Optimal Routing of Delivery Vehicles

by

Lambertus Johannes Blom

s25283652

Submitted in partial fulfillment of the requirements for the degree of

BACHELORS OF INDUSTRIAL ENGINEERING

in the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

UNIVERSITY OF PRETORIA

October 2010

Optimal Routing of Delivery Vehicles

Lambertus Johannes Blom

02/10/2010

Executive Summary

Transportation is an essential part of any business today. The growth in the global market has forced companies to rethink their strategies when it comes to transportation. The aim of this project is to develop a model that will efficiently utilize a fleet of vehicles to serve several customers with known demand. The business in question is *Tombstone Land*. This business's core competency is the production of granite tombstones. Currently, they have multiple showrooms across the country which showcases various packages of tombstones. Unfortunately, no structure exists within the company to optimize their vehicle routing. Although ample off-the-shelf solutions exist, developing a custom fit, user friendly model that will cope with the current size of the business, is a viable alternative. It will also provide the business with the relevant information should they decide to purchase software to cope with the growth of their business.

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial intelligence
CVRP	Capacitated Vehicle Route Problem
GA	Genetic Algorithm
MDVRP	Multi Depot Vehicle Routing Problem
OVRP	Open Vehicle Routing Problem
OR	Operations Research
SA	Simulated Annealing
SDVRP	Site-Dependant Vehicle Routing Problem
TS	Tabu Search

Chapter 1

Introduction

Tombstone Land is situated in Bronkhorstspruit. The core competency of this business is the production of granite tombstones. Production began during January 2009 and the business has experienced rapid growth ever since.

Holistically, an order is placed after viewing a range of packages showcased in one of their many showrooms across the country. After the order has been placed, the tombstone along with its various components are manufactured and sent to storage. Given the completion of 'enough' orders that are designated for the same region, the tombstones are transported and erected at different graveyards.

Currently, *Tombstone Land* has no framework in place which allows them to optimize their transportation model with respect to Vehicle Route Planning.

Given the amount of competition a company faces today, it is important to consider all aspects of business and identify all areas of improvement. One of these areas is the timely delivery of products to the customer. With the growth in the global market and a shift in power to empower the consumer, it has become absolutely essential to provide competitive service or products at the right place and at the right time.

Transportation forms a large part of the expenditures of most businesses today. For this reason, savings in transportation cost or mismanagement thereof has a notable influence on the financial standing of the business.

1.1 Research Question

Considering a business model like that of *Tombstone Land,* it is apparent that transportation plays an important role. To complete an order, delivery must take place and therefore transportation cost must be incurred. Having numerous showrooms across the country and the ever increasing cost of fuel, one cannot simply transport on a rule of thumb or best guess basis. Research on how vehicles must be routed is to be conducted as to reassure the financial validity of incurring the transportation cost, or stated differently, spend only what is needed to deliver the product to the customer and therefore select the shortest or most cost-effective route. This lead to the research question:

How should Tombstone Land route their vehicles to minimize transportation cost, while at the same time still meet all customer demands?

1.2 Research Design

The main deliverable of this project is to develop a computer program or algorithm that may be used to optimally determine the routes *Tombstone Land* must select for their delivery vehicles, to meet all customer demands while simultaneously lowering or minimizing transportation cost.

This model must ensure that certain restraints (e.g. only one vehicle visits one node or stop) are adhered to, as this will model real world situations.

In order to accomplish this, reference to previous literature on the matter will be conducted. The findings should include some form of solution to the problem, while a study of the business must provide context or constraints to the problem.

1.3 Research Mythology

Figure 1 illustrates the Operations Research process when modeling a problem. This model will be followed when attempting to solve the proposed problem. (Rardin 1998)

Problem: It is important to understand the problem in context. Sufficient information of the project environment along with previous cases should provide a good foundation to identify the core of the problem that *Tombstone Land* faces.

Model: The identified problem must be modeled. An objective function describes the main deliverable while the constraints add real world boundaries to the solution space.

Solution: Solving the model will generate numerical data for a feasible solution. Different techniques exist on how to improve on this solution in order to ensure that one does not stagnate on a local optimum but possibly find the global optimum to the problem.

Decision: Analyzing the numerical data from the solution will provide one with information. This information may be used to make insightful decisions or possibly lead to a new problem which will then again be analyzed and solved using the discussed mythology.

1.4 Document Structure

Chapter 2 covers a literature review of the VRP and the problems they face. Furthermore, findings related to the business are documented and a generally accepted notation for the problem discussed. The physical design of the model is discussed in chapter 3 while the computational evaluation of the model is cover by chapter 4. Chapter 5 provides a short conclusion.

Chapter 2

Literature Review

A review on past literature was done in order to understand the problem better and to possibly find a feasible solution to the problem. Ample research with respect to transport and the problems it faces have been conducted in the past. New frontiers concerning this problem and new philosophies on how to optimize this area of business are still being discovered today. Concerning the VRP problem, a formal definition will be given, the different VRP models will be discussed as well a generally excepted/standard notation of an algorithm noted.

2.1 The VRP definition

A Vehicle routing problem according to (Laporte 1992) and (Van Breedam 2001) may be described as follows: Finding a route that a fleet of vehicles must take to serve a number of customers at stops or nodes. The vehicles depart from and return to a single depot or point.

2.2 The VRP models

Various types of VRP problems exist. A brief description of those will follow.

As noted in an article by (Pisinger & Ropke 2007), most papers on heuristic solutions to VRP's target a specific type of problem to obtain a fine-tuned solution. The five different VRP models discussed in the article are as follows:

VRPTW - Vehicle Routing Problem with Time Windows

This type of VRP is an extension of the CVRP in the sense that time windows are associate to customers, which governs the time in which a customer must be visited.

CVRP - Capacitated Vehicle Routing Problem

In this type of VRP, minimum cost routes are selected to travel from the depot to make deliveries and then return to the depot. The assumption of a homogeneous fleet with a known capacity is made.

MDVRP - Multi-Depot Vehicle Routing Problem

An extension of the CVRP, this type of VRP allows for multiple depots.

SDVRP - Site-Dependant Vehicle Routing Problem

Customers may be served by a sub set of vehicles which need not have the same capacity.

OVRP - Open Vehicle Routing Problem

This type of VRP is similar to the CVRP, but with the exception that vehicles need not return to the depot after delivery has been completed.

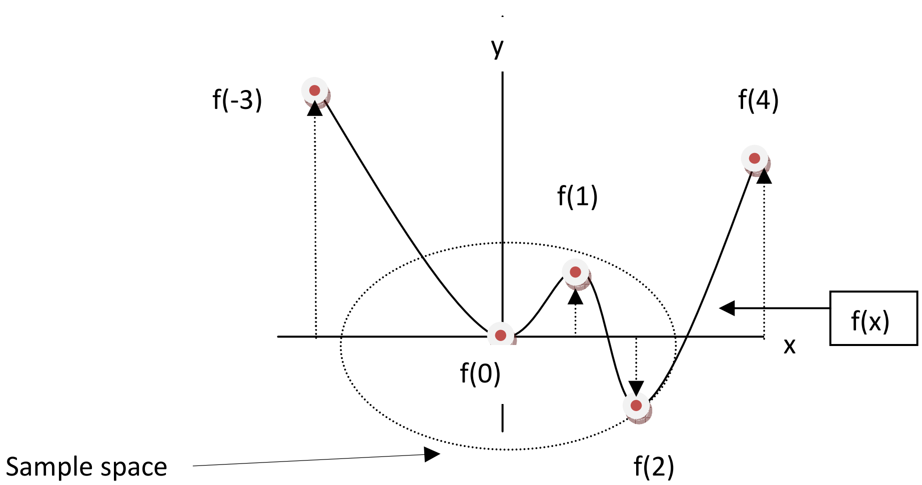
2.3 Heuristics

According to (Winston & Venkataramanan 2004), heuristics are classified as the 'approximate rule-of-thumb techniques' used to solve difficult problem in operations research. Furthermore, a comment is made on the problem these techniques face namely, that they tend to focus on optimizing local optimums or minimums rather than finding the global optimum.

Local and global optimum

In an effort to explain the difference between the local and global optimum, a reference is made to (Stewart 2004):

Figure 1: Arbitrary function f(x)

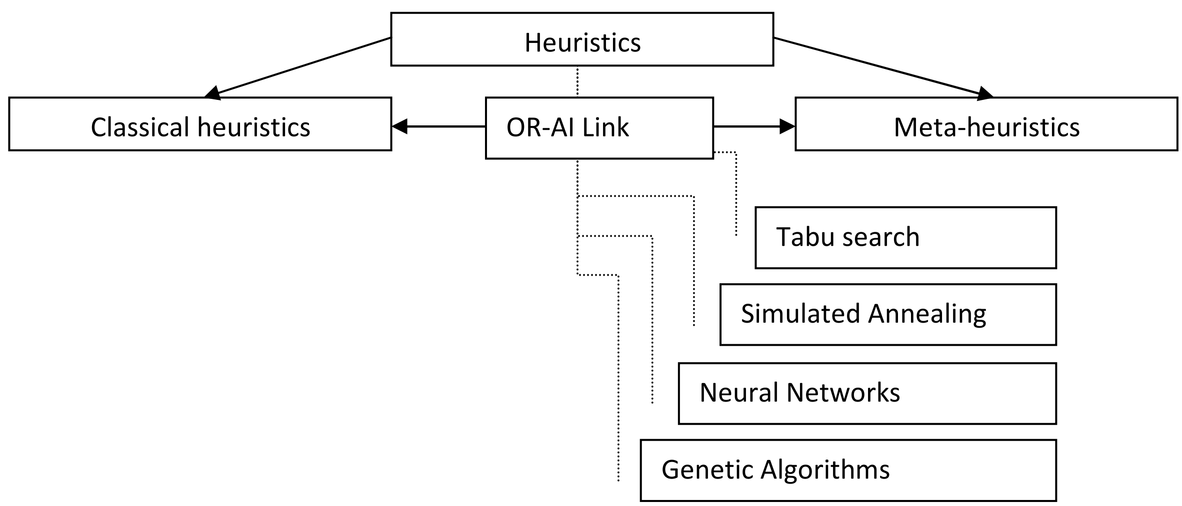


The above function shows arbitrary function f(x) with parameters $-3 \leq x \leq 4$. Note that f(1) is a local maximum, while f(0) is a local minimum. Furthermore, f(2) is both a local minimum and global minimum, while f(4) is neither a local maximum (because it occurs at the endpoint) or global maximum. The absolute/global maximum occurs at f(-3), but this is not a local maximum because it occurs at an endpoint.

From this discussion it becomes apparent why certain models rarely find the global/absolute minimum or maximum to an optimization problem: The sample space rarely accounts for all the possibilities in a real world context.

A discussion on heuristics follows:

Figure 2: Holistic view of Heuristics



The figure shows a pictorial representation of heuristics. Both the TS and SA methods will be discussed in brief as they may possibly be used to solve the problem discussed in this project.

2.3.1 Tabu Search

The TS is a meta-heuristic that may be used to superimpose upon another heuristic to shorten computational time (Raza & Al-Turki 2007) .

By making use of short-term and long-term memories, certain moves are forbidden. In an effort to stop cycling around local minimums/maximums and move away from them, short-term memory is used. Long-term memory helps to guide searches to an area which may contain the optimum solution. (Winston & Venkataramanan 2004)

2.3.2 Simulated Annealing

One characteristic of SA algorithms is that they accept better and worse solutions (with some probability constraint) in order to move away from local neighborhoods. This model was developed from the analogy that followed from the annealing of metal. (Raza & Al-Turki 2007)

The use of a simple swapping strategy and the fact that cost functions are derived easily when using this method makes it very attractive. The SA method is memory-less and optimizes without the use of previous data. Control over parameters of this algorithm may be enforced so convergence takes place in a short period of time. (Winston & Venkataramanan 2004)

2.4 Standard notation of the VRP and algorithm

According to (Joubert 2004) , the following model may be used to model the VRP problem:

Let:

N	be the total number of customers
q_i	be the demand for node i, where $i = \{1,2,\ldots,N\}$
d_{ij}	be the distance traveled between node i and j, where $i, j = \{1,2,...,N\}$
c_{ij}	be the cost associated to the distance traveled between node i and j, where $i, j = \{1,2,\ldots,N\}$
K	be the number of vehicles available in the fleet
p	be the capacity of each vehicle in the homogeneous fleet

The principal decision variable , choosen to be x_{ijk}, is defined as:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travels from node } i \text{ to } j, \text{ where } i, j = \{1,2,\ldots N | i \neq j\}, \\ & k = \{1,2,\ldots,K\} \\ 0 & \text{otherwise} \end{cases}$$

The objective function is:

$$\min z = \sum_{i=0}^{N} \sum_{j=0,j\neq i}^{N} \sum_{k=1}^{K} c_{ij}x_{ijk} \qquad \textbf{(1.1)}$$

subject to:

$$\sum_{j=1}^{N} x_{0jk} = \sum_{j=1}^{N} x_{j0k} = 1 \qquad \forall k = \{1,2,\dots,K\} \qquad \textbf{(1.2)}$$

$$\sum_{j=1}^{N} \sum_{k=1}^{K} x_{0jk} \leq K \qquad \textbf{(1.3)}$$

$$\sum_{i=1;i\neq j}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall j \in \{1,2,\dots,N\} \qquad \textbf{(1.4)}$$

$$\sum_{j=1;j\neq i}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall i \in \{1,2,\dots,N\} \qquad \textbf{(1.5)}$$

$$\sum_{i=1}^{N} q_i \sum_{j=0;j\neq i}^{N} x_{ijk} \leq p \qquad \forall k = \{1,2,\dots,K\} \qquad \textbf{(1.6)}$$

$$x_{ijk} \in \{0,1\} \qquad \textbf{(1.7)}$$

The objective of this function is to minimize the cost enquired when traveling between points i and j. The element c_{ij} is a function of the distance traveled between points i and j multiplied by the cost per distance. The meaning of these equations is as follows:

(1.2) ensures that all the vehicles leaving the depot will again return to the depot (This problem is not an open vehicle routing problem aka. OVRP and therefore, all vehicles must return to the depot after their delivery has been completed)

(1.3) ensures that no more than the maximum vehicles available in the fleet are assigned to a route (e.g 5 vehicles assigned to routes while only having a fleet of 4 vehicles)

(1.4)&(1.5) ensures that all the nodes are visited once and only once.

(1.6) ensures that the total demand on a set route does not exceed the vehicle capacity (e.g. vehicle visiting 2 nodes with cumulative demand of 5 units can only transport a maximum of 4 units)

2.5 Findings relevant to Tombstone Land

Figure 4 is a crude illustration of the current state of how *Tombstone Land* utilizes their fleet for deliveries. A vehicle is sent from the depot to do a delivery in a region or area. Often, the vehicle is underutilized or more than one vehicle has to be sent to do a delivery- which is a clear violation of the proposed algorithm's constraints discussed previously.

Tombstone Land's fleet is made up of a homogeneous collection of five (5) vehicles, each with a maximum capacity of eight (8) units.

Showrooms are located at the following places:

Moloto
Soshanguve
Everton
Nigel
Pumulani
Nqutu
Witbank
Hammanskraal
Tembisa
Elukwati
Tsakane
Krugersdorp
Carolina
Bloedrivier
Hebron

These showrooms service the areas around them with respect to graveyard locations. Because of the negligible distance from the showrooms to the graveyards when compared to the distance from the depot to the showrooms, and the fact that the customer has to meet the delivery vehicle at the showrooms, the showrooms will be considered as the point of delivery for the vehicles.

Table 2 shows the approximate distance matrix with respect to showroom locations across the country, whereas Table 3 to shows the cost matrix associated with the distances traveled. The conversion factor from distance to cost is as follows:

R2/ km	Operating and maintenance cost
20ℓ/100 km = 0.2l/1 km= R1.326/km	Consumption
R3.326/km	Total cost (conversion factor of 3.326)

Chapter 3

Model Formulation

This chapter focuses on the development of the model used to solve the Vehicle Routing Problem faced by *Tombstone Land*. The model was developed from the knowledge gained from previous literature discussed in Chapter 2.

Using both the short-term and long-term memories of the TS method along with the technique of SA which accepts better and worse values, the model calculates an optimal or near optimal route plan for the fleet.

Some requirements were stipulated as to guide the design of the model itself. These requirements are as follow:

I. The program should be user friendly.

II. The distances from the depot to nodes and distances between nodes must be accessible and inexpensive to change.

III. The program must aid the operator in the use of the program in such a manner that the assumptions inherently made to solve the problem is not violated (e.g. warning message is displayed when demand exceeds the vehicle capacity).

3.1 Input data

Given the nature of the Vehicle Routing Problem, data related to a specific problem must be obtained and entered into the algorithm. The data includes the following:

- The number of customers or nodes.
- The demand at each node.
- The distances from the depot to the nodes.
- The distances between the nodes.

As this algorithm was developed specifically for *Tombstone Land,* assumptions concerning input data are as follows:

- The demand at any node does not exceed the vehicle capacity.
- The vehicle capacity is restricted to eight (8) units.
- The fleet consists of a Homogeneous fleet of five (5) vehicles.
- Distances to and between nodes are obtained from the n x n symmetrical distance matrix located in the Excel spreadsheet.
- The demand for the depot is set at zero (0).
- At any time, the depot is considered to be Node 1.
- The amount of nodes does not exceed sixteen (16).

3.1.1 Excel data input

The data required to solve a specific problem can be interred into the designed Excel spreadsheet. The spreadsheet contains the distance matrix which shows the distances to and between all nodes. A **demand** row captures the demand at each node while assuming the demand at the depot is zero (0). If the demand for a particular node is equal to zero, that node is removed from the second distance matrix which will serve as the input distance matrix for calculations done in MATLAB. In an effort to prevent demand data exceeding vehicle capacity being entered into the algorithm, the **demand** row was split into two new demand rows namely, **Demand1** and **Node** respectively. The **Node** cell assigns a number to the locations, as the output of the model is in a numerical format. Thus, routing a vehicle to Node 8 is equivalent to routing the vehicle to Krugersdorp. The logic behind the addition of the new demand rows are as follows:

```
If (Demand) > 8
        then    Demand1=8;
        else    Demand1=Demand;
end
```

The value of **Demand1** is used as the input demand because it is always less than eight (8) (adhering to the constraint in demand) or equal to the user input demand.

Table 1: Glossary of algoritm variables

Term	Description
VCAP	The vehicle capacity
VFLEET	The number of vehicles in the fleet
Distance	Distance to or between nodes
Demand	Demand at a node
Route	Matrix containing viable vehicle routes
x	Matrix of vehicles being routed
Cdem	Cumulative demand of a route
TotalDistanceOld	Cumulative distance traveled by fleet (Old system routing, AS-IS)
VF	Solution: Number of Vehicles routed
VC	Solution: Needed vehicle capacity
row	The row size of **Route**
col	The column size of **Route**
Last	Last node of a route
TotalDistance	Cumulative distance traveled by fleet (Solution, TO-BE)
DistanceSavings	Improvement or saving obtain by algorithm
BestRoute	The route plan which results in the minimum fleet travel distance
BestSavings	The distance saved when using the optimum route plan
counter	Counts the number of times the program is initiated
countrow	Counts the number of non-zero rows in the route matrix

3.1.2 MATLAB data input

The relevant data from Excel is imported into MATLAB for the necessary computations to occur.

This is done by using the built-in *xlsread* statement:

```
VCAP=xlsread('TOMB',2,'VCAP');
VFLEET=xlsread('TOMB',2,'VFLEET');
Distance = xlsread('TOMB',2,'DIST');
Demand=xlsread('TOMB',2,'Demand1');
x=(zeros(16,16));
Route=(zeros(VFLEET,VCAP));
Cdem=0;
TotalDistanceOld=0;
```

The extract describes how data is imported from the Excel. The name of the spreadsheet is 'TOMB', while data is imported from sheet two (2). A search for the cells named VCAP, VFLEET, DIST and Demand1 is conducted and assigned to the appropriate variable.

3.2 Algorithm logic

The following section will discuss the model logic in various sections.

3.2.1 Modeling the AS-IS state of the route plan

In order to compare the improvement made by the algorithm when compared to the old routing system or the AS-IS state of routing, the state must first be modeled.

The extract routes vehicles to nodes with a demand greater than zero and then back to the depot. The total distance traveled by the fleet is assigned to the variable *TotalDistanceOld*.

```
for i=1:16
   if Demand(i)>0
          x(1,i)=1;
          x(i,1)=1;
          for j=1:16
              TotalDistanceOld=TotalDistanceOld+x(i,j)*Distance(i,j);
          end
   end
end
```

3.2.2 Route creation

In order for vehicles to be assigned to routes, feasible routes must first be determined. The algorithm extract begins by randomly selecting a node. Afterwards, testing is done to determine whether a vehicle still has the sufficient capacity to serve the next node. If this is viable, the node is added as a new stop in a route. The second part of the extract removes null values (e.g. 0 2 0 3 0 4 becomes 234).

```
for j=1:16
   for i=1:16
        i=randint(1,1,[1,16]);
        if (Cdem+Demand(i)<=VCAP)&&(Demand(i)>0)&&(i~=j)
            Route(j,i)=i;
            Cdem=Cdem+Demand(i);
            Demand(i)=0;
        end
   end
   Cdem=0;
end
```

```
[VF,VC]=size(Route);
for k=1:VF
   j=1;
   for i=1:VC
       if Route(k,i)>0
           Route(k,j)=i;
           Route(k,i)=0;
           j=j+1;
       end
   end
end
```

3.2.3 Route assignment

After the creation of routes, vehicles are assigned to serve a route. The non-zero rows of the route matrix represent a route assigned to a single vehicle. Thus, if it is calculated that there are more than five (5) non-zero rows in the solution, the solution violates a constraint and the route is reset. This is done by the '**countrow'** function. In an effort to explain the data contained in the route matrix, please refer to the following example:

Route=

3	5	7	8
11	12	14	0
13	0	0	0
15	16	0	0

Four vehicles are required to respectively complete the following routes:

- ✓ 1 -3, 3-5 ,5-7 ,7-8 ,8-1
- ✓ 1-11, 11-12, 12-14, 14-1
- ✓ 1-13, 13-1
- ✓ 1-15, 15-16, 16-1

The node numbers relate to locations found in the Excel spreadsheet.

```
[VF,VC]=size(Route);
Last=8;
for k=1:VF
  if Route(k)>0
   for i=1:8
     x(1,Route(k))=1;
     while Route(k,(i))>0&&Route(k,(i+1))>0
       x((Route(k,(i))),(Route(k,(i+1))))=1;
       i=i+1;
     end
     for j=1:8
       while Route(k,j)>0
         Last=Route(k,j);
         j=j+1;
       end
     end
     x(Last,1)=1;
   end
  end
```

```
end
countrow=0;
for i=1:VF
    if Route(i,1)>0
        countrow=countrow+1;
    end
end
if countrow>5
    x=(zeros(16,16));
end
```

3.2.4 Solution

The algorithm concludes by providing a solution to the given problem. The cumulative distance traveled by the fleet is calculated and the compared to the old routing system in the form of a *Distance Savings* variable. A counter determines how many times the algorithm will search for a better solution. If a solution improves on the previous solutions, it is regarded as the optimal solution.

```
for i=1:16
    for j=1:16
        TotalDistance=TotalDistance+Distance(i,j)*x(i,j);
    end
end
DistanceSaving=TotalDistanceOld-TotalDistance;
if (DistanceSaving>BestSavings)&&(TotalDistance>0)
    BestSavings=DistanceSaving;
    BestRoute=Route;
end
```

Chapter 4

Computational Evaluation

Several instances will be tested in this section to determine whether the model provides an optimum or near optimum solution.

Three solutions to a problem will be given and compared:

- ✓ The current routing method used by *Tombstone Land.*
- ✓ An initial solution.
- ✓ The iterative solution provided by the developed program.

4.1 Direct routing

The method currently being used by *Tombstone Land* is that of direct routing. This method routes a vehicle to one demand point and back again. The first column of table 5 shows this distance calculated for the 20 different simulations of demand. It became apparent that this is a far from optimum way of routing the fleet.

4.2 The initial solution

The program used for the initial solution to the VRP is an adaptation of the main program. The two main differences is that the program only finds the first feasible solution and that it starts with the first demand point with a demand larger than zero. Taking less than one minute to solve, the computation time for the initial solution is far less than that of the iterative program.

4.3 The iterative solution

As noted in the previous paragraph, the iterative program is fairly similar to that of the initial solution. However, the computational time is a function of the amount of iterations the user wishes to run. Running about 500 iterations takes about 25 minutes and provides an average improvement of 6.04% when compared to the initial solution and an average improvement of 38.37% when compared to the direct routing solution.

4.4 Demand simulation and results

In order to test the model, demand was simulated for the various nodes (Refer to Table 4).

Using the simulated demand as input, all three methods of routing was tested and the total distance cover by the fleet was recorded in Table 5. In all 20 instances the initial solution was better than that of the direct routing.

The three methods were compared and the results documented in Table 6.

On average, the initial solution provided an improvement of 34.34% when compared to that of direct routing and the iterative solution provided (on average) a further improvement of 6.04%.

Chapter 5

Conclusion

Although some assumptions made in this model may conflict with 'real world' situations, the findings speak for themselves. The manner in which *Tombstone land* route their vehicles are far from optimal. The point is not to prove them wrong but to highlight an area with much potential for improvement. This solution is tailor made for the business and provides them with an improved method of routing their vehicles. The main goal behind the exercise is to point out the importance transportation plays in a business and how the mismanagement of this facet of business can severely impact the growth and prosperity of the business.

List of figures

Figure 3: Operations Research Process

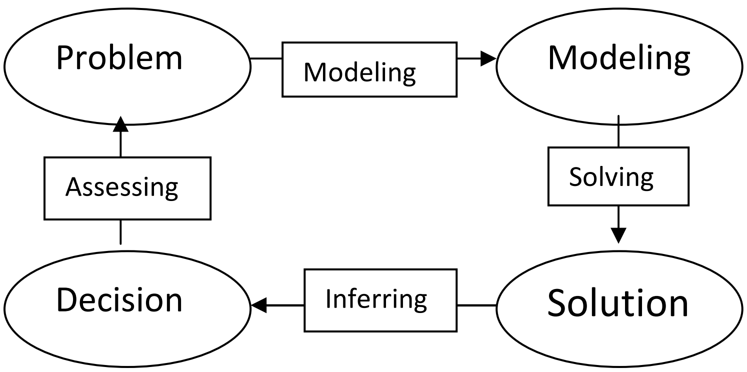


Figure 4: AS-IS state of Vehicle Routing Plan for *Tombstone Land*

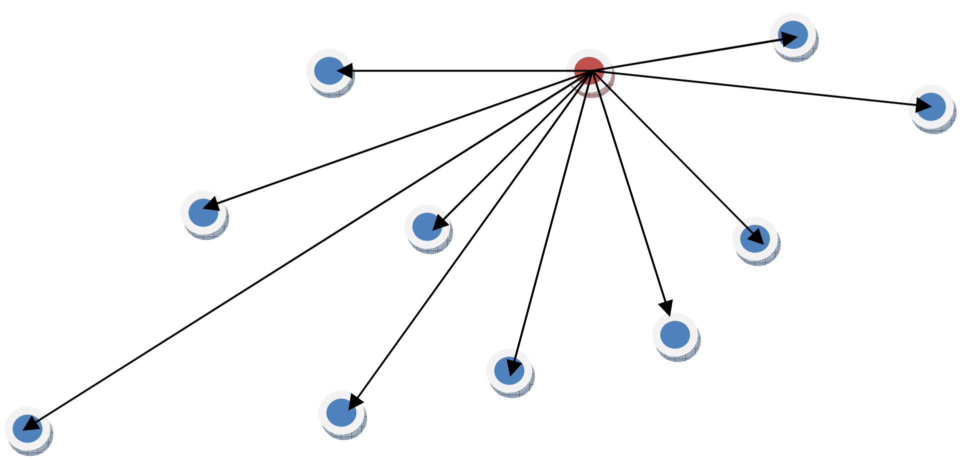


Figure 5: Proposed TO-BE state of the Vehicle routing Plan

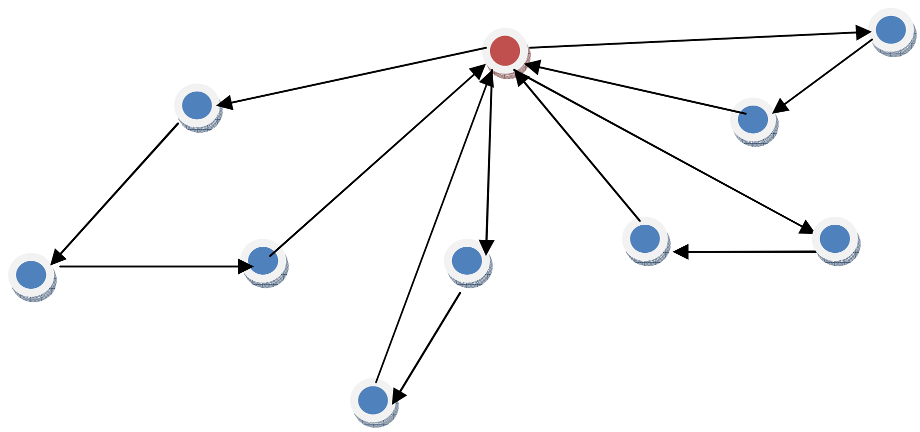


Figure 6 : Input spreadsheet

	Bronkhorstspruit	Bloedrivier	Carolina	Elukwatini	Everton	Hammanskraal	Hebron	Krugersdorp	Moloto	Nigel	Nqutu	Pumulani	Soshanguve	Tembisa	Tsakane	Witbank	
Bronkhorstspruit	0	417	370	345	160	200	450	92	220	120	380	180	180	130	115	280	
Bloedrivier	417	0	50	365	475	485	320	450	450	355	65	455	480	416	370	285	
Carolina	370	50	0	410	218	210	230	262	250	338	633	250	210	275	325	430	
Elukwatini	345	365	410	0	256	225	310	290	175	225	300	210	245	230	230	100	
Everton	160	475	218	256	0	51	291	67	85	120	420	46	26	60	110	235	
Hammanskraal	200	485	210	225	51	0	250	113	50	141	422	30	24	83	130	222	
Hebron	450	320	230	310	291	250	0	350	245	378	600	272	266	328	370	377	
Krugersdorp	92	450	262	290	67	113	350	0	140	97	410	96	92	60	85	250	
Moloto	220	450	250	175	85	50	245	140	0	135	390	90	66	92	130	182	
Nigel	120	355	338	225	120	141	378	97	135	0	305	112	133	65	13	160	
Nqutu	380	65	633	300	420	422	600	410	390	305	0	400	425	365	320	221	
Pumulani	180	455	250	210	46	30	272	96	90	112	400	0	35	53	102	202	
Soshanguve	180	480	210	245	26	24	266	92	66	133	425	35	0	71	121	232	
Tembisa	130	416	275	230	60	83	328	60	92	65	365	53	71	0	50	193	
Tsakane	115	370	325	230	110	130	370	85	130	13	320	102	121	50	0	170	
Witbank	280	285	430	100	235	222	377	250	182	160	221	202	232	193	170	0	
Demand	0	0	2	0	4	0	1	1	0	0	2	3	7	1	5	3	29
Demand1	0	0	2	0	4	0	1	1	0	0	2	3	7	1	5	3	
Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	Bronkhorstspruit	**Bloedrivier**	**Carolina**	**Elukwatini**	**Everton**	**Hammanskraal**	**Hebron**	**Krugersdorp**	**Moloto**	**Nigel**	**Nqutu**	**Pumulani**	**Soshanguve**	**Tembisa**	**Tsakane**	**Witbank**	
Bronkhorstspruit	0	0	370	0	160	0	450	92	0	0	380	180	180	130	115	280	
Bloedrivier	0	0	50	0	475	0	320	450	0	0	65	455	480	416	370	285	
Carolina	370	50	0	0	218	0	230	262	0	0	633	250	210	275	325	430	
Elukwatini	0	0	0	0	256	0	310	290	0	0	300	210	245	230	230	100	
Everton	160	475	218	256	0	0	291	67	0	0	420	46	26	60	110	235	
Hammanskraal	0	0	0	0	0	0	250	113	0	0	422	30	24	83	130	222	
Hebron	450	320	230	310	291	250	0	350	0	0	600	272	266	328	370	377	
Krugersdorp	92	450	262	290	67	113	350	0	0	0	410	96	92	60	85	250	
Moloto	0	0	0	0	0	0	0	0	0	0	390	90	66	92	130	182	
Nigel	0	0	0	0	0	0	0	0	0	0	305	112	133	65	13	160	
Nqutu	380	65	633	300	420	422	600	410	390	305	0	400	425	365	320	221	
Pumulani	180	455	250	210	46	30	272	96	90	112	400	0	35	53	102	202	
Soshanguve	180	480	210	245	26	24	266	92	66	133	425	35	0	71	121	232	
Tembisa	130	416	275	230	60	83	328	60	92	65	365	53	71	0	50	193	
Tsakane	115	370	325	230	110	130	370	85	130	13	320	102	121	50	0	170	
Witbank	280	285	430	100	235	222	377	250	182	160	221	202	232	193	170	0	
	VCAP	8															
	VFLEET	5															

Figure 7: Initial solution output (MATLAB)

```
Command Window
>> TombInt

TotalDistanceOld =

        5028

The total distance traveld by the fleet when routed according to
the following route plan:

Route =

     2     4     5    11     0     0     0     0     0     0     0     0     0     0
     6     8    14     0     0     0     0     0     0     0     0     0     0     0
     7     9     0     0     0     0     0     0     0     0     0     0     0     0
    10     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0

is

TotalDistance =

        3496


DistanceSaving =

        1532

>> |
```

Figure 8: Iterative solution (MATLAB)

```
Command Window
The total distance traveled by the fleet,
when routed directly to and from demand points is:

TotalDistanceOld =

        5028

The total distance traveled by the fleet when routed according to
the following route plan:

BestRoute =

     5     7     0     0     0     0     0     0     0     0     0     0     0     0
     4     6     8     0     0     0     0     0     0     0     0     0     0     0
     2    11    14     0     0     0     0     0     0     0     0     0     0     0
     9    10     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0

is

BestTotalDistance =

        3128

The amount of distance saved is:

BestSavings =

        1900
```

List of Tables

Table 2: n x n symmetric distance matrix

	Bloedrivier	Bronkhorstspruit	Carolina	Elukwatini	Everton	Hammanskraal	Hebron	Krugersdorp	Moloto	Nigel	Nqutu	Pumulani	Soshanguve	Tembisa	Tsakane	Witbank
Bloedrivier	0	417	50	365	475	485	320	450	450	355	65	455	480	416	370	285
Bronkhorstspruit	417	0	370	345	160	200	450	92	220	120	380	180	180	130	115	280
Carolina	50	370	0	410	218	210	230	262	250	338	633	250	210	275	325	430
Elukwatini	365	345	410	0	256	225	310	290	175	225	300	210	245	230	230	100
Everton	475	160	218	256	0	51	291	67	85	120	420	46	26	60	110	235
Hammanskraal	485	200	210	225	51	0	250	113	50	141	422	30	24	83	130	222
Hebron	320	450	230	310	291	250	0	350	245	378	600	272	266	328	370	377
Krugersdorp	450	92	262	290	67	113	350	0	140	97	410	96	92	60	85	250
Moloto	450	220	250	175	85	50	245	140	0	135	390	90	66	92	130	182
Nigel	355	120	338	225	120	141	378	97	135	0	305	112	133	65	13	160
Nqutu	65	380	633	300	420	422	600	410	390	305	0	400	425	365	320	221
Pumulani	455	180	250	210	46	30	272	96	90	112	400	0	35	53	102	202
Soshanguve	480	180	210	245	26	24	266	92	66	133	425	35	0	71	121	232
Tembisa	416	130	275	230	60	83	328	60	92	65	365	53	71	0	50	193
Tsakane	370	115	325	230	110	130	370	85	130	13	320	102	121	50	0	170
Witbank	285	280	430	100	235	222	377	250	182	160	221	202	232	193	170	0

Table 3: n x n symmetric cost matrix

	Bloedrivier	Bronkhorstspruit	Carolina	Elukwatini	Everton	Hammanskraal	Hebron	Krugersdorp	Moloto	Nigel	Nqutu	Pumulani	Soshanguve	Tembisa	Tsakane	Witbank
Bloedrivier	0	1387	166	1214	1580	1613	1064	1497	1497	1181	216	1513	1596	1384	1231	948
Bronkhorstspruit	1387	0	1231	1147	532	665	1497	306	732	399	1264	599	599	432	382	931
Carolina	166	1231	0	1364	725	698	765	871	832	1124	2105	832	698	915	1081	1430
Elukwatini	1214	1147	1364	0	851	748	1031	965	582	748	998	698	815	765	765	333
Everton	1580	532	725	851	0	170	968	223	283	399	1397	153	86	200	366	782
Hammanskraal	1613	665	698	748	170	0	832	376	166	469	1404	100	80	276	432	738
Hebron	1064	1497	765	1031	968	832	0	1164	815	1257	1996	905	885	1091	1231	1254
Krugersdorp	1497	306	871	965	223	376	1164	0	466	323	1364	319	306	200	283	832
Moloto	1497	732	832	582	283	166	815	466	0	449	1297	299	220	306	432	605
Nigel	1181	399	1124	748	399	469	1257	323	449	0	1014	373	442	216	43	532
Nqutu	216	1264	2105	998	1397	1404	1996	1364	1297	1014	0	1330	1414	1214	1064	735
Pumulani	1513	599	832	698	153	100	905	319	299	373	1330	0	116	176	339	672
Soshanguve	1596	599	698	815	86	80	885	306	220	442	1414	116	0	236	402	772
Tembisa	1384	432	915	765	200	276	1091	200	306	216	1214	176	236	0	166	642
Tsakane	1231	382	1081	765	366	432	1231	283	432	43	1064	339	402	166	0	565
Witbank	948	931	1430	333	782	738	1254	832	605	532	735	672	772	642	565	0

Table 4: Demand Simulation

						Demand Simulation										
	Bronkhorstspruit	Bloedrivier	Carolina	Elukwatini	Everton	Hammanskraal	Hebron	Krugersdorp	Moloto	Nigel	Nqutu	Pumulani	Soshanguve	Tembisa	Tsakane	Witbank
Simulation 1	0	4	2	5	3	1	3	2	3	2	0	1	4	0	0	5
Simulation 2	0	2	1	5	4	4	3	3	0	4	4	5	0	1	2	2
Simulation 3	0	4	0	0	1	2	0	0	3	3	3	8	0	1	4	4
Simulation 4	0	1	0	6	5	0	0	0	1	2	1	1	3	3	0	0
Simulation 5	0	8	8	8	0	0	0	0	2	3	0	0	6	5	0	0
Simulation 6	0	0	0	3	7	0	1	0	4	4	0	0	0	5	3	5
Simulation 7	0	2	0	5	0	5	0	0	2	1	0	3	5	5	3	5
Simulation 8	0	1	1	2	1	3	1	2	0	5	3	2	4	1	2	2
Simulation 9	0	0	3	2	5	1	4	0	3	2	3	1	5	5	0	5
Simulation 10	0	4	4	0	3	1	5	6	1	3	1	0	1	2	4	4
Simulation 11	0	0	3	0	3	3	1	2	1	5	0	1	5	2	1	4
Simulation 12	0	3	3	0	2	5	1	0	3	1	2	0	1	5	5	2
Simulation 13	0	0	0	6	4	3	1	1	0	5	5	5	5	0	3	2
Simulation 14	0	3	0	1	0	5	4	3	1	0	1	3	5	3	1	2
Simulation 15	0	3	3	2	5	1	2	4	0	3	3	2	5	0	0	1
Simulation 16	0	0	2	4	0	1	2	0	2	3	4	5	2	0	5	5
Simulation 17	0	0	0	2	3	2	4	3	0	5	3	1	6	1	3	3
Simulation 18	0	8	2	0	0	1	1	3	5	0	0	6	5	3	2	4
Simulation 19	0	0	4	1	0	5	0	4	0	2	5	4	3	4	3	2
Simulation 20	0	2	0	2	3	4	5	2	3	5	1	0	0	1	0	0

Table 5: Simulation results

	AS-IS	Initial	Iterative
Simulation 1	6028	3801	3692
Simulation 2	6478	3825	3788
Simulation 3	4404	3366	2510
Simulation 4	4264	2558	2313
Simulation 5	3564	3099	3045
Simulation 6	3640	2508	2396
Simulation 7	4374	2847	2817
Simulation 8	6838	4091	3400
Simulation 9	6030	3981	3607
Simulation 10	6228	3855	3576
Simulation 11	4994	3104	2490
Simulation 12	6044	3587	2988
Simulation 13	5004	2997	0
Simulation 14	5978	3891	3395
Simulation 15	6348	4082	3634
Simulation 16	5680	3730	3460
Simulation 17	5264	3582	3015
Simulation 18	5268	3305	3416
Simulation 19	4784	3315	2941
Simulation 20	5028	3496	3074

Table 6: Improvement per method

	AS-IS/Initial	AS-IS/Iterative	Initial/Iterative
	36.94	38.75	1.81
	40.95	41.53	0.57
	23.57	43.01	19.44
	40.01	45.76	5.75
	13.05	14.56	1.52
	31.10	34.18	3.08
	34.91	35.60	0.69
	40.17	50.28	10.11
	33.98	40.18	6.20
	38.10	42.58	4.48
	37.85	50.14	12.29
	40.65	50.56	9.91
	40.11	0.00	0.00
	34.91	43.21	8.30
	35.70	42.75	7.06
	34.33	39.08	4.75
	31.95	42.72	10.77
	37.26	35.16	-2.11
	30.71	38.52	7.82
	30.47	38.86	8.39
Average	34.34	38.37	6.04

Bibliography

Joubert, JW 2004, 'An initial solution heuristic for the vehicle routing and scheduling problem', Master's Dissertation, Industrial and Systems Engineering, University of Preoria, etd-08272007-161812, University of Preoria, Pretoria.

Laporte, G 1992, 'The vehicle routing problem: An overview of exact and approximate algorithms', *European Journal of Operational Research*, no. 59, pp. 345-358.

Pisinger, D & Ropke, S 2007, 'A general heuristic for vehicle routing problems', *Computers & Operations Research*, no. 34, p. 2403 – 2435.

Rardin, R 1998, *Optimization in Operations Research*, Prentice Hall, New Jersey.

Raza, SA & Al-Turki, UM 2007, 'A comparative study of heuristic algorithms to solve maintenance scheduling problem', *Journal of Quality in Maintenance Engineering*, vol 13, no. 4, pp. 398-410.

Stewart, J 2004, *Calculus Early Transcendentals*, 5th edn, Tomson, Brooks; Tomson, Cole.

Van Breedam, A 2001, 'Comparing decent heuristics and metaheuristics for the vehicle routing problem', *Computers and Operations research*, no. 28, pp. 289-315.

Winston, WL & Venkataramanan, M 2004, *Introduction to mathematical programming*, 4th edn, Hinrichs, Curt.