DIPLOMARBEIT

Titel der Diplomarbeit

Ring Star Problem for Disaster Relief

Verfasser

Göran Sjöström

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften
(Mag. rer. soc. oec)

Wien, November 2011

Studienkennzahl lt. Studienblatt:	A 157
Studienrichtung lt. Studienblatt:	Internationale Betriebswirtschaft
Betreuer:	O.Univ.Prof. Dipl.-Ing. Dr. Richard F. Hartl

ACKNOWLEDGEMENT

First of all I like to express my gratitude to the staff of the Chair of Production and Operations Management at the University of Vienna for their great support.

Especially I would like to extend my thanks to Mag. Dipl.-Ing. Dr. Verena Schmid for her patience and support by frequent questions. Additionally great thanks to Mag. Attila Kovacs for his creativity and input when help was needed, especially during the implementation of the C++ code.

My gratitude goes as well to my parents Harriet and Anders Sjöström and my brother Daniel for supporting me to start my studies after 4 years in the industry and their steady support during these years.

Vienna, November 2011

Contents

List of Figures

List of Tables

Abstract

In disaster situations, quick relief for the affected population is the key. Ring Star Problem (RSP) can be used to model problems related to Disaster Relief. The idea of RSP lies in the assumption that in post-disaster not all of the affected population can be supplied directly with aid. Therefore they would need to travel to the closest location where help can been supplied. For solving the problem several approaches were implemented. An integrative exact weighted sum approach and as well two step methods. Additionally different Large Neighbourhood Search (LNS) approaches were created. The data was then evaluated by comparing the achieved results of costs and the needed computing time for achieving results. Instances up to 100 nodes could be solved to optimality with the integrative weighted sum approach. The LNS solved up to 100 nodes efficiently.

1 Introduction

Disasters occur and hit mankind since the beginning of time. For many centuries, when disasters hit - both man-made and natural - people were left on their own. In modern time this ought not to happen. Especially with modern technologies of communication and transportation the disaster relief has the opportunity to react swiftly. In spite of this, even today it is not always possible to help quickly. When taking into consideration natural disasters of the last decade - tsunami in South-East Asia, hurricane over New Orleans/USA, flooding in Pakistan, earthquake in Haiti, earthquake and tsunami in Japan - quick reaction for helping in the affected regions was the most important issue. From first-aid, medical care, food deliveries to building of wells - the logistic behind it has been the key for quick relief.

When disasters hit countries like Haiti, which already before the earthquake had been one of the poorest countries worldwide, it was clearly shown how devastating the effect is when the little existing infrastructure gets destroyed. Thus victims were completely dependent on quick international help.

But even in the more industrialized nations, natural disasters demonstrate dramatically the limits of the existing infrastructures. The hurricane Katrina in 2005, which hit most of the southern parts of the United States of America and especially New Orleans, gave a clear picture of the limitation that the people had to help themselves. Even USA, what most people would call "developed country", had enormous problems in organizing an immediate and efficient aid-program for the affected region.

During 2011 the earthquake and the following devastating tsunami which hit the east coast of Japan, and the starvation disaster in East Africa, demonstrated the idea of the importance of rapid international help in supplying goods.

Especially the situation in Somalia illustrates the difficulties of providing rapid help assistance. In this case, the country government had restricted sphere of influence. So the government is not in the position to guarantee and ensure the coordination of the NGOs aid projects. Moreover the aid organizations must rely on the local war lords. In spite of this, studies illustrate that, for the people itself it works better that way, then with an unstable government (T. and Leeson [2007]).

Despite that political situation in Somalia, the almost non existing infrastructure was and still is another big challenge. The worst dry season since over 60 years and the war-related circumstances and constraints to access of productive resources hit the region hard (Sietz et al. [2011]). As a result whole villages were not having enough to eat and drink. Cattle's were starving and died out of thirst. Plants were dying because of lack of water and in result the people were starting to starve. As a result large groups of people were fleeing from their acreage into the cities in need for help.

On this kind of situation we want to apply our Ring Star Problem (RSP). It is a location allocation problem. The aim is to locate a tour through a subset of vertices of a graph with the goal of minimizing two different costs: the costs of length of the tour between nodes, and the costs of assigning the non-visited to the closest visited nodes.

But what does it mean when applied on a real world problem? Let us take the situation in Somalia into consideration, with many small villages and very little, to non existing infrastructure. When help organizations are supplying food and water it needs to be delivered to as many people as possible. The goal is to get aid to the whole population. In this detail lies the problem. Many small villages are not reachable by big trucks. The aim is to find the best and closest spots to these villages, that can be visited by the trucks. To these hubs it must be possible for the people of the surrounding villages and towns to reach with their limit resources so that their survival can be guaranteed.

In this matter we do have a truck, which leaves the depot with the urgently needed supplies. The route is being organized in such a manner that the villages/hubs visited are chosen in such a way that the population of the remaining villages can travel to these points for aid with the limited resources available

Lets take the example of the emergency service into consideration. The locations of the service spots are chosen so that the people in the nearby area will not have any problems to reach it in case of an emergency. Either within a certain limit of time, within a certain amount of travelling distance or at a certain limit of costs.

For these problems two objectives have to be taken into consideration in planning the deliveries. First the villages/cities (hubs) have to be selected, which the trucks are able to reach. The surrounding non-visited villages are then assigned to the closest hub with the assignment costs of "distance multiplied with population". This will be done with the p-Median; explained in detail in Section (4.2).

Then the route of the vehicle has to be planned to optimality with the help of the *Travelling Salesman Problem* (TSP), with just the costs of "distance between hubs"; explained in detail in Section (4.3).

Since the assignment costs are very large compared to the ring costs of the classical TSP it is a *NP-hard* problem.

2 Literature Overview

As Labbé et al. [1995] introduced it as part of the problem locating one or several facilities with regard to existing one in order to optimize some economic criterion. The first location model was mentioned by Alfred Weber in a book published 1909. Until 1960 no significant progress occurred. Then facility location analysis became an active field of research.

Labbé et al. [2004] where the first who clearly introduced the RSP with their polyhedral analysis and exact algorithm, which is an extension of their former works. There they gave the clear objective of minimizing two kind of costs - ring costs and assignment costs.

RSP is applied in the field of telecommunication. A network of terminals or concentrators are installed and connected as a ring and the other customer locations assigned to these concentrators. For detailed informations refer to "An exact algorithm for solving the Ring Star Problem" by Kedad-Sidhoum and Nguyen [2010].

On other real world problems, this method is being applied as well. Like planning of public transport where the decision has to be made where all the stops have to be created so that they are in a reasonable distance for potential users (Labbé et al. [2005]).

A problem related to RSP is the *Median Cycle Problems (MCP)*, where the objective is to minimize the routing cost, but subject to an upper bound on the total assignment. It was the first time formulated and solved with an exact branch and bound method and with a heuristic procedure by Labbé et al. [1999]. In further studies this method is used and applied to find solutions in real world problems. The design of ring-shaped infrastructures such as circular metro-lines and motorways are examples. The target is to minimize the travel cost of the vehicle, while bounding the average accessibility cost for non-directly served population. The stops correspond to metro-stations and by the motorway to junctions with smaller streets. Check the paper of Labbé et al. [2005] for additional information.

The *Bi-Objective Ring Star Problem* is another adaptation of the basic problem. Liefooghe et al. [2008] and Liefooghe et al. [2010] are two papers especially assigned to this topic. The difference to the classic RSP is the main objective in minimizing the costs of the cycle/ring of visited nodes and additionally the minimization of the assignment costs as two separate corresponding costs. In our classical RSP model these two objectives are taken into one weighted sum approach and formed into one objective.

Doerner et al. [2007] made an adaptation of the RSP and Bi-Objective RSP for the tour planning of mobile healthcare facilities. The target was to select locations for a tour, from which the people in need where in a defined reasonable distance to the location chosen, so that they could be helped. For this special problem, additional criteria where added like certain percentage coverage of the population. This problem was a *Multi-objective Optimization Problem (MOP)* as a set where more then two objective functions were optimized.

Some additional variations of the RSP were created for the practical usage. One of it, is the *m-Ring Star Problem.* In this method more than one tour is taken into consideration. This variation is of interest in the telecommunication. A further adaptation is the *Capacitated m-Ring Star Problem* analysed by Mauttone et al.

[2008] . Here the additional constraint exist in a limited amount of Q customers per cycle.

The *Steiner Ring Star Problem* is another variation created by Lee et al. [1996]. The aim is to find a minimum cost cycle while only considering certain fix defined nodes (Steiner nodes), to which the customers must be connected (each customer to only one node)

When trying to achieve the optimal results in the coming pages, we implemented some meta-heuristic approaches which have been studied since the early 1980´s. As Osman and Kelly [1996] mentioned in his book *Meta-Heuristic: Theory & Applications*, meta-heuristics are designed to attack hard combinatorial optimization problems where classical heuristics have failed to be effective and efficient.

Especially the method of the *Large Neighbourhood Search* (LNS) introduced by Shaw [1998], on which our implementation is based, has to be mentioned. The idea behind the LNS is the gradually destroying and repairing of a solution to improve the final results. As it was one of the first ideas, this meta-heuristic was just working by accepting improving solutions.

Further adaptation appeared, especially the one by Schrimpf et al. [2000], where a temporary worse solution was accepted to improve the performance of searching the neighbourhood for reaching global optimum.

3 Problem Description and Definition

For the description of the classic RSP we apply the explanation mentioned by Liefooghe et al. [2010].

The RSP model aims at minimizing the sum of the assignment costs of arcs directed from every non-visited node to a visited and the costs of the routing of the visited nodes.

This model can as well be split into two partial problems. The assignment will be modelled with the p-Median as described in Section (4.2). On the other hand the minimization of the sum of the touring cost between the visited nodes, calculated with the TSP, which will be explained in Section (4.3).

In this section we introduce the weighted sum approach which represents a combination of the two sub problems into one single model.

Taking the RSP into consideration, it is a NP-hard combinatorial problem since the particular case of visiting the whole set of nodes is equivalent to a traditional TSP (Liefooghe et al. [2008]). This means that for each additional location in the problem set, the solving time for the exact method rises exponential.

NP stands for non-deterministic polynomial. So the problem class of NP can be solved in polynomial time in a non-deterministic turning machines (Chapter 11 in Burke and Kendall [2005]). For this reason different heuristics for solving big instances are used (please see in the Chapter 5 for more details about the heuristics used). These methods are solving small problem sets to optimality, and big problem sets close to optimality, but to a much lower computing time compared to the exact mathematical method.

3.1 The Model

Let $G = (V, E, A)$ be a complete mixed graph where $V = \{v_0, v_1, ..., v_n\}$ is a set of vertices. $E = \{[v_i, v_j] | v_i, v_j \in V, i < j\}$ is a set of edges, and $A = \{[v_i, v_j] | v_i, v_j \in V \setminus \{v_0\}\}$ is a set of arcs. Vertex v_0 is the depot. Each edge has a non-negative distance cost d_{ij}, and each arc $(v_j, v_i) \in A$ we assign some assignment cost d_{ji}. Each vertex v_j has a demand in term of its population pop_j, except v_0 where $pop_j = 0$. The total amount of vertices to be visited along the resulting route is devoted as p.

x_{ij} is a binary decision variable, which will evaluate to 1 if and only if edge $[v_i, v_j]$ is part of the resulting tour.

z_{ji} is another binary decision variable. It defines that vertex v_j is assigned to v_i if $z_{ji} = 1$, and not assigned if $z_{ji} = 0$. In this matter binary variable y_i models if vertex v_i is on route if $y_i = 1$ and jf variable $y_i = 0$ it declares that vertex v_i is not routed

We define variable u_i as the rank of routed vertex v_i to avoid sub-cycles.

3.2 1-Step Exact

For the weighted sum approach as the 1-Step Combined version we did us following model:.

$$\sum_{j=1}^{n} pop_j \sum_{i=1}^{n} d_{ji} \cdot z_{ji} + \sum_{k=0}^{n} \sum_{l=0}^{n} d_{kl} \cdot x_{kl} \ \rightarrow \min \tag{1}$$

$$s.t. \sum_{i=1}^{n} z_{ji} = 1 \qquad \forall\ j = 1, \ldots, n \tag{2}$$

$$y_i - z_{ji} \geq 0 \qquad \forall\ i = 1, ...n,\ j = 1, \ldots, n \tag{3}$$

$$\sum_{i=0}^{n} y_i = p \qquad \forall\ i = 0, \ldots, n \tag{4}$$

$$\sum_{\substack{l=0\\ l\neq k}}^{n} x_{kl} = y_k \qquad \forall\ k = 0, \ldots, n \tag{5}$$

$$\sum_{\substack{k=0\\ l\neq k}}^{n} x_{kl} = y_l \qquad \forall\ l = 0, \ldots, n \tag{6}$$

$$u_k - u_l + p \cdot x_{kl} \leq p - 1 \qquad \forall\ k, l \in V, k \neq l \tag{7}$$

$$z_{ji} \in \{0, 1\} \qquad \forall\ i = 1, \ldots, n,\ j = 1, \ldots, n \tag{8}$$

$$y_i \in \{0, 1\} \qquad \forall\ i = 1, \ldots, n \tag{9}$$

$$x_{kl} \in \{0, 1\} \qquad \forall\ k = 0, \ldots, n,\ l = 0, \ldots, n \tag{10}$$

The objective function (1) minimizes the sum of the weighted distance travelled by assigned population and the distance travelled in routing.

Constraint (2) ensures that each location j has to be assigned to exact one visited node i. But more then one location j can be assigned to hub i.

Constraint (3) ensures that each location j may only be assigned to an opened hub y_i. Constraint (4) sums up all the opened locations to guarantee that exactly p locations are opened.

Constraints (5) as the in-degree constraint, and (6) as the out-degree constraint, show that each location on tour may only be entered and left once. Non of these locations are allowed to be visited twice.

To avoid sub-tours we add additional constraints (7). It ensures that if edge $[v_k, v_l]$ is on tour, the rank of u_k is lower then rank of location u_l.

Constraints (8), (9) and (10) state that variables z_{ji}, y_i and x_{kl} are binary.

3.3 Solution Representation

In Figure 1 a possible solution of a RSP is shown. Let the depot v_0 be the whole black location. It's the start- and end location for the ring tour of the opened locations. The non opened are being assigned towards the closest opened location. Under the condition that non is being assigned directly to the depot

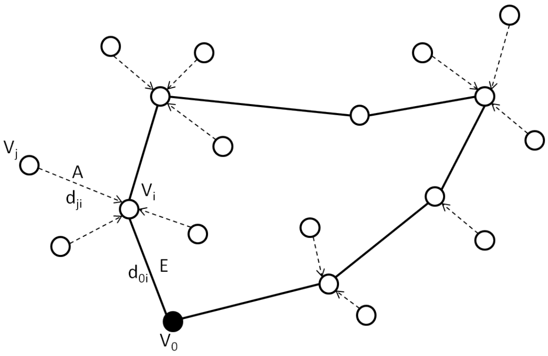


Figure 1: Example for the RSP

4 Solution Approaches

4.1 Overview

We tried to solve the problem stated in Section 3.2 with the help of CPLEX. Throughout the rest of this thesis we will refer to "1-Step Exact" for this integrated weighted sum approach.

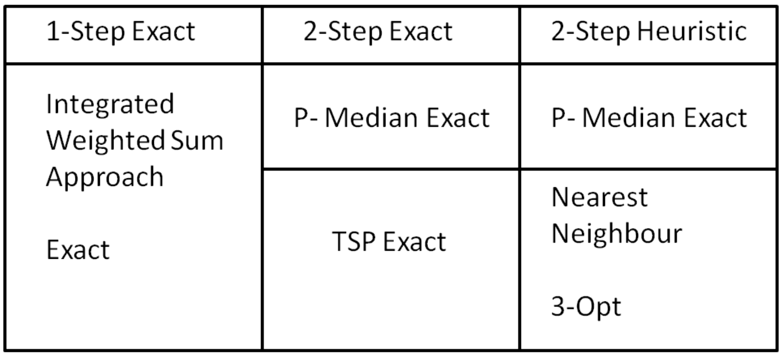
For comparison reason we implemented additional approaches. So we could analyse the effects with respect to the resulting solution quality and the run-time required for solving.

As a kind of solution approach we separated the model into the two known sub problems (for details please see Section 4.2 and 4.3). The first sub-problem (p-Median) is solved by using CPLEX.

The second sub-problem, the routing, we then used two different approaches, with the p-Median solution as the starting point. In our first trial we solved the routing part as well exact by using the TSP-Model in CPLEX (2-Step Exact).

The second method we solved by using a heuristic. First the initial solution of the exact p-Median was routed with the nearest neighbour and then the improvement heuristic of 3-opt was applied for improving the routing itself (2-Step Heuristic).

In following figure we illustrate these three approaches and their composition.



1-Step Exact	2-Step Exact	2-Step Heuristic
Integrated Weighted Sum Approach Exact	P- Median Exact	P- Median Exact
	TSP Exact	Nearest Neighbour 3-Opt

Figure 2: Solution Approaches 1-Step Exact and 2-Step with Exact Initial Solution

For improving results in matter of computing time we then implemented some additional approaches, only based on heuristics. These methods are explained in detail in the Chapter 5.

4.2 P-Median

The P-Median problem has been studied in many fields of locations problems. It's a rather basic model, where a fixed number of locations p are being selected as a hub and "opened". The depot as base is not being taken into consideration for the assignment. All the remaining locations are then being assigned to one of the closest of these opened locations for a fixed amount of service.

The target in this model is to locate a given amount of locations p, such that the lowest cost of assignment of all the remaining locations is achieved.

$$\sum_{j=1}^{n} pop_j \sum_{i=1}^{n} d_{ji} \cdot z_{ji} \ \rightarrow \min \tag{11}$$

$$s.t. \sum_{i=1}^{n} z_{ji} = 1 \qquad \forall\, j = 1, \ldots, n \tag{12}$$

$$y_i - z_{ji} \geq 0 \qquad \forall\, i = 1, \ldots, n,\ j = 1, \ldots, n \tag{13}$$

$$\sum_{i=1}^{n} y_i = p \tag{14}$$

$$z_{ji} \in \{0, 1\} \qquad \forall\, i = 1, \ldots, n,\ j = 1, \ldots, n \tag{15}$$

$$y_i \in \{0, 1\} \qquad \forall\, i = 1, \ldots, n \tag{16}$$

The main objective, as given in (11) is to minimize the total distance travelled by the population

Constraint (12) ensures that each location j has to be assigned to exact one hub i. But more then one location j can be assigned to one site i.

Constraint (13) ensures that each assigned location j may only be assigned to an opened site y_i. And in addition Constraint (14) sums up all the opened locations to guarantee that exactly p locations are opened.

4.3 Travelling Salesman Problem

For decades the TSP has been known and analysed. The idea behind it is that a salesperson is required to visit certain amount of locations p within a given area and that he starts from one location x_{0j} (home or depot) and returns to it at the end x_{i0}. The goal is to visit each customer only once with the lowest sum of costs or distance travelled. All has to fit in one big tour.

For the formulation of the TSP we used the model of Miller et al. [1960]

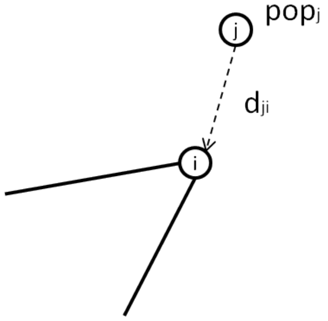


Figure 3: Assignment of non-visited j with its population *pop* to visited node i

$$\sum_{i=0}^{n}\sum_{j=0}^{n} d_{ij} \cdot x_{ij} \ \rightarrow \min \tag{17}$$

$$\sum_{\substack{i=0\\ i\neq j}}^{n} x_{ij} = 1 \qquad \forall\ j = 1, \ldots, n \tag{18}$$

$$\sum_{\substack{j=0\\ i\neq j}}^{n} x_{ij} = 1 \qquad \forall\ i = 1, \ldots, n \tag{19}$$

$$u_i - u_j + p \cdot x_{ij} \leq p - 1 \qquad \forall\ i, j \in V, i \neq j \tag{20}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\ i = 0, \ldots, n,\ j = 0, \ldots, n \tag{21}$$

In the objective function (17) the sum of the distance between all visited nodes have to be minimized.

Constraints (18) as the in-degree constraint, and (19)as the out-degree constraint, show that each location on tour may only be entered and left once. Non of these locations are allowed to be visited twice.

To avoid subtours we add additional constraint (20). It guarantees that if vertex x_{ij} is on tour, the rank of u_i is lower then rank of location u_j.

5 Large Neighbourhood Search (LNS)

The LNS, as all other meta-heuristics, is based on the idea of destroying and repairing best found solutions, to reach optimal solutions. (Illustrated in Figure 4) .

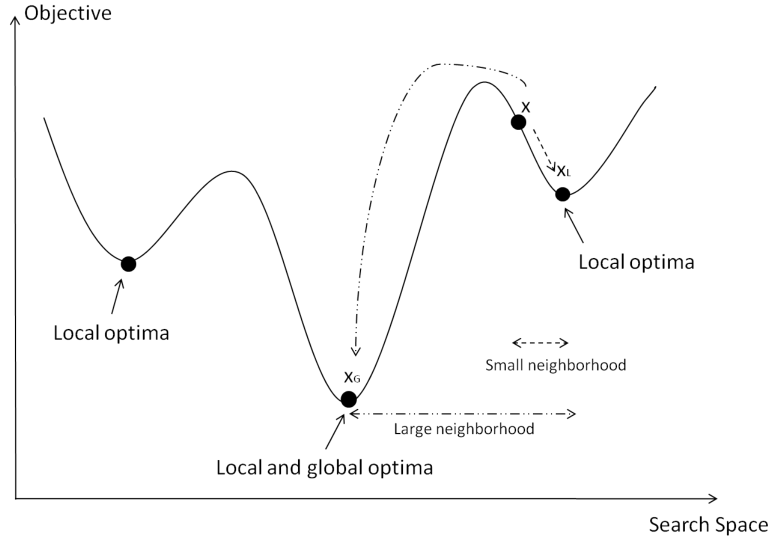


Figure 4: Meta-heuristic Search with Local and Global Optima

Shaw [1998] introduced this heuristic the first time. Its basic principle can be described as a gradually destroying and repairing a existing solution. A further adaptation is based on the idea of Schrimpf et al. [2000] *ruin and recreat* with accepting even slight worse results for a certain amount of iterations. As Pisinger and Ropke [2010] mentioned the *degree of destruction* is of high importance. If only a small part of the solution is destroyed, then the repair operator might have troubles by exploring the large neighbourhood. But on the other hand even destroying a big part of the existing solution could resolve in poor quality solutions. Especially in this matter a certain degree of destruction, based on the instance size is a common tested possibility. But even randomness is an important tool for the meta-heuristics, as the quality of destroying can be improved.

When destroying the current solution, an infeasible solution is created. Therefore the repair operator has to bring the solution back to feasibility. This can work in many different ways. Either the repair operator is checking for itself for the best greedy heuristic, by inserting the best/cheapest possible. Another method is by checking in combination of the destroy and repair operator. In that matter a broader neighbourhood can be surveyed.

In Algorithm 1 we want to represent the basic idea of the LNS. Let x be the current solution and x^b the best solution. In row 4 we destroy $d(.)$ and repair $r(.)$ our current solution $r(d(x))$ and the solution obtained is referred to as x^t. Then we check in our if-function, if this temporary solution will be accepted for further calculation. This acceptance criteria can be based on different ideas. The simplest way is to accept improving results.

Afterwards the LNS works by comparing if the objective value of temporary solution $c(x^t)$ is better then the formally best known objective value $c(x^b)$.

Regarding the stopping criteria the most common ways are setting limits of computing time or a certain amount of iterations.

When applying the LNS on the RSP, it works through gradually destroying (deleting node from tour) and repairing (inserting new node) by taking the assignment into consideration. Through this method we always take the problem as a whole into consideration. Therefore we will refer to it as "1-Step Heuristic" throughout the thesis.

For solution representation we stated two different vectors with all relevant informations. The first vector illustrates each opened location. The position of each location on the vector defines the routing order.

The second vector defines to which hub each location is being assigned.

As it will be explained in the following sections, we used different kind of

Algorithm 1 Large Neighbourhood Search

```
INPUT: feasible starting solution $x$
$x^b \leftarrow x$;
repeat
    $x^t \leftarrow r(d(x))$;
    if accept $(x^t, x)$ then
        $x \leftarrow x^t$;
    end if
    if $c(x^t) < c(x^b)$ then
        $x^b \leftarrow x^t$;
    end if
until stop criterion is met
return $x^b$
```

destroy and repair operators.

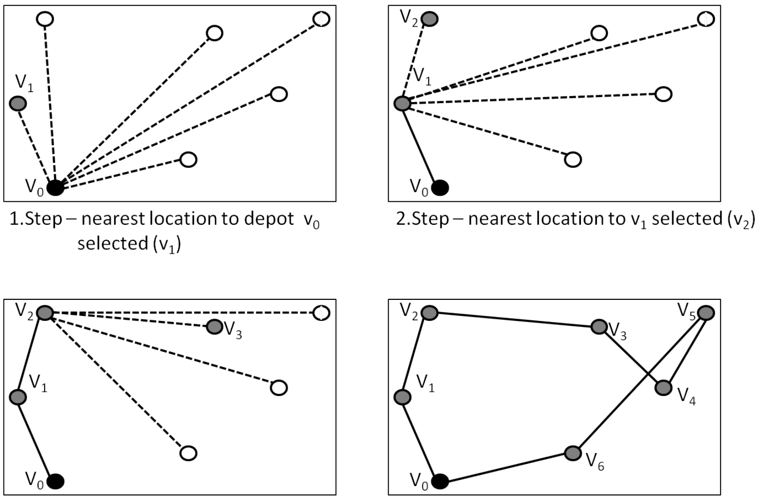
5.1 Starting Solution

In the beginning a random starting solution is being created. Out of all locations are randomly p locations selected as hubs. In the following step all the remaining locations are being assigned to their nearest hub.

For the routing the opened location are then being arranged with help of the *Nearest Neighbour* heuristic starting from the depot. Afterwards the tour length is locally improved with a *3-opt* heuristic. These achieved results are then the base for further calculation and improving steps of the implemented LNS-Heuristics.

5.1.1 Nearest Neighbour

The Nearest Neighbour heuristic is a greedy algorithm for solving TSP. The aim is to find a cheap tour by starting at one point (mostly the depot), visiting all possible locations and finally ending the tour again at the starting point . In each step, the closest non visited node will be added. Figure 5 illustrates the first three steps and the final solution achieved.



1.Step – nearest location to depot v_0 selected (v_1)

2.Step – nearest location to v_1 selected (v_2)

3.Step – nearest location to v_2 selected (v_3)

Final Solution – all locations assigned

Figure 5: Nearest Neighbour Heuristic

This heuristic tends to generate poor solutions. Especially the last routing decisions tend to result in high routing costs.

5.1.2 Local Search

For the Local Search we used the 3-Opt. It was introduced by Lin [1965] and aims in improving the existing tour by switching parts of it.

We define parts of the tour with $i, i+1, j, j+1, k, k+1$. In the first step we start deleting edges and adding new ones. The edge between i and $i+1$ will be deleted and a new edge between i and $j+1$ added. The same happens with j and $j+1$, which will be deleted and edge j and $k+1$ added. Finally the edge $k, k+1$ deleted and $k, i+1$ added. Figure 6 illustrates the moves graphically. As it can be seen, the part of $k+1, i$ will be inserted between $j, j+1$.

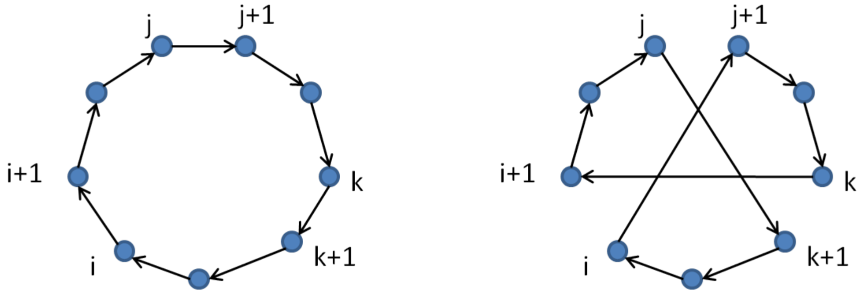


Figure 6: Local Search 3-Opt Improvement Heuristic

5.2 Destroy Operator

The Destroy Operator is a key part in search for the optimal solution.

By destroying the current solution we gradually delete hubs from the tour, and then reassign the non-routed locations to the remaining tour-members. This removal can be applied in different ways.

- Randomized - naive version and
- Biased - weighted savings potential

As Randomized already indicates, the selection of a hub for deleting works purely naive. The operator selects randomly one of the hubs for deleting.

The Biased approach on the other hand defines the highest cost saving potential for each hub. The one with the best potential will then have the highest probability of being deleted.

In Figure 7 we illustrate the basic idea behind it. Location k as part of the current solution is on the tour. The destroy operator selects this hub k for deleting. Therefore the route has to be reconnected with the edge (v_i, v_j). The formally to k assigned locations have as well to be reassigned to the new closest hub.

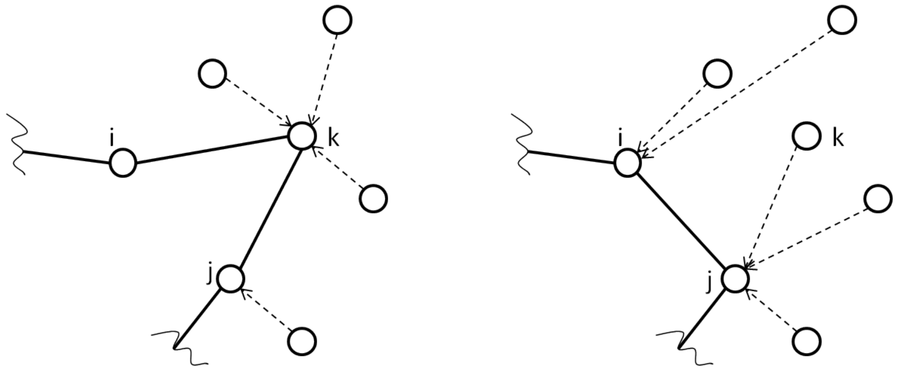


Figure 7: Illustration of deleting location k from tour, the reconnecting of the tour and reassignment of non visited locations including location k

As a consequence to this destroy operation, the now existing solution is not feasible any more. Therefore we had to implement as well a repair operation. In Section 5.3 we will explain the repair steps in detail.

5.2.1 Random Destroy

As the name of this method already indicates, is it based on randomization. This very simple operations works by deleting randomly one of the opened locations k, on the tour. So if there are p hubs in the beginning, we delete one location from the tour and have just $p - 1$ left. All the assigned locations to this deleted location will then be assigned to the next best opened hub. The closed hub will be assigned as well to the remaining opened locations.

As this method is purely randomized, it works to optimality with smaller problem sets. The probability of finding superior solutions or even the global optimum rises because of its big share in existing solution. The meaning of it is, that when having just 6 hubs and 12 assigned nodes, it will randomly find the optimal solution within a certain time to a much higher probability. In comparison when having for example 6 hubs and 30 assigned locations the probability of finding the optimal solution in the same time is much lower.

With increasing size of benchmarks, this method is limited, just because of the amount of possibilities for random deleting hubs.

5.2.2 Biased Destroy

The idea of the biased destroy method is based on the evaluation of all hubs that can be deleted. For each of these location a value of $cost_k$ is being calculated.

These $cost_k$ represent the value of deleting location k from the tour. Reassigning the formally assigned locations of k to the new closest location $c(j)$. And reconnection of the tour between v_i and v_j. In function (22) all the values are added into one $cost_k$.

$$cost_k = -d_{i,k} - d_{k,j} + d_{i,j} - \sum_{j=1}^{n} pop_j \cdot z_{j,k} \cdot (d_{j,c(j)} - d_{j,k}) \tag{22}$$

For all opened locations these costs will be calculated. Each will receive a defined percentage calculated out of the sum of all savings divided by its value. For choosing the selection point a random percentage number will be selected. The location representing this percentage will then be deleted from the tour. As an effect of this selection process, the one location with the highest percentage, has the highest probability of being selected. Figure 8 shows how this idea of the biased approach works.

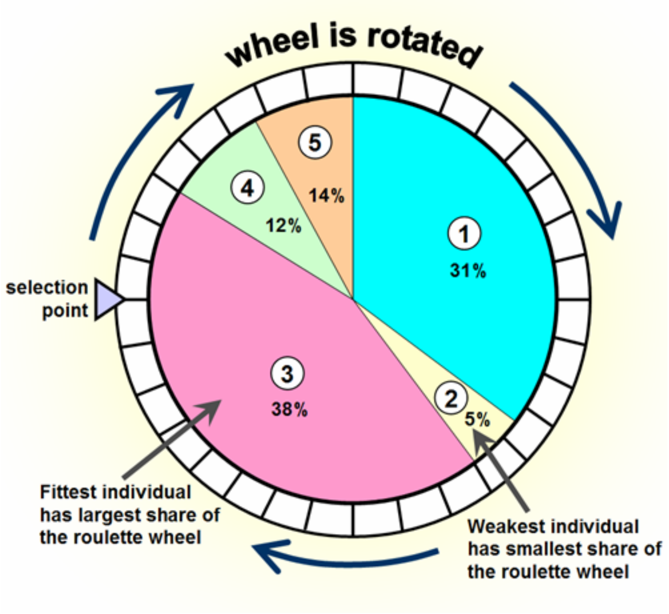

wheel is rotated
5
14%
4
12%
1
31%
selection point
3
38%
2
5%
Fittest individual has largest share of the roulette wheel
Weakest individual has smallest share of the roulette wheel

Figure 8: Biased Destroy - Selection Process (Newcastle Engineering Design Centre [2011])

The now existing tour is actually not feasible, as we have one location to little on the existing tour ($p-1$). Therefore the insertion part of the new hub will be done with the *Repair Operator* explained in following section.

5.3 Repair Operator

Regarding the insertion/repair of the needed hub (as we do have $p-1$), we implemented a *greedy* Best Insertion Heuristic. As we do have an infeasible solution we need to insert a new location to satisfy the condition of p locations. All the non opened locations are being checked for the cost of insertion. Under the condition that the hub deleted before is tabu listed. The allowed non opened locations will be checked for the sum of:

- best insertion place w on tour with the lowest costs/distance and

- reassignment of all the non opened locations

The one location l with the lowest summed costs/distance will then be opened and added to the tour at the best possible spot w. Then the possible non opened locations will be assigned to the new hub (as calculated in the reassignment part).

5.4 Best Pair Selection

Best Pair Selection is a special heuristic we implemented. Compared to the Destroy Operators implemented in Section 5.2, the advantage of this heuristic is in the approach of combining **destroy and repair** operators in one single operation. As we combine these two operations we do get the advantage of finding the best possible exchange of pairs in our RSP.

In this method we will look at each single opened hub on the tour. For each location of it we will check for the consequence if deleting it from the tour, and instead inserting one of the formally non-routed locations on the tour. The best pair locations, with the highest cost reduction, are then being selected as the best suitable options for deleting and inserting to the tour.

5.5 Post Optimizer

In the last step the tour will be once more optimized with the help of the 3-opt method. This final step tries to improve the existing solution even more.

5.6 Acceptance Criteria

Based on the former mentioned idea of Schrimpf et al. [2000] with the adaptation of even accepting slightly worse results, we created the parameter of T_{max}. This gave us the possibility of calculating with even worse results for a certain amount of iterations in dependence to the size of p. (number of iterations = $\lfloor \frac{p \cdot T_{max}}{100} \rfloor$)

5.7 Termination Criteria

For termination of the LNS search we set the program to run for a certain amount of iterations without improvement $IT_{max} = 200$.
The meaning was, that the program could run until it found a better solution, then it had 200 more iterations to search the neighbourhood for superior results. When superior was found, the counter for IT_{max} was set to cero and starts from new for searching the neighbourhood.

6 Computational Experiments

All the calculating was done on the same desktop computer for comparison reason. As software we used the "Microsoft Visual C++ 2008 Express Edition". In addition we used the "IBM ILOG Cplex 12.1" with the use of a C++ API. All calculations were executed on a PC with 2.20 GHz and 1GB RAM.

On each set we made 5 independent runs with parameter of $IT_{max} = 200$ and $T_{max} = 40\%$. The runs are being compared with two values. The *solution quality* (total costs) and the *runtime* (in seconds) needed for finding the best solution. Best values in each instance are then imprinted "bold" for easier comparison.

6.1 Data

In order to test the performance of the proposed algorithm we used the instances proposed by Hoshino and de Souza [2010]. The instances have been modified in the following way.

- All nodes are customer nodes.
- We set the numbers of vehicle to 1.

- The capacity of the vehicle is supposed to be unlimited.
- All customers are eligibly to be selected as hubs.
- The demand (namely in terms of pop_j) was set to 100 at each location.

In our calculation we used following instances:

- $12A$ and $12B$
- $25A$ and $25B$
- $50A$ and $50B$
- $75A$ and $75B$
- $100A$ and $100B$

In total $n = 12$ nodes to the maximum of $n = 100$ nodes were in the set. In all these instances both categories A and B were used. B has compared to A larger distances between the different vertices.

The number of hubs to be opened has been fixed to $p = \lfloor \frac{n}{2} \rfloor$. The size of population pop was set to 100.

6.2 1-Step Exact versus 2-Step Exact

By comparing these two methods we want to analyse the effects in terms of solution quality when looking at the problem in one integrative model compared to the splitting of assignment and routing into two separate calculations.

In the Tables 1 and 2 we compare the results of the combined 1-Step Exact approach to the 2-Step Exact solutions.

Comparing the results of the 2-Step Exact version with the 1-Step Exact, we can clearly see that solution quality only has a small gap of 0.13%. As the 1-Step

Exact is minimizing the assignment costs and at the same time minimizing the routing costs of the opened hubs, it has the possibility to switch certain locations from opened to closed location and vice versa. The 2-Step method is in that way limited, as it first minimizes the assignment costs and determines which locations have to be routed. Only then it is minimizing the routing costs.

In terms of runtime of these settings we do have one big advantage in the 2-Step Exact version: the run-time for finding a solution is by far lower compared to the 1-Step Exact. The reason lies in the complexity of the integrative version. As the program is optimizing both objectives the runtime increases significant by increasing problem/node size. So the average run time of the 2-Step method is just 0.32% (Class A) and 0.17% (Class B) of the 1-Step.

Even as the 1-Step Exact method can find the optimal results in the considered instances, the 2-Step Exact method results are very competitive. When taking the runtime as well into consideration, we can clearly state that the 2-Step Exact approach is a powerful tool in solving these problem sets.

Class A	1-Step Exact		2-Step Exact			
Set	Solution	Runtime	Solution	Runtime	Gap Solution	Gap Runtime
12A	**7533**	0.53	7547	**0.19**	0.19%	-64.60%
25A	**10996**	1.24	10998	**0.20**	0.02%	-83.56%
50A	**16710**	64.59	16729	**2.34**	0.11%	-96.37%
75A	**19898**	1835.28	19923	**10.53**	0.13%	-99.43%
100A	**22451**	48541.10	22492	**146.80**	0.18%	-99.70%
Avg. A	**15517.60**	10088.55	15537.80	**32.01**	0.13%	-99.68%

Table 1: Class A - 1-Step Exact compared with 2-Step Exact

Class B	1-Step Exact		2-Step Exact			
Set	Solution	Runtime	Solution	Runtime	Gap Solution	Gap Runtime
12A	**52731**	0.25	52829	**0.17**	0.19%	-31.20%
25A	**76972**	1.23	76986	**0.20**	0.02%	-83.55%
50A	**116970**	45.08	117103	**2.31**	0.11%	-94.87%
75A	**139286**	1454.92	139461	**12.28**	0.13%	-99.16%
100A	**157157**	57349.20	157444	**87.95**	0.18%	-99.85%
Avg. B	**108623.20**	11770.14	108764.60	**20.58**	0.13%	-99.83%

Table 2: Class B - 1-Step Exact compared with 2-Step Exact

6.3 2-Step Method: Exact versus Heuristic

In this section we want to compare the quality of solution and computing time between the two different "2-Step" methods. Again we first solved the *P-Median* exact with CPLEX for the selection of the best hubs and the assignment of the non visited. This starting solution was then used for the routing operations.

In our first method we used CPLEX to calculate the exact solution of the TSP. We call this method "2-Step Exact" in the Tables 3 and 4.

Alternatively then a heuristic of creating the tour with a the Nearest Neighbour heuristic and afterwards to improve the route with the help of the 3-Opt improving heuristic. This method we simply call "2-Step Heuristic" in the Tables 3 and 4.

As both methods used the same starting solution it was only a matter of rearranging the hubs in a certain order, so that the tour length is minimized.

When comparing the results, the 2-Step Heuristic achieved real good results. As having a gap of just 0.10% on the average, we can clearly see that the quality of solution is high. Then even taking the run-time into consideration we can see the big advantage of this Heuristic method. It was by far faster in computing time. Compared on the average the Heuristic has a gap to the Exact method of

−97.44% in class A, and −96.01% in class B. In that way the heuristic only needs 2.56% of the Exact runtime in class A and 3.99% in class B.

We can clearly state that the solutions for both methods are not that much apart. Compared to the Exact method, the Heuristic results are indeed very promising. The gaps were just at 0.10% on average and the computing time by far lower.

Class A	2-Step Exact		2-Step Heuristic			
Set	Solution	Runtime	Solution	Runtime	Gap Solution	Gap Runtime
12A	**7547**	0.66	7556	**0.06**	0.12%	-90.40%
25A	**10998**	0.20	11003	**0.11**	0.05%	-46.31%
50A	**16729**	2.34	16733	**0.22**	0.02%	-90.66%
75A	**19923**	10.53	19953	**0.98**	0.15%	-90.66%
100A	**22492**	146.80	22520	**2.74**	0.12%	-98.14%
Avg. A	**15537.80**	32.11	15553	**0.82**	0.10%	-97.44%

Table 3: Class A - 2-Step comparison of Exact and Heuristic method

Class B	2-Step Exact		2-Step Heuristic			
Set	Solution	Runtime	Solution	Runtime	Gap Solution	Gap Runtime
12B	**52829**	0.17	52892	**0.06**	0.12%	-63.37%
25B	**76986**	0.20	77021	**0.09**	0.05%	-54.19%
50B	**117103**	2.31	117131	**0.22**	0.02%	-90.53%
75B	**139461**	12.28	139671	**1.00**	0.15%	-91.86%
100B	**157444**	87.95	157640	**2.73**	0.12%	-96.89%
Avg. B	**108764.60**	20.58	108871	**0.82**	0.10%	-96.01%

Table 4: Class B - 2-Step comparison of Exact and Heuristic method

6.4 1-Step Heuristic

As already explained in Section 5, we proposed three different LNS approaches for solving the available instances. The goal was to find the best suitable operator for calculating the problem-sets within reasonable computing time.

When applying on the different problem-sets we first of all calculated the solutions of the three approaches as explained earlier:

- **h1:** Random Destroy
- **h2:** Biased Destroy
- **h3:** Best Pair Selection

These three heuristic approaches (h1, h2 and h3) will show us how they work as single operations. With the Random Destroy operation h1 we just destroy the existing solution by randomly selection hubs for deleting from the tour. Afterwards the Repair Operator as, explained in Section 5.3, is being applied for recreating a feasible solution in a greedy way. By using the Biased Destroy h2 on all instances, the selection of nodes for deleting from the tour, are weighted with their saving potential. The probability of deleting the most suitable hub increases in that matter. The Repair Operator is afterwards applied for creating feasible solution. In the Best Pair Selection h3 the whole process of destroying and recreating feasible solutions is analysed in one step to guarantee the best opportunistic selection of hubs and locations.

In addition we created different combinations of these single operators. The goal was to make our Large Neighbourhood Search even stronger in combining the strength of 2 different methods. Therefore we created following combinations:

- **h4:** Biased Destroy + Random Destroy
- **h5:** Biased Destroy + Best Pair Selection

- **h6:** Random Destroy + Best Pair Selection

As a last approach we combined all three operators into one heuristic method h7. By combining all of them we want to see the strength as a triple combination, but as well the effect compared to the heuristics where always two approaches are combined.

- **h7:** Biased Destroy + Random Destroy + Best Pair Selection

For each of these combinations of LNS heuristics (h4, h5, h6 and h7), each partial operator was selected with the same probability.

For evaluation purposes we applied each of these seven LNS heuristics on all instances. The average achieved solutions of each heuristic, on each instance, were taken into further consideration. We ranked these results in term of quality, compared to the other heuristics.

In order to compare the seven variants of the heuristics we ranked them from 1 to 7. 1 as a rank, indicates the best achieved results and value 7 indicates the worst.

The results shown in Figures 9 and 10 are based on these ranked values.

Analysing the results in Figure 9 for class A, the heuristic h7 (Biased Destroy + Random Destroy + Best Pair Selection) did achieve the best results with a median of just 1 and even a value of 1 for the 75% quantile over all test instances. The quality of solutions over all instances are in this matter very high. Analysing the results of the single heuristics h1, h2 and h3 we can see that the quality especially of the Random Destroy h1 operation is very low with a value of 7 for the 25% quantile. The Best Pair h3 method on the other hand even outperforms the combination approach of h4 (Biased Destroy + Random Destroy). When only considering the pair combinations of h4, h5 and h6, the operator h5 outperforms the others with a median of 2.

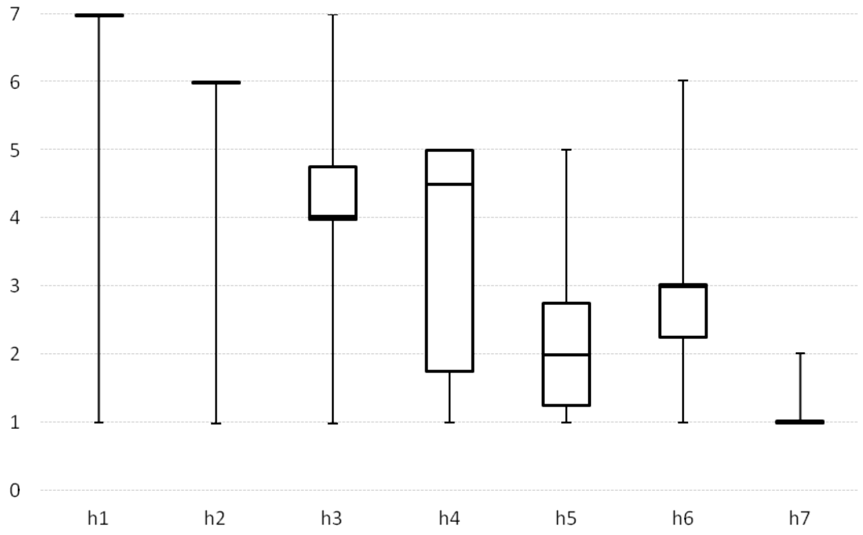


Figure 9: Class A - Rank comparison of Heuristics

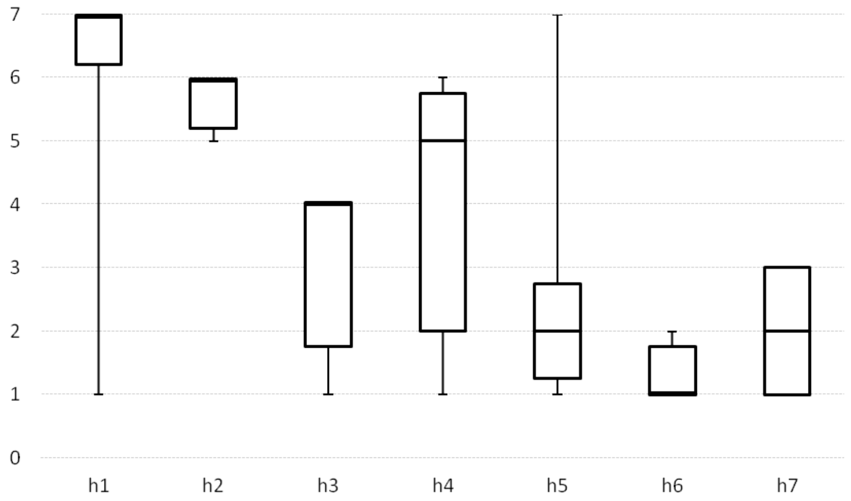


Figure 10: Class B - Rank comparison of Heuristics

Going into details with the results of class B in Figure 10 we again can see that the performance of the single operations h1 and h2 are limited. Random Destroy h1 did achieve a rank with a median of only 7, as it is not really directed towards a constructive destroy search. Due to the fact that it just randomly selects hubs for deleting, it works on small instances (the probability of finding the optimal solution is significant higher) and has its limitation on the bigger instances. Heuristic h3 is in that matter again of interest. As explained in Section 5.4, it is designed as a real goal oriented search in the large neighbourhood space. So it even outperforms heuristic h4 (Biased Destroy + Random Destroy). By analysing the combinations of the different operators, h6 achieved the best solution quality and did even outperform the triple combination of h7.

For considering the best heuristic for further studies, we compared the average values over all instances for both classes. In Table 5 these ranks are illustrated.

Rank Comparison in Average

	h1	h2	h3	h4	h5	h6	h7
Average Class A	6	7	4	5	2	3	**1**
Average Class B	5	7	4	6	2	**1**	3

Table 5: Class A and Class B - Heuristic comparison based on Ranks

The results demonstrate that heuristic h5 is managing well in both instances. Even though it is not the best in either class, it still performs on average best over both instances. Especially the part of ”Best Pair” with its goal oriented destruction and insertion and in combination with ”Roulette Wheel” is performing good.

Based on these results we did state heuristic h5 as the best suitable method for LNS. Therefore we applied it in further calculations for comparison of the results with the 1-Step Exact method. From now on we will refer to it as the ”1-Step Heuristic” method.

6.5 1-Step Exact vs. 1-Step Heuristic

As we have determined the best 1-Step Heuristic, it is of big interest to compare these results with our optimal results. Especially as the goal was to find results as close as possible to optimal results with our LNS heuristic. But within a reasonable time compared to the exact method of "1-Step Exact". Only with the comparison of these values the real quality of the heuristic h5 can be determined.

Results in the following Tables 6 and 7 illustrate the solutions of costs and run-time of the "1-Step Combined" version compared to the "1-Step Heuristic" of h5. For the heuristic h5 we used the best achieved results of 5 runs in each set. For the time we took the average run-time of the 5 runs, when the solutions have been found (not the total runtime).

On the sets of 12A, 12B and 25A, 25B our heuristic achieved the optimal solutions of our Exact solution method. Even so in 75A and 50B. In the remaining sets the gaps between both solutions are very small and at a maximum of 0.02%. So even these results are very promising especially when taking the runtime into consideration. For the set of 100A, the 1-Step Exact method needed almost 13.5 hours for terminating. Heuristic h5 on the other hand found results, only 0.01% off, in less then 10 minutes on average.

These results illustrate clearly that our applied heuristic h5 is achieving solutions of very high quality. A gap of just 0.01% to the optimal solution demonstrates this quality clearly. Even the runtime is of such high quality that our LNS heuristic can be seen as a great method for solving.

6.6 Sensitivity Analysis

As mentioned in earlier stages we calculated all the heuristic instances with a parameter of T_{max} for achieving optimal results.

Class A	1-Step Exact		1-Step Heuristic h5			
Set	Solution	Runtime (RT)	Solution	Avg. RT	Gap Solution	Gap RT
12A	**7533**	0.53	**7533**	**0**	0.00%	$\leq$ -99.99%
25A	**10996**	1.24	**10996**	**0.12**	0.00%	-90.43%
50A	**16710**	64.59	16711	**19.33**	0.01%	-70.07%
75A	**19898**	1835.28	**19898**	**220.00**	0.00%	-88.01%
100A	**22451**	48541.10	22456	**587.61**	0.02%	-98.79%
Avg. A	**15517.60**	10088.59	15518.80	**165.41**	0.01%	-98.36%

Table 6: Class A - 1-Step Exact compared with 1-Step Heuristic h5

Class B	1-Step Exact		1-Step Heuristic h5			
Set	Solution	Runtime (RT)	Solution	Avg. RT	Gap Solution	Gap RT
12B	**52731**	0.25	**52731**	**0.028**	0.00%	-88.88%
25B	**76972**	1.23	**76972**	**0.11**	0.00%	-90.91%
50B	**116970**	45.08	**116970**	**13.72**	0.00%	-69.56%
75B	**139286**	1454.92	139307	**204.83**	0.02%	-85.92%
100B	**157157**	57349.20	157192	**1070.65**	0.02%	-98.13%
Avg. B	**108623.20**	11770.14	108634.40	**257.87**	0.01%	-97.81%

Table 7: Class B - 1-Step Exact compared with 1-Step Heuristic h5

The idea behind this T_{max} is to avoid the local optima by partially accepting even worse results. This parameter is in dependence of size of p. For example if $p = 6$ and $T_{max} = 40\%$, the program is allowed to calculate for a maximum of 2 iterations (= 40% of p) with even worse achieved results. But these amount of iterations were not always of same size. It was given the possibility to either calculate 0, 1 or 2 iterations with worse results.

In following we calculated on the problem sets with different values of T_{max}.

As we wanted to see the clear effect on all the test instances we made a summary over all instances with each setting. We started with a $T_{max} = 0\%$, in which case just the close neighbourhood is being analysed. Then the values were increased to $10\%, 20\%, 40\%$ and 60%. For evaluation we summed up the results of the *Average Solutions* and *Average Computing Time* needed for finding the solutions.

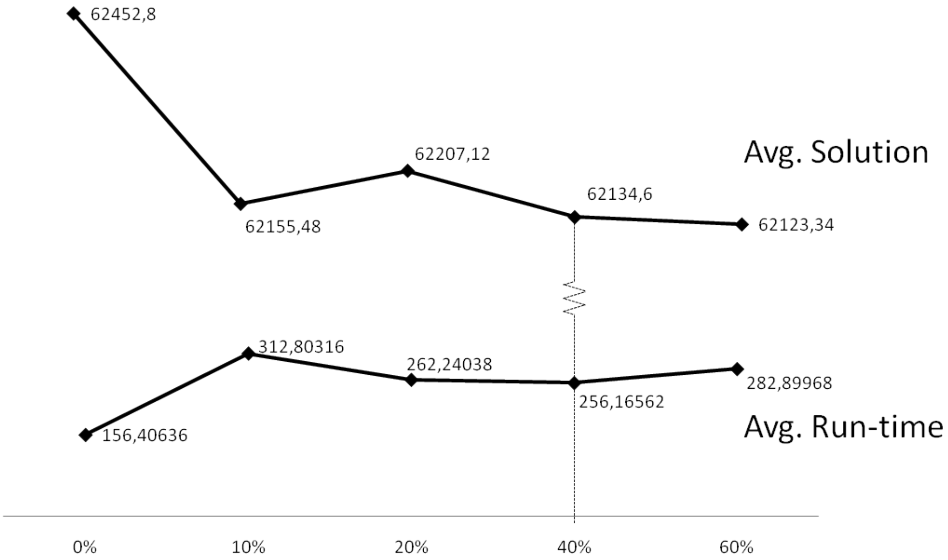


Figure 11: Summed Results for T_{max} variation

The results in figure 11 show that $T_{max} = 40\%$ should be used as the best suitable index for calculating the selected heuristic of h5. Using this value we will get promising results, in combination with a comparable fast computing time.

7 Conclusion

This diploma thesis analyses different kind of solution methods for solving a Ring Star Problem (RSP) for Disaster Relief. This problem has its practical usage in terms of supplying aid to disaster affected areas, where help-organisations might not be able to reach the whole population. Therefore the Ring Star with its routing and the assignment of non routed locations towards a routed one gives a practical solution approach of how to get the maximum out of the limited resources available.

Different approaches were implemented. Initially a integrative exact weighted sum approach was implemented. This method summed up both partial problems of the RSP (p-Median and TSP) into one combined objective.

Additionally we applied two different 2-step approaches by first solving the assignment of non routed to a certain amount of hubs (p-Median) and afterwards the routing of these hubs by using the classical TSP approach. Once we used the exact method for solving the TSP. As another method we applied heuristics for solving the routing.

At last we implemented different LNS operators with the goal of being more efficient in term of runtime with high solution quality. For the efficient LNS 1-Step Heuristic we analysed different destroy and repair operators. We came to the conclusion that our version with a Biased Destroy (+Greedy Repair) in combination with a Best Pair selection the most efficient heuristic is. This method worked best on all instances, when taking average results into consideration.

The results achieved in all different methods, as explained before, indicate that we have to interpret the results in different ways. Either we want to have optimal results, which do need a long computing time (1-Step Combined). Or we can accept a very little gap of maximum $0,02\%$ with a rather promising computing time (1-Step Heuristic). Or as a final method we accept the results to be up to around $0,20\%$ off from the optimal results but with a very fast computing time (2-Step Exact).

8 Outlook

In further studies it would be of great interest to see the effect of the created heuristic when adding certain additional constraints. One special field would be, that if certain locations are not permitted to be visited as a constraint - like not being able to reach. In a way the opposite idea of the "Steiner Nodes" which have to be visited.

The idea of adding a restriction of certain limited capacity of the routed vertices would make the problem even less complex to solve. Especially in practical matter this capacity restriction would be of interest. When depots for relief are created, they are not able to supply infinite goods. Therefore the idea of a certain capacity constraint could be taken into consideration.

This capacity restriction can be taken into consideration as well for a vehicle driving the given tour. It might only have limited loading space available. Therefore it would not be able to supply all warehouses in just one tour.

As all vehicle have a certain limit of range (for example a maximum travel distance) the idea of limiting the maximum tour length could be an additional constraint.

All these additional approaches would make the Ring Star Problem even more specific to certain real world needs. In this matter the explored approaches in this diploma thesis can help to solve the relief distribution in coming disasters.

References

E. K. Burke and G. Kendall, editors. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques.* Springer-Verlag, 2005.

K. Doerner, A. Focke, and W. J. Gutjahr. Multicriteria tour planning for mobile healthcare facilities in a developing country. *European Journal of Operational Research*, 179(3):1078 – 1096, 2007.

E. A. Hoshino and C. C. de Souza. Instances for the capacitated m-ring-star problem. `www.ic.unicamp.br/~cid/Problem-instances/CmRSP/`, 05. May 2010.

S. Kedad-Sidhoum and V. H. Nguyen. An exact algorithm for solving the ring star problem. *Optimization*, 59(1):125–140, 2010.

M. Labbé, D. Peeters, and J.-F. Thisse. Chapter 7 location on networks. In C. M. M.O. Ball, T.L. Magnanti and G. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 551 – 624. Elsevier, 1995.

M. Labbé, G. Laporte, I. R. Martín, and J. J. S. González. The median cycle problem. *Université de Montréal*, 1999.

M. Labbé, G. Laporte, I. R. Martín, and J. J. S. González. The rings star problem: Polyhedral analysis and exact algorithm. *Networks*, 43(3):177–189, 2004.

M. Labbé, G. Laporte, I. R. Martín, and J. J. S. González. Discrete optimization: Locating median cycles in networks. *European Journal of Operational Research*, 160(2):457–470, 2005.

Y. Lee, S. Chiu, and J. Sanchez. A branch and cut algorithm for the steiner ring star problem. *Journal on Computing*, 8(3):194–201, 1996.

A. Liefooghe, L. Jourdan, M. Basseur, E.-G. Talbi, and E. K. Burke. Metaheuristics for the bi-objective ring star problem. In *Proceedings of the 8th European conference on Evolutionary computation in combinatorial optimization*, EvoCOP'08, pages 206–217, Berlin, Heidelberg, 2008. Springer-Verlag.

A. Liefooghe, L. Jourdan, and E. G. Talbi. Metaheuristics and cooperative approaches for the bi-objective ring star problem. *Comput. Oper. Res.*, 37(6): 1033–1044, June 2010.

S. Lin. Computer solutions of the traveling salesman problem. *Bell Syst. Tech. J.*, 44:2245–2269, 1965.

A. Mauttone, S. Nesmachnow, A. Oivera, and F. Amoza. Solving a ring star problem generalization. In *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, pages 981 –986, dec. 2008.

C. Miller, A. Tucker, and R. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7 (4):326–329, 1960.

N. U. Newcastle Engineering Design Centre, Merz Court. `http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php/`, 20. November 2011.

I. H. Osman and J. P. Kelly, editors. *Meta-heuristics: Theory & Applications.* Kluwer Academic Publisher, 1996.

D. Pisinger and S. Ropke. Large neighbourhood search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146, pages 399–419. Springer, 2010.

G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171, 2000.

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. pages 417–431. Springer-Verlag, 1998.

D. Sietz, M. K. Lüdeke, and C. Walther. Categorisation of typical vulnerability patterns in global drylands. *Global Environmental Change*, 21(2):431 – 440, 2011. Special Issue on The Politics and Policy of Carbon Capture and Storage.

P. T. and Leeson. Better off stateless: Somalia before and after government collapse. *Journal of Comparative Economics*, 35(4):689 – 710, 2007.

ZUSAMMENFASSUNG DEUTSCH

In Zeiten von Katastrophen ist die schnelle Hilfe für die betroffene Bevölkerung von größter Wichtigkeit. Mit Hilfe des Ring Star Problem (RSP) als Modell für Katastrophenhilfe, können hierfür unterschiedlichste Situationen in Betracht gezogen werden, um eine effiziente Hilfeleistung zu garantieren.

Da die gesamte betroffene Bevölkerung nicht immer direkt mit Hilfslieferungen versorgt werden kann, wird mithilfe des RSP versucht diese Lieferungen an angrenzende Gebiete, zu welchen der Zugang möglich ist, die betroffene Bevölkerung zu erreichen. Diese Gebiete/Ortschaften werden dann mit Hilfe einer Routenplanung miteinander verbunden, um eine effiziente und schnelle Hilfslieferung in alle betroffenen Gebiete zu garantieren.

Zur Untersuchung dieses Problems wurden Datensätze in den Größen von 12 bis 100 Ortschaften untersucht.

Zur Lösung dieses Problems wurden unterschiedliche Ansätze untersucht. Zu einem wurde eine exakte Lösungsmethode mithilfe einer CPLEX Schnittstelle im Microsoft Visual C++ implementiert. Diese Methode wurde mit einer integrativen gewichteten Zielfunktion gelöst. Als Grundlage hierfür diente das p-Median, für die Zuweisung der nicht besuchten Gebiete zu den besuchten Ortschaften, sowie das Travelling Salesman Problem (TSP) für die effiziente Routenplanung. Die erzielten Werte dieser exakten Methode dienten in weiterer Folge als Vergleichswerte für weitere Lösungsansätze.

Infolge wurde das Problem in zwei Teilprobleme zerlegt (p-Median und TSP). Diese wurden zu einem Teil sukzessiv mithilfe der CPLEX Schnittstelle gelöst. Zum Anderen wurde nur der Zuweisungsteil, das p-Median, mit CPLEX gelöst

und im Anschluss das Routing der besuchten Orte mithilfe von Heuristiken.

Weiters wurde das gesamte Problem mittels der Large Neighbourhood Search (LNS) Metaheuristik gelöst. Dies erfolgte durch sukzessives Zerstören und Reparieren von bestehenden Zuweisungen und Routen.

Die Auswertungen der gesamten Daten ergaben ein klares Bild bezüglich der Effizienz der erstellten, heuristischen Lösungsansätze. Die Zerlegung des RSP in zwei Teilprobleme und hierbei die Kombination von exakter und heuristischer Methoden stellte eine äußerst effektive Lösungsmethode dar. So erreichte diese Methode im Durchschnitt eine Lösungsgüte von 99,77%, und benötigte hierfür lediglich 0,01% (0,82 Sekunden) der Rechenzeit im Vergleich zur optimalen Lösung. Die Metaheuristik konnte eine noch bessere Lösung erzeugen, welche im Durchschnitt nur 0,01% von der optimalen Lösung entfernt war. Die Rechenzeit lag im Durchschnitt bei lediglich 2,19% der exakten Lösung, was einer durchschnittlichen Dauer von 257,87 Sekunden entspricht.

Curriculum Vitae

Personal Data

First- and Surname:	Göran Sjöström
Birth-date:	03.Juli 1980
Nationality:	Sweden

Education

2004 - 2011 University Vienna, Master in International Business Administration, Specialization: International Management and Production Management

1995 - 2000 Higher Institute of Technical Education in Dornbirn, Industrial Engineering and Management, Focus: Textile Management

1990 - 1995 Secondary school ”Bundesgymnasium” Bludenz

1986 - 1990 Primary School Bludenz and Thüringen

Working Experience

September 2008 - October 2011 Accounting Department at Foto Leutner GmbH, Vienna, Austria

February 2009 - October 2011 Teaching Assistant at Chair of Production and Operations Management University Vienna, Austria

February 2005 Assistant of Sales Director at Aplicaciones Mecánicas del Caucho, S.A., San Sebastian, Spain

February 2002 - July 2004 Sales Executive at Thomas Trading International AB, Västra Frölunda/Gothenburg, Sweden

Februar 2001 - July 2001 Junior Sales Executive at International Delton Fabrics LTD, Hong Kong, China

October 2000 - December 2000 Assistant of the Management Board at Thomas Trading International AB, Västra Frölunda/Gothenburg, Sweden

Language Skills

German	mother tongue
Swedish	business fluent
English	business fluent
Spanish	fluent written and spoken
French	basic knowledge

Further Training

September - December 2001 Spanish language lectures at Universidad Autonoma de Puebla, Puebla, Mexico

April - June 2002 Swedish language lectures at Folkuniversitetet Gothenburg

Additional Information

Driving License	B
Computer Skills	Microsoft Office, Excel Solver, C++, SAP, Navision

November, 2011