	500	0.4598
Dry biscuits (tennis)	25 g	25	8.99	9.95	200	200	1.12375
Crumpets	40 g	40	35.99	N. A.	500	1	2.8792
Pop corn	25 g	25	8.49	9.95	500	500	0.4245
Oils	CHO = 0g Protein = 0g Fat = 5g Kilojoules = 190						
Canola	1 tsp.	5	16.99	18.95	750	750	0.113267
Corn	1 tsp.	5					N. A.
Flaxseed oil	1 tsp.	5					N. A.
Olive oil	1 tsp.	5	39.99	49.95	500	250	0.3999
Peanut oil	1 tsp.	5	51.49	N. A.	750	1	0.343267
Safflower oil	1 tsp.	5					N. A.
Sesame oil	1 tsp.	5	48.99	47.95	210	210	1.141667
Soybean oil	1 tsp.	5					N. A.
Sunflower oil	1 tsp.	5	11.99	15.95	500	750	0.106333
Mayonnaise	1 tsp.	4.6	12.99	19.95	375	750	0.12236
Salad dressing	1 tsp.	4.9	16.99	19.95	340	300	0.244856
Margarine	1 tsp.	4.67	9.99	19.99	250	500	0.186613

Appendix E – VBA (Visual Basic for Applications) code

E.1. VBA code that activates and solves the goal programming model by passing the daily energy value to the spreadsheet object that calculates the number of exchange group selections for each group

```
Private Sub CommandButton1_Click()
    'Set the objective cell equal to the calculated energy value
    Sheet14.Cells(1, 3) = Round(Sheet1.Cells(18, 2), 0)
    'Activate the spreadsheet that solves the GP model
    Sheet14.Activate
    'Solve the model created with Solver
    SolverSolve UserFinish:=True
    SolverFinish KeepFinal:=1
    solversave ("Y13") 'Save solution
End Sub
```

E.2. Excerpt of the VBA code used for the execution of the multi-objective linear programming model which solves the least cost and highest preference problem

```
'Full cream diary group
'Determine if the number of selections to be made from group > 0
If Sheet4.Range("c2").Value > 0 Then
    Sheet4.Range("m6").Formula = "=SUM(M7:M8)"           'formula for minimize Q
    Sheet4.Range("m7").Formula = "=D1*((M3-N3)/N3)"      'formula for calc target cost
    Sheet4.Range("m8").Formula = "=E1*((M5-N5)/N5)"      'formula for calc target pref
    solverreset 'reset solver for next problem
    'start cost target: set obj cell, set max or min, set cells to change
    solverok setcell:=Sheet4.Range("M3"), maxminval:=2, bychange:=Sheet4.Range("K3:K7")
        solveradd cellref:=Sheet4.Range("K3:K7"), relation:=5 'add constraints
        solveradd cellref:=Sheet4.Range("K8"), relation:=2, formulatext:=Sheet4.Range("C2").Value
    SolverSolve UserFinish:=True 'accept solution
    Sheet4.Range("N3") = Sheet4.Range("M3") 'set cell M3 to cost target value
    solverreset
    'start pref target: set obj cell, set max or min, set cells to change
    solverok setcell:=Sheet4.Range("M4"), maxminval:=2, bychange:=Sheet4.Range("K3:K7")
        solveradd cellref:=Sheet4.Range("K3:K7"), relation:=5 'add constraints
        solveradd cellref:=Sheet4.Range("K8"), relation:=2, formulatext:=Sheet4.Range("C2").Value
    SolverSolve UserFinish:=True 'accept solution
    Sheet4.Range("N5") = Sheet4.Range("M5") 'set cell M5 to pref target value
    solverreset
    'start minimize Q: set obj cell, set minimise, set cells to change
    solverok setcell:=Sheet4.Range("M6"), maxminval:=2, bychange:=Sheet4.Range("K3:K7")
        solveradd cellref:=Sheet4.Range("K3:K7"), relation:=5 'add constraints
        solveradd cellref:=Sheet4.Range("M5"), relation:=1, formulatext:=1000
        solveradd cellref:=Sheet4.Range("K8"), relation:=2, formulatext:=Sheet4.Range("C2").Value
        solveradd cellref:=Sheet4.Range("M7:M8"), relation:=1, formulatext:=Sheet4.Range("M6").Value
    SolverSolve UserFinish:=True 'accept solution
Else
    Sheet4.Range("m6:m8").Value = 0 'selections of food items if number = 0
End If
```

Appendix F – Eating plans generated by the expert system

This appendix provides an example of the eating plans generated by the expert system. For the complete set of eating plans please refer to the enclosed disk.

F.1. Application of the consultation session of subject 6 with equal cost and preference priority, automatic daily energy calculation.

Personalised diet for a total of 6164 kJ		
Total carbohydrates: 188.55g	Total cost of this diet: R 30.45	
Total protein: 68.89g	Personal food preference satisfied: 82.81%	
Total fat: 47.04g		
Breakfast	Portion size	
Egg (boiled)	50 g	Protein
Porridge (cooked)	125 ml	Starch
Main meal		
Beetroot	125 ml	Vegetable
Gem squash	125 ml	Vegetable
Carrots	125 ml	Vegetable
Beef, lamb, pork, poultry, mince	30 g	Protein
Fish (white)	30 g	Protein
Rice (white/brown, cooked)	80 g	Starch
Potato (cooked)	100 g	Starch
Snacks		
Low fat milk powder	60 ml / 35 g	Low fat dairy
Apple	80 g	Fruit
Apple juice	90 g	Fruit
Orange	100 g	Fruit
Peanut butter	30 g + 2 fat exchanges	Protein
Pop corn	25 g	Starch
Any Time		
Full cream milk	250 ml	Full cream dairy
Low fat milk	250 ml	Low fat dairy
Butter milk from low fat	250 ml	Low fat dairy
Tomato	125 ml	Vegetable
Pawpaw	100 g	Fruit
Cheese, cheddar, sweetmilk	30 g	Protein
Bread, whole wheat/brown/white	1 slice	Starch

Appendix F

Cake flour	20 g	Starch
Maize flour	25 g	Starch
Canola	1 tsp.	Oil
Sunflower oil	1 tsp.	Oil
Mayonnaise	1 tsp.	Oil

References

- Albright, S.C. (2012) *VBA for Modelers: Developing decision support systems with Microsoft (R) Office Excel (R)*, 4th edition, South-Western, Cengage Learning.
- Aldeseit, B., Majdalawi, M.I. and Ata, M. (2012) 'Application of linear programming technique to formulate least cost balanced ration for calves-fattening in Jordan', *Journal of Animal and Veterinary Advances*, vol. 11, no. 17, pp. 3119-3124.
- Allen, I.E. and Seaman, C.A. (2007) *Likert scales and data analyses*, [Online] Available: <http://mail.asq.org/quality-progress/2007/07/statistics/likert-scales-and-data-analyses.html> [18 December 2013].
- Anderson, A.M. and Earle, M.D. (1983) 'Diet planning in the Third World by linear and goal programming', *The Journal of the Operational Research Society*, vol. 34, no. 1, pp. 9-16.
- Anonymous (2008) *Major components of expert systems*, [Online], Available: <http://year12ipt.ash.com/untitled-7.html> [22 January 2014].
- Arsene, O., Dumitrache, I. and Miha, I. (2011) 'Medicine expert system dynamic Bayesian Network and ontology based', *Expert Systems with Applications*, vol. 38, no. 1, pp. 15253-15261.
- Barbara, D., Couto, J., Jajodia, S., Popyack, L. and Wu, N. (2001) 'ADAM: Detecting intrusions by data mining', *Proceedings of the IEEE Workshop on Information Assurance and Security*, West Point, pp. 11-16.
- Bassham, S., Fletcher, L.R. and Stanton, R.H.J. (1984) 'Dietary analysis with the aid of a microcomputer', *Journal of Microcomputer Applications*, vol. 7, no. 3-4, pp. 279-289.
- Berry, D.C., Gillie, T. and Banbury, S. (1995) 'What do patients want to know: An empirical approach to explanation generation and validation', *Expert Systems with Applications*, vol. 8, no. 4, pp. 419-428.
- Bilton, J. (2011) *Technological Determinism*, [Online], Available: <http://atschool.eduweb.co.uk/trinity/determin.html> [16 May 2012].

- Bogg, D.A., Rosenberg, L., Cozier, Y.C., Wise, L.A., Coogan, P.F., Ruiz-Narvaez, E.A. and Palmer, J.R. (2011) 'General and abdominal obesity and risk of death among black women', *The New England Journal of Medicine*, vol. 365, no. 10, pp. 901-908.
- Boose, J.H. (1986) *Expertise transfer for expert system design*, 1st edition, Amsterdam: Elsevier Science Publishers B. V.
- Bourguet, J.-R., Thomopoulos, R., Mugnier, M.-L. and Abécassis, J. (2013) 'An artificial intelligence-based approach to deal with argumentation applied to food quality in a public health policy', *Expert Systems with Applications*, vol. 40, no. 1, pp. 4539-4546.
- Cadenas, J.M., Pelta, D.A., Pelta, H.R. and Verdegay, J.L. (2004) 'Application of fuzzy optimization to diet problems in Argentinean farms', *European Journal of Operational Research*, vol. 158, no. 1, October, pp. 218–228.
- Caradas, A.A., Lambert, E.V. and Charlton, K.E. (2001) 'An ethnic comparison of eating attitudes and associated body image concerns in adolescent South African schoolgirls', *Journal of Human Nutrition and Dietetics*, vol. 14, no. 2, December, pp. 111-120.
- Chen-Chin, L. and Tomsovic, K. (1986) 'An expert system assisting decisions-making of reactive power/voltage control', *Power Systems, IEEE Transactions on*, vol. 1, no. 3, August, pp. 195-201.
- Chen, Y., Hsu, C.-Y., Liu, L. and Yang, S. (2012) 'Constructing a nutrition diagnosis expert system', *Expert Systems with Applications*, vol. 39, no. 1, pp. 2132-2156.
- Clancey, W.J. (1988) ' Knowledge-based tutoring: The GUIDON program', *Journal of Computer Assisted Learning*, vol. 4, no. 4, pp. 250-252.
- Cole, T.J., Freeman, J.V. and Preece, M.A. (1995) 'Body mass index reference curves for the UK, 1990', *Archives of Disease in Childhood*, vol. 73, pp. 25-29.
- Dantzig, G.B. (1963) *Linear Programming and extensions*, 1st edition, New Jersey: Princeton University Press.
- Dantzig, G.B. (1990) 'The diet problem', *Interfaces*, vol. 20, no. 4, pp. 43-47.

- Darby-Dowman, K. and Wilson, J.M. (2002) 'Developments in linear and integer programming', *The Journal of the Operational Research Society*, vol. 53, no. 10, pp. 1067-1071.
- Darmin, N., Ferguson, E.L. and Briend, A. (2006) 'Impact of a cost constraint on nutritionally adequate food choices for French women: An analysis by linear programming', *Journal of Nutrition Education and Behavior*, vol. 38, no. 2, pp. 82-90.
- David, O., Schneider, I.W. and Leavesley, G.H. (2013) *Metadata and modeling frameworks: The object modeling system example*, [Online], Available: <http://deprojects.googlecode.com/svn/tg/doc/artigos%20e%20teses/Metadata%20and%20Modeling%20Frameworks%20The%20Object%20Modeling%20System%20Example.pdf> [17 December 2013].
- De Villiers, M.R. (2005) 'Three approaches as pillars for interpretive information systems research: development research, action research and grounded theory', Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries, White River, pp. 142-151.
- Dixit, A. and Saxena, P.C. (2011) 'Umbrella: A new component-based software development model', 2009 International Conference on Computer Engineering and Applications, Singapore, pp. 61-65.
- Droy, J.-M., Darmoni, S.J., Massari, P., Blanc, T., Moritz, F. and Leroy, J. (1993) 'SETH: An expert system for the management on acute drug poisoning', *Computer Methods and Programs in Biomedicine*, vol. 43, pp. 171-176.
- Dubois, D. and Prade, H. (1978) 'Operations on fuzzy numbers', *International Journal of Systems Science*, vol. 9, no. 6, pp. 613-626.
- Eiselt, H.A. and Sandblom, C.-L. (2000) *Integer programming and network models*, 1st edition, Heidelberg: Springer-Verlag.
- Elliot, E.J. (2007) 'Acute gastroenteritis in children', *British Medical Journal*, vol. 334, no. 7583, pp. 35-40.

- Emrouznejad, A. and Ho, W. (2012) *Applied Operational Research with SAS*, Boca Raton: CRC Press, Taylor & Francis Group.
- Feeley, A., Musenge, E., Pettifor, J.M. and Norris, S.A. (2012) 'Changes in dietary habits and eating practices in adolescents living in urban South Africa: The birth to twenty cohort', *Journal on Nutrition*, vol. 28, no. 7, pp. e1-e6.
- Ferguson, E.L., Darmon, N., Briend, A. and Premachandra, I.M. (2004) 'Food-based dietary guidelines can be developed and tested using linear programming analysis', *The Journal of Nutrition*, vol. 134, no. 4, pp. 951-957.
- Fikes, R. and Kehler, T. (1985) 'The role of frame-based presentation in reasoning', *Communications of the ACM*, September, pp. 904-920.
- FNIC (2013) *United States Department of Agriculture National Agriculture Library*, [Online], Available: <http://fnic.nal.usda.gov/dietary-guidance/myplatefood-pyramid-resources> [15 December 2013].
- Frega, R., Lanfranco, J.G., De Greve, S., Bernardini, S., Geniez, P., Grede, N., Bloem, M. and De Pee, S. (2012) 'What linear programming contributes: World Food Programme experience with the "Cost of the Diet" tool', *Food and Nutrition Bulletin*, vol. 33, no. 3, pp. S228-S234.
- Fryer, S., Waller, G. and Kroese, B.S. (1997) 'Stress, coping, and disturbed eating attitudes in teenage girls', *The International Journal of Eating Disorders*, vol. 22, no. 4, pp. 427-436.
- Gagne, J.M.L., Andersen, M. and Norford, L.K. (2011) 'An interactive expert system for daylighting design exploration', *Building and Environment*, vol. 46, no. 11, pp. 2351-2364.
- Garille, S.G. and Gass, S.I. (2001) 'Stigler's diet problem revisited', *Operations Research*, vol. 49, no. 1, pp. 1-13.
- Gass, S.I. (2003) *Linear programming: Methods and applications*, 5th edition, New York: McGraw-Hill.
- Gennari, J.H., Musen, M.A., Ferguson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F. and Tu, S.W. (2003) 'The evolution of Protégé: an environment for

- knowledge-based systems development', *International Journal of Human-Computer Studies*, vol. 58, no. 1, pp. 89-123.
- Giarratano, J.C. and Riley, G.D. (2005) *Expert systems: Principles and programming*, 4th edition, Boston: Thomson Course Technology.
- Goedecke, J.H., Jennings, C.L. and Lambert, E.V. (2006) 'Obesity in South Africa', in *Chronic Diseases of Lifestyle in South Africa: 1995 - 2005*, Cape Town: South African Medical Research Council.
- Graciani, C., Guitiérrez-Naranjo, M.Á., Pérez-Hurtado, I., Riscos-Núñez, A. and Romero-Jiménez, A. (2013) 'A Rete-based algorithm for rule selection in P systems', *International Journal of Unconventional Computing*, vol. 9, no. 1, pp. 367-384.
- Grzegorz, J.N., Bobek, S., Ligeza, A. and Kaczor, K. (2011) 'Algorithms for rule inference in modularized rule bases', *Rule-Based Reasoning, Programming, and Applications: 5th International Symposium, RuleML 2011 - Europe, Barcelona*, pp. 305-312.
- Haeussler, E.F.J., Paul, R.S. and Wood, R.J. (2008) *Introductory Mathematical Analysis for Business, Economics, and the Life and Social Sciences*, 12th edition, Upper Saddle River: Pearson Perntice Hall.
- Hamzah, P., Shuib, A., Tumin, H., Shamsudin, M.R. and Yusof, S.M. (2011) 'Two-stage LP diet model for Malay households', *IEEE Colloquium on Humanities, Science and Engineering Research, Penang*, pp. 562-567.
- Haven, J., Burns, A., Britten, P. and Davis, C. (2006) 'Developing the consumer interface for the MyPyramid food guidance system', *Journal of Nutrition Education and Behavior*, vol. 38, no. 6, pp. S124-S135.
- Head, K.A. and Kelly, G.S. (2009) 'Nutrients and botanicals for treatment of stress: adrenal fatigue, neurotransmitter imbalance, anxiety, and restless sleep', *Alternative Medicine Review*, vol. 14, no. 2, pp. 114-140.
- Hobday, R.A., Thomas, S., O'Donovan, A., Murphy, M. and Pinching, A.J. (2008) 'Dietary intervention in chronic fatigue syndrome', West Smithfield: The British Dietetic Association (Research report).

- Hornick, B.A., Krester, A.J. and Nicklas, T.A. (2008) 'Menu modeling with MyPyramid food patterns: Incremental dietary changes lead to dramatic improvements in diet quality of menus', *Journal of the American Dietetic Association*, vol. 108, no. 12, pp. 2077-2083.
- Huang, M.-J. and Chen, M.-Y. (2007) 'Integrated design of the intelligent web-based Chinese Medical Diagnostic System (CMDS) – Systematic development for digestive health', *Expert Systems with Applications*, vol. 32, no. 1, pp. 658-673.
- Hughes, J.A. and Sharrock, W.W.W. (1997) *The Philosophy of Social Research*, 3rd edition, New York: Longman Publishing.
- Huth, M. and Ryan, M. (2004) *Logic in Computer Science: Modelling and reasoning about systems*, 2nd edition, Cambridge: Cambridge University Press.
- Igumbor, E.I., Sanders, D., Puoane, T.R., Tsolekile, L. and Schwarz, C. (2012) "Big Food," the Consumer Food Environment, Health, and the Policy Response in South Africa', *PLoS Medicine*, vol. 9, no. 7, July, pp. 1-7.
- Jaworowska, A., Blackham, T., Davies, I.G. and Stevenson, L. (2013) 'Nutritional challenges and health implications of takeaway and fast food', *Nutrition Reviews*, vol. 71, no. 5, pp. 310-318.
- Jones, D. and Mehrdad, T. (2010) *Practical goal programming*, 1st edition, New York: Springer.
- Kahraman, A. and Seven, H.A. (2005) 'Healthy daily meal planner', Proceedings of the 2005 workshops on Genetic and evolutionary computation, New York, pp. 390-393.
- Kimani-Murage, E.W. (2013) 'Exploring the paradox: double burden of malnutrition in rural South Africa', *Global Health Action*, January, pp. 193-205.
- Labadarios, D., Steyn, N.P. and Nel, J. (2011) 'How diverse is the diet of adult South Africans?', *Nutrition Journal*, vol. 10, no. 33, pp. 1-11.
- Lancaster, L.M. (1992) 'The evolution of the diet model in managing food systems', *Interfaces*, vol. 22, no. 5, pp. 59-68.

- Larman, C. and Basili, V.R. (2003) 'Iterative and incremental developments: A brief history', *IEEE Computer Society*, vol. 36, no. 6, pp. 47-56.
- Lawler, E.L. and Wood, D.E. (1966) 'Branch-and-bound methods: A survey', *Operations Research*, vol. 14, no. 4, pp. 699-720.
- Liao, S.-H. (2005) 'Expert system methodologies and applications - a decade review from 1995 to 2004', *Expert Systems with Applications*, vol. 28, pp. 93-103.
- Linde, P. and Delpont, C. (2012) 'A linear programming approach to the construction of human diets at minimum cost with taste beneficiaries', Potchefstroom: NWU, Potchefstroom Campus (Honors project report).
- Lindqvist, U. and Porras, P.A. (1999) 'Detecting computer and network misuse through the production-based expert system toolset (P-BEST)', Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, CA, pp. 146-161.
- Lo, C.-C., Chen, C.-H., Cheng, D.-Y. and Kung, H.-Y. (2011) 'Ubiquitous Healthcare Service System with Context-awareness capability: design and implementation', *Expert Systems with Applications*, vol. 38, no. 1, pp. 4416-4436.
- López-Nores, M., Blanco-Fernández, Y. and Pazos-Arias, J.J. (2010) 'Electronic health records as a new dimension of user profiles in recommender systems', *Global Telemedicine and eHealth Updates*, vol. 3, no. 1, pp. 137-140.
- López-Nores, M., Blanco-Fernández, Y., Pazos-Arias, J.J. and Gil-Solla, A. (2012) 'Property-based collaborative filtering for health-aware recommender systems', *Expert Systems with Applications*, vol. 39, no. 1, pp. 7451-7457.
- Mahan, L.K. and Escott-Stump, S. (2004) *Krause's food, nutrition and diet therapy*, 11th edition, Philadelphia: Saunders.
- Maillot, M., Vieux, F., Amiot, M.J. and Darmon, N. (2010) 'Individual diet modeling translates nutrient recommendations into realistic and individual-specific food choices', *The American Journal of Clinical Nutrition*, vol. 91, pp. 421-430.
- Mamat, M., Rokhayati, Y., Noor, N.M.M. and Mohd, I. (2011) 'Optimizing human diet problem with fuzzy price using fuzzy linear programming approach', *Pakistan Journal of Nutrition*, vol. 10, no. 6, pp. 594-598.

- Masić, I., Ridanović, Z. and Pandza, H. (1995) 'Medical expert systems', *Medicinski Arhiv*, vol. 49, no. 3-4, pp. 107-112.
- McCarl, B.A. (2007) *Chapter II: Linear programming*, [Online], Available: <http://agecon2.tamu.edu/people/faculty/mccarl-bruce/mccspr/new02.pdf> [15 January 2014].
- McKenzie, Y.A., Alder, A., Anderson, W., Wills, A., Goddard, L., Gulia, P., Jankovich, E., Mutch, P., Reeves, L.B., Singer, A. and Lomer, M.C.E. (2012) 'British Dietetic Association evidence-based guidelines for the dietary management of irritable bowel syndrome', *Journal of Human Nutrition and Dietetics*, vol. 25, pp. 260-274.
- Midle-Busch, A., Blaschek, A., Borggäfe, I., Heinen, F., Straube, A. and Von Kries, R. (2010) 'Associations of diet and lifestyle with headache in high-school students: Results from a cross-sectional study', *The Journal of Head and Face Pain*, vol. 50, no. 7, July, pp. 1104-1114.
- Mino, Y. and Kobayashi, I. (2009) 'Recipe recommendation for a diet considering a user's schedule and the balance of nourishment', IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, pp. 383-387.
- Monyeki, M.A., Neetens, R., Moss, S.J. and Twisk, J. (2012) 'The relationship between body composition and physical fitness in 14 year old adolescents residing within the Tlokwe local municipality, South Africa: The PAHL study', *BMC Public Health*, vol. 12, no. 1, pp. 374-381.
- Moore, J.H. and Weatherford, L.R. (2001) *Decision modelling with Microsoft Excel*, 6th edition, New Jersey: Pearson Education International.
- Moraes, L.E., Wilen, J.E., Robinson, P.H. and Fadel, J.G. (2012) 'A linear programming model to optimize diets in environmental policy scenarios', *Journal of Dairy Science*, vol. 95, no. 3, pp. 1267-1282.
- Nash, J.C. (2000) 'The (Dantzig) simplex method for linear programming', *Computing Science & Engineering*, vol. 2, no. 1, pp. 29-31.
- Naufel, K.Z., Briley, K.E., Harackiewicz, L.K., Marzec, K.P. and Nielsen, M.E. (2010) 'How do Psychology students use web-based information? Trends and

implications from a descriptive study', *North American Journal of Psychology*, vol. 12, no. 1, pp. 1-14.

Ness, R.J. (1997) 'Rescue support apparatus', (Patent: US 5,644,294).

Neumark-Sztainer, D. (2009) 'Preventing obesity and eating disorders in adolescents: What can health-care providers do?', *Journal of Adolescent Health*, vol. 44, no. 3, pp. 206-213.

Neumark-Sztainer, D., Wall, M., Larson, N.I., Eisenberg, M.E. and Loth, K. (2011) 'Dieting and disordered eating behaviors from adolescence to young adulthood: Findings from a 10-year longitudinal study', *Journal of the American Dietetic Association*, vol. 111, no. 7, pp. 1004-1011.

Neumark-Sztainer, D., Wall, M.E.M.E., Story, M. and Hannan, P.J. (2006) 'Overweight status and weight control behaviors in adolescents: Longitudinal and secular trends from 1999 to 2004', *Journal of Preventative Medicine*, vol. 43, no. 1, pp. 52-59.

Neumark-Sztainer, D., Wall, M., Story, M. and Standish, A.R. (2012) 'Dieting and unhealthy weight control behaviors during adolescence: Associations with 10-year changes in body mass index', *Journal of Adolescent Health*, vol. 50, no. 1, pp. 80-86.

Nicklas, T.A. (1995) 'Dietary studies of children: The Bogalusa Heart Study experience', *Journal of the American Dietetic Association*, vol. 95, no. 10, pp. 1127-1133.

Nienaber-Rousseau, C., Venter, C.S. and Wright, H.H. (2012) 'Voedingstatus', Potchefstroom: NWU, Potchefstroom Campus (Study guide: VOED 122 PAC).

Niu, S.H., Ong, S.K. and Nee, A.Y.C. (2013) 'An improved intelligent water drops algorithm for solving multi-objective job shop scheduling', *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2431-2442.

Oates, B.J. (2006) *Researching information systems and computing*, London: SAGE Publications Ltd.

- Ogbuji, Q.C. (2010) 'Obesity and reproductive performance in women', *African Journal of Reproductive Health*, vol. 14, no. 3, pp. 143-151.
- Orlikowski, W.T. and Baroudi, J.J. (1991) 'Studying Information Technology in organizations: Research approaches and assumptions', *Informations Systems Research*, vol. 2, no. 1, pp. 1-28.
- Pazos-Arias, J.J., López-Nores, M., García-Duque, J., Díaz-Redondo, R.P., Blanco-Fernández, Y., Ramos-Cabrer, M., Gil-Solla, A. and Fernández-Vilas, A. (2008) 'Provision of distance learning services over interactive digital TV with MHP', *Computers and Education*, vol. 50, no. 3, pp. 927-949.
- Peterson, C.D., Norris, S.A. and Pettifor, J.M. (2006) 'Eating attitudes in a group of 11-year-old urban South African girls', *South African Journal of Nutrition*, vol. 19, no. 2, pp. 80-85.
- Pfeuffer, F., Stiglmayr, M. and Klamroth, K. (2012) 'Discrete and geometric Branch-and-bound algorithms for medical image registration', *Annals of Operations Research*, vol. 196, no. 1, pp. 737-765.
- Pick'nPay (2012) *Pick n Pay Online*, [Online], Available: [https://shop.pnp.co.za/b2c_pnp/b2c/display/\(layout=5.1-6_2_4_82_83_8_3&area=%24ROOT\)/.do?rf=y](https://shop.pnp.co.za/b2c_pnp/b2c/display/(layout=5.1-6_2_4_82_83_8_3&area=%24ROOT)/.do?rf=y) [6 August 2012].
- Puoane, T., Bradley, H. and Hughes, G.D. (2005) 'Obesity among black South African women', *Human Ecology*, no. 13, pp. 91-95.
- Puoane, T., Steyn, K., Bradshaw, D., Laubscher, R., Fourie, J., Lambert, V. and Mabananga, N. (2002) 'Obesity in South Africa: The South African demographic and health survey', *Obesity Research*, vol. 10, no. 10, October, pp. 1038-1048.
- Ragsdale, C.T. (2007) *Managerial decision modelling*, 1st edition, Mason, Ohio: Thomson Higher Education.
- Rankin, D. (2012) Registered Dietician [*Personal interview*], 24 July, Potchefstroom.
- Reddy, S.P., Resnicow, K., James, S., Funani, I.N., Kambaran, N.S., Omardien, R.G., Masuka, P., Sewpaul, R., Vaughan, R.D. and Mbewu, A. (2012) 'Rapid increases in overweight and obesity among South African adolescents:

- Comparison of data from the South African National Youth Risk Behaviour Survey in 2002 and 2008', *American Journal of Public Health*, vol. 102, no. 2, pp. 262-268.
- Render, R., Stair, R.M. and Hanna, M.E. (2009) *Quantitative analysis for management*, 10th edition, Upper Saddle River: Pearson Prentice-Hall.
- Rolf, H.L. (2013) *Finite Mathematics*, 8th edition, Boston: Cengage Learning.
- Romm, N.R.A. (1991) *The methodologies of positivism and marxism, a sociological debate*, London: MacMillan.
- Romm, N. (2010) 'Experimental research: Studying variables to examine causal effects in terms of mitigating against the potential of racism', *New Racism*, pp. 103-169.
- Rossouw, H.A., Grant, C.C. and Viljoen, M. (2012) 'Overweight and obesity in children and adolescents: The South African problem', *South African Journal of Science*, vol. 108, no. 5/6, pp. 1-7.
- Ruis-Mezcua, B., Garcia-Crespo, A., Lopez-Cuadrado, J.L. and Gonzalez-Carrasco, I. (2011) 'An expert system development tool for non AI experts', *Expert Systems with Applications*, vol. 38, no. 1, pp. 579-609.
- Russel, S.J. and Norvig, P. (2010) *Artificial Intelligence: A modern approach*, 3rd edition, Upper Saddle River: Prentice Hall.
- Schönfeldt, H.C., Gibson, N. and Vermeulen, H. (2010) 'The possible impact of inflation on nutritionally vulnerable households in a developing country using South Africa as a case study', *Nutrition Bulletin*, vol. 35, no. 3, pp. 254-267.
- Senekal, M., Steyn, N.P., Mashego, T.B. and Nel, J.H. (2001) 'Evaluation of body shape, eating disorders and weight management related parameters in black female students of rural and urban origins', *South African Journal of Psychology*, vol. 31, no. 1, pp. 45-53.
- Sepkowitz, K.A. (2001) 'AIDS - The first 20 years', *The New England Journal of Medicine*, vol. 344, no. 23, pp. 1764-1772.

- Sing, N.H. and Chang, Y.S. (2005) 'An agent to detect online false and misleading claims for weight-loss products in Korea', *Expert Systems with Applications*, vol. 29, no. 1, pp. 153-161.
- Sklan, D. and Dariel, I. (1993) 'Diet planning for humans using mixed-integer linear programming', *British Journal of Nutrition*, vol. 70, no. 1, pp. 27-35.
- Soden, P.M. and Fletcher, L.R. (1992) 'Modifying diets to satisfy nutritional requirements using linear programming', *British Journal of Nutrition*, vol. 68, no. 3, pp. 565-572.
- Sterling, L., Petot, G., Marling, C., Kovacic, K. and Ernst, G. (1996) 'The role of common sense knowledge in menu planning', *Expert Systems with Applications*, vol. 11, no. 3, pp. 301-308.
- Stewart, M.L. and Schroeder, N.M. (2013) 'Dietary treatments for childhood constipation: efficacy of dietary fibre and whole grains', *Nutrition Reviews*, vol. 71, no. 2, pp. 98-109.
- Steyn, N.P., Labadarios, D., Maunder, E., Nel, J. and Lombard, C. (2005) 'Secondary anthropometric data analysis of the National Food Consumption Survey in South Africa: the double burden', *Journal of Nutrition*, vol. 21, no. 1, January, pp. 4-13.
- Steyn, N.P., Labadarios, D. and Nel, J.H. (2011) 'Factors which influence the consumption of street foods and fast foods in South Africa - a national survey', *Nutrition Journal*, vol. 10, no. 104, pp. 1-10.
- Steyn, N.P., Senekal, M., Brits, S., Alberts, M., Mashego, T. and Nel, J.H. (2000) 'Weight and health status of black female students', *South African Medical Journal*, vol. 90, no. 2, pp. 146-152.
- Stigler, G.J. (1945) 'The cost of subsistence', *Journal of Farm Economics*, vol. 27, no. 2, pp. 303-314.
- Stupar, D., Eide, W.B., Bourne, L., Hendricks, M., Iverson, P.O. and Wandel, M. (2012) 'The nutrition transition and the human right to adequate food for adolescents in the Cape Town metropolitan area: Implications for nutrition policy', *Food Policy*, vol. 37, no. 3, pp. 199-206.

- Su, C.-T. and Yang, C.-H. (2008) 'Feature selection for the SVM: An application to hypertension diagnosis', *Expert Systems with Applications*, vol. 34, no. 1, pp. 754-763.
- Taylor, E. (2007) 'n Model van die faktore wat die sukses van onderrigleer van tegnologie-gebaseerde onderwerpe beïnvloed', Potchefstroom: NWU, Potchefstroom Campus (Thesis - PhD).
- Taylor, B.W.I. (2013) *Introduction to Management Science*, 11th edition, Harlow: Pearson Education Limited.
- Temin, M., Levine, R. and Stonesifer, S. (2010) 'Start with a girl: A new agenda for global health', *Issues in Science & Technology*, vol. 26, no. 3, pp. 33-40.
- Thagard, P. (2012) *Cognitive Science*, [Online], Available: <http://plato.stanford.edu/archives/fall2012/entries/cognitive-science/> [21 January 2014].
- Ting, S.L., Kwok, S.K., Tsang, A.H.C. and Lee, W.B. (2010) 'CASESIAN: A knowledge-based system using statistical and experiential perspectives for improving the knowledge sharing in the medical prescription process', *Expert Systems with Applications*, vol. 37, no. 1, pp. 5336-5346.
- Tobin, P.K. (2006) 'The use of stories and storytelling as knowledge sharing practices: A case study in the South African mining industry', Pretoria: University of Pretoria (Thesis - DPhil).
- Torelli, P., Evangelista, A., Bini, A., Castellini, P., Lambru, G. and Manzoni, G.C. (2009) 'Fasting headache: A review of the literature and new hypothesis', *American Headache Society*, May, pp. 744-752.
- Van der Merwe, M.T. (2012) 'Obesity in childhood and adolescence', *The South African Medical Journal*, vol. 102, no. 5, p. 289.
- Vanderbrei, R.J. (2008) *Linear programming: Foundations and extensions*, 1st edition, New York: Springer.

- Wagner, W.P., Chung, Q.B. and Najdawi, M.K. (2003) 'The impact of problem domains and knowledge acquisition techniques: A content analysis of P/OM expert system case studies', *Expert Systems with Applications*, vol. 24, no. 1, pp. 79-86.
- Walker, A.R.P. (1995) 'Nutrition-related diseases in Southern Africa: With special reference to urban African populations in transition', *Nutrition Research*, vol. 15, no. 7, pp. 1053-1094.
- Wang, X., Bai, Y., Cai, C. and Yan, X. (2010) 'A production rule- based knowledge system for software quality evaluation', Computer Engineering and Technology (ICCET), 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, 208-211.
- Wang, M.-H. and Su, Y.-T. (2012) 'Using extension method for health analysis system of a fitness device', *Expert Systems with Applications*, vol. 39, no. 1, pp. 12833-12840.
- Whitney, E. and Rolfes, S.R. (2008) *Understanding Normal and Clinical Nutrition*, 8th edition, Belmont: Thomson Wadsworth.
- Williams, H.P. (2013) *Model building in mathematical programming*, 5th edition, West Sussex: John Wiley & Sons.
- Woolworths (2012) *Woolworths*, [Online], Available: <http://www.woolworths.co.za/> [7 August 2012].
- World Health Organisation, *BMI Classification* (2006), [Online], Available: http://apps.who.int/bmi/index.jsp?introPage=intro_3.html [10 September 2013].
- Youm, S., Lee, G., Park, S. and Zhu, W. (2011) 'Development of remote healthcare system for measuring and promoting healthy lifestyle', *Expert Systems with Applications*, vol. 38, no. 1, pp. 2828-2834.