PROBABILISTIC VEHICLE ROUTING PROBLEMS

by

Antoine Jézéquel

Dipl. Ing., Ecole Centrale de Lyon, France (1984)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE DEGREE OF

MASTER OF SCIENCE
IN TRANSPORTATION

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August, 1985

Antoine Jézéquel

The author hereby grants to M.I.T. permission to reproduce and distribute copies of this thesis document in whole or in part.

Signature of author ____________________________ Amedeo R. Odoni
Department of Civil Engineering August 15, 1985
Thesis Supervisor
Certified by ________________________________
Accepted by ________________________________

Chairman, Departmental Committee of Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
ARCHIVES

JAN 02 1986
PROBABILISTIC VEHICLE ROUTING PROBLEMS

by Antoine Jézéquel

Submitted on August 9, 1985, in partial fulfillment of the requirements for the degree of Master of Science in Transportation.

ABSTRACT

This thesis presents an examination of the Probabilistic Traveling Salesman Problem (PTSP) methodology on a practical basis.

First we review the definition and the formulation of this new combinatorial optimization problem. Then we define what a "good" tour can be "in the PTSP sense". We see that in the Euclidean case, such a "good" tour is the product of a compromise between a tour of short length and a tour including zigzag patterns.

We assess the efficiency of several different heuristics for the PTSP. We establish that two procedures proposed in Jaille[1985] (namely the Almost Nearest Neighbour Algorithm and the Supersavings Algorithm) behave poorly. Then we show that a manual-solution procedure is very efficient and very appropriate for capturing the conceptual basis of the apparently self-contradictory definition of the "good" PTSP tour. This manual procedure is the most efficient for coverage probabilities greater than or equal to 0.5. For coverage probabilities less than 0.5, another heuristic derived from the Clarke-Wright Savings Algorithm (it consists of maximizing the savings in expected distance) behaves well. We show that a solution produced with this latter procedure for a given probability can also be a very good solution for another value, most of the time a higher value, of the coverage probability. Thus we propose a global procedure where the problem is attacked as a whole in terms of the entire probability range. The results formerly obtained are greatly improved.

In Chapter 3, we present a vehicle routing problem under uncertainty whose formulation is directly derived from the PTSP methodology. This problem is called the "Probabilistic Vehicle Routing Problem" (PVRP). We introduce the notion of risk and point out that we have to face important and somewhat surprising complications. We derive an upper bound for the expected length of a subtour. Then we propose a solution procedure based on the investigations of Chapter 2. Finally, we consider possible generalizations and relaxations of the formulation of the PVRP.

Thesis Supervisor: Amedeo R. Odoni

Title: Professor of Aeronautics and Astronautics and of Civil Engineering
ACKNOWLEDGEMENTS

I wish to express my deep gratitude to Professor Amedeo Odoni, whose patience, help, support, and many suggestions made this thesis possible.

I would like to thank Patrick Jaillet for his useful explanations and comments at the early stage of this thesis.

Sincere thanks to Craig Dunbar and Philippe Delquié. Their suggestions, criticisms and mischievousness were almost always welcomed.

Finally, and most of all, I am forever indebted to my parents for their constant encouragement, support and understanding in completing this thesis and throughout my stay at M.I.T.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>1</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>2</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>3</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>6</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 2: A PRACTICAL VIEW OF THE PTSP, THE</td>
<td></td>
</tr>
<tr>
<td>EFFICIENCY OF SEVERAL HEURISTICS</td>
<td>11</td>
</tr>
<tr>
<td>2.1 A definition of the PTSP</td>
<td>11</td>
</tr>
<tr>
<td>2.2 The expected length of a PTSP tour</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Notation and assumptions</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 The expected length of a PTSP tour</td>
<td>16</td>
</tr>
<tr>
<td>2.3 The star-shaped example revisited</td>
<td>21</td>
</tr>
<tr>
<td>2.4 Practical approaches to the PTSP</td>
<td>24</td>
</tr>
<tr>
<td>2.4.1 Brief review of the alternatives</td>
<td>24</td>
</tr>
<tr>
<td>2.4.2 A practical view to the optimization problem</td>
<td>25</td>
</tr>
<tr>
<td>2.4.3 The Almost Nearest Neighbour Algorithm</td>
<td>26</td>
</tr>
<tr>
<td>2.4.4 The Supersavings Algorithm</td>
<td>30</td>
</tr>
<tr>
<td>2.4.5 Another heuristic based on the Supersavings approach</td>
<td>35</td>
</tr>
<tr>
<td>2.4.6 The manual approach</td>
<td>37</td>
</tr>
<tr>
<td>2.5 Some test cases</td>
<td>38</td>
</tr>
<tr>
<td>2.5.1 Computational results</td>
<td>38</td>
</tr>
<tr>
<td>2.5.2 Analysis and comments</td>
<td>41</td>
</tr>
<tr>
<td>2.6 NEWSAVE : a global procedure for the PTSP</td>
<td>47</td>
</tr>
<tr>
<td>2.6.1 The performances of NEWSAVE</td>
<td>47</td>
</tr>
<tr>
<td>2.6.2 A global procedure for the PTSP based on NEWSAVE</td>
<td>49</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.1</td>
<td>Graphical example of a PTSP</td>
</tr>
<tr>
<td>2.2</td>
<td>The star-shaped example</td>
</tr>
<tr>
<td>2.3</td>
<td>An illustration of the $L_t^{(r)}$'s</td>
</tr>
<tr>
<td>2.4</td>
<td>An illustration of the $L_{1,t}^{(r)}$'s</td>
</tr>
<tr>
<td>2.5</td>
<td>An illustration of the tactical difference between the classic Nearest Neighbour Algorithm and ANNA</td>
</tr>
<tr>
<td>2.6</td>
<td>Merging of subtours</td>
</tr>
<tr>
<td>2.7</td>
<td>Merging of pair of nodes</td>
</tr>
<tr>
<td>2.8</td>
<td>Illustration of the poor behaviour of ANNA</td>
</tr>
<tr>
<td>2.9</td>
<td>Tour drawn for problem 7 by means of MANUAL</td>
</tr>
<tr>
<td>2.10</td>
<td>Tour obtained for problem 7 by means of CNSA</td>
</tr>
<tr>
<td>3.1</td>
<td>Travelling an extra distance</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.1</td>
<td>Relative difference in expected distance for the two different tours of the star-shaped example</td>
</tr>
<tr>
<td>2.2</td>
<td>Number of best performances</td>
</tr>
<tr>
<td>2.3</td>
<td>Number of worst performances</td>
</tr>
<tr>
<td>2.4</td>
<td>Number of best performances after improvement</td>
</tr>
<tr>
<td>3.1</td>
<td>Lower bounds for the minimum size of the routing problem for risk to be taken</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

Distribution problems and especially vehicle routing problems have been and are the subject of intensive research. An exhaustive review of these problems is available in Bodin & Golden[1983]. Perhaps the best known and the most studied of all routing problems is the famous Travelling Salesman Problem (denoted in the following as the TSP). The TSP is often formulated as follows: a vehicle with unlimited capacity is based at a depot; the vehicle services a set of nodes each with unit demand; only routing costs are considered and the problem is to find a path through this set of nodes which minimizes the total routing costs (which are assumed to be a linear function of the total distance travelled). An excellent review of various approaches to this problem is contained in Parker & Rardin[1983].

With several vehicles of common capacity all based at the same single depot and with known node demands, one can define a standard Vehicle Routing Problem (VRP).

These routing problems have been mostly studied in a deterministic context where the elements of the problem are known with certainty before the problem is solved. Compared with the huge amount of literature available for deterministic routing problems (more than 650 references quoted in Bodin & Golden[1983]), only a few results taking account of any kind of uncertainty are readily available. To date, one can
identify four different set of problems which are characterized by the introduction of non-deterministic elements:

- the distances between points are non-deterministic.
  
  This type of problem is the oldest one. It is mainly concerned with the derivation of asymptotic bounds for the TSP in the plane. A good review of this problem can be found in Jaillet[1985].

- the demand at each node is non-deterministic.
  
  This problem is a variation of the standard VRP where the demand at each node is a random variable. In Stewart & Golden[1983] this problem is formulated as a Stochastic Vehicle Routing Problem (SVRP) and solved by means of stochastic programming methods. The formulation used by these methods renders the problem deterministic. Therefore, the SVRP is solved through the traditional heuristics used for the deterministic version of the problem.

- a third interesting variation can be found in Odoni[to be published,1985].

  The author examines the mail delivery in a rectilinear street with regularly spaced addresses. Using a geometrical probability approach, he derives for several route patterns the route length variability with respect to coverage probability (the probability that a given address requires mail delivery).

- the fourth approach (and the one we are concerned with in this thesis) is the product of the doctoral research of
Patrick Jaillet[1985].

Jaillet formulated a variation of the TSP known as the Probabilistic Vehicle Routing Problem (PTSP) where each point of the set of nodes requires a visit only with a given probability during any given instance of the problem (the coverage probability). A tour is defined a priori covering the entire set of nodes. The nodes which do not request service on a given instance are skipped. Jaillet derived a closed-form expression for the expected length of such a tour. Among many results, he also proposed procedures to solve this new optimization problem.

As far as we know, no attempt has been made so far to apply or to examine Jaillet's results and procedures on a practical basis.

In this thesis, in a first part we take a close look at the PTSP and we define what a "good tour" is in the PTSP sense. On this basis, we assess the efficiency of several different heuristics. In a second part, we formulate a Probabilistic Vehicle Routing Problem (PVRP) and we also propose an appropriate solution procedure.

Finally, having cleared some of the undergrowth from the unexplored land of the PTSP, in Chapter 4 we propose some directions for future research.
CHAPTER 2

A PRACTICAL VIEW OF THE PTSP,

THE EFFICIENCY OF SEVERAL HEURISTICS

2.1. A definition of the PTSP

The PTSP has been originally formulated as follows:
"Consider a problem of routing through a set of n points. On any given instance of the problem only a subset of k out of the n points (0 ≤ k ≤ n) must be visited, with the number k determined according to a known probability distribution. We wish to find a priori a tour through all n points. On any given instance of the problem, the k points present will then be visited in the same order as they appear in the a priori tour. The problem of finding such an a priori tour which is of minimum length in the expected value sense is defined as a "probabilistic traveling salesman problem" (PTSP)."

Based on the "Refuse Collection Example" given in Larson & Odoni[1981], a simple example of a PTSP is given in figure 2.1.

Thus, an optimal PTSP tour should be a tour (an a priori tour) that would "behave well" when its pattern is modified because of the missing nodes which are skipped. Obviously, this problem is more complicated than the TSP because it includes the TSP itself. The key contribution of Jaillet to this formulation is the derivation of a closed form expression for the expected length of the a priori tour. Before having a
close look at this expression in the following section, let us consider some interesting features and properties of the PTSP as derived by Jaillet.

The point that provides the essential motivation for our investigation is that it can be shown through a simple example that the optimal TSP may a very poor solution to the PTSP. The example given in Jaillet [1985] is the star-shaped tour as indicated in figure 2.2. 24 nodes are set on two concentric circles. The nodes correspond to the vertices of two concentric regular 12-gons. The ratio of the two diameters is 1.59. The inside 12-gon is such that each of its vertices is positioned between two successive vertices of the outside 12-gon. Figure 2.2. presents two alternatives tours. Tour A is the optimal TSP tour. Tour B is an alternative tour whose length is 0.7% greater than A's. Nevertheless, for a coverage probability of 0.9 (the probability that each node requests service) the expected length of tour B is 7.0% smaller than A's and 31.0% smaller for $p = 0.5$.

In view of the fact that the optimum solution to the TSP does not solve automatically the PTSP, one must develop some special solution procedures for deriving good tours in the PTSP sense. In the next section, we concentrate on the derivation of the expression for the expected distance of a PTSP tour. Then we return to the star-shaped example to define what should be a good tour "in the PTSP sense" with the help of some more exhaustive computations.
Figure 2.1: Graphical example of a PTSP
The original set of nodes

Tour A

Tour B

Figure 2.2: The star-shaped example
2.2. The expected length of a PTSP tour

2.2.1. Notation and assumptions

We first introduce some notation and assumptions necessary to present the closed form expression giving the expected length of a tour. For the sake of consistency, we use the same nomenclature as Jaillet [1985].

Let \( G = (N,A,D) \) be a complete, directed graph where:
- \( N \) is the set of points of cardinality \( N \);
- \( A \) is the set of arcs joining the nodes of \( N \);
- \( D \) is the distance (cost) matrix;
- \( d(i,j) \) is the cost of traveling from node \( i \) to node \( j \) through arc \((i,j)\).

\( t = (i_1,...,i_N,i_1) \) will represent a sequence of nodes forming a Hamiltonian circuit of the graph.

The length \( L(t) \) of the tour \( t \) is given by:

\[
L(t) = \sum_{j=1}^{N} d(i_j,i_{j+1}) \quad \text{where } i_{N+1} = i_1 \quad (2.1)
\]

We can notice that in the general case \( i_j \) is taken as \( i_{(j \mod N)} \).

In other respects, a black node of \( G \) will be a node that will always request a visit. In the PTSP, the depot is the typical case of a black node. A white node of \( G \) will be a node that will not always require a visit. We assume that
each white node is present with probability \( p \), independently of each other. Accordingly, in this case, the probability distribution for the number \( k \) of nodes present is the binomial one.

2.2.2. The expected length of a PTSP tour

First we present the expected length of a tour with no black nodes and \( n \) white nodes. Given a tour \( t \) of \( G \), \( t = (i_1, \ldots, i_n, i_1) \), we build \( n-1 \) quantities defined by:

\[
L_t(r) = \sum_{j=1}^{n} d(i_j, i_{j+r+1}), \quad r \in [0..n-2]
\] (2.2)

where as before: \( i_k = i(k \mod (n)) \).

\( L_t(0) \) is simply the length of the tour \( t \). \( L_t(r) \) is the sum of \( n \) elements of the distance matrix, each representing the distance from node \( i_j \) to its \((r+1)\)th successor with respect to the tour. Figure 2.3 gives an illustration of what these quantities are using once again the refuse collection example of Larson & Odoni[1981].

To derive the expected length of the tour, Jaillet first derives the conditional expected length of the tour, given that \( k \) nodes are missing from the original tour:

if \( n = k \) or \( n = k-1 \):

\[
E[L(t)/k \text{ missing nodes}] = 0
\]

if \( k \in [0..n-2] \):

\[
E[L(t)/k \text{ missing nodes}] = \frac{1}{\binom{n}{k}} \sum_{r=0}^{k} \binom{n-2-r}{k-r} L_t(r)
\] (2.3)
Noticing the fact that there are \( \binom{n}{k} \) different and equiprobable possible ways of having \( k \) missings nodes out of \( n \), calling \( D_q \) the resulting length of the tour for the \( q^{th} \) way, \( q \in [1..k] \), then one can write:

\[
E[L(t)/k \text{ missing nodes}] = \frac{1}{n \binom{n}{k}} \sum_{q=1}^{\binom{n}{k}} D_q \quad (2.4)
\]

Then the conditional expected distances can be derived by means of a combinatorial argument which states that sum (2.4) can be reorganised and expressed as a combination of the quantities \( L_r(t) \) previously defined. For details see Jaillet [1985], pp. 26-31.

Then recalling that we assumed a binomial distribution for the number \( k \), we can derive the unconditional expected distance for the tour \( t \):

\[
E[L(t)] = p^2 \sum_{r=0}^{n-1} (1-p)^r L_r(t) \quad (2.5)
\]

This expression has been used to estimate the expected lengths associated with our earlier star-shaped example.

For the applications we are going to study, we need a similar expression for the case where one node is black, namely when there is a depot. Suppose we have \( n \) white nodes, one black node and a coverage probability \( p \). Then Jaillet also derived the expected length of such a tour which is given by the
following expression:

\[
E[L(t)] = p^2 \left[ \sum_{r=0}^{n-2} (1-p)^r L_{1,t}^{(r)} \right] + p(1-p)^{n-1} L_{1,t}^{(n-1)} \tag{2.6}
\]

The quantities \( L_{1,t}^{(r)} \) are similar to the \( L_t^{(r)} \) 's defined before but they take account for the fact that one point (the depot) is always visited. \( L_{1,t}^{(r)} \) is defined as follows:

\[
L_{1,t}^{(0)} = d(i_j,i_{j+1})
\]

\[
L_{1,t}^{(r)} = d_{1,t}(i_j,i_{j+r+1}) \quad r \in [1..n-1] \tag{2.7}
\]

where \( i_j = i(j \mod (n+1)) \).

\( d_{1,t}(...) \) is obtained from \( d(...) \) according to the following rules:

for \( 1=j=n-r+1 \):

\[
d_{1,t}(i_j,i_{j+r+1}) = d(i_j,i_{j+r+1})
\]

for \( n-r+1<j=n+1 \):

\[
d_{1,t}(i_j,i_{j+r+1}) = d(i_j,i_1) + d(i_1,i_{j+r+1})
\]

The second term indicates, only that node \( i_1 \) (the depot) cannot be skipped. Figure 2.4 illustrates this point and present a graphical representation of the \( L_{1,t}^{(r)} \)'s. We can notice that these expressions may be extended to the case where there are \( m \) black nodes \( (m>1) \) (see Jaillet[1985], pp. 38-40) and where the probability distribution is more general than the binomial distribution.
Figure 2.3: An illustration of the $L_n(r)_{\leq}$
Figure 2.4: An illustration of the $L_{1,t}^{(s)}$'s
2.3. The star-shaped example revisited

We now return to the star-shaped example in order to provide a better understanding of the PTSP. Let us first mention some interesting properties of the PTSP which go to improve our understanding of the problem. Jaillet showed that:

- the optimal PTSP in the plane can intersect itself in the Euclidean case. For the TSP, it is trivial to see that the optimal tour cannot intersect itself since, thanks to the triangular inequality, it suffices to "uncross" a path to obtain an immediate reduction of the length of the tour;
- for small size problems (up to four nodes with distance matrix symmetric), the optimal tour is also the optimal PTSP;
- in the Euclidean case, if the set of nodes is itself the convex hull of the set of nodes, the optimal PTSP is the optimal TSP.

Thus, as already mentioned, there is no straightforward relationship from the TSP to the PTSP. Let us rework the star-shaped example to have a more precise and practical idea about the kind of tours that will be good "in the PTSP sense". We slightly modify the example by turning one of the nodes into a black node, node 1 for example. Then for a range of probability between 1 and 0.1 we derive the relative difference between the expected distances for the optimal TSP
tour A and the alternative tour B. The results are listed in Table 2.1.

Table 2.1: Relative difference in expected length for two different tours for the star-shaped set of nodes.

<table>
<thead>
<tr>
<th>p</th>
<th>1</th>
<th>.9</th>
<th>.8</th>
<th>.7</th>
<th>.6</th>
<th>.5</th>
<th>.4</th>
<th>.3</th>
<th>.2</th>
<th>.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>D%</td>
<td>-.6</td>
<td>-6</td>
<td>-11</td>
<td>-16</td>
<td>-20</td>
<td>-23</td>
<td>-24</td>
<td>-22</td>
<td>-17</td>
<td>-9</td>
</tr>
</tbody>
</table>

Why does tour B behave better than tour B in the PTSP sense? Consider tour A. When a node is missing, it is skipped. But with this configuration, there is not that much reduction in the length of the tour. It keeps more or less the same shape. It does not really "shrink". On the other hand, with tour B, the absence of a node induces a relatively large reduction in the distance travelled. The tour "shrinks". Both tours start with about the same length in the deterministic case (p = 1), then rapidly, reductions occur in the pattern of tour B for probabilities even close to 1.

Why are the relative differences in expected length not as large in the all-white nodes case? (p = .9, 7%; p = .5, 31%). Because the presence of the black node prevents the tour B from "shrinking" too much. It induces reduced flexibility in the patterns of both tours which affects more tour B which is otherwise more likely to "shrink" dramatically.
One can also notice that the relative difference in expected distance reaches a maximum for \( p = 0.4 \) and then diminishes. Therefore, it is not guaranteed that tour B is the best solution for any value of the coverage probability.

This set of facts and observations leads to the following statements:

- For any problem of practical interest (more than four nodes, depot included), the optimal TSP may not be a good solution for the PTSP. On the other hand, there may also exist problems for which the TSP is a good solution for the PTSP.

- Different PTSP tours may be optimal for different values of the coverage probability for a given problem, i.e. a given graph. Thus, the procedures to solve this problem must be \( p \)-dependent.

- A good PTSP tour tour should be a tour that "shrinks" when some nodes are missing, i.e. "takes advantage" of the fact that nodes are missing. This means that patterns that include nodes laying out more or less in line are not desirable. Rather, one should prefer to come out with paths that include a lot of zigzags. The tour must be compact in order to minimize the \( L_{1,t}^{(r)} \)'s.

In the following section, we examine a manual procedure based on these remarks and two heuristics among those proposed in Jaillet [1985] for solving this problem. We also propose a new solution-procedure.
2.4. Practical approaches to the PTSP

2.4.1. Brief review of the alternatives

In chapter 5 of Jaillet [1985], four exact optimization methods based on mathematical programming formulations are proposed, as well as a Branch and bound approach. The problem is successively formulated as an integer nonlinear programming problem, a mixed integer linear problem and finally a pure integer linear program. Jaillet states that the last two techniques should be discarded because of the huge number of variables they require. Then he suggests that for some cases, the first formulation may be turned into a Branch and bound approach which is considered "promising".

"When one notes that the TSP is a special case of the PTSP and that the most efficient exact methods for this problem have been built on more than two decades of intensive research, one realizes that this area must be left as a topic for future research."

Such exact approaches are beyond the scope of this thesis since we are interested here in a more "geometrical" approach. To conclude this very brief review of the possible exact methods, it must also be said that Jaillet also proved the inadequacy of dynamic programming for solving the PTSP.

Jaillet also suggests several heuristics procedures among which we can mention "hill-climbing" methods, the spacefilling curves approach and partitionning methods (for
the PTSP in the plane). However, only two methods among those proposed in Jaillet[1985] retain our attention here for two reasons:
- these are the only ones for which a detailed procedure is suggested in Jaillet [1985].
- they are directly prompted by and adapted from well-known and classic heuristics for the TSP and for routing problems in general.

These two approaches are represented by two algorithms: the Almost Nearest Neighbour Algorithm and the Supersavings Algorithm.

Before going into a detailed description of these two algorithms, we first expose some features of the optimization problem.

2.4.2. A practical view of the optimization problem

We still consider the problem where there is one and only one one black node (the depot). In this case, the expected length of the tour is given by:

$$E[L(t)] = \sum_{r=0}^{n-1} \alpha_r L_1, t^{(r)}$$ (2.8)

where $\forall r \in \{0..n-2\}$, $\alpha_r = p^2(1-p)^r$ and $\alpha_{n-1}=p(1-p)^{n-1}$ (2.9)

One can easily show that $L_{1, t}^{(n-1)}$ is tour independent. Thus one can see that the optimization problem can be decomposed into n-1 problems.
The optimal TSP (denoted \( t_1 \)) tour solves \( \text{Min}(\alpha_{0L_1}, t^{(O)}) \).

The optimal PTSP tour (denoted \( t_p \)) solves \( \text{Min}(\sum_{r=0}^{n-1} \alpha_{rL_1}, t^{(r)}) \).

We denote by \( v_k \) the tour that solves \( \text{Min}(\sum_{r=0}^{k} \alpha_{rL_1}, t^{(r)}) \). Thus \( v_0 \) corresponds to \( t_1 \) and \( v_{n-1} \) corresponds to \( t_p \). We can state that it is always possible to choose \( k \) such that the solution will be a good approximation for the PTSP (trivially true).

Denote such a value of \( k \) as \( k^* \). Then, on a practical basis, \( k^* \) is going to be very small compared to \( n \), the size of the problem. We call the \( \alpha_r \)'s of equation (2.8) the weights of the objective function. They are easily computable. Only \( n-1 \) depends on the size of the problem. For high coverage probability (\( p > 0.5 \)), the \( \alpha_r \)'s decrease sharply with \( r \) and a low value for \( k^* \) (for example 2, 3 or 4) would be suitable.

This also shows that the higher \( p \) is, the closer the PTSP tour will "resemble" the TSP tour. For small coverage probabilities (\( p \leq 0.5 \)), the decrease in the \( \alpha_r \)'s is less steep and therefore we may wish to choose a higher value for \( k^* \). We can conclude that problems involving a low coverage probability may be harder to solve since the associated optimization problem is much more complicated. We will designate \( k^* \) as "the level of optimization".

2.4.3. The Almost Nearest Neighbour Algorithm

The Almost Nearest Neighbour Algorithm (ANNA) is
similar to the well-known Nearest Neighbour Algorithm used in deterministic contexts. In the latter, at each step, the node chosen as the next node on the tour is the closest node to the last node added to the tour. In other words, at each step, the node chosen is the one which minimizes the increase in travel distance. With ANNA, at each step, the node chosen will be the one that will minimize the increase in expected distance. For example, for the optimization problem corresponding to $v_1$, adding node $j+1$ to the path \([\text{depot}, 1, 2, \ldots, j]\) leads to an increase in the expected distance of:

$$\alpha_{0\text{d}(j,j+1)} + \alpha_{1\text{d}(j-1,j+1)}$$

since the increase will be equal to $d(j,j+1)$ for $L_1, t^{(0)}$ and to $d(j-1,j+1)$ for $L_1, t^{(1)}$. In this case we can identify a difference in the tactical behaviour of the two algorithms. Consider figure 2.5. When the deterministic Nearest Neighbour Algorithm will be clearly indifferent between nodes $s$ and $t$, ANNA (with $k^* = 1$) will prefer $s$ to $t$ since $d(s,j-1)$ is less than $d(t,j-1)$. In other words, the choice of $s$ will tend to create the desired zigzag patterns.

More generally, the increase in expected distance for the optimization problem corresponding to $v_k$ will be:

if $k<j$ : $\alpha_{0\text{d}(j,j+1)} + \alpha_{1\text{d}(j-1,j+1)} + \ldots + \alpha_{k\text{d}(j-k,j+1)}$ \hspace{1cm} (2.10)

if $k\geq j$ : $\alpha_{0\text{d}(j,j+1)} + \ldots + \alpha_{j-1\text{d}(1,j-1)} + \alpha_{j\text{d}(\text{depot},j+1)}$
Figure 2.5: An illustration of the tactical difference between the classic Nearest Neighbour Algorithm and ANNA.
Thus, the procedure proposed in Jaillet [1985] is the following:

1) Choose the level of optimization $k^*$ as explained before;
2) Start from the depot, pick up the closest node and label it node 1;
3) Use the minimization of increase in expected distance as a criterion to build step by step the tour.

We label this procedure as the algorithm ANNA0.

A simple variation of this algorithm may also be noted. As a matter of fact, since there is a depot which always requests a visit, the arcs (depot,1) and (depot,n) will have more weight in the expression of the expected distance since they always appear in the $L_{1,t(r)}$'s. This is true in general for all the arcs adjacent to the depot. Thus, the idea is to minimize them first. For this reason, a second procedure is proposed:

1) Choose $k^*$;
2) Find the closest node to the depot and label it node 1;
3) Find the second closest node to the depot and label it node n;
4) Starting from node 1, proceed as in 3) of ANNA0.

We will refer to this procedure as ANNA1.

One can imagine and construct many derivatives and variations of these two procedures. For example, one can grow
simultaneously two trees up to $k^*$ nodes long each, and then join them. However, we believe that these two variations are sufficient to capture the conceptual basis of this approach and to assess its efficiency. If promising, one should definitely go further in this direction.

2.4.4. The Supersavings Algorithm

This approach is based on the famous "savings" method whose basic development is given in Clarke & Wright[1964]. We are interested in the savings in expected distance made possible by merging two subtours. Jaillet derived the following result:

"Given a graph $G$ with $n$ white nodes and one black node (say node $n+1$); consider three subtours of $G$ having only the black node in common and spanning (together) every node of $G$ (see figure 2.6); assume every node is relabelled so that the three subtours are:

$$t_1 = (n+1, 1, 2, \ldots, i_1, n+1)$$
$$t_2 = (n+1, i_1+1, \ldots, i_2, n+1)$$
$$t_3 = (n+1, i_2+1, \ldots, n, n+1)$$

Consider now the "merging" of $t_1$ and $t_2$ in a subtour $t_{12}$ as follows:

$$t_{12} = (n+1, 1, 2, \ldots, i_1, i_1+1, \ldots, i_2, n+1)$$

Then we have:

$$E[L(t_1)] + E[L(t_2)] - E[L(t_{12})] = \sum_{r=0}^{i_2-1} \alpha_r S^{(r)}$$  \hspace{1cm} (2.11)
with \( S(r) = \sum_{k=1}^{u} s(i_1-k, i_1+1+r-k) \)

where \( s(i, j) = d(i, n+1) + d(n+1, j) - d(i, j) \)

\[ l = \max(0, i_1+1+r-i_2) \]

\[ u = \min(r, i_1-1) \]

In general, we call the left term of equation (2.11) the "supersavings" associated with the subtours \( t_1 \) and \( t_2 \). Then, the general idea is to form subtours constituting a partition of the set of nodes, then to compute all the associated supersavings, then to merge these subtours in order to maximize the supersavings realized. Recalling the discussion on the structure of the optimization problem, in solving \( v_1 \), the supersavings associated with linking two end-point nodes \( i \) and \( j \) will be:

\[ \lambda_{0S(0)} + \lambda_{1S(1)} \]

where \( S(0) = s(i, j) \) and \( S(1) = s(i-1, j) + s(i, j+1) \).

Thus, in solving \( v_1 \), it is logical to merge subtours composed of two nodes or more specifically, the general tour being built is merged at each step with a subtour composed of two nodes. In solving \( v_k \), we will merge the general tour and subtours composed of \( k+1 \) nodes. Figure (2.7) illustrates this point.
Figure 2.6: Merging of subtours
Figure 2.7: Merging of pairs of nodes
We can now propose a formulation for the Supersavings algorithm.

A) Choose $k^*$ such that $v_{k^*}$ is a good approximation for the PTSP;

B) To determine the best tour corresponding to $v_{k^*}$, choose the best among the $k^* + 1$ tours obtained as follows:

1) For $v_0$, use the Clarke–Wright Savings Algorithm (CWSA);

2) For $v_1$:
   - form subtours of two nodes (using CWSA);
   - compute all the four possible supersavings associated with each pair of nodes for all these pairs of nodes;
   - order and list those supersavings from largest to smallest;
   - choose the two pairs of nodes associated with the largest supersavings as an embryo of the tour;
   - starting at the top of the list previously derived, form larger subtours by linking pair of nodes until a final tour is formed.

3) For $v_2$:
   - form subtours of three nodes (using CWSA);
   - compute all the supersavings associated with those triplets of nodes;
   - proceed as before namely use the maximization of supersavings as a criterion to link those triplets until a tour is formed.

4) For $v_3$:
   - form subtours of two nodes (using CWSA);
- compute all the associated supersavings;
- form subtours of four nodes based on previous supersavings;
- compute all the new supersavings associated with these quadruplets;
- form the tour based on the supersavings associated with the quadruplets as already exposed.

And so on and so forth....

One can continue this process up to $v_{k^*}$. Yet, for practical purposes, one should probably stop at $k^* = 3$. There is a trade-off between the length of the original subtours and the accuracy of the supersavings computed. We think that three is, as a first step, a very reasonable level of optimization. Within the same algorithm, it leads for each coverage probability to the comparison of four different tour-solutions.

We denote $SAVEk^*$ the heuristic that solves $v_{k^*}$, i.e. $SAVE0$ is the CWSA etc....If this approach proves promising, we may increase $k^*$.

2.4.5. Another heuristic based on the Supersavings approach

The two first procedures we have described are directly derived from propositions explicitly made in Jaillet[1985]. Using the same kind of ideas, it is possible to derive some other simple methods. The one we propose now is a straightforward adaptation of the CWSA. In CWSA, at each step, the objective function is to maximize the savings in
distance. In our approach, we set an objective function which maximizes the savings in expected distance (the so-called supersavings). Based on the discussion of 2.4.3., consider the subtour \( t_1 \):

\[
t_1 = [\text{depot}, 1, 2, \ldots, j-1, j, \text{depot}]
\]

and the subtour \( t_2 \) composed of only node \( j+1 \):

\[
t_2 = [\text{depot}, j+1, \text{depot}]
\]

There are two possible ways to merge \( t_1 \) and \( t_2 \). We may obtain either:

\[
[\text{depot}, 1, 2, \ldots, j-1, j, j+1, \text{depot}]
\]

or

\[
[\text{depot}, j+1, 1, 2, \ldots, j-1, j, \text{depot}].
\]

For the optimization problem corresponding to \( v_1 \), adding node \( j+1 \) to \( t_1 \) leads to two possible supersavings:

\[
\alpha_{0s}(j, j+1) + \alpha_{1s}(j-1, j+1)
\]

or

\[
\alpha_{0s}(1, j+1) + \alpha_{1s}(2, j+1)
\]

where in general \( s(k, 1) \) is the usual savings in distance.

More generally, for \( v_k \), the computation of the objective function will be:

\[
\alpha_{0s}(j, j+1) + \alpha_{1s}(j-1, j+1) + \ldots + \alpha_{k-1s}(j-k, j+1)
\]

if \( k < j \)

or

\[
\alpha_{0s}(1, j+1) + \alpha_{1s}(2, j+1) + \ldots + \alpha_{k-1s}(k+1, j+1)
\]

if \( k = j \)

or

\[
\alpha_{0s}(j, j+1) + \ldots + \alpha_{j-1s}(1, j+1)
\]

if \( k > j \)
Thus this new procedure is formulated as follows:

1) Choose the level of optimization $k^*$ as already exposed;
2) Compute all the classic savings;
3) Choose the pair of nodes associated with the largest savings as an embryo of the tour;
4) Use the maximization of the savings in expected distance as a criterion to build the tour.

We denote NEWSAVE this heuristic.

2.4.6. The manual approach

If any formal argument is still needed, the inadequacy of dynamic programming for solving the PTSP (see Jaillet[1985], pp.157-162) offers convincing evidence that the PTSP can best be solved if one considers the problem as a whole. Computerized heuristic algorithms are usually myopic by construction. At each step, they try to optimize a secondary objective function but they mostly ignore what is to come or what is left over.

Yet, in the Euclidean case, there exists one tool that provides a general view at the problem and allows a good understanding of the structure and the configuration of the set of nodes. It is simply the human eye. Thus, we consider a tour construction "by hand" (or "by eye").

Based on the discussion of section 2.3., we draw the tour following as much as possible a zigzag pattern. We avoid alignments of nodes that do not lead to reductions in
travelled distance. The general shape of the tour must be jagged. Using our understanding of the problem and our experience, we seek a compromise between a tour of short length (that would behave well for \( p \) close to 1) and a tour including a lot of zigzags (that would supposedly behave well for small values of the coverage probability).

One can notice that a priori this procedure is not \( p \)-dependent since we propose only one tour. However, we think that one cannot be too specific with such a method. The physical meaning of \( p = 0.5 \) is quite obvious. It may be on the other hand very challenging to figure out the behaviour of a tour under a coverage probability of 0.1. We hope that the tour so constructed will "behave well" in general and that this experience will improve our understanding of the problem.

2.5. Some test cases

2.5.1. Computational results

ANNA0, ANNA1, NEWSAVE and the four SAVE's are implemented on a microcomputer, namely an IBM PC AT. The computer code is written in Turbo Pascal. The memory capacity allows us to handle problems up to 100 nodes. For 8 different problems (see Appendix A for the details of the problems), we first drew a tour based on MANUAL. Then we executed the different computerized algorithms for 10 values of the coverage probability (from 0.1 to 1).

In Table 2.2 we list the number of times (out of 8) a
method gives the best results for a given coverage probability. In Table 2.3, we do the same but with the worst performances. The sum in some rows may exceed 8 indicating that some methods perform equally well or poorly in some cases.

Table 2.2: Number of best performances

<table>
<thead>
<tr>
<th></th>
<th>MANUAL</th>
<th>ANNA0</th>
<th>ANNA1</th>
<th>NEWSAVE</th>
<th>SAVE0</th>
<th>SAVE1</th>
<th>SAVE2</th>
<th>SAVE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td></td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>+++++</td>
<td>+++++</td>
<td>+++++</td>
</tr>
<tr>
<td>0.9</td>
<td>6</td>
<td></td>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>6</td>
<td></td>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>5</td>
<td></td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>6</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>5</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>4</td>
<td></td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Table 2.3: Number of worst performances

<table>
<thead>
<tr>
<th>p</th>
<th>MANUAL</th>
<th>ANNA0</th>
<th>ANNA1</th>
<th>NEWSAVE</th>
<th>SAVE0</th>
<th>SAVE1</th>
<th>SAVE2</th>
<th>SAVE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>+++++</td>
<td>+++++</td>
<td>+++++</td>
</tr>
<tr>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>0.6</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>0.5</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>0.4</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0.3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.2</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.5.2. Analysis and comments

First consider Table 2.3. It is clear that ANNA0 and ANNA1 behave badly for the whole probability range. The examination of the tours produced explains this poor performance. Two points may be highlighted:

- during the construction of a tour, some points are temporarily forgotten. They are left over. They are like "stranded". They are recovered at the end of the construction. Thus, the path may cross several times the node field. It results in a considerable increase in the length of the tour which therefore is not compact at all.

- the patterns observed seem pretty erratic. The tour tends to make loops, to intereset itself many times. For relatively low value of p, the tour often makes some kind of "turns". All those loops, turns and "contortions" lead to a large increase in the length of the tour which reverberates on the expected length of the tour.

A very undesirable consequence of this behaviour is that in some cases (problems 2, 5 and 8), the expected distances of the tour-solutions produced for p = 0.9 are even greater than the simple lengths of the tours produced by the algorithms for p = 1 when one should expect these lengths to be at least the upper bounds for the solutions of the problems. Figure 2.8 based on problem 3 illustrates the poor behaviour of these algorithms. For p = 1 and problem 1 (the refuse collection example of Larson and Odoni[1981]), ANNA1 gives the best performance as well as MANUAL and SAVEO which
ANNAO with $p=0.1$

ANNAO with $p=0.5$

Figure 2.8: Illustration of the poor behaviour of ANNA

42
produce the same tour. This single good result obtained for a ten-nodes problem is not significant.

Finally, the Almost Nearest Neighbour Algorithm should be rejected despite its a priori attractive and simple formulation. Its essential weakness is its myopicity. We do not think that it is worthwhile to try to improve it.

SAVE3 also behaves poorly. The only case where it behaves better is problem 2 with p = 0.1. This not really significant. SAVE2 is never really good or bad. These two procedures may also be discarded.

Now consider Table 2.2. It appears clearly that MANUAL, NEWSAVE and also SAVE1 but to a lesser extent give the best results. For p >= 0.5, MANUAL dominates the other procedures. For p<0.5, MANUAL behaves as well as NEWSAVE. SAVE1's better performances are obtained for low and high values of the coverage probability but SAVE1 is dominated in general by the two previously cited procedures. SAVE0 (Clarke-Wright) is never truly good or bad, and therefore must be discarded.

The good behaviour of MANUAL confirms to some extent our good understanding of the PTSP. We wanted to draw "tours of short length with as many zigzags as possible", "tours with jagged general patterns". We give in Figure 2.9 the tour drawn for problem 7. This tour dominates all the other solutions we have obtained by any other methods for the whole probability range but for p=1 where the CWSA gives a better tour. The tour produced by CWSA is of shorter length but it behaves very poorly in the PTSP sense. This tour illustrates
Figure 2.9: Tour drawn for problem 7 by means of MANUAL
Figure 2.10: Tour obtained for problem 7 by means of CNSA
what a "non desirable" pattern is. This tour is shown in figure 2.10. However, one must keep in mind two important points.

First MANUAL is a subjective procedure that can only be applied to Euclidean problems. It is operator-dependent and cannot be exactly reproduced even with the same operator. A single person trying to draw a path at different times would certainly come out with different results (we must recall that this procedure has been used with regard to providing a better understanding of the PTSP and not a priori as an effective procedure in itself).

The second point is that in many cases, the superiority of MANUAL derives partially from the fact that it often provides a very good solution for the TSP. As a matter of fact, in the expression for the expected distance, the simple length of the tour \( L_{1,t}^{(0)} \) has the heaviest weight (100*p% exactly) whereas the weights of the other quantities (the \( L_{1,t}^{(r)} \)'s) decrease fast to zero for \( p >= 0.5 \). We can suppose that in this case the influence of this first term is such that it allows the expected distance distance to remain the lowest in this range. That would explain why MANUAL work best for \( p >= 0.5 \). So, for the test problems we have considered, any algorithm or procedure that would provide a good solution for the TSP would be likely to give a good solution for this probability range. For problems involving special features as symmetry for example (the star-shaped example), it may not be true.
So, we can expect to be able to improve our current solutions for \( p \geq 0.5 \) since it is surely possible to find better solutions for the TSP than the ones we propose by means of MANUAL. As a matter of fact, if it were possible to solve the TSP optimally "by hand", one would know this (see Krolak[1971] for an interesting man-machine interactive procedure to solve the TSP).

For \( p < 0.5 \), NEWSAVE produces good tours and behaves as well as MANUAL. In the next section, we concentrate on this procedure and we show how to greatly improve it.

2.6. NEWSAVE: a global procedure for the PTSP

2.6.1. The performance of NEWSAVE

We now consider NEWSAVE. Contrary to MANUAL, its solutions are \( p \)-dependent. The question that may arise is whether a tour obtained with a given coverage probability, say \( p_1 \), could be a good tour for another value, say \( p_2 \), of the coverage probability.

Thus, for each of the solutions given by NEWSAVE, we compute the 9 other expected distances (recall that we have ten different values for the coverage probability). If there is a decrease in the expected distance, we retain the tour concerned as the new best solution. Then, gathering the results, we build Table 2.4 where the number of best solutions for each algorithm after these new computations is indicated.

One can see that the performance of NEWSAVE is greatly
improved. NEWSAVE dominates the other procedure for \( p \leq 0.6 \) and behaves as well as MANUAL for \( p > 0.6 \). SAVE1 is now somewhat marginal in terms of efficiency.

<table>
<thead>
<tr>
<th>( p )</th>
<th>MANUAL</th>
<th>ANNA0</th>
<th>ANNA1</th>
<th>NEWSAVE</th>
<th>SAVED</th>
<th>SAVE1</th>
<th>SAVE2</th>
<th>SAVE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td></td>
<td>1</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>4</td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>4</td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>4</td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4 : Number of times the algorithms perform best after improvement

What happened is that when NEWSAVE gives the best tour for a given \( p \), this same tour tends also to be the best solution for the immediate neighboring values of the coverage probability. It is especially true for the upper values. This effect vanishes somewhat when \( p \) is increased too much. Curiously, it hardly improves, if at all, the expected
distances for lower values of \( p \). For example, in problem 3, the best tour we find for \( p = 0.5 \) is obtained by solving the problem with \( p = 0.2 \). Likewise, the best solution for \( p = 0.9 \) is given by solving the problem for \( p = 0.4 \). Complete and detailed results for problem 3 and the drawing of the solutions obtained are given in Appendix B.

It is difficult at this stage to explain the reasons that cause such a behaviour. Rather than speculating on the inadequacy of the heuristic or on the very complex structure of the space of the solutions, we take this behaviour as a fact, an interesting one since it allows us to improve our tour-solutions.

2.6.2. A global procedure for the PTSP based on NEWSAVE

The previous discussion leads us to propose a more global heuristic to solve the PTSP. Rather than trying to solve the problem for a given probability, the problem will be attacked as a whole (in terms of the entire probability range). The new procedure is formulated as follows:

1) Define a discrete set of coverage probabilities in the range \([0..1] \): \( \{p_1, p_2, \ldots, p_i, \ldots, p_n\} \)

2) Obtain by any available method a good solution for the TSP associated with this set of nodes;

3) For each \( p_i \) of the \( p_i \)'s, compute the expected distance of the tour, at this stage, each \( p_i \) is associated with the same tour;

4) For each \( p_i \) do:
- solve the problem by means of NEWSAVE;
- compute all the expected distances for the new tour obtained for all the \( p_i \)'s;
- if, for any \( p_i \), the expected distance is less than that of the tour previously associated with that \( p_i \), discard the old tour and associate the new tour with \( p_i \).

So, at the end, one comes out with \( n \) tours each associated with a given probability \( p_i \). We label this procedure GLOBALSAVE.

Using the ideas and the methods provided in Golden[1976], it is possible to obtain variations of this algorithm and to handle large size problems. A simple variation would be obtained not by considering the classic savings in distance \( s(i,j) \) for nodes \( i \) and \( j \):

\[
s(i,j) = d(\text{depot},i) + d(\text{depot},j) - d(i,j)
\]

but slightly different savings \( s'(i,j) \) defined by:

\[
s'(i,j) = d(\text{depot},i) + d(\text{depot},j) - m d(i,j)
\]

where \( m \) is a parameter.

According to the value of \( m \), we may enhance or not the importance of the internodal distance. There is no a priori method for determining a best value for \( m \) for any given problem. One can use different \( m \)'s and then choose the best solution produced. We leave this area for future work.
2.7. Conclusions

In this chapter, we first reviewed the PTSP formulation as designed by Jaillet. Then we defined what a good tour "in the PTSP sense" could be and accordingly we proposed a manual procedure to solve the problem. We assessed the efficiency of two heuristics derived from Jaillet[1985] (ANNA and SAVE), of a manual procedure (MANUAL) and of another method we designed (NEWSAVE). We showed that ANNA and SAVE behaved poorly, that MANUAL was dominant for coverage probabilities greater than 0.5 and that NEWSAVE behaved well in the remaining part of the probability range. Then we noticed that a tour produced by NEWSAVE for a given value of $p$ could also be a very good tour for another value of the coverage probability. Accordingly, we designed a new procedure where the problem was no longer solved for a given $p$ but rather was solved as a whole in term of the entire probability range. This last procedure based on NEWSAVE produced very good results. It has been called GLOBALSAVE.
CHAPTER 3

THE PROBABILISTIC VEHICLE ROUTING PROBLEM

3.1. Introduction

In Chapter 2, we introduced the PTSP, we examined some of its occasionally surprising features and we described what a good tour might be "in the PTSP sense". We also assessed the efficiency of several different heuristics and finally we proposed a global procedure that seems to be promising according to our computational results.

In this chapter, we formulate the Probabilistic Vehicle Routing Problem (PVRP). First we review another approach to vehicle routing problems under uncertainty and then we point out why the PTSP formulation leads us to consider the PVRP. In a second part, we propose a heuristic directly derived from GLOBALSAVE. In a last part we extend the PVRP formulation to more general cases.

3.2. Brief review of another attack on the problem: the SVRP

The Stochastic Vehicle Routing Problem (SVRP) is a generalization of the standard Vehicle Routing Problem (VRP) where the demands at each node are random variables with a known probability distribution. The problem is to find a
priori a set of routes of minimum expected distance. A very
good source on this problem can be found in Stewart[1981].

This problem is usually formulated by means of
chance-constrained models which render the problem
deterministic. Thus the solution-procedures used are directly
derived from the ones applied for the VRP. In Stewart and
Golden[1982], algorithms using a penalty function procedure
are presented.

3.3. Formulation of the PVRP

Like the SVRP, the PVRP is a vehicle routing problem
where the demands at each node are non-deterministic. However,
the PVRP will be directly derived from the PTSP formulation
and contrary to the SVRP, we will actually compute the
expected distances of the routes we want to minimize.

More specifically, in a first step, we consider the case
where these demands can take either the values 0 or 1. 0
corresponds to the case where the node does not request a
visit and accordingly is skipped in the tour as in the PTSP.
If the node is to be visited, the quantity to be serviced is
one unit. Thus, the problem consists of determining a set of
a priori routes through the set of nodes to operate our fleet
of vehicles with limited capacity such that these routes are
of minimum total expected distance. One can notice that the
PVRP we consider is a kind of m-PTSP. However, as explained
in the following, this is not exactly a m-PTSP.

Consider now a given route (or subtour) serviced by only one vehicle of limited capacity. We know that on any given instance of the problem, some nodes may be missing and thus will not be visited. In other words, the number of points actually visited will always be less than or equal to the total number of points in the subtour. So, it could be interesting to design routes for which the maximum number of points on the route exceeds the capacity of the vehicle (since at each node there is potentially unit demand). We want to take such a risk to achieve a better efficiency. Such a policy may actually allow us to reduce the size of our fleet and the associated costs.

"On the average", because some nodes will typically be missing, the vehicle will be able to visit all the points in a given instance of the tour in one pass. However, in some cases, the total demand on a given instance of the tour under this approach may exceed the capacity of the vehicle. Assuming that a client requesting service and not being visited is a client lost, we want to achieve a level of service of 100%. We follow this operating policy: if on a given day, on a given route, capacity is reached and if there are still some other nodes left to be visited, the vehicle goes back to the depot (to perform any requested operations such as unloading or reloading the vehicle), and then continues the tour starting with the first node in the a priori tour that has not yet been visited. We can quantitatively define a
general level of risk associated with each subtour which is simply the probability that the number of nodes requesting service exceeds the capacity of the vehicle.

Following this operating policy, some extra distance may be travelled by the vehicle and to be consistent, one would be willing to compute its expected value. We will see in the next section that this operating policy introduces an important complication in the computation of the expected distance.

Based on the previous description, the PVRP is formulated as follows: based at a depot a fleet of vehicles each with the same limited capacity, say $K$, services a set of nodes. At each node, either one unit is requested with probability $p$ or none with probability $(1-p)$. In the latter case, the node is simply skipped. All the points are independent. The problem consists of designing routes such that the expected total distance travelled by the fleet is minimized, given that the general level of risk is less than a given level. More formally:

$$\text{Min}\{ \text{expected total distance} \}$$

s.t. $\text{Prob[ for any subtour, demand exceeds capacity ]} < \text{Risk}$

We note $N$ the number of points in a given subtour. In general, if $(m-1)K < N ≤ mK$, the vehicle which services the tour would have in the worst case to go back $m-1$ times to the depot. We limit our research to the case where $N ≤ 2K$ when, at most, the vehicle may have to go back on time to the depot. Even in this case, we show that the derivation of a closed
form expression for the expected distance is hardly tractable.
We derive in the next section an upper bound for the total expected distance. For m>2, we have not obtained any similar results.

3.4. An upper bound for the total expected length

We consider a given subtour and use the following notation:
- N is the number of points in the subtour,
- K is the capacity of the vehicle ( N <= 2K ),
- k is the number of nodes which do not request service on a given instance of the subtour, note that k follows a binomial distribution,
- p is the coverage probability.

For k >= N-K , the vehicle is able to visit all the points present. The conditional expected distances are directly given by equation (2.3).
For k <N-K , the vehicle is not able to visit all the points in one trip. We have to take account of this fact and introduce some modifications in the expressions for the conditional expected distances. For example, if all nodes are present:

\[ E[L(t)/\text{no node missing}] = L_{1},t^{(0)} + s(K,K+1) \]

where
\[ s(K,K+1) = d(\text{depot},K) + d(\text{depot},K+1) - d(K,K+1) \]

In general, the remaining conditional expected distances are composed of three terms:

- the usual conditional expected distances for the entire subtour.

- a first correction term due to the fact that some terms of the distance matrix are missing in the \( L_{1,t}^{(r)} \)'s already computed. For example, it is easy to show that, whatever the configuration of the problem may be, if \( k \) nodes are missing, it is impossible to go directly from node \( K \) to node \( K+k+1 \). As a matter of fact, even if all the \( k \) missing nodes are located between node \( K \) and node \( K+k+1 \), this implies that the vehicle has reached capacity at node \( K \). Therefore, at this point, it would go back to the depot and then continue the tour starting with node \( K+k+1 \). The distance \( K,K+k+1 \) is thus never travelled. We must not take account of the term \( d(K,K+k+1) \) which is normally part of the computation of the expected distance. This kind of reasoning can be extended to some other terms.

- another correction term due to the fact that the vehicle has to travel an extra distance going back and forth to the depot.

Let us first derive this last quantity.

The vehicle has to go back to the depot. It can be easily realized that we have to count how many times terms of the type \( d(\text{depot},K+1) \) will appear for \( i \) belonging to the range
Consider the term \( d(\text{depot}, K+i) \). In this case, \( i \) nodes among the first \((K+i-1)\) nodes of the tour are missing and \((k-i)\) nodes among the \((N-K-i)\) remaining are missing. So there are \( \binom{K+i-1}{i} \binom{N-K-i}{k-i} \) different ways for \( d(\text{depot}, K+i) \) to appear.

Now, the vehicle returns to servicing nodes in the tour. For \( i \) belonging to the range \([1..K+1]\), we want to count how many times the generic term \( d(\text{depot}, K+i) \) appears. In this case, \((i-1)\) nodes are missing among the \((K+i-1)\) first nodes and \((k-i+1)\) must be missing among the \((N-K-i)\) remaining nodes.

So there are \( \binom{K+i-1}{i-1} \binom{N-K-i}{k-i+1} \) different ways for \( d(\text{depot}, K+i) \) to appear. Figure 3.1 illustrates this point. We use the following notation:

Given \( a, b \geq 0, \binom{a}{-b} = 0 \) and \( \binom{a}{b} = 0 \) if \( b > a \)

Thus for a given \( k \), the conditional extra distance \( ED(k) \) is given by equation (3.1):

\[
ED(k) = \sum_{i=0}^{k+1} d(\text{depot}, K+i) \left[ \binom{K+i-1}{i} \binom{N-K-i}{k-i} + \binom{K+i-1}{i-1} \binom{N-K-i}{k-i+1} \right]
\]

Let us derive now the first correction term which is a perturbation term. The operating policy that eventually forces the vehicle to go back and forth to the depot not only introduces an extra distance to be travelled but also perturbs the computation of the normal conditional expected distances. We may recall that the expected distance is basically computed
Figure 3.1: Travelling an extra distance
in two steps. First the quantities \( L_{1,t}^{r}(r) \) are formed. Then one counts how many times those quantities appear. Now, the fact is that some elements of the distance matrix that would normally be part of the \( L_{1,t}^{r}(r) \)'s cannot be considered at all since they cannot appear in any actual configuration (like \( d(K,K+k+1) \) in \( L_{1,t}^{(k)} \) for example). Others do not appear as many times as they would in the original infinite vehicle capacity case. The terms concerned are the ones involving nodes located after node \( K \) on the subtour. An exact enumeration can be undertaken for \( L_{1,t}^{(0)} \) and \( L_{1,t}^{(k)} \) but it becomes extremely complicated otherwise and accordingly, a complete derivation has not been fully made. We note \( P(k) \) the perturbation term with \( P(0) = d(K,K+1) \).

We can now write the expected total distance covered:

\[
E[L(t) \text{ with } K] = E[L(t)] + \sum_{k=0}^{N-K-1} (ED(k) - P(k))(1-p)^k p^{N-k} \quad (3.2)
\]

where \( E[L(t)] \) is given by equation (2.6), \( ED(k) \) is known and \( P(k) \) is not fully derived.

Clearly, we can obtain now a lower bound and an upper bound for the expected distance of the tour under our operating policy. The lower bound is simply \( E[L(t)] \). As a matter of fact, forcing the vehicle to go back and forth is like adding a kind of black node to the tour. It cannot reduce the expected distance.

The upper bound is given by:
\[ E[L(t)] = \sum_{k=0}^{N-K-1} ED(k)(1-p)^k p^{N-k} - p^N d(K, K+1) \]  

(3.3)

This upper bound is so far the only expression we have that relates the capacity K and the expected length of the tour. One can notice that the value of the bound depends on the way one travels through the tour since the expression is not symmetric with respect to the tour. The extra distance is all the smaller as the nodes located after the Kth node (counted in the way the tour is travelled through) are close to the depot.

Thus, whenever we build a subtour, we will compute the upper bound for the two possible ways and indicate in which one the lowest result is obtained.

3.5. The level of risk

In this section we take a closer look at what we defined as the general level of risk. We examine some practical implications on the size of the subtours we want to build.

The general level of risk is the probability that the vehicle will not be able to visit all the nodes present in only one pass. Or, in other words, it is the probability that for a given subtour on a given day, the number of nodes requesting service exceeds the capacity of the vehicle. With our assumption that the number of nodes requesting service follows a binomial distribution, it can be expressed as:
Level of risk = \[ \sum_{i=1}^{N-K} \text{Prob}[K+i \text{ nodes present out of } N] \]

Level of risk = \[ \sum_{i=1}^{N-K} N^{K+i}(1-p)^{N-K-i} \]

The constraint becomes:

\[ \sum_{i=1}^{N-K} N^{K+i}(1-p)^{N-K-i} < R \]

Given \( K \) and \( R \), the maximum risk allowable, the question is how many nodes to include in the subtour such that the constraint is not violated. In order to be more specific, let us consider the following example where \( p=0.7, K = 4 \) and \( R = 10\% \).

For \( N = 5 \), \( \text{Prob}[\text{capacity exceeded}] = 0.168 \),

for \( N = 6 \), \( \text{Prob}[\text{capacity exceeded}] = 0.42 \),

for \( N = 7 \), \( \text{Prob}[\text{capacity exceeded}] = 0.647 \).

Clearly in this case, we cannot take any risk. As soon as the number of points in the subtour is greater than the vehicle's capacity, the constraint (Level of risk < R) is violated. So, given \( R \) and \( p \), we are interested in the minimum size of the problem that allows us to take some risk. To appreciate this point we derive simple results shown in Table 3.1 which give, for different \( R \)'s and \( p \)'s, a lower bound for the minimum size of the problem for a risk to be taken.

For example, it can be seen that with \( R = 0.1 \) and \( p = .7 \), one must consider \( K = 6 \) and \( N = 7 \) at least (we have derived a
Table 3.1: Minimum sizes for the routing problem

<table>
<thead>
<tr>
<th>( p )</th>
<th>( N_{\text{min}} )</th>
<th>( K_{\text{min}} )</th>
<th>( N_{\text{min}} )</th>
<th>( K_{\text{min}} )</th>
<th>( N_{\text{min}} )</th>
<th>( K_{\text{min}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>0.55</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>0.6</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>0.65</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>0.7</td>
<td>7</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>0.75</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>10</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td>0.8</td>
<td>11</td>
<td>10</td>
<td>14</td>
<td>13</td>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td>0.85</td>
<td>15</td>
<td>14</td>
<td>19</td>
<td>18</td>
<td>29</td>
<td>28</td>
</tr>
<tr>
<td>0.9</td>
<td>22</td>
<td>21</td>
<td>29</td>
<td>28</td>
<td>44</td>
<td>43</td>
</tr>
<tr>
<td>0.95</td>
<td>45</td>
<td>44</td>
<td>59</td>
<td>58</td>
<td>90</td>
<td>89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( R = 10% )</th>
<th>( R = 5% )</th>
<th>( R = 1% )</th>
</tr>
</thead>
</table>

lower bound. From Table 3.1, we can see that in many cases, the constraint on the level of risk dictates the design of routes for which the number of points will be equal to the capacity of the vehicle. This will be especially true for high values of the coverage probability. In this case, we are able to compute exactly the expected length of the subtour. Another implication is that, to take risks, it is necessary to have vehicles of large capacity.
Thus, to set the size of the subtour, we compute the level of risk with hypothetical subtours of size \( K + i \), \( i = 1, 2, \ldots \) until either:
- the constraint is violated, or
- the level of risk reaches a maximum and then decreases without violating the constraint.

In any procedure to solve the PVRP, the first step would be to determine \( N \), the maximum number of nodes a subtour can have such that the constraint on the level of risk is not violated. However, \( N \) should only be a benchmark. There may exist specific configurations where it would make more sense to build a subtour with less than \( N \) nodes. It might also make sense to slightly violate the constraint if required by a "common sense" approach. We have to adapt any approach to the actual structure and configuration of the problem we may attempt to solve.

3.6. A heuristic for the PVRP

In a deterministic environment, several strategies that may be followed to solve the one depot, multiple vehicle, node-covering problem can be identified. The classification given in Bodin et al[1983] includes seven different approaches: 1) cluster first, route second 2) route first, cluster second 3) savings/insertion 4) improvement/exchange 5) mathematical-programming-based 6) interactive optimization 7) exact procedures.
Obviously, this classification can also be used for the non-deterministic case. The most efficient heuristic we have for the PTSP, namely NEWSAVE, is a savings based procedure. It is a sequential method where at each step we choose the node that maximizes the savings in expected distance (the supersavings) associated with the tour being built. Thus the tour grows at its two ends.

One can easily derive from NEWSAVE a procedure that can be applied for the PVRP. The tour construction process can be stopped at any stage, as soon as the number of nodes included in the tour equal the predetermined size of the subtour, or as soon as the constraint on the level of risk is violated.

An alternative procedure would be to use a route first, cluster second strategy. In a first step we would build a grand tour by means of NEWSAVE (or rather GLOBALSAVE). Then we would break this tour into several subtours of adequate size. For example, if tour \( t \) composed of \( N \) nodes were to be broken into tours \( t_1 \) and \( t_2 \), we would compute the relationship between the different expected distances:

\[
E[L(t)] + \left( \text{supersavings associated with the merging of } t_1 \text{ and } t_2 \right) = E[L(t_1)] + E[L(t_2)]
\]

The exact expressions for these supersavings would be easily computable as indicated in 2.4.4. \( t_1 \) and \( t_2 \) could then be further subdivided into smaller subtours and we could define a partition of the grand tour as a solution to the problem. The
point is that nothing guarantees that this tour construction procedure minimizes the supersavings associated with the merging of the subtours obtained. Therefore nothing guarantees that the expected lengths of the subtours are actually minimized. This alternative approach should be mentioned but, for that reason, we do not consider it further.

In conclusion, the procedure we propose is formulated as follows:

1) Compute the maximum number of points that can be permitted in a subtour, namely $N_{\text{max}}$;

2) Set the size of all the subtours to be equal to $N_{\text{max}}$ (but one that may be equal to the remaining of the division of the total number of nodes $n$ in the problem by $N_{\text{max}}$);

3) Based on the actual configuration of the set of nodes, eventually modify the sizes of the subtours;

4) Define a probability range and a set of discrete coverage probabilities as in NEWSAVE;

5) For each subtour, while the size of the subtour is not reached, apply NEWSAVE for each value in the probability range and retain the best solution for the probability that is originally considered;

6) Compute the bounds for each subtours.

As can be seen, this procedure uses a composite strategy. The first third steps are related to a cluster first, route second approach, as we set the sizes of the subtours. The
last third ones are based on the supersavings approach described at the end of the second chapter. We do not solve the problem for all the values of the probability range but for one value only. This is because each value of the coverage probability \( p \) is associated with a certain level of risk that determines the size of the subtours.

In the next section, we consider how the formulation of the PVRP can be extended and applied to more general cases.

3.7. A generalisation for the PVRP

3.7.1. The different possible extensions of the formulation

Let us recall the basic hypothesis of our formulation. We assumed:

- a binomial distribution for the number of points requesting service,
- the same coverage probability for all the points,
- the fact that the quantities serviced at each node could either be 0 or one unit.

The first two assumptions are relevant to the PTSP and all its derivatives in general. The third one is more specific to the PVRP itself. These assumptions may appear somewhat restrictive. We shall consider a relaxation of these hypothesis and show that in the context of the PTSP (PVRP) formulation, it is possible to adapt our methods and formulations to more general cases.
3.7.2. The coverage probability

So far, it has been assumed that all the members of the set of nodes requested service with the same probability \( p \) independently of each other. In this case, the probability distribution for the number of points requesting a visit is the binomial one.

Consider now the case where some points do not request service with probability \( p \) but rather with probability \( p' \). For example, assume a coverage probability of 0.5 for all the nodes but one, which requests visits with probability 0.75. If we replace this last node by two superimposed nodes each requesting service independently with probability 0.5, we can compute the probability \( P \) that this particular point requests a visit:

\[
P = 1 - \text{Prob}[\text{the point does not request service}]
\]
\[
P = 1 - \text{Prob}[\text{both superimposed nodes do not request service}]
\]
\[
P = 1 - (1-0.5)^2 = 0.75
\]

Thus by replacing this particular node by two identical nodes, we obtain a new point that behaves as the old one in terms of requesting service and we have retrieved our binomial distribution since now all nodes request service with the same probability again.

This "trick" can be used in more general cases as mentioned in Jai1et[1985],p. 46. The idea is that for any set of nodes labelled \( i \) ( \( i \in [1..n] \) ) requesting service independently of each other with probability \( p_i \), \( i \in [1..n] \), it is possible to replace each node by \( n_i \) nodes superimposed.
each requesting service with probability $P_0$, such that:

$$P_1 = 1 - (1 - P_0)^N.$$ 

We may have either a good approximation or an exact reproduction of the original problem where our original assumptions are met.

The major drawback of this approach is that it may increase the total number of nodes considerably. It is even impossible a priori to define an upper bound for the size of the problem in the worst case. This would increase the computational burden as well as the storage requirements for the execution of the algorithms. However, we think that this extension is very interesting and that it may allow to handle easily a considerable expanded range of applications.

3.7.3. The probability distribution

Consider now the case where the probability distribution for the number of nodes $W$ requesting service is not binomial but of some general kind (and exactly known). As shown in Jaillet[1985], it is possible to derive a closed-form expression for the expected length of such a tour under this new assumption:

$$E[L(t)] = \sum_{r=0}^{n-2} L_r(r) \left( \sum_{k=r}^{n-2} \binom{n-2-r}{k-r} \binom{n}{k} \text{Prob}[W=n-k] \right)$$ (3.5)

The probabilistic weights can also be rapidly computed and therefore our heuristics for the PTSP are still valid.
For the PVRP, the computation of the expected lengths of the subtours and the level of risk now require more detailed attention and the derivation of an upperbound is burdensome but can still be undertaken.

Basically, the introduction of a general probabilistic distribution does not change either our approach or our solution procedures as long as the nodes are indistinguishable.

3.7.4. More general assumptions on the quantities demanded

So far, it has been assumed that at each node the quantities took either the values 0 or 1. We assume now that each node \( i \in [1..n] \) requests either 0 or \( Q_i \) units.

If all the \( Q_i \)'s are equal and if the capacity \( K \) is a multiple of \( Q_i \), the problem is exactly the same as in our basic formulation.

If all the \( Q_i \)'s are equal but if the capacity \( K \) is not a multiple of \( Q_i \), the problem still remains the same. We have a surplus of capacity that cannot be used in any way. If \( K \) would be equal to the immediately smaller multiple of \( Q_i \), it would yield the same results.

Suppose now that at each node \( i \in [1..n] \), the demand is either 0 or \( Q_i \) units where the \( Q_i \)'s have no special properties. Two difficulties with regard to the PVRP formulation arise as soon as we try to take a risk, i.e., as soon as we design routes where the total demand exceeds the
capacity of the vehicle. If we do not want to take risks, we can simply apply the proposed procedure for the PVRP. The vehicle never has to go back to the depot and we can compute the exact values of the expected lengths of the subtours. However, it would make little sense to design such routes and not to take advantage of the fact that typically some nodes are going to be missing and that the actual demand will always be less than or equal to the total potential demand.

The first difficulty lies in the fact that it is impossible to compute \textit{à priori} a general level of risk. The composition of each subtour must be known and even with this information, the computation of the level of risk would require a complete enumeration of all the possible cases.

Likewise, the second difficulty is that it would also be burdensome to derive an upperbound for the expected length of the subtour. So the problem is to find a way to determine the sizes of the subtours, and to find a measure through which to evaluate quantitatively the subtours built.

One way to handle those two difficulties is the following. Let $Q_m$ be the average of the $Q_i$'s. Then one can compute a "virtual risk" assuming that all the $Q_i$'s take a single value $Q_m$. Thus, one can set a "virtual" maximum size of a subtour. This "virtual risk" is an estimate of the actual risk which becomes less reliable as the standard deviation of the $Q_i$'s becomes larger. Likewise, once the tour is obtained, we can set the upper bound derived in the first formulation as an estimate of the actual expected length of
the tour. We would have to compute the "virtual capacity" $K_v$, $K_v = \text{Int}(k/Q_m)$. Clearly, a lower bound is still given by the expected length of the tour in the PTSP sense.

The procedure proposed is thus the following:

1) Compute $Q_m$;
2) Compute the virtual capacity $K_v = \text{Int}(K/Q_m)$;
3) Set the sizes of the subtours as for the strict PVRP;
4) Solve the problem as for the strict PVRP;
5) Compute the estimates of the expected lengths of the subtours.

One can imagine another procedure where the construction of each subtour is stopped when the potential demand of the subtour, i.e., the sum of the demands associated with each of the nodes included in the subtour, exceeds a given level. This level might be a function of the capacity of the vehicle, of the demands themselves or of any related quantity. Such a statistical approach would require an exhaustive review of an important number of test cases. We leave this area for future work.

3.8. Conclusions

In this chapter, we presented an application of the PTSP formulation to vehicle routing problems and thus we defined the PVRP. We showed that this problem was more complicated than the "simple" PTSP. We introduced the notion of risk and we computed an upper bound for the expected distance of a
tour. Finally, we presented some possible extensions of the formulation of the PVRP in order to handle more general cases.
CHAPTER 4

CONCLUSIONS

4.1. Summary

This thesis has presented an examination of the PTSP methodology on a practical basis.

First we reviewed the definition and the formulation of this new combinatorial optimization problem. Then we defined what a "good" tour could be "in the PTSP sense". We saw that in the Euclidean case, such a "good" tour was the product of a compromise between a tour of short length and a tour including zigzag patterns.

We assessed the efficiency of several different heuristics for the PTSP. We established that two procedures proposed in Jaillet[1985] (namely the Almost Nearest Neighbour Algorithm and the Supersavings Algorithm) behaved poorly. Then we showed that a manual-solution-procedure was very efficient and very appropriate to capture the conceptual basis of the apparent contradictory definition of the "good" PTSP tour. This manual procedure was the most efficient for coverage probabilities greater than or equal to 0.5. For coverage probabilities less than 0.5, another heuristic derived from the Clarke–Wright Savings Algorithm (it consists of maximizing the savings in expected distance) behaved well. We showed that a solution produced with this latter procedure for a given probability could also be a very good solution for
another value, and most of the time a higher value, of the coverage probability. Thus we proposed a global procedure where the problem was attacked as a whole in terms of the entire probability range. The results formerly obtained were greatly improved.

In Chapter 3, we presented a vehicle routing problem under uncertainty whose formulation was directly derived from the PTSP methodology. This problem was called the "Probabilistic Vehicle Routing Problem" (PVRP). We introduced the notion of risk and pointed out that we had to face important and somewhat surprising complications. We derived an upper bound for the expected length of a subtour. Then we proposed a solution-procedure based on the investigations of Chapter 2. Finally, we considered possible generalizations and relaxations of the formulation of the PVRP.

4.2. Directions for future work

At this stage, it is possible to identify several areas where future work may prove fruitful. The first group are concerned with the algorithmic investigation for the PTSP. The second are related to the PVRP formulation. Many questions can be raised.

- since the manual approach has proved promising, is it possible to implement a man-machine interactive procedure (as in Krolak[1971]) as an effective solution-procedure?
- is there an analytical reason that would explain the behaviour with respect to \( p \) of the solutions produced by NEWSAVE or is there a bias in the formulation of the algorithm?

- since we have seen that very often a solution produced with a low probability value could also be a better solution for a higher value of \( p \), is it possible to improve the solutions for the deterministic (\( p = 1 \)) by solving the problem for high values of the coverage probability (some recent computations allow us to suggest this point)? In other words, to which extent does the PTSP provide insights for the TSP?

- how would SAVE behave if at each step the supersavings associated with merging with the entire main tour were considered rather than only the supersavings with only the pairs (or triplets or whatever ...) of nodes located at the ends of the main tour?

- how efficient would all the other possible solution procedures mentioned in Jaillet[1985] be?

- how efficient would the implementable exact procedures proposed in Jaillet[1985] be?

- is it possible to derive closed-form expressions or at least recursive methods to calculate an exact value for the expected distance of a subtour for the FVRP?

- is it possible to find similar expressions for the case where there is no a priori constraint on the size of the subtour with respect to the capacity of the vehicle?
- what kind of other appropriate solution procedures for the generalised PVRP can be designed?

Finally, we hope that this thesis has contributed to clear some of the undergrowth from the unexplored land of the PTSP. We hope that these first results and practical reflexions will stimulate interest and provide additional motivation for future work.
REFERENCES


# APPENDIX A

## REFERENCES OF THE TEST CASES

<table>
<thead>
<tr>
<th>Test case</th>
<th>Origin</th>
<th># of white nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Refuse collection example of Larson &amp; Odoni[1981]</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Problem # 2 of Eilon[1971]</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Problem # 3</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Problem # 4</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>Problem # 5</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>Problem # 7</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>Problem # 8</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>Problem # 9</td>
<td>75</td>
</tr>
</tbody>
</table>
APPENDIX B

DETAILED RESULTS FOR PROBLEM 3

This Appendix presents the detailed results for test case 3. It shows how the initial set of solutions obtained by means of the heuristic NEWSAVE can be greatly improved to finally yield a new set of solutions that dominates the previous one. In the table next page, the row "p" contains the coverage probabilities. Rows "MANUAL" and "NEWSAVE" contain the results originally obtained respectively with MANUAL and NEWSAVE for the coverage probabilities of row "p". Then the best results obtained from the local optima produced in a first step by NEWSAVE, after recomputing all the expected distances, are presented in row "NEWSAVE improved". The coverage probabilities corresponding to these local optima are presented in row "Associated p".

Figure B1 presents the tour drawn by means of MANUAL. Figure B2 shows the tour produced by the CWSA. Figures B3, B4, B5, B6 and B7 present the tours produced by means of NEWSAVE for respectively $p=0.9$, $p=0.8$ and $p=0.7$, $p=0.6$ and $p=0.5$, $p=0.4$ and $p=0.3$, $p=0.2$ and $p=0.1$. It can be seen in the table that this last tour is the best tour for the range $0.1--0.7$; that the tour produced with $p=0.3$ (or 0.4) shown in figure B6 is the best tour for the range $0.8--0.9$ and that MANUAL still produces the best solution for $p = 1$. 

81
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>1</td>
<td>0.9</td>
<td>0.8</td>
<td>0.7</td>
<td>0.6</td>
<td>0.5</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>MANUAL</td>
<td>288.5</td>
<td>277.9</td>
<td>265.7</td>
<td>252</td>
<td>236.5</td>
<td>218.9</td>
<td>198.4</td>
<td>173</td>
<td>138.8</td>
<td>87.3</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>NEWSAVE</td>
<td>295.6</td>
<td>290.5</td>
<td>272</td>
<td>257.8</td>
<td>244.7</td>
<td>224.1</td>
<td>196.5</td>
<td>171.8</td>
<td>137.8</td>
<td>87.1</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>NEWSAVE improved</td>
<td>290.3</td>
<td>277.6</td>
<td>264.1</td>
<td>249.5</td>
<td>233</td>
<td>215.3</td>
<td>195.3</td>
<td>170.9</td>
<td>137.8</td>
<td>87.1</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Associated</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>p</td>
<td>&amp; 0.3</td>
<td>&amp; 0.3</td>
<td>&amp; 0.3</td>
<td>&amp; 0.1</td>
<td>&amp; 0.1</td>
<td>&amp; 0.1</td>
<td>&amp; 0.1</td>
<td>&amp; 0.1</td>
<td>&amp; 0.1</td>
<td>&amp; 0.1</td>
</tr>
</tbody>
</table>
Figure B1: Tour drawn for problem 3 by means of MANUAL
Figure B2: Tour obtained for problem 3 with CWSA
Figure B3 : Tour obtained for problem 3 by means of NEWSAVE

with $p = 0.9$
Figure B4: Tour obtained for problem 3 by means of NEWSAVE

with $p = 0.8$ and $p = 0.7$
Figure B5: Tour obtained for problem 3 by means of NEWSAVE

with $p = 0.5$ and $p = 0.6$
Figure B6: Tour obtained for problem 3 by means of NEWSAVE

with $p = 0.3$ and $p = 0.4$
Figure B7: Tour obtained for problem 3 by means of NEWSAVE

with $p = 0.1$ and $p = 0.2$