Developing Decision Support for FoodBank South Africa's Allocation System

An application of Operational Research techniques to aid decision-making at a not-for-profit organization

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

DEPARTMENT OF STATISTICAL SCIENCES

NEIL MARK WATSON

submitted in partial fulfilment of the Masters in Operational Research for Development degree in the Faculty of Science

June 27, 2011

Supervisors: Prof. Theodor Stewart and Dr. Leanne Scott

Plagiarism Statement

I know the meaning of plagiarism and declare that all of the work in this dissertation, save for that which is properly acknowledged, is my own.

Signed

Neil Mark Watson

Abstract

There is a wealth of literature pertaining to the application of hard Operational Research (OR) techniques (simulation, linear programming, goal programming etc.) in determining optimal ordering, inventory and allocation policies for goods within distribution systems in developed economies. In contrast, there is a dearth of research relating to similar applications in developing economies, or more particularly in the unique context of a developing country. This study aims to assist decision making at a not-for-profit organization (NPO), Foodbank South Africa (FBSA), within its allocation system through a combined 'soft-hard' OR approach. Two problem-structuring tools (soft OR), Causal Mapping (CM) and Soft System Methodology's Root Definitions (RDs), are used to structure the organization with respect to its goals (in order to gain a comprehensive understanding of the decision-context in which decisions are made) and gain a better understanding of the 'decision–issues' within the allocation system at the Cape Town (CT) warehouse. A simulation approach (hard OR) is applied to the daily operations at the CT warehouse, wherein 'decision-rules' related to allocating available food stocks that are utilized by floor managers are simulated on a daily basis over a fixed period of time. The simulation model is run iteratively, and optimized with respect to each of the built-in parameters defining these decision-rules. The predicted output of the simulation model is a range of optimum allocation policies that could be utilized. A decision support tool is developed that will automate daily allocation decisions, depending on the particular allocation policy adopted.

Keywords: Cognitive/causal mapping, Decision Support System, Foodbanking, Operation Research, Operational Research for Development (ORD), Problem–structuring, Root Definitions, Simulation.

Contents

Acknowledgements

List of Acronyms

CM – Causal Mapping/Map

CSIR – Centre for Scientific and Industrial Research

CT – Cape Town

CV – Coefficient of Variation

DA – Decision Analysis

DM/s – Decision Maker/s

DR/s – Decision Rule/s

FAO – Food and Agriculture Organization

FAST – FBSA's (Agency) Allocation Support Tool

FBCT – Foodbank Cape Town

FBSA – Foodbank South Africa

FIFO – First–In–First–Out

GFN – Global Foodbanking Network

LIFO – Last–In–First–Out

MCDA – Multiple Criteria Decision Analysis

NAC – National Agency Coordinator

NPO – Not–for–profit Organization

ORD – Operational Research for Development

OR – Operations/Operational Research

ORers – Operations/Operational Researchers

PSMs – Problem Structuring Methods

RD – Root Definition

SA – South Africa

SCA – Strategic Choice Approach

SODA – Strategic Options Development Analysis

SSDM – Soft Systems Dynamics Methodology

SSM – Soft Systems Methodology

UCT – University of Cape Town

UN – United Nations

US – United States

VBA – Visual Basic for Applications

VSM – Viable Systems Model

WHO – World Health Organization

List of Figures

List of Tables

Chapter 1

Introduction

1.1   Introduction to this study

The volume of research into the use of OR techniques to aid decision making in developing countries, with a strong focus on development (ORD), pales in comparison to the vast record of OR interventions in developed countries. There has been extensive application of 'hard' OR techniques to problems of ordering, storing and allocating items within distribution systems in developed countries. However, there is a dearth of similar research in developing countries like South Africa, with even fewer studies attempting to incorporate the use of 'soft' OR in their approach.

This study is a humble attempt at filling part of this 'research gap' by applying a combined 'soft–hard' OR approach to aid decision making in the area of allocation at a not–for–profit organization, FBSA, that represents the largest hunger–relief network in SA.

The study uses two problem–structuring tools (soft OR) to gain a greater understanding of FBSA in terms of its goals, and consequently a good appreciation of the context in which decisions in the organization are made, as well as a better understanding of the 'decision–issues' within the allocation system at the CT warehouse. A simulation model (hard OR) is developed to imitate daily allocation decisions, with the end–objective of assisting decision–making by developing an optimal range of allocation policies. A decision support system (DSS) is developed to help FBSA manage their agency database and automate some of the daily allocation decisions.

It is envisioned that the output of this study will be helpful on a practical level to FBSA and thus, albeit indirectly, better the plight of the millions of hungry people in SA. It is hoped that on a theoretical level, this study will motivate others to pursue similar ORD work in this and other areas of need in SA.

1.2 Organization of dissertation

This half–dissertation is organized as follows. First, a background to the problem will be provided in Chapter 2 by discussing the issues of hunger and food insecurity (section 2.1), and Foodbanking (section 2.2). Chapter 3 outlines the process of how the topic for this study came to be selected (sections 3.2 – 3.3). A review of OR work conducted in similar areas is provided in Chapter 4, where emphasis is placed on the allocation of inventory (section 4.2), the use of simulation in food–related problems (section 4.3), and the differences in approach between traditional OR and ORD (section 4.4).

Chapter 5 deals with the process of structuring the problem, providing an introduction to problem–structuring (section 5.1), outlining the methods used in structuring the problem (section 5.2) and then describing their application (sections 5.3 and 5.4).

Chapter 6 outlines the process of building the simulation model used in this study, from data collection and sorting (section 6.1) to distribution fitting (section 6.2) to a detailed description of the code used to create the model (section 6.3) as well as a comprehensive account of the development and function of the Decision Support System, FAST (FBSA's Agency Allocation Support Tool) (section 6.4). Chapter 7 details the validation (section 7.1), verification (section 7.2) and credibility (section 7.3) tests carried out on the simulation model. Chapter 8 describes the allocation models considered in this study (section 8.1), the code that runs the simulation model (section 8.2) and the simulation experiments conducted (section 8.3), the results of which are presented and discussed (section 9.1) in Chapter 9.

Finally, Chapter 10 concludes this study with some pertinent conclusions (section 10.1), reflection (section 10.2) and recommendations (section 10.3) for further study.

Chapter 2

Background to Study

2.1   Hunger and food insecurity

Hunger (the want of food) and food insecurity (limited or no access to adequate food sources) are worldwide pandemics. Recent statistics [29] indicate that more than 1.02 billion people (about 15% of the world's population) 'go hungry', i.e. are undernourished, everyday. More than 265 million of these people live in sub-Saharan Africa, with approximately 14.5 million undernourished people in SA [73]. Nearly all undernourished people live in developing countries.

Hunger and food insecurity are global crises that are on the rise. As recently as 2006, the estimated number of undernourished people was 854 million [29] - i.e. over the past four years, this number has increased by about 12%. Despite the best intentions of the world's humanitarian powers (United Nations (UN), World Health Organization (WHO), etc.), who have recognized the severity of hunger and its crippling effects on countries and the world, the problem is not being dealt with effectively. When the Millennium Development Goals were adopted by 192 UN member states and 23 world organizations in 2001 [21], the first of these was to eradicate extreme poverty and hunger by 2015. Unfortunately, this goal looks less attainable with every passing day.

What makes the problem of world hunger and food insecurity so perplexing is that it *can be prevented*. The world does produce enough food to feed everyone in it on a daily basis. In their 2009 annual report [29], The Food and Agriculture Organization (FAO) state that *"World agriculture produces 17% more calories per person per day than it did 30 years ago, despite the 70% population increase (since then)"* [73].

There is enough food to provide every person in the world with at least 2720 kilo calories (Kcal) daily, which is more than what is needed for basic human functioning [29]. So, what is the problem then? Surely if the world can produce enough food for everyone, then world hunger and food insecurity should cease to exist, provided that everyone can have access to this food? A

critical part of the problem is that food is wasted[1]. It is easier, cheaper and less time-consuming to throw away excess or unwanted food than it is to distribute it to other people. The logistics of distributing this 'waste' are also complex. But it can be done, with the advent of foodbanking as proof.

2.2 Foodbanking

2.2.1 What is foodbanking?

Essentially, 'foodbanking' is the process by which food, that would otherwise be disposed of as waste, is recovered and distributed to hungry people. Across the world, billions of tons of edible food is thrown away daily simply because it is in excess or it is near its 'sell-by-date'. Should this food be quickly and efficiently distributed to those in need, millions of hungry people could be fed on a daily basis. Foodbanking is an attempt to perform this task.

Although the practice of foodbanking differs somewhat from place to place, the basic 'building blocks' of a foodbanking system are the same everywhere. Usually, a foodbanking system consists of[2] (see Figure 2.1 below for a diagrammatic view of foodbanking):

- A group of donor organizations (retailers, manufacturers, producers), on both a national and local level e.g. Pick 'n Pay

- A 'foodbank' - usually a warehouse of some kind where food is sorted and stored for delivery (any single foodbank is usually part of a larger national network of foodbanks)

- A number of authorized social service organizations ('agencies') to which food is donated to feed the hungry

- People suffering from hunger and food insecurity

The social service organizations include: HIV/AIDS homes; soup kitchens; children's homes; school feeding programs and other non–profit organizations. Thus, a foodbank fulfils the role of *"obtaining, storing and transporting food in a safe and coordinated way to serve an entire community or large geographical area"* [60]. Food is either received or bought (usually at a reduced cost) from the donors and brought to the foodbank where it is sorted and stored. The food received can broadly be classified into perishable goods (those goods that would become inedible within a few days) and non-perishable goods (those goods that would still last at least a week or more). Perishable goods are typically received and delivered to agencies on the same day, or the next day. Food is either delivered to or collected by agencies on a regular basis – daily or weekly, depending on

---

[1]The waste of food is only *a part* of the problem. The real heart of the problem lies with government legislation, policy and spending on the poor and hungry. However, such an exploration is not within the scope of this study.
[2]Diagram obtained online from http://www.foodbanking.org/about/foodbanking.html
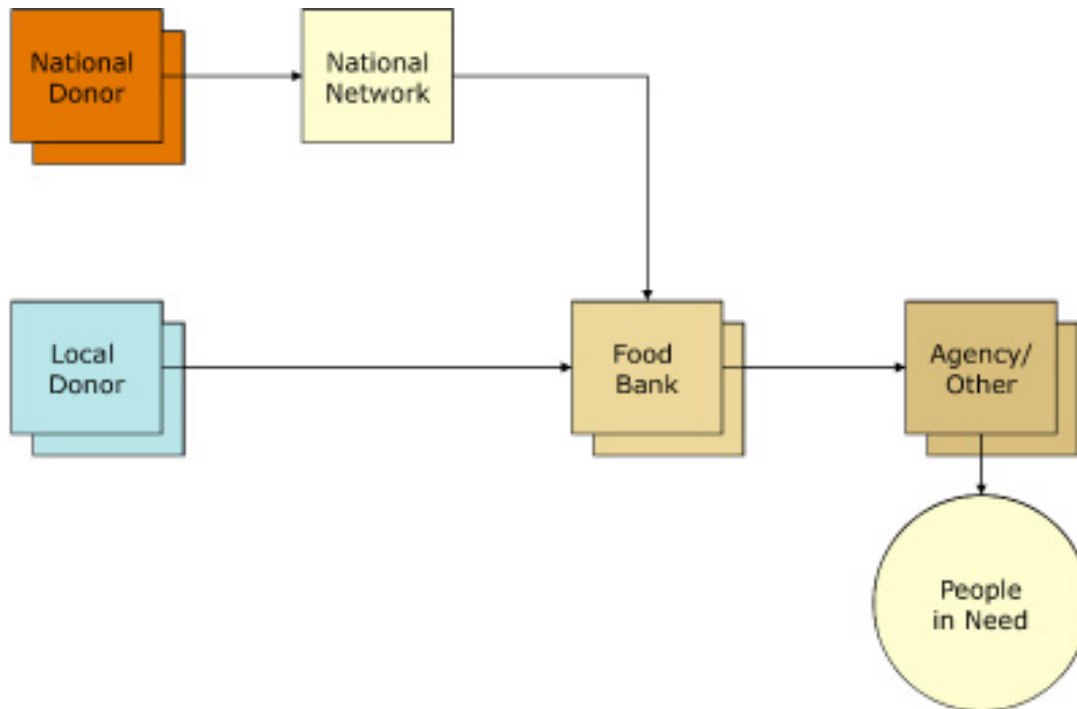
Figure 2.1: Diagrammatic view of foodbanking

the agreement between the particular agency and the foodbank. Although many foodbanks have programs that deliver food directly to communities, the fundamental purpose of a foodbank is to facilitate an adequate flow of food to hungry people through agencies [56].

Foodbanking is an innovative and revolutionary concept that is spreading across the world, holding much promise as an effective weapon in the war on hunger. Although the most notable output of foodbanking is the feeding of millions of people worldwide, it also serves as a vehicle to generate greater public awareness of and involvement in the fight to end hunger [55]. Foodbanking provides a powerful platform for the voice of those campaigning against hunger to influence food policy decisions and legislation to positively impact on the food security of impoverished communities and individuals. Foodbanking also indirectly combats other food-related problems like hunger-related diseases through the adequate provision and storage of clean, safe food [55].

Foodbanking as a practice has evolved during the four decades since its inception. Most modern foodbanks no longer solely consist of a warehouse that sorts and stores food for delivery to agencies – they often have innovative additional programs that provide hungry people with food directly. FBSA has two such programs: 'Lunch Buddies', which aims to provide children in underprivileged schools with a nutritious lunch and 'Fruit in Schools', where children in other schools are provided with a sandwich and a piece of fruit for their school lunch [31].

Why does foodbanking work?

The Global Foodbanking Network (GFN) outlines five reasons why the foodbanking concept is both popular and effective worldwide [56]. These are that Foodbanking is:

- *Universally supported – People everywhere recognize and respect the conviction that no one should go hungry. When it comes to hunger, there is no 'they' to oppose the 'we' who work to end it.*

- *Practical and efficient – Food banking appeals to the heart and the head; it feeds people while reducing waste.*

- *Scalable – Food banks can start at the community level and expand and network to feed a state, a nation and the world.*

- *Adaptable – Food banks can operate in different ways to suit different cultures and economies.*

- *Non-competitive – Food banking does not interfere with commercial channels of food distribution. It is an effective, cost-reducing outlet for businesses, governments and farmers.*

What is important to note here is that foodbanking has already proven to be successful (in the sense that it does work):

- In the United States (US), foodbanking has been practised for the past three decades, and there are now well over 300 foodbanks across the country.

- It has already made a significant impact in SA over the past 18 months [59].

There are now foodbanks operating in 18 countries worldwide, many of which are situated in developing countries.

2.2.2 History of foodbanking

John van Hengel is largely accredited as the 'father of foodbanking' [57]. Whilst working for St. Vincent de Paul, an international Catholic organization dedicated to fighting poverty and disadvantage in the US, van Hengel discovered that many grocery stores simply disposed of food that had passed or was near its expiration date. He acted quickly by arranging a meeting with a number of local stores' managers, pleading with them to send their unwanted food to St. Vincent de Paul [6]. Soon the volume of donated food was superfluous to the needs of the organization. This prompted van Hengel to develop the concept of a food 'bank', whereby individuals and organizations could make 'deposits' of food or money, and agencies could make 'withdrawals'. He approached St. Mary's Basilica with the proposal of using their building as a central location where agencies could access food for their 'clients' at no cost. Thus, in 1967, St Mary's Foodbank was established. In its first year of operation, St. Mary's helped to distribute more than 100 000 Kg of food to hungry people [6].

Soon foodbanks were established in other cities as the concept grew in popularity. A prominent businessman in the food industry, Alan Merret, made a significant contribution in developing the

resource network of foodbanks across the US [6]. In 1976, van Hengel founded Second Harvest (now known as Feeding America) - a consulting organization created to assist others interested in starting foodbanks. Today Feeding America has a network of more than 200 foodbanks across America and is recognized as the largest domestic hunger–relief organization in the US [6].

In 1986, van Hengel established Food Banking Incorporated to serve as a consulting organization to foodbanks around the world. It is now known as the GFN, and over the past two decades has assisted in establishing foodbanks in Africa, Asia, Australia, Canada, Eastern Europe and South America [6].

2.2.3   Foodbanking in South Africa

SA's foodbanking organization, FBSA, officially opened for business on 2 March 2009 with the commissioning of the FBCT warehouse in Philippi (just outside CT). However, foodbanking as a concept had already been practised for a number of years within the country. The problem of hunger and malnutrition has existed for decades in SA. It is a problem 'without borders' in that hunger has no favourite group of people or place in which to live. Within any major and minor urban or rural community there are people desperate for food in order to survive.

In response to this need, hundreds of hunger relief organizations have been founded across the country, many of which operate on a similar model to that of foodbanking. What is unique about FBSA is that it is a *nationwide* organization, with its aim the eradication of hunger in SA. To date, it is the single greatest hunger relief organization that SA has seen. It relies on the 'goodwill' of people (in the form of donations of food and money) and collaboration between key powers in the food industry and the government of SA. FBSA is not attempting to replace existing hunger relief organizations, rather they hope to build a nationwide network of foodbanks that will assist these organizations in combating hunger wherever they are situated. At present, there are five foodbanks in operation in five major cities in SA: Cape Town, Durban, Johannesburg, Pietermaritzburg and Port Elizabeth[3].

Essentially, FBCT was formed by the amalgamation of three separate organizations: Feedback, the Lions Feeding Scheme and the Robin Good Foundation. All three of these organizations had been operating with a similar model to that of foodbanking for a number of years. More importantly, people involved in these organizations were all united under a common vision - to see food being distributed to those in need within the greater CT region. This common vision played a vital role in promoting 'buy–in' amongst the relevant people from these organizations to the vision of establishing FBSA. The GFN played an important role in establishing FBSA, by conducting an in–depth feasibility study that included the establishment of the South African Forum for Food Security, securing support from the South African Government and drafting a memorandum of understanding with them that detailed how it would partner with FBSA in

---

[3]For the purposes of this study, all analysis and modelling is based on data obtained from FBCT. Moreover, FBSA has its head offices situated in Ndabeni in CT. Thus, attention will be levelled primarily at FBCT in this half–dissertation. However, it is hoped that the output of this study will be useful to all foodbanks nationwide.

developing foodbanks across the country [58].

FBSA has made significant strides in its war on hunger over the past year: in volume of food redistributed; increased collaboration with government and other key players in the business and private sector of SA; innovative strategic development, and establishment of new foodbanks across the country. At present, FBSA is providing more than 1.2 million meals a month to over 900 welfare agencies across SA [58]. In its first year of operation, it distributed 5.6 million Kg of food valued at R76 million to the needy at a cost (to FBSA) of less than R1 per meal. FBSA feeds approximately 66000 hungry South Africans everyday [16].

Foodbanking in the South African context

It is important to highlight elements of the practice of foodbanking that are unique to SA. The South African context is unique. Firstly, SA has the highest prevalence of HIV/AIDS in the world [22], with a large proportion of those suffering with the disease amongst the poorest socio–economic class in SA. It is thus no stretch of the imagination to reason that many people suffering from hunger in SA are also suffering with HIV/AIDS. The reason why this is an important issue is that people living with the disease have specific dietary requirements that would need to be accounted for in order to provide them with adequate nutrition. Secondly, the vast majority of SA's poor live in rural areas where poverty is rife. The level of unemployment in these areas is also very high, which further compounds the dire economic situation of these areas.

The above two points mean that in order for foodbanking to be effective in SA, the foodbanking network must extend into rural areas. Initially, FBSA attempted to address this issue by making foodbanks in urban areas 'responsible' (provide food for people and stimulate creation of new foodbanks) for the rural areas within a 300 km radius [4]. However, FBSA has now developed a comprehensive agriculture–based rural development programme, 'Agri–foodbank' [32] to not only bring food to the hungry in rural areas, but also to train and equip local farmers, thus creating sustainable food sources and employment within these areas[4].

The need for food aid to be accompanied by development programmes in rural areas is now widely recognized. Franklin and Harrel [33] (pg. 100) state that *"A newer concept to nutrition is that the nutritional well–being of a society is both a consequence and cause of the developmental processes within that society. The choices regarding the acquisition of food and other nurturing behaviour of individuals must therefore be given explicit consideration in the formulation of developmental assistance policies and in the design of developmental programmes and interventions, particularly those that seek to incorporate the poor into the measured productive processes of any economy"*. Elsewhere they note that, by itself, food aid has little nutritional impact and actually results in the 'client' spending less time on productive agricultural activities [33]. However, when coupled with integrated rural development programmes (e.g. farmer training and other skill–development programmes), food aid can

---

[4]Note that at the time of writing, the Agri-foodbank model had only been proposed to government and other key stakeholders. However, it was generating considerable interest and support, and plans for a pilot study are well under way.

promote sustainable development. Thus, with the proposed Agri–foodbank model, FBSA has in its vision for foodbanking in SA to be much more than just the redistribution of food to the hungry. Such innovative vision contributes to foodbanking in SA being unique.

Another issue that is prominent, albeit not necessarily unique, in SA is that of corruption and poor management of resources. This is a major obstacle in the way of any organization that wishes to partner with provincial or national government. It is difficult to deal with since it is rooted in the mindsets of individual people who are in positions of power. Fortunately, thus far FBSA has enjoyed substantial support from government and maintains a strong partnership with both local and national government departments. However, FBSA has no direct control over what support they receive as an NPO and this does pose a challenge.

It is essential that the mindset towards poverty of *all* South Africans is changed if there is going to be any significant progress made in the war on hunger. Fighting hunger in SA consists of more than the redistribution of food to the hungry (although this is obviously a crucial element). Any solution to the crises of hunger and malnutrition needs to be sustainable and should empower people to rise out of their state of poverty and hunger. A *sustainable* solution is one that promotes sustainable development through education (about food), training of farmers to produce their own food and providing a market for these farmers to sell their produce. A solution will only be sustainable if it is accompanied by a shift in the mindset of people towards poverty (government, food industry, suppliers, agencies, public, and even people within FBSA) which is probably one of the greatest challenges facing FBSA since they are trying to influence something that is intangible.

Chapter 3

Problem Definition

3.1 Introduction

The purpose of this chapter is to highlight a few of the many challenges facing FBSA/FBCT, select a subset of these challenges for focus in this study, and justify that selection[1]. What is hoped is that any reader contemplating undertaking research with FBSA specifically, or with any similar organization, will gain a greater understanding of some areas in which further research would prove useful.

It is important to emphasize here that FBSA is a new organization. It has only been in operation for the better part of two years. In many ways FBSA exists in a continual flux of change. There are at least two reasons for this. Firstly, FBSA is still in the process of establishing its 'identity' i.e. its short, medium and long–term goals; how the organization is structured; its strategies for combating hunger etc. Changes that improve the organization are being welcomed at present[2]. Secondly, FBSA has to change consistently to try to meet the demands of its 'clients', the hungry people of SA. When the long–term vision of an organization is the elimination of hunger in a country, one can understand that the demand on the organization always exceeds what they can supply. Expansion of the organization, both geographically and operationally, is thus a key ingredient of their strategy. However, FBSA recognizes that more than just the right scale of operation is needed – effective foodbanks and innovative development programs are also essential ingredients. With this long–term expansion and development comes change.

Thus, the challenges facing FBSA are numerous and varied. In fact, the nature of the challenges are continually changing. As one challenge is met, it opens the door to a different one. As

---

[1]Please note that this chapter was included specifically to identify a range of problems that FBSA was facing (for the possibilities of future research) and then describe the process of narrowing these problems down to the one we eventually chose. It was not included to describe what allocation problems are, but rather to indicate how we arrived at the decision to work on the allocation system at FBCT.

[2]Throughout the duration of this study, it was thus very important to hold regular meetings with representatives from FBSA/FBCT in order to be updated on any significant changes relevant to our research. Initially, the gap between these meetings was a month or more. However, it was soon discovered that a great deal can change within a month and thus more regular contact was sought.

Rosenhead [66] (pg. 761), referring to the responsive nature of management developments, put it: *"Broadly the successes of one advance sow the seeds of the next set of problems, requiring yet further innovations in organizational structure and management technique"*.

One thing that does not change is the large quantity of these challenges. It is impossible for any organization to consistently meet every challenge it faces, and it is certainly implausible for any masters dissertation to cover all the issues in which research would prove valuable. It is thus far more beneficial to adopt a 'laser–focus' approach and single out one or two issues for thorough research, than to attempt many issues half–heartedly with the hope of helping an organization reach workable solutions to these problems. History tells many a story of ambitious dissertations that promised much yet delivered very little that was helpful. Consequently, although it would have been possible to help FBSA with many of its problems, only a subset of these was chosen for the focus of this study. What follows is a summary of some of the issues identified in initial meetings held with FBSA, a detailed description of the issues selected for this study, and a justification of that selection.

3.2   Summary of issues

A series of meetings were held with FBSA during January–February 2010. The meetings served to establish a good relationship with members of FBSA and gain a good understanding of the types of issues the organization was facing, as well as promote their understanding of OR and how it could be useful to them. From the outset, it was conveyed that only a small subset of the challenges facing them would be 'taken on' in this study. The question was thus one of determining what problem/s would be amenable to an OR intervention that would prove useful to FBSA. As expected, there were numerous issues highlighted by FBSA at the very first meeting. These issues were explored at subsequent meetings, until a general agreement was reached as to what issues this study would have as its focus.

Some of the issues identified at the initial meetings were to:

1. Determine the optimum (i.e. minimum cost) vehicle routes for delivering and collecting food

2. Develop some form of an activity–based costing of meals (a minimum–cost model where drivers could decide to 'buy' a certain food delivery)

3. Determine the optimum locations for establishment of new foodbanks across SA (accessibility, number of people to be impacted and cost)

4. Develop a comprehensive 'poverty map' of areas in SA

5. Further develop their allocation model to address issues such as allocation policies and the efficiency of the allocation process[3]

---

[3]'Further' is deliberately used here to acknowledge the work of another University of Cape Town (UCT) MSc ORD

6. Assist with the management and maintenance of their fleet of vehicles (minimum–cost model)

3.3 Selection and justification of the problem for this study

Initially, an agreement was reached that this study would focus on issues 1 and 6. However, it was soon learnt that FBSA had previously been promised a student from Stellenbosch University through the Centre for Scientific and Industrial Research (CSIR). A meeting was held with a representative from the CSIR where a consensus was reached that the Stellenbosch student would be tackling the vehicle routing and maintenance since the OR department there has access to specialized routing software. Another reason for this decision was that routing–type problems tend to be very computational, i.e. more in the mould of 'hard' OR, while this study aims to combine 'hard' and 'soft' OR techniques in its approach to the problems it tackles. The use of combined OR approaches in the developed context is well documented and has yielded promising results. However, there is a dearth of similar research in the developing context, as noted by White *et al.* [87] (pg. 5): *"However, we note a lack of combinations of 'hard' and 'soft' methods, which may provide powerful applications of efficient planning alongside consideration of local aspirations"*. Evidently, there is a need for more such research, and it is hoped that this study will contribute to an emerging pool of similar studies in the future.

Issue 2 did not appear to be as important as the other issues, and by itself did not promise to provide a suitable scope for a masters half–dissertation. Issues 3 and 4 certainly seemed amenable to OR work, but after further exploration it was agreed that each problem was far too large in scope for a half–dissertation. However, these issues would provide exciting topics for a doctoral study.

Hence, attention was placed on issue 5, that of the allocation process at FBSA. Initially this issue was not highlighted by FBSA, but after bringing it to their attention and examining the current process, it was quickly agreed that there was great scope for work to be done in this area. Mention has already been made of the work done by another UCT student, Tim Blake [10], in formulating an allocation policy. Having had a positive experience with Tim, FBSA was keen to accommodate another student. A few more meetings were held in which the current allocation process was examined. A number of issues within the allocation process surfaced:

- Although an existing allocation policy was in operation, the need to develop a more robust policy, or set of policies, became evident

- The accuracy of food provided to agencies, fairness of distribution between agencies and time taken to allocate were highlighted as important criteria[4] to consider in improving the

---

student, Timothy Blake [10] helped to aid decision making at FBCT by developing a formal allocation system (amongst other things).

[4]After further exploration, it became obvious that the time to allocate and accuracy/fairness of distribution are 'competing' criteria, thus any allocation model would not necessarily be able to optimize both simultaneously.

current allocation process

- The tension between FBSA aiming to satisfy more of their current agencies' need and to support more agencies over time (i.e. allocation strategies)

- The unpredictability of supply over time, including possible seasonal effects

- The need for a more regular review of allocation policy

- Improving the speed at which the physical (in–warehouse) allocation takes place at the FBCT warehouse[5]

It soon became evident that there was ample opportunity to apply OR to some of the issues raised in investigating the allocation process. After subsequent meetings were held to 'flesh–out' these issues, it was decided that this study would have as its primary aim to develop a simulation model to formulate a range of robust allocation policies that would seek to improve the efficiency of the allocation process and the fairness and accuracy of allocations to agencies. In addition, a decision support system would be developed to help automate some of the decisions involved in the daily physical allocation process. This automation would be done with a view to decrease the time taken to perform allocations. It is clear that such a problem could be tackled with OR, and that developing such an allocation model would add great value to FBCT directly and FBSA indirectly.

---

[5]It is important to note here that the issues above were raised at different meetings over a period of a few months, with agreement on what the most important issues were only being reached over time. This is evidence of FBSA still undergoing changes as it continues to develop as an organization.

Chapter 4

Literature Survey

4.1 Introduction

ORD is an emerging field. Whilst OR work in development–type settings and on development problems has been carried out for a few decades, there is not a great wealth of literature on the subject [87]. The most pertinent research is usually found in the form of reviews or critiques [11, 87, 43], and virtually no research relating specifically to OR in food allocation problems in a developing context (developing countries and/or developing economies) was found.

The predominant area of food–related OR research in developing countries, particularly in so–called 'emerging nations' like SA, is in the agricultural sector [87]. Here research has focused primarily on predicting [15, 72] and optimizing [40] crop yields , with attention also being given to agricultural decision making [7], food security problems [72], and applying planning models in agricultural supply chains [5].

There is a wealth of literature on the application of OR techniques in a developed context (developed countries and/or developed economies) in dealing with issues such as the ordering [35, 34, 52], allocation [65, 26, 63, 64, 71] and storage of stock (inventory) [53, 38], and the use of simulation to improve warehousing practice and food systems [81, 62], some of which will be discussed here.

The 'spotlight' of this review of literature will be on the allocation of inventory (4.2) and simulation studies (4.3) in food–related problems, as these incorporate both the focus and method of analysis in this study. A brief discussion of the differences of approach between traditional OR and ORD will also be provided.

4.2 Allocation of inventory

The allocation of perishable items is an area that has received significant attention in the OR literature. This research includes studies on allocation in warehouse–retailer/customer systems

[12, 41, 26, 51, 64, 46] ; supply chains [80, 8] and general manufacture–distribution systems [28, 79]. It is of interest that there has been little research of a similar vein regarding non–perishable items. This is perhaps understandable in that non–perishable products, by their nature, pose a significantly smaller and less critical problem than do perishable products as the time constraint on allocation is far more relaxed.

What follows is a detailed discussion of two case studies pertaining specifically to the allocation of perishable items, as this is the major area of focus for this study, and one case study relating to the allocation of any inventory. After this, a review of other OR work conducted in this area will be provided.

4.2.1 Case studies

Optimal Issuing Policies for Perishable Inventory. Pierskalla and Roach [63] develop optimal issuing policies for certain types of perishable inventory problems. Their context is that of a perishable inventory that is divided into categories based on its 'shelf' age, where there is demand for items of each category. They consider a few contrasting objective functions and demonstrate that for the majority of these objectives the optimal issuing policy is of the First–In–First–Out (FIFO) type where the 'oldest' unit that satisfies the particular demand is issued first. The issuing of blood at a hospital or blood bank is used as the context for the study. A number of factors that are felt to influence both ordering and issuing policies for blood are listed as follows [63] (pg. 603) :

a) *The supply of blood is random, and comes from many sources such as volunteers, corporate blood plans, paid donors, etc.*

b) *The demand for blood is random*

c) *Much of the blood demanded is not used, and is returned to the inventory*

d) *Blood is a perishable commodity and, for practical purposes, it is assumed to deteriorate to lower freshness categories on a step function basis over a 21 – or 28–day horizon*

e) *The demand for blood of a particular category may be satisfied from a fresher category but not an older category*

Assumptions of model. The following assumptions are incorporated into the mathematical model used in the study (pg. 604):

1. *The items deteriorate over time on a step function basis*

2. *The demands for the items occur periodically and initially will be assumed to be known (i.e. deterministic)*

3. *Replenishment of the inventory may be made by items of any age*

4. *The quantity of items added to the inventory is assumed to be known, initially*

5. *The demand for an item of a given freshness level may be satisfied from the given level or any higher level, i.e., any 'younger' item in the stock*

6. *The model is dynamic in the sense that there is a time horizon of n periods considered (n is any positive integer)*

With regards to the demand process, two further assumptions are made. Firstly, if a stockout[1] occurs, then all demand is lost. Secondly, all demand is back–logged and then satisfied by the next supply of stock.

Model description. The mathematical model used in the study is detailed (on pg. 605) by describing the notation used in the study as follows:

- $n \equiv$ *number of periods the [allocation] process operates*

- $M \equiv$ *number of age categories for the deteriorating item*

- $p_j - p_{j-1} \equiv$ *length of age category j in periods for $j = 1, \ldots, M$ and $p_0 = 1$ (i.e., category j consists of all items of age less than p_j periods and greater than or equal to p_{j-1} periods)*

- $V_j \equiv$ *non-negative value of one unit of stock in [age] category j (by assumption 1, $V_1 \geq V_2 \geq \ldots \geq V_M$)*

- $D_{ij} \equiv$ *total demand filled for items of category j in period i*

- $I_{ij} \equiv$ *the non-negative inventory of items of category j remaining after demands are filled in period i*

- $R_i \equiv$ *cumulative value in period i of all filled demands (insofar as possible) plus the value of the stock on hand at the end of period i*

- $R_i = R_{i-1} + \sum_{j=1}^{M} V_j D_{ij} + \sum_{j=1}^{M} V_j I_{ij} - \sum_{j=1}^{M} V_j I_{i-1,j}, \quad i = 1, \ldots, n$

- $I_{0j} \equiv$ *initial inventory and $R_0 = \sum_{j=1}^{M} V_j I_{0j}$*

- $R = (R_1, R_2, \ldots, R_n)$

- $S_{ij} \equiv$ *non-negative stockout of items of category j at the end of period i*

- P *is any policy which states which items are to be used to fill the demands*

P represents a feasible allocation policy if all demands are satisfied whenever stock is available to satisfy these demands. FIFO represents the feasible policy in which the 'oldest' item satisfying assumption 5 is allocated, while LIFO (Last–In–First–Out) represents the feasible allocation policy in which the 'youngest' item satisfying assumption 5 is allocated.

Three different objective functions are considered in the study:

1. Maximize Total Current Utility (TCU), where
   TCU = value of all previous satisfied demands + value of items currently in stock

2. Minimize the number of back–logged items and, in the case of a stockout, minimize the total number of lost demands

3. Minimize the total amount of items that reach the last age category (the oldest items that may be assumed to be obsolete)

The study successfully demonstrates that the FIFO policy is optimal in all demand cases for all three objectives, except in the case of lost demands where it is not optimal for the first objective. Further generalizations and extensions of the model are discussed, in particular extending the

---

[1]Here, a 'stockout' occurs when there is no inventory and no supply of items.

model to allow for both the demand and incoming stock processes to be random. It is noted that, provided both processes are assumed to be independent of the allocation process, the FIFO policy would still remain optimal as above.

Applicability of approach to our study. Pierskalla and Roach's [63] study serves as a good example of how traditional hard OR techniques were applied in an allocation–type problem[2]. In examining the list of factors that influence the ordering and allocating policies, some are applicable to perishable foods, in particular the fact that the supply of foods to FBCT is random (a) and that perishable food does decay[3] (d). However, there are significant differences. In the context of FBCT, one cannot assume that the demand for food is random (b). On the contrary, the demand is constant because it is ever–increasing and *always* exceeds the supply. Also, since the demand always exceeds the supply, there is never any 'returning' inventory (c). Despite these differences, it is interesting to note that the FIFO approach shown to optimize most of the objectives was precisely the policy in operation at FBCT for allocating their perishable inventory at the beginning of this study.

Optimal Myopic Allocation of a Product With Fixed Lifetime [64]. Prastacos develops optimal myopic[4] allocation rules for a perishable product for two classes of allocation policies in a one–distribution centre, n–demand–location regional system. The two classes of allocation policies considered are:

a) Rotation, where any unused allocated stock at a demand–location remaining at the end of an allocation period is returned to the regional distribution centre to be re–allocated in the next period, provided that it is not deemed obsolete

b) Retention, where any unused allocated stock at a demand–location at the end of an allocation period remains there and is not returned to the distribution centre

Assumptions of model. The following assumptions are made in the formulation of the model used in the study [64]:

1. Random amounts of a fresh perishable product are acquired (either produced or sourced) by the regional distribution centre

2. After the the product is acquired, it is distributed to n demand–locations throughout the region

3. The time between two successive 'acquisitions' is constant and denotes one (allocation) period

---

[2]It is obvious that the study was conducted prior to the 'Ackoff Revolt' of 1978–79, when the concepts of 'hard' or 'soft' OR did not exist.

[3]This rate of decay is far more rapid in the case of the perishable food that FBCT receives, which has a 'shelf' life of at most a day.

[4]Here 'myopic' refers to allocation rules that allocate stock in such a way as to minimize the total cost incurred by the regional distribution centre per period of allocation.

4. The product has a fixed lifetime, equal to L periods

5. There are no time delays in any delivery process (i.e. any allocation)

6. Each demand–location stores the product in order to meet a random demand in a particular period. These demands are assumed to form a set of independent and identically distributed random variables with a mean $< \infty$

7. The value of the product remains constant over its entire lifetime

8. Each demand–location uses a FIFO issuing policy to satisfy its demand

Two further assumptions relating to costs incurred by the distribution centre are included:

1. If the demand exceeds the stock at a particular location, this excess demand is either lost or supplied (during that period) from outside the region at unit cost s, and no transferral of stock between demand–locations is allowed

2. If a unit becomes obsolete, it is discarded at a unit cost w

Model description. The model used in the study is formulated as follows [64]:

- $Q^{(t)} \equiv$ amount of fresh product available at beginning of period t

- $D_k^{(t)} \equiv$ random demand to be satisfied at demand–location k during period t, $k = 1, 2, \ldots, n$

- $F_k(x) = P(D_k^{(t)} \leq x) \equiv$ the (stationary) probability distribution of demand at location k

- $S_k^{(t)} \equiv$ amount of product short at demand–location k in period t

- $W_k^{(t)} \equiv$ amount of product outdated at demand–location k in period t

- $X^{(t)}(j) \equiv$ amount of product of age j at the distribution centre at beginning of period t, before allocation is made

- $Y_k^{(t)}(j) \equiv$ amount of product of age j at the demand–location at beginning of period t, before allocation is made

- $\mathbf{X}^{(t)}$ and $\mathbf{Y}_k^{(t)}$ are the respective inventory vectors

As fresh product is only available at the distribution centre, we have:

$$X^{(t)}(0) = Q^{(t)}; \quad Y_k^{(t)}(0) = 0.$$

The objective is to minimize the total cost to the distribution centre *per period,* which is expressed mathematically as: Minimize

$$C(\mathbf{X}^{(t)}, Y^{(t)}; \pi) = sES(\mathbf{X}^{(t)}, Y^{(t)}) + wEW(\mathbf{X}^{(t)}, Y^{(t)}),$$

where π represents the allocation rule for the allocation of the inventory $\mathbf{X}^{(t)}$.

The study shows that the optimal allocation rule for a rotational class of policies when there is continuous demand is to *"equalize the probabilities of stock becoming outdated across all the demand–locations to the lowest value (that is consistent with regional constraints) by allocating stock that is subject to outdating; then equalize the shortage probabilities across all demand–locations to the lowest feasible value by allocating the remaining quantity"* [64] (pg. 908). An algorithm that is a close approximation to the above rule is developed for rotational policies with discrete demand and is also demonstrated to be optimal.

For the retention class of policies, the assumption is made that a demand–location keeps all products it receives until they are used or become outdated. This in turn implies that the expected quantity of product to outdate in any given period is independent of the allocation rule, since the $L - 1$ period's old quantities form the inventory for the demand locations at the beginning of every period. The objective is thus simplified to minimize the next period's expected shortages. A very close approximation to the rotational myopic rule above is used to equalize the shortage rates of the current period, which in turn minimizes the expected shortages of the following period, proving to be the optimal myopic allocation rule for retention policies with continuous demand[5]. The optimal myopic allocation rules are extended for cases of exponential, normal, uniform and poisson demand processes.

Prastacos then lists the following features of the optimal allocation policy (for the rotational class of policies considered above) [64](pg. 912–913):

1. *It minimizes both the expected shortage and the expected outdate costs of next period, and not just the sum of the two. In managing the inventory of one individual location there is a 'tradeoff' decision point that has to be established between the two costs, on the basis of the unit shortage and outdate costs at that location. No such decision is to be made by the regional centre.*

2. *The optimal allocation in the region is independent of the unit costs s and w. This is a very significant result since the estimation of these unit costs in many cases is purely subjective.*

3. *The form of the optimal policy remains the same for both cost considerations and for both classes of policies examined.*

4. *The rule is simple to implement, and can be computed in closed form for most probability distributions.*

Applicability of approach to our study. The cost–minimization approach used is very much in the classic hard OR mould and is not particularly relevant to our study, as our objective is not primarily to minimize costs incurred to FBCT, although it is certainly hoped that by aiding their decision–making with regards to allocation, FBCT may be able to operate more efficiently. However, the regional one–distribution centre (FBCT), n demand–locations (agencies) context is a mirror of the context in which FBCT operates which makes the results of the study worthy of consideration. The rotational class of policies is irrelevant to our context as none of the agencies' needs are ever fully satisfied (so great is the demand) and thus no stock is returned to the warehouse.

---

[5]Prastacos indicates that this rule can also be approximated for discrete demand.

The retention policy is a more accurate representation of the allocation process at FBCT as, if we take each period to be a single day, FBCT receives the majority of their fresh produce each morning (beginning of the period) which is then allocated on that day. The use of an objective function that incorporates the allocation rule is particularly pertinent to our study as we will attempt to optimize a set of rules (policy) via a simulation study. That the optimum myopic rule derived in the study is independent from the unit costs (2) and transferable across the different classes of policies (3) are attributes that we will aspire to when we develop a set of optimal allocation policies. The 'ease of implementation' (4) of the optimal rule is considered to be a very important property that this study will endeavour to replicate.

Two–interval Inventory–allocation Policies in a One–warehouse N–Identical–retailer Distribution System [46]. McGavin *et al.* developed a model to determine optimal allocation policies that minimize lost sales per retailer ('lost sales/retailer') between warehouse replenishments, assuming a fixed warehouse–replenishment schedule (replenishment cycle) [46] and stochastic demand for a single product. They define an allocation policy by four key decisions [46] (pg. 1092):

1. *The number of withdrawals from warehouse stock, where each withdrawal is an opportunity to allocate the withdrawn stock to any (or all) of the N retailers*

2. *The times between successive withdrawals, which divide the replenishment cycle into intervals*

3. *For each withdrawal, the quantity of stock to be withdrawn from the warehouse*

4. *For each withdrawal, the division of withdrawn stock among the retailers*

The study assumes that decisions 1 to 3 are made when the warehouse is replenished, whilst decision 4 depends on the retailer inventories when stock is withdrawn.

McGavin *et al.* note that the majority of research conducted into inventory–allocation policies has revolved around so–called 'ship–all' policies where there is a single withdrawal of all warehouse stock immediately after replenishment, and thus they proposed to investigate *'two-interval'* policies, where two withdrawals are made from warehouse stock, focusing on the withdrawal quantities and times, and the division of withdrawn stock among retailers.

Assumptions/description of model. The following assumptions are made in the formulation of the model used in their study:

- Warehouse stock is replenished solely from an outside supplier at the beginning of a replenishment cycle, i.e. neither the warehouse nor the retailers have any inventory at this time

- The first withdrawal of stock takes place immediately after the warehouse has been replenished

- The delivery to retailers is immediate

- During the first interval (time between the first and second withdrawals) each retailer satisfies random demand for the product from its inventory, and then first–interval lost sales/retailer are totalled across all retailers

- At the onset of the second interval, the warehouse determines current retailer inventories and then divides remaining stock between them

- Each retailer attempts to satisfy a random demand for the product during second interval, and the second–interval lost sales/retailer are totalled, after which total lost sales/retailer are determined

- The intervals are not necessarily equal

- Each demand process is independent from and identical to the demand process at every other retailer

- Demand for the product at each retailer is generated by a known stochastic process with stationary and independent increments

- Inventory holding costs are ignored

The study shows that, in order to minimize the lost sales/retailer between warehouse replenishments, the optimal allocation policies are those that seek to divide available stock to maximize the minimum retailer inventory across all retailers i.e. warehouse stock is allocated to bring all retailer inventories to some stationary base–level at the beginning of a period, but if there is insufficient warehouse stock, then all stock is distributed so as to maximize the minimum inventory levels across all retailers. This holds for any given withdrawal quantity and interval length. Such an allocation is called a *'balancing division'* [46]. Other interesting results obtained by conducting a comparative numerical study of the optimal two–interval allocation policy and other policies (e.g. ship–all) were that:

a) The risk–pooling[6] benefits of the optimal two–interval allocation policy over a ship–all policy are scenario–specific. Those scenarios where the demand uncertainty is great and the service level[7] is high typically demonstrate significant risk–pooling benefits

b) With regards to the optimal allocation policy itself, greater demand uncertainty generally led to a greater first withdrawal and shorter first interval; whilst higher service levels led to a greater first withdrawal and longer length of the first interval

McGavin *et al.* also developed an *'infinite–retailer'* model which is used to construct optimal two–interval allocation policies for a distribution system with infinite retailers. Apart from the number of retailers, it is identical to the two–interval model described above. The model is then used to derive two heuristic policies for finite–retailer systems. Computer simulation was used to compare the optimal two–interval allocation policies with the two heuristic policies based on the infinite retailer model with respect to the expected lost sales/retailer. The results of

---

[6]Note that distribution systems 'pool' risk by stocking retailers through warehouses [46].

[7]A measure related to the number of lost sales/retailer. High service level corresponds to low number of lost sales/retailer, low service level corresponds to high number of lost sales/retailer.

the simulation study suggested that the infinite–retailer heuristics *generally*[8] provide excellent allocation policies for N–retailer systems.

Applicability of approach to our study. The study is a good example of the use of simulation within an OR–type intervention. By necessity, FBCT's perishable allocation policy closely resembles the 'ship–all' type as the perishable goods that come in are usually allocated on the same day otherwise they will become unsuitable for consumption. However, investigating the use of a two–interval allocation policy is certainly something to consider when the simulation model used in our study is developed. This could be particularly pertinent in modelling the allocation of perishable and non–perishable goods together, where there would be essentially two withdrawals (one for perishable and one for non–perishable) per day.

The major differences to note are that the study assumes that: the retailers are identical, which is not true for the FBCT distribution system in which no two agencies (retailers) are the same; that stock is simultaneously distributed to all retailers, which does not account for a separation of distribution to retailers by day of the week, which is the case at FBCT where certain agencies are supplied on certain days of the week; and that, although the demand at the various agencies supported by FBCT may vary from week to week, it is always far greater than what is supplied[9] and is thus essentially viewed as constant[10]. The approach is also one of minimization of costs incurred by the agencies which, as noted above, is not the focus of our study.

The concept of a 'balancing division' approach to allocation is something that will be investigated in our simulation study. The versatility of the optimal two–interval allocation policy, in that it applies for any given withdrawal quantity and interval length, is important for its applicability to our study, as the range of allocation policies developed for FBCT will need to be robust.

Of interest are the results (a) and (b) obtained from the comparative numerical study of the optimal two–interval allocation policy with other policies. Since the demand for food is so colossal, there is very little demand uncertainty and the service level provided by FBCT is very low[11]. Thus, by (a), the implication is that a two–interval policy would not necessarily demonstrate significant risk–pooling benefits. It would be of interest to observe how a two–interval allocation policy performs under the conditions of low demand uncertainty and low service level.

4.2.2 Other literature

Prastacos extends his work in *Allocation of a Perishable Product Inventory* [65] to tackle the problem faced by the regional distribution centre when it seeks to allocate stock in the region

---

[8]There are certain scenarios, such as those with small numbers of retailers, where the heuristics do not perform particularly well.

[9]The demand processes at the various agencies in the FBCT system can be viewed as identical in this regard.

[10]For the purposes of the simulation run in our study, it may be assumed to be a large random variable.

[11]At present, FBSA aims to provide each agency with one meal per person per day. This would equate to only satisfying one third of the demand at each agency, if we assume that a normal person eats three meals a day.

in such a way that it is used most efficiently at the demand locations. This is an important consideration[12] but not one that is currently a priority at FBCT as their focus is on the efficiency of the allocation process and fairness of allocation to agencies. Federgruen *et al.* [26] developed a model, based on Prastacos's [64], that incorporates the transportation costs for two types of deliveries to the demand–locations: individual deliveries to each demand–location and deliveries to a combination of demand–locations along multi–stop routes performed by a fleet of vehicles from the regional distribution centre. As indicated in section 3.3, the vehicle routing and transportation are not topics that will be covered in this half–dissertation, but it is of importance to note that FBCT currently employs the multi–stop delivery pattern.

Epstein *et al.* [23] detail how a combinational auction procedure, based on an integer linear programming model, resulted in substantial improvements in the provision of meals to school children in Chile (a developing country). They note that in Chile more than 30 percent of children under age 18 live below the poverty line. Thus, in order to make the concept of 'equal opportunities for everyone' a reality, Chile needs to compensate those children who have suffered from social deficits (i.e. previously disadvantaged). This is achieved through welfare programmes that provide free meals to school children in lower socioeconomic regions, with the hope that this will lower school absenteeism and drop–out rates and improve school performance. Chile's school system employed mathematical modelling to design the combinational auction process, in which firms 'bid' for contracts to feed, in total, 1.3 million children from low–income families at their schools across the country. Essentially, a 'combinational' auction process is one in which firms can present multiple bids for contracts to supply meals in different regions and at different levels of nutritional support for a certain price. Hence, each bid represents a combination of regions, support levels and price. The use of the model revolutionized both the auction process and the provision of meals to school children in the following ways:

- It resulted in a transparent and objective auction process, and encouraged competition amongst bidding firms

- Firms could compose flexible regional bids that could include their scale economies, resulting in a more efficient allocation of meals

- An optimal assignment of contracts (i.e. a solution) was found by the model[13]

- The price–quality ratio of meals was improved, amounting to yearly savings of approximately US$40 million, which is equivalent to the cost of providing meals to 300 000 children per year

- Both the nutritional quality and food structure of the meals provided was improved

- The infrastructure of the meals service and the working conditions of firms' employees in the schools improved

---

[12]This consideration is one that would most probably grow in importance as FBCT meets the initial challenges it is facing.

[13]The entire problem was NP-complete with more than 10000 binary variables, so this was quite an accomplishment!

The study is an excellent example of the successful use of OR to improve decision–making in the allocation of food in a developing country[14]. Epstein *et al.* [23] (pg. 11) state that mathematical tools for decision making *"give decision makers better analytical capabilities and a deeper understanding of their problems, and perhaps more important ... can provide transparency and encourage competition. These are key issues in achieving efficiency and improving the allocation of resources, which can directly improve the quality of people's lives"*. It is hoped that our study will augment both the transparency and efficiency of FBSA's allocation process, and thus indirectly impact the quality of life of thousands of South Africans.

Muckstad and Roundy [51] developed a mathematical model and corresponding algorithm to tackle the problem of planning economic delivery, i.e. allocation, intervals for k items in a one–warehouse, N–retailer system where the demand for the product is constant. The model incorporates holding (inventory) costs, and fixed ordering and placement costs incurred by the warehouse during the delivery process. It assumes that the warehouse has limited delivery capacity and that equally–spaced delivery times are desirable. No back–ordering is permitted. The problem is structured using graph theory and results in a non–linear integer programming problem which is solved via finding the optimal solution to its continuous relaxation.

Hill [41] used a simulation study to estimate the order of magnitude of benefits to a company in a warehouse–branch retail chain of introducing a stock–allocation policy. He considered four allocation policies (A, B, C and D), ranging in complexity from a common practical 'first–come–first–serve' policy (A) to a complex policy incorporating branch inventory reviews and the estimation of expected lost sales (D). The simulation compared the benefits of policies B, C and D to the 'base case' policy A. He concluded that the introduction of a more sophisticated allocation system (i.e. B, C or D) would result in increased warehouse service levels, but the order of magnitude of this benefit would depend on the specific parameters to which a company was operating.

4.3   Simulation in other food–related problems

Simulation has been used in the food industry in fast–food chains [61, 78], automated food plants [62], planning distribution systems [79] and even in the spray–drying of food products [77]. Van der Vorst *et al.* [81] developed a method to model the dynamic behaviour of food supply chains and assess different designs of the supply chain framework and operational management by employing discrete–event simulation . They considered a typical producer–distribution centre–retail outlet supply chain. The simulation was aimed at aiding the decision–making processes involved when redesigning a chilled–food–product supply chain, by assessing the effects of varying the barometers of logistical performance. They posit that improvements to supply chain performance are most often hindered by uncertainties relating to decision making (e.g. uncertainties related to supply, process and demand). Van der Vorst *et al.* draw on Silver *et al.*

---

[14]This was the only study relating specifically to the allocation of food in a *developing* country that was found in all the Operations/al Research Journals available to the author.

[75] in asserting that quantitative models must be more realistic representations of a problem in order to become more useful to managers. They argue that models should incorporate some of the common 'givens' as decision variables. The results of the simulation indicated that, for the chilled–food–product supply chain, system performance was markedly improved when: new information systems were introduced, ordering and delivery frequencies were increased, and the producer's lead time[15] was reduced. However, they do concede that *"it is impossible to predict the exact benefits that would actually be obtained, because it was impossible to simulate human interventions"* [81] (pg. 364). This 'human factor' will be important to bear in mind when analysing the results of our simulation study.

Martin [45] developed a simple algorithm, based on the nearest–neighbour heuristic, to simulate travel and delivery times for vehicle routes for a centralized bakery. The bakery was seeking help with the problem of selecting new routes at minimal cost and in the shortest time possible. He found that designing optimal minimum–distance routes did not significantly reduce costs incurred by the bakery, but replacing the physical testing of new routes with a simulated route testing saved the bakery both time and money. He emphasized the importance of the transparency of the model developed, stating *"The clients rejection of unfamiliar or overly complicated solutions dictated a simple and familiar solution"* [45] (pg. 40). He implies that the real challenge of the intervention was more one of getting management to trust the simulation model enough to use it. The study is an excellent example of how a simulation model can be both effective (in that it achieved the objectives of the study) and transparent (in that it was easy to understand and use). Such a simulation model is what we are hoping to build as part of the output of our study.

Pidd [62] describes the use of simulation in two case studies of automated food plants. In the first study, discrete–event simulation is used to improve the overall efficiency of a plant. Pidd notes that most of the problems in automated food plants arise from the dynamic interactions between components of the plant, and posits that simulation provides a good environment for experimenting with changes in these interactions without incurring too much loss (in time or produced product etc.). He also provides a three–part typology (in the form of questions Operational researchers (ORers) should ask themselves) to aid the choice of type of simulation study to be done [62] (pg. 685):

- *Time handling? – Should time be moved forward in fixed increments or between events?*

- *Stochastic or Deterministic? – Generally, some of the behaviour of a system has to be modelled stochastically.*

- *Discrete entities or Continuous variables? – The temptation is to proceed with the assumption that all variables are continuous. However, all entities in the system should be carefully examined, as first impressions can be deceiving. A combined continuous–discrete simulation may be the way forward.*

Such a guideline will be useful when it comes to deciding on the type of simulation model to be employed in our study. It is of interest that Pidd indicates that the simulation program was designed to allow engineers at the plant, who had little knowledge of simulation models, to experiment with the operation of the plant to improve its performance. Once again, the

---

[15]Here, 'lead time' refers to the time between beginning and completion of the production process.

importance of the transparency (see section 5.2) of methodology is evident. He also mentions that much of the benefit of the OR intervention was derived in coercing the client group, via the simulation model, to adopt a more analytical approach to the problem.

In the second study, simulation was used in the design of an integrated plant, as well as ensuring that a particular plant design should result in the plant operating to a specified performance level. Pidd does note that simulation models form only a part of the design and operation of these plants, and cannot incorporate all aspects of plant design and operation.

4.4 Traditional OR vs. ORD

Mention has been made of the more traditional 'hard OR' approach, both in the literature reviewed above and in stating that our study will adopt a less conventional combined 'soft-hard' OR approach. What is also crucial to emphasize are the differences between traditional OR and ORD[16] that necessitate a different approach when conducting an OR intervention in a developing context. What follows is a brief summary of the key differences between traditional OR and ORD, taken from an essay[17] written by the author titled *'What Distinguishes Operational Research for Development from Operational Research as a More General Approach to Management Problems'* [84].

The primary differences between ORD and OR exist in:

- The contexts in which they are applied, i.e. the developing and developed contexts

- Their objectives

- The roles that the various tools employed by ORD and OR assume

- Their approaches to the problems they face

The contexts within which ORD and OR are applied are vastly different. Stewart [76] (pg. 1) posits that, while there are obvious similarities in OR approaches in all contexts (i.e. between the OR and ORD approaches), there are *"unique contextual features in developing regions that give rise to a need for special approaches or at least a change in emphasis"*. The lack of infrastructure in the developing context leads to uncertainties surrounding the availability and quality of various systems (transport systems and other public systems) that are taken for granted in the developed context where the focus of OR is in improving the efficiency of already existing infrastructure [76]. The chronic shortage of resources and uncertainties surrounding their availability and delivery is unique to the developing context. Other uncertainties such as the lack of: adequate data, research support and proper understanding of problems by leaders occur to a greater degree in the developing context [36]. The time pressures experienced

---

[16]ORD should be considered to be the *appropriate* application of OR to development problems *anywhere,* with the primary purpose of promoting sustainable self-development of those people affected by the problem at hand.

[17]This essay is available by request from the author of this dissertation. Please contact him via email at: neilmarkwatson@gmail.com should you wish to obtain it.

by those working in the developing context are unique from the perspective of the seriousness of the consequences of running over time as the very livelihood of people can be affected [82].

The social climate in the developing context is unique, where 'what the State says, goes'. This results in resistance to changes of any sort - such as the introduction of new methods and approaches to problems [36]. The dependency of less developed countries on developed countries for resources etc. hinders the development of local OR communities, and the existing decision-making culture is inimical to analytical formulation and analysis [36]. Higher degrees of corruption, conformity and time-wasting are more prevalent in the developing context [37].

The cultural context in developing areas is unique. Community-based societies are more common in the developing context. Unique challenges are posed by cultural differences such as language barriers to effective communication of ideas, strong traditional mandates and beliefs of communities and a more acute distrust of outsiders [76]. This distrust often emanates from the lack of transparency of approach which generates limited or no understanding amongst the 'client'. With this in mind, the importance of effective communication between helper and client/s in an intervention in the developing context is inarguable regarding the client's understanding of the intentions of the helpers, and the helpers understanding of the needs and desires of the client. This is emphasized by Stewart [76] (pg. 4) who writes: *"There needs to be continual interaction between planners, modellers and stakeholders in striving towards a shared understanding and consensus of what the most desirable plans or courses of action are"*.

ORD and OR differ in their objectives. ORD is chiefly concerned with promoting sustainable self-development across all spheres of society. OR is preoccupied with improving the efficiency of existing systems; whilst ORD seeks to assist in helping to provide these systems and improve their effectiveness. There is a greater need for both the practical application of OR in the developing context and transparency of the OR methodologies utilized in this application. The degree of transparency of any OR process is directly related to the degree to which it will be understood by those it seeks to help.

The roles that various OR tools assume in ORD and OR are different. In the developing context, there is a greater need for an OR tool to be more than just a part of the OR process, and for it to be effective in tackling problems in situations where there is often a severe shortage of technology. It is critical that OR tools are relatively simple and transparent, and amenable to the facilitation of community involvement for them to be effective in the developing context. The extent of the need for simplicity and transparency of decision models is unique to the developing context.

OR and ORD differ in their approaches. In light of the various uncertainties facing communities in the developing context, along with the greater emphasis on community-based development, the importance of the active involvement of the community in any intervention becomes even more apparent. Thus, any OR tool/methodology applied within this context must incorporate facilitating this involvement as part of its role. This is not necessarily unique to the developing context in that many OR tools applied to more general management problems seek to facilitate the involvement of all stakeholders. However, the need is more acute in the developing context.

Historically, OR has tended to be more reactive than proactive in its approach to problems. The need for a more proactive OR approach has been recognized, and the need for ORD to be proactive in its approach is more pressing than in OR in the developed context [82]. This is particularly apparent with respect to ORDs drive towards the promotion of sustainable self-development, where forward-thinking and planning is essential to ensure any measure of sustainability. Whilst the optimization paradigm of OR has become far less prevalent in the modern day practice of OR in general, there appears to be less scope for its use in ORD than in OR.

4.5 Reflection on literature

It is important to note here that the approaches employed in the majority of the literature reviewed are in the classical OR mould – that of optimization models with minimization of costs/maximization of profits as their objectives. As stated above, this study will employ a combined soft–hard OR approach. However, the approaches and models considered in this chapter do provide useful information and guidelines to consider when building the simulation model for our study.

Our approach will also endeavour to take into account the unique aspects of the developing context, as outlined above. It is recognized here that the environment within which we are operating does not necessarily incorporate *all* of the unique characteristics of the developing context (e.g. there is not a chronic shortage of technology at FBCT), but certain features (e.g. simplicity and transparency of approach, objectives, unique uncertainties, community involvement) are pertinent to the context of our study. The objective in our study differs considerably from the more traditional OR approaches in that it is not expressly related to cost or profit. Rather, the principal aim of the study is to aid decision–making at FBCT with regards to allocation decisions, in the hope that the allocation system will become more efficient, fair and accurate. Should FBCT derive financial benefit as a result of this, that would be a bonus.

Chapter 5

Problem Structuring

5.1 Introduction

The problems facing organizations today are by nature ill-defined and 'messy' [3]. It is very seldom that one is faced with a problem to which it is easy to see the 'right' answer/s. The reality is that most problems are complex, involving people with differing *'weltaanschauung'* ('worldviews') [14] i.e. possibly conflicting views of the problem and/or interests in a particular outcome. Consequently, any solution offered to a problem is rarely consummate. In theory, there exists an 'optimal' solution (that can be found), but in reality no solution is ever completely 'optimal' (at least not over time). Problems are seldom completely solved, rather they are 'alleviated' [13] or 'finished' [18]. Today, many ORers prefer to speak of 'problematical situations' [48], acknowledging the innate presence and influence of humans (their worldviews, desires, behaviour) as both providing and influencing the context within which problems arise[1]. The use of 'problem structuring' – exploring the problematical situation using any number or variety of tools in order to give it more 'structure' – as a method of gaining a greater understanding of the problem at hand is thus invaluable with regards to:

a) Ensuring that the *right* problem is tackled, and

b) Having a sufficient understanding of the right problem in order to find an appropriate 'solution'

Dewey's maxim, *'A problem well put is a problem half solved'* [17] has proven to be more truthful than witty, as many ORers could attest.

In order for an OR intervention to be successful it is essential to gain a good understanding of the problematical situation before developing an appropriate solution. If the problem at hand is not explored and structured before an attempt is made to solve it, one runs the risk of either solving the wrong problem or finding a suitable solution which has little relevance or applicability to the target audience ('target audience' here referring to those people who will be affected by whatever action is taken in solving the problem).

Whilst more theoretical OR research is indeed valuable, when it comes to an OR intervention, especially in the developing context, there is a heightened need for appropriate action to be taken to solve the

---

[1]The term 'problematical situation' will be used to refer to both the actual issue/s to which a solution is sought (the 'problem') and the human context of the situation within which the problem has arisen.

problem at hand [76]. Bearing this in mind, one must be careful not to act too soon without an adequate understanding of the problem. The skill for an ORer is to find the right balance of time between careful structuring of the problem and taking action. Both elements should be present, even though the majority of people evaluate any intervention primarily by the action taken in solving a problem and the results of that action, with the initial problem structuring work going largely unnoticed.

'Problem structuring' as an idea began to surface in the mid-1960's amongst European and United Kingdom ORers. Actual 'problem structuring methods' (PSMs) were developed and formalized in the 1980's. The need for problem structuring was birthed in the 60's in response to a dissatisfaction amongst many ORers with the growing realization that OR was becoming increasingly ineffective in its ability to tackle the real-world problems of the day. As Rosenhead states: *"OR was thrown up by a situation where traditional management methods were proving inadequate to handle the growing complexity of organizational arrangements. Problem structuring methods in turn were generated out of a sense that the trajectory of OR had led it away from important areas of social decision-making."* [66] (pg. 759).

Over the past three decades, PSMs have grown in both theory and application. Rosenhead and Mingers [67] (pg. 842) posit that presently PSMs are *"characterized as a family of methods for supporting decisions by groups of diverse composition within a complex environment to agree on a problem focus and make commitments to a series of actions"*. Rosenhead [66] (pg. 762) purports that PSMs are *"appropriate for situations characterized by multiple actors, differing perspectives, partially conflicting interests, significant intangibles and perplexing uncertainties"*. There are many different PSMs – Soft Systems Methodology (SSM), Soft Systems Dynamics Methodology (SSDM), Strategic Choice Approach (SCA), Strategic Options Development Analysis (SODA), Viable Systems Model (VSM), Decision Analysis (DA) and Multiple Criteria Decision Analysis (MCDA) – as well as combinations of parts of these methods being practised by ORers today. Rosenhead and Mingers [48] conjecture that over the past three decades SSM, SODA and SCA have become distinctly known and extensively applied[2]. PSMs and are now viewed by most ORers, excepting many in the US [66], as being essential to any OR intervention, with their implementation often constituting the entire intervention.

Although PSMs differ in both their underlying theory and application, all have as their primary aim that of promoting greater understanding of the problem at hand in order to develop an efficacious solution. Today, PSMs serve to foster greater comprehension of the problem among both the ORer and the target audience. Many PSMs (SSM, SODA) are utilized to facilitate the negotiation of an agreed solution (among stakeholders) through discussion and the development of a common understanding of the problem [20]. An integral feature of this facilitation is the use of a model to portray various versions of the problematical situation in order to promote discussion among the stakeholders [86].

Today, many ORers combine parts of different PSMs in their problem structuring, much to the dismay of authors of specific PSMs who feel aggrieved that the 'purity' of their methodology is compromised [20]. An ORer should always seek to improve whatever situation in which they intervene by using any tools/methodologies available – to use 'all the arrows in one's quiver', so to speak. However, an ORer should seek to apply only those PSMs, or parts of PSMs, that they are well schooled in. The use of PSMs is an art in that, while there are many different ways to 'paint' the problem–situation, it takes a lot a practice and skill to produce a picture that everyone can appreciate and understand.

As mentioned above, knowing when a sufficient understanding of the problem has been developed in order to take action is also a practised competency. The danger with PSMs is that one can get so engrossed

---

[2]The ORD masters course offered at UCT reflects this assertion. The majority of the Problem Structuring Methods course focused on these three approaches, although other approaches were also covered.

in structuring the problem that a solution, let alone an *appropriate* solution, is never reached. This aptly called 'paralysis by analysis' should be avoided.

A large part of the problem structuring process in this study came via the numerous meetings held with members of FBSA (the 'client'). The value of meeting a client group face–to–face and listening to their views of the problem should never be underestimated. Establishing a good rapport is essential if any further progress is to be made, and holding a few meetings with the client in which they express their views, and leave feeling that their take on the problem has been appreciated is essential to establishing this rapport. Maintaining regular contact with the client is also paramount to the success of the intervention [79]. The aphorism 'No one cares how much you know until they know how much you care' is particularly relevant to practitioner–client relationships in a developing context.

5.2 Problem structuring methods used in this study

No specific PSMs were specified to be utilized in this study. One cannot tailor a problem to a particular PSM. The approach was one of first establishing a good relationship with FBSA before even mentioning 'problem–structuring'. However, a great deal of useful information was gained from the initial meetings with FBSA that helped guide the selection of the PSMs used in this study. Mingers and Rosenhead [49] (pg. 842) suggest that, in order for PSMs to be effective, they should:

- *Enable alternative perspectives to be considered with each other*

- *Be transparent to a range of participants*

- *Operate iteratively*

- *Allow contingent solutions*

These characteristics served as guidelines (a sort of 'check-list') throughout the problem–structuring process. Rosenhead posits that a successful application of a PSM in an OR intervention generates *"collaborative action [among stakeholders] towards a desired future"* [66] (pg. 764). However, the generation of a greater understanding of a problematical situation, even if only for the ORer, through the use of PSMs is also invaluable. It is important to note here that the PSMs were used in this study primarily to generate greater understanding of the problematical situation for both the practitioner and the client. Thus, whilst they were not expressly used to generate action to improve the problematical situation, they still proved to be very beneficial to both parties.

The goal in this section is to use whatever PSM, or parts of PSMs, that are suitable with respect to both their appurtenance to our problematical situation and their transparency. Here, 'transparency' refers to how easily the particular methods/tools employed are understood by the client. A lack of understanding among the client group regarding a particular PSM is a great hindrance to facilitating discussion and, hence, generating a greater understanding of the problem. No particular PSM was applied in its entirety, primarily because it was felt that doing so would be too time consuming, as it was evident from very early on in the definition of the problem that a combination of soft and hard OR methods would be needed in this study. Mingers [48] posits that it is both possible and beneficial to abstract pieces of different methodologies at the level of techniques/tools, and use these tools to augment the problem–structuring approach being employed. This is precisely what was done in this study, where two specific tools within recognized PSMs were employed as part of a broader problem–structuring process:

- Cognitive/causal mapping from SODA

- Root definitions from SSM

What follows are explanations of these two tools and then a discourse of their application to help structure FBSA as an organization (via CM) and the allocation system at FBCT (via RDs). It is hoped that the structuring of FBSA as an organization and of the allocation system at FBCT will jointly provide us with sufficient understanding of the problematical situation in order to determine the way forward to discover an appropriate solution.

5.2.1 Causal mapping

CM is a visual tool used to represent the thoughts of an individual or group of individuals about a particular issue, placing emphasis on causal relations between concepts. Concepts are represented as nodes and causal relationships through links between nodes [54]. It is a powerful tool for problem–structuring because, as Narayanan [54] (pg. 5) states, causal maps *"invoke the notion of causation, and users of the tool observe that causal analysis is built into our natural language"*. It is thus a relatively transparent and 'user–friendly' tool for both practitioner and client, and has been widely used in many problem–structuring interventions [50] (see [49, 19, 48]).

CM and Cognitive Mapping (as used in SODA) are not synonymous, although they are acutely similar. The two primary differences exist in the technique for eliciting the maps (more defined in the case of cognitive mapping) and the links between nodes (which do not necessarily represent causal relationships between those concepts in cognitive mapping). However, for the purposes of this study it is not a gross error to refer to the two techniques interchangeably, as many authors do. Most ORers view a causal map as being a particular form of a (more general) cognitive map [70], whilst some see cognitive mapping as being based on CM[3] [83]. The point here is not to enter into a comparison of the two tools, but rather to note that there is a distinction and, further, to state that CM was the tool used in this study.

CM has been utilized since the mid–70's when Axelrod [9] introduced it to management studies in exploring the belief systems of managers and decision makers (DMs) [44]. Over the past 40 years, CM as a technique has grown both in the scope of its application and in the diversity of CM approaches. Scavarda *et al.*, in their review of CM practice [69] (pg. 1), list the following ways of applying CM (for both practitioners and researchers):

1. *Diagnostic tool – help to identify and solve possible causes of a problem*

2. *Communication tool – can communicate causal relationships effectively and efficiently*

3. *Risk Mitigation tool – help anticipate unintended consequences and mitigate risks*

4. *Control tool – help identify best location for metrics and controls*

These are but a few recognized uses of CM out of an abundance of documented applications. It is important to note here that CM is not primarily used to solve problems directly. Rather, it aims to provide DMs with a visual representation of the issue at hand, stimulating reflection and discussion of one another's perspectives. In this way, CM aims to *facilitate* the process of decision–making [27]. This is a crucial role since individuals' (within and without an organization) beliefs and thoughts may influence the decisions made by an organization – although causal maps rarely capture all the thoughts of everyone present, rather the beliefs of those stakeholders deemed to be most important [27].

---

[3]This is the view adopted for the purposes of this study.

Vidal [83] (pg. 1023) concludes that CM is effective in helping the user/s to: *"evaluate and explore values, goals, strategies and actions; link strategic thinking to action plans; communicate ideas in a large group to identify conflicts; to get a holistic and systematic view of the situation; and to support the elaboration and agreement on an action plan"*. When causal maps are used in a decision–making context they often denote a 'means–ends' structure, whereby decision options are means of achieving the DMs goals (ends) [50]. Within this context, there are two approaches that can be followed to establish the relationships between concepts. The 'inductive/bottom-up' approach where lower-level decision options are first explored, moving 'up' towards forming and exploring the end goals, and the 'deductive/hierarchical' approach where higher-level goals are first explored, moving 'down' towards the formation and exploration of decision options [70].

There are copious examples of the use of CM in a variety of fields. Causal maps have been used in: investigating causal learning in children [39]; gaining greater understanding of the reasons for the failure of complex projects [2]; helping to quantitatively describe biological processes that occur at the cellular level [85]; and to compare poverty activists and non–activists regarding their beliefs about the causes of poverty in developing nations [42]. CM has even been used to assist sheep farmers in New Zealand to understand and visually delineate their farm systems [25].

Today several different types of causal maps, each with their own approaches, exist: Ishikawa (fishbone) diagrams; impact wheels; issue trees; strategy maps; risk assessment management tools and cause–and–effect diagrams [69]. Generally, there are two basic approaches to constructing causal maps - brainstorming, whereby groups of individuals meet to devote serious thought to an issue, and interviews, usually with one or two individuals who express their beliefs about an issue [69]. The thoughts and beliefs of participants are duly recorded, analysed and then captured in short phrases (10 to 12 words seems to be the recommended length [27]), or 'concepts' which form the nodes of the causal map. Time is then spent analysing the relationships between concepts. If a relationship does exist, a link is drawn between the two concepts. In CM, this link is also given a direction (indicated by an arrow head) and a sign to indicate whether one concept reinforces (+) or hinders (-) the other that it leads to[4]. In Figure 5.1 below, having a 'Well-managed flow of food from donators to Foodbanks to agencies reinforces having 'Well supported agencies in areas around foodbanks', whilst having 'Insufficient funds to enable procurement of food for agencies would hinder 'Well supported agencies in areas around foodbanks' (hence the '-' sign adjacent to the link).

Causal maps can consist of any number of concepts. Most often it is recommended to keep the number to between 40 to 50, but there is no upper limit (sometimes maps may consist of hundreds of concepts). It all depends on the level of complexity of the issue and to what depth of detail the user wishes to explore. It is common experience among many causal mappers that there exists a tradeoff between the degree of complexity elicited by a causal map and the user–friendliness of the map [27], the balance of which is dependent on the target audience. Bearing in mind the importance of transparency (section 5.2), a somewhat 'less is more' mantra is often adopted with respect to the complexity of the causal maps generated. Scavarda *et al.* [69] (pg. 1) argue that causal maps should exhibit the following characteristics:

1. *Parsimonious – show no synonymous nodes and only important links*

2. *Precise – the definitions of nodes should be precise*

3. *Complete – no important nodes or links are missing*

---

[4]If the relationship is deemed to be reinforcing, it is usually just indicated by an arrow with no sign.
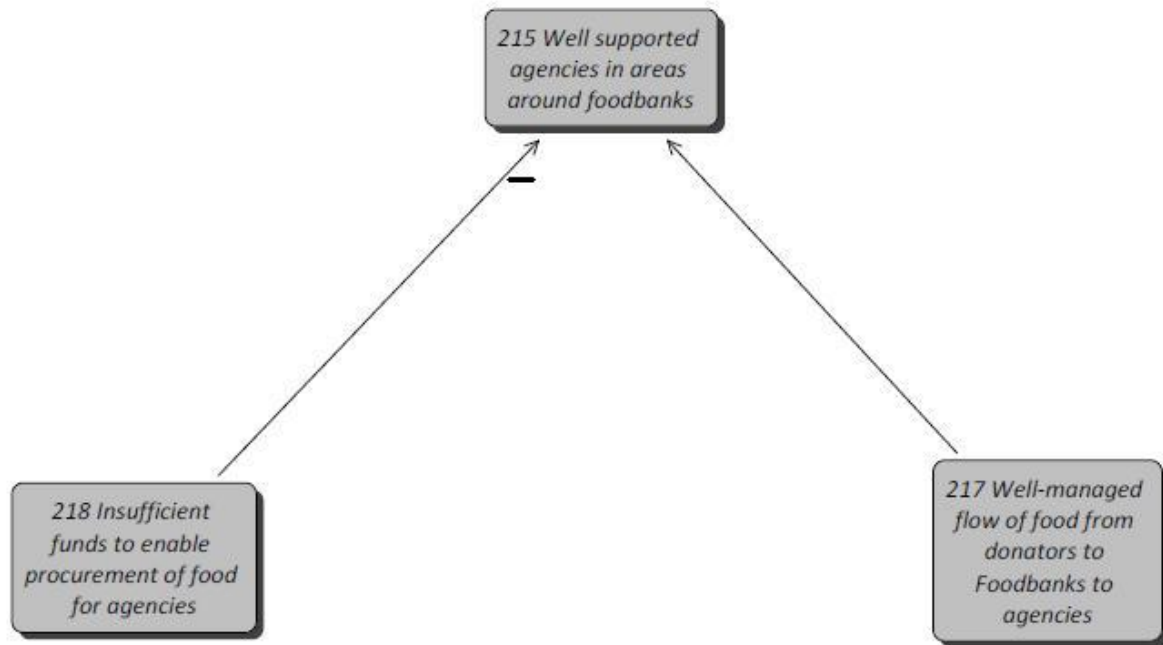
Figure 5.1: Example of part of a causal map

4. *Accurate – links between nodes and the signs assigned to each link are correct*

5. *Visual – similar nodes are close together and arcs do not cross unnecessarily*

The above characteristics served to guide the creation of the causal maps used in this study.

5.2.2 Root definitions

A 'RD' is a tool employed within SSM as a preliminary step in building a model to *describe* the problematical situation as a system that the practitioner intends to improve. Although SSM is not employed in its entirety in this study, it is important to supply a summary of its entire process in order to understand the context of how a RD is utilized. Rosenhead and Mingers [48] (Chapter 1) provide a concise summary of the SSM methodology: *"SSM is an action–oriented process of inquiry into problematical situations in the everyday world; users learn their way from finding out about the situation to defining/taking action to improve it. The learning emerges via an organized process in which the real situation is explored, using as intellectual devices – which serve to provide structure to discussion – models of purposeful activity built to encapsulate pure, stated worldviews."* Thus, SSM is a learning cycle that consists of the following elements [48]:

- Finding out – gaining a greater understanding of the problematical situation by conducting three analyses:

    1. Analysing the intervention itself
    2. Analysing the social context of the situation
    3. Analysing the politics of the situation

- Making purposeful activity models

- Using these purposeful activity models to stimulate discussion/debate about the problematical situation and how it can be improved

- Defining/taking action to improve the situation

Within the 'finding out' phase of the SSM process, analysing the intervention itself (1) entails thinking about how the practitioner employs SSM in order to address the perceived content of the problematical situation [48]. Usually, there are three roles that exist in any problematical situation:

- The 'client' – the person or group of people that caused the intervention to happen

- The 'practitioner' – the person/s conducting the intervention

- The 'issue owners/stakeholders' – those people who care about or are affected by the current situation and any action taken to improve it

Thus, the analysis of the intervention itself consists of identifying and naming the practitioner, client and stakeholders and then investigating the worldviews of the stakeholders, with the purpose of moving towards constructing purposeful activity models of the situation[5]. 'Purposeful activity' here refers to the notion that in any real–world situation, there exist people who are trying to act purposefully, not just by instinct or randomly [48]. RDs play an integral role in the formation of these purposeful activity models, and consequently a key role in the entire SSM process.

As stated above, every problematical situation involves people, each acting purposefully with their own worldview [48]. A RD is essentially a written statement that describes a problematical situation as a purposeful activity system from a particular person's viewpoint or worldview. What is crucial to understand here is that there may be copious RDs for any problematical situation[6], since each RD is contingent upon the worldview of one person (or group of people) who have an interest in the system. Usually, all people who have an interest in the system and its outcomes are referred to as 'issue owners' or 'stakeholders'. Stakeholders can include anyone within the system, affected by the system, or anyone who 'owns' the system in the sense that they can exercise control over the system.

RDs are invariably created by considering the elements of the mnemonic 'CATWOE', illustrated in Figure 5.2 (adapted from [14]), and the 'PQR' formula [48]: Do 'P' by 'Q' in order to help achieve 'R'. The PQR formula can also be interpreted as answering the questions: Do 'What'? (P), 'How' (Q) and 'Why' (R) about the purposeful activity system under investigation.

C	*Customer/s*	Who would be the victims/beneficiaries of the activity?
A	*Actor/s*	Who would do the activities?
T	*Transformation Process*	What is the purposeful activity?
W	*Worldview*	What view of the world makes this definition meaningful?
O	*Owner/s*	Who could stop this activity?
E	*Environmental Constraints*	What constraints in the environment are taken as given?

Figure 5.2: The CATWOE mnemonic

---

[5]Note that for the purposes of this study, it was felt that focusing attention on the intervention itself would yield great insight into the problem, and that conducting analyses of the social and political contexts would be too time consuming and superfluous to what was needed.

[6]Hence the metaphor 'root' that alludes to the fact that each RD is but one core way of describing the system [48].

The core of a RD is the transformation process (T) that converts/transforms some entity into a different state [14]. When constructing RDs, users generally start by determining T and W and then progress on to the other elements of CATWOE [48].

Following the formulation of RDs within the guideline of CATWOE, the SSM process exhorts the user to envision a model of the purposeful activity system and define criteria by which the performance of the notional system could be judged. There are three criteria (three E's) that are pertinent for any system (adapted from [48]):

- Efficacy – whether the transformation is working, i.e. achieving its intended outcome

- Efficiency – whether the transformation is being achieved using a minimum of resources

- Effectiveness – whether the transformation is helping to achieve some longer–term or higher–level goal

Considering such criteria promotes greater understanding of the purposeful activity system being investigated [48], which in turn provides a better understanding of the problem at hand.

5.3 Structuring of FBSA

5.3.1 Causal mapping of FBSA

Approach followed in this study

The CM technique employed in this study most closely resembles cognitive mapping, as used in SODA by Eden and Ackermann [1]. Although the cognitive mapping approach was by no means rigorously adhered to, it did serve as a guideline in creating the causal maps in this study[7]. The approach followed was unique in that the maps were initially created solely by the author of this study from information gained from FBSA's website, meetings with FBSA, and documentation obtained from them. It is noted that this is a significant departure from the traditional brainstorming or interview approaches, but it still proved very useful in structuring the problem, a well as strengthening a mutual understanding (between the author and the client) of the organization itself.

A deductive approach was adopted in creating the maps, where the links between concepts were causal in nature. The 'opposite poles' concept from cognitive mapping [1] was used, albeit infrequently. The maps were created[8] iteratively in the sense that the author would create and present a map to the supervisors of this study who would provide constructive feedback, which would then be incorporated and a subsequent presentation of the improved map made. This process would continue until there was general agreement between the author and the supervisors that the map was ready to be presented to representatives from FBSA. The next step was to hold a meeting with FBSA, in which the maps would serve as a communication tool (section 5.2.1) in an interview–type setting, to stimulate discussion of the issues represented on the map. This was found to be very beneficial to both the author and the representatives from FBSA, one of whom expressed a keen interest in using CM in some of his own work within the organization.

---

[7]The tutorial 'Getting Started with Cognitive Mapping', provided with Banxia's Decision Explorer Software, proved to be very useful in this regard.

[8]The causal maps in this study were created using the Banxia Decision Explorer (COPE) software, ©1991-2007 Banxia Software.

Another significant departure from traditional CM approaches is that the chief outcome of the entire CM exercise was to ensure that the author had a good understanding of FBSA as an organization in terms of its goals. This was certainly seen to fall under structuring of the problem, as the organization itself serves as the context from which the problem emanated. Thus, understanding the organization's vision, strategic objectives and goals would go a long way towards a good structure of the problem. The 'means–ends' context was used in creating the maps, where different colours for concepts were used to indicate where they belonged on the hierarchy (overall goal, strategic objective, possible action). Concepts were also worded in terms of 'states of being' i.e. they were represented as a desired state of being rather than in the more established 'action–orientation' [1]. For example, instead of a concept reading 'Establish more foodbanks' it would read 'Having a sufficient number of foodbanks'.

Causal maps of FBSA

Figure 5.3 represents a small causal map of the 'overall' or 'blanket' goals of FBSA. Note that this map was the final product of the iterative approach described above (section 5.3.1), thus reflecting the beliefs/worldviews of representatives from FBSA. 'Blanket' is used here as a metaphor to indicate that these goals 'cover' or encompass all of FBSA's other goals. The blanket goals of an organization are usually long–term goals that can be perceived as describing the vision of that organization.
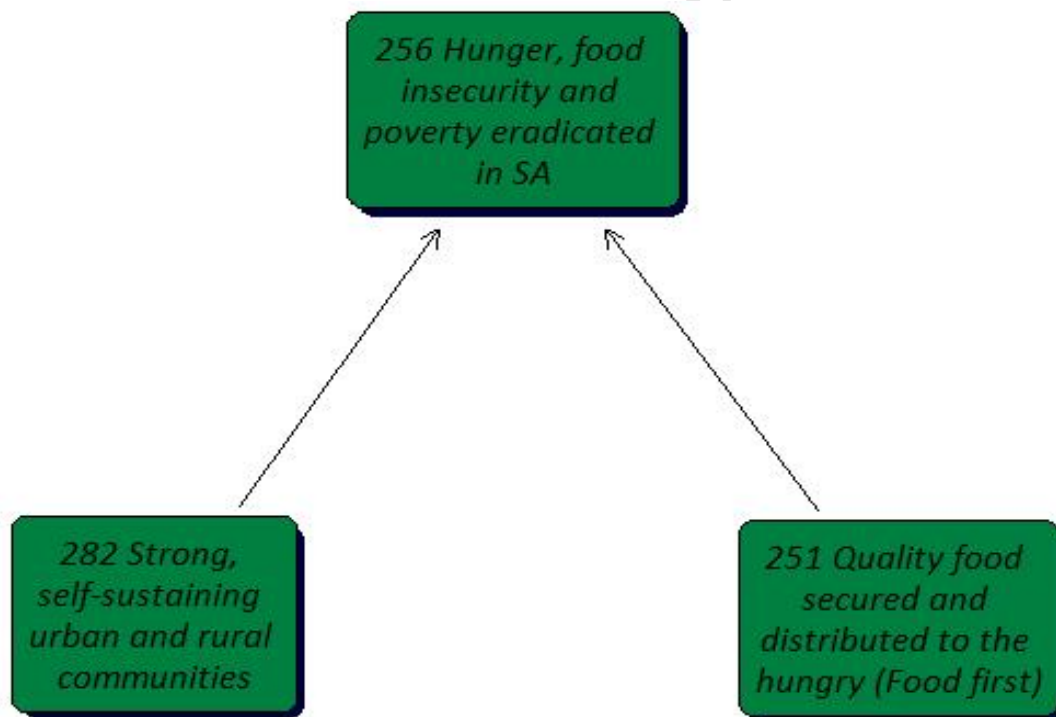
Figure 5.3: Causal map of overall goals of FBSA

Thus the vision of FBSA is to see hunger, food insecurity and poverty eradicated in SA. This concept (256)[9] was derived from FBSA's website. Following the deductive approach, the next step was to ask the question 'What would be necessary/helpful to FBSA to achieve this state (i.e. the elimination of hunger,

---

[9]Note that the numbers of each concept are inconsequential.

food insecurity and poverty from SA)?' In answer to this question, the concepts 'Strong, self–sustaining urban and rural communities' and 'Quality food secured and distributed to the hungry (Food first)' were developed i.e. FBSA believes that both securing and distributing quality food to the hungry (which they are currently doing to a degree) and having strong, self–sustaining urban and rural communities will play an integral part in eradicating hunger, food insecurity and poverty in SA. Both these concepts are long–term goals and thus also form part of FBSA's vision. These two concepts also reflect two pillars of FBSA's strategy – that of the *distribution* of quality food and community *development*.

The concept of having *self–sustaining* urban and rural communities reflects FBSA's belief that development is only true development if it is sustainable, and that their role in the fight against hunger involves more than just distributing food. FBSA is not looking to assuage the problem of hunger – they want to end it. This is obviously a bold, possibly implausible vision, but it is important with a view to motivating people within (staff) and outside (government, businesses, public etc.) of FBSA to support them in this effort. 'Alleviating hunger' is much less emotive than 'Eliminating hunger'.

Figure 5.4 depicts both the overall goals and strategic objectives of FBSA. This is an expanded causal map of Figure 5.3, assembled after further exploration of FBSA's overall goals[10]. The strategic objectives of an organization are those that are still viewed as being long–term goals, but represent more specific, strategic targets to work towards.

The map portrays the third pillar of FBSA's approach – *advocacy* (concepts 235, 234, 190). FBSA wants to become the foremost 'voice' on issues of hunger and poverty in SA, campaigning on behalf of the thousands of social welfare organizations across SA who are committed to fighting hunger (190). FBSA also recognizes the importance of involving the South African public – garnering their support through raising awareness of the plight of the millions of hungry people in SA (234). Every little contribution to the cause makes a difference, hence the more people involved the greater the impact. In fact, FBSA needs public cooperation in order to eradicate hunger, food insecurity and poverty in SA. Equally as important, if not more so, is the ability to influence government and other key stakeholders in the food industry with regards to the legislation pertaining to food policy and funding (235). This is a crucial element in the fight against hunger. Legislation plays an important role in the sense that it can either be a great help or hindrance to the efforts of FBSA. Legislation can also mobilize large–scale efforts across a country for a particular cause. Thus FBSA would like to be in the position where they are recognized as a credible 'thought leader' in matters pertaining to food policy (190), which will help them to influence government and other key stakeholders on issues relating to food policy decisions (235). If FBSA can play a role in the formulation/change of legislation to aid their cause, it would be a huge step towards eradicating hunger (189) in SA, a large element of which would be reaching a state of being able to distribute quality food to all hungry South Africans (185).

In order to reach the state of being able to distribute quality food to all hungry South Africans (185), FBSA recognize that they need to have an effective (and efficient) network of community–based foodbanks in rural and urban areas across SA (192)[11]. Recall that one of the problems highlighted by FBSA was the determination of where the next few foodbanks should be established (see section 3.2). Having such a network is also essential for FBSA to extend their influence into the rural areas of SA (195) and to ensure that FBSA continues to provide adequate support to agencies across SA (191). Increased support from

---

[10]Note that a more comprehensive map of FBSA in terms of its overall, strategic and short–term goals was developed, but is too large to be included in this document. It is available from the author on request.

[11]Note that 'effective (and efficient) network' here refers to both the operation of individual foodbanks in the network and the network as a whole.
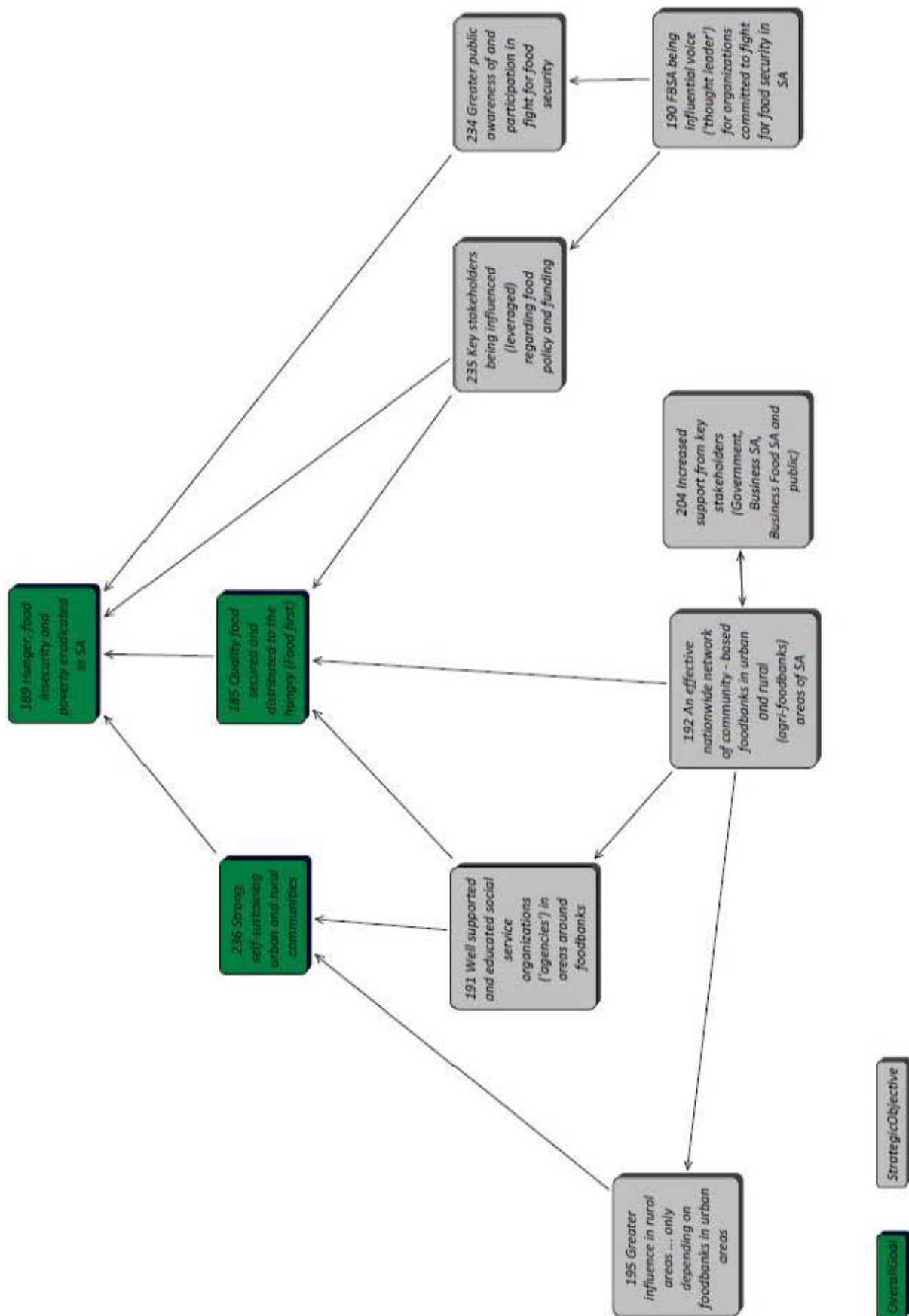
Figure 5.4: Causal map of overall goals and strategic objectives of FBSA

key stakeholders in the food industry (204) is essential for FBSA to continue to establish foodbanks across the country and thus build an effective nationwide network. However, demonstrating the effectiveness of the current network is important to help win more support from these stakeholders (hence the two arrow heads on the link between these concepts).

Having an effective nationwide network (192), well supported agencies (191) and being able to influence key stakeholders on food policy (235) would all help FBSA to effectively distribute food to hungry people across SA. Thus these concepts all play a role in the *distribution* pillar of FBSA's approach.

The *development* pillar in FBSA's approach is its most recent addition. FBSA realize that to effectively eliminate hunger, food insecurity and poverty in SA (189), they need to stimulate sustainable development of communities in urban and rural areas. FBSA's vision for the development of communities is to be able to continue to support agencies in urban and rural areas (191), whilst simultaneously increasing their influence in rural areas (195) through their Agri–foodbank development initiative (192) (see section 2.2.3). Note that concept 195 is written in the form of a contrasting pole – at the time the map was created, FBSA was discussing their strategy regarding the rural areas: whether they continue to 'reach out' to rural areas via the closest urban foodbank, or to employ a more direct approach (Agri–foodbank model). Hence this choice is represented in the concept with the ellipsis, which can be read 'versus'.

Important information learnt from structuring of FBSA

A wealth of valuable information about FBSA as an organization and hence the context in which decisions are made at FBSA was unearthed via the CM process:

- A good understanding of the vision (overall goals) of FBSA was gained, and that this vision 'casts a shadow' over all of FBSA's other strategic goals

- By unpacking this vision, we learnt what FBSA's strategic objectives are

- Understanding FBSA's overall goals and strategic objectives gave us insight into their strategy to achieve their vision and its structure, viz. the advocacy, development and distribution 'pillars' of their approach. Specifically, it was learnt that:

  – These three pillars are equally important and interdependent

  – FBSA is not trying to alleviate hunger, but eliminate it – this is reflected in the three–pillared approach, especially in their emphasis on sustainable development of communities

  – FBSA is seeking to expand both their network (in terms of having a sufficient number of foodbanks) and their influence in rural areas

  – FBSA's approach is holistic – it incorporates a variety of strategies along different avenues that focus on logistical (distribution), social (development, advocacy) and political (advocacy) aspects of the problem of eliminating hunger

- FBSA has specific strategic targets for the next few years (e.g. implementation of Agri–foodbank model)

- FBSA's strategy is dynamic in that it is still being developed (i.e. decisions about the strategy are still being made – think of concept 195) and it needs to respond to new challenges in innovative ways (e.g. Agri–foodbank model)

- FBCT's allocation system would fall under the distribution pillar of FBSA's strategy

- Effective and efficient allocation of food to agencies is key in helping to achieve the strategic objective of having well–supported agencies in areas around foodbanks, which plays an integral part in achieving the overall goal of distributing food to the hungry

In summary, we have gained a good understanding of FBSA as an organization and its approach to ending hunger, food insecurity and poverty in SA. We have also gained a better perspective of where FBCT's allocation system fits into the 'bigger picture' of FBSA and its vision and goals – FBSA wants to see effective and efficient allocation of food to the agencies it supports. This is important for us to understand, as this goal becomes a criterion by which the operation of any particular foodbank (especially its allocation system) is measured, and is thus something that will always be considered when decisions about FBCT's allocation system are made. Hence, a better understanding of the context in which 'allocation decisions' are made was gained.

5.4   Structuring of FBCT's allocation system

5.4.1   Application of root definitions

Approach followed in this study

It was decided to use RDs to gain a better understanding of the allocation system at FBCT. First, the relevant stakeholders in the allocation system are identified. Then their roles within the system are defined, and a RD for each stakeholder is formulated. A comparison between the RDs of each stakeholder will be conducted, in the hope that this will shed more light on the problem. Subsequent to this, criteria will be developed (three E's) against which the performance of the allocation system should be measured.

First, in carrying out an analysis of the intervention itself (section 5.2.2), we identify:

- The practitioner – Neil Watson, assisted by Prof Theodor Stewart and Dr Leanne Scott

- The client – FBCT

- The issue owners/stakeholders – various people involved in FBCT's allocation system, as detailed below.

Root definitions of stakeholders in allocation system

The allocation system at FBCT consists of the following stakeholders[12], and their roles within the system:

- Food Donors/suppliers: the organizations/individuals who donate food to FBCT

- FBCT Ground-staff: the people working at FBCT on a daily basis

- FBCT Warehouse manager: oversees the operation of the FBCT warehouse

---

[12]Note that this list is not exhaustive – it is intended to represent a cross–section of the stakeholders in FBCT's allocation system that fulfil different roles, in order for us to appreciate the influence of their different worldviews in gaining a better understanding of the problem. Also, to formulate a RD for every possible stakeholder would be too time consuming and unnecessary for the purposes of this study.

- FBSA National Agency Coordinator (NAC): has to meet the agencies supported by FBCT each quarter and explain the current allocation policy to them

- Social Welfare Organizations (agencies): the organizations in and around CT that are supported by FBCT

After the initial meetings held with representatives from FBSA, subsequent meetings with representatives from FBCT were held and the practitioner spent three full working days at the FBCT warehouse where the allocation system was observed. Thus, a large amount of information about the allocation system and its stakeholders was accumulated, which enabled the stakeholders to be identified and an initial structure for the problem to be built.

The root definitions

Food Donors/Suppliers
The food donors supply food to FBCT[13]. It is no secret that the majority of this food is actually the 'left–overs' of these organizations. It is difficult to gauge just how concerned the donors are with regards to the allocation of this food to agencies, but we make the assumption that they are at least somewhat concerned, as it would be much more efficient for them to throw away the food they donate. They do not derive any financial benefit from their donation, but their public image most probably benefits from their association with FBCT. Thus, their outlook on the problem probably does not extend further than ensuring that their 'waste' is efficiently removed and that they gain some form of recognition for their donation. The CATWOE elements for the Food Donors/suppliers could be as in Figure 5.5.

C	Agencies
A	Management and ground–staff at FBCT
T	waste food → food delivered to agencies
W	donating food to FBCT is a viable alternative to throwing it away, and is beneficial to our public image
O	the management at FBCT
E	the existing warehouse and transportation capacity at FBCT

Figure 5.5: CATWOE elements for the food donors/suppliers

Thus, a RD of the allocation system for the Food Donors/suppliers could be formulated as:

A FBCT–owned and staffed system to efficiently recover waste food and deliver it to agencies, with the available warehouse and transportation capacity, in order to supply those agencies with edible food that can be given to hungry people.

Ground–staff at FBCT
The ground–staff refers to those people who work at the FBCT warehouse everyday, and the drivers of the trucks that collect and deliver food. In spending time observing and conversing with various ground–staff at the FBCT warehouse, it became evident that the majority of them enjoy their jobs and

---

[13]At present, FBCT generally collects this food from the donor in the evening or early morning.

want to perform them well in order to improve the operation of the warehouse and the allocation system. They demonstrated genuine interest in being part of the hunger–relief network that FBCT represents in the greater CT region. They are concerned that the warehouse and allocation system runs efficiently as this will reflect positively on their individual and group performance, which will result in a less stressful working environment. They are not as concerned about the fairness and accuracy of the allocation of food to agencies. The CATWOE elements for the ground–staff could be as in Figure 5.6.

C	Agencies
A	Management and ground–staff at FBCT
T	current allocation system → more efficient and fair allocation system
W	An efficient warehouse operation and allocation system is desirable as it reflects well on our individual and group performance and will result in a less stressful working environment
O	the management at FBCT
E	existing working environment, warehouse and transport infrastructure at FBCT

Figure 5.6: CATWOE elements for the ground–staff

Thus, a RD of the allocation system for the ground–staff could be formulated as:

A FBCT–owned and staffed system to efficiently allocate and deliver food to agencies, with the available warehouse and transport capacity, in order to supply agencies with edible food and provide a less stressful working environment.

FBCT general manager
This refers to the general manager of the FBCT warehouse. His primary concern is that the warehouse as a whole, and consequently the allocation system, operates effectively and efficiently. He is constantly monitoring the performance of the warehouse and seeking ways to improve its operation. He is also concerned about the fairness and accuracy of the allocation. He is located at the FBCT warehouse and intimately involved in its operation. He cares about the well–being of his staff and holds regular meetings with them to discuss his and their concerns. He would like to improve the efficiency and effectiveness of the warehouse using only the available resources. The CATWOE elements for the FBCT general manager could be as in Figure 5.7.

C	Agencies
A	Management and ground–staff at FBCT
T	current allocation system → more efficient and fair allocation system
W	improving the effectiveness and efficiency of the FBCT warehouse and allocation system using only available resources is desirable
O	the management at FBCT
E	quantity of donated food, available staff, equipment, and warehouse and transport infrastructure at FBCT

Figure 5.7: CATWOE elements for FBCT general manager

Thus, a RD of the allocation system for the FBCT general manager could be formulated as:

A FBCT–owned and staffed system to efficiently and fairly allocate food to agencies , using only available resources, in order to supply agencies with edible food and improve the overall efficiency of the FBCT warehouse.

FBSA NAC
The NAC is also in charge of Information and Communication Technology for FBSA. Once every quarter, he has to appear before representatives from the agencies supported by FBCT and explain the allocation system to them. He is spearheading a project to garner more accurate information about the agencies being supported by FBCT. His primary concern is improving the fairness and accuracy[14] of the allocation system. He is also concerned that the allocation system is efficient. The CATWOE elements for the FBSA NAC could be as in Figure 5.8.

C	Agencies
A	Management and ground–staff at FBCT
T	current allocation system $\rightarrow$ more efficient and fair allocation system
W	an allocation system that results in accurate and fair allocations to agencies is desirable
O	the management at FBCT
E	quantity of donated food and warehouse and transport infrastructure at FBCT

Figure 5.8: CATWOE elements for the FBSA NAC

Thus, a RD of the allocation system for the FBSA NAC could be formulated as:

A FBCT–owned and staffed system to efficiently produce fair and accurate allocations of food to agencies, in order that agencies can be supplied with edible food in a way that they feel is transparent and fair.

Agencies
The agencies are those social welfare organizations in the greater CT region that are supported by FBCT. They are largely unaware of how the FBCT warehouse operates. Their primary concern is that they are allocated enough of the right types of food to satisfy the needs of the hungry people they are supporting. They are not too concerned with the efficiency of the allocation system. However, they are concerned that the allocation system is fair and transparent. The CATWOE elements for the Agencies could be as in Figure 5.9.

Thus, a RD of the allocation system for the Agencies could be formulated as:

A FBCT–owned and staffed system that performs fair and accurate allocations of food to agencies in order to provide them with a sufficient supply of the right types of food to satisfy the needs of the hungry people they are feeding.

---

[14]Accuracy in the sense of supplying the right categories of food to a particular type of agency.

C	Agencies
A	Management and ground–staff at FBCT
T	current allocation system → more efficient and fair allocation system
W	receiving a sufficient supply of the right types of food to satisfy the needs of the people we are feeding is important
O	the management at FBCT
E	the quantity of donated food and the existing allocation system as it operates

Figure 5.9: CATWOE elements for the agencies

5.4.2 Reflection on root definitions

If the RD's for the stakeholders are compared, it is easy to see that they are similar. All recognize that those who stand to benefit from the system are the agencies (customers), and the system could be changed by the management at FBCT (owners), whilst the management and ground staff at the FBCT warehouse are the people who actually 'do' the allocation system (actors). The differences exist in people's worldviews (which is to be expected), the transformation process and the environmental constraints. It is worthwhile to briefly discuss these differences here.

The stakeholders who represent FBCT (ground–staff, general manager and the FBSA NAC) are all concerned, to different degrees, about the effectiveness and efficiency of the allocation system as a whole. The food donors/suppliers are only interested in the efficiency with which their waste food is taken from them, which is reflected in their worldview and transformation process. The FBSA NAC and the agencies are primarily concerned with the fairness and accuracy of the allocation system. Thus we have potentially conflicting interests. The issues of fairness and accuracy add significant complexity to the problem of improving the allocation system in that incorporating them may result in more time–consuming allocations and thus a reduced efficiency of the system – which is what the FBCT general manager (and management in general) would not want. However, seeking to improve just the efficiency of the system may result in biased and inaccurate allocations – which is what the FBSA NAC and the agencies would not want. This tension between the two issues is something that any solution will have to address.

The transformation process for all the stakeholders, barring the food donors/suppliers, is essentially the improvement of the current allocation system into one that is more efficient and fair. All of the stakeholders with this transformation have interests in both the efficiency and fairness of the allocation system, albeit with different primary concerns.

The environmental constraints for all stakeholders, except the agencies (who simply have to accept the allocation system as it stands), essentially involve the FBCT warehouse's resources – whether it be the number of staff, equipment or warehouse or transport capacity. These are constraints that are assumed to be relatively inflexible for the time being as FBCT is a NPO that does not have a limitless budget to hire more staff or purchase more equipment or trucks, and there are no immediate plans to relocate to another larger warehouse. Another crucial constraint is the amount of donated food that FBCT receives on a daily basis – which is invariably far too little to satisfy the needs of the agencies presently being supported. This is also a relatively inflexible constraint in that, although the quantity of donated food is gradually increasing over time (as FBSA sources more donors/suppliers), the list of social welfare organizations requesting support is also growing. At present there are more than 2000 organizations across SA that are on a waiting list to receive support from FBSA. These constraints will have to be incorporated into any proposed solution.

Criteria for efficiency, efficacy and effectiveness

It is now important that we outline the criteria, as explained in section 5.2.2, by which the performance of the allocation system will be judged, as these will serve as good measures of the appropriateness of the solution that we develop:

1. Efficacy – criteria for efficacy indicate whether the transformation is working [48]

   - The primary intended outcomes of the transformation are improving the *efficiency*, *fairness* and *accuracy* of the allocation system. With regards to improving:
   - Efficiency – a reduction in the time taken to perform allocations would be desirable
   - Fairness – an equal percentage of satisfied need across all agencies would be desirable
   - Accuracy – a higher percentage of satisfied need for particular food types across all agencies would be desirable

2. Efficiency – criteria for efficiency indicate whether the transformation is being achieved with the minimum use of resources [48]

   - Any proposed changes to the allocation system should not exceed the current resource use
   - In addition, any proposed changes that reduce resource use whilst either maintaining the current allocation system or improving it would be desirable

3. Effectiveness – criteria for effectiveness indicate whether the transformation is helping to achieve a higher–level goal [48]

   - The higher–level goal for this transformation would be to improve the support provided to agencies by FBCT
   - If we are able to meet the criteria for the efficacy and efficiency of the transformation, this would most certainly contribute to a significant improvement in the support offered by FBCT, and would help reach the state of having well–supported agencies in CT (concept 191 in Figure 5.4)

5.5   Reflection on problem structuring

In summary, we are seeking to improve the efficiency, fairness and accuracy of the allocation system within the current resource (infrastructural and donation–quantity) constraints i.e. we are seeking a solution that will not exceed the current resources available and has to factor in the relatively insufficient supply of donated food. The solution will also be judged according to whether it fits into the strategic and overall goals of FBSA (effectiveness), which are important to consider.

In reflecting on Mingers and Rosenhead's suggestion for the application of PSMs to be effective [49] (see section 5.2), it is encouraging to note that both PSMs were utilized iteratively (in the sense that they were employed, reviewed and re–employed numerous times until their output was deemed to be satisfactory). They most certainly were understood by all the participants involved (i.e. had a high level of transparency), and through promoting discussion (CM), they enabled peoples' different perspectives and worldviews (RD's) to be considered. As noted before, the PSMs were not utilized with the intent of arriving at solutions, but rather to pave the way forward to finding such a solution.

The use of CM has helped us gain a good understanding of FBSA's overall and strategic goals, providing the context in which decisions are made. In particular, it helped to facilitate discussion with members of FBSA, generating shared learning of FBSA's goals between the practitioner and client. Applying the CATWOE mnemonic in formulating RDs for the stakeholders in the allocation system helped to define what the problem is and how we can measure the appropriateness of any proposed solution. Thus, these tools have greatly increased our understanding of FBSA, FBCT's allocation system and the problem of improving this system. However, these tools are not suitable to help develop a solution to the problem. In order to improve the allocation system at FBCT, we need to model it and simulate it. Building a simulation model of the allocation system will help us understand the allocation decisions made on a daily basis, and also allow us to investigate the effect of any proposed changes (i.e. new or different allocation decisions) to the system. Thus, we will now develop a simulation model of FBCT's allocation system.

Chapter 6

Model Building

6.1   Data collection and sorting

Initially, the only available data were that of the allocations made to agencies from FBCT. This was obtained in Excel spreadsheet format from FBSA for the period March 2009 – June 2010. Since FBCT has a policy of 'whatever fresh produce comes in on a day, must go out on that day', it was felt that it would be reasonable to assume that the total weight of allocations of perishable food to agencies on any particular day would be a good approximation of the total weight of incoming perishable donations for that day. However, after spending time at the FBCT warehouse, it was soon realized that often some of the frozen goods that arrive (where frozen goods are classified as being perishable) are kept in cold storage for one or two days. This also happens with some of the dairy products and other fresh produce. Thus, we could no longer justify our previous assumption[1].

Data for incoming goods were eventually obtained for the period March 2009 – June 2010. However, this data were incomplete (there were no data available for October–December 2009, and a few months had a number of days missing). A new 'online' data capturing process, whereby data would be entered onto a system as incoming goods were weighed (as opposed to data being hand–recorded and then entered by a different person at a later stage), was commissioned at FBCT at the beginning of March 2010. It was felt that this new process would be less prone to human error (such as incorrect entries/calculations, data going missing) and result in more reliable data. Thus, data of incoming goods (perishable and non–perishable) were obtained for the period March–June 2010 (119 days). This data were 'cleaned' and sorted in Microsoft Excel using *Visual Basic for Applications* (VBA), whereby a macro was written to remove unnecessary 'noise' and sort the data into the relevant food categories. The data were split into 'perishable' foods, those foods that would become inedible within 2 to 3 days, and 'non–perishable' foods, those that would last at least a month based on what type of food it was[2] (see Appendix A for this classification.).

---

[1]Upon comparing data for incoming goods with this data at a later stage, the two showed very little correlation.
[2]Note that the majority of food received by FBCT is already near or past its expiration date

6.2 Distribution fitting

Since the daily inflows of perishable and non–perishable food would be used as inputs for the simulation model, it was necessary to model them with a distribution. A free trial version of Mathwave's *EasyFit* software was downloaded from *www.mathwave.com* and used to fit over 50 continuous distributions to the data. *EasyFit* estimates the parameters of distributions using well known estimation methods (maximum likelihood estimation, method of moments, method of l-moments and least-squares estimation). Statsoft's Statistica 9 was also employed to fit distributions to the data. From the outset, it was decided that we would be seeking to fit only one or two–parameter distributions as it was felt that any distribution with three or more parameters would bring unnecessary added complexity to the simulation model. Distributions were fitted to:

1. Total daily inflow of perishable food
2. Total daily inflow of non–perishable food

6.2.1 Total daily inflow of perishable food

The top 3 best-fitting two–parameter distributions, according to a χ^2 goodness–of–fit test, were:

1. Frechet, FRE(1.77, 1.45)
2. Log–logistic, LLOG(2.52, 1.99)
3. Log–normal, LN(0.70, 0.69)

Goodness–of–fit

EasyFit automatically performs three 'goodness–of–fit' tests: the Anderson–Darling, Kolmogorov-Smirnov and Chi-square (χ^2) on each of the fitted distributions. The distributions are then ranked according to their performance on each test. It was decided to rank the distributions according to the χ^2 goodness–of–fit test. The top 3 distributions with their respective test statistics for the χ^2 goodness–of–fit test are summarized in Table 6.1.

Table 6.1: Best–fitted two–parameter distributions ranked according to their scores on the χ^2 test for the daily inflow of perishable goods, ($\alpha = 0.05$)

Distribution	Statistic	*Dof*	Critical Value	Reject?	P-value	Rank
Frechet	3.01	6	12.59	No	0.81	1
Log–logistic	4.44	6	12.59	No	0.62	15
Log–normal	4.66	6	12.59	No	0.59	16

Selection and justification of a distribution

From Table 6.1, it is clear that all three of the distributions are good fits to the data. Upon examining histograms of the data with each of the distributions superimposed, it was discovered that both the Frechet and Log–logistic distributions significantly overestimate the occurrence of smaller donations (≤ 2 tons). Although the Log–normal also overestimates the smaller donations, it provides a better fit in this region than do the other two. Thus, it was decided that the Log–normal would be the most parsimonious distribution to model the daily perishable donations. A histogram of the weights of the daily perishable donations with the best–fitted Log–Normal distribution was computed in Statistica and is displayed in Figure 6.1. The best–fitted Log–normal distribution will be used to generate n random weights of daily perishable inflows for the simulation model.
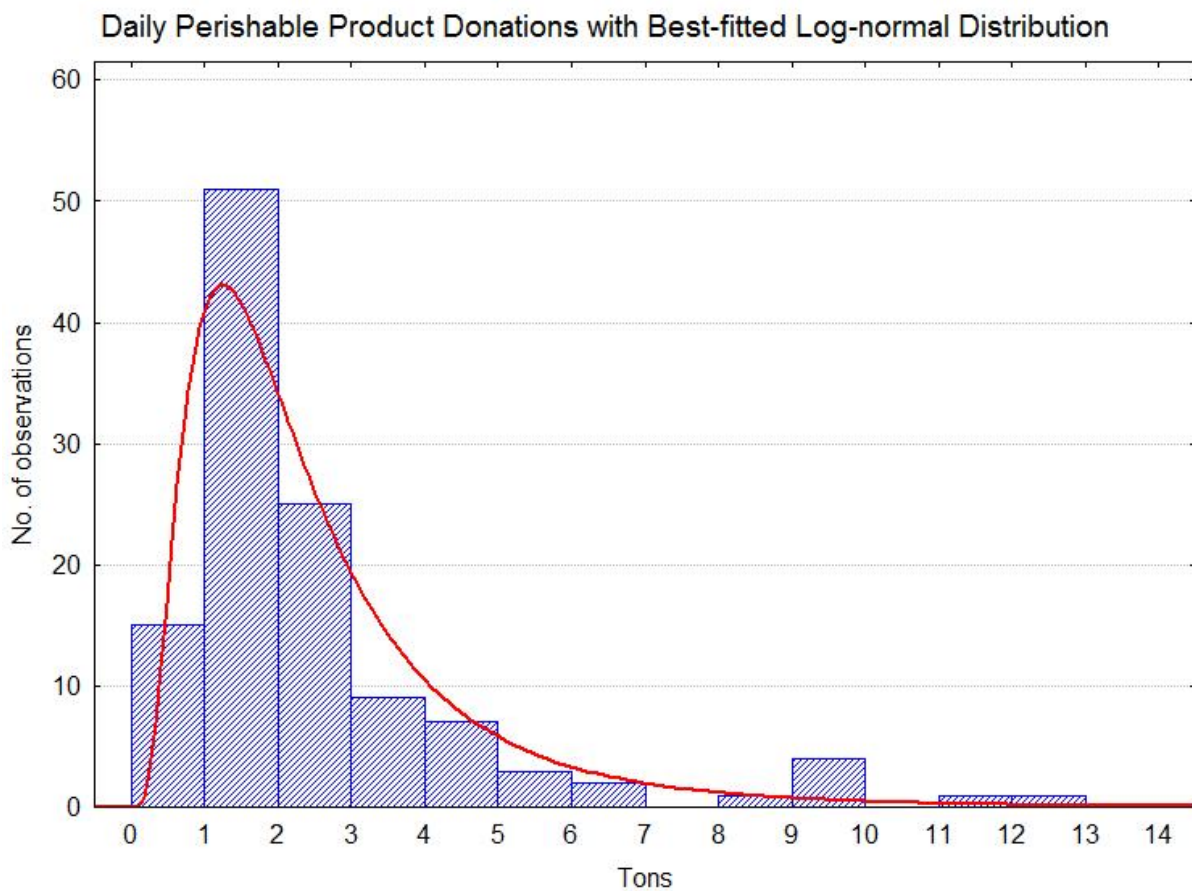
Figure 6.1: Histogram of daily perishable donations with best–fitted Log–normal distribution

Perishable food proportions

Daily perishable food donations can be categorized into six types (see Table A.1 in Appendix A): bakery; cooked food; dairy; fruit; meat and vegetables. For the purposes of the simulation study, it was decided to generate separate distributions for each of these categories of perishable foods, as it was felt that this would provide a more accurate representation of the donations process at FBCT. Whilst the available data enabled distributions to be fit to the total daily inflow of perishable donations, they were not detailed enough to fit the separate distributions for each category. These distributions were fitted in the following

manner. Firstly, estimates of the proportion of daily perishable donations of each of these categories were obtained by computing the total average proportions over the period March to June 2010. These are summarized in Table 6.2.

Table 6.2: Category proportions of daily perishable donations

Category	Proportion (%)
Bakery	35.27
Cooked food	2.02
Dairy	7.34
Fruit	17.27
Meat	8.34
Vegetables	29.77

Secondly, these category proportions were used to scale the mean of the best–fitted Lognormal distribution (i.e. the mean of the underlying normal distribution) in order to 'fit' Lognormal distributions to each of the perishable food categories according to their proportion of the total daily perishable donations. The assumption is made that donations of each perishable food product type are independent, and that the physical processes that result in these donations are the same. The coefficient of variation (CV) for the Lognormal distribution is independent of its mean[3], and thus the Lognormal distributions of the perishable food categories are assumed to have the same CV and consequently the same sigma parameter, but different means according to their representative proportions. These distributions will be used to generate n random daily donations of each perishable food category. The perishable food categories and their respective Lognormal distributions are summarized in Table 6.3.

Table 6.3: Perishable food categories and their fitted Lognormal distributions

Category	Distribution
Bakery	Lognormal(-0.34, 0.69)
Cooked food	Lognormal(-3.20, 0.69)
Dairy	Lognormal(-1.91, 0.69)
Fruit	Lognormal(-1.05, 0.69)
Meat	Lognormal(-1.78, 0.69)
Vegetables	Lognormal(-0.51, 0.69)

6.2.2   Total daily inflow of non–perishable food

Upon examining the data for daily non–perishable food donations, it became clear that there was a large portion (approximately 40%) of days on which no perishable goods were received. It was thus decided to model the total daily weight of non–perishable donations as follows:

---

[3]This can easily be proven by computing the CV with the moment generating functions of the Lognormal distribution.

Let $X_{np} \equiv$ the total daily weight of non–perishable donations. There are two sets of values that we consider for X_{np}, viz. $X_{np} = 0$ (if no donations are received) or $X_{np} > 0$ (if a donation is received). From the above observation, it is evident that we assume $P(X_{np} = 0) = 0.4$ and thus $P(X_{np} > 0) = 0.6$. Thus, the probability of either receiving no donation or a donation on a particular day can be expressed as:

$$F_x(X_{np}) = P(X_{np} \leq x) = \begin{cases} 0.4, & \text{if} \quad x = 0 \\ 0.6 \times C(x), & \text{if} \quad x > 0 \end{cases}$$

where $C(x)$ is the best–fitted (continuous) conditional distribution of X_{np} (see below).

Practically, this can be incorporated into the simulation model by determining the distributions of all the non–zero daily non–perishable donations, i.e. $X_{np}|X_{np} > 0$, and then multiplying the $(n \times 1)$ vector of n simulated daily non–perishable donations by a $(n \times 1)$ vector of n simulated Bernoulli random variables, Z, where the probability of a success (i.e. $Z = 1$) is 0.6. The Bernoulli random variable with success parameter of 0.6 will take on a value of 0 approximately 40% of the time, thus reflecting the assumption that 40% of the time no non–perishable donations are received.

We now determine the conditional distribution of X_{np} given that a donation is received, i.e. $X_{np}|X_{np} > 0$. We thus consider all the data values > 0, and fit two–parameter distributions with *EasyFit* (as in section 6.2.1).

The top three best-fitting two–parameter distributions, according to a χ^2 goodness–of–fit test, were:

1. Gamma, GAM(0.49, 2.91)

2. Weibull, WEI(0.50, 1.07)

3. Log–normal, LN(-0.99, 3.24)

Goodness–of–fit

It was decided to rank the distributions according to the χ^2 goodness–of–fit test. The top three distributions with their respective test statistics for the χ^2 goodness–of–fit test are summarized in Table 6.4.

Table 6.4: Best–fitted two–parameter distributions ranked according to their scores on the χ^2 test for the daily inflow of non–perishable goods, ($\alpha = 0.05$)

Distribution	Statistic	*Dof*	Critical Value	Reject?	P-value	Rank
Gamma	1.86	6	12.59	No	0.93	1
Weibull	3.56	5	12.59	No	0.62	6
Log–normal	6.13	5	12.59	No	0.29	17

Selection and justification of a distribution

From Table 6.4, it is clear that all three of the distributions are good fits to the data. Upon examining histograms of the data with each of the distributions superimposed, it was discovered that all three:

- Significantly overestimate the occurrence of very small donations ($X_{np} \leq 0.25$ tons)

- Underestimate the occurrence of small donations ($0.25 < X_{np} \leq 1.5$)

- Provide good estimates of the occurrence of large donations ($X_{np} > 1.5$ tons)

Of the three distributions, the Gamma provided the best fit for the smaller donations. Thus, it was decided to use the Gamma distribution to model the total daily weight of non–perishable donations given that a donation occurs, i.e.

$$P(X_{np} \leq x | X_{np} > 0) \sim GAM(0.49, 2.91)$$

A histogram of the weights of the daily non–perishable donations with the best–fitted Gamma distribution was computed in Statistica and is displayed in Figure 6.2.
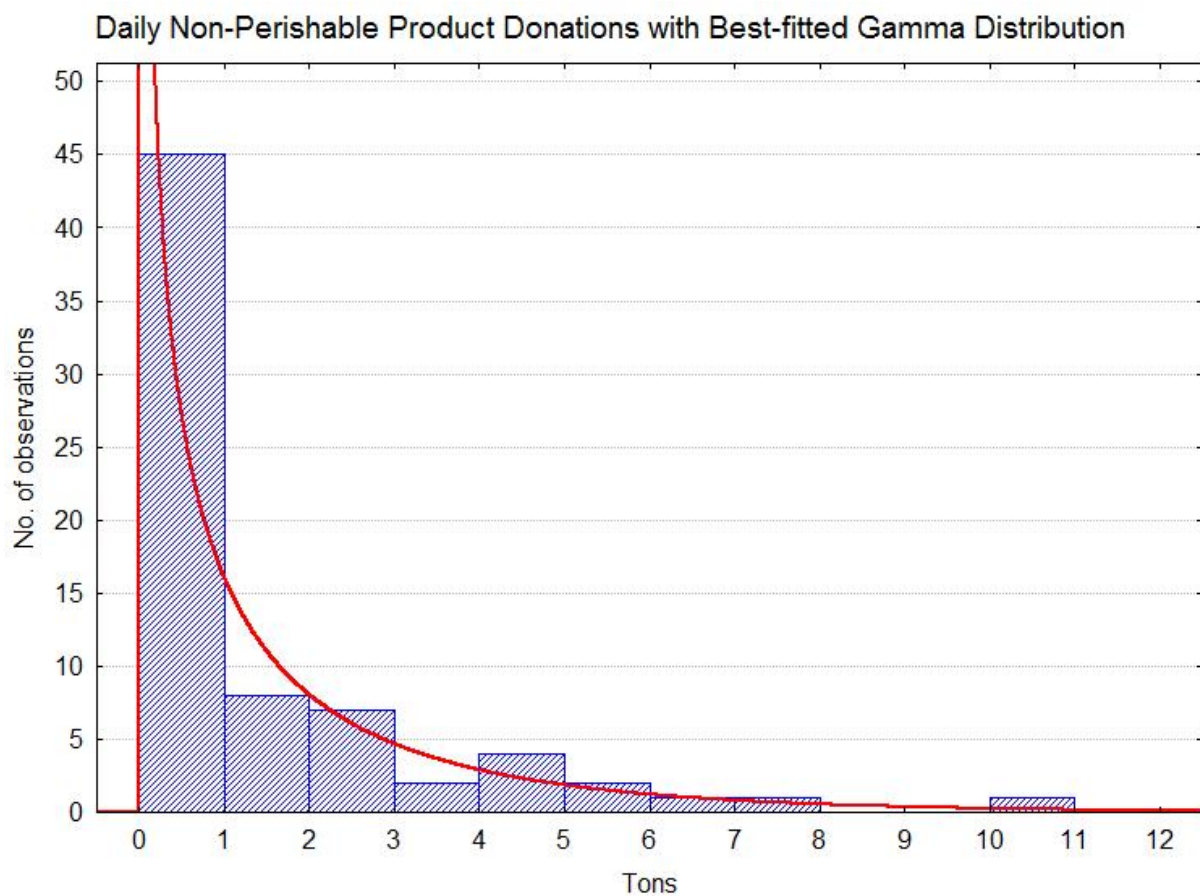
Figure 6.2: Histogram of daily non–perishable donations with best–fitted Gamma distribution

Non–perishable food proportions

Daily non–perishable food donations can be categorized into six types (see Table A.1 in Appendix A): boxed; canned; condiments; luxury foods; luxury goods; other. In order to provide a more accurate representation of the donations process at FBCT for the purposes of the simulation study, separate distributions for each of these six categories were generated (in the same fashion as the perishable food categories in section 6.2.1) as follows. Estimates of the proportion of daily non–perishable donations of each of these categories were obtained by computing the total average proportions over the period March

Table 6.5: Category proportions of daily non–perishable donations

Category	Proportion (%)
Boxed	17.38
Canned	22.54
Condiments	21.52
Luxury foods	24.42
Luxury goods	8.69
Other	5.45

to June 2010. These are summarized in Table 6.5.

The category proportions were used to scale the scale parameter of the best–fitted Gamma distribution in order to 'fit' Gamma distributions to each of the non–perishable food categories according to their proportion of the total daily non–perishable donations. We make the assumption that donations of each non–perishable food product type are independent, and that the physical processes that result in the donations are the same. The CV for the Gamma distribution is independent of its scale parameter[4], and thus the Gamma distributions of the non–perishable food categories are assumed to have the same CV and consequently the same shape parameter, but different scale parameters according to their representative proportions. These distributions will be used to generate n random daily donations of each non–perishable food category. The non–perishable food categories and their respective Gamma distributions are summarized in Table 6.6.

Table 6.6: Non–perishable food categories and their fitted Gamma distributions

Category	Distribution
Boxed	Gamma(0.49, 0.51)
Canned	Gamma(0.49, 0.66)
Condiments	Gamma(0.49, 0.63)
Luxury foods	Gamma(0.49, 0.71)
Luxury goods	Gamma(0.49, 0.25)
Other	Gamma(0.49, 0.16)

6.3 Simulation model

Simulation can be defined as *"the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system"* [74] (pg. 2). This definition serves as a good description of the purpose of the simulation experiments to be conducted in this study. In reflecting on Pidd's guidelines for the type of simulation study to be conducted [62] (see section 4.3), it is obvious that the incoming

---

[4]This can be easily proven by computing the CV with the moment generating functions of the Gamma distribution.

donations will be modelled stochastically, whilst time will be moved forward in fixed increments of one day. Each day could be viewed as an event, wherein the amounts of donated foods (stock) and allocated foods could be seen as discrete random variables. Other elements of the model (such as the classification of what types of foods are allocated to which categories of agency) are predetermined and thus deterministic. Hence, a stochastic discrete–event simulation model in which time is moved forward in fixed increments (days) will be developed. The model (simulation model) that serves as the basis to these experiments will now be detailed. The simulation model (and the other allocation models derived from it) described in subsequent sections serves two primary purposes (see section 6.3.3 for more information). It will be utilized to:

- Perform daily allocations at the FBCT warehouse (see section 6.4.1) and provide members of FBSA with a tool to review allocation policies by simulating them and investigating their performances (see section 6.4.2), as two components of the DSS FAST

- Conduct a simulation study of a number of allocation policies with a view to developing a range of 'optimal' allocation policies to be considered by FBSA for future use (see sections 6.3.3, 6.3.4 and Chapter 8 – Simulation Experiments)

Before these purposes are explored further, a brief justification of the software used in developing and running the model is provided, followed by a description of the allocation system at FBCT (in order to provide some context to the simulation model).

6.3.1 Justification of programmes used

It was decided to build a model to simulate the daily allocation decisions made at the FBCT warehouse using *Microsoft Excel 2007* and *Visual Basic for Applications* (VBA). The justification for the use of these programs is that:

a) Although not the most sophisticated programming language, VBA is still very powerful (especially for interfacing with Excel) and particularly useful for the type of model we will be constructing

b) Working with Excel and VBA allows us to seamlessly integrate powerful optimization programs (Solver and Whats Best!) into our simulation model

c) The author is familiar with Excel and VBA (as a programming language) and, more importantly,

d) The target audience at FBCT are used to working with Excel, thus enhancing the transparency of the model[5]

A simulation model provides an ideal platform for investigating both subtle and major changes to the variables incorporated in the model. The basic 'structure' and mathematical formulation of the model are described in Figure 6.3 and section 6.3.4 respectively, whilst the actual VBA code is provided in Appendix B.

---

[5]Ultimately, people from FBCT need to be able to understand and use the model. Hence, using a more sophisticated program and language, while it may provide more efficient simulations, would be useless if it is not transferable to the client.

6.3.2   Description of allocation system at FBCT

Essentially, the model is designed to depict the daily operation of the allocation system at FBCT. The idea is NOT to develop the most accurate model (in principle, no model is ever entirely accurate as it is simply impossible to incorporate all the variables present in every situation), but rather a defensible/justifiable model of the observed situation. In the time spent at FBCT, much was learnt about the daily allocation procedure and the decisions incorporated in making allocations. A typical daily allocation procedure can be described as follows:

- It begins with the first delivery of perishable goods (mostly bakery goods, fruit and vegetables) at 7am in the morning

- The goods are unpacked and sorted into their categories, with food that is unfit for eating being thrown away. There are usually between four and six people involved in this process

- The goods are weighed (with these weights being recorded as the data we used to model incoming daily donations) and are then moved to another section of the warehouse. There are usually two people involved in the weighing process and a forklift is used to move the goods

- One of the two floor managers, both with considerable 'warehousing' experience, examine the number of crates of food and the list of agencies that are to be given food on that particular day. The list details the name and type of agency, as well as the number of people being supported by that agency.

- The floor manager then decides:

  1. What types of food to give each agency, based on the category of agency

  2. How many crates of the types of food should be given to each agency, based on how many people are being supported by the agency

- Once these two decisions have been made for a particular agency, the food to be given to that agency is weighed and then stored together

- In the meantime, at least one more delivery of perishable goods arrives and the allocation process begins again for those deliveries

- Whilst the perishable goods are being allocated, some of the non–perishable inventory is also allocated to the agencies by the other floor manager. At present, the allocation of non–perishable foods does not happen on a daily basis, but rather sporadically depending on amount of non–perishable stock and on the amount of perishable stock available to be allocated on a particular day

- Once all the food has been allocated, it is divided into two categories:

  1. Food to be delivered to agencies (by truck)

  2. Food that is to be collected by agencies from the warehouse later that day

- After the food has been divided into these two categories, the food to be delivered is loaded onto the trucks and taken to the relevant agencies, whilst the food to be collected is stored in a section of the warehouse. The allocation process is now complete.

The primary decisions involved in the above allocation process include:

1. What types of perishable food to give to each agency

- This decision is predetermined, according to the category of agency. See Table A.2 in Appendix A for details of this classification

2. How much of each type of perishable food to give to each agency, based on the number of people being supported by each agency

   - This decision is made by the floor manager overseeing the allocation. It is made by *visually* estimating how much food there is (i.e. how many crates) and then dividing this food between the agencies

3. How much of what type of non–perishable goods to give to each agency

   - This decision is made by a floor manager, in consultation with the warehouse manager, and is based on what is available in the non–perishable inventory

   - As mentioned above, at present FBCT does not allocate non-perishable stock everyday, but rather sporadically – at most once a week, and often only every two or three weeks. In meeting with FBCT, they indicated their desire for a more 'scientific' allocation process that would incorporate a more regular allocation of non–perishable goods. However, no data specifically relating to this allocation of non–perishable stock were available at the time of writing this study.

   - Seeing that we have assumed that FBCT only receives non–perishable goods sixty percent of the time (see section 6.2.2), and FBCT would like to work towards a more regular allocation of non–perishable goods, for the sake of model simplicity it does not seem unreasonable to mirror their sporadic allocation of non–perishables by assuming that when non–perishable goods are received, they are allocated that day (as is the case for the perishable goods). This would result in non–perishable goods being allocated sixty percent of the time i.e. approximately three out of every five days[6]

4. Which food is to be delivered (by which driver)

   - This decision is also largely predetermined according to the current delivery routes being employed. However, this decision does not form part of the scope of this dissertation

From the above description, it is easy to see that the current allocation system is heavily based on the expertise and perception of one or two FBCT staff. As it stands, it is very prone to human error and relies primarily on subjective decisions[7]. There is little underlying methodology employed by the DMs, who make decisions based on what they see and what they feel to be fair.

By their own admission, the management at FBCT emphasized that the system is not 'scientific', and they recognize the need to improve it. The purpose of the proposed simulation model is to depict the daily operation of the allocation system described above, with particular emphasis placed on the allocation decisions made, in the express hope that such improvement can be made.

What follows is an outline of the simulation model, after which the mathematical formulation of the model will be provided.

---

[6]It is noted here that a more sophisticated model that would store and allocate non–perishable goods based on the amount of perishable goods in inventory would certainly be of great value. However, such an addition was not possible due to time constraints.

[7]Subjective in the sense that they are being made by at most two people.

6.3.3 Outline of simulation model

Figure 6.3 depicts a broad, holistic outline of the simulation model that serves as the basis for the simulation study[8]. It is important to comment here that due to the limitation in both the scope for this half–dissertation and the available time to complete this research, explained in section 10.2.1, the 'holistic' outline presented here was only explored to a preliminary extent. There is certainly room for further simulation studies to be carried out within the framework that Figure 6.3 provides, as discussed in section 10.3.

The model consists of two 'layers' or sub–models – one within the other. The first, inner layer is of the allocation system at FBCT. This model will have as its inputs the available stocks of perishable and non–perishable goods. As mentioned above, the model will simulate the daily allocation decisions that are made. These decisions, or 'decision rules' (DRs), are functions of variable parameters, and any subset of them collectively forms an 'Allocation policy'. They are expressed as linear algebraic equations in the model as detailed in section 6.3.4. Thus, the DRs with their parameters also form part of the input into the daily allocation model, as reflected in Figure 6.3.
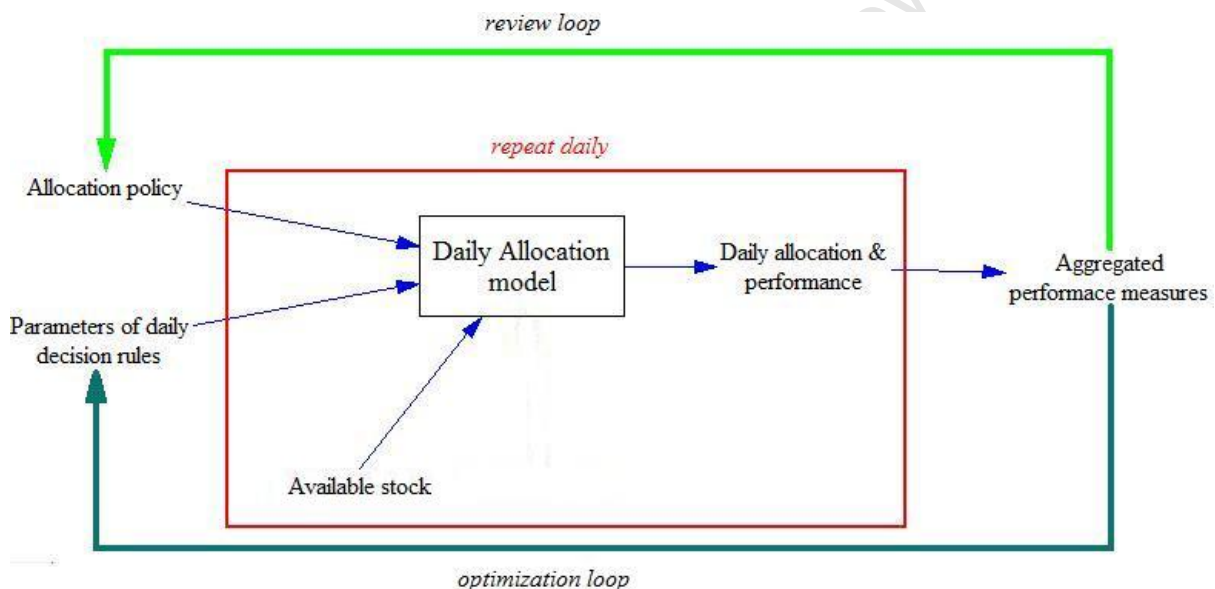
Figure 6.3: Broad Outline of Simulation Model

The following DRs will be modelled in the simulation model:

1. What types of perishable food go to which category of agency

   - This DR is currently predetermined (see Table A.2 in Appendix A)

2. How much of each type of perishable food is given to each agency

   - This DR will vary according to the intended outcome of the allocation; e.g. equal allocation across all agencies, equal percentage of need satisfied across all agencies etc.
   - It will be one of the major focus areas of our simulation model

---

[8]This diagram was constructed in Vensim Professional for Windows Version 5.9, Copyright © 1998–2008 Ventana Systems, Inc.

3. What types of non–perishable food go to which category of agency

   - This DR is also currently predetermined like DR 1
   - This DR depends on DR's 1 and 2

4. How much of each type of non–perishable food is given to each agency

   - This DR depends on the intended outcome of the allocation of non–perishable goods, e.g. to simply 'top–up' the amount of food received by agencies and/or to provide specific non–perishable foods to specific agencies

Note that while it would be interesting to evaluate changes in the classification of which food types are allocated to each agency category, there is good reason why the classification stands as it does. FBSA has conducted research surveys of their agencies and thus has a good understanding of the needs of each type of agency category. This is precisely the reason why such a classification (see Table A.2 in Appendix A) exists. For example, it makes no sense to allocate raw meat to a school where there are no facilities to cook it – it would go to waste. Likewise, allocating large amounts of vegetables to agencies that have no cooling facilities (fridges) would also result in a waste of food. Whilst it would be worthwhile conducting further research into this classification to investigate the possibility of changes and their effects, for the purposes of this dissertation (as well as a result of limitations in the scope and time for this study – see section 10.2.1) it was decided to leave the classification as it stands.

The 'performance' of the allocation system will be measured as a function of a number of performance measures such as:

a) The average Kcal/person value in each agency

b) The average minimum Kcal/person value in each agency

c) The average % of agency's target nutritional need satisfied

d) The average variance in % of agency's target nutritional need satisfied

e) The average number of agencies receiving $< 50\%$ of target % need

These performance measures are explained in section 6.4.2 and in the simulation results (Chapter 8).

Regarding point a) above, it was decided to measure the performance of the allocation system primarily according to the nutritional value of the food received. Essentially, this is what is most important, and it is a more precise measure than just Kg's of allocated food – variances in which may result in significant differences in the nutritional value, and consequently the satisfaction of hunger, at a particular agency. For example, although agency A and B may receive the same percentage of needed Kg's of food, agency A may receive only dry goods, whilst agency B may receive a combination of fruit, vegetables, meats and dry goods that is of far greater nutritional value than agency A's allocation.

In order to incorporate the 'nutritional value' performance measure into our simulation model, an average nutritional value (Kcal/Kg) for each food category was calculated by computing the average Kcal/Kg value across the most common foods in each category. The nutritional value along with examples of the most common foods of each category are detailed in Table A.1 in Appendix A[9]. The target nutritional value for each agency depends on the average daily human nutritional requirements, the target level of

---

[9]The average nutritional value of an average cooked meal was calculated based on providing 200g of vegetables, 200g of meat and 100g of fruit.

need to be satisfied, and the number of people being supported by the agency. According to the FAO [30], the average human being needs approximately 2000–2200 Kcal per day to meet basic nutrition needs (for simplicity, we will use a figure of 2100 Kcal). Thus, if the allocation target is to provide one meal per day per person being supported by an agency[10] (i.e. 33.3% of need satisfied $\equiv$ 700 Kcal per person) that supports 50 people, then the total nutritional target for the agency would be 35000 Kcal per day.

The first layer of the simulation model is repeated daily, each repetition representing a separate simulation. The second, outer, layer consists of two parts: 'Optimization' and 'Review'.

Optimization. The aim of simulating the daily allocation system is to find the values of the parameters incorporated within the DRs being modelled that correspond to the best or 'optimal' performance of the system. This is reflected in Figure 6.3 by the 'optimization loop' where 'loop' indicates that many simulations will be run in order to determine these optimum parameter values. Once a specified number of simulations of the inner layer have been run, aggregated performance measures of the system are calculated. These aggregated performance measures are used to determine a set of 'optimal' parameter values of the DRs for each allocation policy considered.

Review. The second part of the outer layer has as its focus the longer–term allocation decisions, in particular the decision about which DRs to use to form an allocation policy. This is indicated by the 'review loop' in Figure 6.3. Such a review of allocation policy takes place, at most, once every quarter at FBCT. The aim for this part of the simulation would be to derive a range of allocation policies, with their corresponding performance measures, that could be considered by FBCT when reviewing their operating allocation policy.

6.3.4 Mathematical formulation of simulation model

Note that it is not plausible to express the full model algebraicly at this stage because two separate models are simulated in Chapter 8 and these two models ('max–min' and 'number–based') are only detailed in sections 8.1.1 and 8.1.2 respectively[11].

Let:

- food type (i):

  (1) $\equiv$ Baked goods

  (2) $\equiv$ Boxed goods

  (3) $\equiv$ Canned goods

  (4) $\equiv$ Condiments

  (5) $\equiv$ Cooked foods

  (6) $\equiv$ Dairy goods

  (7) $\equiv$ Fruit

  (8) $\equiv$ Luxury foods

  (9) $\equiv$ Luxury goods

  (10) $\equiv$ Meat

  (11) $\equiv$ Vegetables

  (12) $\equiv$ Other goods

- Agency category (j):

---

[10]This is FBSA's current allocation target.

[11]It is the author's opinion that trying to include this formulation here would make reading more confusing.

$$(1) \equiv \text{Clinic}$$
$$(2) \equiv \text{Educare}$$
$$(3) \equiv \text{Feeding Scheme}$$
$$(4) \equiv \text{Nutritional Centre}$$

$$(5) \equiv \text{Satellite}$$
$$(6) \equiv \text{School}$$
$$(7) \equiv \text{Shelter}$$
$$(8) \equiv \text{Soup Kitchen}$$
$$(9) \equiv \text{Support Group}$$

- $Y_i \equiv$ amount of food type i available (stock) to be allocated

- $X_{ip} \equiv$ amount of food type i allocated to agency p, $i = 1, 2, \ldots 12$; $p = 1, 2, \ldots s$

$$X_{ip} \geq 0 \quad \forall\, i,\, p$$

- $T_p \equiv$ the number of people supported by agency p, $p = 1, 2, \ldots s$

- $W_i \equiv$ the total number of people allocated food type i across all agencies, $i = 1, 2, \ldots 12$

- $\sum_{p=1}^{s} X_{ip} \leq Y_i$, $i = 1, 2, \ldots 12$

- $S_{ij} \equiv$ binary variable that indicates whether food type i can be allocated to agency category j, where

$$S_{ij} = \begin{cases} 1, & \text{if food type } i \text{ can be allocated to agency category } j \\ 0, & \text{otherwise} \end{cases} \qquad \begin{array}{l} i = 1, 2, \ldots 12 \\ j = 1, 2, \ldots 9 \end{array}$$

- $v_i \equiv$ the value[12] associated with the amount of allocated food i

- $V_p = \sum_{i=1}^{12} v_i X_{ip} \,/\, T_p \equiv$ total value per person in agency p

- $DR(z) \equiv z^{th}$ Decision Rule, $z = 1, 2, \ldots, m$

- $A_t = \{DR(z)\} \subseteq \{DR(z) : z = 1, 2, \ldots, m\} \equiv$ Allocation policy t, $t = 1, 2, \ldots, w$

Two allocation models, based on the simulation model presented here, are simulated in the simulation study (see Chapter 8). The algebraic formulations of each of these models are presented in sections 8.1.1 and 8.1.2 respectively.

6.4   Decision support system - FAST

FAST was created to assist FBSA with allocation decisions. FAST was 'trialled' by FBCT and feedback from that process is included in section 9.3. FAST comprises two major functions:

- A database management system and daily allocation tool

- A simulation model to investigate the long–term performances of different allocation policies

---

[12]The 'value' here will primarily refer to the nutritional value per Kg but may change depending on the simulation experiment being run.

The two functions are clearly related to the inner and outer layers of the (broader) simulation model presented in Figure 6.3. What follows is a detailed description/Userguide for each of these components, accompanied by actual 'screenshots' of the FAST system as it is being used[13].

6.4.1 FAST as a DSS for performing daily allocations

This part of FAST has been developed to help aid the decision–making of those people involved in performing daily allocations at FBSA (i.e. at any one of the FBSA warehouses countrywide), with the hope of improving both the accuracy and efficiency of the allocation system as a whole. It has been created in such a way as to make it user–friendly and simple to operate. It covers two major areas of the allocation system:

1. The database of agencies
2. Daily allocation to agencies

The purpose of this section is to provide a brief summary of the functionality of FAST (i.e. what it can do), as well as highlight certain actions that the user should avoid so as to maximize their benefit from using FAST. The functionality will be explained by considering the two major areas of the allocation system (as above).

What FAST can do – the database of agencies

Essentially, for this area FAST serves as a database management system. The *'Home'* spreadsheet (see Figure 6.4) displays the following information that is used by FAST:

- The *Kcal/Kg nutritional value* of all the 12 food types allocated by FBSA
- A list of *depots* to which food is delivered
- A 'Delivery type' *colour key*

These values may be changed by the user, e.g. a new depot may be added.

The user can click on the blue *'FAST'* bar in the Excel spreadsheet which will display the *'FAST Homepage'* userform[14] (Figure 6.5).

The *FAST homepage* presents the user with the two major areas (frames) of the allocation system, 'Agencies' and 'Allocation'. Within the *Agencies frame*, three options are provided:

- Add new agency
- Change existing agency
- Remove existing agency

---

[13]Note that a 'FAST–Userguide' document was prepared for FBSA for technical support while they were testing the FAST system.

[14]Note that a 'userform' is classified as any object (usually a box shape) that the user interacts with in order to perform specific tasks.

Figure 6.4: Excel spreadsheet displaying blue *'FAST'* bar

Figure 6.5: FAST – Home page

By selecting one of these three options and pressing the *'Next'* button, the user will be able to perform the selected operation, each of which will now be discussed.

Add a new agency

Upon selecting the *'Add new agency'* option and pressing the *'Next'* button, the *'FAST – Add Agency'* userform appears (Figure 6.6). The user is prompted to input the following details for the new agency:

- The agency's *name*

- The agency's *Pastel code* - this is a 6 digit code used to identify the agency in FBSA's online Pastel database system[15]

- The *type* of agency (e.g. Shelter, Clinic etc.)[16]

- The *delivery type*:

    - Collection: if the agency collects its food from the Foodbank's warehouse

    - Delivery: if food is delivered directly to the agency

    - Depot: if food is delivered to a depot within the agency's vicinity. The agency will then collect its food from the depot

- If the delivery type is 'Depot', then the *depot's name* is selected from the list of available depots

- The *number of beneficiaries* (people) that the agency is supporting

- The agency's *location* (i.e. street address or suburb)

- The *day/s the agency will be allocated food* - the user can select multiple days and even indicate that they are 'Not sure yet'

- *Daily allocation numbers* – If the foodbank supports the agency on more than one day, it may only provide enough food on a particular allocation day for a subset of the total number of people supported by the agency. The user may select the *'Daily Allocation Numbers'* button to bring up the *'FAST – Agency's Daily Allocation Numbers'* userform (Figure 6.7) in which they may specify the number of people food will be allocated for on each particular allocation day[17].

The user *must* provide the agency's name, type, number of beneficiaries, and select an option for its allocation day/s before they can add the agency to the database. This information is essential for FAST to perform allocations - without it, FAST will not be able to operate properly.

Once the user has input the necessary information, they can click the *'Next'* button on the userform[18], which will bring up a *'Yes/No'* userform displaying the message *"Are you sure you want to add this agency?"* – in case the user has realized that they want to change the agency's details and/or not add the agency. By clicking *'Yes'* on this userform, the agency will be added to the database in the following manner:

---

[15]Pastel is a type of accounting software utilized by FBSA.

[16]Note that, at present, FBSA categorizes all its agencies into the following nine categories: Clinic, Educare, Feeding Scheme, Nutritional Centre, Satellite, School, Shelter, Soup Kitchen and Support Group.

[17]Note that if the user does not select this button, the agency will be allocated food based on the number for 'Number of beneficiaries' for each allocation day.

[18]Note that whenever a *'Back'* or *'No'* button is present on a userform, the user may select it if they wish to return to the previous userform. Similarly, whenever a *'Cancel'* button is present, the user may select it to stop whatever they are currently doing and return to the *FAST Homepage*.

Figure 6.6: FAST – Add agency

- It will be added to the *'All Agencies'* spreadsheet - this sheet is a complete list of the agencies supported by the foodbank, providing the following information about each agency:

  - The agency's *PASTEL code*
  - The agency's *name*
  - The agency's *type*
  - The *number of people* supported by the agency
  - The agency's *location*
  - The *delivery type* to the agency
  - The *depot name*, if the delivery type to the agency is 'Depot'
  - The agency's *allocation days*
  - An *'allocation check–box'* to indicate whether the agency has been allocated food - this is included for the purposes of recording allocations
  - An *invoice* box - included for the purposes of recording allocations to each agency

Figure 6.7: FAST – Agency's daily allocation numbers

- It will be added to each day of the week on which it is to be allocated food (i.e. its allocation days), with all of the above information (except the allocation day/s) detailed for each allocation day (see Figure 6.8)

	Code	Name	Type	# ppl	Location	Delivery Type	Depot Name
1							
9	SUN003	Sunrise Educare	Educare	76		Collection	
10	VUS001	Vusisizwe	Educare	118		Collection	
11	CHR001	Christine Revell	Shelter	49		Depot	Athlone Youth Centre
12	CIT001	GH Stark and Rehoboth (City Mission)	NutritionalCentre	238		Depot	Athlone Youth Centre
13	JTS001	JT's Aftercare	Educare	30		Depot	Athlone Youth Centre
14	PIN002	Pinkster Church Philidelphia	SupportGroup	90		Depot	Athlone Youth Centre
15	SAF002	Safeline	SupportGroup	30		Depot	Athlone Youth Centre
16	SIV001	Sivuyile Educare Centre	Educare	70		Depot	Athlone Youth Centre
17	BET005	Betty's Soup Kitchen	SoupKitchen	350	Nyanga	Depot	Blue Downs
18	BUL001	Bulelani ku Yehova	Educare	70		Depot	Blue Downs
19	FAI006	Faith Welfare	NutritionalCentre	112		Depot	Blue Downs

Home / AllAgencies / **Monday** / Tuesday / Wednesday / Thursday / Friday / Saturday / S

Figure 6.8: FAST – agency's information in the Excel spreadsheet

For the example presented in Figure 6.6, the agency 'Betty's Soup Kitchen' will be added to the *All Agencies*, *'Monday'* and *'Wednesday'* spreadsheets, where the different allocation numbers for Monday (200) and Wednesday (150) (see Figure 6.7) will be recorded on their respective days and the total number of beneficiaries (350) the agency supports recorded on the *All Agencies* spreadsheet.

Change an existing agency

Upon selecting the *'Change existing agency'* option in the *Agencies* frame on the *FAST Homepage* userform (Figure 6.5) and pressing the *'Next'* button, the *'FAST – Change Agency'* userform appears (see Figure 6.9).

Figure 6.9: FAST – Change agency

The user is prompted to select whichever agency's details they desire to change from a list, after which they can select the *'Next'* button which will display the *'FAST – Change Agency's Details'* userform (see Figure 6.10).

Figure 6.10: FAST – Change agency's details

This userform displays all of the agency's current details in the frame on the left and then provides options to change each one of these details in the frame on the right. The user can change anything, from the agency's name to its allocation days and allocation numbers. Once the user has made the desired changes, they can select the *'Change agency'* button which will bring up a *'Yes/No'* userform displaying the message *"Are you sure you want to change this agency's details?"* – in case the user has decided that they do not want to change the agency's details. By clicking *'Yes'* on this userform, the agency's details will be changed in the database in the following manner:

- The agency's details will be changed in the *All Agencies* spreadsheet
- The agency's details will be removed from every other sheet, and re–written onto the spreadsheets corresponding to the allocation days selected by the user on the userform[19]

Remove an existing agency

Upon selecting the *'Remove existing agency'* option in the *Agencies* frame on the *FAST Homepage* userform (see Figure 6.5) and pressing the *'Next'* button, the *'FAST – Remove Agency'* userform appears (see Figure 6.11).

The user is prompted to select the agency they wish to remove from the list provided. At this stage, the user has two options:

- Completely remove the agency from the database

---

[19]This is to cater for the possibility that the user may change the agency's allocation days.

Figure 6.11: FAST – Remove agency

- Remove the agency from only a particular day it is allocated food (on the assumption that it receives food on more than one day, or that the user still wishes to keep the agency's details in the database but not include it in allocations)

Should the user wish to completely remove the agency, they should leave the list (under *'Please select the day that you wish to remove the agency from:'*) blank, otherwise they should select the particular day from which they wish to remove the agency from this list. If the user leaves the list blank, the agency will be removed from the *All Agencies* spreadsheet as well as any other spreadsheet corresponding to its allocation days. If a particular day is selected from the list, the agency will be removed from only that day, and still remain in the *All Agencies* and other spreadsheets corresponding to its remaining allocation days.

Once the user has decided which agency and to what extent they would like to remove it, they can select the *'Remove agency'* button which will bring up a *'Yes/No'* userform displaying the message *"Are you sure you want to remove this agency?"* – in case the user has decided that they do not want to remove the agency. By clicking *'Yes'* on this userform, the agency will be removed from the database to the extent the user has specified.

What FAST can do – Allocating to agencies

Upon selecting the *'Allocate to agencies'* option in the *'Allocate'* frame on the *'FAST Homepage'* userform (see Figure 6.5) and pressing the *'Next'* button, the *'FAST – Allocate to Agencies'* userform appears (see Figure 6.12).

The user is prompted to select the day for which they would like to perform an allocation. Should the user wish to perform an allocation to all of the agencies in the database, they can leave this list blank. The user is then required to enter (non–negative) integer values for the amounts of each type of food

Figure 6.12: FAST – allocate to agencies

they want to allocate. The user has complete freedom to enter whatever values they desire, although it is envisioned that these values will usually correspond to the stock that is available at the foodbank at the time of allocation. Should the user wish to only allocate perishable or non–perishable items, they can leave the right or left column values at '0' respectively.

Once the user has filled in the allocation values, they can select the *'Next'* button on the userform which will display the *'FAST – Allocate to Agencies: Allocation Options'* userform (Figure 6.13).

Here the user can select one of two types of allocation to perform[20] :

- A *'Max–Min FoodValue Allocation'*[21] – this type of allocation attempts to 'balance' the Kcal/person value across all agencies, i.e. maximize the minimum Kcal/person value across all the agencies

---

[20]The formulations of both of these allocation models are provided in sections 8.1.1 and 8.1.2 in Chapter 8.
[21]'Max–Min FoodValue' will be shortened to 'max–min' for ease of use for the remainder of this document.
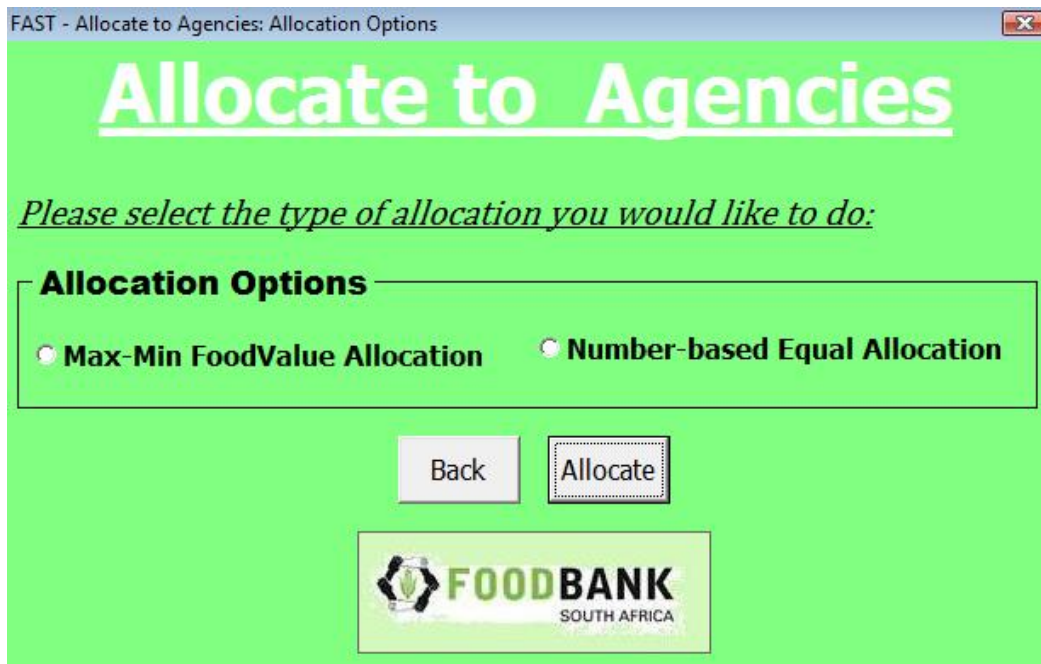
Figure 6.13: FAST – Allocate to agencies: allocation options

- A *'Number–Based Equal Allocation'*[22] – this type of allocation divides the available food between the agencies according to the number of people being supported by each agency[23]

Both kinds of allocation take into account the different types of agencies with respect to the fact that the types of food allocated to an agency depends on what category of agency it is. For example, raw meat will not be given to a School – thus any agency classified as a 'School' will not be allocated meat. Essentially, the FAST allocation system is imitating the decisions made by the floor managers at each foodbank who decide how much and what types of food are to be allocated to each agency.

For the *'Max–Min Allocation'* the number of people being supported by each agency is also included by constraining the amount of each type of food allocated to an agency to be greater than or equal to half of its 'pro–rata' amount, but less than or equal to double this amount, where the 'pro–rata' amount is calculated as:

$$\frac{amount\ of\ food\ type\ x\ \times\ number\ of\ people\ supported\ by\ agency}{total\ number\ of\ people\ across\ all\ agencies\ that\ are\ allocated\ food\ type\ x}$$

Once the user has selected the type of allocation they wish to perform, they can select the *'Allocate'* button which will bring up a *'Yes/No'* userform displaying the message *"Are you sure you want to perform the 'Max-Min FoodValue Allocation'"* or *"Are you sure you want to perform the 'Number–based Equal Allocation'"* (depending on which allocation type has been selected) – in case the user has decided that they do not want to perform that type of allocation. By clicking *'Yes'* on this userform, FAST will perform the specified allocation for the agencies on the day selected by the user.

Once the allocation has been performed, FAST outputs the resulting allocation food values for each agency on the spreadsheet corresponding to the day selected, as in Figure 6.14. Thus, the user can identify

---

[22]'Number–based Equal' will be shortened to 'number–based' for ease of use for the remainder of this document.
[23]Note that this is essentially the current allocation policy being employed by FBSA.

how much of each type of food (rounded off to the nearest Kg) to give to each agency on that day. The allocation values will also be recorded in a *'Records'* spreadsheet, where the date, time of allocation, day for which allocation is performed and type of allocation will be displayed.

Code	Name	Baked food	Cooked food	Dairy	Fruit	Meat	Vegetables	Boxed food	Canned food	Condiments	Luxury food	Luxury goods	Other
BEA001	Beautiful Gate	58	128	151	95	68	59	99	124	99	110	157	58
MAN001	Manenberg Lions	19	0	0	0	23	59	0	0	0	0	0	19
PHA001	Phakama	17	37	44	28	20	17	29	36	29	32	46	17
SIY013	Siyakhana Support Group	5	0	13	8	0	5	8	10	8	9	0	5
STR001	Street Committee	19	0	0	0	23	20	0	0	0	0	0	19
TEH001	Tehillah Nutri Centre	24	0	0	0	28	24	0	0	0	0	0	24
BAT001	Bathandwa	3	7	8	5	4	3	5	6	5	6	8	3
LIT016	Little Flower	14	0	37	23	0	14	24	30	24	27	0	14
NYA001	Nyanga Nutritional Centre	77	0	0	0	91	78	0	0	0	0	0	77
FAM001	The Family	24	0	63	40	0	24	41	52	41	46	0	24
ZAN001	Zanokhanyo	2	5	6	4	3	2	4	5	4	5	7	2
COU001	Counselling & Social Dev Prog	8	17	0	13	9	8	13	0	13	15	21	8
OLI001	Olivia Abrahams	10	21	0	16	11	10	16	21	16	0	0	10
LOG001	The Logos Assemblies of God	7	0	0	12	0	0	0	0	0	14	0	7

Figure 6.14: FAST – Allocate to agencies: allocation values in Excel spreadsheet

It is envisioned that, while these allocation values may not be strictly adhered to, they will serve as an accurate guideline as to how the available stock at the foodbank should be allocated to the agencies being considered. The two types of allocations will result in different allocation food values, and these can be compared by accessing the *'Records'* spreadsheet and examining the results of each type of allocation.

Optimal operation of FAST – what NOT to do

The purpose of this section is to highlight certain actions which the user should not undertake, in order to preserve the optimal operation of the FAST system. It is hoped that, in avoiding these actions, the user may gain maximum value from utilizing the system, whilst avoiding potential operational problems.

Actions the user should avoid

The following is a summary of actions that the user should NOT perform:

- Manually entering *any* information on *any* of the spreadsheets, except the *'Home'* spreadsheet. This includes:
  - Manually adding an agency into any spreadsheet
  - Manually changing any of an agency's details in any spreadsheet
  - Manually removing an agency from any spreadsheet
- Changing any of the information on the *'Home'* spreadsheet other than adding, changing or removing a depot from the list of depots, or changing the Kcal/Kg nutritional values

What is very important for the user to understand is that the FAST system will not operate correctly if there is incorrect and/or missing information. This is why the user is *strongly* discouraged from entering information into the system via any route other than FAST. For example, if the user examines the agency *'Type'* column on the *All Agencies* spreadsheet, they will notice that the *Feeding Scheme*, *Nutritional Centre*, *Soup Kitchen* and *Support Group* agency types are entered as 'FeedingScheme', 'NutritionalCentre', 'SoupKitchen' and 'SupportGroup' respectively, i.e. without any spaces between the words. Thus, if the user were to manually change the type of an agency (even if they do so on each sheet where the agency is

present) to 'Support Group' (i.e. with a space) then when they attempt to perform an allocation, FAST will return an error and will not be able to perform the requested allocation[24]

6.4.2 FAST as a DSS for reviewing allocation policies

This part of FAST has been developed to assist FBSA in medium to long–term decisions regarding which allocation policy/ies they should employ. FAST as a DSS for performing daily allocations is intended for use on a *daily* basis, i.e. to assist the user in making daily allocation decisions. The decision support provided by FAST as a DSS for reviewing allocation policies is intended for those decisions pertaining to the adoption/removal/changing of allocation policies at FBSA, i.e. decisions that will occur on a far less frequent basis – perhaps only a few times per year[25]. Thus this function of FAST is one that the user would spend a considerable amount of time utilizing as they simulate and examine results, then re–simulate and re–examine results, perhaps make some adjustments to an allocation model, and then re–simulate and so on and so forth. The thinking behind this decision support has been outlined in section 6.3.3. The purpose of this function of FAST is to provide the user with a simple tool to simulate a particular allocation policy for whatever length of time they desire,[26] and present them with some descriptive performance measures of that policy. For the purposes of this study, only the two allocation policies already presented are included for analysis, viz.:

- 'Max–Min Allocation'

- 'Number–based Allocation'

What follows is a discussion/Userguide for this function of FAST, accompanied by screenshots as it is being employed.

Figure 6.15 depicts the *'Simulation Homepage'*. It displays the following:

- *Number of days to simulate (n)* – the period of time the allocation policies are to be simulated for

- *Number of simulations to run* – the number of times the allocation policies will be simulated for the period specified (above)

- *Nutritional Value (Kcal/Kg)* – the nutritional value of all the 12 food types allocated by FBSA

- *Simulation Results* – a table displaying the following information about the most recent simulation run:

    - The number of agencies to which food was allocated

    - The total number of people supported by these agencies

    - The names of the allocation policies simulated

    - The average Kcal/person value across all agencies

        * This is a good indication of the overall performance of an allocation policy – the higher this value, the better the performance of the policy

---

[24]Additional information and troubleshooting tips on these and other errors are provided in the FAST userguide prepared for FBSA. It is available on request from the author.

[25]Note that FBSA would like to establish a quarterly review of their allocation policies.

[26]A Monte Carlo simulation procedure is employed, where the user can simulate a certain number of days (a maximum of 365) as many times as they wish to (i.e. for as many iterations as they desire).

* The user should consider the 'average variance in agency's percentage of target need satisfied' in conjunction with this statistic

– The average minimum Kcal/person value across all agencies

* This is a good indication of the performance of an allocation policy – the higher this value, the better the performance of the policy
* At present, FBSA's *target* Kcal/person value for each agency is 700 Kcal/person per day

– The average agency's percentage of target need satisfied

* This performance measure is directly related to the Kcal/person value – the closer the Kcal/person value is to the target Kcal/person value in each simulation, the higher the average percentage of target need satisfied will be
* The higher this percentage, the better the performance of the allocation policy

– The average variance in agency's percentage of target need satisfied

* A high value corresponds to large variability in the agency's percentage of target need satisfied i.e. some days the policy performs very well, whilst other days it performs very poorly
* The lower this value, the better the performance of the policy

– The average number of agencies receiving less than 50 percent of their target need

* If this number is small, it indicates a good performance of the allocation policy
* The closer this number is to the number of agencies being simulated (42), the poorer the performance of the allocation policy in question
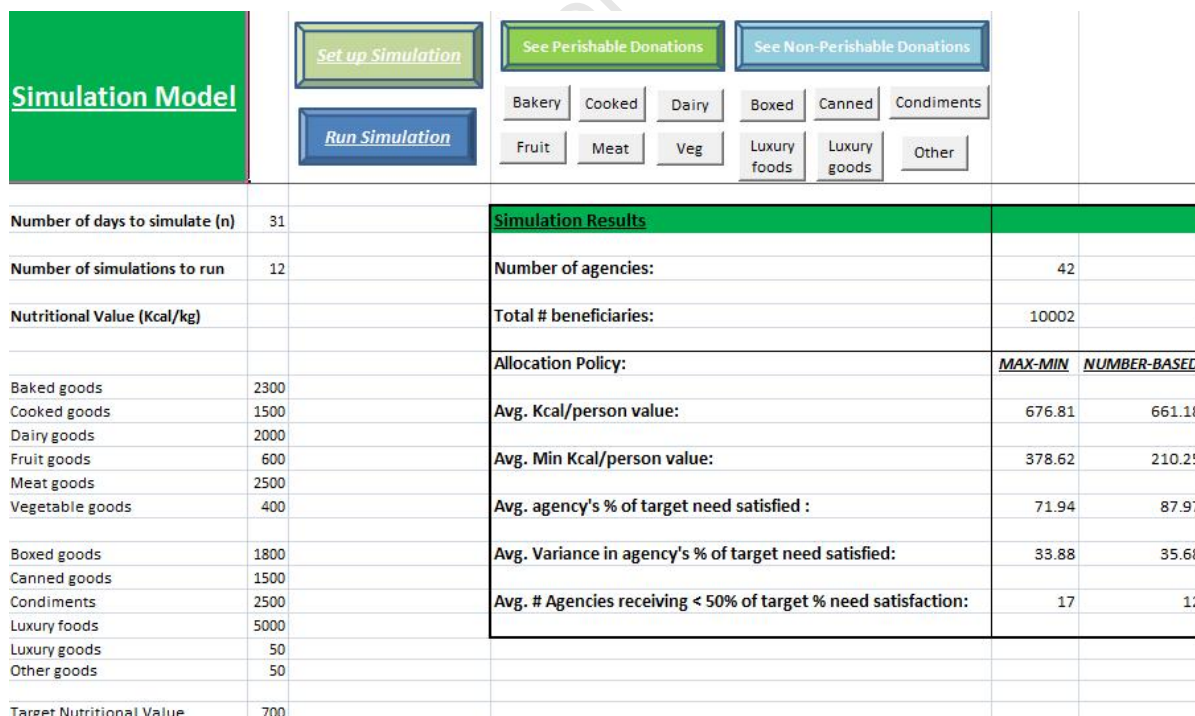
Figure 6.15: FAST – Simulation homepage

For example, in Figure 6.15, we can see the Simulation Results table detailing the above performance measures and descriptive statistics for the two allocation policies that have been simulated for a period of

one year ('31' days for '12' iterations). If the user compares the results for the two allocation policies, they should be able to deduce that, *from this simulation run*, the 'Max–min' allocation policy compared to the 'Number–based' policy:

- Provided approximately 15 Kcal more per person per agency (across all agencies being allocated to)

- Resulted in a far superior average minimum Kcal/person per agency value (approximately 1.8 times as large)

- Satisfied less of the average agency's target need (71.94% to 87.97%, or approximately 16% less)

- Resulted in a slightly smaller average variance in the average agency's percentage of target need satisfied, although a figure of around 33% is still less than desirable

- Resulted in 5 more agencies, on average, (in total, 17 – roughly forty percent of the total number of agencies) receiving less than half of their target need

It should be easy for the user to deduce that, while the 'Max–min' allocation policy outperforms the 'Number–based' policy on three out of the five criteria, it does not necessarily mean that of itself it has performed well. The results actually indicate that both policies produce allocations that result in large variances in the amount of Kcal each person at an agency receives, whilst the average number of agencies receiving less than half of their target need is also far too large for both policies. However, one must place these results in context: FBSA's target allocation value of 700 Kcal per person across all agencies, or 'one meal per person per day' is, as it should be, ambitious for their current state of operation, i.e. with the current quantities of perishable and non–perishable donations being received by FBSA, they cannot meet this target. Hence the very reason why they are continually expanding their operations, making every effort to meet this allocation target in the *near* future, the realization of which will only prompt them to set a new, equally ambitious, allocation target. Bear in mind, FBSA wants to *eliminate* hunger in SA (see Figures 5.3 and 5.4) i.e. complete satisfaction of every agency's actual need. This information should be taken into consideration when examining the results of the simulation experiments in Chapter 9 of this dissertation.

Simulation setup

To set up a simulation, the user simply needs to select the *'Simulation Setup'* button on the homepage, which will display the *'FAST – Simulation Setup'* userform (see Figure 6.16).

The user is prompted to input the number of days they would like to simulate, where they can input any number from 1 to 365 (i.e. between one day and an entire year). The user is also required to indicate how many times (the number of iterations) they would like to simulate the number of days they have indicated. Once the user is satisfied with the values they have entered, they can select the *'OK'* button on the userform. The values input by the user are written onto the spreadsheet in the relevant place.

FAST is now ready to run a simulation[27]. To run a simulation, the user can select the blue *'Run Simulation'* button on the homepage. After selecting the button, some time (anywhere between 1 minute and 5 hours, depending on the length of the period to be simulated and the number of iterations) must be allowed for the simulation to run. Once the simulation is complete, the *Simulation Results* table will be updated with the new information for the user to examine.

---

[27]Details of the different allocation models are provided in section 8.3.
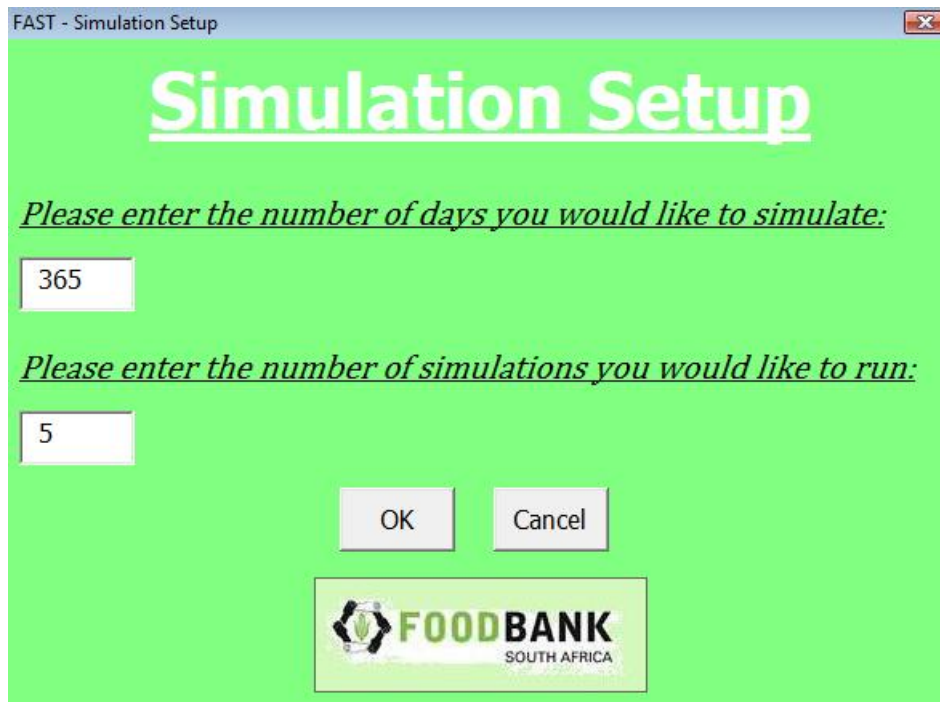
Figure 6.16: FAST – Simulation setup

The user may also examine the simulated donations by selecting any one of the food category buttons on the *Simulation Homepage* (see Figure 6.15). For example, should the user wish to view the simulated *Vegetable* donations, they can simply select the *'Veg'* button on the homepage which will display the area of the spreadsheet where these donation values are stored, as in Figure 6.17. The simulated donation values are measured in kilograms (Kg) and the user should get a feel for the range in size of donation for each of the twelve food types, as well as an appreciation of the stochastic nature of the arrival process of these donations. In particular, the user should notice the presence of '0' values amongst the non–perishable donations, corresponding to the assumption that 40% of the time, no non–perishable goods are received. Once the user is satisfied, they can select the *'Back to TOP'* button which will return them to the *Simulation Homepage*.

Figure 6.17: FAST – Vegetable donations

Chapter 7

Model Validation, Verification and Credibility

The simulation model (incorporating both allocation policies) to be used in this study has now been fully developed (for its purpose). As with any simulation model, it is imperative that it is validated and verified in order to establish whether its output is accurate (i.e. can be trusted), thus ensuring that it has a certain level of credibility amongst both its developer/s and proposed users (i.e. FBSA).

Model validation has been defined by Schlesinger et al. [24] (pg. 103) as *"substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model"*. Sargeant [68] (pg. 1) defines model verification as *"ensuring that the computer program of the computerized model and its implementation are correct"* and model credibility as being *"concerned with developing in (potential) users the confidence they require in order to use a model and in the information derived from that model"* [68] (pg. 1). A useful phrase to help distinguish between model *validation* and *verification* is to regard model validation as being chiefly concerned that the 'right' simulation model has been built, whilst model verification has as its primary concern that the simulation model was built right [74].

The validity/verification/credibility of a model should be established with respect to the intended purpose of that model [68]. The purpose of the simulation model in this study is to help aid the decision–making of the DMs involved with the allocation system at FBCT (and potentially other foodbanks in the national network in the future). It is *not* intended to serve as an accurate prediction tool. Therefore, conducting extensive validation and verification tests is not necessary. Each of model validation, verification and credibility will now be explored, with some of their techniques applied to the simulation model.

7.1   Model validation

Upon concluding the development of a simulation model, it is important to answer the question: 'Does the model function as the developer intended it to?' In answering this question, one is testing whether the behaviour of the model is a valid representation of the real–world system being simulated [74]. Three areas of the model should be examined [47]:

- The model's assumptions (are they reasonable?)

- The input parameters and (their) distributions

- The output values and conclusions

As most validation techniques inherently test the model's assumptions, and each run of the model involves both the input and output values, the above three areas will be examined concurrently rather than individually.

There are at least four broad methods to conducting model validity [68]:

1. The model developers decide on the validity of the model: Whilst this decision is based on various tests conducted on the model during and after its development, it is obviously subjective

2. The (potential) users of the model are involved in its development: This involvement spans the entire development phase, and involves periodic validations of the model by these users

3. A third party validates the model: Someone outside of the model developers, but familiar with simulation techniques validates the model

4. A scoring model is used to validate the model: Upon performing various tests on the model, scores or weights are subjectively assigned to each test and then aggregated into category scores and finally an overall score

Sargeant [68] argues that method three should only be used for large–scale models that are developed by several teams, which is not the case in this study. He discourages the use of a scoring model (4) and notes that it is used very sparingly in practice. Thus our attention will be placed on the first and second methods.

There are several validation techniques discussed in the literature (Sargeant details at least fifteen [68]). Since it has been argued that it is not necessary to conduct an extensive validation of the model, only a subset of these are mentioned here of which only three will be applied. The following validation techniques are frequently applied in the literature [68]:

- Graphical animation: The operational behaviour of the model is displayed graphically in 'real–time' as the model runs

- Comparison with other models: If similar models exist that are considered (known) to be valid, then various results from the model being validated can be compared with similar results from these valid models

- Degeneracy tests: the degeneracy of the model's behaviour is examined by choosing an appropriate set of values of the input and internal parameters and observing the behaviour of the model for this set

- Extreme condition tests: Extreme (unlikely) values of the input and internal parameters are selected (e.g. 0) and the model's behaviour and outputs are examined for these values. The model should still produce plausible results for such extreme values

- Face validity: People who have a good understanding of the system are asked whether they think the model's behaviour and outputs are reasonable (i.e. a good representation of the real–world system)

- Internal validity: The degree of (internal) stochastic variability of the model is investigated by conducting a number of runs of the model

- Parameter variability–sensitivity analysis: The values of the input and internal parameters are varied to determine their effect on the behaviour and/or output of the model

    - The relationships that exist in the real–world model should be evident in the simulation model

    - Two types of this analysis exist:

        * Qualitative – where just the change in direction of the outputs is considered
        * Quantitative – where both the change in direction and magnitude of the outputs are considered

- Predictive validation: The model is utilized to predict the real–world system's behaviour, after which comparisons between the model's forecast and the system's behaviour are made to ascertain whether they are similar

Graphical animation analysis of the model was implausible as the model does not run through a graphical interface. Seeing as there were a variety of other validation techniques to apply, applying this technique was deemed superfluous and therefore not explored. Similarly, *comparison with other models* was deemed to be too time–consuming and thus unnecessary for this study. *Predictive validation* is not useful for the purposes of the simulation model which, as expressed above, does not include it serving as a prediction tool. The simulation experiments were also run for a representative list of agencies (see section 8.1), which would make comparisons with the real–world system complicated and relatively meaningless.

Face validity is essentially a qualitative technique and would fall under the second validation method (2) outlined above. Whilst it was not explicitly employed in the development of the model, relevant members from FBSA/FBCT (those with a good understanding of the allocation system at FBCT) were involved in the development of the model by holding frequent meetings with them in which the model (at its various stages of development) was presented and discussed, taking note of their feedback. Thus, such face validity tests were implicitly conducted. A large emphasis was placed on the *transparency* of the model (see sections 5.2 and 6.3.1) which also increased through these regular meetings, ensuring that the final model was well understood and validated by relevant members of FBSA/FBCT.

Similarly, although not openly defined as a validation technique, the χ^2 goodness–of–fit tests performed on each of the three distributions fitted to the perishable and non–perishable goods (see sections 6.2.1 and 6.2.2) would certainly fall under the first validation method (1), in particular the internal validity technique. In each case, all three fitted distributions were not rejected at a significance level of $\alpha = 0.05$ which indicated that each could be used as a suitable approximation of the distribution of daily (perishable or non–perishable) donations. It needs to be emphasized here that great care was taken to ensure that the 'right' (appropriate) model of the allocation system was built. The entire problem-structuring process (Chapter 5) had as its primary aim promoting a greater understanding of FBSA and the allocation system at FBCT in order to ensure that the author of this study would have sufficient knowledge to develop an appropriate simulation model. Once again, the three days spent (by the author) at FBCT observing the allocation system first–hand (see section 5.4.1) was with a view of gaining a greater understanding of FBCT's allocation system. Thus, while the first validation method (1) generally results in subjective decisions regarding the validity of a model, these decisions are often based upon careful, comprehensive research (e.g. problem–structuring chapter) and sound statistical tests (e.g. goodness–of–fit) and should not easily be dismissed as evidence for the validity of the model.

It is can easily be argued that *degeneracy* and *extreme condition* tests would fall under the (broader) *parameter variability–sensitivity analysis* technique, and since only rudimentary validation is being conducted here, no additional tests will be conducted for this validation technique. Hence, our focus will be on

degeneracy tests, *extreme condition tests* and *internal validity*, all of which fall under the first validation method (1) described above.

The input parameters of the model are the amounts of daily donated perishable and non–perishable foods (see Figure 6.3 in section 6.3). The internal parameters are those relating to the nutritional value per Kg (Kcal/Kg) of each type of food (see Table A.1 in Appendix A), the classification of which types of food are allocated to each agency category (see Table A.2 in Appendix A) and the lower and upper bounds placed on the quantities of allocated food in the max–min model (see section 8.1.1). As already discussed (in section 6.3.3), the Kcal/Kg values of each type of food were calculated as the average of the most common food types in that category. It is this author's view that the calculated values are a reasonable approximation, as a variety of reliable sources were investigated to find the Kcal/Kg values of each of these food types, which were chosen based on the food types that the author observed during the time spent at FBCT. There would be very little, if any, variation in these values. Thus, they will remain fixed. The internal parameters relating to the classification of which food types are allocated to each category of agency will remain fixed, for the reasons provided in section 6.3.3. The lower and upper bounds in the max–min model will be varied to investigate their effect on its behaviour and output. The same performance measures (explained in section 6.4.2) were chosen to represent the output of the model as they are the key statistics of interest to both the developer (the author) and the future users of the model (members of FBSA/FBCT).

Each experiment consisted of running the model once for a length of thirty–one days (i.e. for one month). It is felt that thirty–one runs of the model is a sufficient length to investigate the effects of any changes to its parameters. What follows is a discussion of the application of each of the validation techniques being considered here. It is important to re–emphasize that the following tests are not extensive i.e. they do not involve several changes to or combinations of different parameters. Rather, simple changes are made to parameters and the behaviour of the model is observed.

7.1.1 Internal validity

As mentioned above, the χ^2 goodness–of–fit tests performed on the distributions fitted to the incoming donations data would fall under this method. Hence, some internal validity tests have already been conducted (with positive results). Since internal validity is primarily concerned with investigating the degree of stochastic variability within the model, the performance measure we are most interested in is the *average variance in % of agency's target nutritional need satisfied*. Table 7.1 shows the results generated from running the simulation model once for thirty–one days.

Table 7.1: Simulation results for internal validity tests

Allocation policy	*Max–min foodvalue*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	594.54	575.46
Avg. min. Kcal/person value	415.95	248.36
Avg. % of agency's target nutritional need satisfied	69.04	78.04
Avg. variance in % of agency's target nutritional need satisfied	21.45	22.01
Avg. num. of agencies receiving $< 50\%$ of target % need	18	15

The average variance in the % of agency's target nutritional need satisfied for both allocation policies is slightly disconcerting. It would appear that the model exhibits a higher degree of stochastic variability than what would be desired. However, this result can be explained as follows. The only stochastic element (for the purposes of this simulation study) are the perishable and non–perishable food types' donations. It is obvious that the variation in the fitted donation distributions would contribute towards, but would not by itself result in, the model's observed variation. If Table A.2 in Appendix A is examined, it is evident that certain agency categories are allocated more types of food than others. For example, feeding schemes and shelters are allocated all twelve food types, whilst nutritional centres are only allocated three food types and clinics only one. If we had a subset of nine agencies (one from each category), all with the same number of beneficiaries, then it stands to reason that feeding schemes and shelters would receive more food (hence nutrition) than nutritional centres and clinics. This effect is both somewhat amplified and reduced by the differing numbers of beneficiaries between agencies and the different nutritional values of the twelve food types. Hence, it is reasonable to assume that the food allocation classification would contribute towards the model's observed variation[1]

It is expected that the observed variation will increase somewhat for runs of the model longer than one month, simply because the above mentioned causes of variation will become more apparent. However, this variation should decrease (and eventually stabilize) as the lengths of the model runs increase.

7.1.2 Degeneracy tests

In testing for degeneracy in the model's behaviour, one is essentially trying to answer questions such as 'Will the average percentage satisfaction of an agency's need continue to increase if its allocated quantity of food increases?' Table 7.2 summarizes the original input and internal parameter values and their changed values for a number of different experiments conducted here.

Table 7.2: Changes to the model's input and internal parameters for the degeneracy tests

	Parameter values			
Experiment	*Perishable stock*	*Non–perishable stock*	*Upper bound*	*Lower bound*
Original	$Y_i = LN(Rnd(), \mu_i, \sigma_i)$	$Y_i = GAM(Rnd(), \mu_i, \sigma_i)$	2	2
1	$Y_i = [2][LN(Rnd(), \mu_i, \sigma_i)]$	$Y_i = [2][GAM(Rnd(), \mu_i, \sigma_i)]$	2	2
2	$Y_i = [0.5][LN(Rnd(), \mu_i, \sigma_i)]$	$Y_i = [0.5][GAM(Rnd(), \mu_i, \sigma_i)]$	2	2
3	$Y_i = LN(Rnd(), \mu_i, \sigma_i)$	$Y_i = GAM(Rnd(), \mu_i, \sigma_i)$	4	4
4	$Y_i = LN(Rnd(), \mu_i, \sigma_i)$	$Y_i = GAM(Rnd(), \mu_i, \sigma_i)$	1	1

The model's behaviour and output was noted for each experiment, and is summarized in Tables A.3, A.4, A.5, A.6 and A.7 in Appendix A. What follows is a brief discussion of the results for each of the four experiments, in which any 'out of the ordinary' results are highlighted.

Experiment 1
Here the quantities of both the perishable and non–perishable food types (indicated by the μ_i and σ_i in the donation distributions in Table 7.2 above) were doubled. We would expect the *average Kcal/person*

---

[1]It is obvious that this is only a conjecture. The author is confident that it accounts for the majority of the observed variation, but recognizes that some statistical testing would need to be conducted to confirm this assumption. However, such testing is beyond the scope of this half–dissertation.

value, the *average minimum Kcal/person value* and the *average % of agency's target nutritional need satisfied* to increase proportionately, the *average variance in % of agency's target nutritional need satisfied* to increase and the *average number of agencies receiving < 50% of target % need* to decrease. The model behaves as expected (with the *average number of agencies receiving < 50% of target % need* decreasing to a greater extent than expected).

Experiment 2
Here the quantities of both the perishable and non–perishable food types were halved. We would expect the *average Kcal/person value*, the *average minimum Kcal/person value* and the *average % of agency's target nutritional need satisfied* to decrease proportionately, the *average variance in % of agency's target nutritional need satisfied* to decrease and the *average number of agencies receiving < 50% of target % need* to increase. The model behaves as expected.

Experiment 3
Here both the lower and upper bounds placed on the quantity of allocated food to each agency in the *max–min* model are doubled (i.e. relaxed). We would expect the *average Kcal/person value* to decrease and the *average minimum Kcal/person value* to increase, resulting in them being closer together as there is now more 'freedom' within which Whats*Best*! can solve the model, resulting in more balanced allocations that have a higher minimum Kcal/person value per agency, but more agencies receiving this value resulting in a lower average Kcal/person value per agency. Since these two changes counteract one another, we would not expect any substantial change in either the *average % of agency's target nutritional need satisfied* or the *average number of agencies receiving < 50% of target % need*. We anticipate a decrease in the *average variance in % of agency's target nutritional need satisfied* for the max–min model. The model behaves as expected. In fact, the *average Kcal/person value* and *average minimum Kcal/person values* are identical.

Experiment 4
Here both the lower and upper bounds placed on the quantity of allocated food to each agency in the *max–min* model are halved (i.e. tightened by essentially removing them from the model). This essentially forces the max–min model to allocate quantities of food to agencies based on the number of people they support, i.e. perform allocations like the number–based model, which essentially rules out the need for an optimizer to solve the model. This should create a problem in the solution of the max–min model, as there is now no 'leeway' for the optimizer (Whats*Best*!) to solve the model – i.e. it is forced to allocate a particular quantity (the pro–rata quantity defined in section 6.4.1) to each agency, which will result in it being unable to change the adjustable cells and hence solve the model. The model behaves as expected and the problem identified here is reflected in the statistic *Number of unsolvable iterations* in Table A.7 (thirty of the thirty–one iterations were unsolvable). In view of this statistic, the results for the max–min model in Table A.7 should be ignored.

Reviewing the results of the four experiments, the model evidently behaves as we would expect it to (at times producing positive results that exceeded our expectations).

7.1.3 Extreme condition tests

In this area we are testing to see if the model still produces plausible results for extreme values of the input parameters. For the scope of this study, we will conduct only two tests: one where the incoming donations are set to '0' value and one where they are increased to ten times their current level.

Experiment 1
Here the incoming perishable and non–perishable donations are set to '0' value. We would expect the

average Kcal/person value, the *average minimum Kcal/person value*, the *average % of agency's target nutritional need satisfied* and the *average variance in % of agency's target nutritional need satisfied* to be '0', whilst the *average number of agencies receiving < 50% of target % need* should be '42'. The results are summarized in Table A.8 in Appendix A. The model behaves as expected.

Experiment 2

Here the incoming perishable and non–perishable donations are increased to ten times their current level. As in the first 'degeneracy' experiment, we would expect the *average Kcal/person value*, the *average minimum Kcal/person value*, the *average % of agency's target nutritional need satisfied* to increase proportionately, the *average variance in % of agency's target nutritional need satisfied* to increase and the *average number of agencies receiving < 50% of target % need* to decrease. The results are summarized in Table A.9 in Appendix A. The model behaves as expected (with the *average number of agencies receiving < 50% of target % need* decreasing to '0' value).

Taking the above results into consideration, we can argue that the model behaves well under extreme conditions.

7.1.4 Reflection

Should further work ever be conducted utilizing the simulation model developed in this study, more extensive validation would be warranted. However, seeing as the model has been tested with three validation techniques and generally performed well (according to our expectations), there is good reason to assume that the simulation model is valid for the purposes of this study.

7.2 Model verification

Model verification essentially ensures that the computer programming underlying the simulation model is correct and that its implementation of the conceptual model (the mathematical representation of the system under study) is appropriate i.e. it implements the model's assumptions correctly [68].

Mellor–Crummey [47] details (amongst others) the following model verification tests:

- De–bugging and error checking of programming code

- On–line graphical visualizations: various parameters in the model are graphed as the model is run, and any changes in these parameters can be observed in 'real time'

- Continuity tests: multiple runs of the model are conducted with slightly different parameter values for each run. The results of each run are investigated for any sudden changes in the model's output

- Degeneracy tests: the model is tested for extreme cases of parameter values to ensure that it works as it is supposed to for these values

- Consistency tests: the model is tested to see if similar results are achieved for parameters that are expected to have similar effects

Regarding the first technique pertaining to de–bugging and error checking of the programmed code, the author (also the developer of the code) thoroughly scrutinized the simulation model's underlying programmed code numerous times. In addition to this, the programming language used in Microsoft Excel VBA, Visual Basic, comes with built–in error checking and de–bugging techniques. If any variable is not

properly defined or dimensioned, or any line of code incomplete, or any logic of a statement or construct incorrect, then upon attempting to run the model an error message will be returned detailing what is wrong, with the user being taken to the line of code where the problem lies (which is also highlighted in yellow). VBA provides useful help topics to assist the user in de–bugging these issues. The Excel add–in Whats*Best*! also comes with built in error–checking techniques as well as status and solution reports that can be generated. Status reports will detail any errors in the model formulation and display their exact location. The solution reports display the values of all the variables in the model as well as the optimal objective value. The developer of the simulation model also programmed VBA to display how many runs of the model contain errors (i.e. are not solved) and only the results from those runs that displayed '0' for this value were considered.

As has already been mentioned, on–line graphical visualizations were not possible to conduct. The remaining tests (or ones very similar in purpose and technique) have all already been conducted as part of the model validation, and therefore do not need to be repeated here. Taking all the above into consideration, it is clear that the underlying code has been comprehensively checked (and re–checked) to ensure that it implements the conceptual model correctly.

7.3 Model credibility

As mentioned above, model credibility involves the instilling of confidence amongst any user of the model (i.e. developer or client) in both utilizing the model and trusting the output of the model. A concept very closely related to this is that of the *transparency* of a model, already mentioned numerous times in this study. It is this author's belief that involving the potential users of a model in its development (as was done in this study through holding regular meetings with FBSA/FBCT) results in the model acquiring a high level of transparency, and consequently a high level of credibility.

Upon completing the development of the simulation model, and specifically the DSS FAST (in which the simulation model is embedded), FAST was demonstrated to both the management and ground staff at FBCT. Both the management (all having at least good basic computer skills) and the ground staff (some having no knowledge of how to utilize a computer) understood the logic behind the model - how it simulated the arriving donations and performed allocations. Each function of FAST was carefully and thoroughly demonstrated, and the relevant staff who would be operating the model all indicated that they understood the model, it was easy to use and that they would be able to utilize it effectively[2].

---

[2]Note that a verbal agreement was made that the author of this study would be available for the first quarter of 2011 to act as a 'technical adviser' for FAST, ensuring that someone with excellent knowledge of the system would, on request, be able to help with any operational problems if need be.

Chapter 8

Simulation Experiments

The purpose of this chapter is to detail the simulation experiments conducted of the allocation system at FBCT, as outlined in section 6.3.3. The simulation model presented in section 6.3.4 constitutes the 'base–model' upon which the allocation models presented here are built.

8.1   Allocation models

Whilst it was initially envisioned that a range of different allocation models (or policies) would be developed as a result of this simulation study, limitations on both the scope and time available for this half–dissertation[1] resulted in there being only two allocation policies considered, as outlined in section 6.4.1:

- A 'Max–min Allocation' policy, and

- A 'Number–based Allocation' policy

Shannon [74] notes that a great strength of conducting simulation experiments is that one is able to simulate systems already in existence as well as investigate the impact of other feasible systems that could be brought into existence. This is precisely what has been done in our study with the two allocation policies considered. As already noted, the 'Number–based Allocation' policy is essentially the allocation policy that FBSA is currently employing. The 'Max–min Allocation' policy was developed in response to the desire for such an allocation policy expressed by members of FBSA/FBCT management during meetings held with them.

It is important to note that the simulation model presented here is not intended to be a 'carbon copy' of the allocation system at FBCT, but rather a *representation* of the system. Simulation models, in general, are developed to be run rather than solved. They are utilized to conduct experiments to investigate the behaviour of a system and they do not yield an 'optimal' solution to a system [74]. With this in mind, a *representative* list of agencies was constructed by determining:

- The average number of agencies allocated food each day (42)

---

[1]As with most dissertations, a great deal more work could have been conducted if more time was available. See section 10.2.1 for an explanation of both these limitations.

- The average number of each category of agency allocated food each day (see Table 8.1)

- The average number of people supported per agency of each *category of agency* allocated food each day (see Table 8.1)

Table 8.1: Representative agency category information for simulation model

Agency category	Average number	Average number of people per agency
Clinic	1	114
Educare	15	79
Feeding Scheme	3	572
Nutritional Centre	1	605
School	2	396
Shelter	7	213
Soup Kitchen	6	430
Support Group	7	217

This information was derived from FBCT's agency database[2]. The average number of people supported per agency of each category of agency was divided randomly amongst the number of agencies of that category in such a way as to ensure that this number was closely matched.

Before the simulation experiments are presented, the two allocation models will be detailed by providing both a description and mathematical formulation for each one.

8.1.1 'Max–min foodvalue allocation' model

As previously mentioned, this allocation model attempts to 'balance' the Kcal/person value across all agencies, i.e. maximize the minimum Kcal/person value across all the agencies. Performing such an allocation makes sense as it is not only the quantity of food allocated to an agency that is important – the nutritional quality (value) of the allocated food is equally (if not more) important. This 'nutritional quality' of food is what corresponds to FBSA's desire to improve the 'accuracy' of their allocations (see sections 3.3 and 5.4.1).

The allocation is performed by utilizing the Excel add–in Whats*Best!*, a powerful optimization platform, within an Excel VBA code in the following manner:

- The decisions regarding how much of each type of food to be allocated to each agency are represented in the model by using 'adjustable cells'

  - 'Adjustable' simply implies that these cells are allowed to change value, within certain imposed bounds, during the optimization process (i.e. solving of the model)

- The array of adjustable cells for each agency are multiplied by a corresponding array of '1's and '0's, depending on which types of food are to be allocated to that category of agency

---

[2]Note that FBCT allocates food to other 'Satellite' foodbanks only very seldom, and thus this agency category does not form part of the representative list.

- This array forces the model to allocate only those types of food that the particular agency category should receive, as depicted in Table A.2 in Appendix A

  * For example, if the agency is a Clinic, the array would be [1 0 0 0 0 0 0 0 0 0 0 1]. This would force the model to allocate only 'bakery' goods and 'other' goods to the agency

- In this manner, the decision regarding what type of food to allocate to each agency is represented in the model

- The quantity of each type of food to be allocated to the agency is multiplied by the nutritional value (Kcal/Kg) of that type of food (see Table A.1 in Appendix A), these Kcal values are summed across all the food types the agency is allocated, and then divided by the number of people supported by the agency to get the Kcal per person value for that agency

  - These quantities of allocated food are constrained to be greater than or equal to half of the agency's 'pro–rata' amount, but less than or equal to double this amount (as detailed in the subsection 'What FAST can do – Allocating to agencies' in section 6.4.1)

    * These lower and upper bounds on the quantity of allocated food are incorporated into the model to ensure that no agency is allocated too great a percentage of a particular type of food. For example, without these bounds, it is plausible that an agency may be allocated more than half of the bakery stock, and very little else. Such an allocation would be considered to be unbalanced.

    * The bounds also incorporate the number of people supported by each agency into the model (see section 6.4.1)

- The total quantity of each type of food to be allocated is summed, and is constrained to be less than or equal to the available stock[3]

- Finally, one more adjustable cell is added to the model. This adjustable cell is constrained to be less than or equal to each of the Kcal per person values (V_p) for the quantities of allocated food to each agency. Upon solving the model, the value of this cell is equivalent to the maximum minimum foodvalue per person across all the agencies

  - It is the addition of this last adjustable cell that facilitates the 'balancing division', i.e. maximizing the minimum foodvalue across all agencies. Upon solving the model Whats*Best!* will attempt to maximize this cell by adjusting the values of each of the X_{ip}'s (adjustable cells), whilst adhering to the constraints placed on the model, which in turn will result in Whats*Best!* determining the highest common Kcal per person value across the agencies. This value will be the maximum minimum foodvalue per person (of the allocated quantities of food) across all the agencies

The mathematical formulation of the 'Max–min allocation' model is the same as that of the simulation model outlined in section 6.3.4, with the following additional formulation:

- Let $lb_{ip} \equiv$ lower bound placed on quantity of food type i allocated to agency p, and

- $ub_{ip} \equiv$ upper bound placed on quantity of food type i allocated to agency p,

- The 'pro–rata' quantity of food type i to be allocated to agency p is:

$$PR = \frac{X_{ip} \times T_p}{W_i}$$

---

[3]The daily available quantities of each type of food are randomly generated at the beginning of the simulation, details of which are provided in section 8.2.

- The bounds placed on the quantity of food type i allocated to agency p are[4]:

$$\frac{PR}{lb_{ip}} \leq X_{ip} \leq ub_{ip}(PR)$$

- Let $D \equiv$ the adjustable cell representing the maximum minimum foodvalue per person of the allocated quantities of food across all agencies to be maximized. Then:

$$D \leq V_p, \quad \forall\, p = 1, 2, \ldots s$$

Algebraic formulation of 'Max–min foodvalue allocation' model

Maximize D

Subject to:

$$D \leq V_p, \quad \forall\, p = 1, 2, \ldots s$$
$$\frac{PR}{lb_{ip}} \leq X_{ip} \leq ub_{ip}(PR)$$
$$\sum_{p=1}^{s} X_{ip} \leq Y_i, \quad i = 1, 2, \ldots 12$$
$$X_{ip} \geq 0 \quad \forall\, i,\, p$$

8.1.2 'Number–based equal allocation' model

This allocation model is essentially a representation of the allocation system currently in use at FBCT. Available stock is divided amongst agencies exclusively according to the number of people being supported by each agency, hence the name 'number–based allocation'.

The allocation is performed by an Excel VBA code in the following manner:

- The decisions regarding what types of food are allocated to each category of agency are incorporated into the model in the form of a 'Select case' structure:

    - A select case structure is simply a construct of code that performs certain actions depending on what 'case' a variable takes on

    - Here the variable is the *agency category*, 'agencycat', and the cases are the nine different categories of agency

    - For each case, 'allocation binary variables' for each of the 12 food types take on either a '0' or '1' value, depending on what types of food are to be allocated to the particular category of agency

        * For example, if the agency's category is 'Shelter', then VBA will skip all the other cases of the variable 'agencycat' before 'Shelter'[5] and then execute the code that pertains to

---

[4]Note that the default value for both the lower and upper bounds is '2'.

[5]The cases are arranged in alphabetical order in the VBA code

the agency category 'Shelter', i.e. each allocation binary variable will be set to '1' since shelters are allocated all twelve food types

- The decisions regarding the quantity of each food type to be allocated to each agency are included in the model by defining 'allocation factors' for each food type for each agency.

  – Each allocation factor is calculated by taking the product of the *allocation binary variable* corresponding to the agency's category and the *number of people supported by that agency*. This value is then divided by the total number of people across all the agencies who are to be allocated that food type to get the corresponding allocation factor

  – The actual quantity of each food type to be allocated to each agency is then calculated as the product of each agency's allocation factor for each food type with the quantity of available stock of that food type

The mathematical formulation of the 'Number–based allocation' model is the same as that of the simulation model outlined in section 6.3.4, with the following additional formulation:

- Let

  – $Allo_{ip} \equiv$ agency p's allocation binary variable for food type i, $i = 1, 2, \ldots 12$

  – $AgencyAllo_{ip} \equiv$ agency p's allocation factor for food type i, $i = 1, 2, \ldots 12, p = 1, 2, \ldots s$

- Then $\forall\, i = 1, 2, \ldots 12,$ $p = 1, 2, \ldots s$

$$AgencyAllo_{ip} = \frac{Allo_{ip} \times T_p}{W_i}$$

$$X_{ip} = AgencyAllo_{ip} \times Y_i$$

Algebraic formulation of 'Number–based equal allocation' model

This model is not of an optimization–type. The way in which food is allocated is based upon pre–determined decisions (like the classification of what food types are to be allocated to which agency types), which for the purposes of this simulation study are not being varied. Should further work be carried out in which these decision rules can vary, then the model would be of the optimization–type.

8.2   Simulation code

Before detailing the simulation experiments conducted in this simulation study, it is of value to include a brief description of the Excel VBA code that performs the simulations[6]. The VBA code includes both allocation models described above, and each simulation run encompasses simulating each of these models.

The VBA code is structured as follows:

- The values for the number of days to simulate (n) and number of iterations to perform (k) are read in from the Excel spreadsheet into the VBA code (see Figure 6.15)

---

[6]The actual VBA code for the simulation model is included in Appendix B.

- Arrays to store the simulated daily donations of each of the twelve food types are created (defined)

- These daily donations are simulated and displayed on the Excel spreadsheet (see Figure 6.17)

- VBA runs through the representative list of agencies, calculating:

  - The number of agencies of each agency category

  - The total number of people of each agency category to be allocated food (across all agencies)

  - The total number of people to be allocated food per each food type (across all agencies)

  - The total number of people to be allocated food (across all agencies)

- VBA performs the Max–min allocation (as described above)

- VBA performs the Number–based allocation (as described above)

- VBA computes the relevant descriptive statistics and performance measures (as detailed in section 6.4.2) for each simulation run for each allocation model

- Upon simulating the specified number of days for the desired number of iterations, VBA computes the averages of the performance measures and descriptive statistics and summarizes them in a table on the Excel spreadsheet (see Figure 6.15)

8.3   Simulation experiments

The initial experiments conducted here focus on simulating the *Max–min* and *Number–based* allocation models for different lengths of time (periods), observing their individual performances, and comparing their performances. It is important to consider different simulation periods as the performance of each of the models may vary over time. It was decided to consider three time periods:

- Short–term: one month

- Medium–term: one year

- Long–term: five years

Hence, three simulation experiments were conducted using the FAST Simulation DSS[7]. They are outlined in the following sections, with their results and consequent discussion presented in Chapter 9.

8.3.1   Short–term simulation

- Number of days to simulate, $n = 31$

- Number of iterations to run, $k = 1$

---

[7]Whilst it is acknowledged that shorter simulation runs inevitably lead to more variable results, the rationale behind including different lengths of simulations is that it will be of value to observe the extent of the difference in variance of the results obtained for each of these simulation experiments. Another possible simulation strategy may be to settle on one simulation length (at least a year) and run three experiments with that length and then compare the results.

The simulation model was run for thirty one days for one iteration, to represent simulating one month of the allocation system at FBCT. On average, FAST takes approximately four minutes to perform a simulation of this length. It was decided that the shortest time period over which to simulate the model should be one month, as simulating the model over a week would not allow for the variations inherent in any stochastic simulation to be accounted for. This simulation experiment should provide a good initial indication of how both allocation models will perform, but should not be viewed as a blueprint for their performances, as it will be important to see how they perform over longer periods of time. The results of this simulation are summarized (see Table 9.1) and discussed in section 9.1.1.

8.3.2 Medium–term simulation

- Number of days to simulate, $n = 31$
- Number of iterations to run, $k = 12$

The simulation model was run for thirty one days for twelve iterations, to represent simulating one year of the allocation system at FBCT. On average, FAST takes approximately 50 minutes to perform a simulation of this length. Simulating the allocation policies for this length of time should provide us with results that are a good indication of each policy's performance. Since a year is a significantly longer time period than one month, the variations in the stochastic elements of each allocation model should be well accounted for. Thus, the results of this simulation experiment will supply us with a far more robust indication of how the two policies perform. The results are summarized (see Table 9.3) and discussed in section 9.1.2.

8.3.3 Long–term simulation

- Number of days to simulate, $n = 365$
- Number of iterations to run, $k = 5$

The simulation model was run for three hundred and sixty five days for five iterations, to represent simulating five years of the allocation system at FBCT. On average, FAST takes approximately four hours to perform a simulation of this length. The results obtained from this experiment will be the most robust indication of the performance of each of the allocation policies. It is important to simulate the model for this length of time as a longer simulation period will sufficiently cater for variations in the stochastic elements of each allocation model, and provide a long–term view of the performance of each policy. However, these benefits of longer–term simulations only extend so far, and are offset by the greater amount of time needed to perform them. Thus, it was decided that simulating the model for longer than five years would be unnecessary. The results of this experiment are summarized (see Table 9.4) and discussed in section 9.1.3.

Chapter 9

Results and Discussion

The purpose of this chapter is to provide a summary of the results of the simulation experiments carried out in Chapter 8, accompanied by a comprehensive analysis and discussion of these results (section 9.1). The feedback obtained from members of FBCT on the operation of the DSS FAST will also be included and discussed (section 9.3).

9.1 Simulation results and discussion

Each of the three simulation experiments' results are presented in tabular form, followed by an analysis and discussion. A summary of the number of days (n) and number of iterations (k) of each experiment is provided at the outset of each section.

9.1.1 Short–term simulation

$n = 31$
$k = 1$

Table 9.1 summarizes the results obtained from simulating the two allocation policies for a period of one month. The representative list of agencies (42 in total), amounting to a total of 10002 people, were allocated food on a daily basis for 31 days[1].

Analysis and discussion of short–term results

What follows is an analysis and discussion of each of the (five) performance measures detailed in Table 9.1.

---

[1]Note that the representative list of agencies was the same in all three experiments. Hence both the number of agencies (42) and total number of people allocated food (10002) is the same in each experiment, and no further comment will be made on these two statistics.

Table 9.1: Short–term simulation results

Allocation policy	*Max–min*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	574.85	534.57
Avg. min. Kcal/person value	344.93	191.35
Avg. % of agency's target nutritional need satisfied	62.26	74.11
Avg. variance in % of agency's target nutritional need satisfied	20.66	18.78
Avg. num. of agencies receiving < 50% of target % need	22	15

Average Kcal/person value

As indicated in section 6.4.2, this statistic is a good measure of the overall performance of each allocation policy. At present, FBSA's allocation target is 'one meal per person per day' for each agency, which equates to approximately 700 Kcal per person (see section 6.3.3). Thus, the closer this value is to the target value of 700 Kcal, the better the performance of the policy. The max–min policy results in an average value of approximately 575 Kcal/person, whilst the number–based policy results in an average value of approximately 535 Kcal/person. Thus, based on this simulation experiment, the max–min policy outperformed the number–based policy by providing, on average, 40 Kcal more per person across all the agencies in the representative list. That both policies average more than 530 Kcal/person is an encouraging statistic as it indicates that, on average, either policy satisfies at least 75% of a person's daily nutritional requirements (max–min: approximately 82%, number–based: approximately 76% for this simulation experiment). This would be encouraging for FBSA, as it shows that with the current volume of donated goods they are close to reaching their allocation target.

Average minimum Kcal/person value

This performance measure focuses on the minimum Kcal/person value across all agencies. Ideally, FBSA would want this value to be much the same as the average Kcal/person value, as this would indicate a low variance in the Kcal/person values across all the people being supported by the agencies. One would expect the max–min policy to outperform the number–based policy here, as maximizing the minimum Kcal/person value is precisely the objective of the max–min model. Thus, the fact that the max–min policy results in an average minimum Kcal/person value (approximately 345 Kcal/person) that is more than 1.8 times the corresponding value that the number–based policy achieves (approximately 191 Kcal/person) should not come as a complete surprise. What is more important to note is that the max–min policy's statistic is consequently much closer to its average Kcal/person value (approximately 60% of this value) when compared with the number–based's corresponding statistic (only approximately 36%). Thus, the max–min policy significantly outperforms the number–based with respect to this performance measure.

Figure 9.1 displays (graphically) the *average Kcal/person* and *average minimum Kcal/person* values for both allocation policies. From the figure it is evident that the max–min policy distinctively outperforms the number–based with respect to both of these performance measures for the short–term simulation.

Average % of agency's target nutritional need satisfied

This statistic presents a more salient view of the performance of each policy. Whilst it is useful to know how each policy performs with respect to the individuals being supported by each agency, the interpretation of the previous two statistics can be slightly ambiguous in the sense that they measure the performance of each policy *across* all agencies, and do not focus on performance with respect to *each* agency. The average
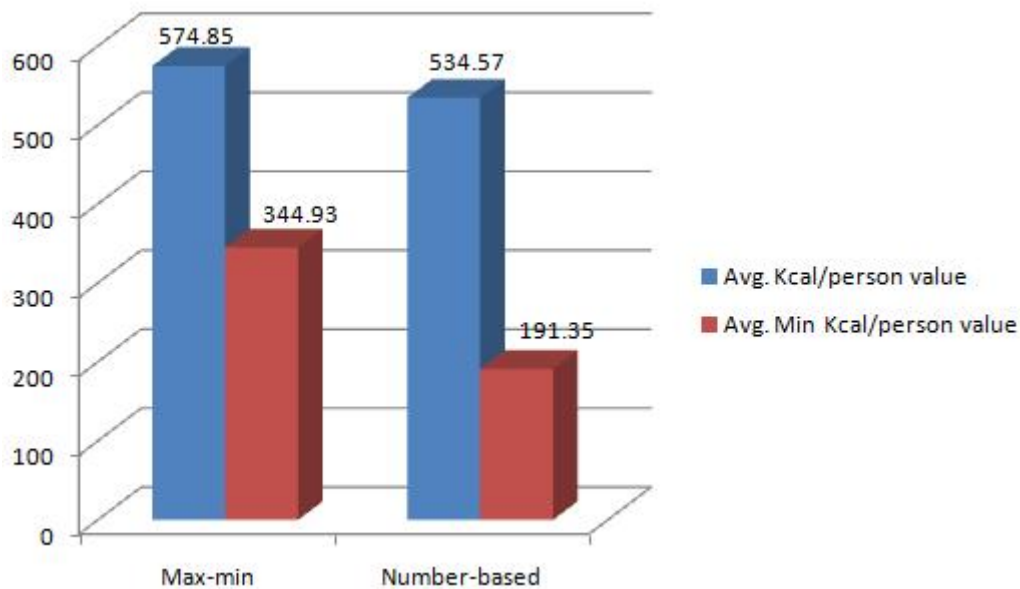
Figure 9.1: Average Kcal/person and minimum Kcal/person values for each allocation policy

% of an agency's target nutritional need satisfied measures the performance of each policy with respect to each agency being supported, and thus provides us with a more holistic performance measure. Obviously, FBSA would want this measure to be as close to 100% as possible. However, with their present resources (donations, warehouse space, delivery vehicles etc.) a value of approximately 75% would be deemed acceptable for now.

On average, the max–min policy satisfies only approximately 62% of an agency's target need[2]. This value seems confusing when viewed in light of the previous two statistics. However, it is important to interpret this statistic correctly. The max–min model attempts to *balance* the Kcal/person value in each agency *across all forty–two agencies*. It is critical to understand the implications of this. What this implies is that the max–min model will often arrive at a solution in which approximately 90% of the agencies are allocated quantities of food that result in them receiving the same Kcal/person value (the *minimum Kcal/person value* as above). Whilst this is favourable in terms of achieving nutritionally balanced allocations across the agencies, the Kcal/person value arrived at may be substantially lower than what the *average minimum Kcal/person value* would suggest, thus resulting in a lower percentage of an agency's target need satisfied value on average. This is visible in Table 9.2, which depicts a subset of *minimum Kcal/person* values for fifteen agencies for five solutions of the model.

The reason for this discrepancy is that, whilst the majority of agencies receive the minimum Kcal/person value, a few may receive considerably more than this (sometimes five or six times the amount, as in the first and third columns of Table 9.2), thus resulting in an inflated value for the *average minimum Kcal/person* statistic. The same would not apply to the number–based model which, although having a far lower *average minimum Kcal/person value*, would allocate food in such a way that perhaps only one or two agencies would have this value with the rest having higher values.

---

[2]Where 'target' refers to FBSA's allocation target of 700 Kcal per person per day which is easily translated into the agency's total Kcal needs per day.

Table 9.2: A sample of minimum Kcal/person values from five runs of the max–min model

Agency	Kcal/person	values			
Care Haven	1681.21	363.96	2271.29	335.18	331.90
Enkosi	1585.26	363.96	2011.16	233.67	331.90
Little Angels	1823.41	363.96	2372.28	335.18	336.23
Malindi	1681.21	363.96	2271.29	128.27	331.90
Noah	1824.32	363.96	2375.98	233.67	331.90
St Pauls Primary	603.79	363.96	593.99	233.67	445.97
The Ark	568.01	363.96	567.82	115.06	331.90
Ubuntu Open Door	568.01	363.96	567.82	115.06	331.90
Yakulunthu	396.50	363.96	126.46	233.67	331.90
Zingisani	390.34	363.96	87.28	115.06	331.90
Min. Kcal/person value	390.34	363.96	87.28	115.06	331.90

Hence the reason for the *average % of agency's target nutritional need satisfied* being a more reliable measure of a policy's performance.

Average variance in % of agency's target nutritional need satisfied

This statistic measures the variation in the percentage of an agency's need that was satisfied by each allocation policy. It is an important performance measure, and a critical indication of the overall performance of each policy. Ideally, FBSA would want this statistic to be as close to '0' as possible. However, as detailed in section 7.1.1, there are elements of the simulation model that will give rise to some variation. The number–based model results in a slightly lower statistic (18.78) than the max–min model (20.66). The statistics for both policies are higher than what would be desired – the reasons for which were explained in section 7.1.1. The performance of both policies in this area is poor, but it is important to remember that FBSA is a young, burgeoning organization that is constantly growing and improving. They are investigating the option of procuring food from wholesalers in order to increase the amount of incoming food and the certainty of the quantity of this food. Should they opt to incorporate this into their operations, it would certainly help to reduce the variation (and hence uncertainty) in the quantity of incoming food and consequently in the quantity of allocated food. FBSA certainly would like to see the *average variance in % of agency's target nutritional need satisfied* decrease in the near future.

Average number of agencies receiving < 50% of target % need

This statistic gives us another indication of how each policy is performing. It is obvious that FBSA would always want this statistic to be '0', thus ensuring that, at the very least, they are satisfying half of each agency's needs. Bearing in mind that the number of agencies in the representative list is forty–two, the results for both policies are very alarming. On average, the max–min policy has twenty–two (more than half) of the agencies having less than half of their needs satisfied, whilst the number–based policy has fifteen (approximately 36%). These statistics give us an indication of just how far away FBSA is from achieving their allocation target at present. These results can also be explained by the logic presented in the *Average % of agency's target nutritional need satisfied* section above: the max–min policy results in most of the agencies having a Kcal/person value of or close to the *minimum Kcal/person value* which may sometimes fall under 350 Kcal (i.e. less than half of the allocation target), resulting in most of the agencies receiving less than fifty percent of their target need in those cases. Once again, the same would not be true of the number–based policy, which would generally result in some agencies frequently receiving less than fifty percent of their target need (e.g. clinics), whilst others (e.g. shelters) most often receive more than half of their target need.

FBSA would be resolved to see this statistic always reading a '0' value in the future.

9.1.2   Medium–term simulation

$n = 31$
$k = 12$

Table 9.3 summarizes the results obtained from simulating the two allocation policies for a period of one year. As indicated earlier, it is believed that simulating the model for a period of one year will yield results that are a more accurate representation of the performances of each policy, as the inherent variation present in the model will be more accounted for by the length of a year.

Table 9.3: Medium–term simulation results

Allocation policy	*Max–min*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	660.14	632.32
Avg. min. Kcal/person value	354.54	201.07
Avg. % of agency's target nutritional need satisfied	68.82	84.31
Avg. variance in % of agency's target nutritional need satisfied	32.43	34.85
Avg. num. of agencies receiving $< 50\%$ of target % need	18	14

Analysis and discussion of medium–term results

What follows is an analysis and discussion of each of the performance measures detailed in Table 9.3. The results will also be compared with those of the short–term simulation.

Average Kcal/person value
On average, both policies resulted in more than 630 Kcal being provided to each beneficiary across all the agencies, which is over ninety percent of the target Kcal value. This indicates a good performance for both policies and is an encouraging statistic, considering that both policies provided more Kcal per person in this simulation than in the short term simulation. Figure 9.2 shows that on average the max–min policy provided approximately 660 Kcal per person (approximately 85 more than in the short–term simulation) and the number–based provided approximately 632 Kcal per person (approximately 97 more than in the short–term simulation). Whilst these statistics indicate that the number–based policy improved more (with respect to this performance measure) than did the max–min policy, the max–min policy still performed better, providing on average approximately 28 more Kcal per person than the number–based policy.

Average minimum Kcal/person value
As in the short–term simulation, the max–min policy produces a far higher *average minimum Kcal/person* value (approximately 354 Kcal/person) than does the number–based policy (approximately 201 Kcal/person). On average, the statistic for both policies is approximately 10 Kcal/person higher than in the short–term simulation, which is an encouraging result. However, this improvement is smaller than the corresponding improvement in the *average Kcal/person* value, which would suggest that there may be an increase in the
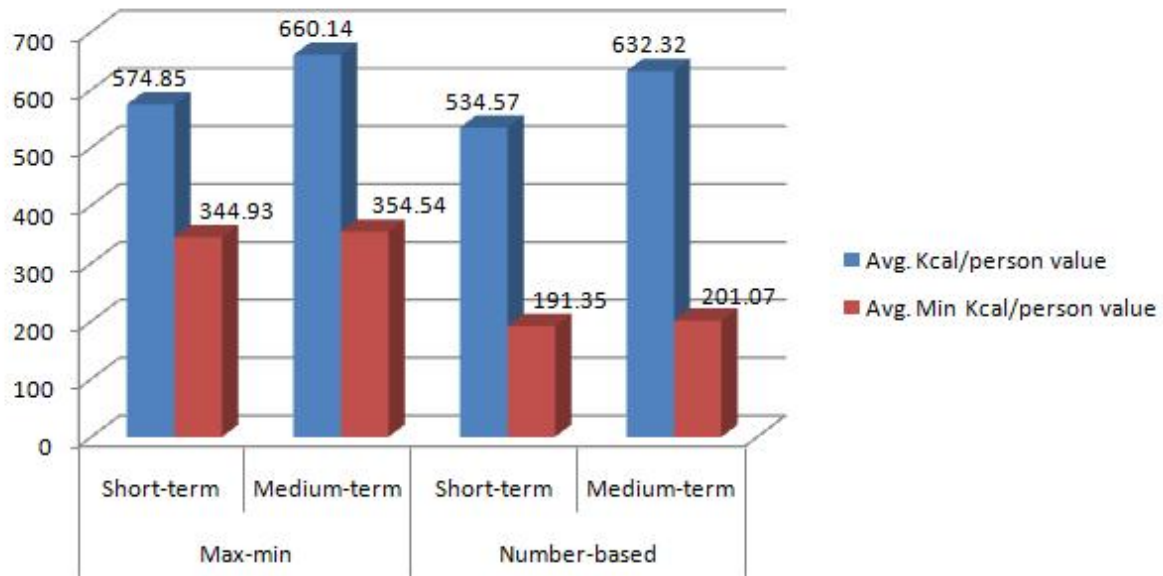
Figure 9.2: Average Kcal/person and minimum Kcal/person values for each allocation policy for both the short and medium–term simulations

average variance in % of agency's target nutritional need satisfied as the values for these two statistics are further apart here (for both policies) than in the short–term results. It is evident that the same explanation provided (in the short–term results) for the large difference in this performance measure between the policies would still apply here. The results for this performance measure are compared graphically with those from the short–term simulation in Figure 9.2. The improvement of each policy can be seen.

Average % of agency's target nutritional need satisfied
Both policies show improvements in this performance measure. The max–min policy satisfies, on average, approximately 69% of an agency's target need (an improvement of approximately 6.5%), whilst the number–based policy satisfies, on average, approximately 84% (an improvement of approximately 10%). Thus, as in the *average Kcal/person* statistic, the number–based policy demonstrates more improvement than its counterpart. As in the short–term results, the number–based policy outperforms the max–min (even more strongly here) with regards to this performance measure. Overall, it is reassuring that both policies do show improvement. Figure 9.3 displays these results for both the short and medium–term simulations. The superior performance of the number–based policy with regards to this performance measure is distinct.

Average variance in % of agency's target nutritional need satisfied
The results obtained for both policies show a large increase in this statistic. Both policies showed an increase of more than 10% in this variation. In fact, it almost doubles for the number–based policy. The results are displayed in Figure 9.3, where the large increase in variation for both policies is evident. These results are certainly alarming, but not completely unexpected. It was predicted that there would be an increase in this variation (see section 7.1), but what is surprising is the magnitude of this increase (in particular for the number–based policy). There is now, on average, more variation in the satisfaction of an agency's target need for the number–based policy than its counterpart. This large variation produced by the number–based policy is also evident in the large difference between its *average Kcal/person* (632.32) and *average minimum Kcal/person* (201.07) values, which is much greater than the corresponding difference in the max–min policy's results. It is difficult to decide whether the increases in this performance measure
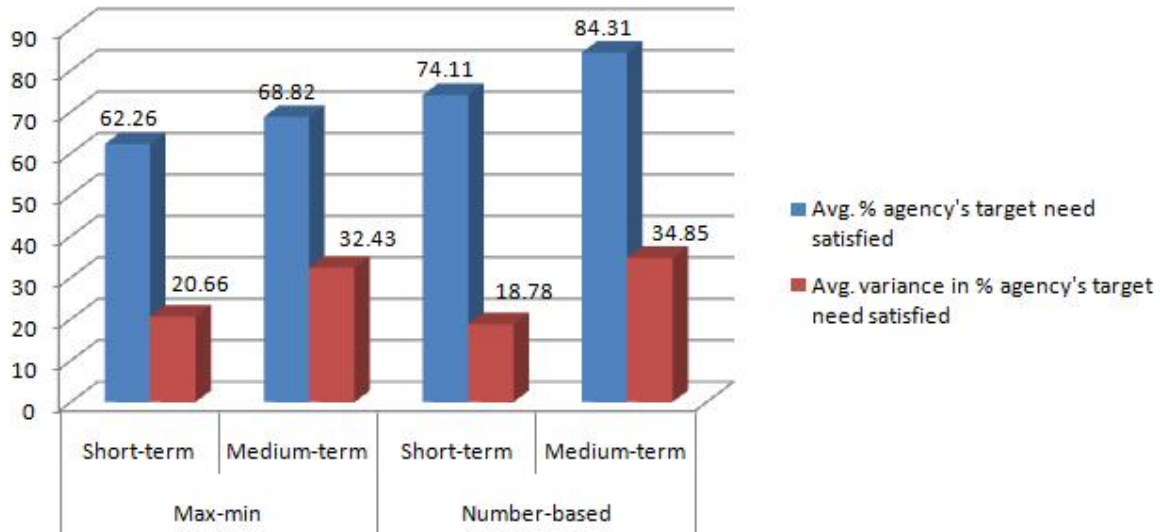
Figure 9.3: Average % of agency's target nutritional need satisfied and the average variance in this % for each allocation policy for both the short and medium–term simulations

for both policies can be solely attributed to the variation inherent within the model finding greater expression due to the longer length of the simulation. However, what is apparent is that the max–min policy performed better than the number–based policy with respect to this performance measure.

Average number of agencies receiving < 50% of target % need

Both policies demonstrated an improvement with respect to this performance measure. On average, the max–min policy resulted in 18 agency's receiving less than half of their target need, four less than in the short–term results (i.e. an improvement of nearly 20%), whilst the number–based policy resulted in 14 agencies having less than half their need satisfied, one less than in the short–term results. Thus, the max-min policy showed a greater improvement than its counterpart. However, the results obtained here are still of concern. It was hoped that both policies would have performed far better than what the results indicate. The continued poor performance of both policies here can be attributed to the increase in variation in the average % of an agency's target need satisfied (above) as follows. Although the increase in the *average Kcal/person* and *average minimum Kcal/person* values would suggest that fewer agencies would receive less than half of their target need, the increase in variation of the average percentage of an agency's target need satisfied counteracts the effects of these improvements, resulting in a higher number of agencies receiving less than half of their needs than would be anticipated.

9.1.3 Long–term simulation

$n = 365$
$k = 5$

Table 9.4 summarizes the results obtained from simulating the two allocation policies for a period of five years. The results obtained from this simulation should provide us with the most accurate reflection of the performance of each policy, and can be viewed as being close approximations to the performance of the model for longer simulation periods. It is believed that the values obtained for each of the five statistics for this simulation run are 'stable' in the sense that they would not change substantially for longer runs of the model. It will be interesting to ascertain to what degree the performances of each policy have changed

from the results obtained for the medium–term simulation.

Table 9.4: Long–term simulation results

Allocation policy	Max–min	Number–based
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	666.97	640.44
Avg. min. Kcal/person value	359.39	201.22
Avg. % of agency's target nutritional need satisfied	69.71	85.33
Avg. variance in % of agency's target nutritional need satisfied	32.80	34.21
Avg. num. of agencies receiving $< 50\%$ of target % need	18	13

Analysis and discussion of long–term results

What follows is an analysis and discussion of each of the performance measures detailed in Table 9.4.

Average Kcal/person value
Both allocation policies demonstrate a marginal increase in their values for this statistic. The trend of the number–based policy showing a greater increase than the max–min continues, as it increases by approximately 8 Kcal per person on average, compared to the increase of approximately 6 Kcal per person achieved by the max–min policy. That both policies do still exhibit improvements on this performance measure for a considerably longer simulation length (five times the medium term length) is encouraging. The smaller increases do indicate that these values are stabilizing. The performances of each policy are displayed in Figure 9.4. Both policies result in more than 90% of the target value of 700 Kcal per person being achieved on average (max–min policy approximately 95% and number–based policy approximately 91%) which is a very positive result.
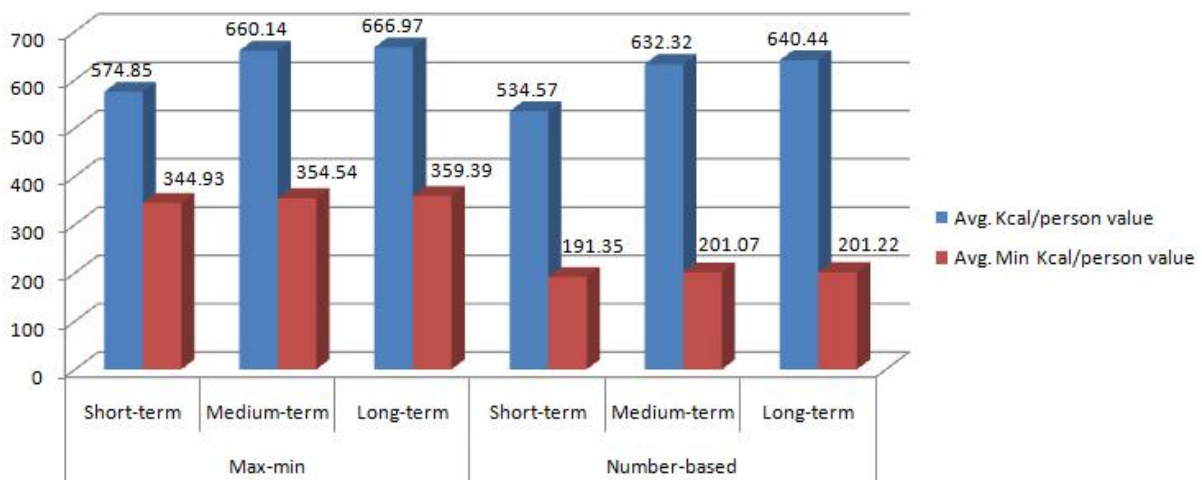
Figure 9.4: Average Kcal/person and minimum Kcal/person values for each allocation policy for the short, medium and long–term simulations

Average minimum Kcal/person value

The max–min policy shows a small increase of approximately 5 Kcal per person, whilst the increase demonstrated by the number–based policy is negligible. The small increases achieved are displayed in Figure 9.4. Thus, it appears that these values are also stabilizing. It is heartening to see that the value for this statistic achieved by the max–min policy is at least half the target value. The max–min policy continues to decisively outperform the number–based policy on this performance measure (as we would expect it to). FBSA would obviously desire to see the performance of both policies with regards to this statistic increase substantially in the near future. By procuring more food it is anticipated that this will be plausible, as indicated by the results of the first degeneracy experiment (see Table A.4 in Appendix A) and the second extreme conditions experiment (see Table A.9 in Appendix A).

Average % of agency's target nutritional need satisfied

Both policies achieve small improvements compared to the medium–term results (each increases by approximately 1 percent). The trend of values stabilizing is evident in these results too, which are displayed in Figure 9.5. The trend of the number–based policy considerably outperforming its counterpart is also evident, with the results indicating that on average it satisfies 15% more of an agency's need than does the max–min policy. This is an important result, as this statistic is the foremost measure of the overall performance of a policy (when considered in conjunction with the *average variance in % of agency's target nutritional need satisfied*).
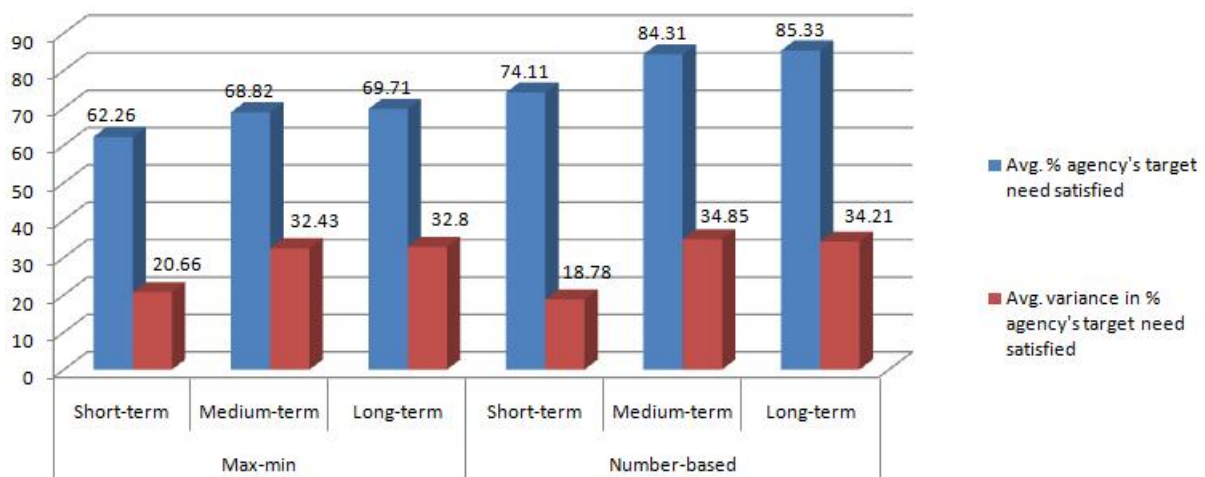
Figure 9.5: Average % of agency's target nutritional need satisfied and the average variance in this % for each allocation policy for the short, medium and long–term simulations

Average variance in % of agency's target nutritional need satisfied

The max–min policy results in a slight increase in variance, whilst the number–based policy actually demonstrates a small decrease (approximately half a percent). It is a relief that the substantial 'jump' in this value that occurred between the short and medium–term results is not repeated here. The trend of stabilizing values is evident (once again), and is visible in Figure 9.5. This variation is still far higher than what would be desired. Once again, it is difficult to decide on what is the primary cause of this variation, and it would be of great value (to FBSA) for further research in this area to be conducted.

Average number of agencies receiving < 50% of target % need

There is no improvement made by the max–min policy for this statistic, whilst the number–based policy improves by one (i.e. one less agency receiving less than half of its target need). That there is minimal

improvement indicates that these values are stabilizing and are still poor. Once again, with an increased inflow of food both policies should improve their performance with regards to this statistic as is indicated by the results of the first degeneracy experiment (see Table A.4 in Appendix A) and the second extreme conditions experiment (see Table A.9 in Appendix A).

9.2 Reflection on results

The discussion of the results of the three simulation experiments presented here has provided us with a good measure of the performance of both the max–min and number–based allocation policies. What is of interest is that one cannot categorically state the one policy is better than the other. Each policy has its advantages and disadvantages when compared to the other:

- The max–min policy consistently outperformed the number–based policy with respect to the *average Kcal/person* and *average minimum Kcal/person* values

- The number–based policy consistently outperformed the max–min policy with respect to the *average % of agency's target nutritional need satisfied* and *average number of agencies receiving < 50% of target % need* values

- Both policies exhibited too high a value for the *average variance in % of agency's target nutritional need satisfied*, with the number–based policy initially performing better than its counterpart. In the long run, the max–min policy demonstrated a slightly lower value than the number–based policy for this performance measure

Thus, no conclusion can be reached as to which is the 'better' policy to implement. One could argue that FBSA should just continue with their current allocation policy (essentially the number–based policy) as it performs best with respect to the *average % of agency's target nutritional need satisfied*, which is the most salient performance indicator. However, since FBSA is intent on improving the accuracy and fairness of their allocations, implementing the max–min policy is worthy of consideration. The decision on what allocation policy/ies to implement in the future is entirely up to FBSA. Both policies can be utilized by FBSA, the choice of which one will depend on which of the three criteria of accuracy, efficiency and fairness FBSA deem to be most important. Should they wish to maintain the efficiency of the allocation system, the number–based policy would be the best option. However, should they wish to improve the accuracy and fairness of the allocation system then the max–min policy should be adopted. It is hoped that the results presented here, along with the DSS FAST (in particular FAST as a DSS for reviewing allocation policies) will assist those members of FBSA in making these decisions.

9.3 Feedback on FAST as a DSS for performing daily allocations

A thorough presentation of FAST to the management and staff of FBCT was made at the end of November 2010. While it was initially agreed that it would be tested during December, unforeseen circumstances at the organization resulted in testing only beginning in mid–January. Hence, there was not enough time for them to investigate FAST properly to provide suitable feedback.

However, it must be noted here that, upon presenting FAST to FBCT, they indicated that they felt it would be of great use to them. It was well understood (i.e. transparent) and they intended to begin testing it on

both a local (CT area) and national level (i.e. at other foodbanks) with almost immediate effect. The FBCT warehouse manager visited the other foodbanks across the country and demonstrated FAST to them. All of the other foodbanks agreed on the value of utilizing the system and expressed their support for testing it. From the initial testing carried out on FAST as a DSS for performing daily allocations[3] thus far, members from FBCT have commented that the system is quick and easy to use, and will definitely be put into operation in the near future.

Whilst the lack of feedback is not ideal, consolation is found in the initial positive response from FBCT. As mentioned earlier, the author will be available to FBCT as a technical adviser for the first quarter of 2011. It is envisioned that FAST will be in operation at FBCT (if not countrywide) by the end of March, after which some concrete feedback could be obtained. However, such feedback would have to form part of another study.

---

[3]Note that no testing of FAST as a DSS for reviewing allocation policies has been conducted so far, and thus no feedback from FBCT/FBSA has been received.

Chapter 10

Conclusion and Recommendations

10.1 Conclusions

Hunger and food insecurity are pandemics that are on the rise. The world does produce enough food on a daily basis to provide sufficient nutrition to every person in it, hence why the problem of hunger is so perplexing. Part of the problem is that a large amount of food is wasted, and foodbanking has developed to help combat hunger by recovering waste food and redistributing it to the hungry as part of its strategy. Foodbanking in SA has exploded over the past two years with the formation of the national NPO FBSA that coordinates a countrywide network of foodbanks. This study is a record of a successful ORD intervention carried out at FBCT, where the decision–making of members of FBSA/FBCT involved in the allocation system has been augmented.

This study is a good example of the application of OR techniques in a developing country/economy in the area of the allocation of inventory. After discussing the issues of hunger and food insecurity, detailing what foodbanking is and summarizing its history in Chapter 2, the problems facing FBSA were discussed and the topic for this study was identified in Chapter 3. A review of the literature pertaining to the allocation of inventory, the use of simulation in the area of allocation and the differences in approach between OR and ORD was provided in Chapter 4. This study has also helped to address an evident 'research–gap' in applying combined 'soft–hard' OR approach through the use of two problem–structuring tools, CM and RDs, and a simulation study of the allocation system at FBCT. Chapter 5 outlined exactly what is meant by problem–structuring, described CM and RDs and detailed their application to FBSA and the allocation system at FBCT respectively. CM helped both the author of this study and members of FBSA gain a greater understanding of FBSA in terms of its goals, and consequently a good appreciation of the context in which decisions in the organization are made. The application of the RDs helped the author to better comprehend the decision–issues within the allocation system at FBCT, by exploring the different worldviews of various stakeholders of the system. The above insight gained from structuring the problem proved to be critical in helping the author to recognize the need for a simulation study of the allocation system at FBCT, as well as guiding the development of the simulation model and the interpretation of the results of the consequent simulation experiments.

The development of the simulation model and the DSS FAST were extensively described in Chapter 6. The simulation model developed was a good representation of the allocation system at FBCT. It was not created to be a carbon–copy of the system or to be used as a prediction tool, but rather to investigate the performances of allocation policies (via a simulation study) to address the accuracy, efficiency and fairness

of FBCT's allocation system. FAST *as a DSS for performing daily allocations* provides FBSA with a tool with which they can manage their database of agencies and perform daily allocations (automating many of the allocation decisions involved in this process), decreasing the time taken to perform these allocations. FAST *as a DSS for reviewing allocation policies* can assist DMs at FBSA/FBCT to review the performances of current and future allocation policies. FAST is transparent and easy to use, and the management at FBCT trusts it enough to test it and intend to implement it on a national level. Overall, the assistance provided by FAST to DMs involved in the allocation system at FBCT/FBSA can help them improve the accuracy, efficiency and fairness of the allocation system as a whole (all of which were important criteria identified by FBSA).

The simulation model developed in Chapter 6 was validated and verified by conducting a series of pertinent tests in Chapter 7. The model invariably behaved as it was expected to, sometimes exceeding these expectations. In involving FBSA/FBCT in the development of the model, its transparency was increased which resulted in it attaining a high level of credibility. A simulation study of the allocation system at FBCT was conducted in Chapter 8 by running the simulation model for three different periods of one month, one year and five years. The results of this simulation study were presented and discussed in Chapter 9. The study helped to investigate the robustness of two allocation policies, the 'Max–min foodvalue' and 'Number–based equal' by computing five statistics to measure their performances for each of the three simulation periods. The results showed that both policies have their advantages and disadvantages when compared to one another, with neither policy being conclusively 'better' than the other. Whilst only these two policies were considered in this study, they can both be utilized by FBSA within their current resource constraints. The results obtained from the simulation study can help the DMs at FBSA/FBCT to decide on which policy to adopt in the future.

From the (limited) feedback obtained from FBSA/FBCT, it is evident that the work conducted in this study will be helpful to them on a practical level, and thus indirectly better the plight of millions of hungry people in SA. On a theoretical level, this half–dissertation should serve as a reference point for future ORD research in similar areas of need in SA and the world. The product of this research is a 'bank' of decision support that can assist FBSA/FBCT and other organizations in their fight against hunger and food insecurity. The decision support provided to FBSA, in the form of the problem–structuring work, the simulation model and results from the simulation study, and the DSS FAST can definitely by utilized to great effect by FBSA in the future to spur them on their journey towards the elimination of hunger, food insecurity and poverty in SA.

10.2 Reflection on study

10.2.1 Limitations to research

As stated in the body of this half–dissertation, limitations in both the scope of, and time available to complete this study resulted in certain aspects of the research being explored only to a preliminary extent. A brief comment on each of these limitations is provided here, followed by a discussion of those areas affected by these limitations.

- *Limitation in scope of study*: Around the 'three–quarter' mark of the period available to conduct this research, FAST had been comprehensively developed, whilst the simulation study was at a preliminary stage. Whilst reviewing the progress of the study with my supervisors, it was agreed that the volume of work already completed was sufficient for the scope of a half–dissertation. It

was thus decided to halt further exploration of the simulation study, and begin writing.

- *Limitation in time available to complete this research*: As with most dissertations, time (the lack thereof) placed limitations on the extent to which some aspects of this study could be explored, most notably the simulation study (as indicated above) and FAST

Effect of limitations on simulation study

As noted, due to the above limitations, the simulation study was halted when still at a preliminary level of exploration. A great deal more work could have been done through the simulation study – most of which is included as topics for future research in section 10.3. In particular, investigating the effect of changing what types of food can be allocated to each category of agency on the variance in the average percentage of agency's target need satisfied was a topic that would have been explored if more time was available. Investigating the effects of both subtle and major changes to the input parameters on the performance of each policy would also have been carried out (although such an investigation was conducted to an extent in the model validation tests). In summary, the simulation study would have been conducted to a fuller extent (as outlined in Figure 6.3).

Effect of limitations on decision support system – FAST

Essentially, the DSS given to FBSA is only a prototype. Undoubtedly, there will be some aspects of FAST that could be improved, as well as adding to its functionality in order to increase its all–round utility. However, there simply was not enough time to make any of these amendments.

10.3 Recommendations for future research

There is large scope for further research to be conducted in the area focussed on in this study. It is hoped that our study will prove to be a useful reference for future research carried out in similar applications of OR in the areas of hunger in developing countries and decision–making in NGO's and NPO's. What follows is a summary of suggestions for future research, some of which are particular to FBSA whilst others could have a wider application.

- Tackle the other issues identified by FBSA in section 3.2, i.e. that of:
  - Determine the optimum (i.e. minimum cost) vehicle routes for delivering and collecting food
  - Develop some form of an activity–based costing of meals (a minimum–cost model where drivers could decide to 'buy' a certain food delivery)
  - Determine the optimum locations for establishment of new foodbanks across SA (accessibility, number of people to be impacted and cost)[1]
  - Develop a comprehensive 'poverty map' of areas in SA
  - Assist with the management and maintenance of their fleet of vehicles (minimum–cost model)

---

[1]Mention must be made of a study in progress by a Stellenbosch student, E.J. Lanz, that is attempting to address this issue.

- Re–fit distributions to the perishable and non–perishable food categories

  – It would be of great interest and value to re–fit these distributions, using the data used in this study (March–June 2010) as well as data since then (i.e. including data from July–December 2010)

  – Re–fitting the distributions to a more comprehensive set of data would result in more confidence in the accuracy of these distributions

  – Such a re–fit would be necessary considering that FBSA is constantly expanding, securing more donors and greater quantities of donated food. Being able to investigate the performance of the two policies considered in this study (as well as others) with updated input distributions would be of great value to FBSA

  – Fitting distributions to each day of the week (i.e. Monday to Sunday) would also increase the accuracy of estimating the incoming donations

- Incorporate the financial donations received by FBSA as an input into the model, with a view to exploring the purchase of extra food in order to increase the overall satisfaction of agencies' needs i.e. investigating a new allocation policy in which food is purchased to supplement the donated food

  – This would require collection of financial data, fitting distribution/s to the data and determining the goodness–of–fit of these distributions (as done for the food donations in this study)

- Incorporate the time taken to perform allocations (i.e. the physical time taking for FBCT staff to start and complete the allocation process) into the model as a performance measure

  – FBSA has expressed their desire for such a measure of performance

- Incorporate a measure of the nutritional quality (in terms of variety) of food allocated to agencies

  – It would be interesting to investigate adding constraints to the model that effect the composition of the food allocated to agencies

    * Bounds could be placed on each food type to ensure, for example, that a certain percentage of food allocated to each agency (based on the nutritional needs of each category of agency) is fruit or vegetables or canned good an so forth

  – Incorporating such constraints would greatly increase the accuracy of allocations made to agencies

- Improve the existing simulation model to include storing and allocating non–perishable goods based on the amount of perishable goods in inventory

  – FBSA has also expressed their desire to move towards such an allocation policy for non–perishable goods

  – It would be of great interest to investigate the effects of such an addition to the model on both the average percentage satisfaction of an agency's need and the variation in that level of satisfaction

- Investigate the use of a two–interval allocation policy (as in McGavin *et al.*'s study [46] in section 4.2)

  – One interval (withdrawal) for perishable stock and another for non–perishable stock

  – FBSA is interested in developing and investigating such an allocation policy in the future

- Investigate the causes of the large variation in the average percentage of agency's target need satisfied, and suggest plausible solutions to this

- Investigate the effects of changing the food classifications to agencies (as detailed in Table A.2 in Appendix A) on the performance of the two policies considered in this study with respect to the five performance measures

    - It would be of particular interest to examine the effect of such a change on the *average variance in % of agency's target nutritional need satisfied*

- Improve FAST by increasing its functionality and user–friendliness

    - Adding new functions, such as being able to filter and/or allocate to groups of agencies based in their location or category

Bibliography

[1] F. Ackermann and C. Eden. *SODA: Journey Making and Mapping in Practice. In: Rosenhead J and Mingers J (eds) Rational Analysis for a Problematic World Revisited*, chapter 3. Wiley, Chichester, 2001.

[2] F. Ackermann and C. Eden. Using Causal Mapping with Group Support Systems to Elicit an Understanding of Failure in Complex Projects: Some Implications for Organizational Research. *Group Decision and Negotiation*, 14(5):355–376, 2005. ISSN 0926-2644.

[3] R. L. Ackoff. The Future of Operational Research is Past. *Journal of the Operational Research Society*, 30(2):93–104, 1979. ISSN 0160-5682.

[4] Foodbank South Africa. Food Banking Explained. `http://www.foodbank.org.za/what-we-do/foodbanking`, Accessed 5 May 2010 .

[5] O. Ahumada and J. R. Villalobos. Application of Planning Models in the Agri-FoodSupply Chain: A Review. *European Journal of Operational Research*, 195:1–20, 2009.

[6] St Mary's FoodBank Alliance. St Mary's Foodbank. `http://www.firstfoodbank.org/history.html`, Accessed 7 May 2010 .

[7] C. B. Alphonce. Application of the Analytic Hierarchy Process in Agriculture in Developing Countries. *Agricultural Systems*, 53:97–112, 1997.

[8] K. R. Apaiah. *Designing Food Supply Chains: A Structured Methodology; a Case on Novel Protein Foods*. PhD thesis, Wageningen University, 2006.

[9] R. M. Axelrod. *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton University Press Princeton, New Jersey, 1976.

[10] T. J. Blake. Aiding Decision Making at Foodbank Cape Town. Master's thesis, University of Cape Town, 2009.

[11] C. T. Bornstein and J. Rosenhead. The Role of Operational Research in Less Developed Countries: A Critical Approach. *Eur. Jnl. Opnl. Res.*, 49(2):156–178, 1990.

[12] P. Chandra. A Dynamic Distribution Model with Warehouse and Customer Replenishment Requirements. *Jnl. Opnl. Res. Soc.*, 44(7):681–692, 1993. ISSN 0160-5682.

[13] P. Checkland. Systems Thinking, Systems Practice. In: Eden, C. and Ackermann, F. Were Next for Problem Structuring Methods. *Jnl. Opnl. Res. Soc.*, 57(7):766–768, 1981. ISSN 0160-5682.

[14] P. Checkland. *Soft Systems Methodology. In: Rosenhead J and Mingers J (eds) Rational Analysis for a Problematic World Revisited*, chapter 4. Wiley, Chichester, 2001.

[15] X. Chen, X. Pan, and C. Yang. On the Study of Chinas Grain Output Prediction. *International Transactions in Operational Research*, 8:429–437, 2001b.

[16] J. de Lijster. Working Partnerships. *Business Report*, page 2, 24 June 2010 .

[17] J. Dewey, L. Hickman, and T. M. Alexander. *The Essential Dewey: Ethics, Logic, Psychology*, volume 2. Indiana University Press, 1998. ISBN 0253211859.

[18] C. Eden. Problem Solving or Problem Finishing? In: Eden, C. and Ackermann, F. Were Next for Problem Structuring Methods. *Jnl. Opnl. Res. Soc.*, 57(7):766–768, 1987. ISSN 0160-5682.

[19] C. Eden and F. Ackermann. *Making Strategy: The Journey of Strategic Management*. Sage Publications Ltd, London, 1998. ISBN 076195225X.

[20] C. Eden and F. Ackermann. Where Next for Problem Structuring Methods. *Jnl. Opnl. Res. Soc.*, 57 (7):766–768, 2006. ISSN 0160-5682.

[21] Wikipedia Online Encyclopaedia. Millennium Development Goals. `http://en.wikipedia.org/wiki/Millennium_Development_Goals`, Accessed 6 May 2010 .

[22] Wikipedia Online Encyclopaedia. HIV/AIDS in South Africa. `http://en.wikipedia.org/wiki/HIV/AIDS_in_South_Africa`, Accessed 13 July 2010 .

[23] R. Epstein, L. Henríquez, J. Catalán, G. Y. Weintraub, and C. Martínez. A Combinational Auction Improves School Meals in Chile. *Interfaces*, 32(6):1–14, 2002. ISSN 0092-2102.

[24] Schlesinger et al. Terminology for model credibility. *Simulation*, 32(3):103–104, 1979.

[25] J. R. Fairweather and L. M. Hunt. Can Farmers Map their Farm System? Causal Mapping and the Sustainability of Sheep/beef Farms in New Zealand. *Agriculture and Human Values*, pages 1–12, 2009. ISSN 0889-048X.

[26] A. Federgruen, G. P. Prastacos, and P. H. Zipkin. An Allocation and Distribution Model for Perishable Products. *Operations Research*, 34(1):75–82, 1986.

[27] J. Fleck, H. Scarbrough, and J. Swan. Instructors Guide. Trainer Notes: Cognitive Mapping Techniques. `http://www.omni.bus.ed.ac.uk/opsman/oakland/inst18.html`, Accessed 18 March 2010 .

[28] M. Folie and J. Tiffin. Solution of a Multi-product Manufacturing and Distribution Problem. *Mngmt. Sci.*, 23(3):286–296, 1976. ISSN 0025-1909.

[29] Food and Agriculture Organization of the United Nations. The State of Food Insecurity in the World 2009. `www.fao.org/docrep/012/i0876e/i0876e00.htm`, Accessed 10 May 2010 .

[30] Food and Agriculture Organization of the United Nations. Energy Requirements and Expenditure. `http://www.fao.org/docrep/U8480E/U8480E04.html`, Accessed 9 September 2010 .

[31] FoodbankCT. Foodbank Cape Town. First Quarterly Newsletter, June 2009 .

[32] FoodbankSA. Agri-FoodBank: A Proposal to Integrate Development Programmes of Multiple Departments. Proposal document, FoodbankSA, 2010.

[33] D. L. Franklin and M. W. Harrell. Resource Allocation Decisions in Low-income Rural Households:: Nutritional Implications for Agricultural and Rural Development Projects. *Food Policy*, 10(2): 100–108, 1985.

[34] B. E. Fries. Optimal Ordering Policy for a Perishable Commodity with Fixed Lifetime. *Opns. Res.*, 23 (1):46–61, 1975.

[35] O. Fujiwara, H. Soewandi, and D. Sedarage. An Optimal Ordering and Issuing Policy for a Two-stage Inventory System for Perishable Products. *Eur. Jnl. Opnl. Res.*, 99:412–424, 1997.

[36] R. D. Galvao. Operational Research for Development. In *Fifth International Conference on Operational Research for Development (ICORD-V)*, pages viii–x, Jamshedpur, India, 19th–21st December, 2005 1992.

[37] R. D. Galvao. The Operational Research for Development Movement Since the ICORD I (1992) Conference. In *Fifth International Conference on Operational Research for Development (ICORD-V)*, pages 1–2, Jamshedpur, India, 19th–21st December, 2005 1992.

[38] C-H. Goh, B. S. Greenberg, and H. Matsuo. Two–Stage Perishable Inventory Models. *Mngmt. Sci.*, 39(5):633–649, 1993.

[39] A. Gopnik, C. Glymour, D. M. Sobel, L. E. Schulz, T. Kushnir, and D. Danks. A Theory of Causal Learning in Children: Causal Maps and Bayes Nets. *Psychological review*, 111(1):3–31, 2004. ISSN 0033-295X.

[40] M. Haouari and M. N. Azaiez. Optimal Cropping Patterns Under Water Deficits. *Eur. Jnl. Opnl. Res.*, 130:133–146, 2001.

[41] R. M. Hill. Allocating Warehouse Stock in a Retail Chain. *Jnl. Opnl. Res. Soc.*, 40(11):983–992, 1989. ISSN 0160-5682.

[42] D. W. Hine, C. J. Montiel, R. W. Cooksey, and J. H. Lewko. Mental Models of Poverty in Developing Nations. *Journal of Cross-Cultural Psychology*, 36(3):283, 2005. ISSN 0022-0221.

[43] Stewart T. J. OR Practice in South Africa. *Eur. Jnl. Opnl. Res.*, 87:464–468, 1995.

[44] L. Markíczy and J. Goldberg. A Method for Eliciting and Comparing Causal Maps. *Journal of Management*, 21(2):305, 1995. ISSN 0149-2063.

[45] E. Martin IV. Centralized Bakery Reduces Distribution Costs Using Simulation. *Interfaces*, 28(4): 38–46, 1998. ISSN 0092-2102.

[46] E. J. McGavin, L. B. Schwarz, and J. E. Ward. Two-interval Inventory-allocation Policies in a One-warehouse N-identical-retailer Distribution System. *Mngmt. Sci.*, 39(9):1092–1107, 1993. ISSN 0025-1909.

[47] J. Mellor-Crummey. Analyzing Simulation Results. Lecture notes, Rice University, 2005.

[48] J. Mingers and J. Rosenhead, editors. *Rational Analysis for a Problematic World Revisited*. Wiley, Chichester, 2001.

[49] J. Mingers and J. Rosenhead. Problem Structuring Methods in Action. In: White, L. Evaluating Problem-structuring Methods: Developing an Approach to Show the Value and Effectiveness of PSMs. *Jnl. Opnl. Res. Soc.*, 57(7):842–855, 2006. ISSN 0160-5682.

[50] G. Montibeller and V. Belton. Causal Maps and the Evaluation of Decision Optionsa Review. *Jnl. Opnl. Res. Soc.*, 57(7):779–791, 2006. ISSN 0160-5682.

[51] J. A. Muckstadt and R. O. Roundy. Multi-item, One-warehouse, Multi-retailer Distribution Systems. *Mngmt. Sci.*, 33(12):1613–1621, 1987. ISSN 0025-1909.

[52] S. Nahmias. Allocation of a Perishable Product Inventory. *Opns. Res.*, 23(4):735–749, 1975.

[53] S. Nahmias. Myopic Approximations for the Perishable Inventory Problem. *Mngmt. Sci.*, 22(9): 1002–1008, 1976.

[54] V. K. Narayanan and D. J. Armstrong. *Causal Mapping for Research in Information Technology*. Idea Group Publishing, London, 2005. ISBN 1591403979.

[55] India Foodbanking Network. Launch of India Foodbanking Network, IFBN. `http://www.indiafoodbanking.org`, Accessed 13 July 2010 .

[56] The Global Foodbanking Network. About Food Banking. `http://www.foodbanking.org/about/foodbanking.html`, Accessed 7 May 2010 .

[57] The Global Foodbanking Network. Winter 2007/2008 News. `http://www.foodbanking.org/news/documents/Winter.html`, Accessed 3 August 2010 .

[58] The Global Foodbanking Network. The Global Foodbanking Network 2009 Annual Report. `http://www.foodbanking.org/news/documents/GFNAnnualReport2009.pdf`, Accessed 8 May 2010 .

[59] South Africa The Good News. Foodbank South Africa. `http://www.sagoodnews.co.za/corporate_social_investment/foodbank_south_africa.html`, Accessed 19 December 2010 .

[60] California Association of Foodbanks. About Food Banking. `http://www.cafoodbanks.org/About_Food_Banking.html`, Accessed 8 May 2010 .

[61] C. Parkan. Simulation of a Fast-food Operation Where Dissatisfied Customers Renege. *Jnl. Opnl. Res. Soc.*, 38(2):137–148, 1987. ISSN 0160-5682.

[62] M. Pidd. Simulating Automated Food Plants. *Jnl. Opnl. Res. Soc.*, 38(8):683–692, 1987. ISSN 0160-5682.

[63] W. P. Pierskalla and C. D. Roach. Optimal Issuing Policies for Perishable Inventory. *Management Science*, 18(11):603–614, 1972.

[64] G. P. Prastacos. Optimal Myopic Allocation of a Product with Fixed Lifetime. *Jnl. Opnl. Res. Soc.*, 29 (9):905–913, 1978.

[65] G. P. Prastacos. Allocation of a Perishable Product Inventory. *Opns. Res.*, 29(1):95–107, 1981.

[66] J. Rosenhead. Past, Present and Future of Problem Structuring Methods. *Jnl. Opnl. Res. Soc.*, 57(7): 759–765, 2006. ISSN 0160-5682.

[67] J. Rosenhead and J. Mingers. Rational Analysis in a Problematic World Revisited. In: White, L. Evaluating Problem-structuring Methods: Developing an Approach to Show the Value and Effectiveness of PSMs. *Jnl. Opnl. Res. Soc.*, 57(7):842–855, 2006. ISSN 0160-5682.

[68] R. G. Sargeant. Verification and Validation of Simulation Models. In *Proceedings of the 2007 Winter Simulation Conference*, pages 124–137, Syracuse, New York, 2007. Syracuse University.

[69] A. Scavarda, T. Bouzdin-Chameeva, S. Goldstein, J. Hays, and A. Hill. A Review of the Causal Mapping Practice and Research Literature. In *Second World Conference on Production and Management (POM) and 15th Annual POM Conference*, Cancun, Mexico, May, 2004 .

[70] A. J. Scavarda, T. Bouzdine-Chameeva, S. M. Goldstein, J. M. Hays, and A. V. Hill. A Methodology for Constructing Collective Causal Maps. *Decision Sciences*, 37(2):263–283, 2006. ISSN 1540-5915.

[71] C. P. Schmidt and S. Nahmias. (S-1, S) Policies for Perishable Inventory. *Mngmt. Sci.*, 31(6): 719–728, 1985. ISSN 0025-1909.

[72] C. Schweigman and T. A. B. Bakker. Operations Research as a Tool for Analysis of Food Security Problems. *Eur. Jnl. Opnl. Res.*, 49(2):211–221, 1990. ISSN 0377-2217.

[73] World Hunger Education Service. World Hunger Facts 2010. `http://www.worldhunger.org/articles/Learn/world`, Accessed 5 May 2010 .

[74] R. E. Shannon. Introduction to the Art and Science of Simulation. In *1998 Winter Simulation Conference*, page 2, Texas, U.S.A, 1998. Texas A&M University.

[75] E. A. Silver, D. F. Pyke, R. Peterson, et al. *Inventory Management and Production Planning and Scheduling*, volume 2. Wiley New York, 1998.

[76] T. J. Stewart. Operational Research for Development in Africa. Technical report, Department of Statistical Sciences, University of Cape Town, 2005.

[77] J. Straatsma, G. Van Houwelingen, A. E. Steenbergen, and P. De Jong. Spray Drying of Food Products: 1. Simulation Model. *Journal of Food Engineering*, 42(2):67–72, 1999. ISSN 0260-8774.

[78] W. Swart and L. Donno. Simulation Modelling Improves Operations, Planning, and Productivity of Fast Food Restaurants. *Interfaces*, 11(6):35–47, 1981. ISSN 0092-2102.

[79] A. J. Taylor. Some Experiences in Planning a Distribution System. *Jnl. Opnl. Res. Soc.*, 33(10): 891–898, 1982. ISSN 0160-5682.

[80] J. G. A. J. Van der Vorst. Performance Measurement in Agri-food Supply-chain Networks. *Quantifying the Agri-food Supply Chain*, pages 15–26, 2006.

[81] J. G. A. J. van der Vorst, A. J. M. Beulens, and P. van Beek. Modelling and Simulating Multi-Echelon Food Systems. *Eur. Jnl. Opnl. Res.*, 122:354–366, 2000.

[82] L. N. van Wassenhove. Humanitarian Aid Logistics: Supply Chain Management in High Gear. In *Fifth International Conference on Operational Research for Development (ICORD-V)*, pages 3–20, Jamshedpur, India, 19th–21st December 2005.

[83] R. V. V. Vidal. Book Reviews: Visible Thinking: Unlocking Causal Mapping for Practical Business Results. *Eur. J. Opns. Res.*, 170:1017–1024, 2006.

[84] N. M. Watson. What Distinguishes Operational Research for Development from Operational Research as a More General Approach to Management Problems. Essay, UCT, 2009.

[85] G. E. Weinreb, T. C. Elston, and K. Jacobson. Causal Mapping as a Tool to Mechanistically Interpret Phenomena in Cell Motility: Application to Cortical Oscillations in Spreading Cells. *Cell motility and the Cytoskeleton*, 63(9):523–532, 2006. ISSN 1097-0169.

[86] L. White. Evaluating Problem-structuring Methods: Developing an Approach to Show the Value and Effectiveness of PSMs. *Jnl. Opnl. Res. Soc.*, 57(7):842–855, 2006. ISSN 0160-5682.

[87] L. White, H. Smith, and C. Currie. OR in Developing Countries: A Review. `doi:10.1016/j.ejor.2010.02.015`, 2010.

Appendix A

Food classifications and Results tables

Classification of food inflows

The data of food inflows were sorted into perishable or non–perishable food according to the following categories in Table A.1 below. Note that the column of examples does not represent all types of foods placed in that category.

Table A.1: Categorization of food

Food type	Category	Examples	Kcal/Kg
Bakery	Perishable	Bread, rolls, cakes, pastries	2300
Cooked foods	Perishable	Prepared meals, cooked meat	1500
Dairy	Perishable	Milk, cheese, butter, yoghurt	2000
Fruits	Perishable	Apples, bananas, oranges, pears	600
Meats	Perishable	Chicken, fish, lamb, steak	2500
Vegetables	Perishable	Lettuce, beans, carrots, potatoes	400
Boxed goods	Non-perishable	Cereals, pastas, dry goods	1800
Canned goods	Non-perishable	Canned fruit, canned meat	1500
Condiments	Non-perishable	Sauces, spices, sugar	2500
Luxury foods	Non-perishable	Biscuits, chips, sweets, juices	5000
Luxury goods	Non-perishable	Toiletries, pharmaceuticals	50
Other	Non-perishable	Toys, furniture, paper	50

Note that, although both 'Luxury goods' and 'Other' goods strictly do not have any nutritional value, they do still have some utility in terms of being useful to the agencies they are given too. Thus they are still assigned (small) nutritional values of 50 Kcal, which is significantly smaller than any of the other food categories, yet indicates that they do have utility.

Table A.2: Categorization of allocation of food to agencies by agency category

Agency Category Product	Clinic	Educare	Feeding Scheme	Nutritional Centre	Satellite	School	Shelter	Soup Kitchen	Support Group
Bakery	1	1	1	1	1	1	1	1	1
Boxed goods	0	1	1	0	1	0	1	1	1
Canned goods	0	1	1	0	1	0	1	1	0
Condiments	0	1	1	0	1	0	1	1	1
Cooked foods	0	0	1	0	1	0	1	1	1
Dairy	0	1	1	0	1	0	1	0	0
Fruits	0	1	1	0	1	1	1	1	1
Luxury foods	0	1	1	0	1	1	1	0	1
Luxury goods	0	0	1	0	1	0	1	0	1
Meats	0	0	1	0	1	0	1	1	1
Vegetables	0	1	1	1	1	0	1	1	1
Other	1	1	1	1	1	1	1	1	1

A '1' indicates that particular type of food is allocated to that category of agency, whilst a '0' indicates that particular type of food is not allocated to that category of agency.

Model validation tests results tables

Degeneracy test results tables

Table A.3: Original results

Allocation policy	*Max–min foodvalue*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	574.85	534.57
Avg. min. Kcal/person value	344.93	191.35
Avg. % of agency's target nutritional need satisfied	62.26	74.11
Avg. variance in % of agency's target nutritional need satisfied	20.66	18.78
Avg. num. of agencies receiving < 50% of target % need	22	15

Table A.4: Experiment 1 results

Allocation policy	*Max–min foodvalue*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	961.11	1034.21
Avg. min. Kcal/person value	832.39	497.00
Avg. % of agency's target nutritional need satisfied	126.06	142.24
Avg. variance in % of agency's target nutritional need satisfied	72.15	77.62
Avg. num. of agencies receiving < 50% of target % need	3	2

Table A.5: Experiment 2 results

Allocation policy	*Max–min foodvalue*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	298.15	315.57
Avg. min. Kcal/person value	180.65	100.02
Avg. % of agency's target nutritional need satisfied	33.17	41.37
Avg. variance in % of agency's target nutritional need satisfied	6.32	7.68
Avg. num. of agencies receiving < 50% of target % need	35	31

Table A.6: Experiment 3 results

Allocation policy	*Max–min foodvalue*
Number of agencies	42
Total number of beneficiaries	10002
Avg. Kcal/person value	413.76
Avg. min. Kcal/person value	413.76
Avg. % of agency's target nutritional need satisfied	59.84
Avg. variance in % of agency's target nutritional need satisfied	9.68
Avg. num. of agencies receiving $< 50\%$ of target % need	22

Table A.7: Experiment 4 results

Allocation policy	*Max–min foodvalue*
Number of agencies	42
Total number of beneficiaries	10002
Avg. Kcal/person value	13.14
Avg. min. Kcal/person value	4.35
Avg. % of agency's target nutritional need satisfied	1.80
Avg. variance in % of agency's target nutritional need satisfied	1.02
Avg. num. of agencies receiving $< 50\%$ of target % need	41
Number of unsolvable iterations	30

Extreme conditions test results tables

Table A.8: Experiment 1 results

Allocation policy	*Max–min foodvalue*	*Number–based*
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	0	0
Avg. min. Kcal/person value	0	0
Avg. % of agency's target nutritional need satisfied	0	0
Avg. variance in % of agency's target nutritional need satisfied	0	0
Avg. num. of agencies receiving $< 50\%$ of target % need	42	42

Table A.9: Experiment 2 results

Allocation policy	Max–min foodvalue	Number–based
Number of agencies	42	42
Total number of beneficiaries	10002	10002
Avg. Kcal/person value	6253.34	6193.30
Avg. min. Kcal/person value	3581.65	1999.73
Avg. % of agency's target nutritional need satisfied	671.24	810.03
Avg. variance in % of agency's target nutritional need satisfied	3281.59	3859.64
Avg. num. of agencies receiving $< 50\%$ of target % need	0	0

Appendix B

VBA code

The Excel VBA code written to simulate the daily allocation system at FBCT is included here. Comments explaining the logic and reasoning behind the various actions within the program are indicated by a ' symbol before the entered text. It is hoped that these comments promote a 'smoother' reading of the code and enhance its transparency.

The VBA code for FAST is included in the CD (Appendix C)[1] provided with the hard copy of this dissertation. A picture of each stage or 'userform' of the DSS is presented, followed by all the code associated with that userform. Comments are included in the code as above.

Should the reader have any queries, please feel free to contact the author:

Neil Mark Watson

Email: neilmarkwatson@gmail.com

---

[1]This code totals to approximately 100 pages. Thus it was decided to include it in CD format.

```vb
Sub Allocation()


    Application.ScreenUpdating = False


    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'This will simulate both the "MAX-MIN FOODVALUE ALLOCATION" and _
    "NUMBER-BASED ALLOCATION for n days, and k iterations
    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Worksheets("Home").Activate

    Range("E24").Value = Time()

    n = Range("B5").Value

    k = Range("B7").Value

    Range("H25:GAA1000").ClearContents
    Range("H25:GAA1000").ClearFormats


    'Dimension the various arrays to be simulated

    'Perishable donations:

    ReDim PerDon(n, k) As Single


    'Non-perishable donations:

    ReDim NonPer(n, k) As Single


    'Bernoulli random variables:

    ReDim Bernoulli(n) As Single


    'Perishable goods:

    ReDim Bakery(n, k) As Single
    ReDim Cook(n, k) As Single
    ReDim Dairy(n, k) As Single
    ReDim Fruit(n, k) As Single
    ReDim Meat(n, k) As Single
    ReDim Veg(n, k) As Single


    'Non-perishable goods:

    ReDim Box(n, k) As Single
    ReDim Can(n, k) As Single
    ReDim Con(n, k) As Single
    ReDim Luxf(n, k) As Single
    ReDim Luxg(n, k) As Single
    ReDim Other(n, k) As Single
```

```
        'Compute (and display) the various arrays

        For j = 1 To k

            For i = 1 To n

                'Perishable donations:

                PerDon(i, j) = LognormalInv(Rnd(), 0.694734283, 0.702423795)

                Range("D104").Offset(i - 1, j - 1).Value = PerDon(i, j)


                'Non-perishable donations:

                Bernoulli(i) = BernoulliRand(0.6)

                NonPer(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 2.9094) * Bernoulli(i)

                Range("D504").Offset(i - 1, j - 1).Value = NonPer(i, j)



                'Perishable donation catgories:

                Bakery(i, j) = 1000 * LognormalInv(Rnd(), 0.694734283, -0.339713647)
                Range("IV104").Offset(i - 1, j - 1).Value = Bakery(i, j)

                Cook(i, j) = 1000 * LognormalInv(Rnd(), 0.694734283, -3.19964888)
                Range("SQ104").Offset(i - 1, j - 1).Value = Cook(i, j)

                Dairy(i, j) = 1000 * LognormalInv(Rnd(), 0.694734283, -1.909407548)
                Range("ACF104").Offset(i - 1, j - 1).Value = Dairy(i, j)

                Fruit(i, j) = 1000 * LognormalInv(Rnd(), 0.694734283, -1.053775499)
                Range("ALX104").Offset(i - 1, j - 1).Value = Fruit(i, j)

                Meat(i, j) = 1000 * LognormalInv(Rnd(), 0.694734283, -1.781683175)
                Range("AVP104").Offset(i - 1, j - 1).Value = Meat(i, j)

                Veg(i, j) = 1000 * LognormalInv(Rnd(), 0.694734283, -0.509245216)
                Range("BFH104").Offset(i - 1, j - 1).Value = Veg(i, j)



                'Non-perishable donation categories:

                Box(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 0.50565372) * Bernoulli(i)
                Range("IV504").Offset(i - 1, j - 1).Value = Box(i, j)

                Can(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 0.65577876) * Bernoulli(i)
                Range("SQ504").Offset(i - 1, j - 1).Value = Can(i, j)

                Con(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 0.62610288) * Bernoulli(i)
                Range("ACF504").Offset(i - 1, j - 1).Value = Con(i, j)

                Luxf(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 0.71047548) * Bernoulli(i)
                Range("ALX504").Offset(i - 1, j - 1).Value = Luxf(i, j)

                Luxg(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 0.25282686) * Bernoulli(i)
                Range("AVP504").Offset(i - 1, j - 1).Value = Luxg(i, j)

                Other(i, j) = 1000 * GammaInv(Rnd(), 0.49405, 0.1585623) * Bernoulli(i)
                Range("BFH504").Offset(i - 1, j - 1).Value = Other(i, j)



            Next i

        Next j
```

```vba
    'Define the different categories of agency as an array

    Dim AgencyCat(1 To 9) As String

    AgencyCat(1) = "Clinic"
    AgencyCat(2) = "Educare"
    AgencyCat(3) = "NutritionalCentre"
    AgencyCat(4) = "School"
    AgencyCat(5) = "Shelter"
    AgencyCat(6) = "SoupKitchen"
    AgencyCat(7) = "SupportGroup"
    AgencyCat(8) = "Satellite"
    AgencyCat(9) = "FeedingScheme"


    'Search the data for the first of each type of category, and calculate the _
    total # ppl being supported by agencies in that category


    'Define a vector of the total # ppl being supported by each agency:

    Dim totalppl(5000) As Integer

    'Define a vector of agency category totals

    Dim cattotal(9) As Integer


    For q = 1 To 9

        Range("E29").Activate

        Set agencat = Range("E30", Range("E30").End(xlDown)).Find(AgencyCat(q))

        If Not agencat Is Nothing Then

            agencat.Select

            firstcat = ActiveCell.Address


            totalppl(1) = ActiveCell.Offset(0, 1).Value



            'Search the data for all other categories of type q:

            For R = 2 To 5000

                Range("E30", Range("E30").End(xlDown)).FindNext(after:=ActiveCell).Select

                selectedcat = ActiveCell.Address



                'Insert an If Then statement here to tell VBA that once it has reached _
                the firstcategory again, then the number of agencies of that category is _
                equal to the number of categories (r) it found less 1 (r-1)(since it has _
                found the firstcategory twice)


                If selectedcat = firstcat Then


                    'Label the number of categories as 'rcats':

                    rcats = R - 1



                    'A GoTo statement is inserted here so that once VBA has run through the _
                    rest of the database and found all other agencies of category q, it must _
                    total the number of ppl being supported by that category of agency

                    GoTo catcontinue

                End If
```

```vba
                totalppl(R) = ActiveCell.Offset(0, 1).Value

        Next R

catcontinue:

        'Sum the total number of ppl per agency category:

        cattotalppl = 0

        For R = 1 To rcats

            cattotalppl = cattotalppl + totalppl(R)

        Next R

        'output the total number of ppl per category onto the spreadsheet:

        cattotal(q) = cattotalppl

        Range("AA2000").Offset(q - 1).Value = cattotal(q)

    Else

        cattotal(q) = 0

        Range("AA2000").Offset(q - 1).Value = cattotal(q)

    End If


Next q


'Calculate the total number of ppl being supported across all categories of agency:

Dim cattotppl(9) As Integer

For q = 1 To 9

    cattotppl(q) = Range("AA2000").Offset(q - 1).Value

Next q


totppl = 0

For q = 1 To 9

    totppl = totppl + cattotppl(q)

Next q


'Calculate the total number of ppl to be given each type of food

'Bakery
bktotppl = totppl

'Boxed goods
bxtotppl = cattotppl(2) + cattotppl(5) + cattotppl(6) + cattotppl(7) + cattotppl(8) + cattotppl(9)

'Canned goods
cantotppl = cattotppl(2) + cattotppl(5) + cattotppl(6) + cattotppl(8) + cattotppl(9)

'Condiments
contotppl = cattotppl(2) + cattotppl(5) + cattotppl(6) + cattotppl(7) + cattotppl(8) + cattotppl(9)

'Cooked food
cktotppl = cattotppl(5) + cattotppl(6) + cattotppl(7) + cattotppl(8) + cattotppl(9)
```

```vba
        'Dairy
        drtotppl = cattotppl(2) + cattotppl(5) + cattotppl(8) + cattotppl(9)


        'Fruit goods
        frtotppl = totppl - cattotppl(1) - cattotppl(3)


        'Luxury foods
        lftotppl = cattotppl(2) + cattotppl(4) + cattotppl(5) + cattotppl(7) + cattotppl(8) + cattotppl(9)


        'Luxury goods
        lgtotppl = cattotppl(5) + cattotppl(7) + cattotppl(8)


        'Meats
        mttotppl = cattotppl(5) + cattotppl(6) + cattotppl(7) + cattotppl(8) + cattotppl(9)


        'Vegetable goods
        vgtotppl = totppl - cattotppl(1) - cattotppl(4)


        'Other goods
        othtotppl = totppl



        'Dimension a variable for the category of each agency, 'agencycategory' as a string variable


        Dim agencycategory As String


        'Read in the number of agencies in the data (s):

        s = Range("D30", Range("D30").End(xlDown)).Rows.Count



        'Dimension vectors for the agencies, the number of ppl being supported by each agency _
        and the allocation factor for each type of product that will be allocated to the agencies

        ReDim Agency(s) As String                'Vector for agencies

        ReDim agencynum(s) As Integer            'Vector for number of people being supported by an agency


        'Insert a For Next loop to run through data of agencies, reading in their name, number of ppl _
        being supported and category:

        For p = 1 To s

            Agency(p) = Range("D30").Offset(p - 1).Value

            agencynum(p) = Range("D30").Offset(p - 1, 2).Value


        Next p
```

```
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'MAX-MIN ALLOCATION
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Allocation bound factors:

lb = 2
ub = 2



'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'ADJUSTABLE CELLS
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Insert a For Next loop to run through the agencies and make corresponding _
"perishable" cells adjustable:

For p = 1 To s

    Range("D30", Range("D30").End(xlDown)).Find(Agency(p)).Select

    ActiveCell.Offset(0, 26).Select

    For i = 1 To 6

        ActiveCell.Offset(0, i - (i - 1)).Select

        With ActiveCell
            .Value = "0"
            wbAdjust
        End With


    Next i

Next p


'Insert a For Next loop to run through the agencies and make corresponding _
"non-perishable" cells adjustable:

For p = 1 To s

    Range("D30", Range("D30").End(xlDown)).Find(Agency(p)).Select

    ActiveCell.Offset(0, 33).Select

    For i = 1 To 6

        ActiveCell.Offset(0, i - (i - 1)).Select

        With ActiveCell
            .Value = "0"
            wbAdjust
        End With

    Next i

Next p


'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


For p = 1 To s

    Range("D30", Range("D30").End(xlDown)).Find(Agency(p)).Select

    agencycategory = Range("D30").Offset(p - 1, 1).Value
```

```
        Select Case agencycategory

        Case "Clinic"
            ActiveCell.Offset(0, 41).Value = "1"
            ActiveCell.Offset(0, 42).Value = "0"
            ActiveCell.Offset(0, 43).Value = "0"
            ActiveCell.Offset(0, 44).Value = "0"
            ActiveCell.Offset(0, 45).Value = "0"
            ActiveCell.Offset(0, 46).Value = "0"
            ActiveCell.Offset(0, 48).Value = "0"
            ActiveCell.Offset(0, 49).Value = "0"
            ActiveCell.Offset(0, 50).Value = "0"
            ActiveCell.Offset(0, 51).Value = "0"
            ActiveCell.Offset(0, 52).Value = "0"
            ActiveCell.Offset(0, 53).Value = "1"

            ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"

            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            'CONSTRAINTS
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

            Call ClinicCon

            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Case "Educare"
            ActiveCell.Offset(0, 41).Value = "1"
            ActiveCell.Offset(0, 42).Value = "0"
            ActiveCell.Offset(0, 43).Value = "1"
            ActiveCell.Offset(0, 44).Value = "1"
            ActiveCell.Offset(0, 45).Value = "0"
            ActiveCell.Offset(0, 46).Value = "1"
            ActiveCell.Offset(0, 48).Value = "1"
            ActiveCell.Offset(0, 49).Value = "1"
            ActiveCell.Offset(0, 50).Value = "1"
            ActiveCell.Offset(0, 51).Value = "1"
            ActiveCell.Offset(0, 52).Value = "0"
            ActiveCell.Offset(0, 53).Value = "1"

            ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"

            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            'CONSTRAINTS
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

            Call EducareCon

            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Case "FeedingScheme"
            ActiveCell.Offset(0, 41).Value = "1"
```

```
        ActiveCell.Offset(0, 42).Value = "1"
        ActiveCell.Offset(0, 43).Value = "1"
        ActiveCell.Offset(0, 44).Value = "1"
        ActiveCell.Offset(0, 45).Value = "1"
        ActiveCell.Offset(0, 46).Value = "1"
        ActiveCell.Offset(0, 48).Value = "1"
        ActiveCell.Offset(0, 49).Value = "1"
        ActiveCell.Offset(0, 50).Value = "1"
        ActiveCell.Offset(0, 51).Value = "1"
        ActiveCell.Offset(0, 52).Value = "0"
        ActiveCell.Offset(0, 53).Value = "1"

        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'CONSTRAINTS
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Call FeedingSchemeCon

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Case "NutritionalCentre"
        ActiveCell.Offset(0, 41).Value = "1"
        ActiveCell.Offset(0, 42).Value = "0"
        ActiveCell.Offset(0, 43).Value = "0"
        ActiveCell.Offset(0, 44).Value = "0"
        ActiveCell.Offset(0, 45).Value = "0"
        ActiveCell.Offset(0, 46).Value = "1"
        ActiveCell.Offset(0, 48).Value = "0"
        ActiveCell.Offset(0, 49).Value = "0"
        ActiveCell.Offset(0, 50).Value = "0"
        ActiveCell.Offset(0, 51).Value = "0"
        ActiveCell.Offset(0, 52).Value = "0"
        ActiveCell.Offset(0, 53).Value = "1"

        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'CONSTRAINTS
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Call NutritionalCentreCon

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Case "Satellite"
        ActiveCell.Offset(0, 41).Value = "1"
        ActiveCell.Offset(0, 42).Value = "1"
        ActiveCell.Offset(0, 43).Value = "1"
        ActiveCell.Offset(0, 44).Value = "1"
        ActiveCell.Offset(0, 45).Value = "1"
```

```
        ActiveCell.Offset(0, 46).Value = "1"
        ActiveCell.Offset(0, 48).Value = "1"
        ActiveCell.Offset(0, 49).Value = "1"
        ActiveCell.Offset(0, 50).Value = "1"
        ActiveCell.Offset(0, 51).Value = "1"
        ActiveCell.Offset(0, 52).Value = "1"
        ActiveCell.Offset(0, 53).Value = "1"


        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"


        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'CONSTRAINTS
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Call ShelterCon

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


    Case "School"
        ActiveCell.Offset(0, 41).Value = "1"
        ActiveCell.Offset(0, 42).Value = "0"
        ActiveCell.Offset(0, 43).Value = "0"
        ActiveCell.Offset(0, 44).Value = "1"
        ActiveCell.Offset(0, 45).Value = "0"
        ActiveCell.Offset(0, 46).Value = "0"
        ActiveCell.Offset(0, 48).Value = "0"
        ActiveCell.Offset(0, 49).Value = "0"
        ActiveCell.Offset(0, 50).Value = "0"
        ActiveCell.Offset(0, 51).Value = "1"
        ActiveCell.Offset(0, 52).Value = "0"
        ActiveCell.Offset(0, 53).Value = "1"

        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'CONSTRAINTS
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Call SchoolCon

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


    Case "Shelter"
        ActiveCell.Offset(0, 41).Value = "1"
        ActiveCell.Offset(0, 42).Value = "1"
        ActiveCell.Offset(0, 43).Value = "1"
        ActiveCell.Offset(0, 44).Value = "1"
        ActiveCell.Offset(0, 45).Value = "1"
        ActiveCell.Offset(0, 46).Value = "1"
        ActiveCell.Offset(0, 48).Value = "1"
        ActiveCell.Offset(0, 49).Value = "1"
        ActiveCell.Offset(0, 50).Value = "1"
        ActiveCell.Offset(0, 51).Value = "1"
        ActiveCell.Offset(0, 52).Value = "1"
```

```
        ActiveCell.Offset(0, 53).Value = "1"

        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"


        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'CONSTRAINTS
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Call ShelterCon

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Case "SoupKitchen"
        ActiveCell.Offset(0, 41).Value = "1"
        ActiveCell.Offset(0, 42).Value = "1"
        ActiveCell.Offset(0, 43).Value = "0"
        ActiveCell.Offset(0, 44).Value = "1"
        ActiveCell.Offset(0, 45).Value = "1"
        ActiveCell.Offset(0, 46).Value = "1"
        ActiveCell.Offset(0, 48).Value = "1"
        ActiveCell.Offset(0, 49).Value = "1"
        ActiveCell.Offset(0, 50).Value = "1"
        ActiveCell.Offset(0, 51).Value = "0"
        ActiveCell.Offset(0, 52).Value = "0"
        ActiveCell.Offset(0, 53).Value = "1"

        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"



        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'CONSTRAINTS
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        Call SoupKitchenCon

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Case "SupportGroup"
        ActiveCell.Offset(0, 41).Value = "1"
        ActiveCell.Offset(0, 42).Value = "1"
        ActiveCell.Offset(0, 43).Value = "0"
        ActiveCell.Offset(0, 44).Value = "1"
        ActiveCell.Offset(0, 45).Value = "1"
        ActiveCell.Offset(0, 46).Value = "1"
        ActiveCell.Offset(0, 48).Value = "1"
        ActiveCell.Offset(0, 49).Value = "0"
        ActiveCell.Offset(0, 50).Value = "1"
        ActiveCell.Offset(0, 51).Value = "1"
        ActiveCell.Offset(0, 52).Value = "1"
        ActiveCell.Offset(0, 53).Value = "1"

        ActiveCell.Offset(0, 56).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 57).FormulaR1C1 = "=RC[-29]*RC[-15]"
        ActiveCell.Offset(0, 58).FormulaR1C1 = "=RC[-29]*RC[-15]"
```

```vb
            ActiveCell.Offset(0, 59).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 60).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 61).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 63).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 64).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 65).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 66).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 67).FormulaR1C1 = "=RC[-29]*RC[-15]"
            ActiveCell.Offset(0, 68).FormulaR1C1 = "=RC[-29]*RC[-15]"


            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            'CONSTRAINTS
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            Call SupportGroupCon

            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        End Select

    Next p


    Range("D29").Offset(0, 70).Select
    With ActiveCell
        .Formula = "Vj"
        .Font.Bold = True
        .Font.Underline = True
    End With

    Range("D30").Offset(0, 72).Select
    With ActiveCell
        .Formula = "Kcal/Kg"
        .Font.Bold = True
        .Font.Underline = True
        .Offset(0, 1).Value = "2300"
        .Offset(0, 2).Value = "1500"
        .Offset(0, 3).Value = "2000"
        .Offset(0, 4).Value = "600"
        .Offset(0, 5).Value = "2500"
        .Offset(0, 6).Value = "400"
        .Offset(0, 8).Value = "1800"
        .Offset(0, 9).Value = "1500"
        .Offset(0, 10).Value = "2500"
        .Offset(0, 11).Value = "5000"
        .Offset(0, 12).Value = "50"
        .Offset(0, 13).Value = "50"
    End With

    For p = 1 To s

        Range("D30").Offset(p - 1, 70).FormulaArray = "=(SUMPRODUCT(RC[-14]:RC[-9],R30C77:R30C82) + SUMPRO
DUCT(RC[-7]:RC[-2],R30C84:R30C89))/RC[-68]"

    Next p

    For p = 1 To s

        Range("D30").Offset(p - 1, 71).Select

        wbConstraint Range("BV30").End(xlDown).Offset(2, 0), "<=", ActiveCell.Offset(0, -1), ActiveCell

    Next p


    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'THIS SECTION TOTALS ALL THE AMOUNTS OF ALLOCATED FOOD (ADJUSTABLE CELLS)
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Range("BX34").Select
    With ActiveCell
        .Formula = "Totals"
        .Font.Bold = True
        .Font.Underline = True
    End With

    Range("BY34").Offset(0, 0) = "=sum($BH$30:" & Range("BH30").End(xlDown).Address & ")"
    Range("BY34").Offset(0, 1) = "=sum($BI$30:" & Range("BI30").End(xlDown).Address & ")"
```

```
        Range("BY34").Offset(0, 2) = "=sum($BJ$30:" & Range("BJ30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 3) = "=sum($BK$30:" & Range("BK30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 4) = "=sum($BL$30:" & Range("BL30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 5) = "=sum($BM$30:" & Range("BM30").End(xlDown).Address & ")"

        Range("BY34").Offset(0, 7) = "=sum($BO$30:" & Range("BO30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 8) = "=sum($BP$30:" & Range("BP30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 9) = "=sum($BQ$30:" & Range("BQ30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 10) = "=sum($BR$30:" & Range("BR30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 11) = "=sum($BS$30:" & Range("BS30").End(xlDown).Address & ")"
        Range("BY34").Offset(0, 12) = "=sum($BT$30:" & Range("BT30").End(xlDown).Address & ")"

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        'Set up the constraints to ensure the available stock is used:

        wbConstraint "BY34:CD34", "<=", "BY36:CD36", "BY35:CD35"
        wbConstraint "CF34:CK34", "<=", "CF36:CK36", "CF35:CK35"


        'Set up the cell to be maximized, "D":

        Range("BT30").End(xlDown).Offset(2, 0).Select
        With ActiveCell
            .Formula = "D"
            .Font.Bold = True
        End With



        Range("BV30").End(xlDown).Offset(2, 0).Select
        With Selection
            .Value = "0"
            wbAdjust
        End With


        'Tell the WhatsBest! Solver which cell it must maximize:

        wbBest Range("BV30").End(xlDown).Offset(2, 0), "Maximize"

        Dim lngSolutionStatus As Long


        'Dimension arrays to store the allocated amounts of each food category to each agency


        'Perishable goods:

        ReDim Bakerygoods(s, n, k) As Single
        ReDim Cookedgoods(s, n, k) As Single
        ReDim Dairygoods(s, n, k) As Single
        ReDim Fruitgoods(s, n, k) As Single
        ReDim Meatgoods(s, n, k) As Single
        ReDim Vegetablegoods(s, n, k) As Single

        'Non-perishable goods:

        ReDim Boxedgoods(s, n, k) As Single
        ReDim Cannedgoods(s, n, k) As Single
        ReDim Condimentgoods(s, n, k) As Single
        ReDim Luxfoodgoods(s, n, k) As Single
        ReDim Luxgoods(s, n, k) As Single
        ReDim Othergoods(s, n, k) As Single


        'Dimension an array to store the total Kcal value per person at _
        each agency and the percentage of satisfied target nutritional need per person:

        ReDim VKcalppl(s, n, k) As Single
        ReDim PNS(s, n, k) As Single              'PNS = Percentage of Need Satisfied
        ReDim MinKcalpp(n, k) As Single           'MinKcalpp = Minimum Kcal value per person
        ReDim Kcalpp(n, k) As Single              'AvgKcalpp = Acerage Kcal value per person

        'Read in the target nutritional value per person:

        TN = Range("B26").Value
```

```vb
        'Define variables to be used to compute descriptive statistics:

        Dim E1 As Single        'To compute average proportion of agencies' need satisfied
        Dim E2 As Single        'To compute variance of proportion of agencies' need satisfied
        Dim E3 As Integer       'To compute the number of times an agency's satisfied proportion _
        of need falls below a critical level
        Dim E4 As Single        'To record the Min Kcal/person value for each day and iteration
        Dim E5 As Single        'To record how many times a NON-optimal solution is obtained
        Dim E6 As Single        'To record the average Kcal/person value for each day and iteration

        'Initialize these variables to zero value

        E1 = 0
        E2 = 0
        E3 = 0
        E4 = 0
        E5 = 0
        E6 = 0


        'For each agency (p) , calculate the amount of each type of product received on a particular _
        day (i) , for a set amount of iterations (j), and then convert this amount into total _
            nutritional value (Kcal) received


        'This multiple loop construct will simulate the "MaxMin Foodvalue allocation:

        For j = 1 To k

            For i = 1 To n

                Range("BX36").Select
                With ActiveCell
                    .Formula = "Stock"
                    .Font.Bold = True
                    .Font.Underline = True

                    'Perishable goods:

                    .Offset(0, 1).Value = Bakery(i, j)
                    .Offset(0, 2).Value = Cook(i, j)
                    .Offset(0, 3).Value = Dairy(i, j)
                    .Offset(0, 4).Value = Fruit(i, j)
                    .Offset(0, 5).Value = Meat(i, j)
                    .Offset(0, 6).Value = Veg(i, j)

                    'Non-perishable goods:

                    .Offset(0, 8).Value = Box(i, j)
                    .Offset(0, 9).Value = Can(i, j)
                    .Offset(0, 10).Value = Con(i, j)
                    .Offset(0, 11).Value = Luxf(i, j)
                    .Offset(0, 12).Value = Luxg(i, j)
                    .Offset(0, 13).Value = Other(i, j)
                End With


                For p = 1 To s

                    Range("D30", Range("D30").End(xlDown)).Find(Agency(p)).Select

                    agencycategory = Range("D30").Offset(p - 1, 1).Value


                    Select Case agencycategory

                    Case "Clinic"

                        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                        'ALLOCATION VALUE BOUNDS
                        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

                        'Lower bound (minimum) allocation values:
                        ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)

                        ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

                        'Upper bound (maximum) allocation values:
                        ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
```

```vb
            ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)



        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Case "Educare"

        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'ALLOCATION VALUE BOUNDS
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        'Lower bound (minimum) allocation values:
        ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)
        ActiveCell.Offset(0, 103).Value = (Dairy(i, j) * agencynum(p)) / (drtotppl * lb)
        ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
        ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)

        ActiveCell.Offset(0, 108).Value = (Box(i, j) * agencynum(p)) / (bxtotppl * lb)
        ActiveCell.Offset(0, 109).Value = (Can(i, j) * agencynum(p)) / (cantotppl * lb)
        ActiveCell.Offset(0, 110).Value = (Con(i, j) * agencynum(p)) / (contotppl * lb)
        ActiveCell.Offset(0, 111).Value = (Luxf(i, j) * agencynum(p)) / (lftotppl * lb)
        ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

        'Upper bound (maximum) allocation values:
        ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
        ActiveCell.Offset(0, 132).Value = (Dairy(i, j) * agencynum(p) * ub) / (drtotppl)
        ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
        ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)

        ActiveCell.Offset(0, 137).Value = (Box(i, j) * agencynum(p) * ub) / (bxtotppl)
        ActiveCell.Offset(0, 138).Value = (Can(i, j) * agencynum(p) * ub) / (cantotppl)
        ActiveCell.Offset(0, 139).Value = (Con(i, j) * agencynum(p) * ub) / (contotppl)
        ActiveCell.Offset(0, 140).Value = (Luxf(i, j) * agencynum(p) * ub) / (lftotppl)
        ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)

        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Case "FeedingScheme"

        'Lower bound (minimum) allocation values:
        ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)
        ActiveCell.Offset(0, 102).Value = (Cook(i, j) * agencynum(p)) / (cktotppl * lb)
        ActiveCell.Offset(0, 103).Value = (Dairy(i, j) * agencynum(p)) / (drtotppl * lb)
        ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
        ActiveCell.Offset(0, 105).Value = (Meat(i, j) * agencynum(p)) / (mttotppl * lb)
        ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)

        ActiveCell.Offset(0, 108).Value = (Box(i, j) * agencynum(p)) / (bxtotppl * lb)
        ActiveCell.Offset(0, 109).Value = (Can(i, j) * agencynum(p)) / (cantotppl * lb)
        ActiveCell.Offset(0, 110).Value = (Con(i, j) * agencynum(p)) / (contotppl * lb)
        ActiveCell.Offset(0, 111).Value = (Luxf(i, j) * agencynum(p)) / (lftotppl * lb)

        ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

        'Upper bound (maximum) allocation values:
        ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
        ActiveCell.Offset(0, 131).Value = (Cook(i, j) * agencynum(p) * ub) / (cktotppl)
        ActiveCell.Offset(0, 132).Value = (Dairy(i, j) * agencynum(p) * ub) / (drtotppl)
        ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
        ActiveCell.Offset(0, 134).Value = (Meat(i, j) * agencynum(p) * ub) / (mttotppl)
        ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)

        ActiveCell.Offset(0, 137).Value = (Box(i, j) * agencynum(p) * ub) / (bxtotppl)
        ActiveCell.Offset(0, 138).Value = (Can(i, j) * agencynum(p) * ub) / (cantotppl)
        ActiveCell.Offset(0, 139).Value = (Con(i, j) * agencynum(p) * ub) / (contotppl)
        ActiveCell.Offset(0, 140).Value = (Luxf(i, j) * agencynum(p) * ub) / (lftotppl)

        ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)

    Case "NutritionalCentre"

        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'ALLOCATION VALUE BOUNDS
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        'Lower bound (minimum) allocation values:
```

```
                ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)

                ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)
                ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

                'Upper bound (maximum) allocation values:
                ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)

                ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)
                ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)




                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            Case "Satellite"

                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                'ALLOCATION VALUE BOUNDS
                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

                'Lower bound (minimum) allocation values:
                ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)
                ActiveCell.Offset(0, 102).Value = (Cook(i, j) * agencynum(p)) / (cktotppl * lb)
                ActiveCell.Offset(0, 103).Value = (Dairy(i, j) * agencynum(p)) / (drtotppl * lb)
                ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
                ActiveCell.Offset(0, 105).Value = (Meat(i, j) * agencynum(p)) / (mttotppl * lb)
                ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)

                ActiveCell.Offset(0, 108).Value = (Box(i, j) * agencynum(p)) / (bxtotppl * lb)
                ActiveCell.Offset(0, 109).Value = (Can(i, j) * agencynum(p)) / (cantotppl * lb)
                ActiveCell.Offset(0, 110).Value = (Con(i, j) * agencynum(p)) / (contotppl * lb)
                ActiveCell.Offset(0, 111).Value = (Luxf(i, j) * agencynum(p)) / (lftotppl * lb)
                ActiveCell.Offset(0, 112).Value = (Luxg(i, j) * agencynum(p)) / (lgtotppl * lb)
                ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

                'Upper bound (maximum) allocation values:
                ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
                ActiveCell.Offset(0, 131).Value = (Cook(i, j) * agencynum(p) * ub) / (cktotppl)
                ActiveCell.Offset(0, 132).Value = (Dairy(i, j) * agencynum(p) * ub) / (drtotppl)
                ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
                ActiveCell.Offset(0, 134).Value = (Meat(i, j) * agencynum(p) * ub) / (mttotppl)
                ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)

                ActiveCell.Offset(0, 137).Value = (Box(i, j) * agencynum(p) * ub) / (bxtotppl)
                ActiveCell.Offset(0, 138).Value = (Can(i, j) * agencynum(p) * ub) / (cantotppl)
                ActiveCell.Offset(0, 139).Value = (Con(i, j) * agencynum(p) * ub) / (contotppl)
                ActiveCell.Offset(0, 140).Value = (Luxf(i, j) * agencynum(p) * ub) / (lftotppl)
                ActiveCell.Offset(0, 141).Value = (Luxg(i, j) * agencynum(p) * ub) / (lgtotppl)
                ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)


                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            Case "School"

                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                'ALLOCATION VALUE BOUNDS
                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

                'Lower bound (minimum) allocation values:
                ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)

                ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
                ActiveCell.Offset(0, 111).Value = (Luxf(i, j) * agencynum(p)) / (lftotppl * lb)
                ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

                'Upper bound (maximum) allocation values:
                ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)

                ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
                ActiveCell.Offset(0, 140).Value = (Luxf(i, j) * agencynum(p) * ub) / (lftotppl)
                ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)


                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```vb
        Case "Shelter"

            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            'ALLOCATION VALUE BOUNDS
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

            'Lower bound (minimum) allocation values:
            ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)
            ActiveCell.Offset(0, 102).Value = (Cook(i, j) * agencynum(p)) / (cktotppl * lb)
            ActiveCell.Offset(0, 103).Value = (Dairy(i, j) * agencynum(p)) / (drtotppl * lb)
            ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
            ActiveCell.Offset(0, 105).Value = (Meat(i, j) * agencynum(p)) / (mttotppl * lb)
            ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)

            ActiveCell.Offset(0, 108).Value = (Box(i, j) * agencynum(p)) / (bxtotppl * lb)
            ActiveCell.Offset(0, 109).Value = (Can(i, j) * agencynum(p)) / (cantotppl * lb)
            ActiveCell.Offset(0, 110).Value = (Con(i, j) * agencynum(p)) / (contotppl * lb)
            ActiveCell.Offset(0, 111).Value = (Luxf(i, j) * agencynum(p)) / (lftotppl * lb)
            ActiveCell.Offset(0, 112).Value = (Luxg(i, j) * agencynum(p)) / (lgtotppl * lb)
            ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

            'Upper bound (maximum) allocation values:
            ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
            ActiveCell.Offset(0, 131).Value = (Cook(i, j) * agencynum(p) * ub) / (cktotppl)
            ActiveCell.Offset(0, 132).Value = (Dairy(i, j) * agencynum(p) * ub) / (drtotppl)
            ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
            ActiveCell.Offset(0, 134).Value = (Meat(i, j) * agencynum(p) * ub) / (mttotppl)
            ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)

            ActiveCell.Offset(0, 137).Value = (Box(i, j) * agencynum(p) * ub) / (bxtotppl)
            ActiveCell.Offset(0, 138).Value = (Can(i, j) * agencynum(p) * ub) / (cantotppl)
            ActiveCell.Offset(0, 139).Value = (Con(i, j) * agencynum(p) * ub) / (contotppl)
            ActiveCell.Offset(0, 140).Value = (Luxf(i, j) * agencynum(p) * ub) / (lftotppl)
            ActiveCell.Offset(0, 141).Value = (Luxg(i, j) * agencynum(p) * ub) / (lgtotppl)
            ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)


            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


        Case "SoupKitchen"

            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            'ALLOCATION VALUE BOUNDS
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

            'Lower bound (minimum) allocation values:
            ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)
            ActiveCell.Offset(0, 102).Value = (Cook(i, j) * agencynum(p)) / (cktotppl * lb)

            ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
            ActiveCell.Offset(0, 105).Value = (Meat(i, j) * agencynum(p)) / (mttotppl * lb)
            ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)

            ActiveCell.Offset(0, 108).Value = (Box(i, j) * agencynum(p)) / (bxtotppl * lb)
            ActiveCell.Offset(0, 109).Value = (Can(i, j) * agencynum(p)) / (cantotppl * lb)
            ActiveCell.Offset(0, 110).Value = (Con(i, j) * agencynum(p)) / (contotppl * lb)
            ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

            'Upper bound (maximum) allocation values:
            ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
            ActiveCell.Offset(0, 131).Value = (Cook(i, j) * agencynum(p) * ub) / (cktotppl)

            ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
            ActiveCell.Offset(0, 134).Value = (Meat(i, j) * agencynum(p) * ub) / (mttotppl)
            ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)

            ActiveCell.Offset(0, 137).Value = (Box(i, j) * agencynum(p) * ub) / (bxtotppl)
            ActiveCell.Offset(0, 138).Value = (Can(i, j) * agencynum(p) * ub) / (cantotppl)
            ActiveCell.Offset(0, 139).Value = (Con(i, j) * agencynum(p) * ub) / (contotppl)
            ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)


            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


        Case "SupportGroup"
```

```
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                'ALLOCATION VALUE BOUNDS
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

                'Lower bound (minimum) allocation values:
                ActiveCell.Offset(0, 101).Value = (Bakery(i, j) * agencynum(p)) / (bktotppl * lb)
                ActiveCell.Offset(0, 102).Value = (Cook(i, j) * agencynum(p)) / (cktotppl * lb)
                ActiveCell.Offset(0, 104).Value = (Fruit(i, j) * agencynum(p)) / (frtotppl * lb)
                ActiveCell.Offset(0, 105).Value = (Meat(i, j) * agencynum(p)) / (mttotppl * lb)
                ActiveCell.Offset(0, 106).Value = (Veg(i, j) * agencynum(p)) / (vgtotppl * lb)

                ActiveCell.Offset(0, 108).Value = (Box(i, j) * agencynum(p)) / (bxtotppl * lb)

                ActiveCell.Offset(0, 110).Value = (Con(i, j) * agencynum(p)) / (contotppl * lb)
                ActiveCell.Offset(0, 111).Value = (Luxf(i, j) * agencynum(p)) / (lftotppl * lb)
                ActiveCell.Offset(0, 112).Value = (Luxg(i, j) * agencynum(p)) / (lgtotppl * lb)
                ActiveCell.Offset(0, 113).Value = (Other(i, j) * agencynum(p)) / (othtotppl * lb)

                'Upper bound (maximum) allocation values:
                ActiveCell.Offset(0, 130).Value = (Bakery(i, j) * agencynum(p) * ub) / (bktotppl)
                ActiveCell.Offset(0, 131).Value = (Cook(i, j) * agencynum(p) * ub) / (cktotppl)
                ActiveCell.Offset(0, 133).Value = (Fruit(i, j) * agencynum(p) * ub) / (frtotppl)
                ActiveCell.Offset(0, 134).Value = (Meat(i, j) * agencynum(p) * ub) / (mttotppl)
                ActiveCell.Offset(0, 135).Value = (Veg(i, j) * agencynum(p) * ub) / (vgtotppl)

                ActiveCell.Offset(0, 137).Value = (Box(i, j) * agencynum(p) * ub) / (bxtotppl)

                ActiveCell.Offset(0, 139).Value = (Con(i, j) * agencynum(p) * ub) / (contotppl)
                ActiveCell.Offset(0, 140).Value = (Luxf(i, j) * agencynum(p) * ub) / (lftotppl)
                ActiveCell.Offset(0, 141).Value = (Luxg(i, j) * agencynum(p) * ub) / (lgtotppl)
                ActiveCell.Offset(0, 142).Value = (Other(i, j) * agencynum(p) * ub) / (othtotppl)


        End Select

    Next p

    Columns("F:FZ").EntireColumn.AutoFit


    On Error Resume Next


    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'SOLVE THE MODEL
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    wbSolve lngSolutionStatus

    Select Case lngSolutionStatus

    Case 1

        'MsgBox "The model is: Globally Optimal"

    Case 2

        'MsgBox "The model is: Globally Optimal, " & Range ("WBMAX")

    Case 3

        'MsgBox "The model is: Infeasible"
        E5 = E5 + 1

    Case 4

        'MsgBox "The model is: Unbounded"
        E5 = E5 + 1

    Case 5
```

```vba
            'MsgBox "The model is: Feasible"
            E5 = E5 + 1

        Case 6

            'MsgBox "The model is: Infeasible or Unbounded"
            E5 = E5 + 1

        Case 7

            'MsgBox "The model is: Near Optimal"
            E5 = E5 + 1

        Case 8

            'MsgBox "The model is: Locally Optimal"
            E5 = E5 + 1

        Case 9

            'MsgBox "The model is: Locally Infeasible"
            E5 = E5 + 1

        Case 10

            'MsgBox "The model is: Cutoff"
            E5 = E5 + 1

        Case 11

            'MsgBox "The model is: Numerical Error"
            E5 = E5 + 1

        Case 12

            'MsgBox "The model is: Unknown"
            E5 = E5 + 1

        Case 13

            'MsgBox "The model is: Unloaded"
            E5 = E5 + 1

        Case 14

            'MsgBox "The model is: Loaded"
            E5 = E5 + 1

        Case Else

            'MsgBox "Solution status unknown"
            E5 = E5 + 1

    End Select




    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'COMPUTE DESCRIPTIVE STATISTICS
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Worksheets("Home").Activate

    For p = 1 To s

        Range("D30", Range("D30").End(xlDown)).Find(Agency(p)).Select


        Bakerygoods(p, i, j) = ActiveCell.Offset(0, 56).Value
        Cookedgoods(p, i, j) = ActiveCell.Offset(0, 57).Value
        Dairygoods(p, i, j) = ActiveCell.Offset(0, 58).Value
        Fruitgoods(p, i, j) = ActiveCell.Offset(0, 59).Value
        Meatgoods(p, i, j) = ActiveCell.Offset(0, 60).Value
        Vegetablegoods(p, i, j) = ActiveCell.Offset(0, 61).Value
```

```
                    Boxedgoods(p, i, j) = ActiveCell.Offset(0, 63).Value
                    Cannedgoods(p, i, j) = ActiveCell.Offset(0, 64).Value
                    Condimentgoods(p, i, j) = ActiveCell.Offset(0, 65).Value
                    Luxfoodgoods(p, i, j) = ActiveCell.Offset(0, 66).Value
                    Luxgoods(p, i, j) = ActiveCell.Offset(0, 67).Value
                    Othergoods(p, i, j) = ActiveCell.Offset(0, 68).Value

                    VKcalppl(p, i, j) = ActiveCell.Offset(0, 70).Value

                    'Compute the percentage satisfied nutritional need per person at each agency:

                    PNS(p, i, j) = (VKcalppl(p, i, j) / TN) * 100

                    'Compute descriptive statistics:

                    E1 = E1 + PNS(p, i, j)

                    E2 = E2 + (PNS(p, i, j) / 100) ^ 2

                    If PNS(p, i, j) < 50 Then

                        E3 = E3 + 1

                    End If


                Next p

                MinKcalpp(i, j) = Range("BV73").Value

                E4 = E4 + MinKcalpp(i, j)

                Kcalpp(i, j) = Application.WorksheetFunction.Average(Range("BV30"), Range("BV30").End(xlDown))

                E6 = E6 + Kcalpp(i, j)

            Next i

        Next j

        AVGNDSTSFCTN = Round(E1 / (s * n * k), 2)
        AVGVARNDSTSFCTN = Round(((E2) / (s * n * k) - (AVGNDSTSFCTN / 100) ^ 2) * 100, 2)
        AVGMINKCALPP = Round(E4 / (n * k), 2)
        AVGNUM = Round(E3 / (n * k), 0)
        AVGKCALPP = Round(E6 / (n * k), 2)

        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'NUMBER-BASED ALLOCATION
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''



    'Dimension variables for each of the types of food to be allocated to each category of _
    agency as integer variables (that will take on a value of 0 or 1)

    Dim allobk As Integer
    Dim allobx As Integer
    Dim allocan As Integer
    Dim allocon As Integer
    Dim allock As Integer
    Dim allodr As Integer
    Dim allofr As Integer
    Dim allolf As Integer
    Dim allolg As Integer
    Dim allomt As Integer
    Dim allovg As Integer
    Dim allooth As Integer


    'Dimension vectors for the allocation factor for each type of product that will be allocated _
    to the agencies

    ReDim agencybkallo(s) As Single      'Bakery allocation factor for the agency

    ReDim agencybxallo(s) As Single      'Boxed goods allocation factor for the agency

    ReDim agencycanallo(s) As Single     'Canned goods allocation factor for the agency

    ReDim agencyconallo(s) As Single     'Condiments allocation factor for the agency

    ReDim agencyckallo(s) As Single      'Cooked food allocation factor for the agency

    ReDim agencydrallo(s) As Single      'Dairy allocation factor for the agency

    ReDim agencyfrallo(s) As Single      'Fruit allocation factor for the agency

    ReDim agencylfallo(s) As Single      'Luxury foods allocation factor for the agency

    ReDim agencylgallo(s) As Single      'Luxury goods allocation factor for the agency

    ReDim agencymallo(s) As Single       'Meat allocation factor for the agency

    ReDim agencyvgallo(s) As Single      'Vegetables allocation factor for the agency

    ReDim agencyotallo(s) As Single      'Other goods allocation factor for the agency

    Dim agencycategorys As String


    'Insert a For Next loop to run through data of agencies, reading in their name, number of ppl _
    being supported and category:

    For p = 1 To s

        Agency(p) = Range("D30").Offset(p - 1).Value

        agencynum(p) = Range("D30").Offset(p - 1, 2).Value

        agencycategorys = Range("D30").Offset(p - 1, 1).Value



        'Insert a Select Case construct to make 'decisions' as to what types of products are allocated _
        to each agency based on their category (according to Table 9.2 in Appendix A)

        Select Case agencycategorys

        Case "Clinic"
            allobk = 1
```

```
            allobx = 0
            allocan = 0
            allocon = 0
            allock = 0
            allodr = 0
            allofr = 0
            allolf = 0
            allolg = 0
            allomt = 0
            allovg = 0
            allooth = 1

        Case "Educare"
            allobk = 1
            allobx = 1
            allocan = 1
            allocon = 1
            allock = 0
            allodr = 1
            allofr = 1
            allolf = 1
            allolg = 0
            allomt = 0
            allovg = 1
            allooth = 1

        Case "FeedingScheme"
            allobk = 1
            allobx = 1
            allocan = 1
            allocon = 1
            allock = 1
            allodr = 1
            allofr = 1
            allolf = 1
            allolg = 0
            allomt = 1
            allovg = 1
            allooth = 1

        Case "NutritionalCentre"
            allobk = 1
            allobx = 0
            allocan = 0
            allocon = 0
            allock = 0
            allodr = 0
            allofr = 0
            allolf = 0
            allolg = 0
            allomt = 0
            allovg = 1
            allooth = 1

        Case "School"
            allobk = 1
            allobx = 0
            allocan = 0
            allocon = 0
            allock = 0
            allodr = 0
            allofr = 1
            allolf = 1
            allolg = 0
            allomt = 0
            allovg = 0
            allooth = 1

        Case "Shelter"
            allobk = 1
            allobx = 1
            allocan = 1
            allocon = 1
            allock = 1
            allodr = 1
            allofr = 1
            allolf = 1
            allolg = 1
            allomt = 1
            allovg = 1
```

```vb
            allooth = 1

        Case "SoupKitchen"
            allobk = 1
            allobx = 1
            allocan = 1
            allocon = 1
            allock = 1
            allodr = 0
            allofr = 1
            allolf = 0
            allolg = 0
            allomt = 1
            allovg = 1
            allooth = 1

        Case "SupportGroup"
            allobk = 1
            allobx = 1
            allocan = 0
            allocon = 1
            allock = 1
            allodr = 0
            allofr = 1
            allolf = 1
            allolg = 1
            allomt = 1
            allovg = 1
            allooth = 1

        End Select



        'Agency's allocation factors for each type of product are calculated as the number of ppl _
        the agency supports divided by the total number of ppl across all agencies who are to _
            receive that product, this factor is multiplied by a binary variable that indicates _
            whether that agency is to receive a certain product type based on its category

        agencybkallo(p) = allobk * (agencynum(p) / bktotppl)

        agencybxallo(p) = allobx * (agencynum(p) / bxtotppl)

        agencycanallo(p) = allocan * (agencynum(p) / cantotppl)

        agencyconallo(p) = allocon * (agencynum(p) / contotppl)

        agencyckallo(p) = allock * (agencynum(p) / cktotppl)

        agencydrallo(p) = allodr * (agencynum(p) / drtotppl)

        agencyfrallo(p) = allofr * (agencynum(p) / frtotppl)

        agencylfallo(p) = allolf * (agencynum(p) / lftotppl)

        agencylgallo(p) = allolg * (agencynum(p) / lgtotppl)

        agencymallo(p) = allomt * (agencynum(p) / mttotppl)

        agencyvgallo(p) = allovg * (agencynum(p) / vgtotppl)

        agencyotallo(p) = allooth * (agencynum(p) / othtotppl)


    Next p



    'Dimension arrays to store the allocated amounts of each food category to each agency


    'Perishable goods:

    ReDim Bakerygood(s, n, k) As Single
    ReDim Cookedgood(s, n, k) As Single
    ReDim Dairygood(s, n, k) As Single
    ReDim Fruitgood(s, n, k) As Single
    ReDim Meatgood(s, n, k) As Single
    ReDim Vegetablegood(s, n, k) As Single
```

```vb
        'Non-perishable goods:

        ReDim Boxedgood(s, n, k) As Single
        ReDim Cannedgood(s, n, k) As Single
        ReDim Condimentgood(s, n, k) As Single
        ReDim Luxfoodgood(s, n, k) As Single
        ReDim Luxgood(s, n, k) As Single
        ReDim Othergood(s, n, k) As Single



        'Dimension arrays to store the allocated nutritional value (Kcal) of each food category to _
        each agency


        'Perishable goods:

        ReDim BakeryKcal(s, n, k) As Single
        ReDim CookedKcal(s, n, k) As Single
        ReDim DairyKcal(s, n, k) As Single
        ReDim FruitKcal(s, n, k) As Single
        ReDim MeatKcal(s, n, k) As Single
        ReDim VegetableKcal(s, n, k) As Single


        'Non-perishable goods:

        ReDim BoxedKcal(s, n, k) As Single
        ReDim CannedKcal(s, n, k) As Single
        ReDim CondimentKcal(s, n, k) As Single
        ReDim LuxfoodKcal(s, n, k) As Single
        ReDim LuxgoodKcal(s, n, k) As Single
        ReDim OtherKcal(s, n, k) As Single



        'Read in Kcal/kg values for each product category


        'Perishable goods:

        BkKcal = Range("B12").Value
        CkKcal = Range("B13").Value
        DKcal = Range("B14").Value
        FKcal = Range("B15").Value
        MKcal = Range("B16").Value
        VKcal = Range("B17").Value


        'Non-perishable goods:

        BxKcal = Range("B19").Value
        CaKcal = Range("B20").Value
        CoKcal = Range("B21").Value
        LfKcal = Range("B22").Value
        LgKcal = Range("B23").Value
        OKcal = Range("B24").Value



        'Dimension arrays to store the total nutritional value of food allocated to an agency, the _
        total nutritional value per person being supported by that agency, and the percentage of _
            satisfied target nutritional need per person

        ReDim VTotKcals(s, n, k) As Single
        ReDim VKcalppls(s, n, k) As Single
        ReDim PNSat(s, n, k) As Single                    'PNSat = Percentage of Need Satisfied
        ReDim MinKcal(n, k) As Single
        ReDim Kcal(n, k) As Single


        'Define variables to be used to compute descriptive statistics:

        Dim S1 As Single        'To compute average proportion of agencies' need satisfied
        Dim S2 As Single        'To compute variance of proportion of agencies' need satisfied
        Dim S3 As Integer       'To compute the number of times an agency's satisfied proportion _
        of need falls below a critical level
        Dim S4 As Single        'To record the Min Kcal/person value for each day and iteration
        Dim S5 As Single        'To record the average Kcal/person value for each day and iteration

        'Initialize these variables to zero value
```

```vba
        S1 = 0
        S2 = 0
        S3 = 0
        S4 = 0
        S5 = 0

    'For each agency (p) , calculate the amount of each type of product received on a particular _
    day (i) , for a set amount of iterations (j), and then convert this amount into total _
        nutritional value (Kcal) received


    For j = 1 To k

        For i = 1 To n

            For p = 1 To s

                    Bakerygood(p, i, j) = agencybkallo(p) * Bakery(i, j)
                    Cookedgood(p, i, j) = agencyckallo(p) * Cook(i, j)
                    Dairygood(p, i, j) = agencydrallo(p) * Dairy(i, j)
                    Fruitgood(p, i, j) = agencyfrallo(p) * Fruit(i, j)
                    Meatgood(p, i, j) = agencymallo(p) * Meat(i, j)
                    Vegetablegood(p, i, j) = agencyvgallo(p) * Veg(i, j)


                    Boxedgood(p, i, j) = agencybxallo(p) * Box(i, j)
                    Cannedgood(p, i, j) = agencycanallo(p) * Can(i, j)
                    Condimentgood(p, i, j) = agencyconallo(p) * Con(i, j)
                    Luxfoodgood(p, i, j) = agencylfallo(p) * Luxf(i, j)
                    Luxgood(p, i, j) = agencylgallo(p) * Luxg(i, j)
                    Othergood(p, i, j) = agencyotallo(p) * Other(i, j)


                    'Convert amount of goods received into Kcal:

                    BakeryKcal(p, i, j) = BkKcal * Bakerygood(p, i, j)
                    CookedKcal(p, i, j) = CkKcal * Cookedgood(p, i, j)
                    DairyKcal(p, i, j) = DKcal * Dairygood(p, i, j)
                    FruitKcal(p, i, j) = FKcal * Fruitgood(p, i, j)
                    MeatKcal(p, i, j) = MKcal * Meatgood(p, i, j)
                    VegetableKcal(p, i, j) = VKcal * Vegetablegood(p, i, j)


                    BoxedKcal(p, i, j) = BxKcal * Boxedgood(p, i, j)
                    CannedKcal(p, i, j) = CaKcal * Cannedgood(p, i, j)
                    CondimentKcal(p, i, j) = CoKcal * Condimentgood(p, i, j)
                    LuxfoodKcal(p, i, j) = LfKcal * Luxfoodgood(p, i, j)
                    LuxgoodKcal(p, i, j) = LgKcal * Luxgood(p, i, j)
                    OtherKcal(p, i, j) = OKcal * Othergood(p, i, j)


                    'Compute the total nutritional value of food allocated to each agency:

                    VTotKcals(p, i, j) = (BakeryKcal(p, i, j) + CookedKcal(p, i, j) + DairyKcal(p, i, j) _
                        + FruitKcal(p, i, j) + MeatKcal(p, i, j) + VegetableKcal(p, i, j) _
                        + BoxedKcal(p, i, j) + CannedKcal(p, i, j) + CondimentKcal(p, i, j) _
                        + LuxfoodKcal(p, i, j) + LuxgoodKcal(p, i, j) + OtherKcal(p, i, j))


                    'Compute the total nutritional value per person supported by each agency:

                    VKcalppls(p, i, j) = VTotKcals(p, i, j) / agencynum(p)

                    Range("G30").Offset(p - 1, 0).Value = VKcalppls(p, i, j)

                    'Compute the percentage satisfied nutritional need per person at each agency:

                    PNSat(p, i, j) = (VKcalppls(p, i, j) / TN) * 100

                    'Compute descriptive statistics:

                    S1 = S1 + PNSat(p, i, j)

                    S2 = S2 + (PNSat(p, i, j) / 100) ^ 2
```

```vba
                  If PNSat(p, i, j) < 50 Then

                      S3 = S3 + 1

                  End If


          Next p

          MinKcal(i, j) = Application.WorksheetFunction.Min(Range("G30", Range("G30").End(xlDown)))

          S4 = S4 + MinKcal(i, j)

          Kcal(i, j) = Application.WorksheetFunction.Average(Range("G30"), Range("G30").End(xlDown))

          S5 = S5 + Kcal(i, j)

          Range("G30", Range("G30").End(xlDown)).ClearContents


      Next i


  Next j


  AVGNEEDSTSFCTN = Round(S1 / (s * n * k), 2)
  AVGVARNEEDSTSFCTN = Round(((S2) / (s * n * k) - (AVGNEEDSTSFCTN / 100) ^ 2) * 100, 2)
  AVGNUMBER = Round(S3 / (n * k), 0)
  AVGMINKCAL = Round(S4 / (n * k), 2)
  AVGKCAL = Round(S5 / (n * k), 2)

  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  'Construct a results table to display performance of allocation policy:
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


  'First, clear any previous results tables:

  Worksheets("Home").Activate

  Range("Z30").ClearContents

  Range("D5:F20").Select
  With Selection
      .ClearContents
      .ClearFormats
  End With

  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  'This section creates headings for the table:

  Range("D5").Formula = "Simulation Results"
  Selection.Font.Bold = True
  Selection.Font.Underline = xlUnderlineStyleSingle
  With Selection.Font
      .Name = "Calibri"
      .Size = 14
      .Underline = xlUnderlineStyleSingle
  End With


  Range("D7").FormulaR1C1 = "Number of agencies:"
  Range("D9").FormulaR1C1 = "Total # beneficiaries:"
  Range("D11").FormulaR1C1 = "Allocation Policy:"
  Range("D13").FormulaR1C1 = "Avg. Kcal/person value:"
  Range("D15").FormulaR1C1 = "Avg. Min Kcal/person value:"
  Range("D17").FormulaR1C1 = "Avg. agency's % of target need satisfied :"
```

```vba
        Range("D19").FormulaR1C1 = "Avg. Variance in agency's % of target need satisfied:"
        Range("D21").FormulaR1C1 = "Avg. # Agencies receiving < 50% of target % need satisfaction:"

        Range("D6:D22").Select

        With Selection.Font
            .Size = 13
            .Bold = True
            .Underline = False
        End With
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        'This section draws borders around and in the table:

        Range("D5:F5").Select
        With Selection.Interior
            .Pattern = xlSolid
            .PatternColorIndex = xlAutomatic
            .color = 5287936
        End With
        Range("D5:F22").Select
        Selection.Borders(xlDiagonalDown).LineStyle = xlNone
        Selection.Borders(xlDiagonalUp).LineStyle = xlNone
        With Selection.Borders(xlEdgeLeft)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        With Selection.Borders(xlEdgeTop)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        With Selection.Borders(xlEdgeRight)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        Selection.Borders(xlInsideVertical).LineStyle = xlNone
        Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

        Range("D5:F22").Select
        Selection.Borders(xlDiagonalDown).LineStyle = xlNone
        Selection.Borders(xlDiagonalUp).LineStyle = xlNone
        With Selection.Borders(xlEdgeLeft)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        With Selection.Borders(xlEdgeTop)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        With Selection.Borders(xlEdgeRight)
            .LineStyle = xlContinuous
            .Weight = xlMedium
        End With
        Selection.Borders(xlInsideVertical).LineStyle = xlNone
        Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

        Range("E5:F22").Select
        With Selection.Borders(xlEdgeLeft)
            .LineStyle = xlContinuous
            .Weight = xlThin
        End With

        Range("D10:F10").Select
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .Weight = xlThin
        End With

        ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```vba
    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'Display Simulation Results:


    Range("E7").Value = s
    Range("E9").Value = totppl
    Range("E11").Formula = "MAX-MIN allocation"
    Range("F11").Formula = "NUMBER-BASED allocation"

    Range("E13").Value = AVGKCALPP
    Range("F13").Value = AVGKCAL

    Range("E15").Value = AVGMINKCALPP
    Range("F15").Value = AVGMINKCAL

    Range("E17").Value = AVGNDSTSFCTN
    Range("F17").Value = AVGNEEDSTSFCTN

    Range("E19").Value = AVGVARNDSTSFCTN
    Range("F19").Value = AVGVARNEEDSTSFCTN

    Range("E21").Value = AVGNUM
    Range("F21").Value = AVGNUMBER

    Range("E23").Value = E5

    Range("E6:F22").Select

    With Selection.Font
        .Size = 12
        .Bold = False
        .Underline = False
    End With

    Range("E13:F19").Select
    Selection.NumberFormat = "0.00"

    Range("E21:F21").NumberFormat = "0"


    Range("E11:F11").HorizontalAlignment = xlCenter
    Range("E11:F11").Font.Underline = True
    Range("E11:F11").Font.Italic = True
    Range("E11:F11").Font.Bold = True

    Range("E13:F21").HorizontalAlignment = xlRight


    Range("L4:L10").ClearContents

    Columns("F:FZ").EntireColumn.AutoFit

    Range("D5").Select

    Range("E25").Value = Time()

    Application.ScreenUpdating = True



End Sub
```

```vb
    Sub ClinicCon()

        'Lower bound constraints:
        ActiveCell.Offset(0, 87).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 12).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        'Upper bound constraints:
        ActiveCell.Offset(0, 16).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 12).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    End Sub


    Sub EducareCon()

        'Lower bound constraints:
        ActiveCell.Offset(0, 87).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        'Upper bound constraints:
        ActiveCell.Offset(0, 16).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    End Sub
```

```vba
    Sub FeedingSchemeCon()

        'Lower bound constraints
        ActiveCell.Offset(0, 87).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address


        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        'Upper bound constraints:
        ActiveCell.Offset(0, 16).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    End Sub
```

```vba
    Sub NutritionalCentreCon()

        'Lower bound constraints:
        ActiveCell.Offset(0, 87).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 5).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 7).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        'Upper bound constraints:
        ActiveCell.Offset(0, 16).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 5).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 7).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    End Sub



    Sub SchoolCon()

        'Lower bound constraints:
        ActiveCell.Offset(0, 87).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 3).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 7).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        'Upper bound constraints:
        ActiveCell.Offset(0, 16).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 3).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 7).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    End Sub



    Sub ShelterCon()

        'Lower bound constraints
        ActiveCell.Offset(0, 87).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
```

```vb
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    'Upper bound constraints:
    ActiveCell.Offset(0, 16).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

End Sub


Sub SoupKitchenCon()

    'Lower bound constraints:
    ActiveCell.Offset(0, 87).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
```

```vb
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        ActiveCell.Offset(0, 3).Activate
        wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

        'Upper bound constraints:
        ActiveCell.Offset(0, 16).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 2).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 1).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

        ActiveCell.Offset(0, 3).Activate
        wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    End Sub


Sub SupportGroupCon()

    'Lower bound constraints:
    ActiveCell.Offset(0, 87).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -31), ">=", ActiveCell.Offset(0, 14), ActiveCell.Address

    'Upper bound constraints:
    ActiveCell.Offset(0, 16).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address
```

```
    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 2).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

    ActiveCell.Offset(0, 1).Activate
    wbConstraint ActiveCell.Offset(0, -59), "<=", ActiveCell.Offset(0, 15), ActiveCell.Address

End Sub
```