Moments over the Solution Space of the Travelling Salesman Problem

by

Paul John Sutcliffe

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science

Faculty of Engineering and Information Technology
University of Technology, Sydney

Certificate of Authorship and Originality

I, Paul John Sutcliffe certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Paul John Sutcliffe
The candidate and author

Dedication

To my wife, my parents and sisters

Acknowledgments

Firstly, I would like to thank my supervisors Professor Jenny Edwards and Doctor Andrew Solomon. Andrew, thanks for everything you taught me. Also thanks for not groaning too loudly at the clumsiness of some of these proofs. I'm hoping our next collaboration will be Über slick. Jenny thanks for your help in all things and particularly for preventing me from "impaling" when I hoped to "imply" and "exiting" when "existing" was intended.

Sincere thanks to the staff and students of the operations research and industrial engineering group at the University of Texas at Austin for making my brief stay there so fruitful. I am particularly grateful for the guidance and encouragement of Professor J. Wesley Barnes and Doctor Bruce Colletti.

I am indebted to Simon de Givry, Thomas Schiex and Christophe Hitte, for their kind provision of the canine RH data set and for its translation into TSP instances.

I would like to extend my thanks to David Plyler, Len Hill, John Bance and Doctor Neil Thomson.

I am grateful for the financial support in the form of a scholarship from the University of Technology Sydney.

Finally, I thank my wife Lisa for her love and extreme patience.

Abstract

In this thesis we consider the statistical properties of the symmetric travelling salesman problem (TSP). Previous work on the statistical properties of the problem has been largely limited to the Euclidean case with vertex coordinates as random variables with known distribution embedded in $\mathbb{R}^d$, and to the case of *independent identically distributed* random edge costs. Furthermore, this previous work did not extend to computing the moments, beyond the mean. In the work presented here we consider the more general case of problem instances specified as a set of edge costs and with no (known) embedding or coordinate system available.

For an instance of the problem on n vertices with fixed edge costs we give constructive proofs that the population variance of tour costs over the solution space can be computed in $\mathcal{O}(n^2)$, the third central moment can be computed in $\mathcal{O}(n^4)$ and the fourth central moment can be computed in $\mathcal{O}(n^6)$. These results provide direct methods to compute the moments about the origin and factorial moments of these orders with the corresponding computational complexity. In addition the results provide tractable methods to compute, among other statistics, the standard deviation of tour costs, the skewness of the probability distributions of tour costs over the solution space and kurtosis of this distribution.

In the case of the stochastic TSP with *edge costs* defined as independently distributed random variables with (not necessarily identical) known mean and variance we provide a $\mathcal{O}(n^4)$ algorithm to compute the variance of tour costs.

Given a subgraph S of a tour in an n city TSP, we provide an $\mathcal{O}(n^2)$ algorithm to compute the expected tour costs over the solution space of those tours containing S. This is useful in analysing and constructing algorithms such as Gutin's greedy expectation heuristic.

We demonstrate that the probability distribution of gains over the 2-opt landscape of an n city TSP can be computed in $\mathcal{O}(n^4 \log(n))$. This result provides a tractable algorithm to compute, among other statistics the moments of gains over the landscape. The result also provides the 2-opt neutrality (the number of neighbouring solutions with identical cost) of a instance. The result has natural generalisation to the 3-opt landscape (at higher computational complexity). We relate the variance of tour costs over the solution space to that of the gains over the 2-opt landscape of a problem, providing an $\mathcal{O}(n^2)$ method to compute the variance of gains over the landscape.

We apply our method to compute the low order moments of the distribution of tour costs to several empirical studies of the solution space. Among other results we: confirm the known relationship between the standard deviation of tour costs and the optimal tour cost; we show a correlation between the skewness and the optimal tour cost; we demonstrate that the moments can be used to estimate the complete probability distribution of tour costs.

Contents

Chapter 1

Introduction

The travelling salesman problem (TSP) is a classic problem in combinatorial optimisation. The problem is this: given a finite set of cities together with the distances between these cities, find the tour of shortest distance which visits each city just once and returns to the city of origin. The problem is of interest for three interrelated reasons.

Firstly, the problem and equivalent problems occur in numerous practical situations where an ordering of objects needs to be determined to optimise a cost. As a result of this, the TSP has scientific and commercial significance.

Secondly, the problem is at the heart of the most famous open theoretical problem in computer science, namely, is $\mathcal{P} = \mathcal{NP}$. That is, is the class of problems that can be solved in polynomial time with a deterministic algorithm the same as the class that can be solved in polynomial time by a non-deterministic algorithm?

Thirdly, as pointed out by Lawler et al. [78], because the TSP is comparatively well studied, it makes a good "test bed" for novel optimisation techniques. Ironically, this is made even more true by the existence of good algorithms to solve and approximate certain subclasses of the TSP. This is the case because it is seldom efficient to test and refine a novel idea on a novel problem. So, for example, key general optimisation techniques such as simulated annealing and genetic algorithms were developed initially on the TSP, but soon found application to other problems. Extensive general references to the problem include [2, 47, 78, 97].

1.1 Definition of the TSP

A pair of sets (V, E) with $E \subseteq V \times V/(x, y) \sim (y, x)$ and $x \neq y$ form a *graph* or *undirected graph*, G. An element of V is a *vertex* while $\{x, y\} \in E$ is called an *edge*. A graph $G = (V, E)$ is *complete* if, for each $x, y \in V$, there is an edge $\{x, y\} \in E$. The complete graph with n vertices is denoted K_n.

graph
undirected graph
vertex
edge
complete graph
K_n

Given these definitions and the above description of the problem it is natural to define the symmetric TSP in terms of a complete *undirected* graph $G = (V, E)$ with the vertices V representing cities, and the edges E representing the connections between cities. The number of vertices is the *size* or *order* of the TSP. We label the set of n vertices as $\{1, 2, \ldots, n\}$, and an n-cycle permutation of these is a *tour* or *solution*, π. The set of all tours, is the *solution space* Θ. The distance between cities (or the *cost* of an edge), is a function cost : $E \to \mathbb{R}$ which we extend to the function, $\text{cost}(\pi) : \Theta \to \mathbb{R}$, defined as the cost of a tour.

size
order
tour
solution
solution space Θ
cost

$$\text{cost}(\pi) = \sum\nolimits_{i=1}^{n} \text{cost}(\{\pi(i), \pi((i \mod n) + 1)\}). \tag{1.1}$$

Where we wish to emphasise that the domain of this function is Θ we write $\text{cost}_\Theta(\pi)$ and similarly for restriction to some subset of Θ. We may also abbreviate either of these cost functions to c.

$\text{cost}_\Theta(\pi)$
c

The TSP is to find some n-cycle permutation, π, of V for which $\text{cost}(\pi)$ is minimum. Such a permutation, π^* is called a *global minimum tour*. The cost, c associated with a π^* is a *global minimum cost* and is denoted c^*. It is easy to see that for an n city instance, the number of tours is $|\Theta| = (n-1)!/2$.

global minimum tour
global minimum cost
c^*

1.1.1 Example TSPs

Figure 1.1 shows a simple TSP with just 4 cities. If we endow this problem with the Euclidean metric such that each of the short edges has cost 1 then the tour on the left is a global minimum with cost 4. The two other tours have cost $2(1+\sqrt{2})$.

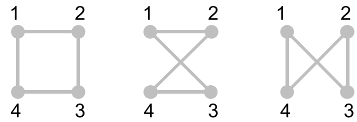


Figure 1.1: *A four city TSP. The only tours are* $(1,2,3,4)$, $(1,2,4,3)$, *and* $(1,3,2,4)$.

For each circular permutation, π, of the integers $1, 2, \ldots, n$, there are $2n$ distinct *linear permutations* of these integers corresponding to π. So in the figure the tours $(1,2,3,4)$ and $(2,3,4,1)$ are identical being simply shifted by 1 city. Similarly the tours $(1,2,3,4)$ and $(1,4,3,2)$ are identical, with this rearrangement being a traversal of the tour in the opposite direction. Therefore given a tour, π, it is convenient to fix a linearization of π such that the first vertex is 1 and the second vertex is smaller than the last. That is we write π as $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ with $\pi_1 = 1$ and $\pi_2 < \pi_n$. It is easy to see that there is a bijection between the circular permutations and this linearization. Using this convention, the three tours of Figure 1.1 are $(1,2,3,4)$, $(1,2,4,3)$ and $(1,3,2,4)$. Given a tour π in a TSP with graph (V,E), by $E_\pi \subseteq E$ we mean the set of edges of E in π. So in the case of the tour $\pi = (1,2,3,4)$ in Figure 1.1, $E_\pi = \{\{1,2\},\{2,3\},\{3,4\},\{1,4\}\}$.

linear permutations

$E_\pi \subseteq E$

Figure 1.2 shows a low cost tour in a small but typical TSP. The instance, pr76.tsp, has 76 cities and originated from the manufacture of printed circuit boards [96]. This example has approximately 1.2×10^{109} tours. It is worth noting that it is estimated that there are only 10^{80} (non-virtual) elementary particles in the universe [108]. This corresponds to the number of tours in a TSP with around 60 cities. This illustrates the futility of naive approaches to computing the moments of tour costs over the solution space of the problem.

Figure 1.2: *A low cost tour in a 76 city TSP (pr76.tsp). The problem is a typical, but small, example from the manufacture of printed circuit boards. The instance has approximately* 1.2×10^{109} *tours. This number dwarfs the number of elementary particles in the universe by a large factor.*

1.2 Motivation for this Research

This thesis is concerned with understanding the statistical properties of the solution space of the TSP and other constructs, such as combinatorial landscapes of the problem. We take the view that, a *clear* understanding of properties of the problem will lead to the development of heuristics and optimisation algorithms that exploit these insights. To this end, our primary goal is to demonstrate that the TSP is amenable to statistical analysis.

1.3 Outline and Author's Contribution

This thesis is arranged in two parts. Chapters 1 to 4 introduce and review the literature on the TSP and matters relating to the author's contribution. Chapters 5 to Chapters 11 provide the author's contribution.

More specifically, the remainder of the thesis is set out as follows. In Chapter 2 we discuss the relationships between the TSP and other problems in terms of computability and approximation. We also describe certain subtypes of the problem and its relationship with the minimum spanning tree problem. In Chapter 3 we briefly survey some of the most well known heuristics and algorithms to address the TSP. Chapter 4 surveys the statistical literature on the problem.

Chapter 5 begins the author's contribution. In it we introduce a class of TSPs we term *perfect* for which each solution has a distinct cost, that is, the cost function of an instance is an injective function to the real numbers. In Chapter 6 we prove that the expected value of tours over certain subsets of the solution space can be computed in polynomial time on the number of cities of an instance. In Chapter 7 we consider the population variance of tour costs over the solution space of both fixed cost and stochastic TSPs. Constructive proofs are given showing that in both these cases, the variance can be computed in polynomial time. We also, in this chapter, confirm the known empirical evidence of a relationship between the variance of an instance and the cost of its optimal solution. In Chapter 8 we prove that the third central moment of the distribution of costs over the solution space can be computed in polynomial time. This provides the well known statistic, the skewness of the distribution. We provide empirical evidence of a correlation between the skewness of the distribution of costs and the optimal tour cost. These empirical results suggest that it may be possible to construct models to estimate the likely optimal tour cost based on: the problem type of an instance, its size, mean, standard deviation and skewness. In Chapter 9 we extend our results to compute the fourth central moment of tour costs. In the penultimate chapter, Chapter 10, we show a relationship between the variance of tour costs over the solution space of an instance and the statistical properties of the landscape arising from the well known 2-opt move. Finally in Chapter 11 we consider future research arising from this thesis. In particular we consider the problem of estimating the probability distribution of tour costs of an instance using the moments.

Chapter 2

The Relationships Between the TSP and other Problems

As we pointed out in the introduction, the TSP is associated with the class of problems termed $\mathcal{NP}$-complete. We first, in the following two sections, briefly elaborate on these comments. Complete discussion can be found in Garey and Johnson [39], Johnson [59], Cormen et al. [27] and Ausiello et al. [5]. In Section 2.3 we consider certain sub-types and variations of the TSP. Section 2.4 provides some additional graph theoretical definitions. Finally, in Section 2.5 we discuss the relationship between the TSP and other problems.

2.1 Computability of the TSP

By the *space complexity* of a problem we mean the number of bits required to solve an instance of a problem under a reasonable encoding scheme. A problem is in the class *polynomial time computable* or $\mathcal{P}$ if there exists a polynomial time algorithm to solve it. It is in *nondeterministic polynomial time* or $\mathcal{NP}$ if there is some nondeterministic algorithm that computes it in polynomial time. The *decision problem* associated with an optimisation problem is a variation of that problem which frames it as a boolean valued function. In the case of the TSP, the decision problem is: given a TSP, is

space complexity
polynomial time
$\mathcal{P}$
nondeterministic polynomial
$\mathcal{NP}$
decision problem

there a tour with cost less than or equal to k? An instance of this decision problem is an instance of a TSP together with the number k. Clearly the decision problem of the TSP is in $\mathcal{NP}$, since, given a TSP and bound, k, we can nondeterministically "guess" at an appropriate tour π and then in polynomial time verify that $\text{cost}(\pi) \leq k$.

Let α be a decision problem and let D_α be the domain of the problem, that is the set of all problem instances of α. Let $Y_\alpha \subseteq D_\alpha$ be the set of all positive instances ("yes" instances) of α. Similarly let β be a decision problem with domain D_β and positive instances $Y_\beta \subseteq D_\beta$. The problem α is *polynomial time transformable* to β if there is a polynomial time computable function $f : D_\alpha \rightarrow D_\beta$ such that $\beta(f(a)) \in Y_\beta$, if and only if $a \in Y_\alpha$. If this is the case, we write $\alpha \leq_{\mathcal{P}} \beta$. This type of transformation is termed a *Karp reduction*. The existence of a Karp reduction means we can decide if a is an element of Y_α in the sum of the time it takes to compute $b = f(a)$ and the time it takes to decide if b is in Y_β. A problem β is $\mathcal{NP}$*-complete* if

polynomial time transformable

$\leq_{\mathcal{P}}$

Karp reduction

$\mathcal{NP}$*-complete*

$$
\begin{array}{ll}
\bullet & \beta \in \mathcal{NP} \\
\bullet & \alpha \leq_{\mathcal{P}} \beta \text{ for all } \alpha \in \mathcal{NP}.
\end{array} \tag{2.1}
$$

Theorem 1 *The decision problem of the TSP is* $\mathcal{NP}$*-complete.*

Proof: A full proof is given in Johnson [59] It relies on showing firstly, that the decision problem is in $\mathcal{NP}$, (this is shown above) and secondly, that there is a polynomial time Karp reduction from a known $\mathcal{NP}$-complete problem, the Hamiltonian cycle problem. By the definition of $\mathcal{NP}$-completeness, this is sufficient to provide such a reduction from any problem in $\mathcal{NP}$. $\square$

A *polynomial time Turing reduction* $\alpha \leq_{\mathcal{P}T} \beta$ from a problem α to a problem β is one that acts like a program to solve α by calling a polynomial time subroutine to solve β. In contrast to a Karp reduction, the algorithm to solve β may be called multiple times.

polynomial time Turing reduction

$\leq_{\mathcal{P}T}$

Informally, the problem is $\mathcal{NP}$*-hard* if an algorithm to solve it would provide an algorithm to solve any $\mathcal{NP}$-complete problem. More precisely, an optimisation problem, o, is $\mathcal{NP}$-hard if there is an $\mathcal{NP}$-complete decision problem δ and a polynomial time Turing reduction from δ to o, that is

$\mathcal{NP}$*-hard*

$\delta \leq_{\mathcal{P}T} o$. The most important consequence of this is that a polynomial time algorithm to solve an $\mathcal{NP}$-hard problem would imply that $\mathcal{P} = \mathcal{NP}$.

Theorem 2 *The TSP is $\mathcal{NP}$-hard.*

Proof: This is obvious since if we know a global minimum tour for an instance, we can compute its cost and compare this to a give bound, in polynomial time. □

An optimisation problem, o, is *$\mathcal{NP}$-easy* if there is an $\mathcal{NP}$-complete decision problem δ and a polynomial time Turing reduction from o to δ, that is $o \leq_{\mathcal{P}T} \delta$. Finally, a problem is *$\mathcal{NP}$-equivalent*, if it is both $\mathcal{NP}$-hard and $\mathcal{NP}$-easy. As we will now see the TSP is, in addition to being $\mathcal{NP}$-hard, $\mathcal{NP}$-easy. At first sight this is rather surprising. It certainly justifies attention to the properties, statistical and otherwise, of the problem's solution space.

$\mathcal{NP}$-easy

$\mathcal{NP}$-equivalent

Algorithms 1 and 2 from Reinelt [97], demonstrate, for *fixed* precision arithmetic[1], that given a polynomial time TSP decision function, TSP Decision(G,b), there is a polynomial time function to solve the TSP optimisation problem.

Theorem 3 *The TSP is $\mathcal{NP}$-easy.*

Proof: By Algorithms 1 and 2. □

Its easy to see that $\mathcal{P} = \mathcal{NP}$, if any $\mathcal{NP}$-complete problem is also in $\mathcal{P}$. Likewise if an $\mathcal{NP}$-complete problem is proven not to be in $\mathcal{P}$ then $\mathcal{P} \neq \mathcal{NP}$. The key fact is, that no one has been able to prove either of these statements, so the question of the equality $\mathcal{P}$ and $\mathcal{NP}$ remains a famous and intriguing open problem. Although it is fair to say that it is most unlikely that $\mathcal{P} = \mathcal{NP}$, a proof of this may never be found.

[1] Each edge cost can be scales to an integer value.

Algorithm 1 Compute Global Optimal Cost
Pseudo code to find the global minimum cost of a TSP with graph G. We assume the existence of TSP Decision(G, b) which returns true, if and only if $c* \leq b$ where $c*$ is the global minimum cost of a TSP with graph G.

Input: G a graph of a TSP with n vertices.
Input: c_l the largest edge cost.
Output: c^* the global minimum tour cost.

$c_{upper} \leftarrow nc_l$
$c_{lower} \leftarrow -nc_l$
while $c_{lower} < c_{upper}$ **do**
  $k \leftarrow \lceil \frac{c_{lower}+c_{upper}}{2} \rceil$
  if TSP Decision(G, k) **then**
    $c_{upper} \leftarrow k$
  else
    $c_{lower} \leftarrow k + 1$
  end if
end while
$c^* \leftarrow k$
return c^*

Algorithm 2 Compute Global Optimal Tour
Pseudo code to find a tour with global minimum cost

Input: A graph G of a TSP with n vertices.
Input: The global minimum tour cost of the TSP c^*.
Output: A global minimum tour π^*.

Let R be an empty graph with edge cost function R.cost
$R \leftarrow G$
for all $i, j \in 1 \ldots n$ **do**
  R.cost$(i, j) \leftarrow c^* + 1$
  if $false =$ **TSP Decision**(R, c^*) **then**
    R.cost$(i, j) \leftarrow G$.cost(i, j)
  end if
end for
Delete from R any edges of cost $c^* + 1$.
return R as a tour π^*

2.2 Approximation of the TSP

Since the likelihood of discovering a polynomial time procedure for any $\mathcal{NP}$-complete problem is low, much interest has focused on finding methods that give near optimal solutions. Such an algorithm is termed an *approximation algorithm*. Let c^* be the global minimum cost of a minimization problem and c be the value provided by some approximation algorithm α. We say α is *ratio bound* by $\rho(n)$ if $\frac{c}{c^*} \leq \rho(n)$ for any input size n. So $\rho(n)$ is the factor

approximation algorithm
ratio bound

by which c may be greater than the global minimum. If $\rho(n)$ is independent of n we write the ratio bound as ρ. Similarly α has a *relative error bound* $\epsilon(n)$ if $\frac{|c-c^*|}{c^*} \leq \epsilon(n)$ for any input size n. If $\epsilon(n)$ is independent of n we write the relative error bound as ϵ.

relative error bound

We are frequently interested in approximation algorithms which in addition to a problem instance of size n, also take as input, a relative error bound $\epsilon > 0$, and produce a solution with error bound ϵ. Such an algorithm is called an *approximation scheme*. It is a *polynomial-time approximation scheme* if it runs in polynomial time in n, for fixed ϵ. It is a *full polynomial-time approximation scheme* if, there is a two variable polynomial p, such that the run time of the algorithm is bounded by $p(n, \frac{1}{\epsilon})$.

approximation scheme
polynomial-time
full polynomial-time

Sahni and Gonzalez [103] show that there is unlikely to be any approximation scheme for the TSP which is guaranteed to work "well" for all problem instances.

Theorem 4 *Unless $\mathcal{P} = \mathcal{NP}$ no polynomial-time approximation scheme exists for the TSP with $\rho \geq 1$.*

Their proof relies on showing that if such an approximation scheme existed it could be used to solve the Hamiltonian cycle problem in polynomial time. However, as noted above, the Hamiltonian cycle problem is known to be $\mathcal{NP}$-complete.

2.3 Types and Variations of the TSP

2.3.1 Euclidean and Geometric TSPs

A TSP is termed Euclidean where its edge costs can be embedded in $\mathbb{R}^d$ under the conventional Euclidean metric. If this is the case, then there is a natural coordinate system available and an n city problem can be expressed compactly as n Cartesian coordinates. For $d = 2$, this situation occurs frequently and directly from the specification of many practical problems.

In the case of the statistics literature, it is typical to consider problems with real valued coordinates in $[0, 1]^d$, [114, 124]. The cost of an edge between two vertices p and q with coordinates $(p_1, p_2 \ldots, p_d)$ and $(q_1, q_2 \ldots, q_d)$ is

$$\mathrm{cost}(\{p, q\}) = \left(\sum_{i=1}^{d} (p_i - q_i)^2 \right)^{\frac{1}{2}}. \tag{2.2}$$

A second common variation has cities positioned in a bounded subset of $\mathbb{Z}^d$ with a discretised Euclidean metric given by

$$\mathrm{cost}(\{p, q\}) = \left\lceil \left(\sum_{i=1}^{d} (p_i - q_i)^2 \right)^{\frac{1}{2}} \right\rceil. \tag{2.3}$$

It is tempting to view these arrangements as simply, a specialisation, of the TSP. However, as Arora [4] relates, there is a technical difficulty here, since the decision problem associated with the Euclidean TSP is $\mathcal{NP}$-hard, but it is not *known* to be in $\mathcal{NP}$. This occurs because there is no known polynomial time algorithm that decides, given a set C of integers and an integer k, if it is true that: $\sum_{c_i \in C} \sqrt{c_i} \leq k$.

A common generalisation of the Euclidean TSP is a *geometric TSP* where the edge costs have embedding in a specified metric space, for example on a sphere. The exact metrics used for this case and several others are provided in Reinelt [96] along with an extensive library, TSPLIB [57], of mostly two dimensional Euclidean cases.

geometric TSP

In spite of the complication introduced by the taking of roots, the Euclidean and geometric problems are amenable to approximation. They are succinctly represented and invite heuristics based on geometric consideration. Arora provides a probabilistic approximation scheme for discrete geometric cases [3, 72]. For any fixed $\epsilon > 0$, Arora's algorithm provides, with probability at least $\frac{1}{2}$, a ratio bound of $(1 + \epsilon)$ in run time $\mathcal{O}\left(n(\log(n))^{\frac{k}{\epsilon}}\right)$ for some constant k. These results do not extend to the case of the full polynomial-time approximation scheme.

Theorem 5 *Unless $\mathcal{P} = \mathcal{NP}$, no full polynomial-time approximation scheme exists for the Euclidean TSP.*

A proof is provided by Johnson and Papadimitriou [63].

2.3.2 TSP with Triangular Inequality

The *TSP with triangular inequality* is a widely occurring problem. As the name suggests, for any three cities p, q, r we have

TSP with triangular inequality

$$\text{cost}(\{p, q\}) \leq \text{cost}(\{p, r\}) + \text{cost}(\{r, q\}). \tag{2.4}$$

Frequently we do not have a convenient coordinate system available and, as with the general TSP, the edge costs are provided explicitly. Where this last property is true, the TSP with triangular inequality is a clearly specialization of the TSP. In Sections 3.1.2 and 3.1.3 we discuss two key approximation algorithms. The first of these, the doubling algorithm, is ratio bound by a factor of 2 and, the second, the Christofides algorithm by $\frac{3}{2}$.

The following simple but useful result is known as *the shortcut lemma*. It extends some of the properties of the triangular inequality to larger paths and is instrumental in proving the efficacy of the Christofides algorithm. Figure 2.1 illustrates the situation.

shortcut lemma

Lemma 6 (Shortcut) *Let $G = (V, E)$ be a graph of a TSP with edge costs obeying the triangular inequality. Let $e = \{v_1, v_m\}$ be an edge in G and let $P = (v_1, v_2, \ldots v_m)$ be a path in G. Then* $\text{cost}(e) \leq \text{cost}(P)$.

Proof: By $m - 2$ applications of the triangular inequality starting with $\text{cost}(\{v_1, v_3\}) \leq \text{cost}(\{v_1, v_2\}) + \text{cost}(\{v_2, v_3\})$. $\square$

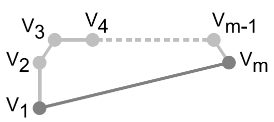


Figure 2.1: *An illustration of the shortcut lemma.*

Parameterised Triangular Inequality

Bender and Chekuri [11] describe a generalisation of the triangular inequality, the *parameterised triangular inequality*. Here, any three edges of the problem are constrained by

parameterised triangular inequality

$$\text{cost}(\{p,q\}) \leq \alpha(\text{cost}(\{p,r\}) + \text{cost}(\{r,q\})), \tag{2.5}$$

where p, q, r are cities and the parameter α reflects the degree to which the triangular inequality is obeyed. Among other results these authors provide a 4α ratio bound algorithm for TSPs obeying Equation 2.5.

2.4 More Graph Theoretic Definitions

Given undirected graphs $G = (V, E)$ and $G' = (V', E')$ if $V' \subseteq V$ and $E' \subseteq E$ then G' is a *subgraph* of G written $G' \subseteq G$. If $G' \subseteq G$ and $V' = V$ then G' is a *spanning subgraph* of G. A *walk* W in a graph $G = (V, E)$ is a sequence of not necessarily distinct vertices $(v_1, v_2 \dots v_k)$ with each $\{v_i, v_{i+1}\} \in E$ for all $i \in [1 \dots k-1]$. A *path* P in G is a walk with distinct vertices. A *cycle* C in $G = (V, E)$ is a sequence of vertices $(v_1, v_2 \dots v_k, v_1)$ with $(v_1, v_2 \dots v_k)$ forming a path and with $\{v_k, v_1\} \in E$.

subgraph
spanning subgraph
walk
path
cycle

Clearly a walk, path or cycle in a graph G is a subgraph of G. If, as with the TSP, there is a cost function from the edges of a graph to the real numbers, it is reasonable to extend this function to the graph. So, by the cost of a graph $G = (V, E)$ with respect to a function cost : $E \to \mathbb{R}$, we mean the function cost : $G \to \mathbb{R}$ defined as $\sum_{e \in E} \text{cost}(e)$.

A directed graph is a pair of disjoint sets V and E along with maps *init* : $E \to V$ and *ter* : $E \to V$. These maps assign each edge $e \in E$ an *initial vertex*, *init*(e), and a *terminating vertex*, *ter*(e). An edge e with *init*(e) = *ter*(e) is termed a *loop*. In addition, under this arrangement there may be multiple edges between vertices. The variation of an undirected graph that allows multiple edges and loops is termed a *multi-graph*. The definitions of path, walk and cycle have natural extension to directed graphs and multi-graphs [15, 32].

initial vertex
terminating vertex
loop
multi-graph

2.5 Other Problems

2.5.1 Minimum Spanning Trees

A *spanning tree*, T of a graph $G = (V, E)$ is a non-cyclic subgraph that connects all of V. A *minimum spanning tree*(MST) is a spanning tree with the sum of edge costs a minimum. In contrast to the TSP, there are several effective algorithms to compute a MST. The algorithms of Kruskal and Prim both have a nominal complexity in $\mathcal{O}(|E| \log |V|)$ [27]. However, in the case of Prim's algorithm a complexity in $\mathcal{O}(|E| + |V| \log |V|)$ is possible by use of a Fibonacci heap [69].

spanning tree
minimum spanning tree
MST

The next lemma demonstrates the close connection between the MST and the TSP with the triangular inequality.

Lemma 7 *Let $G = (V, E)$ be a graph of a TSP with non-negative edge costs and with a global minimum tour π^* and let T be a Minimum Spanning Tree over G. Then* $\text{cost}(T) \leq \text{cost}(\pi^*)$.

Proof: We can form a spanning tree of G by removing a single edge from π^* to form a path, P. Clearly $\text{cost}(T) \leq \text{cost}(P) \leq \text{cost}(\pi^*)$. □

Given this result, it is unsurprising that the MST and its variation find application in the computation of lower bounds for the cost of a minimum tour. The most important approach is the one tree described by Held and Karp [2, 60]. Furthermore, the connection between the MST and the TSP is exploited in several approximation procedures, most notably the doubling and Christofides algorithms discussed in Sections 3.1.2 and 3.1.3.

2.5.2 Hamiltonian Cycle Problem

A *Hamiltonian cycle* in an undirected graph is a cycle that visits each vertex of the graph. The *Hamiltonian cycle problem* is: given a graph, G, does G have a Hamiltonian cycle? Clearly this is very similar to the TSP. Indeed, the TSP is frequently called the *weighted Hamiltonian cycle problem*. Each

Hamiltonian cycle
Hamiltonian cycle problem
weighted Hamiltonian cycle problem

tour in a TSP is simply a Hamiltonian cycle in a complete graph and any Hamiltonian cycle is simply a cyclic permutation of vertices.

The Hamiltonian cycle problem is known to be $\mathcal{NP}$-complete and the proof of this is *not* dependent on the TSP being $\mathcal{NP}$-complete [27]. In Chapter 5 we provide a reduction from the Hamiltonian cycle problem to the TSP which, among other properties, has an injective cost function to the real numbers.

2.5.3 Asymmetric TSP

The *asymmetric travelling salesman problem* (ATSP) [2, 47, 78, 97] is, as the name suggests, a variation of the TSP in which the cost in moving from vertex x to vertex y need not be equal to that of moving from y to x. The problem is typically modelled on a directed graph with a tour being a permutation of the vertices of that graph.

asymmetric travelling salesman problem

2.5.4 Summary

In this chapter we have provided a brief review of the computational properties of the TSP and its relationships to other combinatorial optimisation problems. In the next chapter we provide a review of the most common methods employed to address the problem.

Chapter 3

Survey of TSP Algorithms

In this chapter we provide a survey of the most common TSP algorithms. We restrict our attention largely to methods that apply to the general TSP and the TSP with the triangular inequality.

There are essentially two approaches to finding solutions to the TSP. Firstly, a tour can be constructed. Secondly a process of improvement can be applied to a tour already constructed. We begin with the tour construction methods. Detailed comparisons of various approaches are given by Reinelt [97], Johnson and McGeoch [61] and Chong [20].

3.1 Tour Construction Methods

3.1.1 Branch and Bound

Branch and bound was discovered independently by at least three groups. Firstly Dantzig et al. [29] applied the method to the ATSP. This extremely significant paper also introduced several other innovations. A more general description was provided by Land and Doig [75] in the context of solving integer programming problems by linear programming. Finally, the approach was described and named branch and bound by Little et al. [81] in an application to the TSP. They noted a number of unpublished papers with similar approaches. The hallmark of the technique is its ability find the global optimal solution, although, of course, in super-polynomial time.

Algorithm 3 provides a branch and bound algorithm for the TSP. Here we construct a tree T, the vertices of which describe a sets of tours D. We do not want to actually store any D, since they can be as large as the solution space S. For each vertex v of T, we simply maintain two edge sets (of G) $v.M \subset E$ and $v.N \subset E$ with $v.M$ a set of edges that must be in a tour in $v.D$ and $v.N$ a set of edges that must be never be in a tour of $v.D$. If we set $v.M = \emptyset$ and $v.N = \emptyset$ then $v.D = S$, which is to say, no edges are compulsory and no edges are excluded. This situation appears at the root of the tree.

Algorithm 3 Branch and Bound

Pseudo code for the branch and bound heuristic.

Input: A graph $G = (V, E)$ of a TSP.
Input: A Tree T with the vertices consisting of two edge sets (of G) $M \subset E$ and $N \subset E$ with M a set of edges that must be in a tour and N a set of edges that must never be in a tour. For a vertex v denote these two sets as $v.M$ and $v.N$.
Output: A global optimal tour π^*.

mark the root vertex r of T active
$r.M \leftarrow \emptyset$
$r.N \leftarrow \emptyset$
compute an upper bound C_{upper} on the cost of a optimal tour
$C_{best} \leftarrow C_{upper}$
while T has an active vertex v **do**
  choose an unused edge $e \in E$ with e consistent with $v.M$
  mark e as used
  create in T non-active child vertices v_{left}, v_{right} of v
  $v_{left}.M \leftarrow v.M \bigcup \{e\}$
  $v_{left}.N \leftarrow v.N$
  $v_{right}.M \leftarrow v.M$
  $v_{right}.N \leftarrow v.N \bigcup \{e\}$
  for all v_c the two child vertices of v **do**
    compute C_{lc} a lower bound on the cost of any solution consistent with v_c
    if $C_{lc} < C_{best}$ **then**
      if $v_c.M$ describes one tour, π, and $C_{best} > cost(\pi)$ **then**
        $\pi* \leftarrow \pi$
        $C_{best} \leftarrow cost(\pi)$
      else
        mark v_c active
      end if
    end if
  end for
end while
return π^*

The upper bound of line 4 is computed as the product of $|V|$ and the largest cost of an edge in E. The quality of the lower bound at line 15 determines the efficiency of the method. If a poor lower bound is used, say, by choosing a number less than the cost of any solution in S, then every solution in S will require examination. A more useful lower bound is given by Algorithm 4. By *consistent* on line 7, we mean the set $v.M \bigcup \{e\}$ forms a tour or could form a tour with the addition of extra edges.

Algorithm 4 Lower Bound
A Lower bound for the cost of tours consistent with M and N

Input: A graph $G = (V, E)$ of a TSP.
Input: $M \subset E$, a set of edges that must be in a tour.
Input: $N \subset E$, a set of edges that must never be in a tour.
Output: A lower bound c_{lower} on the tour costs, consistent with M.
1: let S be the $|V| - |M|$ edges of $E - (M \bigcup N)$ of least cost
2: $c_{lower} \leftarrow \sum_{e \in M} \text{cost}(e) + \sum_{e \in S} \text{cost}(e)$
3: **return** c_{lower}

3.1.2 The Doubling Algorithm

An *Eulerian circuit*, O in a graph $G = (V, E)$ is a walk that visits each vertex of V at least once, and each edge in E once and only once, and returns to its start vertex. A graph is Eulerian if it has an Eulerian circuit. Euler observed that Eulerian graphs are precisely those connected graphs with the degree of every vertex divisible by 2 [32]. It is easy to see that a tour π can be constructed from an Eulerian circuit, O, with the vertices of π ordered by the order of their first appearance in O. *Eulerian circuit*

The doubling algorithm, Algorithm 5, exploits the properties of the minimum spanning tree, detailed in Section 2.5.1 to construct a tour. A proof of its performance is given in Theorem 8.

Theorem 8 *The Doubling Heuristic, Algorithm 5, is ratio bounded by a factor of 2 for a TSP obeying the triangular inequality.*

Proof: Let π^* be the optimal tour. We have by Lemma 7, $\text{cost}(T) \leq \text{cost}(\pi^*) \leq \text{cost}(\pi)$. The triangular inequality gives $\text{cost}(\pi) \leq \text{cost}(O) \leq$

Algorithm 5 Doubling Algorithm
Pseudo code for the doubling algorithm.

Input: A graph $G = (V, E)$ of a TSP.
Output: A tour π.
1: compute a Minimum Spanning Tree T of G
2: construct a Multi-graph D from T by *doubling* each edge in T to form a pair of parallel edges in D
3: compute an Eulerian circuit O of D
4: convert O to a tour π with the vertices of π ordered by their first appearance in O
5: **return** π

$\text{cost}(D) = 2\text{cost}(T)$ as required. □

The complexity of the algorithm is in $\mathcal{O}(n^2)$, this being the cost of computing a minimum spanning tree.

3.1.3 Christofides' Algorithm

This algorithm has the distinction of providing the best known performance guarantee for instances of the TSP where the triangular inequality holds. Under this condition, the algorithm constructs tours with a ratio bound of $\frac{3}{2}$ in the worst case. Algorithm 6 gives the method and we provide a proof in Theorem 9 (This is essentially that given in Johnson and Papadimitriou [64]). Both the algorithm and its proof rely on the properties of a matching and a perfect matching. We define these here.

A *matching* on a graph $G = (V, E)$ is a subset of its edges $M \subseteq E$ such that no two edges of M are adjacent. Each vertex is therefore matched, by an edge, with just one other vertex. A *perfect matching* is a matching $M \subseteq E$ such that $|M|$ is the largest size possible. A *minimum cost perfect matching* is, as the name suggests, a perfect matching such that the sum of its edge cost is a minimum, over the set of possible perfect matchings.

matching

perfect matching

minimum cost perfect matching

The algorithm exploits the relationship between a minimum cost spanning tree and a TSP in much the same way as the doubling heuristic. The central insight of Christofides was to find a tighter method, than the doubling heuristic, to construct an Eulerian graph, by use of a perfect match.

Algorithm 6 **Christofides' Algorithm**
Pseudo code for the Christofides' algorithm.

Input: A graph $G = (V, E)$ of a TSP. A start vertex $s \in V$.
Output: A tour π.
1: compute a minimum spanning tree T of G
2: $V' \leftarrow$ the odd degree vertices of T
3: construct a minimum cost perfect matching, M on induced subgraph of G subject to V'. So M has the odd degree vertices of T, paired via edge costs in G
4: construct a multi-graph D from M and T such that the common edges of M and T form pairs of parallel edges in D
5: compute an Eulerian circuit O of D
6: convert O to a tour π with the vertices of π ordered by their first appearance in O
7: **return** π

Theorem 9 *Christofides' Algorithm is ratio bounded by a factor of $\frac{3}{2}$ for a TSP obeying the triangular inequality.*

Proof: By the elementary properties of graphs, every graph has an even number of odd degree vertices. So the set V' of Algorithm 6 has an even number of elements. In addition, each of the vertices in this set appears at some position in any tour. Let $(v'_1, v'_2 \dots v'_{|V'|})$ be the order in which these vertices occur in the optimal tour π^*. Construct a cycle $C = (v'_1, v'_2 \dots v'_{|V'|}, v'_1)$ in G. By the Shortcut Lemma, Lemma 6, the sum of the edge costs in $\text{cost}(C) \leq \text{cost}(\pi^*)$ since each edge in C is either a edge in π^* or circumvents a number of edges in π^*.

The alternating edges of C, $M_1 = \{\{v'_1, v'_2\}, \{v'_3, v'_4\}, \dots \{v'_{|V'-1|}, v'_{|V'|}\}\}$ and $M_2 = \{\{v'_2, v'_3\}, \{v'_4, v'_5\}, \dots \{v'_{|V'|}, v'_1\}\}$ both form perfect matchings since both have the same number of vertices as M, paired as edges. Now $\text{cost}(C) = \text{cost}(M_1) + \text{cost}(M_2)$ So $\text{cost}(M_1) + \text{cost}(M_2) \leq \text{cost}(\pi^*)$. However both M_1 and M_2 are perfect matchings and so $\text{cost}(M_1) \geq \text{cost}(M)$ and $\text{cost}(M_2) \geq \text{cost}(M)$, giving $\text{cost}(M) \leq \frac{1}{2}\text{cost}(\pi^*)$.

Any tour π produced by Algorithm 6 takes its edges from either T or M, giving $\text{cost}(\pi) \leq \text{cost}(T) + \text{cost}(M)$. T is a minimum cost spanning tree so $\text{cost}(T) \leq \text{cost}(\pi^*)$ and by the condition above $\text{cost}(M) \leq \frac{1}{2}\text{cost}(\pi^*)$ Giving $\text{cost}(\pi) \leq \frac{3}{2}\text{cost}(\pi^*)$ which implies the result. $\square$

The worst case runtime of the algorithm is in $\mathcal{O}(n^3)$, this being the cost of computing the minimum perfect matching [64]. However Johnson and McGeoch [61] note that application of the algorithm of Gabow and Tarjan [38] has the potential to reduce this to $\mathcal{O}(n^{2.5})$.

3.1.4 Nearest Neighbour

The nearest neighbour algorithm (Algorithm 7) is the simplest of the tour construction methods. Clearly the run time of the algorithm is $\mathcal{O}(n^2)$ for a TSP of size n cities. In terms of performance, Rosenkrantz et al. show that in instances obeying the triangular inequality, the heuristic has a ratio bound in $\theta(\log(n))$ [64].

Algorithm 7 Nearest Neighbour Algorithm
Pseudo code for the nearest neighbour heuristic.

Input: A graph $G = (V, E)$ of a TSP.
Input: A start vertex $s \in V$.
Output: A tour π.

```
P ← ∅
l ← s
U ← V − {s}
while |U| > 0 do
    choose a vertex u ∈ U with min(cost({l, u}))
    P ← P ⋃ {{l, u}}
    U ← U − {u}
    l ← u
end while
return P as a tour π
```

3.1.5 Clarke and Wright

The Clarke Wright Algorithm [24] (Algorithm 8) was developed to solve a generalisation of the TSP, the vehicle routing problem. It is helpful to view the algorithm as operating on a multi-graph $G = (V, E)$.

Given a TSP on n cities and a start vertex s, the algorithm works by first conducting $n - 1$ two edge cycles each starting at, and returning to, s. The algorithm then merges cycles iteratively until only a single tour is left. In terms of this merge operation, let P and Q be two edge disjoint cycles in G, with a single common vertex s. The two cycles may be written as $P = (s, p_1, p_2, \ldots, p_l, s)$ and $Q = (s, q_1, q_2, \ldots, q_m, s)$.

The four possible Clarke and Wright mergers are then:

$$\begin{aligned}
&(s, p_1, p_2, \ldots, p_l, q_1, q_2, \ldots, q_m, s)\\
&(s, p_1, p_2, \ldots, p_l, q_m, \ldots, q_2, q_1, s)\\
&(s, p_l, \ldots, p_2, p_1, q_1, q_2, \ldots, q_m, s)\\
&(s, p_l, \ldots, p_2, p_1, q_m, \ldots, q_2, q_1, s).
\end{aligned} \tag{3.1}$$

Algorithm 8 Clarke and Wright Algorithm
Pseudo code for the Clarke Wright algorithm for the TSP.

Input: A multi graph $G = (V, E)$ of a TSP with $n = |V|$ vertices.
Input: A start vertex $s \in V$.
Output: A tour π.

```
construct n − 1 two edge cycles in G each connecting s with each remaining vertex of V
while G has more than 1 cycle do
    for all pairs of cycles P, Q ∈ G do
        compute a Clarke and Wright merge cycle of minimal costs (Equation 3.1)
    end for
    merge the two best cycles P, Q found
end while
return the remaining cycle as a tour π
```

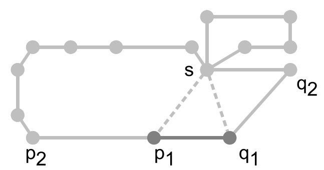


Figure 3.1: *The Clark and Wright heuristic. Here the two cycles can be merged by removing the edges $\{s, p_1\}$, $\{s, q_1\}$ and replacing them with $\{p_1, q_1\}$.*

Figure 3.1 illustrates a merge. It is clear from this graph that the method exploits the triangular inequality. The run time of the algorithm for a TSP with n cities is in $\mathcal{O}(n^2 \log(n))$ with a space complexity in $\mathcal{O}(n^2)$ [61].

3.1.6 Insertion Heuristics

The insertion heuristic works by constructing a tour from some path, P, by a process of inserting vertices into P. Algorithm 9 provides the general arrangement.

Algorithm 9 General Insertion Method
Pseudo code for insertion heuristic.

Input: A graph $G = (V, E)$ of a TSP with $n = |V|$ vertices.
Input: A path $P = v_1, v_2 \dots v_k$ of length $k < n$ in G.
Output: A tour π.
1: **while** $|P| \neq n$ **do**
2:     **choose** some vertex q with $q \notin P$
3:     **choose** an insertion point, i, to place q in P
4:     $P \leftarrow (v_1, v_2 \dots, q, v_i \dots)$
5: **end while**
6: return P as a tour π

At line 3 of this procedure, if $i > 1$, the act of inserting q reduces to removing the edge $\{v_{i-1}, v_i\}$ from the partial tour and replacing it with the two edges $\{v_{i-1}, q\}$ and $\{q, v_i\}$. Reinelt [97] lists nine insertion heuristics. Most have run time $\mathcal{O}(n^2)$ with two variants having run time $\mathcal{O}(n^2 \log(n))$. The initial path P typically consists of between 0 and 2 vertices.

3.1.7 The Greedy Heuristic

The greedy heuristic constructs a tour iteratively, by inserting an edge of lowest cost into a set T, *consistent* with the requirement to eventually result in a tour. The arrangement is given in Algorithm 10. Reinelt [97] reports a proof by Frieze that where the triangular inequality holds, the greedy heuristic is ratio bound by $\log(n)$.

Algorithm 10 Greedy Method
Pseudo code for greedy heuristic.

Input: A graph $G = (V, E)$ of a TSP with $n = |V|$ vertices.
Output: A tour π.
$T \leftarrow \varnothing$
$L \leftarrow E$
sort list L by edge cost in ascending order.
while $|T| \neq n$ **do**
    choose lowest $e_i \in L$ with the set $T \bigcup \{e_i\}$ consistent with a tour
    $T \leftarrow T \bigcup \{e_i\}$
    $L \leftarrow L - \{e_i\}$
end while
return set T as a tour π

3.1.8 Gutin and Yeo Algorithm

More recently Gutin and Yeo [48] have provided an approximation heuristic they term the *greedy expectation heuristic*. The authors provide details for both the ATSP and quadratic assignment problems.

For the ATSP, the algorithm operates by recursively constructing a tour. The algorithm starts with an empty tour and a complete directed graph K. At each step in the process an edge, e, is selected from the incumbent K such that the average cost of tours containing e is minimised. This edge is added to the partially completed tour. The recursion repeats with a modified K (excluding e and certain associated edges). It terminates when a complete tour is constructed.

3.1.9 Linear and Integer Programming

Linear programming concerns the optimisation of linear cost functions of variables subject to linear constraints. If the function is to be minimised, the problem can be expressed as follows:

$$\begin{aligned} \text{Minimize } z &= \sum_{j=1}^{n} c_j x_j \\ \text{Subject to } & \sum_{j=1}^{n} a_{i,j} x_j = b_i, \quad i = 1 \dots m \\ \text{and } & x_j \geq 0, \qquad\qquad j = 1 \dots n. \end{aligned} \tag{3.2}$$

The problem where the constraints are expressed as inequalities can be transformed into this form by the inclusion of extra variables termed *slack variables*. The most well known algorithm to address linear programming is the *simplex method* of Dantzig [104]. Smale proves the number of pivot operations required to solve a problem by the simplex method is, with high probability, $\mathcal{O}(\max(m, n))$ [104, 110]. However the algorithm is formally non-polynomial [91].

slack variables
simplex method

The special, and more difficult case, of *integer programming* occurs where some or all of the x_j are constrained to be integers. This situation occurs in the case of the TSP and many other combinational problems.

integer programming

Orman and Williams [89] survey eight integer programming formulations of the TSP. The best known of these is as follows. Let $G = (V, E)$ be the graph of the instance and let x_e be 1 if there is an edge $e \in E$ in a tour and 0 otherwise. Let c_e be the cost of edge $e \in E$ and let I_v be the set of edges incident to vertex $v \in V$.

I_v

$$
\begin{array}{lcll}
\text{Minimize } z & = & \sum_{e \in E} c_e x_e & \\
\text{Subject to} & & \sum_{e \in I_v} x_e = 2, & \text{for all } v \in V \\
\text{and} & & \sum_{e \in C} x_e \leq |C| - 1, & \text{for all cycles } C \text{ in } G, |C| < |V|.
\end{array}
\tag{3.3}
$$

A non-integer solution for an integer programming problem is a *linear programming relaxation* of the problem. The formulation in Equation 3.3 becomes such a relaxation if the x_e are allowed to take values in the interval $[0, 1]$. The linear programming relaxation of an instance of a TSP has application as a lower bound on the minimum solution cost of the instance.

linear programming relaxation

Methods that apply this lower bound to the branch and bound arrangement of Section 3.1.1 are termed *branch and cut algorithms*. The technique allows a *provably* globally optimal tour of an instance to be found in super-polynomial time. Applegate et al. [2] provide a recent and detailed survey of this approach.

branch and cut algorithms

The central difficulty in most applications of the linear programming formulations of the TSP and similar problems is that the number of constraints is super-polynomial on the size of the problem. For example, in Equation 3.3, the last term, which prevents cycles with fewer than $|V|$ vertices, expands to an exponential number of constraints on the problem size.

3.1.10 Neural Net Methods

Several artificial neural net methods to find approximate solutions to the optimisation problems have been proposed, starting with the Hopfield Tank model [56], the Potts model and the Kohonan self organising network. While it shows promise for other combinatorial problems, the Hopfield Tank network fails to provide good results even for quite small instances of the TSP and its architecture is not amenable to larger problems [92, 111, 122].

3.2 Tour Improvement and Landscapes

Tour improvement heuristics take an existing tour and seek to transform it to one with lower cost. The simplest approach, and one which is remarkably successful in the case of the TSP, is to start with a randomly generated tour and apply a process of iterative improvement. Each step in this process is a "small change" in the tour of that step, made in such a way that its cost decreases. The process ends when no further improvement can be made. Precisely what is meant by a small change in a tour gives rise to the concepts of a neighbourhood of a solution and to a landscape of a problem. We define these here.

3.2.1 Neighbourhoods and Landscapes

A landscape of a combinatorial optimisation problem (COP) is a formalisation of that problem which collects the three key aspects of the problem into a single mathematical entity [7]. More formally, a *landscape* of a combinatorial optimisation problem is the triple, $\mathfrak{L} = (S, f, \mathfrak{N})$, where the *solution space* S is the set of all solutions to the optimisation problem, the *objective function*, $f : S \rightarrow \mathbb{R}$ determines the cost of a particular solution to the problem, and the *neighbourhood*, $\mathfrak{N}$ is a collection of operations defined on the solution space. More precisely, a *move* is some function $m : S \rightarrow S$ which transforms an element s of S to a neighbouring element s' of S. A specific set of move functions forms a *neighbourhood* $\mathfrak{N}$. The multi-set $\mathfrak{N}(s)$ is the collection of images of $s \in S$ under all the moves of $\mathfrak{N}$. If $f(s) \leq f(s')$ $\forall s' \in \mathfrak{N}(s)$ then s is a *local minimum.* If $f(s) \geq f(s')$ $\forall s' \in \mathfrak{N}(s)$ then s is a *local maximum*. A *local optimum* is either a local maximum or a local minimum. If $f(s) \leq f(s')$ $\forall s' \in S$ then s is a *global minimum*. If $f(s) \geq f(s')$ $\forall s' \in S$ then s is a *global maximum*. If s is either of these, it is a global optimum denoted s^*. We can depict $\mathfrak{N}$ using a directed graph with vertices

COP

landscape

solution space

objective function

neighbourhood

move

neighbourhood

local, minimum, maximum, optimum

S and edges $s \to s'$ if $s' \in \mathfrak{N}(s)$. If s and s' are two solutions of a minimization problem with objective function, f then the *gain* of going from s to s' is the function $\mathrm{gain}(s, s') = f(s) - f(s')$. The terms gain and *move value* are synonymous. *gain*

3.2.2 Landscapes and the TSP

In terms of the TSP, the objective function is simply the cost of a tour as defined in Chapter 1. A move, as defined above, is then, a permutation τ on the tour π which transforms π to a neighbouring solution $\pi' = \tau(\pi)$. A specified multi-set of the images of such moves $\mathfrak{N}(\pi) = \left\{\pi'_1, \ldots, \pi'_{|\mathfrak{N}(\pi)|}\right\}$ defines the neighbourhood of a tour. The move functions can be thought of as *perturbation operations* which transform a tour to another tour. *perturbation operations*

Thus a landscape in the context of a TSP is the triple $\mathfrak{L} = (\Theta, \mathrm{cost}, \mathfrak{N})$, where Θ is the set of all tours on that TSP, $\mathrm{cost}(\pi) \to \mathbb{R}$ is the objective function, and $\mathfrak{N}$ is a multi-set consisting of the neighbours of each tour under a move. So $\mathfrak{N}(\pi)$ denotes the neighbours of a particular tour π.

If $\mathrm{cost}(\pi) \leq f(\pi')\ \forall \pi' \in \mathfrak{N}(\pi)$, then π is a local minimum and $\mathrm{cost}(\pi)$ is the cost of this local minimum. The local maximum and optimum are similarly defined. If π and π' are two solutions of a TSP then the *gain* in moving from π to π' is the function $\mathrm{gain}(\pi, \pi') = \mathrm{cost}(\pi) - \mathrm{cost}(\pi')$. A positive gain, $\mathrm{gain}(\pi, \pi')$, indicates π' is an improved (lower cost) solution compared to π. *gain*

Figures 3.2 and 3.3 show the landscape of a five vertex TSP under a typical move operation, the 2-opt or inversion move. This move is discussed in more detail in the next section.

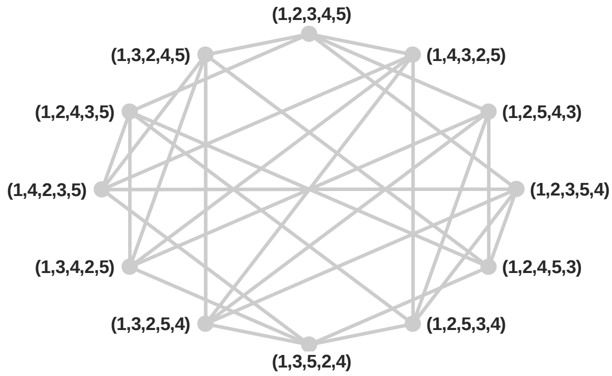


Figure 3.2: *The 2-opt landscape of a five city TSP. Each of the twelve vertices of the landscape is a tour and these are shown in Figure 3.3. In this graph each pair of directed edges connecting two tours is shown as a single undirected edge.*

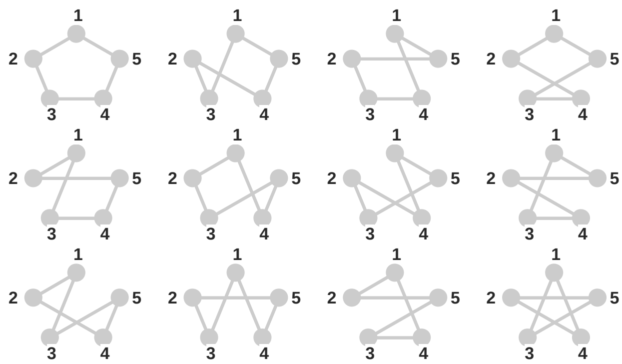


Figure 3.3: *The twelve tours in the 2-opt landscape of a five city TSP. Each tour is a vertex of the graph shown in Figure 3.2.*

3.2.3 Basic Tour Perturbation Operations

Jang et al. [58] note three basic tour perturbation methods: inversion, translation and swap. Colletti [26] considers group theoretical approaches to these moves and neighbourhood operations in both the TSP, ATSP and other variants of problem. Here we discuss the three basic moves.

Inversion

The first and most important operation we consider is the *inversion* move of Croes [28]. This consists simply of reversing a section of a tour. For this reason, the terms, inversion and *reversal* are synonymous. Given a tour

inversion

reversal

$$\pi = (i_1, \dots, i_{p-1}, i_p, \dots, i_q, i_{q+1} \dots, i_n) \tag{3.4}$$

an inversion produces a neighbouring tour

$$\pi' = (i_1, \ldots, i_{p-1}, i_q, i_{q-1}, \ldots, i_{p+1}, i_p, i_{q+1}, \ldots, i_n). \tag{3.5}$$

This has the effect of removing the two edges $\{i_{p-1}, i_p\}$ and $\{i_q, i_{q+1}\}$ then replacing them with the two edges $\{i_{p-1}, i_q\}$ and $\{i_p, i_{q+1}\}$. Figure 3.4 shows one possible inversion on the tour $(1, 2, 3, 4, 5, 6)$ to produce a tour $(1, 4, 3, 2, 5, 6)$. The path (2,3,4) is traversed in the opposite order in the second tour. This last point is significant for the ATSP.

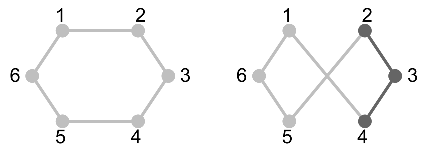


Figure 3.4: *An inversion move on a six city TSP. The darker edges would be traversed in reverse order.*

Translation

A translation consists of moving, without inversion, a path within a tour and placing it between two other vertices. Figure 3.5 provides an example. Here the path (2,3,4,5,6) in the tour (1,2,3,4,5,6,7,8) is translated to be positioned after vertex 7. Thus the tour becomes (1,7,2,3,4,5,6,8). Where k paths are translated the operation is termed k-translation.

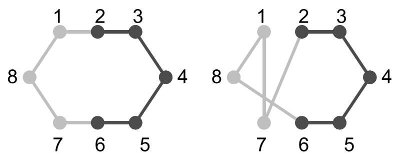


Figure 3.5: *A translation move on an eight city TSP.*

Swap

Here 2 or more vertices have their positions exchanged. If

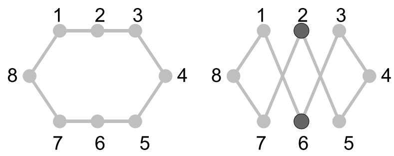
$$\pi = (i_1, i_2, \ldots, i_{p-1}, i_p, i_{p+1} \ldots, i_{q-1}, i_q, i_{q+1}, \ldots, i_n) \tag{3.6}$$

is our initial tour, switching vertices i_p and i_q will yield

$$\pi' = (i_1, i_2, \ldots, i_{p-1}, i_q, i_{p+1} \ldots, i_{q-1}, i_p, i_{q+1}, \ldots, i_n) \ . \tag{3.7}$$

The operation is typically denoted k-swap where k is the number of vertices exchanged. A *2-swap* is simply a 2-translation in which 2 singleton paths are exchanged. Figure 3.6 shows a 2-swap with vertices 2 and 6 exchanged. *2-swap*

Figure 3.6: *A 2-swap move on an eight city TSP.*



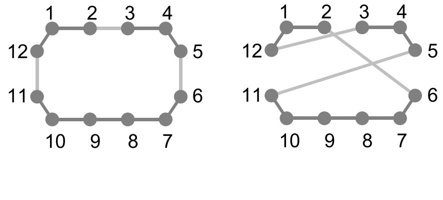
3.2.4 The k-opt Move

Lin [79] significantly extends the work of Flood [35] and Croes [28], by introducing the concept of the *k-optimal* tour. *k-optimal*

> *A tour is said to be k-optimal (or k-opt) if it is impossible to obtain a tour with smaller cost by replacing any k of its edges by any other set of k edges.* Lin [79].

This motivates the definition of the *k-opt* as one that removes k edges and replaces them with k different edges. In the case of a 2-opt move, two edges in a tour are replaced with two other edges (not from the original tour). This is equivalent to an inversion move on the tour. Lin generalized the 2-opt move to 3-opt move in which a tour with three paths undergoes translation with or without inversion. Figure 3.7 shows a 3-opt move in which two paths are inverted. *k-opt* 2-*opt move* 3-*opt move*

Figure 3.7: *A 3-opt move on a twelve city TSP.*



More recently Bentley uses a restricted version of the 3-opt move, termed the 2.5-opt. In the 2.5-opt move, one of the three paths is forced to be a 2.5-*opt move* singleton. Bentley [12] and Johnson and McGeoch [62] discuss this and other variations of the k-opt move.

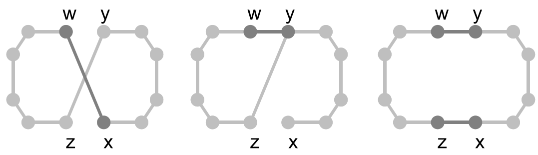


Figure 3.8: *The fast 2-opt move. We always require the first edge removed to be of lower cost than its replacement.*

In Chapter 10 we provide an $\mathcal{O}(n^4 \log(n))$ algorithm to compute the probability distribution of gains over the 2-opt landscape of an n city problem. We also prove that the variance of gains over the 2-opt landscape can be computed in $\mathcal{O}(n^2)$.

Efficient Search Over the k-opt Neighbourhood

For a 2-opt move to be effective in reducing tour length, at least one of the two edges added must be of lower cost than the edge it replaces. In addition, the order that the two edge replacement operations are carried out is immaterial. This motivates a simple but effective heuristic to speed the discovery of improving 2-opt move. We first note that ordered lists of the cost of each edge incident to each vertex can be pre-computed efficiently.

The operation of the Bentley heuristic is illustrated in Figure 3.8. We simply visit each vertex of a TSP as ordered in the current tour. To identify a likely improving 2-opt move, we first find a high cost edge $\{w, x\}$ then examine the edges incident to w to find an edge, $\{w, y\}$, not in the tour with $\text{cost}(\{w, y\}) < \text{cost}(\{w, x\})$. Having chosen these two edges the edges $\{y, z\}$ and $\{x, z\}$ are fully determined by the requirement to maintain a tour rather than two cycles. If $\text{cost}(\{w, y\}) + \text{cost}(\{x, z\}) < \text{cost}(\{w, x\}) + \text{cost}(\{y, z\})$ the actual move is made. This arrangement has a natural extension to k-opt [12, 62].

3.2.5 Distance in Landscapes

Boese [14] describes the bond distance between two tours as the total number of edges of a tour minus the number of edges that are present in both tours. This motivates the following definition. The *bond distance* between two n city tours, π and π', with edge sets E_π and $E_{\pi'}$ is $\mathrm{b}(\pi, \pi') = n - |(E_\pi \cap E_{\pi'})|$. The bond distance is a natural and easy to compute measure of distance. It is also independent of a particular landscape. The *k-opt distance,* $\mathrm{d}_k(\pi, \pi')$ between two tours π and π', is the minimum number of k-opt moves needed to transform π to π'. Given any two tours π and π', Kececioglu and Sankoff [68] prove that $\mathrm{d}_2(\pi, \pi') \leq 2\mathrm{b}(\pi, \pi')$. Solomon et al. [112] prove that finding a minimum length sequence of 2-opt moves to transform a tour π to a second π' is $\mathcal{NP}$-hard.

bond distance

k-opt distance, $\mathrm{d}_k(\pi, \pi')$

3.3 Tour Improvement Algorithms

3.3.1 Iterative Improvement

The simplest heuristic to find a local minimum in a landscape is to inspect the neighbours $\mathfrak{N}(s)$ of a solution s. Each time a solution s' is found with positive gain relative to s, the search is continued from that s' (as s). The process ceases when the neighbourhood of an s contains no improved solutions. This arrangement is termed *iterative improvement*. The key characteristic of iterative improvement is that it is guaranteed to find a local optimum. For this reason, it may be used to augment more complex heuristics. In terms of the TSP, a search that uses k-opt moves (for a fixed k) to iterate over a landscape is termed a *k-opt search*. Most implementations of 2-opt and 3-opt search use some refinement of iterate improvement such as the fast neighbourhood search noted in Section 3.2.4. Where only moves with maximum gain are made, the algorithm is termed *steepest descent*. Algorithm 11 illustrates this arrangement.

iterative improvement

k-opt search

steepest descent

Algorithm 11 Improve
The Iterative Improvement algorithm by Steepest Descent

Input: A COP
Input: A start solution s
Output: A local minimum solution s'

```
repeat
    choose s' ∈ 𝔑(s) with maximum gain g
    if (g > 0) then
        s ← s'
    end if
until g ≤ 0
return s
```

3.3.2 The Metropolis Algorithm

In its original form, the Metropolis algorithm (Algorithm 13) simulates the behaviour of systems governed by statistical mechanics [70]. In the context of optimisation, the algorithm is similar to iterative improvement, but utilises two stochastic mechanisms. These mechanisms allow local minima to be escaped [123].

Firstly, the perturbation function that selects a new solution s', given the current solution s makes a *random choice* from a neighbourhood of s. Secondly the *acceptance function* used to decide if the new solution s' is to be adopted has a random component. So, given a solution s and new solution s', the new solution is always accepted if the gain(s', s) is positive. In the case where gain$(s', s) \leq 0$ the probability of acceptance of s' is $p = e^{\frac{\text{gain}(s',s)}{t}}$. Here t is the *temperature*, a positive control parameter dictating the probability that a solution with non-positive gain will be adopted. Here p approaches 1 as t approaches ∞ and 0 as t approaches 0. This parameter has the same units as the cost function. Its name relates to the physical simulation role of the algorithm. The method is shown in Algorithms 13 and 12.

acceptance function

temperature

3.3.3 Simulated Annealing

Kirkpatrick, Gelatt and Vecchi [70] published a significant paper in which they demonstrate a connection between combinatorial optimisation and statistical mechanics. Their insight was motivated by the observation that if a

Algorithm 12 Accept
The Acceptance Function of the Metropolis algorithm

Input: The gain of a move g
Input: A control parameter t the temperature
if $g > 0$ **then**
  return $true$
end if
return $true$ with probability $e^{\frac{g}{t}}$ otherwise **return** $false$

Algorithm 13 Metropolis
Pseudo code of the Metropolis algorithm

Input: An instance I of a minimization COP
Input: A solution s
Input: A control parameter t the temperature
Input: The number of times to iterate k
repeat
  $s' \leftarrow \text{move}(s)$, a randomly selected neighbour of s.
  $g \leftarrow \text{cost}(s) - \text{cost}(s')$
  if $\text{accept}(g, t)$ **then**
    $s \leftarrow s'$
  end if
until k times
return s

body is heated and then cooled in a controlled manner, it will "freeze" to a stable low energy state. This process can be simulated by repeated applications of the Metropolis algorithm at different temperatures. Kirkpatrick et al. show that this method can be applied to combinatorial optimisation by identifying the cost function with the energy of the physical system, and the solution space with the state space of the physical system.

The scheme by which the system is "cooled" is termed the *annealing schedule*. Hajek [49] proves that simulated annealing will converge to the global optimal solution given infinite time by use of a logarithmic annealing schedule. This is disappointing at the practical level, but an important result nevertheless. Kirkpatrick et al. suggest a simple annealing schedule, where at each iteration the temperature is reduced by a small factor, that is $t \leftarrow \alpha t$ for $\alpha < 1$. This continues to be a widely used scheme.

annealing schedule

Johnson and McGeoch [61] report that, although slower than other algorithms for the TSP, such as Lin-Kernighan, simulated annealing can produce good results in terms of the quality of the optima it finds. Algorithm 14 provides the arrangement.

Algorithm 14 Simulated Annealing
Pseudo code for Simulated Annealing

Input: An instance I of a minimization COP
Input: A start solution s
Input: A start temperature t
Input: A stop temperature t_{limit}
Input: The number of times to iterate, k

```
1: repeat
2:     s ← Metropolis(I, s, t, k)
3:     t ← reduce(t)
4: until t ≤ t_limit
5: return s
```

3.3.4 The Lin-Kernighan Algorithm

In 1973, Lin and Kernighan [80] generalised Lin's 3-opt algorithm to search the solution space in a sequence of specific 2-opt moves. This algorithm has enjoyed great success being considered the "world champion" TSP heuristic for over a decade [61]. The algorithm still provides a basis for effective approaches. The *Lin-Kernighan algorithm* is, however, not straightforward. Johnson and McGeoch [61] comment that

Lin-Kernighan algorithm

> *Many authors rely on implementations that omit key components of the algorithm and end up producing tours that are worse on average than those produced by 3-opt.*

Helsgaun [50] citing a survey by Melamed et al. [85], comments that, as of 1989, no other implementation of the algorithm had shown the efficiency obtained by Lin and Kernighan. These statements are not surprising since, as Papadimitriou [90] points out in his analysis of its "neighbourhood" structure

> *... the algorithm is very complicated with unspecified parameters and details.*

The Lin-Kernighan method addresses the central deficiency of the 2-opt and 3-opt search methods, their inability to escape poor local minima. In the case of 2-opt search, this occurs because, any step that requires more than two edge replacements to make an improvement will not be executed. 3-opt search is limited similarly to three edge replacements. It is inefficient to overcome this problem by simply increasing the number of edge replacements to some k, since the complexity of this approach, for an exhaustive search is $\mathcal{O}(n^k)$, where n is the number of cities in the tour and k is the number of edges to be replaced.

The Lin-Kernighan algorithm (Algorithms 15 and 16) overcomes this problem by exploring a subset of the 2-opt landscape in a sequence of 2-opt moves such that, while not every move need have positive gain, the net gain in executing a subsequence is positive. The construction of this sequence is terminated when the net gain of the sequence becomes non-positive. The solution produced by a subsequence that starts from the initial move and has greatest net gain is then returned. Because not every move in the chain need have positive gain, 2-opt local minima may be escaped. If an improved tour is discovered, it may be refined further by reapplication of the algorithm.

In the Lin-Kernighan algorithm there are two types of 2-opt sequences. Firstly a sequence of the form $\phi_m \ldots \circ \phi_1 \circ \psi(\pi)$. The second form of sequence is $\phi_m \ldots \circ \phi_1 \circ \psi_2 \circ \psi_1(\pi)$. In both cases, any move ϕ_i has the following properties:

- It may have negative gain.
- The first edge added by each ϕ_i is chosen to maximise the gain of the 2-opt move.
- The first edge removed by ϕ_i is the last edge added by the previous move ϕ_{i-1} (or where $i = 1$ ψ or ψ_2).
- Each move ϕ_i removes one and only one edge, that is also in E_π.
- An edge that is removed is not re-added by any subsequent move.

In the first sequence, ψ is the initial 2-opt move. It has the following properties:

- It has positive gain.
- The bond distance $\mathrm{b}(\pi, \psi(\pi)) = 2$.

In the second type of sequence, $\psi_2 \circ \psi_1(\pi)$ are an initial pair of 2-opt moves which have the following properties:

- Together the pair are equal to a single 3-opt move with positive gain.
- The bond distance $\mathrm{b}(\pi, \psi_2 \circ \psi_1(\pi)) = 3$.

The initial step, either ψ or $\psi_2 \circ \psi_1$, is selected as the first move or moves found with positive gain. To discover this, move candidate replacement edges are tested in order of likelihood to maximise the gain of the resulting move. This operation is similar to that discussed in Section 3.2.4. The original formulation of the algorithm considered a maximum of $n * 100$ edges.

Algorithm 15 Lin-Kernighan Inner

Pseudo code of the Lin-Kernighan inner loop

Input: A tour π
Input: A vertex v
Output: A tour π_{best} with $\pi_{best} \leq \mathrm{cost}(\pi)$

$g_{net} \leftarrow 0$
$g_{best} \leftarrow 0$
$c_{last} \leftarrow \mathrm{cost}(\pi)$
$\pi_{best} \leftarrow \pi$
choose based on v, a good start move $\pi' \leftarrow \psi(\pi)$ **or** $\pi' \leftarrow \psi_2 \circ \psi_1(\pi)$
$g \leftarrow c_{last} - \mathrm{cost}(\pi')$
$g_{net} \leftarrow g$
$c_{last} \leftarrow \mathrm{cost}(\pi')$
if $g_{net} > 0$ **then**
    $g_{best} \leftarrow g$
    $\pi_{best} \leftarrow \pi'$
    while $g_{net} > 0$ **do**
        $\pi' \leftarrow \phi(\pi')$
        $g_{net} \leftarrow g_{net} + c_{last} - \mathrm{cost}(\pi')$
        $c_{last} \leftarrow \mathrm{cost}(\pi')$
        if $g_{net} > g_{best}$ **then**
            $g_{best} \leftarrow g_{net}$
            $\pi_{best} \leftarrow \pi'$
        end if
    end while
end if
return π_{best}

Algorithm 16 **Lin-Kernighan Outer**

Pseudo code of the Lin-Kernighan algorithm

```
generate a new tour π
π_best ← π
while still improvement do
    choose a start vertex v
    π ← Lin-Kernighan inner loop(π, v)
    if cost(π) < cost(π_best) then
        π_best ← π
    end if
end while
return π_best
```

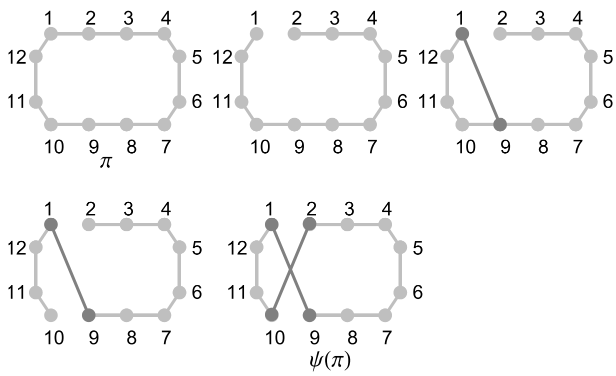


Figure 3.9: *The steps executed by the Lin-Kernighan algorithm in the initial 2-opt move. Read left to right beginning with π and ending with $\psi(\pi)$. The last edge added, {10,2}, will be deleted by ϕ_1, the next 2-opt move in the sequence of moves.*

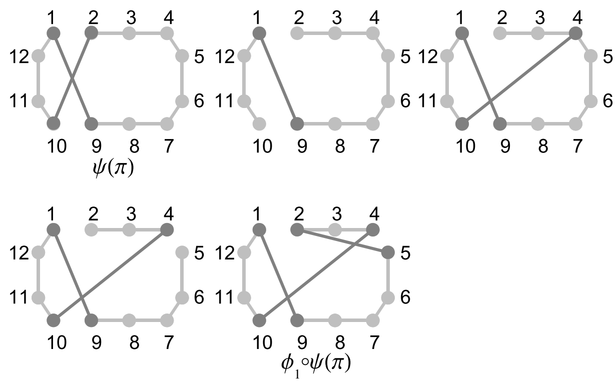


Figure 3.10: *The steps executed by the Lin-Kernighan algorithm in the move ϕ_1 on $\psi(P)$ to form $\phi_1 \circ \psi(P)$.*

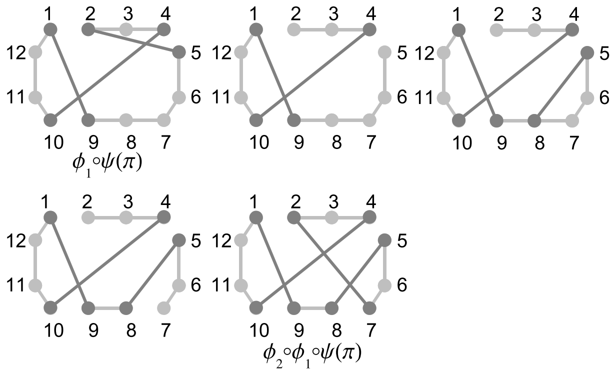


Figure 3.11: *The steps executed by the Lin-Kernighan algorithm in the move ϕ_2 on $\phi_1 \circ \psi(P)$ to form $\phi_2 \circ \phi_1 \circ \psi(P)$.*

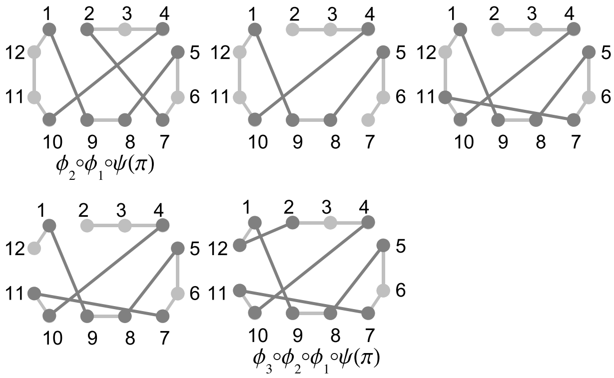


Figure 3.12: *The steps executed by the Lin-Kernighan algorithm in the move ϕ_3 on $\phi_2 \circ \phi_1 \circ \psi(P)$ to form $\phi_3 \circ \phi_2 \circ \phi_1 \circ \psi(P)$.*

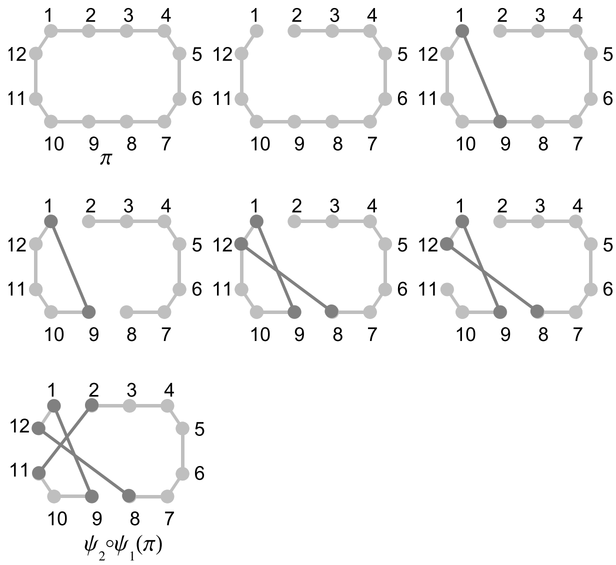


Figure 3.13: *The steps in executing an initial 3-opt move by the Lin-Kernighan algorithm. The last edge added, {11,2}, will be deleted by ϕ_1, the next 2-opt move in the sequence of moves.*

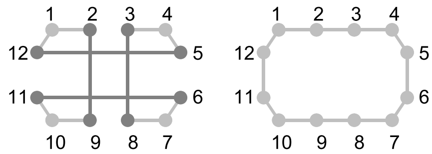


Figure 3.14: *An example of a double bridge move. One tour cannot be transformed to the other by a sequence of 2-opt moves without re-adding an edge.*

The tight criteria for the 2-opt move construction ensures that the algorithm efficiently explores a subset of the 2-opt solution space. Nevertheless, the fact that this is only a subset limits the algorithm's utility. In particular, the *double bridge move* of Figure 3.14 cannot be "deconvolved" by the Lin-Kernighan algorithm since, to do this by a sequence of 2-opt moves would entail removing an edge and subsequently replacing it, an action the algorithm does not allow. *double bridge move*

Number	3-opt search		Lin-Kernighan		Simulated Annealing	
of Runs	Excess	Run Time	Excess	Run Time	Excess	Run Time
1	3.69%	1.6 min.	2.18%	4.0 min.	1.71%	791 min.
10	3.07%	6.1 min.	1.75%	30.3 min.	1.33%	7903 min.
50	2.84%	26.1 min.	1.56%	147.1 min.	1.22%	39513 min.

Table 3.1: *The percentage excess over a theoretical lower bound (Held Karp), together with the run time in minutes for: 3-opt search, Lin-Kernighan and simulated annealing. A range of geometric problems both random and real-world are used. Johnson [59]*

Other Optimisations Noted by Lin and Kernighan

Lin and Kernighan implement two useful heuristics which have general application. We note these here.

The time taken to confirm that a local minimum is in fact a local minimum solution can be significant as its entire neighbourhood must be searched. Lin and Kernighan term this time the *checkout time* and note that is typically 30 to 50 percent of the run time of an iteration. A useful optimisation is, therefore, to compare tours that are likely to be local minima against a set of previously found local minimum. *checkout time*

They also note that local minima tend to have edges in common. They provide a heuristic to exploit this phenomenon which they termed *reduction*. *reduction* Reduction works by first finding between two and five local optima, then computing the set, C, of edges common to all of them. In subsequent iterations, no move is allowed that will result in an element of C being removed from a tour. As new good local minima are found, the set C may be updated.

In fact, the observation that good local minima tend to share many edges is very significant and was later noted by Kirkpatrick et al. [70] and investigated by Boese [14].

Chained and Iterated Lin-Kernighan

In the *chained Lin-Kernighan algorithm*, Martin et al. [83] augment the Lin-Kernighan approach in two ways. Firstly, after each run of the inner loop of the Lin-Kernighan algorithm, the solution is perturbed by application of the double bridge 4-opt move with positive gain. Recall that this move cannot be made by the original algorithm. Their second innovation is to adopt this perturbed tour on a stochastic basis, with the acceptance function of the Metropolis algorithm. Thus a perturbed solution is accepted as the new tour, if its relative gain, g is positive, or with some probability $p = \mathrm{e}^{\frac{g}{t}}$ where t is the temperature parameter. They termed the resulting algorithm, the *large step Markov chain* algorithm.

chained Lin-Kernighan algorithm

large step Markov chain

Johnson produces a simplified version of the chained Lin-Kernighan algorithm termed *iterated Lin-Kernighan*, in which the stochastic acceptance function is removed. So a solution is accepted only if it has positive relative gain. The iterated Lin-Kernighan algorithm chooses a double bridge 4-opt move randomly. Thus it has a stochastic component. Applegate et al. have describe a non-stochastic variant of chained Lin-Kernighan, in which the double bridge perturbation function of Martin et al. is used with a non-random acceptance function [2].

iterated Lin-Kernighan

Helsgaun Variation

More recently, Helsgaun [50] has provided a very successful variation of the Lin-Kernighan approach. The key refinements of Helsgaun's method are: the initial move may be a restricted 4-opt, the sequence of following moves may be up to a restricted 5-opt move. The restrictions on these moves are enforced by allowing edge replacements only from a candidate set. This candidate set is chosen by consideration of the nearest neighbours of a vertex together with the minimum spanning tree associated with the problem. The rationale is that the edges in the minimum tour often appear in the minimum spanning tree. Indeed Helsgaun reports that in a typical case, 70 to 80 percent of the edges of the minimum tour are in the minimum one tree.

3.3.5 Tabu Search

Pseudo code of a simple Tabu Search is shown here as Algorithm 17. Extensive references to Tabu Search are [40–42]. The central feature of the approach is the use of memory in the search in the process. At the simplest level, tabu search operates much like interactive improvement, but with additional restrictions on which solutions in the neighbourhood of some solution s may be visited. At the conceptual level, the restrictions are enforced by maintenance of a set of tabu solutions, T. These are only moved to if there is good reason to do so, with the decision to explore these dependent on the *aspiration* criterion. At the practical level, the tabu set is maintained as a combination of previously visited moves, a history set, and/or set of rules governing which moves are valid given the current solution, its neighbourhood and the history set. *aspiration*

The algorithm may alter the neighbourhood selection process to become more restrictive, favouring those solutions with desirable properties. This process is termed *intensification*. Typically, encountering solutions with properties in common with previously discovered good solutions would invite intensification. The opposite process, widening the neighbourhood selection criterion is termed *diversification*. Glover [40] notes that approximating the minimum cost solution over $N(s) - T$ at line 6 may be appropriate where this set is large. *intensification* *diversification*

In terms of a practical realisation of these ideas to the TSP, Zachariasen and Dam [125] discuss an implementation using the Tabu Search approach

Algorithm 17 Tabu Search
Pseudo code for a simple Tabu search algorithm

```
Input: An instance I of a minimisation COP
Input: An iteration limit k_l.
Output: A solution to I
choose a start solution s
s* ← s
T ← ∅
k ← 0
while N(s) − T ≠ ∅ and k < k_l do
    choose s' ∈ N(s) − T with minimum cost over this set.
    s ← s'
    if cost(s) < cost(s*) then
        s* ← s
    end if
    k ← k + 1
    update(T)
end while
return s*
```

with neighbourhood selection based on each of the 2-opt, 3-opt, the Lin-Kernighan and the flower heuristic. The latter is a modification of the Lin-Kernighan heuristic which results in slightly larger neighborhoods. On 4 geometric problems from TSPLIB ranging in size from 318 to 2392 cities, the Lin-Kernighan-Tabu search combination gave results on average 0.38% above optimal, while the flower Tabu search combination produced results on average 0.28% above optimal. However, the authors concede that the large step Markov chain Lin-Kernighan variant of Martin et al. [83] outperform their implementation with an average tour cost of 0.15% above the optimal cost over the four instances.

3.3.6 Evolutionary Algorithms

Genetic and evolutionary algorithms attempt to mimic the biological process of evolution by natural selection. The approach was initially investigated by Box [16], Holland [54, 55], and Bremermann and Rogson [18] among others. Brady [17] was the first to apply the method to the TSP and this was soon followed by Grefenstette et al. [45] and several others. Also of note is the contribution of Mühlenbein et al. [88] who provide a fast parallelised implementation together with several other optimisations [61].

Algorithm 18 illustrates the general procedure. It starts by choosing a set of solutions, typically randomly, and computing the cost of each. It then begins an iterative simulation of reproduction over this population. The three primary operations involved are: firstly, the *selection* process of pairs of solutions to act as parents; secondly, the *crossover* operation to combine information from the two parents and construct a pair of offspring, c_1 and c_2; and thirdly, the *mutation* operation, which with probability ρ alters each of c_1 and c_2. As with the biological case, this mutation will probably be deleterious or even result in an invalid solution. However the occasional beneficial alteration will increase the fitness of the child solution (reduce its cost) and result in an increased chance of that solution being selected as a parent. In this way, beneficial properties tend to propagate through the population.

selection *crossover* *mutation*

In terms of the TSP, the principal difficulty is providing a workable crossover operation. Various crossover mechanisms are considered in [77] and [94], and most recently in [95]. General implementation issues are discussed in Whitley [121]. It is noteworthy that in several implementations appearing in the TSP literature, the initial population are created randomly and then improved by local search. This is to say, the initial population are already good solutions. Under these circumstances it would be expected that the final solution produced is as well.

3.3.7 Ant Colony Optimisation

Ant colony optimisation simulates the ability of ants to find optimal physical paths through complex environments. Ants do this, in part, by laying trails of chemical pheromones. When an ant is faced with a choice of paths it will follow the path of highest pheromone concentration. However the pheromone trails themselves tend to evaporate.

Consider the case where a trail splits in two and later rejoins. Let one section be longer than the other and let the number of ants entering each section be equal. The ants taking the shorter route will reach the join in

Algorithm 18 Genetic Algorithm
Pseudo code for a simple Genetic Algorithm

Input: An instance I of a minimisation COP
Input: An iteration limit k_l.
Input: A cost limit c_l.
Output: A set, S, of solutions to I
$k \leftarrow 0$
choose An initial population of solutions S.
Compute and store the cost of each solution in S.
$s^* \leftarrow$ the solution in S with minimum cost.
while $cost(s^*) > c_l$ **and** $k < k_l$ **do**
  Create a new population S' by:
  choose $P \subseteq S \times S$ of parent pairs (p_1, p_2) of solutions
  for all pairs $(s_1, s_2) \in P$ **do**
    Construct a pair of child solutions (c_1, c_2) from each pair (p_1, p_2) by:
    $(c_1, c_2) \leftarrow$ **crossover**$((\mathbf{s_1}, \mathbf{s_2}))$
    $c_1 \leftarrow$ **mutate**$(\rho, \mathbf{c_1})$
    $c_2 \leftarrow$ **mutate**$(\rho, \mathbf{c_2})$
    $S' \leftarrow S' \bigcup \{c_1\} \bigcup \{c_2\}$
  end for
  $S \leftarrow S'$
  Compute and store the cost of each solution in S.
  $s^* \leftarrow$ the solution in S with minimum cost.
  $k \leftarrow k + 1$
end while
return s^*

the paths before those taking the longer route. Therefore, an ant coming from the opposite direction at this time will only encounter pheromone on the shorter route and will be more likely to take that route. This will further increase the concentration of pheromone on the shorter route. The net result is that the shortest route will have the highest concentration of pheromone and will be the most attractive route to other ants [31, 107, 115].

Algorithm 19 provides a TSP oriented Ant Colony optimisation routine. The global variable $\tau(i, j)$ simulates the level of pheromone on each edge. The constant γ of Equation 3.9 is the degree of evaporation of a trail. The probability of an ant making a transition from a city i to city j is determined by Equation 3.8 and this is dependent on $\tau(i, j)$, together with the cost of the unvisited cities for an ant. It is noteworthy that this function determines the numerical reciprocal of the cost between i and j. This is in contrast to an actual mechanism outlined above for natural ants (where each ant has no

Algorithm 19 Ant Colony Optimisation
Pseudo code for an Ant Colony Search Algorithm for the TSP

Input: A graph $G = (V, E)$ of a TSP. A start vertex $s \in V$.
Input: An iteration limit l.
Input: The number of ants $a < |V|$
Input: Initial trail intensity t
Input: α Weights effect of cost.
Input: β Weights effect of trail.
Input: $\gamma \leq 1$ Factor by which trail is reduced.
Output: A tour, π

for all $k \leq a$ ants **do**
  choose i_k Randomly position ant k to a distinct city i_k.
  $P_k \leftarrow i_k$ Initial a path P_k for each ant.
end for
for all edges $\{i, j\} \in E$ **do**
  $\tau(i, j) \leftarrow t$
end for
$n \leftarrow 1$
while $n < l$ **do**
  $m \leftarrow 1$
  while $m < |V|$ **do**
    for all ants k, $k \leq a$ **do**
      $i \leftarrow position(k)$
      choose with probability $p(i_k, j, k, P_k)$ given by Equation 3.8, the next city i'_k to move ant k.
      $i_k \leftarrow i'_k$
      $P_k \leftarrow P_k$ append i_k
    end for
    $m \leftarrow m + 1$
  end while
  for all ants k, $k \leq a$ **do**
    compute the cost of P_k as a tour
    update the pheromone level , $\tau(i, j)$, on each edge by Equation 3.9
  end for
  $n \leftarrow n + 1$
  return the best found P_k as a tour π.
end while

knowledge of the distance). The constants α and β weight the effects of the cost function and $\tau(i, j)$. These are analogues to such factors as the clarity of the pheromone trial.

$$p(i, j, k, P_k) = \begin{cases} \frac{\tau(i,j)^{\alpha} 1/\mathrm{cost}(i,j)^{\beta}}{\sum_{q \in V - P_k} \tau(i,q)^{\alpha} 1/\mathrm{cost}(i,q)^{\beta}} & \text{if } j \in V - P_k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

$$\tau(i,j) \leftarrow \gamma\tau(i,j) + \sum_{k=1}^{a} \tau_k(i,j) \tag{3.9}$$

$$\tau_k(i,j) = \begin{cases} \frac{1}{\mathrm{cost}(P_k)} & \text{if } \{i,j\} \text{ is an edge in } P_k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

Bui and Colpan [19] describe a hybrid algorithm, adding a local search phase to ant colony optimisation. They apply their approach to the Euclidean TSP and obtain strong results in terms of tour quality in 40 instances taken from TSPLIB. Specifically, they obtain optimal solutions in 37 instances and very near optimal results in the remaining three instances. The largest problem investigated was 1577 cities in size. Unfortunately no run times were provided.

Among others, Gómez and Barán [44] argue, that both Ant Colony optimisation and Evolutionary algorithms work in large part because of the structure of the landscapes explored by Boese [14]. We discuss this further in Section 4.6 and Theorem 31 of Chapter 6 provides a tractable method to aid in the analysis of the operation of these algorithms.

3.4 Summary and Relevance to this Thesis

In this chapter we have outlined a subset of the optimisation algorithms and heuristics applicable to the TSP. Our exposition has, for the most part, been restricted to techniques with application beyond the Euclidean case. In doing this we have two goals in mind firstly, to provide a review of the literature and secondly to, introduce those techniques for which our upcoming results have most relevance.

Specifically, in Chapter 5 we provide a Karp reduction from the Hamiltonian cycle problem to the TSP that has the property that each tour has distinct cost. We argue that those methods that rely on simple edge cost comparison could be applied to the TSPs produced by our reduction. This would include among others the nearest neighbour algorithm, the greedy algorithm and iterative improvement in various landscapes.

In Theorem 31 of Chapter 6 we provide a constructive proof that, given a subgraph, P, of any tour in an n city TSP, the expected value of all tours containing P can be computed in $\mathcal{O}(n^2)$. This has application in the analysis of several of the methods discussed in this Chapter including the nearest neighbour and greedy methods. In particular, it applies to the branch and bound algorithm of Section 3.1.1 since the expected cost of tours with edges in the sets $v.M$ of Algorithm 3 can be computed. As mentioned above, this theorem also has application in analysing the relationships between the Ant Colony optimisation and Evolutionary algorithms noted by Gómez and Barán [44].

In Chapter 10 we provide a method to compute the expected value of gains over the 2-opt landscape of the TSP. We view this as useful in the analysis of those algorithms that make use of this basic move.

In the next chapter we provide a review of the statistical background underpinning our central results. In addition we give a review of the known statistical properties of the solution space and landscapes of the TSP.

Chapter 4

Statistics and the TSP

In this chapter we briefly review the statistical bases of our results and previous probabilistic work on the TSP.

4.1 Statistical Preliminaries

We start by providing a concise review of the probability theory underpinning the contents of this and subsequent chapters. A complete reference, with proofs, is provided by the first two chapters of Shiryaev [105]. The properties of probability distributions and moments are summarised in Balakrishnan and Nevzorov [6]. Molloy [87] discusses the application of the first and second moments to the analysis of combinatorial problems.

4.1.1 Measure Spaces

A measure space is a formalisation that generalises and makes precise familiar concepts such as lengths and areas. Let Ω be a set and let $\mathcal{A}_\Omega$ be a collection of subsets of Ω. If Ω and $\mathcal{A}_\Omega$ satisfy the following three conditions,

- $\Omega \in \mathcal{A}_\Omega$
- if $\omega \in \mathcal{A}_\Omega$ then $\Omega - \omega \in \mathcal{A}_\Omega$
- if $\omega_i \in \mathcal{A}_\Omega$ for each $i \geq 1$ then $\bigcup\limits_{i=1}^{\infty} \omega_i \in \mathcal{A}_\Omega$

then $\mathcal{A}_\Omega$ is a *σ-algebra* or *σ-field.* and the pair, $(\Omega, \mathcal{A}_\Omega)$, is termed a *measurable space*. Given a measurable space $(\Omega, \mathcal{A}_\Omega)$, a function $m : \mathcal{A}_\Omega \to \mathbb{R}^+$ is a *finite measure* if

σ-algebra

measurable space

finite measure

- $m(\varnothing) = 0$
- $m\left(\bigcup_{i=1}^{\infty} \omega_i\right) = \sum_{i=1}^{\infty} m(\omega_i)$ where the members ω_i of $\mathcal{A}_\Omega$ are pairwise disjoint.

If this is the case then the triple, $(\Omega, \mathcal{A}_\Omega, m)$ is a *measure space*.

measure space

4.1.2 Probability Spaces

A *probability space* is a measure space, $(\Omega, \mathcal{A}_\Omega, \mathrm{P}_\Omega)$, with $\mathrm{P}_\Omega(\Omega) = 1$. As a consequence of this definition, P_Ω has the following properties

probability space

- $\mathrm{P}_\Omega(\emptyset) = 0$
- for $\omega_i, \omega_j \in \mathcal{A}_\Omega$, $\mathrm{P}_\Omega(\omega_i \bigcup \omega_j) = \mathrm{P}_\Omega(\omega_i) + \mathrm{P}_\Omega(\omega_j) - \mathrm{P}_\Omega(\omega_i \bigcap \omega_j)$
- for $\omega_i, \omega_j \in \mathcal{A}_\Omega$ with $\omega_j \subseteq \omega_i$ then $\mathrm{P}_\Omega(\omega_j) \leq \mathrm{P}_\Omega(\omega_i)$.

This formalisation provides a conceptual framework to model situations such as experiments. A typical arrangement is for Ω, to be a set of elementary events, say the outcomes of the experiment. $\mathcal{A}_\Omega$ is then, the collections of all outcomes, and P_Ω provides the probability of the occurrence of any collection of events in $\mathcal{A}_\Omega$.

4.1.3 Distribution Functions and Random Variables

A function $X : \mathcal{A}_\Omega \to \mathbb{R}$ is a *random variable*. It is typical in the statistics literature to use upper case characters to label random variables and lower case characters as the values they may realise on single repeatable experiments.

random variable

Let X be a random variable and let $X \leq x$ be all the events $\omega \in \mathcal{A}_\Omega$ such that $X(\omega) \leq x$ (and similarly for $X = x$ and other relationships). Then $F_X(x) = \mathrm{P}_\Omega(X \leq x)$ is the *cumulative distribution function* or *distribution function* of X.

$X \leq x$

cumulative distribution function

distribution function

Discrete Random Variables

X is a *discrete random variable* if it has a countable range, $R_X = \{x_1, x_2, \ldots\}$. If this is the case then there is a function $\mathrm{p} : \mathbb{R} \to [0, 1]$ termed the *probability mass* or *probability function* defined as

probability mass
probability function

$$\mathrm{p}(x) = \mathrm{P}_\Omega(X = x). \tag{4.1}$$

In this case, the cumulative distribution function is $F_X(x) = \sum_{x_i \in R_X | x_i \leq x} \mathrm{p}(x_i)$. As a consequence of the properties of P_Ω, we have $1 = \sum_{x \in R_X} \mathrm{p}(x)$.

Continuous Random Variables

A function F is *absolutely continuous* on $(-\infty, +\infty)$ if, and only if, there is a function f such that for all $b > a$, $F(b) - F(a) = \int_a^b f(x)\mathrm{d}x$. (Where this is the case, f is the derivative of F.) Let X be a random variable with an absolutely continuous cumulative distribution function, F_x. Then the function, $f : \mathbb{R} \to [0, 1]$ such that $F_X(x) = \int_{-\infty}^{x} f(x)\mathrm{d}x$ is the *probability density function* of X. Again, as a consequence of the properties of P_Ω, we have $1 = \int_{-\infty}^{+\infty} f(x)\mathrm{d}x$.

absolutely continu-ous

probability density function

Support of Distributions

Informally, the support of a cumulative distribution function are those numbers with non-zero probability of occurring. In the case of a discrete random variable with probability mass p, the support is simply the set of numbers, S, such that $\mathrm{p}(x) > 0$ for $x \in S$. In the case of continuous random variables, the situation is less clear since the probability of any particular number occurring is 0. The corresponding concept is the probability of the random variable taking a value between two points. This motivates the following definition.

A number, x, is in the support of a random variable if, and only if, for any real number $\epsilon > 0$, $F_X(x + \epsilon) - F_X(x - \epsilon) > 0$. The set of all x is termed the *support of* F_X. Where the support is *compact* it is closed and bounded [22, 109].

support of F_X

compact support

The General Case

In the most general case, any cumulative distribution function, F can be written as $F(x) = w_d F_d(x) + w_c F_c(x) + w_s F_s(x)$ where F_d is a discrete cumulative distribution function, F_c is an absolutely continuous cumulative distribution function, and F_s is a singular cumulative distribution function. The singular case is pathological and will not be considered further. The non-negative weights w_d, w_c, w_s sum to 1.

The term *probability distribution* is something of a "catch all phrase", it refers to either the probability mass function, in the case of a discrete random variable, the probability density function in the continuous case, or in either case, the cumulative distribution function. The last of these, of course, provides the same "information" as the first two.

probability distribution

4.1.4 Independent Random Variables

Let $X_1, X_2, \ldots, X_m$ be a finite number of random variables with cumulative distribution functions $F_{X_1}, F_{X_2}, \ldots, F_{X_m}$. For each X_i let $X_i \leq x_i$ be the events $\omega_i \in \mathcal{A}_\Omega$ such that $X_i(\omega_i) \leq x_i$. The *joint cumulative distribution function* $F_{X_1,X_2,\ldots,X_m} : \mathbb{R}^m = [0,1]$ is defined as

joint cumulative distribution function

$$F_{X_1,X_2,\ldots,X_m}(x_1, x_2, \ldots, x_m) = \mathrm{P}_\Omega(X_1 \leq x_1 \wedge X_2 \leq x_2 \wedge \ldots \wedge X_m \leq x_m) \quad (4.2)$$

The random variables $X_1, X_2, \ldots, X_m$ with joint domain forming a product space are said to be *independent* or *collectively independent*, if

independent

collectively independent

$$F_{X_1,X_2,\ldots,X_m}(x_1, x_2, \ldots, x_m) = F_{X_1}(x_1)F_{X_2}(x_2)\ldots F_{X_m}(x_m) \quad (4.3)$$

for all $x_1, x_2, \ldots, x_m \in \mathbb{R}$. $X_1, X_2, \ldots, X_m$ are *pairwise independent* when each pair is independent.

pairwise independent

4.1.5 Expected Value and Moments

In the most general case, the *expected value*, $\mathrm{Ex}(X)$, of a random variable X is its Lebesgue integral, that is $\mathrm{Ex}(X) = \int_\Omega X(\omega)\mathrm{P}_\Omega(d\omega)$. In the case of

expected value

discrete random variables, this reduces to $\mathrm{Ex}(X) = \sum_{x \in R_X} x\mathrm{p}(x)$. Where the domain of X is the *finite* set of events Ω, it is convenient to write this as

$$\mathrm{Ex}(X) = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} X(\omega). \tag{4.4}$$

The terms expected value and *mean value* are synonymous. For any finite set of, not necessarily discrete, random variables $X_1, X_2, \ldots, X_m$ it is easy to show that

mean value

$$\mathrm{Ex}\left(\sum_{i=1}^{m} X_i\right) = \sum_{i=1}^{m} \mathrm{Ex}(X_i). \tag{4.5}$$

If two random variables, X_1 and X_2, are independent then

$$\mathrm{Ex}(X_1 X_2) = \mathrm{Ex}(X_1)\mathrm{Ex}(X_2). \tag{4.6}$$

The kth *moment about the mean* or *central moment* is defined as

moment about the mean
central moment

$$\mathrm{Mm_k}(X) = \mathrm{Ex}((X - \mathrm{Ex}(X))^k). \tag{4.7}$$

The second moment about the mean, $\mathrm{Mm_2}(X)$, is the *population variance*, $\mathrm{Var}(X)$, and reflects the spread of values around the mean of X. The positive square root of the variance is the *standard deviation*.

population variance
standard deviation

Moment about the Origin

The kth *moment about the origin* or *raw moment* of X is defined as

moment about the origin
raw moment

$$\mathrm{Mo_k}(X) = \mathrm{Ex}(X^k). \tag{4.8}$$

In the case of the second moment the positive square root of $\mathrm{Mo_2}(X)$, $\sqrt{\mathrm{Mo_2}(X)}$, is termed the *quadratic mean* or *root mean square*. It forms a biased measure of central tendency of $|X|$. Also in the case of the second moment it is useful to note that for any random variable X

quadratic mean
root mean square

$$\mathrm{Var}(X) = \mathrm{Mo_2}(X) - (\mathrm{Ex}(X))^2. \tag{4.9}$$

More generally

$$\mathrm{Mm_k}(X) = \sum_{j=0}^{k} -1^j \binom{k}{j} \mathrm{Ex}(X)^j \mathrm{Mo_{k-j}}(X). \tag{4.10}$$

Luenberger [82] discusses the algebraic properties of the second moment as a vector norm.

Factorial Moments

Where a random variable X has range in $0, 1, 2, \ldots$ the positive *factorial moments* are

factorial moments

$$\mathrm{Mf_k}(X) = \mathrm{Ex}(X(X-1)(X-2)\ldots(X-k+1)). \tag{4.11}$$

Balakrishnan and Nevzorov [6] provide

$$\begin{aligned}
\mathrm{Mf_1}(X) &= \mathrm{Mo_1}(X) \\
\mathrm{Mf_2}(X) &= \mathrm{Mo_2}(X) - \mathrm{Mo_1}(X) \\
\mathrm{Mf_3}(X) &= \mathrm{Mo_3}(X) - 3\mathrm{Mo_2}(X) + 2\mathrm{Mo_1}(X) \\
\mathrm{Mf_4}(X) &= \mathrm{Mo_4}(X) - 6\mathrm{Mo_3}(X) + 11\mathrm{Mo_2}(X) - 6\mathrm{Mo_1}(X).
\end{aligned} \tag{4.12}$$

Statistics Relating to the Moments

The cubed term in the third moment about the mean ensures that information about the sign of each difference from the mean is not entirely lost. It therefore encapsulates some information about the symmetry of a distribution about the mean. This motivates the definition of the well known statistic, the *skewness*, as

skewness

$$\alpha_3(X) = \frac{\mathrm{Mm_3}(X)}{\mathrm{Mm_2}(X)^{3/2}}. \tag{4.13}$$

As with the second moment about the mean, the fourth moment about the mean, $\mathrm{Mm_4}(X)$ provides a measure of the spread of values about the mean. However, its higher exponent results in emphasis to data points distant from the mean. Comparing the second and fourth central moments, therefore reflects the degree of the peakedness of a probability distribution. The, again well known statistic, the *kurtosis* is defined as

kurtosis

$$\alpha_4(X) = \frac{\mathrm{Mm}_4(X)}{\mathrm{Mm}_2(X)^2}. \tag{4.14}$$

Under this definition the kurtosis of a standard normal distribution is 3.[1] A distribution with a kurtosis of less than 3 is termed *platykurtic* while one with kurtosis of greater than 3 is termed *leptokurtic*.

platykurtic
leptokurtic

Let X and Y be random variables with $\mathrm{Var}(X) > 0$ and $\mathrm{Var}(Y) > 0$. The covariance is

$$\mathrm{Cov}(X,Y) = \mathrm{Ex}((X - \mathrm{Ex}(X))(Y - \mathrm{Ex}(Y))). \tag{4.15}$$

4.1.6 Distributions and Processes

The Normal Distribution

Certainly, the most well known distribution is the *normal* or *Gaussian distribution*. The probability density function of a normally distributed random variable X is

normal
Gaussian distribution

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{4.16}$$

where the parameters $\mu = \mathrm{Ex}(X)$ and $\sigma^2 = \mathrm{Var}(X)$. Figure 4.1 shows both the density function and cumulative distribution function of the normal distribution for various parameters. Where $\mu = 0$ and $\sigma^2 = 1$ the distribution is termed a *standard normal distribution*.

standard normal distribution

[1] The kurtosis is often defined with the additive constant -3 to give the standard normal distribution a kurtosis of 0.

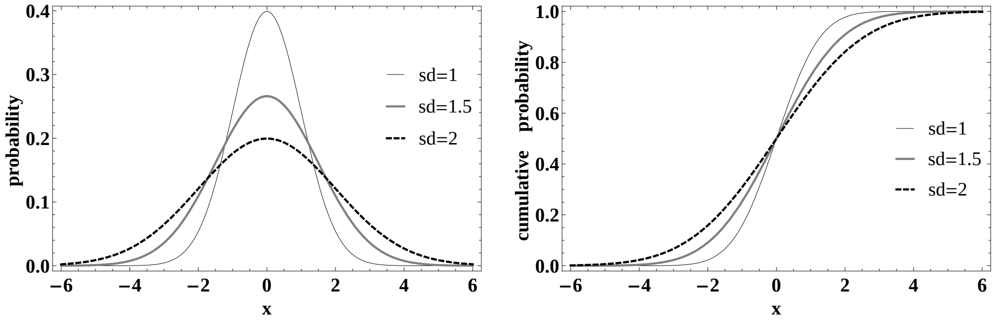


Figure 4.1: *The normal probability distribution with mean 0 and standard deviations 1, 1.5 and 2. The probability density function (left). The cumulative distribution function (right).*

A *normal random variable* is a random variable with a normal distribution. More generally a random variable with distribution $\mathcal{D}$ is termed a $\mathcal{D}$ random variable. *normal random variable*

The Beta and Weibull Distributions

The probability density function of a random variable, X with Beta distribution is

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{4.17}$$

where Γ is the well known generalisation of the factorial function, the gamma function. In this form the distribution has support $[0, 1]$, however a more general form is commonly used which allows the support to be an arbitrary interval. This, more general form, is arrived at by a simple linear transform of X [6]. Figure 4.2 illustrates the effect of changing the parameters α and β.

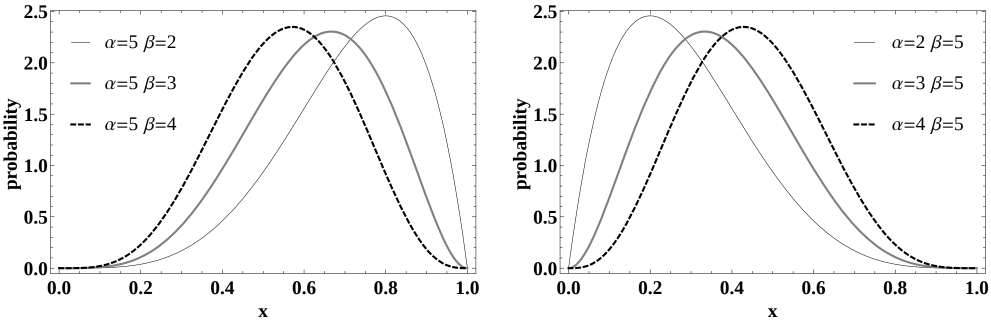


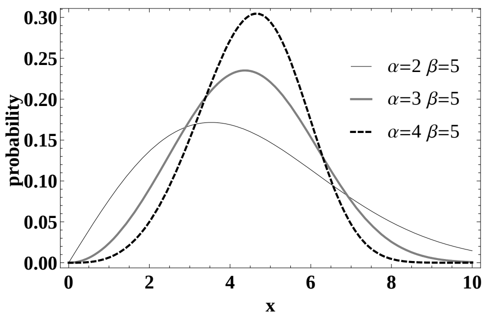
Figure 4.2: *The probability density function of the Beta distribution. The effect of varying the parameters α and β.*

As with the Beta distribution, the Weibull distribution is quite flexible. It finds application in modelling the extreme behaviour of systems [98]. The probability density function of the Weibull distribution is given by

$$f(x) = \begin{cases} \alpha\beta^{-\alpha}x^{(1-\alpha)}e^{-\left(\frac{x}{\beta}\right)^{\alpha}} & 0 \leq x \\ 0 & \text{otherwise.} \end{cases} \tag{4.18}$$

Figure 4.3 shows this function.

Figure 4.3: *The probability density function of the Weibull distribution.*



In Section 4.4.1 we discuss the application of these distributions to the problem of modelling the costs of local optima of the TSP.

The Uniform Distribution

A random variable X has a *uniform distribution* on $[a, b]$ if its probability density function $f(x)$ is *uniform distribution*

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases} \tag{4.19}$$

Poisson Distribution and Process

Let X be a discrete random variable, then X has a *Poisson distribution* with *intensity* λ if its probability mass $\mathrm{p}(x)$ is given by

Poisson distribution
intensity

$$\mathrm{p}(x) = \frac{e^{-\lambda}\lambda^m}{m!}, m = 0, 1, 2 \ldots \tag{4.20}$$

The expected value of this distribution is λ.

A *spatial Poisson process* or *Poisson point process* models a set of points "scattered" in some compact space. More specifically, and in 2 dimensions, let S be a measurable subset of $\mathbb{R}^2$ with area $A(S)$. A set of points $\Pi(S)$ is a spatial Poisson process in $\mathbb{R}^2$, if the number of points in S, $|\Pi(S)|$ is a Poisson random variable with intensity, $\lambda = A(S)$ and, in addition, for any other measurable subset of $\mathbb{R}^2$, T, with S and T disjoint, then $|\Pi(S)|$ and $|\Pi(T)|$ are independent.

spatial Poisson process
Poisson point process

The usefulness of this arrangement to our present purpose is, that under a spatial Poisson process on S, the points in $\Pi(S)$ are uniformly distributed. A proof is of this provided in Karlin and Taylor [65]. The fact that the number of points in disjoint regions are independent, makes the Poisson process more useful than a simple uniform distribution, since it allows a more straight forward analysis of partitioning algorithms.

Example TSP Distribution

Figure 4.4 shows a small geometric problem taken from the TSPLIB data set, along with its probability mass function. The instance, burma14.tsp, consists of 14 locations in Burma. The problem embeds in a sphere of large diameter (the earth) and so has only an approximate embedding in the Euclidean plane. The figure shows a low cost tour (3371 km). The solution space of the instance consists of 113510400 tours ranging in cost from 3323 to 9139 km. We constructed the distribution plots by complete enumeration of this space[2]. Figure 4.5 shows the cumulative distribution function of the problem. The instance has mean tour cost of 6672.15 km, a standard deviation of 709.376 km, a skewness of -0.0632331, and a kurtosis of 2.79716. These statistics, together with the plots, suggest that the cost distribution of the instance is approximately normal. Figure 4.5 shows the cumulative distribution function of the instance and Figure 4.6 shows the probability mass of low cost tours (less than cost 3500).

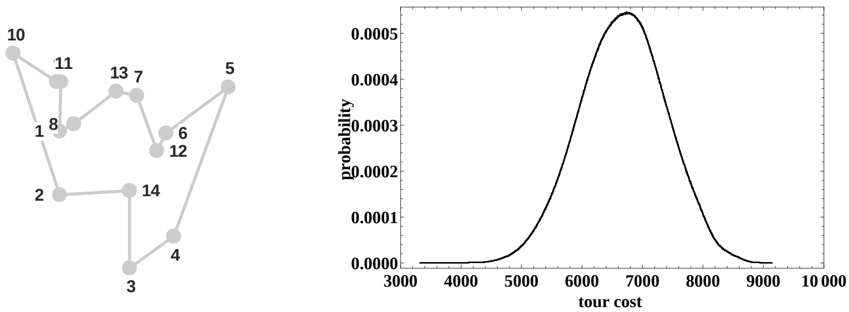


Figure 4.4: *(left) A low cost tour in a 14 city TSP. The problem has an approximate embedding in the plane. (right) The probability mass of the problem. The appearance of continuity in this plot is due to the large number solutions.*

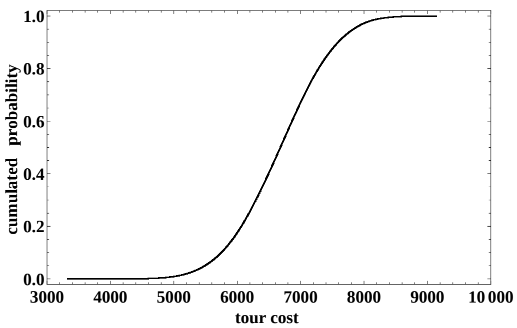


Figure 4.5: *The cumulative distribution function of the problem shown in Figure 4.4. The plot is valid in [3323,9139].*

[2]An instance size of 14 cities is close to the limit for complete enumeration of the solution space in typical contemporary computing environments.

Figure 4.6: *The probability mass of tours with cost less than 3500, for the problem shown in Figure 4.4.*

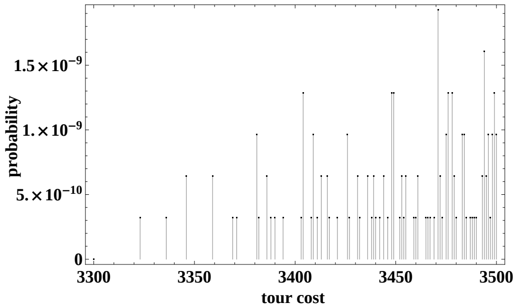
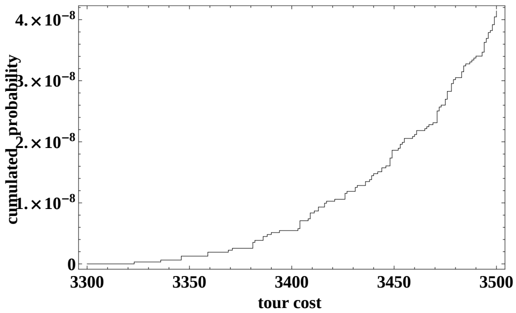


Figure 4.7: *The cumulative distribution function of tours with cost less than 3500, for the problem shown in Figure 4.4.*



4.2 Probabilistic results on the TSP

Previous theoretical work on the probability distribution of the TSP is surveyed in [37, 114, 124]. Much of this work concerns the stochastic d dimensional Euclidean case with *city coordinates* as independent and identically distributed random variables. Many results relate to the key finding of Beardwood, Halton and Hammersley [10] who showed that, where the number of cities, n, is large the cost of the optimal tour is close to the product of $n^{(d-1)/d}$ and a certain constant, c_d. More recently Wästlund [119] has shown a similar result for the random edge cost problem.

4.3 The Stochastic TSP

As with the TSP with fixed edge costs, we define the *stochastic TSP* in terms of a complete *undirected* graph $G = (V, E)$ with the vertices V representing cities, and the edges E representing the connections between cities. A tour, π and the solution space, Θ are defined identically to the fixed cost case.

In the stochastic problem, the cost of each edge, $e \in E$ is a random variable, $\text{Cost} : E \rightarrow \mathbb{R}$ or $C : E \rightarrow \mathbb{R}$ from some distribution. Unless stated otherwise, we do not assume the edge distributions are identical or independent. The cost of a tour is then a random variable $\text{Cost} : E \rightarrow \mathbb{R}$. As before, we may abbreviate this function to C. Also as before, where we wish to emphasise that the domain of the cost function is Θ we write Cost_Θ (or C_Θ) and similarly for restrictions to some subset of Θ. Cost_Θ C_Θ

The stochastic problem not only has theoretical interest, as discussed below, but is also frequently implicit in practical situations, where the edge costs of a problem are not known with any certainty. Indeed, this lack of certainty often justifies the use of approximation algorithms that sacrifice solution quality for execution speed.

It is worth noting that the goal of the fixed cost TSP (to find a tour of minimum cost) may make little sense in a stochastic problem since there are two sources of variation in tour cost. Firstly, the choice of the tour and secondly the random nature of each edge cost. For example, in the case of the trivial TSP, one with just three edges and one solution, if the edge costs are fixed, the variance of tour costs is, of course, 0. In the stochastic case, if the edge costs are independently distributed, the variance of the tour cost over the solution space is the sum of the three edge cost variances [105].

It is occasionally useful to view the fixed cost TSP as a special case of the stochastic TSP. We will apply this approach in Chapter 6.

The *stochastic Euclidean TSP* is defined analogously to the fixed cost Euclidean TSP case, but with *city coordinates* as random variables. The edge costs of the problem are then random variables, since they are a function of random variables. However, properties such as independence between random variables are not preserved by this arrangement. So, where the city coordinates are independent random variables, this property does not extend to the edge costs. *stochastic Euclidean TSP*

4.3.1 Probability Distribution of the Euclidean TSP

Here we present the finding of Beardwood, Halton and Hammersley [10] and discuss its implications for the Euclidean problem.

Theorem 10 (Beardwood, Halton and Hammersley) *Let the vertex coordinates of an n city TSP be independently and identically distributed random variables in $\mathbb{R}^d$ with distribution $\mathcal{D}$ having a compact support and let the random variable C^* denote the cost of an optimal solution. Then with probability 1*

$$\lim_{n\to\infty} \frac{C^*}{n^{(d-1)/d}} = c_d \int_{\mathbb{R}^d} f(x)^{(d-1)/d} \mathrm{d}x \tag{4.21}$$

where $f(x)$ is the probability density of the absolute continuous part of $\mathcal{D}$ and c_d a positive constant dependent only on d, the dimension of the space.

Yukich [124] points out that, as a consequence of Jensen's inequality, a uniform distribution *maximises* the integral on the right hand side of this expression. This indicates that non-uniformity in $\mathcal{D}$ serves to *decrease* the length of optimal tours. This observation allows us to view uniformly distributed problems as representing a "worst case". For these problems Theorem 10 becomes:

Theorem 11 (Beardwood, Halton and Hammersley uniform) *Let the vertex coordinates of an n city TSP be independently uniformly distributed in $[0,1]^d$ and let the random variable C^* denote the cost of an optimal solution. Then with probability 1*

$$\lim_{n\to\infty} \frac{C^*}{n^{(d-1)/d}} = c_d \tag{4.22}$$

where c_d is a positive constant dependent only on d, the dimension of the space.

In relating Theorem 11 to Theorem 10 it is worthwhile to note that for a continuous distribution such as the uniform distribution, $1 = \int_{-\infty}^{+\infty} f(x)\mathrm{d}x$.

In terms of the actual value of the c_d, Rhee [100] proves

$$\lim_{d\to\infty} \frac{c_d}{\sqrt{d}} = \frac{1}{\sqrt{2\pi e}} \tag{4.23}$$

and empirical evidence suggests $0.70 \leq c_2 \leq 0.73$ [114].

In other work characterising the probability distribution of the cost of the optimal tour, C^*, Rhee and Talagrand [101] prove that this distribution has Gaussian tail bounds.

In the next two sections we give an outline of the proof of a simplification of Theorem 11 in the limited case of $d = 2$. Our sketch outlines the full proof provided in Karp and Steele [67]. The next two algorithms, in addition to aiding the outline, closely relate to the polynomial time approximation algorithms for the Euclidean TSP of Karp [66] and Arora [3, 4].

The Strips Algorithm

This algorithm demonstrates the utility of a coordinate system in a metric space. It works by partitioning the plane into "strips" and constructing a tour by connecting cities in these strips. Its operation is illustrated in Figures 4.8 to 4.10. Lemmas 12 and 13 characterise its properties.

Algorithm 20 The Strips Algorithm
Pseudo code for the strips algorithm

Input: The coordinates of an n city 2 dimensional Euclidean TSP in $[0, l]^2$
Input: The number of strips s.
Output: A tour π with $\text{cost}(\pi) \leq l(\frac{n}{s} + s + 2 + \sqrt{2})$

```
partition the plane, [0,l]^2 into s strips.
draw a vertical line through each city bounded by the strip it is contained in
start from the top left of the top strip (strip 1)
repeat
    on odd numbered strips draw horizontally to the right and on even strips draw
    horizontally to the left
    if a vertical line is encountered then
        move along the vertical line appending each city on the line to the tour
    end if
    at the end of each horizontal pass if on the top of the strip draw vertically downward
    one strip.
until completed a horizontal pass of the bottom strip
draw a final line to top left of the first strip
return the tour as return π
```

Lemma 12 *If c is the cost of a tour provided by the strips algorithm (Algorithm 20) on $[0, l]^2$,then $c \leq l(\frac{n}{s} + s + 2 + \sqrt{2})$.*

Proof: By the triangular inequality, the cost of the tour produced is no more than the sum of the distances drawn by the algorithm. There are no more than n vertical lines through each city each of length $\frac{l}{s}$. The vertical distance drawn at the end of each strip is no more than $2l$. The horizontal distance drawn is sl and the final closing line is of length no more than $l\sqrt{2}$.□

Lemma 13 *The expression of Lemma 12 is minimised if* $s = \sqrt{n}$ *and under this condition we have the cost* $c \leq l(2\sqrt{n} + 2 + \sqrt{2})$.

Proof: Routine. □

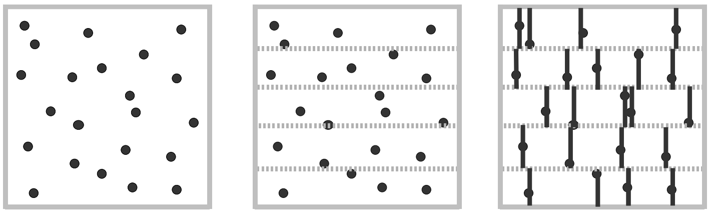


Figure 4.8: *The operation of the strips algorithm on a 22 city TSP. The region is partitioned into horizontal strips. A vertical line is made though each city.*

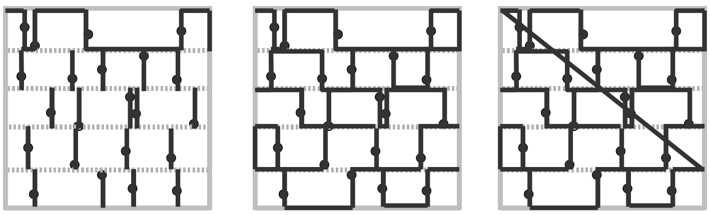


Figure 4.9: *We move horizontally until a vertical line is encountered in the current strip. Each alternate strip is traversed in the opposite direction.*

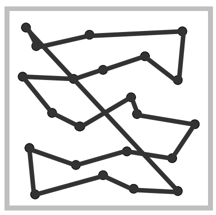


Figure 4.10: *The final tour produced by the strips algorithm. The tour is closed, by a diagonal if necessary. We apply the shortcut lemma, or any other reduction method, as required to produce this. The point of the whole process is that the final tour will be of length no greater than the horizontal, vertical and final diagonal moves.*

The Patches Algorithm

The patches algorithm applies the strips algorithm to construct a tour. The method works by partitioning the space $[0, l]^2$ into squares (patches) such that the optimal tour contained in each small square can be computed. If each small square contains sufficiently *few* cities then the optimal tours (in each small square) can be found in polynomial time on the number of cities in the large square, $[0, l]^2$. The operation of the algorithm is shown in Figure 4.11 and Lemma 14 gives the key result.

Algorithm 21 The Patches Algorithm
Pseudo code for the patches algorithm

Input: The coordinates of an n city 2 dimensional Euclidean TSP in $[0, l]^2$
Input: The number of patches, p, a power of 2.
Output: A tour π

1: partition the plane, $[0, l]^2$ into p patches
2: compute an optimal tour in each patch finding a set P of p tours
3: **choose** one city in each patch
4: apply the strips algorithm to chosen city in each patch to provide a tour π
5: insert the p tours of P into π applying the shortcut lemma as required
6: return the final tour as **return** π

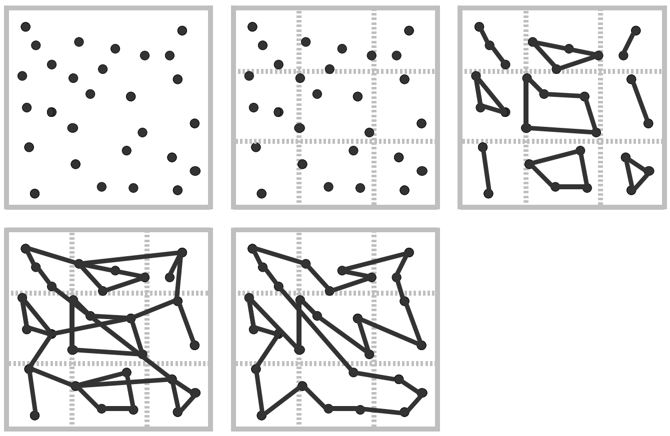


Figure 4.11: *The patch algorithm. The region is partitioned into smaller squares. The strip algorithm is applied to each small square. One large tour is then constructed from each small tour. The shortcut lemma is applied to produce the final tour.*

Lemma 14 *Let p be the number of patches input to Algorithm 21, and let* $\mathrm{cost}(P_i)$ *be the cost of a tour in patch i of the algorithm. Let c be the cost of the tour returned by the algorithm, then* $c \leq \sum_{i=1}^{p} \mathrm{cost}(P_i) + l(2\sqrt{n} + 2 + \sqrt{2})$.

Proof: By Lemma 12 and the shortcut lemma, Lemma 6. $\square$

The Simplified Proof for a Poisson Distribution

We recall from Section 4.1.6 that under a spatial Poisson process, $\Pi(S)$, on a region, S, the points in $\Pi(S)$ are uniformly distributed and, in addition, the number of points in any two subsets of S are independent. Thus a spatial Poisson process can be viewed as providing an "ultra uniform distribution". In a second simplification we consider only the expected value of the optimal tour cost.

Theorem 15 *Let the n cities of a 2 dimensional TSP have a spatial Poisson distribution with intensity 1. Then there is a constant c_2 such that $\lim_{n\to\infty} \frac{\mathrm{Ex}(C^*)}{\sqrt{n}} = c_2$, where the random variable C^* is the cost of the optimal tour.*

Proof:

Let Π be a spatial Poisson process in $\mathbb{R}^2$ with intensity 1. The number of cities in $\Pi([0,l]^2)$ is a random variable with expected value l^2. By Lemma 14,

$$\mathrm{Ex}(\mathrm{Cost}^*(\Pi([0,l]^2)) \leq \sum_{i=1}^{p^2} \mathrm{Cost}(P_i) + l(2p + 2 + \sqrt{2}).$$

Now each of the p^2 smaller square patches P_i has height $s = \frac{l}{p}$, and each has a Poisson distribution, so we can write $\mathrm{Ex}(\mathrm{Cost}^*(P_i)) = \mathrm{Ex}(\mathrm{Cost}^*\Pi([0,s]^2)$ and therefore

$$\mathrm{Ex}(\mathrm{Cost}^*(\Pi([0,l]^2)) \leq p^2\mathrm{Ex}(\mathrm{Cost}^*\Pi([0,s]^2) + l(2p + 2 + \sqrt{2}).$$

Now $l = sp$, so

$$\mathrm{Ex}(\mathrm{Cost}^*(\Pi([0,sp]^2)) \leq p^2\mathrm{Ex}(\mathrm{Cost}^*\Pi([0,s]^2) + 2sp^2 + sp(2 + \sqrt{2}).$$

and dividing by $l^2 = (sp)^2$ gives

$$\frac{\mathrm{Ex}(\mathrm{Cost}^*(\Pi([0,sp]^2))}{s^2p^2} \leq \frac{\mathrm{Ex}(\mathrm{Cost}^*\Pi([0,s]^2)}{s^2} + \frac{2}{s} + \frac{(2+\sqrt{2})}{sp}, p = 1, 2 \ldots \tag{4.24}$$

In this bound, we note that the recursion is in our favour since $sp \geq s$. Karp and Steele [67] and Steele [114] argue that $\mathrm{Ex}(\mathrm{Cost}^*\Pi([0,l]^2)$ is monotone (and so continuous) and by Lemma 13, $\frac{1}{l^2}\mathrm{Ex}(\mathrm{Cost}^*(\Pi([0,l]^2))$ is bounded. Therefore, in view of the recursion above, we have

$$\lim_{l\to\infty} \frac{1}{l^2}\mathrm{Ex}(\mathrm{Cost}^*(\Pi([0,l]^2)) = c_2 \tag{4.25}$$

for some constant c_2. Recalling that the expected number of cities, n, in $\Pi([0,l]^2)$ is l^2, implies the result. □

Karp and Steele [67] and Steele [114] provide stronger results by consideration of $\mathrm{Var}(C^*)$ rather than simply $\mathrm{Ex}(C^*)$ as we do here.

4.3.2 Random Edge Cost Results

Given a graph, $G(V,E)$ a 2-factor is a set of edges $E' \subseteq E$ such that each vertex, V, is incident to precisely two edges of E'. The edges of E' therefore form either, a single Hamiltonian cycle or a set of disjoint cycles. A minimum cost 2-factor is a 2-factor with minimum cost over the set of all 2-factors in a given weighted graph. It is well known that the minimum cost 2-factor of an n vertex complete weighted graph can be computed in $O(n^3)$ [36].

Given these properties, the problem of finding a minimum cost 2-factor in a complete weighted graph is a useful relaxation of the TSP. Frieze [36] introduces the above arrangement as a relaxation of the TSP and exploits it to provide the following remarkable result.

Theorem 16 *Let the edge costs of an n city TSP be uniformly distributed in $[0,1]$. Let the random variable C_n^* denote the cost of an optimal solution of a TSP and let F_n^* be the minimum cost of a 2-factor on the set of edges of the TSP. Then, with high probability, as $n \to \infty$, $|C_n^* - F_n^*| \to 0$.*

Furthermore, Frieze provides a fast algorithm to construct a low cost tour from a minimum cost 2-factor. This allows a low cost tour to be computed in $O(n^3)$ for an n city TSP.

More recently, as yet unpublished, results of Wästlund [119] give

Theorem 17 *Let the edge costs of an n city TSP be uniformly distributed in $[0, 1]$. Let the random variable C_n^* denote the cost of an optimal solution. Then as $n \to \infty$, C_n^* convergence in probability and expectation to a constant c_u given by*

$$c_u = \frac{1}{2}\int_0^\infty y(x)\mathrm{d}x, \tag{4.26}$$

where y is the positive solution to the equation

$$\left(1 + \frac{x}{2}\right) e^{-x} + \left(1 + \frac{y}{2}\right) e^{-y} = 1. \tag{4.27}$$

Numerical techniques give c_u as approximately 2.0415. This work makes rigorous the findings of Mézard and Parisi [86] and Krauth and Mézard [73], who apply the cavity method of statistical mechanics to the TSP.

In more direct significance to the present thesis is the following open problem (problem 44) of Frieze and Yukich [37]. Let $U_i, \ldots, U_n$ be identically distributed random variables in $[0, 1]^d$. Establish that the scaled variance of the TSP functional converges. In dimension $d = 2$, this means showing that $\mathrm{Var}(U_i, \ldots, U_n) \to c$, where c is some positive constant.

4.4 Sampling Results and the TSP

4.4.1 Distribution of Optimal Cost Tour

Reiter and Rice [99] study the cost distribution of local optima, under a gradient maximizing search, in 39 integer programming problems. Their results suggest that the local optima follow a Beta distribution. McRoberts [84] considers plant layout problems and argues that computing the exact solution of combinatorial problems is not only intractable but frequently pointless due to doubt in the instance data. He applies a similar approach to Reiter and

Rice, sampling the costs of local optimal solutions produced by the heuristic of Hillier [52]. This sample data is then used to fit the Weibull distribution which in turn provides an estimate of the global optimum cost. Golden [43] examines six problems from the TSPLIB archive [96]. Distributions of the global optimal tour costs are simulated by generating low cost tours using the 2-opt and 3-opt heuristics. The resulting estimates of optimal solutions are compared to the best solution found by the Lin-Kernighan algorithm. The authors conclude that the Beta distribution provides “a more appropriate distribution” than the Weibull distribution. Indeed they find that the average deviation from the best solution found by the Lin-Kernighan algorithm is less than 1% and the worst is 1.3%.

More recently Vig and Palekar [118], apply sampling techniques similar to Golden, but use the Lin-Kernighan algorithm to find low cost tours. The authors *estimate* moments one to four of the probability distribution of the costs of these tours. They use these estimates to fit various candidate distributions including the Beta, Weibull and Normal cases. Vig and Palekar find that the Beta distribution yields the closest fit.

4.4.2 Relationship Between the Sample Variance of Tour Costs and the Optimal Tour Cost

Basel and Willemain [9] investigate 17 problems from the TSPLIB data set, each with an approximate embedding in the Euclidean plane. They *estimated* both the mean and standard deviation of tour costs by *random sampling* and find a remarkable linear correlation between the square root of an instance size, the number of standard deviations between the mean tour length, and the known optimal tour length. Specifically, they find that if μ_{sample} and σ_{sample} are the sample mean and standard deviation and c^* is the known optimum then $\frac{c^* - \mu_{sample}}{\sigma_{sample}}$ is approximated by $3.289 - 2.309\sqrt{n}$ where n is the number of cities in an instance.

Basel and Willemain adopt a simple sampling approach, generating 20,000 random tours for each instance. The time complexity of their *estimate* of the variance when applied to the non-Euclidean TSP is therefore $O(n^2)$. Our results of Chapter 7 provide an $O(n^2)$ method to *compute* the variance of tour costs. We return to the empirical findings of Basel and Willemain in Sections 7.3 and 11.2.

4.5 Statistical Properties of Landscapes

Here we consider the properties of a general class of optimisation problems, of which the TSP is a notable member.

4.5.1 Elementary Landscapes

Let $\mathfrak{L} = (S, f, \mathfrak{N})$ be the landscape of a combinatorial optimisation problem and let $\mu = \frac{1}{|S|}\sum_{s\in S} f(s)$. So μ is the expected value of the cost of a solution in S.

Grover [46] defines a difference operator ∇^2 as ∇^2 ∇^2

$$\nabla^2 f(s) = \frac{\sum_{s'\in\mathfrak{N}(s)}[f(s') - f(s)]}{|\mathfrak{N}(s)|}. \tag{4.28}$$

This is simply the expected value of the *difference* between $f(s)$ and its $|\mathfrak{N}(s)|$ neighbours. Grover shows that for the TSP, min-cut, graph partitioning, graph colouring, weight partition and the not-all-equal-satisfiability problems, there are move operations that give rise to neighbours such that

$$\nabla^2 \mathbf{f}_\mu + \lambda \mathbf{f}_\mu = 0 \tag{4.29}$$

with $\lambda > 0$ and the vector of normalised costs $\boldsymbol{f}_\mu$ given by $\mathbf{f}_\mu(s) = f(s) - \mu$. $\boldsymbol{f}_\mu$
Grover relates the operator ∇^2 and Equation 4.29 to the wave equations of physics and proves the following theorem.

Theorem 18 (Grover) *Let $\mathfrak{L} = (S, f, \mathfrak{N})$ be the landscape of a COP and let μ be the expected value of solution costs of S. If Equation 4.29 is satisfied then for all local minima $s \in S$ and all local maxima $x \in S$ we have $f(s) \leq \mu$ and $f(x) \geq \mu$.*

Proof: It follows from the definition of a local minimum and the properties of the expected value that $f_\mu(s) \leq \nabla^2 f_\mu(s)$. So in order for Grover's Equation to be satisfied we must have $f_\mu(s) \leq 0$. Similarly for a local maximum. □

Codenotti and Margara [25] state that, among others, the 2-opt move gives rise to landscapes that satisfy Grover's equation. Punnen et al. [93] provide proofs that local minima under both 2-opt and 3-opt have costs less than μ. These authors report that the original proof of the 2-opt result was provided by Rublineckii [102].

Let $\mathbf{A}$ be the adjacency matrix of a neighbourhood $\mathfrak{N}$ so that for $s, s' \in X$, **A** $\mathbf{A}[s, s']$ is the number of edges in $\mathfrak{L}$ from s to s'. A landscape is regular when each vertex in $\mathfrak{L}$ has the same out degree. It is symmetric when $\mathbf{A}$ has that property. For regular and symmetric landscapes Stadler [113] defines a graph *Laplacian operator* $\mathbf{L_s} = d\mathbf{I} - \mathbf{A}$ where the scalar d is the out degree of each *Laplacian operator* vertex in $\mathfrak{L}$ and $\mathbf{I}$ is the identity matrix of size commensurate with $\mathbf{A}$. Stadler denotes landscapes where $\mathbf{f}_\mu$ is an eigenvector of $\mathbf{L_S}$ as *elementary landscapes*, *elementary landscapes* and in these landscapes, $\mathbf{f}_\mu$ satisfies Equation 4.29.

Barnes et al. [7] consider a more general class of landscapes which need not be regular or symmetric. Let the diagonal matrix $\mathbf{D}$ be the degree matrix of $\mathfrak{L}$ so that $\mathbf{D}[s, s]$ is the out degree of a vertex s in the landscape. Then Barnes et al. define a graph Laplacian

$$\mathbf{L_B} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} \tag{4.30}$$

Under this arrangement, a landscape is elementary, if for some α, not necessarily equal to μ, $\mathbf{f}_\alpha(s) = f(s) - \alpha$ is an eigenvector of $\mathbf{L_B}$. If this is the case, Grover's equation (in a more general form) is satisfied, that is to say

$$\nabla^2 \mathbf{f}_\alpha + \lambda \mathbf{f}_\alpha = 0 \tag{4.31}$$

with $\mathbf{f}_\alpha(s) = f(s) - \alpha$.

For regular and symmetric elementary landscapes, Equation 4.31 is satisfied if $\alpha = \mu = \frac{1}{|S|}\sum_{s \in S} f(s)$. In this, more general case, the authors prove two situations arise:

smooth-elementary if $0 < \lambda \leq 1$.

- where $f_\alpha(s) \leq 0$ then $f(s) \leq \nabla^2 f(s) \leq \alpha$
- where $f_\alpha(s) \geq 0$ then $f(s) \geq \nabla^2 f(s) \geq \alpha$

rugged-elementary if $1 \leq \lambda < 2$.

- where $f_\alpha(s) \leq 0$ then $f(s) \leq \alpha \leq \nabla^2 f(s)$
- where $f_\alpha(s) \geq 0$ then $f(s) \geq \alpha \geq \nabla^2 f(s)$

So in *smooth-elementary* landscapes, the neighbours of a solution s, with cost $f(s)$, have a cost on average similar to $f(s)$ and both of these are on the same side of α. In contrast, in *rugged-elementary* landscapes, the cost of s and the cost of its average neighbour are separated by α. More recently Dimova et al. [33] extend this work by considering composite landscapes. These are constructed as sequences of moves on some specified landscape.

smooth-elementary

rugged-elementary

4.5.2 Random Walks on Landscapes

Let $(\Omega, \mathcal{A}_\Omega, \mathrm{P}_\Omega)$ be a finite probability space and let $T \subseteq \mathbb{R}$ be a discrete set. A *random sequence* or *stochastic process with discrete time* or *discrete random process* is a set of random variables, $\mathbf{X} = X_{t \in T}$ indexed by T (typically time) A *realization* or *trajectory* of a random sequence is a set of real numbers $\mathbf{x} = (X_t(\omega_t) = x_t)$ indexed by $t \in T$. Conceptually, a realization of a random process is simply a vector of values that are the outcome of some experiment or set of observations carried out over time. Let $\mathbf{x}$ be a realization of a random process. The process is an *autoregressive process* of order 1 (*AR(1)*) if $\mathbf{x}$ obeys the recurrence equation

random sequence

discrete random process

realization

trajectory

autoregressive process

AR(1)

$$x_t = \phi_1 x_{t-1} + R_t \tag{4.32}$$

with ϕ_1 a constant and each R_t a normal random variable with expected value 0.

Stadler [113] in extending the work of Weinberger [120] shows that a random walk on a symmetric landscape is an AR(1) process if, and only if, the landscape is elementary.

4.6 The Empirical Result of Boese et al.

Kirkpatrick and Toulouse [71], among others, note that local minima in the 2-opt landscape of the TSP tend to have many edges in common. This is to say, the bond distance between the minima is low. Figure 4.12 illustrates that this effect is not restricted to the low cost edges of an instance. Boese et al. in investigating this phenomenon demonstrates correlations between both, the cost of a local minimum and the mean bond distance between the other local minima, and the cost of local minima and the bond distance to a global minimum solution [13, 14]. These correlations are evident in both the 2-opt and 3-opt landscapes of the TSP. The authors apply this result to an optimisation algorithm they term the new adaptive multistart algorithm.

The scatter plots of figures 4.13 and 4.14 show a replication[3] of the result of Boese et al. Here the cost of 2500 2-opt local minima in a 512 city problem are shown. These were produced by randomly generating tours and descending by iterative improvement using the first improving move in a randomly ordered 2-opt neighbourhood. In the case of the relationship between the cost of local minima and the bond distance between the other local minima, the Spearman's non-parametric correlation coefficient is 0.74 (< 0.01). The Spearman's non-parametric correlation coefficient is 0.54 (< 0.01) between the cost of local minima and the bond distance to the global minimum. These are comparable with Boese's result. Both correlations are clear, in the sense of being statistically significant, but in the second result the correlation is rather weak.

[3]This replication formed part of the author's honours thesis

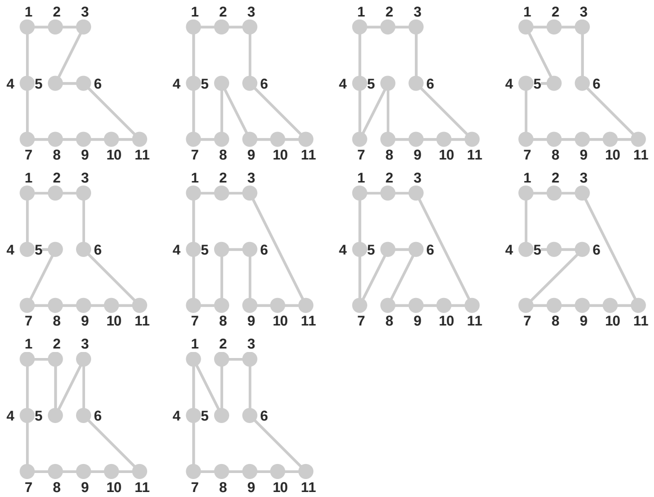


Figure 4.12: *All ten 2-opt local minima of an eleven city Euclidean TSP. The total number of tours in the landscape is 1814400. It is noteworthy that the comparatively long edge,* $\{3, 11\}$, *occurs in 3 of the local minima.*

Figure 4.13: *The cost of local minima versus the mean bond distance to all other local minima in att532. Spearman's non-parametric correlation coefficient is 0.74* (< 0.01).

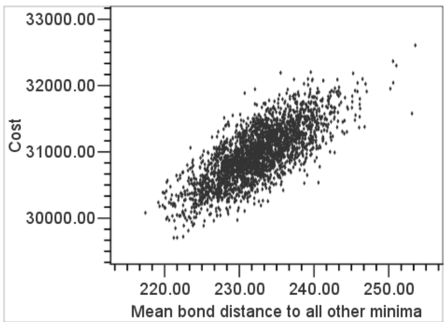
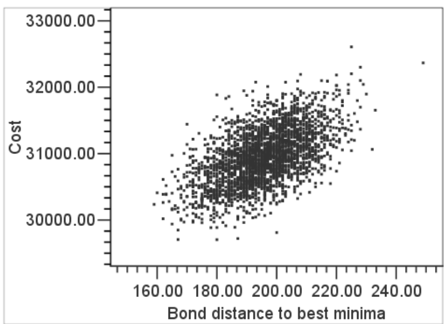


Figure 4.14: *The cost of local minima versus the bond distance to the global minimum for att532. Spearman's non-parametric correlation coefficient is 0.54* (< 0.01).



4.7 Conclusions

In this chapter, we have reviewed the statistical theory underpinning the results appearing in this document. We have also briefly reviewed probabilistic and statistical literature pertaining to the TSP.

In the case of the 2-opt and related elementary landscapes on the general TSP, the empirical results suggest that low cost local minima tend to be clustered together. The key theoretical result on these landscapes is that the local minima have values below that of the expected values of all tour costs. In addition, in these cases, the landscape is smooth with a small change in position from a local minimum resulting in a small change (on average) in solution cost. The fact that more complex landscapes, that are non-regular and non-symmetric, can have similar properties, holds out intriguing possibilities of designing and adapting landscapes to specific instances of problems and to dynamic, run-time, modification of neighbourhoods. It is noteworthy that some of these possibilities are now being realised in practical search algorithms [23]. Our results of Chapter 10 are relevant to this approach. In particular, Theorem 57 provides us with a method to compute the variance of gains over the 2-opt landscape of an n city TSP in $\mathcal{O}(n^2)$. Among other applications for this information, it is usefulness to heuristics based on the 2-opt move that attempt to assess the run-time quality of search paths.

Chapter 5

A Perfect TSP

The results of Chapter 4 indicate that the probability distributions of tour costs of Euclidean TSPs are likely to be Gaussian. The purpose of this chapter is to illustrate that the TSP can give rise to more varied, and interesting cost distributions. In particular our arrangement allows for creation of cost distribution which are markedly platykurtic. We view this as useful for testing algorithms. As we will see, the construction we use is of interest in its own right.

5.1 Definition of a Perfect TSP

Let $B \subseteq \mathbb{R}$ be the range of the cost function of a TSP with solution space Θ. We call a TSP *perfect* if $|\Theta| = |B|$, this is to say, for a perfect TSP, each solution has a distinct cost. In this chapter we show the existence of a class of perfect TSPs, with the following useful properties:

perfect TSP

- The problems have a polynomial size representation in relation to the number of vertices in an instance.
- The Karp reduction to the perfect TSP is from an $\mathcal{NP}$-complete problem, the Hamiltonian cycle problem.
- Given a rational number, c represented in a reasonable form, it can be decided in polynomial time if a tour of cost c exists.

- If a tour of cost, c, exists it is possible in polynomial time to compute it.

It is worthwhile pointing out that the relationship between the edge costs and the tour cost tour costs of the TSP is non-bijective. So that a given TSP, perfect or otherwise, can be produced by many different edge matrices. This is important in understanding the properties of elementary landscapes [8].

5.1.1 An Algorithm to Construct a Perfect TSP

Here we present our reduction algorithm (Algorithm 22), from an instance of a Hamiltonian cycle problem to a TSP. In Theorem 19 we prove that the reduction provides perfect TSPs. Figure 5.1 illustrates the method on a small Hamiltonian cycle problem instance. Theorem 21 shows that it is possible to represent any instance of the TSP constructed by our algorithm in polynomial space on the instance size.

Algorithm 22 Create a Perfect TSP
Pseudo code for a Karp reduction from a Hamiltonian cycle problem to a perfect TSP.

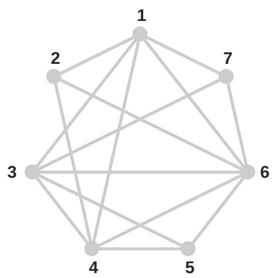
Input: An instance $H = (V_h, E_h)$ Hamiltonian cycle problem
Input: An integer, o, the offset
Input: A positive integer, g, the gap
Output: A perfect TSP T with edge cost function cost
$s \leftarrow o$
$l \leftarrow |E_h| + g + o$
construct a complete graph $T = (V_h, E)$
while unvisited edges e in E **do**
    if $e \in E_h$ **then**
        $\text{cost}(e) \leftarrow 2^s$
        $s \leftarrow s + 1$
    else
        $\text{cost}(e) \leftarrow 2^l$
        $l \leftarrow l + 1$
    end if
end while
return T as a TSP.

The algorithm transforms an instance of the Hamiltonian cycle problem to a TSP, such that, where no edge connects two vertices in the Hamiltonian cycle problem, an edge of cost less than $2^{(|E_h|+o)}$ is added to the TSP. Where an edge exists between two vertices in the Hamiltonian cycle problem, an edge of cost greater than or equal to $2^{(|E_h|+g+o)}$ is added to the TSP. The parameter, g, the gap, therefore determines the difference in cost between the two classes of edges. The parameter, o, the offset, determines the lowest edge cost. Manipulation of the parameters o and g enable the edge costs of the resulting TSP to be bounded.

So for example, given a Hamiltonian cycle problem with n vertices, $|E_h|$ is at most $\frac{1}{2}n(n-1)$. Setting $g = 0$ and $o = -\frac{1}{2}n(n-1)$, will result in the lowest edge cost of $2^{(-\frac{1}{2}n(n-1))}$ and the largest edge cost of $2^0 = 1$.

Theorem 19 *A TSP constructed by Algorithm 22 is perfect.*

Proof: Let Θ be the solution space of an n city TSP (V, E), constructed by Algorithm 22. Let o be the offset parameter and g be the gap parameter of the algorithm. The cost of each edge of such an instance can be written uniquely as 2^{k_i} for some *distinct* integer k_i, $o \leq k_i \leq |E_h| + g + o$. The cost of a tour is then $2^{k_1} + 2^{k_2} + \ldots + 2^{k_n}$, but no two tours in Θ contain the same set of n edges. So by the elementary properties of numbers the cost of each tour in Θ is distinct. $\square$



$$\begin{pmatrix} - & & & & & & \\ 2^0 & - & & & & & \\ 2^1 & 2^{26} & - & & & & \\ 2^2 & 2^5 & 2^7 & - & & & \\ 2^{25} & 2^{27} & 2^8 & 2^{11} & - & & \\ 2^3 & 2^6 & 2^9 & 2^{12} & 2^{13} & - & \\ 2^4 & 2^{28} & 2^{10} & 2^{29} & 2^{30} & 2^{14} & - \end{pmatrix}$$

Figure 5.1: *An example of a 7 vertex Hamiltonian cycle problem with a perfect TSP and the cost matrix computed by Algorithm 22. In this example the parameter o is 0 and g is 10.*

Theorem 20 *Let $H = (V_h, E_h)$ be a Hamiltonian cycle problem. Let T be the n city TSP constructed by Algorithm 22 with input H, o, g. Any tour π in T corresponds to a Hamiltonian cycle in H if, and only if,* $\mathrm{cost}(\pi) < 2^{|E_h|+g+o}$.

Proof: The result follows from the construction of the algorithm and the fact that for any integer k, $2^k > 2^{k-1} + 2^{k-2} + \ldots + 2^{k-n}$. $\square$

The conventional representation of integer edge costs would certainly not produce a polynomial space bound on a TSP constructed with Algorithm 22, since the edge costs are exponential on $|E|$. However a polynomial size representation is possible, but at the cost of limiting the operations available (in polynomial time) to an optimisation algorithm. The scheme of Theorem 21, below, works by simply storing, only the *exponents* of each edge cost, noting that the *mantissa* of the edge cost is always 2.

In the next theorem we view the offset and gap parameters as functions of the size of an instance, n. By *space complexity* we mean the number of bits required to store an instance of a problem under a reasonable encoding scheme. *space complexity*

Theorem 21 *An n city TSP constructed by Algorithm 22 has polynomial space complexity on n if both $|o|$ and g are polynomial bounded on n.*

Proof: Each edge cost is of the form 2^{k_i} with $0 \leq k_i \leq g + o$ and $|E| = \frac{1}{2}n(n-1)$ therefore, we need store only each k_i. The number of bits required to do this is polynomial bounded on n. $\square$

Corollary 22 *The cost of a tour in an n city TSP constructed by Algorithm 22 has polynomial space complexity on n if both $|o|$ and g are polynomial bounded on n.*

Proof: By the argument of the theorem. $\square$

5.1.2 Computability of the Perfect TSP

We recall from Section 2.1 that the decision problem of the TSP is: given a TSP, is there a tour with cost less than or equal to k? Given the above reduction it is ease to show that the decision problem associated with the perfect TSP is $\mathcal{NP}$-complete.

Theorem 23 *The decision problem associated with the perfect TSP is $\mathcal{NP}$-complete.*

Proof: A polynomial time decision algorithm for the perfect TSP could be used to solve the Hamiltonian cycle problem by use of Algorithm 22 and Theorem 25. $\square$

5.2 Tour Operations

Here we consider the operations that can be conducted, in polynomial time, on a TSP with representation as in Theorem 21. First we note that any tour in an n city TSP provided by Algorithm 22 can be represented as its n edge costs, and so can be stored in polynomial space as shown in Theorem 22.

Theorem 24 *Let π and τ be two tours in a TSP constructed using Algorithm 22 with the representation of Theorem 22. Then the relations,* $\mathrm{cost}(\pi) < \mathrm{cost}(\tau)$, $\mathrm{cost}(\pi) = \mathrm{cost}(\tau)$ *and* $\mathrm{cost}(\pi) > \mathrm{cost}(\tau)$ *can be performed in polynomial time.*

Proof: The tours π and τ have equal cost if, and only if, each of their edges have equal cost, since the edge costs of the TSP are distinct powers of 2. In the case of the inequality, let 2^k be the cost of the largest edge in one of, but not both π and τ. If this edge is in π then $\mathrm{cost}(\pi) > \mathrm{cost}(\tau)$ since $2^k > 2^{k-1} + 2^{k-2} + \ldots + 2^{k-m}$ for any finite integer $m > 0$. $\square$

This means many of the heuristics of Chapter 3 can be applied to the perfect TSP of Algorithm 22 under the conditions of Theorem 21.

Theorem 25 *Let Θ, be the solution space of an n city TSP, T, constructed using Algorithm 22. Let c be any number represented as $2^{k_1} + 2^{k_2} + \ldots + 2^{k_q}$ with q and each k_i polynomial bounded on n. Then it can be decided in polynomial time on n, if there is a tour π in Θ with* $\mathrm{cost}(\pi) = c$*. Furthermore, if π exists, it can be computed in polynomial time.*

Proof: If $q \neq n$ then then no tour cost has representation $2^{k_1}+2^{k_2}+\ldots+2^{k_q}$. Otherwise for each k_i determine if there is an edge in T with cost exponent k_i. If all the k_i have a corresponding edge e in T (with cost exponent k_i) then determine if the n edges form a cycle, if so, it provides π. $\square$

5.3 Examples of the Distribution of Costs

Figures 5.2, 5.3 and 5.4 show the distribution of tour costs of the TSP produced by Algorithm 22 with the accompanying 7 vertex Hamiltonian cycle problem. In each case, the gap parameter is 10 and the offset 0. Figure 5.5 shows the lowest cost tours, those that correspond to Hamiltonian cycles of the graph of Figure 5.4. These plots, as with others in this chapter, represent the probability distribution as the number of tours with a given cost.

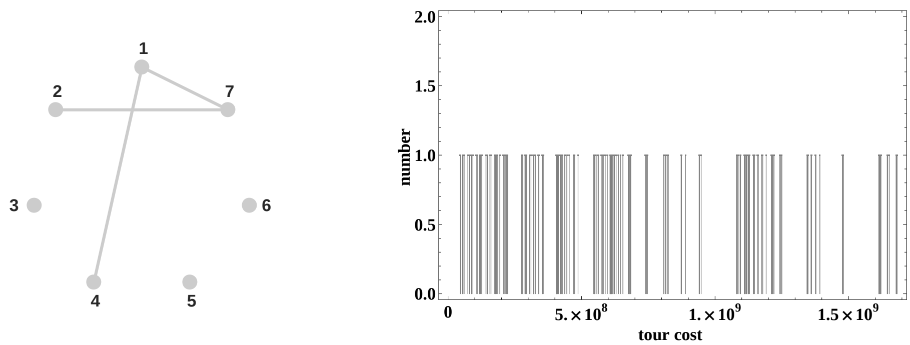


Figure 5.2: *A 7 vertex Hamiltonian cycle problem along with the distribution of costs of the TSP produced by Algorithm 22. The instance was randomly generated with probability of an edge, 0.1. The parameters for Algorithm 22 are gap 10 and offset 0.*

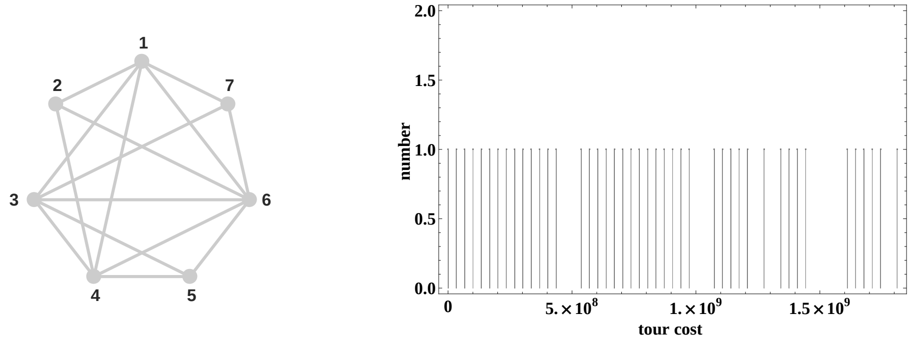


Figure 5.3: *A 7 vertex Hamiltonian cycle problem along with the distribution of costs of the TSP produced by Algorithm 22. The instance was randomly generated with probability of an edge, 0.5. The parameters for Algorithm 22 are gap 10 and offset 0.*

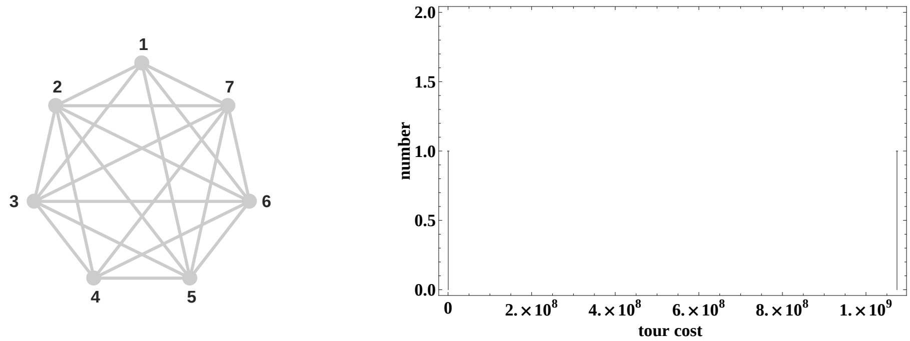
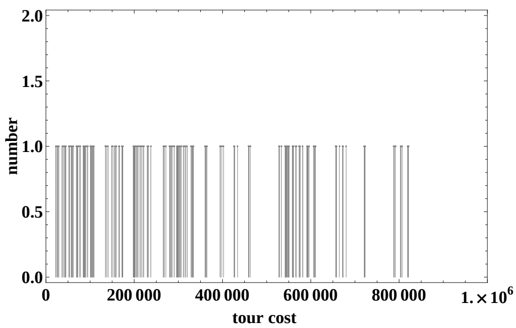


Figure 5.4: *A 7 vertex Hamiltonian cycle problem along with the distribution of costs of the TSP produced by Algorithm 22. The instance was randomly generated with probability of an edge, 0.9. The parameters for Algorithm 22 are gap 10 and offset 0. Figure 5.5 shows details of the low cost tours.*

Figure 5.5: *The low cost tours of the distribution in Figure 5.4. These correspond to the Hamiltonian cycles in the graph of that figure.*



Figures 5.6, 5.7, 5.8 and 5.9 show the distribution of tour costs of the TSP produced by Algorithm 22 over an ensemble of 1000 randomly generated Hamiltonian cycle problems. In each case, the edges were created as a Bernoulli trial, with those in Figure 5.6 having a probability of an edge of 0.1. The probability of an edge occurring in the other ensembles are respectively 0.5, 0.9 and 1.0. In all cases the gap parameter is 10 and the offset 0.

Figure 5.6: *The distribution of tour costs of the TSP produced by Algorithm 22 on an ensemble of 1000 randomly generated 6 vertex Hamiltonian cycle problems each with probability of an edge, 0.1. The parameters for Algorithm 22 are gap 10 and offset 0.*

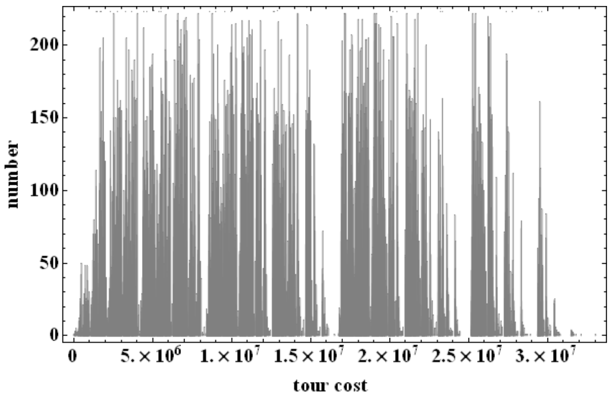
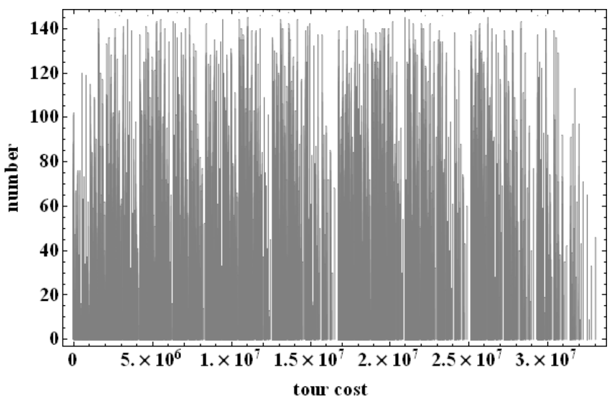


Figure 5.7: *The distribution of tour costs of the TSP produced by Algorithm 22 on an ensemble of 1000 randomly generated 6 vertex Hamiltonian cycle problems each with probability of an edge, 0.5. The parameters for Algorithm 22 are gap 10 and offset 0.*



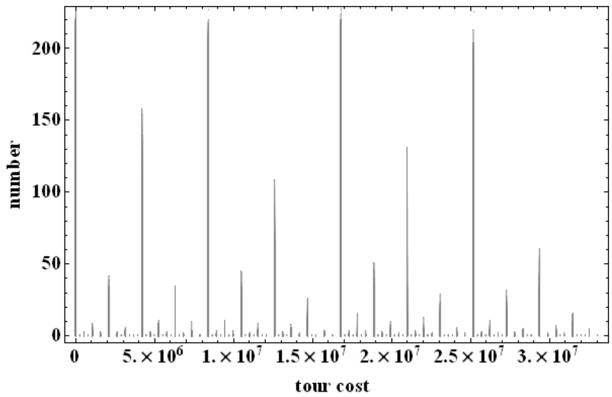


Figure 5.8: *The distribution of tour costs of the TSP produced by Algorithm 22 on an ensemble of 1000 randomly generated 6 vertex Hamiltonian cycle problems each with probability of an edge, 0.9. The parameters for Algorithm 22 are gap 10 and offset 0.*

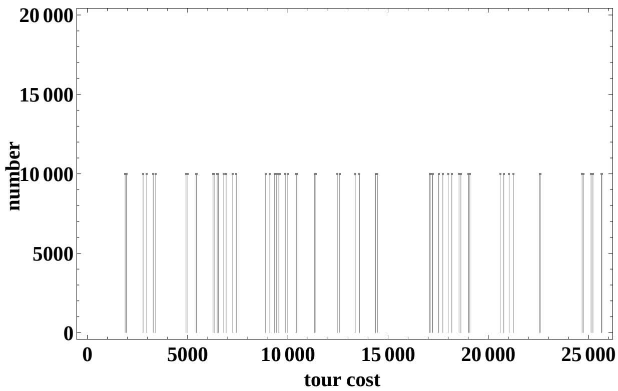


Figure 5.9: *The distribution of tour costs of the TSP produced by Algorithm 22 on an ensemble of 1000 randomly generated 6 vertex Hamiltonian cycle problems each with probability of an edge, 1.0. Each graph is therefore complete. The parameters for Algorithm 22 are gap 10 and offset 0.*

5.4 Conclusion

In this chapter we have given a constructive proof that it is possible to create instances of the TSP with a bijective cost function (but injective to the real numbers). We denote TSPs with this property, perfect TSPs and prove that the decision problem of the perfect TSP is $\mathcal{NP}$-complete.

We presented a polynomial time Karp reduction from the Hamiltonian cycle problem that creates a perfect TSP. We have shown that using this reduction, it is possible in polynomial time, given a rational number, c represented in a reasonable form, to decide if a tour of cost c exists in the solution space of an instance. If such a tour exists it is possible in polynomial time to compute it.

The operations that can be computed on tours, in polynomial time, are limited. However, comparison of tour costs can be performed in fast polynomial time. This makes it possible to use many of the optimisation heuristics discussed in Chapter 3. However, we have little confidence that these standard techniques will be effective on the reduction from the Hamiltonian cycle problem provided. Indeed, the construction described in this chapter was in part motivated by the desire to find problems that were "pathological", in order to study local optima in the 2-opt landscape. As demonstrated empirically, the tour cost distributions resulting from the reduction are non-Gaussian.

Chapter 6

The Expected Value of Tour Costs over Subsets of the Solution Space

In this chapter we provide proofs that the mean tour cost over certain subsets of the solution space of the problem can be computed in fast polynomial time. This result has application in the analysis of algorithms such as branch and discussed in Section 3.1.1, and various other tour construction techniques including the Nearest Neighbour and the Greedy heuristics noted in Sections 3.1.4 and 3.1.7. It also provides an analytical tool to investigate the properties of the solution space discussed in Section 4.6 and also the greedy expectation algorithm of Gutin and Yeo described in Sections 3.1.8. This algorithm operates by a process of tour construction with each additional tour edge chosen to minimise the expected value of tour costs over certain subsets of the solution space. The work of Gutin and Yeo does not extend to the explicit computation of the expected value of tour cost over the subsets of the solution space given here.

6.0.1 The Stochastic TSP as a Generalisation of the Fixed Edge Cost TSP

The elementary properties of the sums of the expected value of random variables noted in Equation 4.5, combined with the fact we have no need, as yet, to compute the product of random variables[1], makes it attractive to present the results of this chapter in terms of a *stochastic TSP*. Here the problem is specified with each edge cost as a random variable from some distribution. For the purposes of computing the expected value of tour costs over the solution space, we assume only that the expected value of the cost of each edge is known and is finite. Other properties of the distributions (of edge costs) are irrelevant to our *present* purpose. In particular, we make no assumption of independence between edge costs.

stochastic TSP

Where, as is typical, the TSP has specified fixed finite edge costs, these results are certainly applicable, since for any fixed $c \in \mathbb{R}$, $\mathrm{Ex}(c) = c$. In this case $\mathrm{Var}(c) = 0$ so the distribution of c is not very interesting, having probability 1 of taking the value c and probability 0 of taking any other value. We proceed with the view that the TSP with fixed finite edge costs is a special case of the stochastic TSP.

6.0.2 The Number of Tours Containing Various Edges

Before providing our main results, we give a useful technical lemma providing the number of tours, that is the number of Hamiltonian cycles, containing various configurations of edges. We will make extensive use of this simple result throughout the remainder of this document.

Lemma 26 *Given a complete graph K, let P be a set of m, non-cyclic, non-singleton paths over K sharing no vertices. Let k be the number of vertices not appearing in any path of P. Then there are $2^{m-1}(k+m-1)!$ Hamiltonian cycles in K containing all the paths in P.*

[1] Computing the product of dependent random variables is less straightforward than computing that of independent random variables.

Proof: Label the paths of P, p_j with $j \in [1 \ldots m]$. We recall that a Hamiltonian cycle is a cyclic permutation of vertices. Therefore, without loss of generality, fix p_1 in position and orientation and write a Hamiltonian cycle as

$$(p_1, i_1, i_2 \ldots, i_q, p_2, i_{q+1} \ldots, p_m \ldots, i_k). \tag{6.1}$$

There are $(k + m - 1)!$ orderings of the free paths and vertices. Each path is at least 2 vertices long and so each of the $m - 1$ free paths has 2 orientations, implying the result. $\square$

6.1 The Expected Value of Tour Costs over the Solution Space

We begin with a well known result giving the mean value of tour costs over the entire solution space [48]. This simple result is typically expressed in terms of the special case of a TSP with fixed edge costs.

Here we consider the probability space $(\Theta, \mathcal{A}_\Theta, \mathrm{P}_\Theta)$ with Θ the solution space of a TSP, that is the set of all tours of an instance. An elementary event in this space is simply a tour. $\mathcal{A}_\Theta$ is the σ-algebra over the subsets of Θ and P_Θ provides the probability of occurrence of any subset in $\mathcal{A}_\Theta$, which is to say, any set of tours. Under this arrangement if π is any tour, then $\mathrm{P}_\Theta(\pi) = \frac{1}{|\Theta|}$. So at the conceptual level, we have a list of all tours with each tour appearing only once. An "unbiased selection" of any tour from this list has probability $\frac{1}{|\Theta|}$ of being a particular tour π.

Let the random variable Cost_Θ denote the cost of a tour in the solution space Θ. Our first goal is to compute $\mathrm{Ex}(\mathrm{Cost}_\Theta)$.

Theorem 27 *Let the random variable* Cost_Θ *be a tour cost of a TSP with edge set E. Let the $|E|$ random variables $C(e)$ be the costs of each edge in $e \in E$. Then the expected value of* Cost_Θ *is* $C(e)$

$$\mathrm{Ex}(\mathrm{Cost}_\Theta) = \frac{2}{(n-1)} \sum_{e \in E} \mathrm{Ex}(C(e)). \tag{6.2}$$

Proof: To compute the expected value of tour costs in Θ we need only sum the expected cost of each tour in Θ and divide this by $|\Theta|$. View each tour in Θ as a set of edges. By Lemma 26, each edge appears in $(n-2)!$ tours. $|\Theta| = \frac{1}{2}(n-1)!$ so, and also by Equation 4.5 we have, $\text{Ex}(\text{Cost}_\Theta) = \frac{2(n-2)!}{(n-1)!} \sum_{e \in E} \text{Ex}(C(e))$, as required. □

6.2 The Expected Value of Tour Costs over Subsets of the Solution Space

In this section, we extend the above result to the conditional case. That is, we wish to compute the expected value of tour costs over each tour in Θ, subject to the condition that certain edges occur in the tours. To this end, let P be a set of paths with the following properties:

- Each path in P is non-cyclic.
- Each path in P has at least one edge and so at least two vertices.
- No two paths share a vertex.

It is worthwhile noting that the paths, when taken together as a single graph, form a subgraph of at least one *tour*. If this subgraph is connected then P contains one path. When we say an edge in P, we mean an edge in one of the paths of P.

Let Θ be the solution space of a TSP and let P be a set of paths with the above properties. Then by $\Theta|P$ we denote the largest subset of Θ such that, $\Theta|P$
if $\pi \in \Theta|P$, then each edge in P is also in E_π, and if $P = \emptyset$ then $\Theta = \Theta|P$.
Let the random variable $\text{Cost}_{\Theta|P}$ be the cost of a tour in $\Theta|P$. $\text{Cost}_{\Theta|P}$

Our goal is to compute the expected value of tour costs over the set of tours $\Theta|P$, that is, $\text{Ex}(\text{Cost}_{\Theta|P})$. We need to consider the number of tours in which an edge can occur. There are four possibilities. An edge e can extend a path in P so that one vertex of the edge is incident to a path in P. An edge e can connect two paths of P, so in this case the edge is incident to two paths of P. The edge could be non-adjacent nor equal to any edge in a path of P. Finally the edge could be in some path of P.

Starting with the first of these possibilities, let $I1_P$ be the set of edges each of which is incident to an end vertex of *one* path of P and not incident to any other path of P. $I1_P$

Lemma 28 *Given a TSP with graph $G = (V, E)$ let P be a set of m, non-cyclic, non-singleton paths over G sharing no vertices. Let k be the number of vertices not appearing in any path of P. Let e be in $I1_P$. Then e occurs in $2^{m-1}(k+m-2)!$ tours also containing the paths of P.*

Proof: The edge e extends a path of P. Therefore applying Lemma 26 (with $k-1$ free vertices) provides the result. □

Let $I2_P$ be the set of edges incident to the end vertices of any two paths of P. So that an edge in $I2_P$ links *two* paths of P to form a single path. $I2_P$

Lemma 29 *Given a TSP with graph $G = (V, E)$ let P be a set of m, non-cyclic, non-singleton paths over G sharing no vertices. Let k be the number of vertices not appearing in any path of P. Let e be in $I2_P$. Then e occurs in $2^{m-2}(k+m-2)!$ tours also containing the paths of P.*

Proof: The edge e combines two paths of P. Therefore applying Lemma 26 (with $m-1$ paths) provides the result. □

Let N_P be the set of edges non-adjacent nor equal to any edge in a path of P. N_P

Lemma 30 *Given a TSP with graph $G = (V, E)$ let P be a set of m, non-cyclic, non-singleton paths over G sharing no vertices. Let k be the number of vertices not appearing in any path of P. Let e be in E if $e \in N_P$. Then e occurs in $2^m(k+m-2)!$ tours also containing the paths of P.*

Proof: The edge e forms a path. Therefore applying Lemma 26 (with $m+1$ paths and $k-2$ free vertices) provides the result. □

Figure 6.1 Illustrates these sets in a twelve city problem.

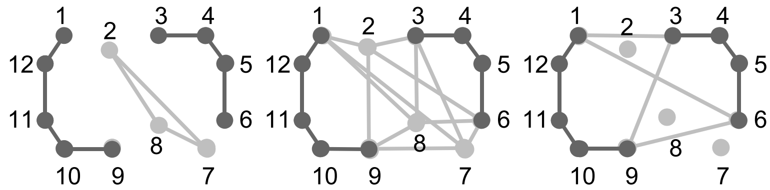


Figure 6.1: *The sets $N_P, I1_P$ and $I2_P$ in a twelve city instance. The dark edges form the two paths in P. (left) The light edges are the set N_P. (centre) The light edges are the set $I1_P$. (right) The light edges are the set $I2_P$.*

Theorem 31 *Given a TSP with graph $G = (V, E)$ let P be a set of m, non-cyclic, non-singleton paths over G sharing no vertices. Let k be the number of vertices not appearing in any path of P. The mean cost of tours containing all the edges in the paths of P is*

$$\mathrm{Ex}(\mathrm{Cost}_{\Theta|P}) \quad = \sum_{e\in P} \mathrm{Ex}(C(e)) + \frac{\sum_{e\in I1_P} \mathrm{Ex}(C(e))}{(k+m-1)} + \frac{\sum_{e\in I2_P} \mathrm{Ex}(C(e))}{2(k+m-1)} + \frac{2\sum_{e\in N_P} \mathrm{Ex}(C(e))}{(k+m-1)} \tag{6.3}$$

Proof: To compute the expected value of tour costs in $\Theta|P$, we need only sum the expected cost of each tour in $\Theta|P$ and divide this by $|(\Theta|P)|$. By Lemma 26, $|(\Theta|P)| = 2^{m-1}(k + m - 1)!$.

Now view each tour as a set of edges. Any edge e in a tour in $\Theta|P$ is in precisely one of $P, I1_P, I2_P$ or N_P.

case 1 If e is in a path of P then e occurs in $2^{m-1}(k + m - 1)!$ tours.

case 2 If e is in $I1_P$ then by Lemma 28 e occurs in $2^{m-1}(k + m - 2)!$ tours.

case 3 If e is in $I2_P$ then by Lemma 29 e occurs in $2^{m-2}(k + m - 2)!$ tours.

case 4 If e is in N_P then by Lemma 30 e occurs in $2^m(k + m - 2)!$ tours.

$$\begin{aligned}
\mathrm{Ex}(\mathrm{Cost}_{\Theta|P}) &= \tfrac{2^{m-1}(k+m-1)!}{2^{m-1}(k+m-1)!}\sum_{e\in P}\mathrm{Ex}(C(e)) \\
&+ \tfrac{2^{m-1}(k+m-2)!}{2^{m-1}(k+m-1)!}\sum_{e\in I1_P}\mathrm{Ex}(C(e)) \\
&+ \tfrac{2^{m-2}(k+m-2)!}{2^{m-1}(k+m-1)!}\sum_{e\in I2_P}\mathrm{Ex}(C(e)) \\
&+ \tfrac{2^{m}(k+m-2)!}{2^{m-1}(k+m-1)!}\sum_{e\in N_P}\mathrm{Ex}(C(e))
\end{aligned}$$

which implies the result. □

Corollary 32 *The sum of Theorem 32 can be computed in $\mathcal{O}(n^2)$ at worst for an n city TSP.*

None of the sets $P, I1_P, I2_P$ or N_P of the theorem is larger than $\mathcal{O}(n^2)$. □

It is worth pointing out that if P is empty then $\Theta|P = \Theta$ and $N_p = E$ and $k = |V|$ and the result reduces to Theorem 27.

6.3 Conclusion and Further Research

In this chapter we have shown that given a subgraph, P, of some tour π on an n city TSP, the expected value of the cost of tours containing all the edges of P can be computed in $\mathcal{O}(n^2)$. The result has application to both the fixed edge cost case and to the stochastic TSP with edge costs as random variables each with arbitrary distribution, but known finite expected value.

It would be interesting, and useful, to extend this result to find the expected value of tour costs of those tours containing P but excluding the tours containing one or more edges from a set X. Unfortunately, where X does not form a disjoint set of non-cyclic paths, the problem will probably reduce to a computation over the power set of X. However for low cardinality X, the problem may be tractable.

The results of this chapter have application in the analysis of heuristics, including but not limited to, the branch and bound, nearest neighbour and greedy expectation algorithms. So for example, in the case of the branch and bound algorithm of Section 3.1.1, the expected cost of tours with edges in the sets $v.M$ of Algorithm 3 can be computed. In the case of the nearest neighbour algorithm, the expected value of all tours containing the partially completed tour, P, at line 6 of Algorithm 7 can be found.

The results also have application in the investigation of the solution space of the problem, particularly the results of Boese et al. on the relationship between the cost of tours and the distance between tours in the 2-opt and 3-opt landscapes. Gómez and Barán [44] argue, that both Ant Colony optimisation and Evolutionary algorithms work in large part because of these features of the solution space.

In the next chapter we consider the second moment of tour costs over the solution space of an instance.

Chapter 7

The Variance of Tour Costs

In this chapter we give a polynomial time algorithm to find the population variance of tour costs over the solution space of a TSP. In practical terms, the algorithm provides an $\mathcal{O}(n^2)$ method to compute the standard deviation of tour costs over the solution space of an n city problem with fixed edge costs. In the stochastic case, where the problem is specified in terms of edge costs which are pairwise independent random variables with known mean and variance, the result provides an $\mathcal{O}(n^4)$ algorithm to compute the variance of tour costs.

7.1 Fixed Edge Costs

Our central theorem will, in large part, be concerned with the number of tours in which certain short paths of edges occur. Our first task is to simply apply Lemma 26 to these configurations of edges. Table 7.1 enumerates the three cases of interest.

7.1.1 The Variance Theorem with Fixed Edge Costs

In order to prove our central theorem we provide some notational machinery. Let Θ be the solution space of a TSP with edge set E and cost function c_Θ, so c_Θ
that $c_\Theta(\pi)$ is the cost of a particular tour π. By, C_Θ, we denote the random C_Θ
variable, the cost of a tour (unspecified) in the solution space, Θ. A_p is the A_p

Table 7.1: *The three ways that, up to two unlabelled edges, can be arranged into paths in tours of size n. The − character represents an edge, so −− means a path with 2 edges and three vertices. The ⇜⇝ symbol, the set (possibly empty) of free vertices between unconnected paths.*

Case	Pattern	Paths m	Free vertices k	Tours	Cities n
1	− ⇜⇝	1	$(n-2)$	$(n-2)!$	$n>2$
2	−− ⇜⇝	1	$(n-3)$	$(n-3)!$	$n>2$
3	− ⇜⇝ − ⇜⇝	2	$(n-4)$	$2(n-3)!$	$n>3$

set of edges adjacent to edge e_p.

We index each π in Θ with an integer $m \in [1 \dots |\Theta|]$. Similarly we label the edges of E as e_i with $i \in [1 \dots |E|]$. We define the function $[1 \dots |\Theta|] \times [1 \dots |E|] : t \rightarrow \{0, 1\}$ as

t_{mi}

$$t_{mi} = \begin{cases} 1 & \text{if edge } e_i \text{ is in tour } m \\ 0 & \text{otherwise.} \end{cases} \tag{7.1}$$

Under this arrangement, if m is the index of a tour π, then the cost of π is

$$c_\Theta(\pi) = t_{m1}c(e_1) + t_{m2}c(e_2) \dots t_{m|E|}c(e_{|E|}) \tag{7.2}$$

and applying Equations 4.4 and 4.7 for $k = 2$ we have.

$$\mathrm{Var}(C_\Theta) = \frac{1}{|\Theta|}\sum_{m=1}^{|\Theta|}((t_{m1}c(e_1) + t_{m2}c(e_2) \dots t_{m|E|}c(e_{|E|}) - \mathrm{Ex}(C_\Theta))^2). \tag{7.3}$$

Now $|\Theta|$ is, of course, factorial on n and so this formulation is impractical for all but the smallest problems. In Theorem 33 we give a polynomial time solution to the problem.

Theorem 33 *Let C_Θ be the random variable, the cost of a tour in the solution space of a TSP with n cities and with edge set E. The variance of C_Θ is*

$$\mathrm{Var}(C_\Theta) = \frac{2\beta_1}{(n-1)} - \frac{4\beta_1 + 2\beta_2}{(n-1)(n-2)} \tag{7.4}$$

with the functions $\beta_1, \beta_2 : E \to \mathbb{R}$ *defined as*

$$\begin{aligned} \beta_1 &= \sum_{e_p \in E} c_0(e_p)^2 \\ \beta_2 &= \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \end{aligned} \tag{7.5}$$

where $c_0(e_p) = c(e_p) - \mathrm{Ex}(C_\Theta)/n$ and A_p is the set of edges adjacent to edge e_p. c_0

Proof: Each tour has only n edges, so for a given tour m, only n of the t_{mi} are 1, the remaining are 0. This gives Equation 7.3 as

$$\mathrm{Var}(C_\Theta) = \frac{1}{|\Theta|} \sum_{m=1}^{|\Theta|} ((t_{m1}c_0(e_1) + t_{m2}c_0(e_2) \ldots t_{m|E|}c_0(e_{|E|}))^2), \tag{7.6}$$

$$\mathrm{Var}(C_\Theta) = \frac{1}{|\Theta|} \sum_{m=1}^{|\Theta|} \sum_{i=1}^{|E|} \sum_{j=1}^{|E|} t_{mi}t_{mj}c_0(e_i)c_0(e_j). \tag{7.7}$$

Expanding the sums we have

$$\sum_{m=1}^{|\Theta|} \sum_{i=1}^{|E|} \sum_{j=1}^{|E|} t_{mi}t_{mj}c_0(e_i)c_0(e_j) \tag{7.8}$$

$$\begin{array}{cccc} = & \alpha_{11}c_0(e_1)c_0(e_1)+ & \ldots & +\alpha_{1|E|}c_0(e_1)c_0(e_{|E|}) \\ + & \alpha_{21}c_0(e_2)c_0(e_1)+ & \ldots & +\alpha_{2|E|}c_0(e_2)c_0(e_{|E|}) \\ \vdots & \vdots & \ddots & \vdots \\ + & \alpha_{|E|1}c_0(e_{|E|})c_0(e_1)+ & \ldots & +\alpha_{|E||E|}c_0(e_{|E|})c_0(e_{|E|}) \end{array} \tag{7.9}$$

where $\alpha_{ij} = \sum_{m=1}^{|\Theta|} t_{mi}t_{mj}$, is the number of tours which contain both edges e_i and e_j.

Let e_i and e_j be any two edges of a TSP. Either these edges are equal, they are adjacent or they are neither equal nor adjacent. Thus each constant α_{ij} is one of three values.

case 1 α_{ii}. By Lemma 26 and Case 1 of Table 7.1 each edge e_i appears in $(n-2)!$ tours over the solution space, thus $\alpha_{ii} = (n-2)!$.

case 2 α_{ij} such that edge e_i is adjacent to e_j. By Lemma 26 and Case 2 of Table 7.1 any two adjacent edges appear in $(n-3)!$ tours so in this case $\alpha_{ij} = (n-3)!$.

case 3 α_{ij} such that edge e_i is non-adjacent to e_j. By Lemma 26 and Case 3 of Table 7.1 the two edges appear in $2(n-3)!$ tours so $\alpha_{ij} = 2(n-3)!$.

We recall that A_p is the set of edges adjacent to an edge e_p and we define N_p to be the set of edges neither adjacent to e_p nor equal to e_p. So Equation 7.8 becomes N_p

$$\begin{aligned} &= (n-2)! \sum_{e_p \in E} c_0(e_p)^2 \\ &+ (n-3)! \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \\ &+ 2(n-3)! \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in N_p} c_0(e_q), \end{aligned} \tag{7.10}$$

$$= (n-2)!\beta_1 + (n-3)!\beta_2 + 2(n-3)!\beta_3, \tag{7.11}$$

where

$$\beta_3 = \sum_{e_p \in E} c_0(e_p) \sum_{f \in N_p} c_0(e_q). \tag{7.12}$$

This gives the variance as

$$\begin{aligned} \mathrm{Var}(C_\Theta) &= \tfrac{2((n-2)!\beta_1 + (n-3)!\beta_2 + 2(n-3)!\beta_3)}{(n-1)!} \\ &= \tfrac{2\beta_1}{(n-1)} + \tfrac{2\beta_2 + 4\beta_3}{(n-1)(n-2)}. \end{aligned} \tag{7.13}$$

However it is easy to see that $\beta_3 = -\beta_1 - \beta_2$, since for any e_p in E we have $E = A_p \bigcup N_p \bigcup \{e_p\}$ and $\sum_{e_p \in E} c_0(e_p) = 0$ (by the definition of c_0). Therefore we have

$$\mathrm{Var}(C_\Theta) = \tfrac{2\beta_1}{(n-1)} - \tfrac{4\beta_1 + 2\beta_2}{(n-1)(n-2)} \tag{7.14}$$

as required. □

A naive computation of β_2 would have time complexity $\mathcal{O}(n^3)$. Figure 7.1

illustrates the relationship, between the set A_p and the sets of edges incident to e_p, that is exploited in the next corollary to improve this complexity.

Corollary 34 *It is possible to calculate the variance of tour costs over the solution space of any n city TSP with fixed edge costs in $\mathcal{O}(n^2)$.*

Proof: The function β_1 above can clearly be found in $\mathcal{O}(n^2)$, since $|E| = (n^2 - n)/2$. Let I_x be the set of edges incident to a city x. Let $e_p = \{x, y\}$ be an edge, then $A_p = (I_x - \{e_p\}) \bigcup (I_y - \{e_p\})$. Thus to find β_2, first compute and store $S_i = \sum\limits_{e \in I_i} c_0(e)$ for each $i \in [1, \ldots, n]$. That is for each of the n cities of the instance we compute and store the sum of the edge costs incident to the city. The time complexity of this phase is $\mathcal{O}(n^2)$ and space complexity is $\mathcal{O}(n)$. The sum β_2 is then I_x

$$\beta_2 = \sum_{e_p = \{x,y\} \in E} c_0(e_p)(S_x + S_y - 2c_0(e_p)), \tag{7.15}$$

and this can clearly be performed in $\mathcal{O}(n^2)$. □

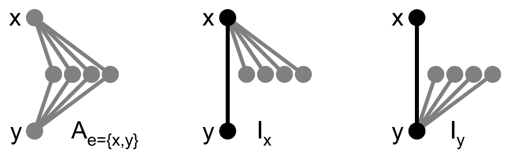


Figure 7.1: *The sets A_p, I_x and I_y in a six city instance.*

Properties of β_1

We define the random variable C_E to be the cost of an unspecified edge in E. C_E In terms of interpreting the above results it is worth noting that β_1 relates to the variance of edge costs over E, specifically

Lemma 35 *Let the random variable C_E be the cost of an edge in E, then*

$$\mathrm{Var}(C_E) = \frac{2\beta_1}{n(n-1)}. \tag{7.16}$$

Proof: This follows directly from Equation 4.7 noting that for an n city TSP $|E| = \frac{1}{2}n(n-1)$ and $\mathrm{Ex}(C_\Theta) = n\mathrm{Ex}(C_E)$. □

7.2 Random Edge Costs

Here we consider the case where the edge costs of an instance are not numbers, but rather *pairwise independently* distributed random variables with known mean and variance. We prove that the variance of tour costs over the solution space under these conditions can be computed in $\mathcal{O}(n^4)$ for an n city TSP.

As before, the random variable C_Θ denotes the cost of an unspecified tour in the solution space Θ of an n city TSP with graph $G = (V, E)$. The $|E|$ random variables $C(e)$, are the costs of each edge $e \in E$. The $|\Theta|$ random variables C_i denote the the cost of each specific tour, tour i, in Θ. We begin with a well known result which enables us to compute the expected value of a random variable in terms of a decomposition of its domain.

$C(e)$

C_i

Let X be a random variable in a probability space $(\Omega, \mathcal{A}_\Omega, \mathrm{P}_\Omega)$. By $\mathrm{Ex}(X|\omega_i)$ we denote the expected value of X with the domain of X restricted to some set of events $\omega_i \in \mathcal{A}_\Omega$. $\mathrm{Ex}(X|\omega_i)$ is termed the *conditional expectation* of X on ω_i. Let the collection of events $\omega_1, \omega_2, \ldots, \omega_q$ be a partition of Ω. This is to say, each pair ω_i, ω_j is disjoint and the union of all the ω_i is equal to Ω. The following lemma gives $\mathrm{Ex}(X)$ as a weighted sum of each $\mathrm{Ex}(X|\omega_i)$.

conditional expectation

Lemma 36 (Conditional Expectation) *Let X be a random variable over a probability space $(\Omega, \mathcal{A}_\Omega, \mathrm{P}_\Omega)$. Let the collection of events $\omega_1, \omega_2, \ldots, \omega_q$ be a partition of Ω, with $P_\Omega(\omega_i) > 0$ for all $i \in [1, q]$. Then,*

$$\mathrm{Ex}(X) = \sum_{i=1}^{q} \mathrm{Ex}(X|\omega_i) P_\Omega(\omega_i). \tag{7.17}$$

Proof: Proofs are given in Chung [22] and Shiryaev [105]. □

We are now able to make rigorous the intuitive notion that, the expected value of C_Θ^2, should be simply, the arithmetic mean of the expected values of each of the C_i^2.

Lemma 37 *Let Θ be the solution space of a TSP and let the random variable C_Θ be the tour cost over all Θ. Let the $|\Theta|$ random variables $C_1, C_2, \ldots, C_{|\Theta|}$*

be the cost of each individual tour in Θ*. Then,*

$$\mathrm{Mo}_2(C_\Theta) \quad = \tfrac{1}{|\Theta|}\left(\mathrm{Ex}(C_1^2) + \mathrm{Ex}(C_2^2) + \ldots + \mathrm{Ex}(C_{|\Theta|}^2)\right) \tag{7.18}$$

Proof: By Equation 4.8 $\mathrm{Mo}_2(C_\Theta) = \mathrm{Ex}(C_\Theta^2)$. Each tour of Θ is an elementary event ω_i with $i \in [1, |\Theta|]$ and for each i, $P_\Theta(\omega_i) = \frac{1}{|\Theta|}$. In addition, the collection of all individual tours naturally partitions Θ. Therefore, writing $\mathrm{Ex}(C_\Theta^2|\omega_i) = \mathrm{Ex}(C_i^2)$ and applying Lemma 36 provides the result. □

Theorem 38 *Let the random variable* C_Θ *be the tour cost of a TSP with graph* $G = (V, E)$*. Let the* $|E|$ *random variables,* $C(e_p)$ *be the costs of each edge in* $e_p \in E$*. If the* $C(e_p)$ *are pairwise independently distributed, with mean* $\mathrm{Ex}(C(e_p))$ *and second moment about the origin* $\mathrm{Mo}_2(C(e_p))$ *then the second moment about the origin of* C_Θ *is*

$$\begin{aligned}\mathrm{Mo}_2(C_\Theta) \quad &= \quad \tfrac{2}{(n-1)} \sum_{e_p \in E} \mathrm{Mo}_2(C(e_p)) \\ &+ \quad \tfrac{2}{(n-1)(n-2)} \sum_{e_p \in E, e_q \in A_p} \mathrm{Ex}(C(e_p))\mathrm{Ex}(C(e_q)) \\ &+ \quad \tfrac{4}{(n-1)(n-2)} \sum_{e_p \in E, e_q \in N_p} \mathrm{Ex}(C(e_p))\mathrm{Ex}(C(e_q)),\end{aligned} \tag{7.19}$$

where A_p *is the set of edges adjacent to an edge* e_p *and* N_p *is the set of edges neither adjacent to nor equal to* e_p*.*

Proof:

Let the random variable C_m be the cost of tour m. By Lemma 37 we have

$$\mathrm{Mo}_2(C_\Theta) \quad = \tfrac{1}{|\Theta|}\left(\mathrm{Ex}(C_1^2) + \mathrm{Ex}(C_2^2) + \ldots + \mathrm{Ex}(C_{|\Theta|}^2)\right). \tag{7.20}$$

Defining t_{mi}, as before, to be 1 if tour m contains edge i and 0 otherwise, we have.

$$\mathrm{Mo}_2(C_\Theta) \quad = \tfrac{1}{|\Theta|} \sum_{m=1}^{|\Theta|} \mathrm{Ex}((t_{m1}C(e_1) + t_{m2}C(e_2) \ldots t_{m|E|}C(e_{|E|}))^2). \tag{7.21}$$

Multiplying out each square and applying Equation 4.5 gives,

$$\mathrm{Mo}_2(C_\Theta) = \frac{1}{|\Theta|}\sum_{m=1}^{|\Theta|}\sum_{i=1}^{|E|}\sum_{j=1}^{|E|} t_{mi}t_{mj}\mathrm{Ex}(C(e_i)C(e_j)). \tag{7.22}$$

Expanding the sums we have

$$\sum_{m=1}^{|\Theta|}\sum_{i=1}^{|E|}\sum_{j=1}^{|E|} t_{mi}t_{mj}\mathrm{Ex}(C(e_i)C(e_j)) \tag{7.23}$$

$$\begin{array}{lllll} = & \alpha_{11}\mathrm{Ex}(C(e_1)C(e_1))+ & \dots & +\alpha_{1|E|}\mathrm{Ex}(C(e_1)C(e_{|E|})) \\ + & \alpha_{21}\mathrm{Ex}(C(e_2)C(e_1))+ & \dots & +\alpha_{2|E|}\mathrm{Ex}(C(e_2)C(e_{|E|})) \\ \vdots & \vdots & \ddots & \vdots \\ + & \alpha_{|E|1}\mathrm{Ex}(C(e_{|E|})C(e_1))+ & \dots & +\alpha_{|E||E|}\mathrm{Ex}(C(e_{|E|})C(e_{|E|})) \end{array} \tag{7.24}$$

where $\alpha_{ij} = \sum_{m=1}^{|\Theta|} t_{mi}t_{mj}$, is the number of tours which contain both edges e_i and e_j.

Let e_i and e_j be any two edges of a TSP, either these edges are equal, they are adjacent or they are neither equal nor adjacent. Thus each constant α_{ij} is one of three values and, as in the proof of Theorem 33, we apply Lemma 26 to resolve these. So Equation 7.23 becomes

$$\begin{array}{ll} = & (n-2)!\sum_{e_p\in E}\mathrm{Ex}(C(e_p)^2) \\ + & (n-3)!\sum_{e_p\in E, e_q\in A_p}\mathrm{Ex}(C(e_p)C(e_q)) \\ + & 2(n-3)!\sum_{e_p\in E, e_q\in N_p}\mathrm{Ex}(C(e_p)C(e_q)), \end{array} \tag{7.25}$$

and

$$\begin{array}{lll} \mathrm{Mo}_2(C_\Theta) & = & \frac{2}{(n-1)}\sum_{e_p\in E}\mathrm{Ex}(C(e_p)^2) \\ & + & \frac{2}{(n-1)(n-2)}\sum_{e_p\in E, e_q\in A_p}\mathrm{Ex}(C(e_p)C(e_q)) \\ & + & \frac{4}{(n-1)(n-2)}\sum_{e_p\in E, e_q\in N_p}\mathrm{Ex}(C(e_p)C(e_q)). \end{array} \tag{7.26}$$

By the definition of Mo_2 we have

$$\sum_{e_p \in E} \mathrm{Ex}(C(e_p)^2) = \sum_{e_p \in E} \mathrm{Mo}_2(C(e_p)). \tag{7.27}$$

Also, since the edge costs are pairwise independently distributed we have

$$\sum_{e_p \in E, e_q \in A_p} \mathrm{Ex}(C(e_p)C(e_q)) = \sum_{e_p \in E, e_q \in A_p} \mathrm{Ex}(C(e_p))\mathrm{Ex}(C(e_q)), \tag{7.28}$$

and similarly for the non-adjacent edge pairs. This gives

$$\begin{aligned} \mathrm{Mo}_2(C_\Theta) &= \tfrac{2}{(n-1)} \sum_{e_p \in E} \mathrm{Mo}_2 C(e_p) \\ &+ \tfrac{2}{(n-1)(n-2)} \sum_{e_p \in E, e_q \in A_p} \mathrm{Ex}(C(e_p))\mathrm{Ex}(C(e_q)) \\ &+ \tfrac{4}{(n-1)(n-2)} \sum_{e_p \in E, e_q \in N_p} \mathrm{Ex}(C(e_p))\mathrm{Ex}(C(e_q)), \end{aligned} \tag{7.29}$$

as required. □

Corollary 39 *Let the random variable C_Θ be the tour cost of an n city TSP with graph $G = (V, E)$. Let the $|E|$ random variables $C(e)$ be the costs of each edge in $e \in E$. If the $C(e)$ are pairwise independently distributed, with known mean and variance then* $\mathrm{Var}(C_\Theta)$ *can be computed in $\mathcal{O}(n^4)$.*

Proof: Apply Equation 4.9 to each edge cost. Then Theorems 38 and 27 and Equation 4.9. □

7.3 Depth and Problem Size

In this section we consider the application of Theorem 33 to the empirical investigation of the relationship between the cost of *low cost* tours and the standard deviation of *all* tour costs.

By the *depth* of a solution, π, we mean the number of standard deviations from the cost, $c_\Theta(\pi)$, of the solution to the mean cost of all tours, $\mathrm{Ex}(C_\Theta)$.

This motivates the following definition: the depth of a solution π of a TSP is

$$\text{depth}(\pi) = \frac{\text{Ex}(C_\Theta) - c_\Theta(\pi)}{\sqrt{(\text{Var}(C_\Theta))}}. \tag{7.30}$$

7.3.1 Real World Problems

We examine two sets of real world problems. In the first case, 95 problem instances were taken from the well known TSPLIB database [96]. The second set, of 39 instances, originates from the genomics community and arises from physical mapping of canine DNA by the radiation-hybrid (RH) method. The specific data set used was obtained from the RHDF9000 dog radiation hybrid panel [34, 53], with each of the 39 TSP instances corresponding to the RH data over a single canine chromosome.

Empirical Results on TSPLIB Instances.

Each of the problems we considered are under 6000 cities in size and has an approximate embedding on a two dimensional surface. In each case the mean and standard deviation of tour costs were determined by application of Theorem 27 and Corollary 39 above.

Known optimal solution costs from [57] for the 95 cases were used to compare the depth of the optimal tours to the size of the problem. Figure 7.2 shows the results of this survey. It indicates a striking linear relationship between the depth of the optimal solutions and the square root of the problem sizes, $\sqrt{n}$. Indeed the Spearman's correlation coefficient between the two is 0.989 with a two tailed level of statistical significance of less than 0.001. It is also significant that this correlation is *stronger* than that observed between $\text{Ex}(C_\Theta)/\sqrt{(\text{Var}(C_\Theta))}$ and $\sqrt{n}$. The results of curve fitting using a linear least squares regression model are summarized in Table 7.2.

Relationship to Other Work

In the case of planar instances our experimental evidence confirms the findings of Basel and Willemain who found the depth of the optimal solution, π^*, relates to the instance size, n, by depth$(\pi^*) = -3.289 + 2.307n^{0.5}$. They *estimated* the mean and standard deviation by random sampling in 17 instances from TSPLIB.

Empirical Results on Canine RH Instances

In each of the 39 cases, the RH panel data was converted to a TSP using the CarthaGène package [30]. The resulting TSP instances have sizes ranging between 68 and 588 cities. For each instance, the optimal solution cost was estimated using the Lin-Kernighan algorithm. Given the size of the problems it is probable that all of the solutions found are in fact optimal, and that any that are not, are within a few percent of optimal. This data coupled with the mean and standard deviation of tour costs was used to approximate the depth of the optimal solutions in each case. Again a striking correlation between the depth of the best solution seen and the square root of the problem size was found, the relationship being near linear as is evident in Figure 7.2. The Spearman's correlation coefficient between the two is 0.988 with a two tailed level of statistical significance of less than 0.001. Again the correlation was *stronger* than that between $\mathrm{Ex}(C_\Theta)/\sqrt{(\mathrm{Var}(C_\Theta))}$ and $\sqrt{n}$. Table 7.2 summarizes the results of curve fitting using least squares regression. In the case of the TSPLIB data a linear model on $\sqrt{n}$ was applied. In the case of the canine RH problems a near linear power model on $\sqrt{n}$ provided the best fit. The resulting best fit expressions are presented in terms of n in the table.

Table 7.2: *Curve fitting using least squares regression*

Problem	Best fit	sig.	df.	ρ^2
TSPLIB	depth $= -4.27 + 2.29n^{0.5}$	< 0.001	93	0.988
canine RH	depth $= 0.806n^{0.796}$	< 0.001	37	0.982

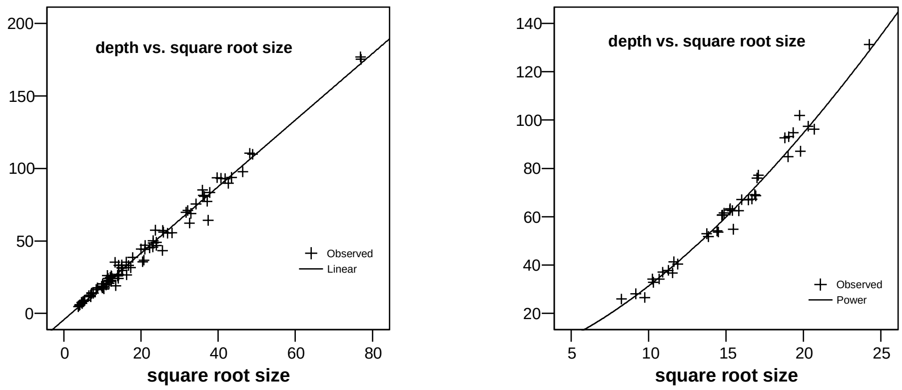


Figure 7.2: *The relationship between the depth of the optimal solution and the square root of the problem size. Left, 95 instances from the TSPLIB problem set. Right, 39 instances each originating from canine RH panel data.*

7.4 Conclusions and Future Research

In this chapter we have demonstrated that the TSP is well enough constrained to be amenable to statistical analysis. In the case where an n city TSP has fixed edge costs, we have proven that the variance of tour costs over the solution space may be computed in time $\mathcal{O}(n^2)$. Where the edge costs are pairwise independently distributed random variables with known mean and variance, we have proven that the variance of tour costs can be computed in $\mathcal{O}(n^4)$.

It seems reasonable that, in the stochastic case this complexity can be improved using similar arguments to the fixed edge cost case. We suspect a computational complexity of $\mathcal{O}(n^3)$ may be achievable. In the stochastic case with city coordinates as independently distributed random variables, the results are not directly applicable, since independence in the vertex coordinates does not imply independence in edge costs. However, it may be possible by consideration of the *covariance* between the edge costs to extend our result to this case.

We anticipate that, by incorporating the argument followed in Theorem 31 of the previous chapter, it will be possible to extend the results of this chapter to consider subsets of the solution space. That is, to compute the variance of tour costs over the set $\Theta|P$. However, we also anticipate this would result in an increase in the computational complexity to $\mathcal{O}(n^4)$ for a fixed cost instance with n cities.

Finally, these results have obvious generalisation to the ATSP and show promise of being generalisable to other variations of the problem.

In the next chapter we extend the central result of this chapter to compute the third moment of tour costs over the solution space of a fixed edge cost instance. In addition, in that chapter we extend our experimental work by examining the relationship between optimal tour cost and skewness in fixed size problems. In Section 11.2 we return to the topic of the relationship between the standard deviation and the optimal tour cost.

Chapter 8

The Third Moment of Costs over the Solution Space

In this chapter we consider the third moment of tour costs over the solution space of the TSP. Our result is confined to the case of the TSP with fixed edge costs. We show that, for an n city problem, the third moment about the mean can be computed in $\mathcal{O}(n^4)$. This result provides a method to compute the third central and factorial moments together with the skewness of the distribution of tour costs.

We apply our method to compute the third moment to two empirical studies. Firstly, we examine skewness versus instance size in four problem types. Secondly, we examine the relationship between skewness versus optimal tour cost for three types of fixed instance size problems. The results of these experiments, together with the empirical results of the last chapter, suggest that it may be possible to construct models to estimate the likely optimal tour cost based on: the problem type of an instance, its size, mean, standard deviation and skewness.

8.1 Third Moment of Tour Costs

Our proof is a natural extension of Theorem 33 in Chapter 7. We begin, as before, by considering the number of tours (Hamiltonian cycles) containing various configurations of edges. The six cases of interest are enumerated in Table 8.1. These are arrived at by Lemma 26.

Lemma 40 (Lemma 26 repeated.) *Given a complete graph K, let P be a set of m, non-cyclic, non-singleton paths over K sharing no vertices. Let k be the number of vertices not appearing in any path of P. Then there are $2^{m-1}(k+m-1)!$ Hamiltonian cycles in K containing all the paths in P.*

Proof: See Lemma 26 of Chapter 6. □

Table 8.1: *The six ways that, up to three unlabelled edges, can be arranged into paths in tours of size n. The $-$ character represents an edge, so $--$ means a path with 2 edges and three vertices. The $\leftrightsquigarrow$ symbol, the set (possibly empty) of free vertices between unconnected paths. The number of paths is given by m, while k is the number of free vertices.*

Case	Pattern	m	k	Num. Tours	Cities n
1	$- \leftrightsquigarrow$	1	$(n-2)$	$(n-2)!$	$n>2$
2	$-- \leftrightsquigarrow$	1	$(n-3)$	$(n-3)!$	$n>2$
3	$- \leftrightsquigarrow - \leftrightsquigarrow$	2	$(n-4)$	$2(n-3)!$	$n>3$
4	$--- \leftrightsquigarrow$	1	$(n-4)$	$(n-4)!$	$n>3$
5	$-- \leftrightsquigarrow - \leftrightsquigarrow$	2	$(n-5)$	$2(n-4)!$	$n>4$
6	$- \leftrightsquigarrow - \leftrightsquigarrow - \leftrightsquigarrow$	3	$(n-6)$	$4(n-4)!$	$n>5$

8.1.1 The Third Moment Theorem

As before, let Θ be the solution space of a TSP with edge set E and cost function C_Θ. We index each π in Θ with an integer $m \in [1 \ldots |\Theta|]$, similarly we label the edges of E as e_i with $i \in [1 \ldots |E|]$. We define the function $[1 \ldots |\Theta|] \times [1 \ldots |E|] : t \rightarrow \{0, 1\}$ as

$$t_{mi} = \begin{cases} 1 & \text{if edge } e_i \text{ is in tour } m \\ 0 & \text{otherwise.} \end{cases}$$

Under this arrangement, if m is the index of a tour π, then the cost of π is

$$C_\Theta(\pi) = t_{m1}c(e_1) + t_{m2}c(e_2) \ldots t_{m|E|}c(e_{|E|}),$$

and specializing Equation 4.7 to $k = 3$, the third moment about the mean μ is

$$\mathrm{Mm}_3(\mathrm{C}_\Theta) = \frac{\sum_{m=1}^{|\Theta|} ((t_{m1}c(e_1) + t_{m2}c(e_2) \ldots t_{m|E|}c(e_{|E|}) - \mu)^3)}{|\Theta|}. \tag{8.1}$$

Now $|\Theta|$ is, of course, factorial on n and so this formulation is impractical for all but the smallest problems. In Theorem 41 we give a polynomial time solution to the problem.

Returning to notational matters, as before A_p is the set of edges adjacent to edge e_p. Let $N_{p,q,\ldots}$ be the set of edges neither adjacent to nor equal to edges $e_p, e_q, \ldots$, so $N_{p,q,\ldots} = E - (A_p \bigcup \{e_p\} \bigcup A_q \bigcup \{e_q\} \ldots)$. Figures 8.1 to 8.4 illustrate these arrangements. $N_{p,q,\ldots}$

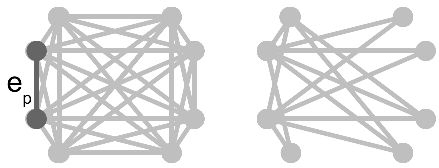


Figure 8.1: *(left) A complete graph with an edge e_p. (right) The set A_p.*

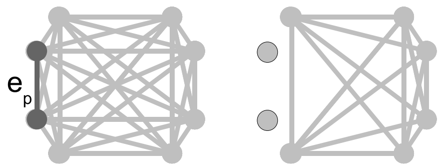


Figure 8.2: *(left) A complete graph with an edge e_p. (right) The set N_p.*

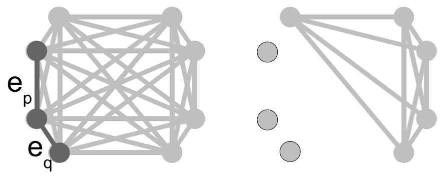


Figure 8.3: *(left) A complete graph with adjacent edges e_p, e_q. (right) The set $N_{p,q}$.*

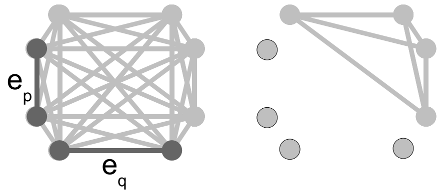


Figure 8.4: *(left) A complete graph with non-adjacent edges e_p, e_q. (right) The set $N_{p,q}$.*

Theorem 41 *Let C_Θ be the random variable, the cost of a tour in the solution space Θ of a TSP with $n > 5$ cities and with edge set E. Let $\mu = \mathrm{Ex}(C_\Theta)$. Then the third moment about the mean of C_Θ is*

$$\mathrm{Mm}_3(\mathrm{C}_\Theta) = \frac{2\gamma_1}{(n-1)} + \frac{2(\gamma_2 + 2\gamma_3)}{(n-1)(n-2)} + \frac{2(\gamma_4 + 2\gamma_5 + 4\gamma_6)}{(n-1)(n-2)(n-3)}$$

with the values $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6$ given by

$$\begin{aligned}
\gamma_1 &= \sum_{e \in E} c_0(e)^3 \\
\gamma_2 &= 3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in A_p} c_0(e_q) \\
\gamma_3 &= 3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in N_p} c_0(e_q) \\
\gamma_4 &= 3 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in A_q - (A_p \bigcup \{e_p\})} c_0(e_r) \\
\gamma_5 &= 3 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r) \\
\gamma_6 &= \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in N_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r)
\end{aligned}$$

where $c_0(e) = c(e) - \mu/n$.

Proof: Consider Equation 8.1. Each tour has only n edges, so for a given m, there are just n t_{mi} which are equal to 1, the remainder being equal to 0. Therefore Equation 8.1 is written

$$\mathrm{Mm}_3(\mathrm{C}_\Theta) = \frac{\sum_{m=1}^{|\Theta|}((t_{m1}c_0(e_1) + t_{m2}c_0(e_2)\dots t_{m|E|}c_0(e_{|E|}))^3)}{|\Theta|},$$

$$= \frac{\sum_{m=1}^{|\Theta|}\sum_{k=1}^{|E|}\sum_{j=1}^{|E|}\sum_{i=1}^{|E|} t_{mi}t_{mj}t_{mk}c_0(e_i)c_0(e_j)c_0(e_k)}{|\Theta|}.$$

The product $t_{mi}t_{mj}t_{mk} = 1$ if, and only if, tour m contains the edges e_i, e_j, e_k and there are six way in which this can occur:

case 1 All of e_i, e_j, e_k are equal. By Lemma 26 and Case 1 of Table 8.1 there are $(n-2)!$ tours containing the edge.

case 2 Two of e_i, e_j, e_k are equal and the third is adjacent. By Lemma 26 and Case 2 of Table 8.1 there are $(n-3)!$ tours containing the three edges so configured.

case 3 Two of e_i, e_j, e_k are equal and the third is non-adjacent to them. By Lemma 26 and Case 3 of Table 8.1 there are $2(n-3)!$ tours containing the 2 edges so configured.

case 4 The three edges e_i, e_j, e_k form a path. By Lemma 26 and Case 4 of Table 8.1 there are $(n-4)!$ tours containing the edges so configured.

case 5 Two of e_i, e_j, e_k are adjacent and the third is non-adjacent to either. By Lemma 26 and Case 5 of Table 8.1 there are $2(n-4)!$ tours containing the three edges so configured.

case 6 All e_i, e_j, e_k are all non-adjacent to each other. By Lemma 26 and Case 6 of Table 8.1 there are $4(n-4)!$ tours containing the three edges so configured.

For each of these six cases, we write the sum of edge cost products as γ_1 to γ_6 in (41). Upon collecting like terms we have:

$$\begin{aligned}\mathrm{Mm}_3(\mathrm{C}_\Theta) =&((n-2)!\gamma_1 + (n-3)!\gamma_2 + 2(n-3)!\gamma_3 \\ &+ (n-4)!\gamma_4 + 2(n-4)!\gamma_5 + 4(n-4)!\gamma_6)/|\Theta| \\ =&\frac{2((n-2)!\gamma_1 + (n-3)!(\gamma_2 + 2\gamma_3) + (n-4)!(\gamma_4 + 2\gamma_5 + 4\gamma_6))}{(n-1)!} \\ =&\frac{2\gamma_1}{(n-1)} + \frac{2(\gamma_2 + 2\gamma_3)}{(n-1)(n-2)} + \frac{2(\gamma_4 + 2\gamma_5 + 4\gamma_6)}{(n-1)(n-2)(n-3)}.\end{aligned}$$

as required. □

8.1.2 Reducing the Computational Complexity

The set A_p is $\mathcal{O}(n)$ in size, while the sets $E, N_p, N_{p,q}$ are all $\mathcal{O}(n^2)$ in size. Therefore the respective computational complexities of finding γ_1 to γ_6 are, at worst: $\mathcal{O}(n^2), \mathcal{O}(n^3), \mathcal{O}(n^4), \mathcal{O}(n^4), \mathcal{O}(n^5), \mathcal{O}(n^6)$. This implies that a naive application of Theorem 41 above would have complexity $\mathcal{O}(n^6)$, being that of the sum γ_6. Here we show that this can be reduced to $\mathcal{O}(n^4)$.

Let I_x be the set of edges incident to the vertex x and let $S_x = \sum_{e \in I_x} c_0(e)$, be the sum of edge costs incident to x. Now $|I_x| = n-1$, so the time complexity of pre-computing all the n values S_x is $\mathcal{O}(n^2)$ and the space complexity of saving them is $\mathcal{O}(n)$.

Lemma 42 *γ_2 can be found in $\mathcal{O}(n^2)$.*

Proof: Recall that $\gamma_2 = 3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in A_p} c_0(e_q)$. Consider the right most sum on A_p, in this expression. We show this can be found in constant time. Writing each edge e_p, as $e_p = \{p_1, p_2\}$ and noting that $A_p = (I_{p1} \bigcup I_{p2}) - \{e_p\}$ gives,

$$\begin{aligned}\gamma_2 &= 3 \sum_{e_p \in E} c_0(e_p)^2 (S_{p1} + S_{p2} - 2c_0(e_p)) \\ &= 6\gamma_1 + 3 \sum_{e_p \in E} c_0(e_p)^2 (S_{p1} + S_{p2}).\end{aligned}$$

This along with $|E| \in \mathcal{O}(n^2)$ implies the result. $\square$

Lemma 43 $\gamma_3 = -\gamma_2 - 3\gamma_1$.

Proof: Recall that $\gamma_3 = 3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in N_p} c_0(e_q)$. Consider the right most sum in this expression. $N_p = E - (A_p \bigcup \{e_p\})$. So

$$\sum_{e \in N_p} c_0(e) \quad = \sum_{e \in E} c_0(e) - \sum_{e \in A_p} c_0(e) - c_0(e_p),$$

but $\sum_{e \in E} c_0(e) = 0$ thus

$$\begin{aligned}
\gamma_3 &= 3 \sum_{e_p \in E} c_0(e_p)^2 \left[- \sum_{e_q \in A_p} c_0(e_q) - c_0(e_p) \right] \\
&= -3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in A_p} c_0(e_q) - 3 \sum_{e_p \in E} c_0(e_p)^2 c_0(e_p) \\
&= -\gamma_2 - 3\gamma_1.
\end{aligned}$$

as required. $\square$

Lemma 44 γ_4 *can be found in* $\mathcal{O}(n^3)$.

Recall that $\gamma_4 = 3 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in A_q - (A_p \bigcup \{e_p\})} c_0(e_r)$. We show that the right most sum can be found in constant time given an e_p and e_q. Let $e_p = \{s, p\}$, let $e_q = \{s, q\}$ be adjacent to it, sharing vertex s, and let $e_{pq} = \{p, q\}$ be adjacent to both. In addition, let I_q be the sets of edges incident to vertex q and let S_q be the pre-computed edge sum. Then $A_q - (A_p \bigcup \{e_p\}) = I_q - (\{e_q\} \bigcup \{e_{pq}\})$ and

$$\gamma_4 \quad = 3 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q)(S_q - c_0(e_q) - c_0(e_{pq})).$$

This along with $|E| \in \mathcal{O}(n^2)$ and $|A_p| \in \mathcal{O}(n)$ implies the result. $\square$

Lemma 45 γ_5 *can be found in* $\mathcal{O}(n^3)$.

Proof: Recall that $\gamma_5 = 3 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r)$. We show that the right most sum can be found in constant time.

Let $e_p = \{s, p\}$, let $e_q = \{s, q\}$ be adjacent to it, sharing vertex s, and let $e_{pq} = \{p, q\}$ be adjacent to both. In addition let I_s, I_p, I_q be the sets of edges incident to the vertices s, p, q respectively and let S_s, S_p, S_q be the pre-computed edge sums. Now $N_{p,q} = E - (I_s \bigcup I_p \bigcup I_q)$, but $\sum_{e \in E} c_0(e) = 0$ and the edges e_{pq}, e_q, e_p are each elements of two of I_s, I_p, I_q so, $\sum_{e_r \in N_{p,q}} c_0(e_r) = c_0(e_p) + c_0(e_q) + c_0(e_{pq}) - S_s - S_p - S_q$ and

$$
\begin{aligned}
\gamma_5 &= 3 \sum_{e_p \in E} \left[c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \left[c_0(e_p) + c_0(e_q) + c_0(e_{pq}) - S_s - S_p - S_q \right] \right] \\
&= 6\gamma_2 + 3 \sum_{e_p \in E} \left[c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \left[c_0(e_{pq}) - S_s - S_p - S_q \right] \right].
\end{aligned}
$$

This along with $|E| \in \mathcal{O}(n^2)$ and $|A_p| \in \mathcal{O}(n)$ implies the result. □

Lemma 46 γ_6 *can be found in* $\mathcal{O}(n^4)$.

Proof: Recall that $\gamma_6 = \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in N_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r)$. We show that the right most sum can be found in constant time. Let $e_p = \{p_1, p_2\}$ and let $e_q = \{q_1, q_2\}$ be non-adjacent. In addition let $I_{p1}, I_{p2}, I_{q1}, I_{q2}$ be the sets of edges incident to these vertices and let $S_{p1}, S_{p2}, S_{q1}, S_{q2}$ be the pre-computed edge sums. Now $N_{p,q} = E - (I_{p1} \bigcup I_{p2} \bigcup I_{q1} \bigcup I_{q2})$, but $\sum_{e \in E} c_0(e) = 0$ and the edges $e_p, e_q, \{p_1, q_1\}$, $\{p_1, q_2\}, \{p_2, q_1\}, \{p_2, q_2\}$ are each elements of two of $I_{p1}, I_{p2}, I_{q1}, I_{q2}$. Therefore write $S_{Np,q} = \sum_{e \in N_{p,q}} c_0(e) = -S_{p1} - S_{p2} - S_{q1} - S_{q2} + c_0(e_p) + c_0(e_q) + c_0(\{p_1, q_1\}) + c_0(\{p_1, q_2\}) + c_0(\{p_2, q_1\}) + c_0(\{p_2, q_2\})$ and

$$
\gamma_6 = \sum_{e_p \in E} \left[c_0(e_p) \sum_{e_q \in N_p} c_0(e_q) S_{Np,q} \right],
$$

as required. □

Theorem 47 *The complexity of computing the third moment about the mean of tour costs over the solution space of a TSP with n cities is $\mathcal{O}(n^4)$.*

Proof: This follows directly from Theorem 41, the comments at the beginning of Sect. 8.1.2, and Lemmas 42 to 46. □

The fact that γ_1, γ_2 and γ_3 can be computed in $\mathcal{O}(n^2)$ opens the possibility of developing fast estimates for the third moment.

8.2 Empirical Results

Here we apply our method of computing the third moment of tour costs of a solution to two simple empirical studies. Firstly we examine the relationship between skewness and problem size in various problem types. Secondly we consider the relationship between skewness and optimal tour cost in problems of fixed size.

8.2.1 Skewness Versus Problem Size

In this experiment we consider the relationship between the skewness of the cost distribution over the solution space and the size (in terms of the number of cities) of an instance. We examine five problem sets, two real world and three randomly generated. The five types are summarized in Table 8.2.

Table 8.2: *Problem types*

Problem Type	Cities	Cases	Description
VLSI	131-984	10	2D Euclid. Metric of TSPLIB [96]
RH Data	68-662	39	Non-Euclid. Genomics problems
Random Euclidean	10-1000	21	2D Euclid. Metric of TSPLIB
Random no embed.	10-1000	21	Rand. uniform int. edge costs
Random no embed.	4-29	25	Rand. uniform int. edge costs

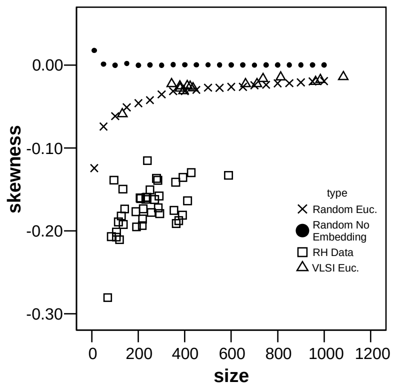


Figure 8.5: *The skewness versus the problem size in the first four problem sets of Table 8.2. The relationships suggests, that in the case of the non RH data sets, the skewness asymptotically approaches 0 with size. The RH data set is somewhat suggestive of convergence but to a lower limit point.*

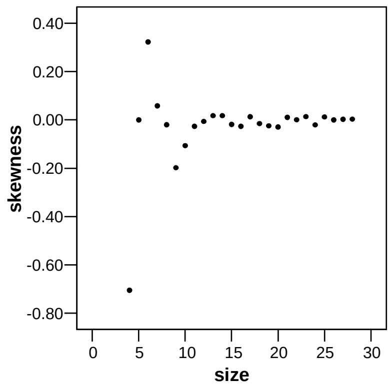


Figure 8.6: *The skewness versus the problem size in the last problem set of Table 8.2. Each of the 25 instances has random integer edge costs from a uniform distribution in [0,999]. The plot suggests that the skewness rapidly converges to 0 with an increase in instance size.*

Of the real world sets, the first set originated in the production of very large scale integrated circuits (VLSI) and uses the 2 dimensional Euclidean metric of [96]. The second set, of 39 instances, approximately obey the triangular inequality, but are non-Euclidean. They originate in the genomics community and arise from physical mapping of canine DNA by the radiation-hybrid (RH) method. The specific data set used was obtained from the RHDF9000 dog radiation hybrid panel [34]. In the case of the random problems, the Euclidean instances have approximate embedding in the plane, having vertex coordinates uniformly distributed in $[0, 999]^2$. The two non-Euclidean sets consists of instances with random integer edge costs with a uniform distribution on $[0, 999]$.

Results, Skewness Versus Size

The skewness of each instance was found using Theorems 33 and 41 in conjunction with Lemmas 42 to 46. Figure 8.5 shows its relationship to problem size. The relationship suggests, that in the case of the non RH data sets, the skewness asymptotically approaches 0 with size. The RH data set is somewhat suggestive of convergence but to a lower limit point. Figure 8.6 shows the pattern of convergence of the final problem set. This consists of 25 small random instances with integer *edge costs* from a uniform distribution in $[0, 999]$. The results suggest the distributions of cost become more symmetric with an increase in instance size in all cases.

8.2.2 Skewness Versus Optimal Tour Cost

In this experiment we consider the relationship between skewness and optimal tour cost. The three problem types considered are shown in Table 8.3. Each of the problem instances is normalised, such that, the expected value of tour costs is 0, and the standard deviation of tour costs is 1. In addition, each instance is of size 100. In the case of the Euclidean TSPLIB problems each problem consists of the first 100 cities in instances from TSPLIB. The original problems all have no fewer than 200 cities. The optimal tour cost for each instance was *estimated* using the Lin-Kernighan algorithm. We expect these to be close to the true optimal tour costs.

Results, Skewness Versus Cost

The plots of Figures 8.7, 8.8 and 8.9 show the correlations between the normalised optimal tour cost and the skewness. Table 8.4 shows the Spearman's non-parametric correlation coefficient between the skewness and normalised optimal tour cost. (We prefer this non-parametric statistic since it is reliable under a wide range of input data distributions). In each case a clear, statistically significant, correlation is apparent. Although, in the case of the truncated TSPLIB problems, the correlation is somewhat weak. The correlations indicate that for each of the three classes of problem considered, instances with high skewness (more positive) tend to have high optimal tour cost, with the converse also holding.

Table 8.3: *Problem Types*

Problem Type	Cities	Cases	Description
Random Euclidean	100	100	2D Euclid. Metric of TSPLIB [96].
Random no embed.	100	100	Rand. uniform int. edge costs.
TSPLIB Truncated	100	48	2D Euclid. Metric of TSPLIB.

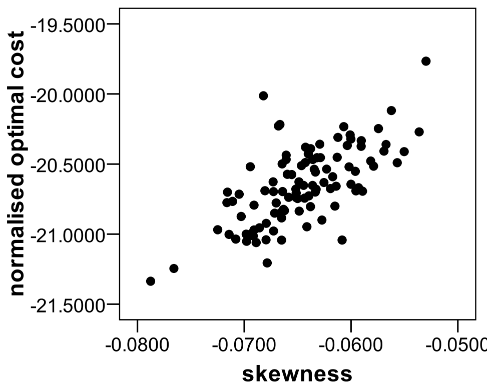


Figure 8.7: *The relationship between skewness and the cost of the optimal solution in 100 normalised random Euclidean problems. The Spearman's non-parametric correlation coefficient is 0.642* (< 0.001).

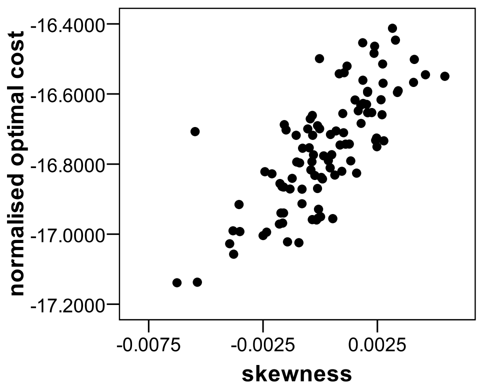


Figure 8.8: *The relationship between skewness and the cost of the optimal solution in 100 normalised random non-Euclidean problems. The Spearman's non-parametric correlation coefficient is 0.763* (< 0.001).

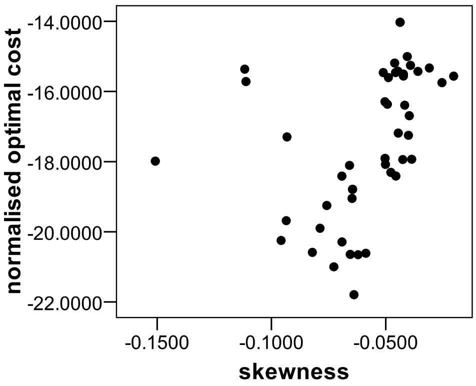


Figure 8.9: *The relationship between skewness and the cost of the optimal solution in 48 truncated (in size) and normalised TSPLIB Euclidean problems. Each problem consists of the first 100 cities in problems from TSPLIB. The Spearman's non-parametric correlation coefficient is 0.586* ($<$ 0.001)*.*

Table 8.4: *Spearman's non-parametric correlation coefficient*

Problem	Cases	Correlation	Significance
Random Euclidean	100	0.642	$< .001$
Random no embed.	100	0.763	$< .001$
TSPLIB Truncated	48	0.586	$< .001$

8.3 Conclusions and Future Research

In this chapter we provide a constructive proof that the third moment of tour costs over the solution space of an n city instance of the TSP can be computed in $\mathcal{O}(n^4)$. Our result is confined to the case of the TSP with fixed edge costs. This result together with the variance of costs over the solution space provides a tractable method to compute the skewness of the cost distribution.

Our $\mathcal{O}(n^4)$ method to compute the third moment relies in part on pre-computing and saving the sum of the edge costs incident to each vertex of an instance. It may be possible, by pre-computing and saving more complex sums of edge costs, to reduce the total computational complexity still further.

In an empirical study of four categories of problems we show that in three problem sets, the skewness asymptotically approaches 0 with size. The behaviour of the remaining set is less clear but is suggestive of convergence, but possibly, to a lower limit point. In all four classes of problems, the results suggest a positive correlation between instance size and tour cost distribution symmetry.

We also provide empirical evidence of a correlation between skewness and optimal tour cost. These two empirical results, together with the empirical results of the previous chapter, suggest that it may be possible to construct models to estimate the optimal tour cost based on the size, mean, standard deviation and skewness of an instance. We return to this topic in Section 11.2.

Chapter 9

The Fourth Moment of Costs over the Solution Space

Here we extend the results of the previous two chapters to consider the fourth moment about the mean of tour costs. We prove that this statistic can be computed in $\mathcal{O}(n^6)$ for an n city instance of the TSP. The theorem provided also suggests that this complexity can be improved. Computing the fourth moment gives a method to compute the kurtosis of the distribution of tour costs over the solution space.

While it is fair to say that at $\mathcal{O}(n^6)$ this complexity is too high to be of use in practical optimisation algorithms, it is a useful result in terms of providing tools to aid the analysis of the solution space. The provision of a method to *compute* the fourth moment is also a valuable tool to aid the development of fast random sampling techniques to *estimate* this moment. We consider an application of this approach in Section 11.1.3.

9.1 The Fourth Moment of Tour Costs

We begin by considering the number of tours (Hamiltonian cycles) containing various configurations of edges. Table 9.1 enumerates the eleven cases to be used.

Lemma 48 (Lemma 26 repeated.) *Given a complete graph K, let P be* *a set of m, non-cyclic, non-singleton paths over K sharing no vertices. Let k be the number of vertices not appearing in any path of P. Then there are $2^{m-1}(k+m-1)!$ Hamiltonian cycles in K containing all the paths in P.*

Proof: See Lemma 26 of Chapter 6. □

Table 9.1: *The eleven ways that, up to four unlabelled edges, can be arranged into paths in tours of size n. The $-$ character represents an edge, so $--$ means a path with 2 edges and three vertices. The $\leftrightsquigarrow$ symbol, the set (possibly empty) of free vertices between unconnected paths. The number of paths is given by m, while k is the number of free vertices.*

Case	Pattern	m	k	Num. Tours	Cities n
1	$- \leftrightsquigarrow$	1	$(n-2)$	$(n-2)!$	$n > 2$
2	$-- \leftrightsquigarrow$	1	$(n-3)$	$(n-3)!$	$n > 2$
3	$- \leftrightsquigarrow - \leftrightsquigarrow$	2	$(n-4)$	$2(n-3)!$	$n > 3$
4	$--- \leftrightsquigarrow$	1	$(n-4)$	$(n-4)!$	$n > 3$
5	$-- \leftrightsquigarrow - \leftrightsquigarrow$	2	$(n-5)$	$2(n-4)!$	$n > 4$
6	$- \leftrightsquigarrow - \leftrightsquigarrow - \leftrightsquigarrow$	3	$(n-6)$	$4(n-4)!$	$n > 5$
7	$---- \leftrightsquigarrow$	1	$(n-5)$	$(n-5)!$	$n > 4$
8	$--- \leftrightsquigarrow - \leftrightsquigarrow$	2	$(n-6)$	$2(n-5)!$	$n > 5$
9	$-- \leftrightsquigarrow -- \leftrightsquigarrow$	2	$(n-6)$	$2(n-5)!$	$n > 6$
10	$-- \leftrightsquigarrow - \leftrightsquigarrow - \leftrightsquigarrow$	3	$(n-7)$	$4(n-5)!$	$n > 6$
11	$- \leftrightsquigarrow - \leftrightsquigarrow - \leftrightsquigarrow - \leftrightsquigarrow$	4	$(n-8)$	$8(n-5)!$	$n > 7$

9.2 The Fourth Moment Theorem

Theorem 49 *Let C_Θ be the random variable, the cost of a tour in the solution space of a TSP with $n > 7$ cities and with edge set E. Let $\mu = \mathrm{Ex}(C_\Theta)$. Then the fourth moment about the mean of C_Θ is*

$$\begin{aligned}\mathrm{Mm}_4(C_\Theta) = \frac{2\delta_1}{(n-1)} &+ \frac{2(\delta_{2a}+\delta_{2b}+2\delta_{3a}+2\delta_{3b})}{(n-1)(n-2)} \\ &+ \frac{2(\delta_{4a}+\delta_{4b}+2\delta_{5a}+2\delta_{5b}+4\delta_6)}{(n-1)(n-2)(n-3)} \\ &+ \frac{2(\delta_7+2\delta_8+2\delta_9+4\delta_{10}+8\delta_{11})}{(n-1)(n-2)(n-3)(n-4)},\end{aligned}$$

with the values $\delta_1, \delta_{2a}, \delta_{2b}, \delta_{3a}, \delta_{3b}, \delta_{4a}, \delta_{4b}, \delta_{5a}, \delta_{5b}, \delta_6, \delta_7, \delta_8, \delta_9, \delta_{10}\delta_{11}$ *given by*

$$\delta_1 = \sum_{e \in E} c_0(e)^4$$

$$\delta_{2a} = 3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in A_p} c_0(e_q)^2$$

$$\delta_{2b} = 4 \sum_{e_p \in E} c_0(e_p)^3 \sum_{e_q \in A_p} c_0(e_q)$$

$$\delta_{3a} = 3 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in N_p} c_0(e_q)^2$$

$$\delta_{3b} = 4 \sum_{e_p \in E} c_0(e_p)^3 \sum_{e_q \in N_p} c_0(e_q)$$

$$\delta_{4a} = 12 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in A_p} c_0(e_q) \sum_{\substack{e_r \in A_q, \\ e_r \notin A_p, \\ e_r \neq e_p}} c_0(e_r)$$

$$\delta_{4b} = 6 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q)^2 \sum_{\substack{e_r \in A_q, \\ e_r \notin A_p, \\ e_r \neq e_p}} c_0(e_r)$$

$$\delta_{5a} = 12 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r)$$

$$\delta_{5b} = 6 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r)^2$$

$$\delta_6 = 6 \sum_{e_p \in E} c_0(e_p)^2 \sum_{e_q \in N_p} c_0(e_q) \sum_{e_r \in N_{p,q}} c_0(e_r)$$

$$\delta_7 = 12 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{\substack{e_r \in A_q, \\ e_r \notin A_p, \\ e_r \neq e_p}} c_0(e_r) \sum_{\substack{e_s \in A_r, \\ e_s \notin A_q, \\ e_s \notin A_p}} c_0(e_s)$$

$$\delta_8 = 12 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{\substack{e_r \in A_q, \\ e_r \notin A_p, \\ e_r \neq e_p}} c_0(e_r) \sum_{e_s \in N_{p,q,r}} c_0(e_s)$$

$$\delta_9 = 3 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in Np,q} c_0(e_r) \sum_{\substack{e_s \in A_r, \\ e_s \in N_{p,q}}} c_o(e_s)$$

$$\delta_{10} = 6 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in Np,q} c_0(e_r) \sum_{e_s \in N_{p,q,r}} c_0(e_s)$$

$$\delta_{11} = \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in N_p} c_0(e_q) \sum_{e_r \in Np,q} c_0(e_r) \sum_{e_s \in N_{p,q,r}} c_0(e_s)$$

where $c_0(e) = c(e) - \mu/n$.

Proof:

Specializing Equation 4.7 to $k = 4$, and proceeding as we did for the third moment we have

$$\mathrm{Mm}_4(C_\Theta) = \frac{\sum_{m=1}^{|\Theta|} \sum_{l=1}^{|E|} \sum_{k=1}^{|E|} \sum_{j=1}^{|E|} \sum_{i=1}^{|E|} t_{mi} t_{mj} t_{mk} t_{ml} c_0(e_i) c_0(e_j) c_0(e_k) c_0(e_l)}{|\Theta|}.$$

The product $t_{mi} t_{mj} t_{mk} t_{ml} = 1$ if, and only if, tour m contains the edges e_i, e_j, e_k, e_l and there are eleven ways in which this can occur.

case 1 All of e_i, e_j, e_k, e_l are equal. By Lemma 48 there are $(n-2)!$ tours containing the edge. The value δ_1 is the sum of terms in this case.

case 2 From e_i, e_j, e_k, e_l there are 2 distinct edges and they form a path. By Lemma 48 there are $(n-3)!$ tours containing the edges. The values δ_{2a}, δ_{2b} are the sums of terms in this case.

case 3 From e_i, e_j, e_k, e_l there are 2 distinct edges and they are non-adjacent. By Lemma 48 there are $2(n-3)!$ tours containing the edges. The values δ_{3a}, δ_{3b} are the sums of terms in this case.

case 4 From e_i, e_j, e_k, e_l there are 3 distinct edges and they form a path. By Lemma 48 there are $(n-4)!$ tours containing the edges. The values δ_{4a}, δ_{4b} are the sums of terms in this case.

case 5 From e_i, e_j, e_k, e_l there are 3 distinct edges two of which form a path, the third is non-adjacent. By Lemma 48 there are $2(n-4)!$ tours containing the edges. The values δ_{5a}, δ_{5b} are the sums of terms in this case.

case 6 From e_i, e_j, e_k, e_l there are 3 distinct edges. All are non-adjacent. By Lemma 48 there are $4(n-4)!$ tours containing the edges. The value δ_6 is the sum of terms in this case.

case 7 Each of e_i, e_j, e_k, e_l are distinct and form a path. By Lemma 48 there are $(n-5)!$ tours containing the edges. The value δ_7 is the sum of terms in this case.

case 8 Each of e_i, e_j, e_k, e_l are distinct, 3 form a path, the other is non-adjacent. By Lemma 48 there are $2(n-5)!$ tours containing the edges. The value δ_8 is the sum of terms in this case.

case 9 Each of e_i, e_j, e_k, e_l are distinct and form 2 non-adjacent paths of 2 edges. By Lemma 48 there are $2(n-5)!$ tours containing the edges. The value δ_9 is the sum of terms in this case.

case 10 Each of e_i, e_j, e_k, e_l are distinct. Two are adjacent. The remaining two are non-adjacent. By Lemma 48 there are $4(n-5)!$ tours containing the edges. The value δ_{10} is the sum of terms in this case.

case 11 Each of e_i, e_j, e_k, e_l are distinct. All are non-adjacent. By Lemma 48 there are $8(n-5)!$ tours containing the edges. The value δ_{11} is the sum of terms in this case.

For each of these cases we write the sum of edge cost products as δ_1 to δ_{11} in (49). Upon collecting like terms we have

$$
\begin{aligned}
\mathrm{Mm}_4(C_\Theta) &= \frac{(n-2)!\delta_1}{|\Theta|} \\
&+ \frac{(n-3)!\delta_{2a}+(n-3)!\delta_{2b}+2(n-3)!\delta_{3a}+2(n-3)!\delta_{3b}}{|\Theta|} \\
&+ \frac{(n-4)!\delta_{4a}+(n-4)!\delta_{4b}+2(n-4)!\delta_{5a}+2(n-4)!\delta_{5b}+4(n-4)!\delta_6}{|\Theta|} \\
&+ \frac{(n-5)!\delta_7+2(n-5)!\delta_8+2(n-5)!\delta_9+4(n-5)!\delta_{10}+8(n-5)!\delta_{11}}{|\Theta|}.
\end{aligned}
$$

Recall $|\Theta| = (n-1)!/2$ so upon cancelation we have the result. □

Lemma 50 *δ_{10} can be found in $\mathcal{O}(n^5)$.*

Proof: Recall that $\delta_{10} = 6 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in Np,q} c_0(e_r) \sum_{e_s \in N_{p,q,r}} c_0(e_s)$.

We prove that the right most sum can be found in constant time. Consider three edges e_p, e_q and e_r with e_p and e_q adjacent, sharing a vertex x and e_r non-adjacent to both e_p, e_q. Writing $e_p = \{p_1, x\}$,$e_q = \{q_1, x\}$ and $e_r = \{r_1, r_2\}$. Let $I_{p1}, I_x, I_{q1}, I_{r1}, I_{r2}$ be the sets of edges incident to these vertices and let $S_{p1}, S_x, S_{q1}, S_{r1}, S_{r2}$ be the pre-computed edge sums. So $S_{p1} = \sum\limits_{e \in I_{p1}} c_0(e)$ and similarly for the other four sums. Now writing the set $N_{p,q,r}$ in terms of E and the five incident sets we have

$$N_{p,q,r} = E - \left(I_{p1} \bigcup I_x \bigcup I_{q1} \bigcup I_{r1} \bigcup I_{r2} \right).$$

However $\sum\limits_{e \in E} c_0(e) = 0$ since the expected value of tour costs is 0. In addition the 10 edges e_p, e_q, e_r, $\{p_1, q_1\}$, $\{p_1, r_1\}$, $\{p_1, r_2\}$, $\{x, r_1\}$, $\{x, r_2\}$ $\{q_1, r_1\}$, $\{q_1, r_2\}$, are each elements of *two* of $I_{p1}, I_x, I_{q1}, I_{r1}, I_{r2}$. Therefore let $S_{Np,q,r}$ be the sum of the edge costs in $N_{p,q,r}$ we have

$$\begin{aligned} S_{Np,q,r} &= \sum_{e_s \in N_{p,q,r}} c_0(e_s) \\ &= -S_{p1} - S_x - S_{q1} - S_{r1} - S_{r2} \\ &+ c_0(e_p) + c_0(e_q) + c_0(e_r) + c_0(\{p_1, q_1\}) + c_0(\{p_1, r_1\}) + c_0(\{p_1, r_2\}) \\ &+ c_0(\{x, r_1\}) + c_0(\{x, r_2\}) + c_0(\{q_1, r_1\}) + c_0(\{q_1, r_2\}) \end{aligned}$$

and

$$\delta_{10} = 6 \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \sum_{e_r \in Np,q} c_0(e_r) S_{Np,q,r}$$

as required. □

Lemma 51 *δ_{11} can be found in $\mathcal{O}(n^6)$.*

Proof: Recall that $\delta_{11} = \sum\limits_{e_p \in E} c_0(e_p) \sum\limits_{e_q \in N_p} c_0(e_q) \sum\limits_{e_r \in Np,q} c_0(e_r) \sum\limits_{e_s \in N_{p,q,r}} c_0(e_s)$. We prove that the right most sum can be found in constant time. Consider three non-equal, non-adjacent edges e_p, e_q, e_r. Writing $e_p = \{p_1, p_2\}, e_q = \{q_1, q_2\}$ and $e_r = \{r_1, r_2\}$. Let $I_{p1}, I_{p2}, I_{q1}, I_{q2}, I_{r1}, I_{r2}$ be the sets of edges incident to these vertices and let $S_{p1}, S_{p2}, S_{q1}, S_{q2}, S_{r1}, S_{r2}$ be the pre-computed edge sums. So $S_{p1} = \sum_{e \in I_{p1}} c_0(e)$ and similarly for the other five sums. Now writing the set $N_{p,q,r}$ in terms of E and the six incident sets we have

$$N_{p,q,r} = E - \left(I_{p1} \bigcup I_{p2} \bigcup I_{q1} \bigcup I_{q2} \bigcup I_{r1} \bigcup I_{r2} \right).$$

However $\sum_{e \in E} c_0(e) = 0$ since the expected value of tour costs is 0. In addition the 15 edges e_p, e_q, e_r, $\{p_1, q_1\}$, $\{p_1, q_2\}$, $\{p_2, q_1\}$, $\{p_2, q_2\}$, $\{p_1, r_1\}$, $\{p_1, r_2\}$, $\{p_2, r_1\}$, $\{p_2, r_2\}$ $\{q_1, r_1\}$, $\{q_1, r_2\}$, $\{q_2, r_1\}$, and $\{q_2, r_2\}$ are each elements of *two* of $I_{p1}, I_{p2}, I_{q1}, I_{q2}, I_{r1}, I_{r2}$. Therefore let $S_{Np,q,r}$ be the sum of the edge costs in $N_{p,q,r}$ we have

$$\begin{aligned}
S_{Np,q,r} &= \sum_{e_s \in N_{p,q,r}} c_0(e_s) \\
&= -S_{p1} - S_{p2} - S_{q1} - S_{q2} - S_{r1} - S_{r2} \\
&+ c_0(e_p) + c_0(e_q) + c_0(e_r) \\
&+ c_0(\{p_1, q_1\}) + c_0(\{p_1, q_2\}) + c_0(\{p_2, q_1\}) + c_0(\{p_2, q_2\}) \\
&+ c_0(\{p_1, r_1\}) + c_0(\{p_1, r_2\}) + c_0(\{p_2, r_1\}) + c_0(\{p_2, r_2\}) \\
&+ c_0(\{q_1, r_1\}) + c_0(\{q_1, r_2\}) + c_0(\{q_2, r_1\}) + c_0(\{q_2, r_2\})
\end{aligned}$$

and

$$\delta_{11} = \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in N_p} c_0(e_q) \sum_{e_r \in Np,q} c_0(e_r) S_{Np,q,r}$$

as required. □

Theorem 52 *The complexity of computing the fourth moment about the mean of tour costs over the solution space of a TSP with n cities is $\mathcal{O}(n^6)$.*

Proof: This follows directly from Theorem 49 and Lemmas 50 and 51. □

9.3 Conclusions and Future Research

In this chapter we have proven that the fourth moment about the mean of tour costs can be computed in $\mathcal{O}(n^6)$ for an n city problem. Among other statistics this result allows the kurtosis of the distribution of tour costs to be found in $\mathcal{O}(n^6)$.

A naive implementation of our key theorem of this chapter, Theorem 49, would have a computational complexity of $\mathcal{O}(n^8)$. We reduced this complexity to $\mathcal{O}(n^6)$ by pre-computing and storing the sum of edges incident to the vertices (cities) of a problem. As were argued in the case of the third moment, it seems reasonable that by pre-computing and storing more complex sums of edges, we could reduce the time complexity of computing the fourth moment still further. We also view our method of *computing* the fourth moment as useful to the task of developing fast methods to *estimate* the fourth moment by random sampling. This may be the subject of future research.

In Section 11.1 of the final chapter of this thesis we provide preliminary results that apply the methods of this, and previous chapters, to the task of estimating the probability distribution of tour costs over the solution space of instances. In the next chapter we turn our attention to the statistical properties of the 2-opt landscape of an instance.

Chapter 10

Statistics over the 2-opt Landscape

In this chapter we consider the statistical properties of the 2-opt landscape. For an n city TSP with fixed edge costs we prove that the probability distribution of gains over the 2-opt landscape can be computed in $\mathcal{O}(n^4 \log(n))$. This result gives a tractable method to investigate the probabilistic properties of the 2-opt move. The methodology adopted implies that statistics such as the neutrality of the landscape can be computed in $\mathcal{O}(n^4)$. The results also suggest that the probability distribution of 3-opt, and other moves, can be similarly determined, but at higher computational complexity.

We relate the variance of gains over the 2-opt landscape to the variance of tour costs over the solution space, as computed in Chapter 7. This provides an $\mathcal{O}(n^2)$ algorithm to compute this landscape statistic. At this computational complexity the variance of gains over the 2-opt landscape could be used to inform search in practical optimisation algorithms.

10.0.1 The 2-opt Landscape of the TSP

We recall from Section 3.2.3 that the 2-opt move reverses a section of a tour. Thus a 2-opt operation on a tour, $\pi = (i_1, \ldots, i_n)$ produces its neighbour

$$\rho = (i_1, \ldots, i_{p-1}, i_q, i_{q-1}, \ldots, i_{p+1}, i_p, i_{q+1}, \ldots, i_n). \tag{10.1}$$

Since the tour graph is undirected, this is equivalent to replacing the two tour edges $\{i_{p-1}, i_p\}$ and $\{i_q, i_{q+1}\}$ with the two edges $\{i_{p-1}, i_q\}$ and $\{i_p, i_{q+1}\}$.

We recall from Section 3.2 that the 2-opt landscape of a TSP is the tuple $\mathfrak{L} = (\Theta, c, \mathfrak{N})$ with Θ the solution space of the problem, c the cost of a tour and $\mathfrak{N}$ a multi-set consisting of the neighbours of each tour under the 2-opt move. So $\mathfrak{N}(\pi)$ denotes the neighbours of a particular tour π. We view $\mathfrak{L}$ as forming a directed graph with tours as vertices and edges connecting neighbouring tours. More succinctly:

Definition 10.0.1 *A* 2-opt TSP landscape $\mathfrak{L}$ *is the triple* $\mathfrak{L} = (\Theta, c, \mathfrak{N})$ *where* $c : \Theta \to \mathbb{R}$ *is the tour cost, and* $(\Theta, \mathfrak{N})$ *is a directed graph with* Θ *the set of tours and* $\mathfrak{N}$ *the set of edges connecting tours by a 2-opt move.*

It is clear that each 2-opt move can be undone by a second application of the 2-opt. For this reason each edge in the 2-opt landscape of a TSP is accompanied by an edge with the same vertices but opposite orientation.

Size of the 2-opt Landscape

Here we characterize two key properties of the 2-opt landscape, the out degree of each vertex (that is, the number of neighbours of each tour) and the number of edges in the graph of the landscape, that is $|\mathfrak{N}|$. (Both of these results are simple and well known. We provide proofs here for completeness.)

Lemma 53 *Each tour in the 2-opt landscape of a TSP of size n cities has* $\frac{1}{2}n(n-3)$ *neighbours.*

Proof: A TSP with $n = 3$ clearly has no 2-opt neighbours. For all other problems, each 2-opt removes two non-adjacent edges of the n edges of a tour. There is only one choice of replacement edges that results in a different tour. There are n ways to choose the first edge removed, leaving $n - 3$ ways of choosing the second. There are two ways of ordering the resulting edge choices, implying the result. $\square$

Corollary 54 *The 2-opt landscape of a TSP of size n cities has* $\frac{1}{4}n!(n-3)$ *edges.*

Proof: There are $\frac{1}{2}(n-1)!$ tours in the landscape. Each tour has one directed edge *to* each of its $\frac{1}{2}n(n-3)$ neighbours implying the result. $\square$

10.1 The Probability Distribution of Gains

Let $\mathfrak{L}$ be the 2-opt landscape of a fixed costs TSP. Then the gain of an edge $l = \{\pi, \rho\}$ in $\mathfrak{L}$ is $\text{gain}(\{\pi, \rho\}) = c_\Theta(\pi) - c_\Theta(\rho)$. Let $G : \mathfrak{L} \to \mathbb{R}$ be the random variable, the gain of an edge in $\mathfrak{L}$. The goal of this chapter is to show that the probability mass function of G can be determined in polynomial time on the number of cities in the TSP. We first give a characterisation of the gross properties of the distribution.

G gain random variable

Lemma 55 *Let the random variable G be the gain of an edge in the 2-opt landscape of a TSP with fixed edge costs. Then the probability distribution of G is symmetric and* $\text{Ex}(G) = 0$.

Proof: For each edge $\{\pi, \rho\}$ in the landscape there is an edge $\{\rho, \pi\}$ with $\text{gain}(\{\pi, \rho\}) = -\text{gain}(\{\rho, \pi\})$. $\square$

10.1.1 Equivalent Edges

Our central results for this chapter rest on the observation that there are many edges in the *landscape* associated with the same two pairs of *tour* edges. This gives rise to the notion of equivalent edges in the landscape and of an equivalence class.

Definition 10.1.1 *Let $l_1 = \{\pi_1, \rho_1\}$ and $l_2 = \{\pi_2, \rho_2\}$ be two edges in $\mathfrak{L}$. We say l_1 and l_2 are equivalent if there are distinct edges $e_p, e_q, e_r, e_s \in E$ such that π_1 can be transformed to ρ_1 by replacing e_p, e_q with e_r, e_s and π_2 can be transformed to ρ_2 by replacing e_p, e_q with e_r, e_s. Equivalent edges are written $l_1 \sim l_2$.*

Lemma 56 *The size of each equivalence class of edges in the 2-opt landscape of an n city TSP is $(n-3)!$.*

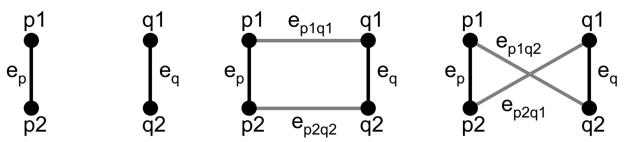
Proof: The total number of moves in the 2-opt landscape (the number of edges in the landscape) is $\frac{1}{4}n!(n-3)$. Each move deletes two non-adjacent edges from a tour. There are $\frac{1}{8}n(n-1)(n-2)(n-3)$ such *unordered* edge pairs in E. For any 2-opt move on a tour π, the two edges removed determine the two replacing them but for each tour π that can be transformed to a tour ρ by exchanging edges $\{x, y\}$ $\{u, v\}$ with $\{x, u\}$ $\{y, v\}$, there is a tour σ that can be transformed to a tour τ by exchanging edges $\{x, y\}$ $\{u, v\}$ with $\{x, v\}$ $\{y, u\}$. Giving $\frac{1}{2} \times \frac{8n!(n-3)}{4n(n-1)(n-2)(n-3)} = (n-3)!$, as required. $\square$

10.2 The Variance of Gains

In this section we consider the relationship between the variance of *tour costs* over the solution space of a fixed edge cost n city TSP and the variance of *gains* over its 2-opt landscape. We use this to provide an $\mathcal{O}(n^2)$ method to compute the variance of G. This is to say we will compute

$$\mathrm{Var}(G) = \frac{4}{n!(n-3)} \sum_{l=\{\pi,\rho\}\in\mathfrak{L}} (\mathrm{gain}(l))^2. \tag{10.2}$$

Figure 10.1: *The edges referred to in Theorem 57.* $e_p = \{p_1, p_2\}$, $e_q = \{q_1, q_2\}$, $e_{p1q2} = \{p_1, q_2\}$ *and* $e_{p2q1} = \{p_2, q_1\}$.



Theorem 57 *Let G be the gain of an edge in the 2-opt landscape of a TSP with fixed edge costs $c(e)$. Let $\mathrm{Var}(C_\Theta)$ be the variance of tour costs. Then*

$$\mathrm{Var}(G) = \frac{4(n-1)\mathrm{Var}(C_\Theta)}{n(n-3)}. \tag{10.3}$$

Proof: Let $[l]$ be the equivalence class containing l. By Lemma 56, the size of each equivalence class is $(n-3)!$ and by the definition of $[l]$, the gain of each edge in $[l]$ is identical and so we may usefully write gain$([l])$ to be the gain of each member of $[l]$.

$$\mathrm{Var}(G) = \frac{4}{n!(n-3)} \sum_{l=\{\pi,\rho\}\in\mathfrak{L}} (\mathrm{gain}(l))^2 \tag{10.4}$$

$$\mathrm{Var}(G) = \frac{4(n-3)!}{n!(n-3)} \sum_{[l]=\in\mathfrak{L}/\sim} (\mathrm{gain}([l]))^2 \tag{10.5}$$

$$\mathrm{Var}(G) = \frac{4\sum_{[l]=\in\mathfrak{L}/\sim}(\mathrm{gain}([l]))^2}{n(n-1)(n-2)(n-3)}. \tag{10.6}$$

We are now in a position to write this expectation in terms of the edges of E.

Let $e_p = \{p_1, p_2\}$ and $e_q = \{q_1, q_2\}$ be any two non-adjacent tour edges. Consider the gain of 2-opt moves that remove these two edges from a tour. There are just two ways to replace the two edges, either $\{p_1, q_1\}, \{p_2, q_2\}$ or $\{p_1, q_2\}, \{p_2, q_1\}$. For any particular tour for which $\{p_1, q_1\}, \{p_2, q_2\}$ is the correct pair of replacement edges (resulting in a valid tour) there is a tour for which $\{p_1, q_2\}, \{p_2, q_1\}$ the correct pair of replacement edges.

Write $e_{p1q1} = \{p_1, q_1\}, e_{p2q2} = \{p_2, q_2\}$, $e_{p1q2} = \{p_1, q_2\}$ and $e_{p2q1} = \{p_2, q_1\}$. Denote the two gains resulting from these two operations as $\mathrm{gain}_1(e_p, e_q)$ and $\mathrm{gain}_2(e_p, e_q)$. Figure 10.1 illustrates this arrangement.

$$\mathrm{gain}_1(e_p, e_q) = c(e_p) + c(e_q) - c(e_{p1q1}) - c(e_{p2q2})$$

$$\mathrm{gain}_2(e_p, e_q) = c(e_p) + c(e_q) - c(e_{p1q2}) - c(e_{p2q1}).$$

Let $c_0(e) = c(e) - \mathrm{Ex}(C_\Theta)/n$ then

$$\mathrm{gain}_1(e_p, e_q) = c_0(e_p) + c_0(e_q) - c_0(e_{p1q1}) - c_0(e_{p2q2})$$

$$\mathrm{gain}_2(e_p, e_q) = c_0(e_p) + c_0(e_q) - c_0(e_{p1q2}) - c_0(e_{p2q1}).$$

So Equation 10.6 becomes

$$\mathrm{Var}(G) = \frac{2\left(\sum_{e_p \in E} \sum_{e_q \in N_p} (\mathrm{gain}_1(e_p, e_q)^2 + \mathrm{gain}_2(e_p, e_q)^2)\right)}{n(n-1)(n-2)(n-3)}. \qquad (10.7)$$

Noting that in each inner loop of this expression we compute two gains and that we sum over all the *ordered* pairs of edges. Consider the sum of $\mathrm{gain}_1(e_p, e_q)^2$ and $\mathrm{gain}_2(e_p, e_q)^2$. For any non-adjacent edges e_p, e_q we have

$$\begin{aligned} \mathrm{gain}_1(e_p, e_q)^2 + \mathrm{gain}_2(e_p, e_q)^2 &= (c_0(e_p) + c_0(e_q) - c_0(e_{p1q1}) - c_0(e_{p2q2}))^2 \\ &+ (c_0(e_p) + c_0(e_q) - c_0(e_{p1q2}) - c_0(e_{p2q1}))^2 \end{aligned} \qquad (10.8)$$

$$\begin{aligned} &= 2c_0(e_p)^2 + 2c_0(e_q)^2 \\ &+ c_0(e_{p1q1})^2 + c_0(e_{p1q2})^2 + c_0(e_{p2q2})^2 + c_0(e_{p2q1})^2 \\ &- 2c_0(e_p)c_0(e_{p1q1}) - 2c_0(e_p)c_0(e_{p1q2}) \\ &- 2c_0(e_p)c_0(e_{p2q1}) - 2c_0(e_p)c_0(e_{p2q2}) \\ &- 2c_0(e_q)c_0(e_{p1q1}) - 2c_0(e_q)c_0(e_{p1q2}) \\ &- 2c_0(e_q)c_0(e_{p2q1}) - 2c_0(e_q)c_0(e_{p2q2}) \\ &+ 2c_0(e_{p1q2})c_0(e_{p2q1}) + 2c_0(e_{p1q1})c_0(e_{p2q2}) + 4c_0(e_p)c_0(e_q). \end{aligned} \qquad (10.9)$$

We note that each term on the first two lines of this expression contains a single edge. Each term on the next four lines of this expression contains two adjacent edges and finally each term on the last line contains two non-adjacent edges. Extending this to each possible non-adjacent edge pair e_p, e_q over E and separating the single edge terms, adjacent edge terms and non-adjacent edge terms, we have,

$$\mathrm{Var}(G) = \frac{8(n-2)(n-3)\beta_1 - 8(n-3)\beta_2 + 16\beta_3}{n(n-1)(n-2)(n-3)} \qquad (10.10)$$

where, as in Theorem 33, the functions $\beta_1, \beta_2, \beta_3$ are given by

$$\begin{aligned} \beta_1 &= \sum_{e_p \in E} c_0(e_p)^2 \\ \beta_2 &= \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in A_p} c_0(e_q) \\ \beta_3 &= \sum_{e_p \in E} c_0(e_p) \sum_{e_q \in N_p} c_0(e_q). \end{aligned} \tag{10.11}$$

$$\mathrm{Var}(G) = \frac{8\beta_1}{n(n-1)} - \frac{8\beta_2}{n(n-1)(n-2)} + \frac{16\beta_3}{n(n-1)(n-2)(n-3)}. \tag{10.12}$$

As we noted in Theorem 33, $\beta_3 = -\beta_1 - \beta_2$. Therefore,

$$\mathrm{Var}(G) = \frac{8(n-2)(n-3)\beta_1 - 8(n-3)\beta_2 + 16(-\beta_1 - \beta_2)}{n(n-1)(n-2)(n-3)}, \tag{10.13}$$

and

$$\mathrm{Var}(G) = \frac{8(n-4)\beta_1}{n(n-2)(n-3)} - \frac{8\beta_2}{n(n-2)(n-3)}. \tag{10.14}$$

Now by Theorem 33, we have

$$\mathrm{Var}(C_\Theta) = \frac{2\beta_1}{(n-1)} - \frac{4\beta_1 + 2\beta_2}{(n-1)(n-2)}. \tag{10.15}$$

Upon rearranging we have

$$\mathrm{Var}(C_\Theta) = \frac{2(n-4)\beta_1}{(n-1)(n-2)} - \frac{2\beta_2}{(n-1)(n-2)}. \tag{10.16}$$

Therefore

$$\mathrm{Var}(G) = \frac{4(n-1)\mathrm{Var}(C_\Theta)}{n(n-3)} \tag{10.17}$$

as required. □

10.3 Computing the Distribution of Gains

Theorem 58 [1] *Let the random variable G be the gain of an edge in the 2-opt landscape of a n city TSP with fixed edge costs. The probability mass function of G can be completely determined in $\mathcal{O}(n^4 log(n))$.*

Proof: Let l be an edge in the 2-opt landscape and let $[l]$ be the equivalence class containing l.

By Lemma 56, the size of each equivalence class is $(n-3)!$ and by the definition of $[l]$, the gain of each edge in $[l]$ is identical and so we may usefully write gain($[l]$) to be the gain of each member of $[l]$. We need to compute and record each gain($[l]$). But the number of equivalence classes $[l]$ is small since there are only $\frac{1}{8}n(n-1)(n-2)(n-3)$ unordered pairs of non-adjacent edge in E. Each gives rise to only two possible gains. So we need only compute each of the $\frac{1}{8}n(n-1)(n-2)(n-3)$ unordered edges in E, and for each compute and record, the two possible gains as a multiset M. Now $|M|$ is $\mathcal{O}(n^4)$ and converting $|M|$ to a set of ordered pairs of gains and multiplicity of that gain can be achieved in $\mathcal{O}(|M|(\log(|M|)) = \mathcal{O}(n^4 \log(n^4)) = \mathcal{O}(n^4 4 \log(n)) = \mathcal{O}(n^4 \log(n))$. □

10.4 Conclusions and Future Research

In this chapter we have shown that the probability distribution of gains over the 2-opt landscape can be computed in $\mathcal{O}(n^4 \log(n))$. This provides a tractable method to investigate the probabilistic properties of the 2-opt move. The method used implies that statistics such as the neutrality of the landscape can be computed, slightly faster, at $\mathcal{O}(n^4)$. The method also implies that the probability distribution of 3-opt, and other moves, can be similarly determined, but at higher computational complexity.

We have related the variance of gains over the 2-opt landscape to the variance of tour costs over the solution space. This provides an $\mathcal{O}(n^2)$ algo-

[1] A weaker preliminary version of Theorem 58 formed part of the author's honours thesis.

rithm to compute this landscape statistic. The relationship provided shows that for a given sized problem, the variance of gains is directly proportional to the variance of tour costs. Hence, it is impossible to construct instances with a relatively large spread of tour costs but low variation of gains and vice versa.

At a computational complexity of $\mathcal{O}(n^2)$ the variance of gains over the 2-opt landscape could be used to inform search in a practical optimisation algorithm. In Section 11.3 we return to these matters and show that in a wide range of landscapes the variance of gains relates to the root mean squared positive and negative gains over the landscapes. In that section we also provide empirical evidence that the variance of gains over the 2-opt landscape relates to the root mean squared gain encountered by iterative improvement to a local minimum.

Chapter 11

Future Research and Applications

In this chapter we conclude by discussing several promising areas for future research. Firstly, in Section 11.1 we provide initial results on estimating the probability distribution of tour costs of a TSP. Secondly, in Section 11.2 we examine *fast* methods to estimate the likely optimal tour cost using the mean and standard deviation of tour costs. Thirdly, in Section 11.3 we relate the standard deviation of gains over any *symmetric* landscape of a COP with the root mean square of positive gains over that landscape. In an empirical study of the 2-opt landscape of the TSP, we provide evidence that the variance of gains over the landscape is predictive of the mean squared gain encountered by iterative improvement from a randomly selected tour to a local minimum. Fourthly, in Section 11.4, we consider the possibility of constructing inner product spaces, with a point in such a space consisting of the vector of all tour costs of a given TSP. Fifthly, Section 11.5 shows how optimisation algorithms can be constructed using Theorem 31. Lastly, we conjecture that our results have application to other $\mathcal{NP}$-hard problems.

11.1 The Moment Problem

In this section we apply our method of computing the moments of tour costs over the solution space of a fixed edge cost instance, to estimate the probability distribution of tour costs of the instance. The general problem is referred to as the *moment problem* or the *inverse moment problem* or *power moment problem*. Writing Equation 4.8 as a Stieltjes Integral [51] we have

moment problem
inverse moment problem
power moment problem

$$\mu_i = \int_a^b x^i \mathrm{d}F(x), i = 0, 1, 2 \ldots \tag{11.1}$$

The problem is to find the probability mass (or density function) $F'(x)$ given a sequence of moments (about the origin) $\mu_0, \mu_1, \mu_2 \ldots$.

A number of cases are of interest, where $a = 0, b = 1$, the problem is known as the *Hausdorff moment problem*, where $a = -\infty, b = \infty$, the problem becomes the *Hamburger moment problem* and finally where $a = 0, b = \infty$, the problem is termed the *Stieltjes moment problem*. In each of these cases, if only a finite number of moments are known, the problem is termed the *truncated moment problem* or *reduced moment problem*. Extensive references include [1, 74, 76, 106].

moment problems of
Hausdorff
Hamburger
Stieltjes
truncated moment problem
reduced moment problem

By $\mathfrak{M}_{2r}$ we denote the set of distributions with moments about the origin $\mu_0, \mu_1, \ldots, \mu_{2r}$. Where $\mathfrak{M}_{2r}$ consists of a single distribution the problem is said to be *determined* and where it consists of more than one distribution the problem is *indeterminate*.

determined
indeterminate

Given the matrix

$$\mathbf{D}_k = \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_k \\ \mu_1 & \mu_2 & \cdots & \mu_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_k & \mu_{k+1} & \cdots & \mu_{2k} \end{pmatrix} \tag{11.2}$$

it is known that Hamburger moment problem has a solution, if and only if, $|\mathbf{D}_k| \geq 0$ for all $k > 0$. The cumulative distribution function has an infinite number of points of increase if $|\mathbf{D}_k| > 0$ for all $k > 0$. Finally the cumulative distribution function has $p + 1$ points of increase if, and only if, $|\mathbf{D}_k| > 0$

for $k = 0, \ldots, p$ and $|\mathbf{D}_k| = 0$ for $k > p$. If this is the case the problem is determined [106, Theorem 1.2].

11.1.1 An Algorithm to Solve the Moment Problem

Tari et al. [116], in a study of the Markov reward model, apply the classical methods detailed in [74] to compute bounds on the probability distribution function consistent with a given finite set of moments.

The method relies on the construction of a discrete probability distribution termed the *reference distribution*. Given moments, $\mu_0, \mu_1, \ldots, \mu_{2r}$ this distribution consists of just r points, $x_1, x_2 \ldots x_r$, with probability mass at each of these points denoted by $p_1, p_2 \ldots p_r$. The reference distribution is a *partial solution* of Equation 11.1 in the sense that

reference distribution

$$\mu_j = \sum_{i=1}^{r} p_i x_i^j, (j = 1, 2 \ldots 2r - 1). \tag{11.3}$$

The values $x_1, x_2 \ldots x_r$, are however not arbitrary, being chosen as the roots of the polynomial $\mathbf{M}_r$ defined as:

$$\mathbf{M}(x)_0 = 1,$$

$$\mathbf{M}(x)_i = \begin{vmatrix} \mu_0 & \mu_1 & \cdots & \mu_i \\ \mu_1 & \mu_2 & \cdots & \mu_{i+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{i-1} & \mu_i & \cdots & \mu_{2i-1} \\ 1 & x & \cdots & x^i \end{vmatrix}.$$

This arrangement, by exploiting the properties of orthogonal polynomials, allows the probability mass of an arbitrary point, c, to be approximated by a process of interpolation over the reference distribution. Tari et al. provide a simplified algorithm and argue that the probability mass of arbitrary point, c, can be estimated as

$$p = \frac{1}{(1, c, c^2 \dots c^r)\mathbf{D}_r^{-1}(1, c, c^2 \dots c^r)^T}.$$

In the cases where $c = x_i$ for each of the $x_1, x_2 \dots x_r$ the corresponding $p_i = p$. Tari et al. argue, that by the properties of orthogonal polynomials, the roots $x_1, x_2 \dots x_r$ are real and distinct.

The authors also give explicit closed form solutions for p in the case where the first five moments are known (including $\mu_0 = 1$). The experimental results in Section 11.1.2 of this chapter were computed by the use of these closed form solutions, in conjunction with the (simplified infinite case) algorithm of Tari et al. The moments were generated using Theorems 31, 33, 41 and 49 of this thesis along with their associated corollaries and lemmas.

11.1.2 Estimated Distributions

Figure 11.1 shows a comparison of the actual cumulative distribution function of a ten city perfect TSP, along with an estimate of the cumulative distribution function of the problem. Figure 11.2 shows this comparison for the case of a 14 city geometric problem.

In each case, the estimate of the cumulative distributions were constructed by first normalising the problems to have a mean of close to zero. Then computing moments 0 to 4 of the normalise problems using the results of the previous chapters. The, simplified infinite case, algorithm of Tari et al. was applied to compute an estimate of the probability mass at 1000 points. These points were within crude upper and lower bounds on the support of the normalised distribution. The resulting estimate of the probability mass function was then, offset by the original problem mean, integrated and scaled to a maximum of 1 to provide an estimate of the cumulative distribution function.

Figure 11.3 shows an estimate of the probability mass function of the corresponding to Figure 11.2. This was computed by differentiation of the cumulative distribution function.

Figure 11.1: *A comparison of an estimate of the cumulative distribution function and the actual cumulative distribution function of a ten city perfect problem. The estimate was generated by consideration of moments zero to four.*

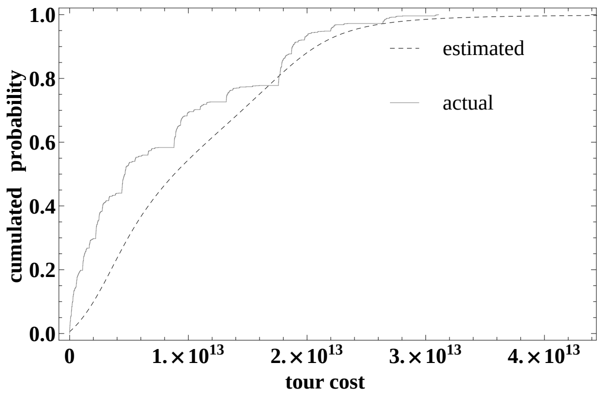


Figure 11.2: *A comparison of an estimate of the cumulative distribution function and the actual cumulative distribution function of a 14 city problem (burma14.tsp). The estimate was generated by consideration of moments zero to four.*

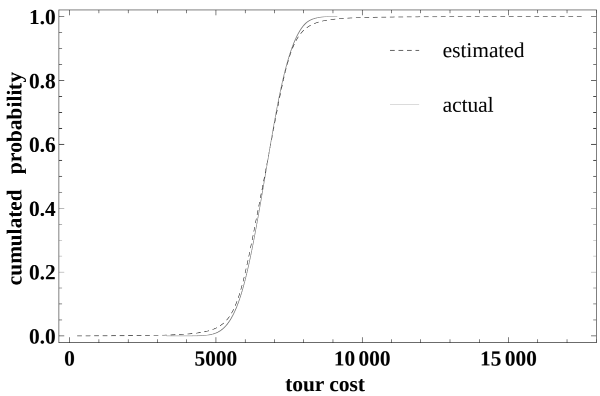


Figure 11.3: *A comparison of an estimate of the probability mass function and the actual probability mass function of a 14 city problem (burma14.tsp). The estimate was generated by consideration of moments zero to four.*

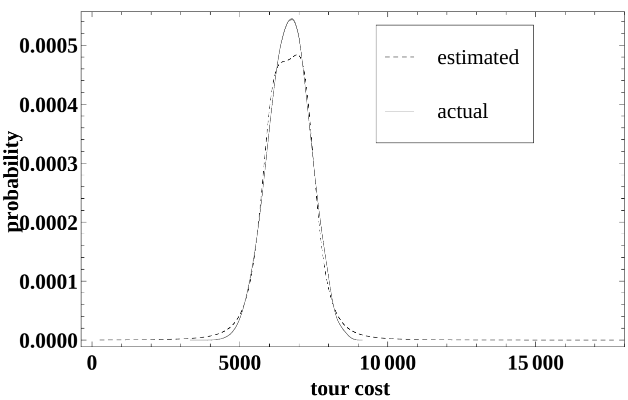


Figure 11.4 shows the estimated cumulative distribution function of a 68 city non-Euclidean problem. The estimate was generated by consideration of moments zero to four. The mean tour cost is 1191860.537 with a standard deviation of 26071.43349. The probable minimum tour cost provided by the chained Lin-Kernighan algorithm is 515370.

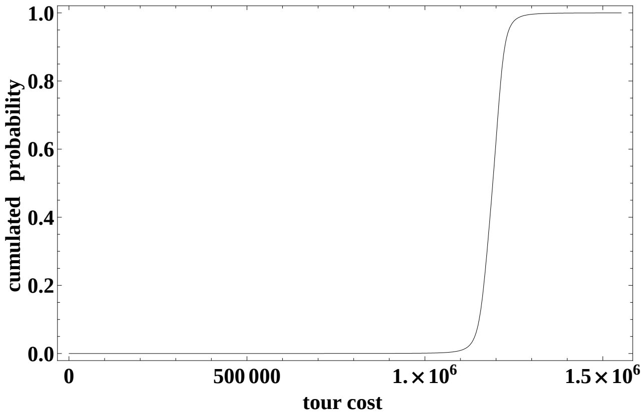


Figure 11.4: *the estimated cumulative distribution function of a 68 city non-Euclidean problem. The estimate was generated by consideration of moments zero to four.*

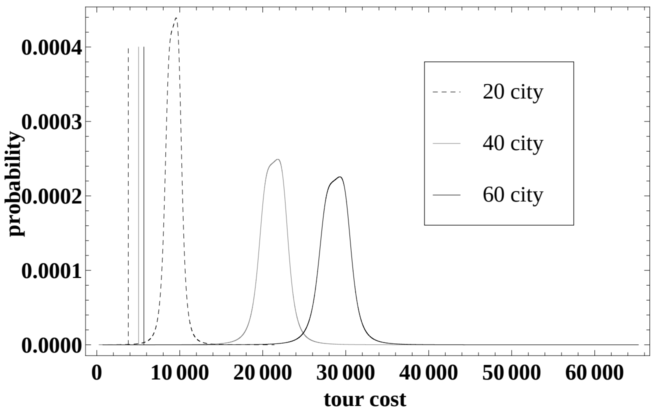


Figure 11.5: *The estimated probability mass functions of three random Euclidean problems along with the optimal tour cost of each problem. The vertical lines mark the optimal tour cost provided by the chained Lin-Kernighan algorithm.*

Figure 11.5 shows three Euclidean problems with vertex coordinates randomly generated from a uniform distribution on $[0, 999]$. The vertical markers in the figure indicate the optimal tour cost provided by the chained Lin-Kernighan algorithm. In each case the plots were generated by consideration of moments zero to four. The plots demonstrate how the minimum tour costs concentrate around a small constant with increase in problem size. This effect was predicted by the work of Beardwood, Halton and Hammersley [10] discussed in Section 4.3.1 of this thesis.

11.1.3 Moment Problem using Linear Programming

In contrast to the above, and most other, approaches to solving the moment problem, which rely on the machinery of real analysis, Chung and Sobel [21] construct a linear programming model to solve the problem.

As in Section 11.1 let $\mathfrak{M}_r$ be the set of distributions with moments $\mu_0, \mu_1, \ldots, \mu_r$. In the case that $\mathfrak{M}_r$ is indeterminate, Chung and Sobel estimate worst case bounds to the distributions in $\mathfrak{M}_r$, that is they estimate

$$L(r,x) = \inf(F(x) : F \in \mathfrak{M}_r) \text{ and } U(r,x) = \sup(F(x) : F \in \mathfrak{M}_r), \quad (11.4)$$

where x is in the support of the cumulative distribution function $F(x)$.

Here we concentrate on the lower bound, $L(r,x)$. The case of the upper bound, $U(r,x)$, has similar treatment. Without loss of generality, let $\{0, 1, 2 \ldots, d\}$ be a user defined discretization of the support of $F(x)$ and write $F(j)$ as F_j. Then the problem of finding $L(r,x)$ is formulated as

$$\begin{array}{lll}
L(r,x) = \text{Minimize } F(x) & & \\
\text{Subject to} & F_j \leq F_{j+1}, & j = 0, 1 \ldots d-1 \\
\text{and} & F_1 \geq 0, & \\
\text{and} & F_d = 1, & \\
\text{and} & \sum_{j=1}^{d} j^k (F_j - F_{j-1}) = \mu_k, & k = 1, 2 \ldots r.
\end{array} \quad (11.5)$$

Upon writing $f_j = F_j - F_{j-1}$ this becomes

$$\begin{array}{lll}
\text{Minimize } \sum_{j=0}^{x} f_j & & \\
\text{Subject to} & \sum_{j=0}^{d} f_j = 1, & \\
\text{and} & f_j \geq 0, & j = 0, 1 \ldots d \\
\text{and} & \sum_{j=1}^{d} j^k (f_j) = \mu_k, & k = 1, 2 \ldots r.
\end{array} \quad (11.6)$$

Under this arrangement each $L(r, x)$ for x in $\{0, 1, 2 \ldots, d\}$ can be computed by solving $d + 1$ linear equations. It is easy to see that the method can be modified to allow more general estimates of the support. Chung and Sobel make these ideas more precise and provide other formulations which are amenable to heuristics such as the simplex method.

In terms of the task of approximating distributions of an instance of the TSP, this construction and its variations are of great interest for two reasons. Firstly, the arrangement does not require the computation of the determinant of the Hankel matrices 11.2 which are known to be poorly conditioned [117]. This means it is attractive to *estimate* the moments (above the second or possibly third) by random sampling. Secondly, it is comparatively easy to add constraints to the formulation. This is not the case in more traditional approaches to the moment problem. Prime candidates for such additional constraints would be:

- upper bounds on the minimum tour cost provided by the Lin-Kernighan algorithm.
- lower bounds on the maximum tour cost provided by the Lin-Kernighan algorithm.
- lower bounds on the minimum tour cost provided by the spanning tree heuristics.

This may be the subject of future research.

11.2 Fast Estimates of the Minimum Tour Cost

The experimental results presented in Section 7.3.1 regarding the depth of the optimal tour cost suggest that this cost is related to the mean and standard deviation of tour costs. In this section we consider, more directly, the relationship between the optimal tour cost and the mean and standard deviation of tour costs. We start, in Section 11.2.1 by considering a statistic that incorporates information from both the mean and the standard deviation of tour costs. We show that this simple function correlates with the optimal tour cost in a wide range of problem types. In Section 11.2.2 we show, in a very constrained setting, that it is possible to estimate the optimal tour cost of an instance using the statistic. Furthermore, in this example, the estimates of optimal tour cost computed are superior to both the nearest neighbour algorithm and the nearest neighbour algorithm augmented with a 2-opt local search phase. Given that, for an n city instance we can compute both the mean and variance of tour costs in $\mathcal{O}(n^2)$, the results suggest that it could be possible to develop a fast ($\mathcal{O}(n^2)$) general estimator of optimal tour costs using these statistics.

11.2.1 Correlation between the Optimal Tour Cost and the Mean and Standard Deviation

Our ultimate aim is to use the mean and standard deviation along with any other pertinent information that can be obtained quickly to provide an estimate of the optimal tour cost. We first note that direct comparison of the optimal tour cost and the standard deviation of tour cost is not straight forward. In many problem types, the mean *positively correlates* with both the optimal tour cost and the standard deviation of tour costs. So that, a naive comparison of optimal tour cost versus standard deviation in these problem types would lead to the perplexing conclusion that an increase in standard deviation is related to an increase in optimal tour cost. In this situation the mean is said to be a *confounding factor* in the relationship between the optimal tour cost and standard deviation of tour costs.

confounding factor

Combining Information from the Mean and Standard Deviation

Our goal in this section is to investigate empirically the relationship between the optimal tour cost and the statistic $\mu - \sigma n^{\alpha}$, where n is the size of the instance in cities and α is a constant dependent on the problem type. This statistic combines information about the optimal tour cost from both the mean and standard deviation. It is motivated by two observations. Firstly, the empirical results of Sections 7.3.1 that suggest the number of standard deviations the optimal tour cost is below the mean is a near linear function of n^α with α dependent on the problem type. Secondly, in the absence of confounding factors, we would expect a *negative correlation* between the standard deviation and the optimal tour cost. So that, an increase in standard deviation is associated with a decease in optimal tour cost.

Study method

Eight problem types are considered in this study, these are summarised in Tables 11.1 and 11.2. These types form the twelve problem sets listed in Table 11.3. Four of the sets consist of a range of instance sizes with the remaining sets of fixed instance size. The set Combined data set 1 contains examples of three problem types.

Table 11.1: *Problem Types*

Problem type	Description
TSPLIB	TSPLIB [96]
	2D Euclidean Metric [96]
RH data	RHDF9000 dog radiation hybrid [34, 53]
	Obeys the triangular inequality
Truncated RH data	First 200 cities of the RH Data set
	Obeys the triangular inequality
Truncated TSPLIB	21 Modified TSPLIB instances with 200 cities
	2D Euclidean Metric [96]
Index data	Based on the Hamming distance between
	words in English text
	Obeys the triangular inequality
Random Euclidean	Uniform random coord. in (0,1000)(0,1000)
	2D Euclidean Metric
Random Euclidean four	Uniform random coordinates in $(0, 1000)(0, h)$
	with h one of 500,1000,1500,2000
	2D Euclidean Metric
Random no embedding	Uniform random integers edge costs in $[0, 999]$
Combined data set 1	See Table 11.2

Table 11.2: *Combined data set 1*

Problem type	Cities	Cases	Description
Truncated TSPLIB	200	21	2D Euclidean Metric of TSPLIB
Index data	200	10	Triangular inequality
Random Euclidean	200	15	2D Euclidean Metric of TSPLIB

Table 11.3:

Problem type	Instances	Cities n
Truncated RH data	26	200
Random no embedding	200	100
Truncated TSPLIB	21	200
Index Data	25	1000
Random Euclidean	100	200
Random Euclidean four	40	1000
Random no embedding	100	1000
Combined data set 1	46	200
TSPLIB	95	14-5934
RH data	39	68-588
Random Euclidean 2D	40	10-1000
Random no embedding	25	100-1200

For each instance in all of the problem sets, the Lin-Kernighan algorithm was applied to provide an estimate of the likely optimal tour cost and the mean and standard deviation were computed using the methods of this thesis. Table 11.4 provides Spearman's non-parametric correlation coefficient between the optimal tour cost discovered by the Lin-Kernighan algorithm and the statistic, $\mu - \sigma n^{\alpha}$, for each of the problem sets considered. For the purposes of this study, all problem sets except the RH Data sets were assigned an α of 0.5. The RH Data sets were assessed with both $\alpha = 0.5$ and $\alpha = 0.796$. These choices were informed by the observations of Section 7.3.1.

Results

Figures 11.6 to 11.14 show plots of relationships between $\mu - \sigma n^{\alpha}$ for nine of the problem sets listed in Table 11.4. In the case of the non-random instances and the set Random Euclidean Four, a clear statistically significant predictive correlation is apparent. In the case of the problem sets with a range of instance sizes, with the exception of the Random No Embedding set, we see quite strong, predictive, but non-linear correlation between $\mu - \sigma n^{\alpha}$ and the optimal tour cost. This suggests that a simple data transformation (or equivalently a refinement of our model statistic) could provide a more useful linear relationship.

Figure 11.10 shows the weak correlation between the optimal tour cost found by the Lin-Kernighan algorithm and $\mu - \sigma n^{0.5}$ for 200 instances of size 100 cities. The problem sets Random Euclidean (200 cities) and Random No Embedding (1000 cities) show no apparent correlation. However, as would be expected from the results of Wästlund and Beardwood, Halton and Hammersley discussed in Section 4.2, the range of optimal solution costs in these sets is narrow. The plot of the set Combined data set 1 in Figure 11.6 indicates that the statistic $\mu - \sigma n^{\alpha}$ is still of interest in the random Euclidean case. That is, when the random Euclidean set is viewed in the larger context of other Euclidean problems the statistic is somewhat predictive of the optimal tour costs. Figure 11.14 illustrates a similar phenomenon in the case of problems with random edge costs. As the size of these instances increases, the range of the optimal solution costs decreases. This type of problem represents a pathological case.

Table 11.4: *Spearman's non-parametric correlation coefficient between the optimal tour cost discovered by the Lin-Kernighan algorithm and the statistic $\mu - \sigma n^{\alpha}$. Where μ is the mean tour cost, σ is the standard deviation of tour costs and n is the number of cities of an instance. NS. indicates no significant correlation.*

Problem type	Instances	Cities n	α	Optimal cost to $\mu - \sigma n^{\alpha}$
Truncated RH data	26	200	0.796	0.775
Truncated RH data	26	200	0.5	0.394
Random no embedding	200	100	0.5	0.373
Truncated TSPLIB	21	200	0.5	0.986
Index Data	25	1000	0.5	0.745
Random Euclidean	100	200	0.5	NS.
Random Euclidean four	40	1000	0.5	0.947
Random no embedding	100	1000	0.5	NS.
Combined data set 1	46	200	0.5	0.906
TSPLIB	95	14-5934	0.5	0.934
RH data	39	68-588	0.796	0.504
RH data	39	68-588	0.5	0.881
Random Euclidean 2D	40	10-1000	0.5	0.963
Random no embedding	25	100-1200	0.5	NS.

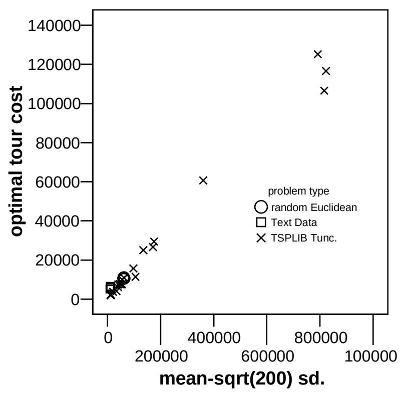


Figure 11.6: *The optimal tour cost versus the statistic $\mu - n^{0.5}\sigma$ in the Combined data set 1 of Table 11.4. The Spearman's non-parametric correlation coefficient is 0.792 (< 0.001). The plot suggests $\mu - n^{0.5}\sigma$ is predictive of the optimal tour cost.*

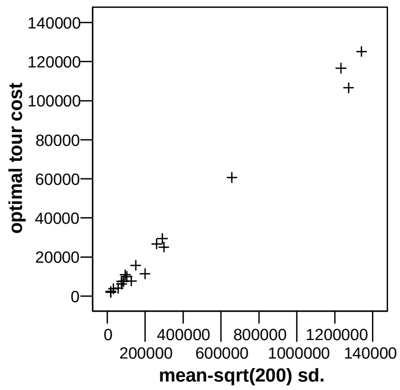


Figure 11.7: *The optimal tour cost versus the statistic $\mu - n^{0.5}\sigma$ in the set Truncated TSPLIB of Table 11.4. The Spearman's non-parametric correlation coefficient is 0.986* (< 0.001)*. The plot suggests $\mu - n^{0.5}\sigma$ is predictive of the optimal tour cost.*

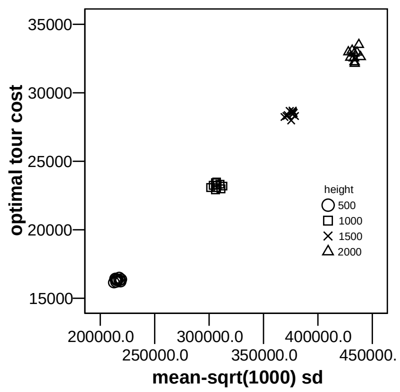


Figure 11.8: *The optimal tour cost versus the statistic $\mu - n^{0.5}\sigma$ in the set Random Euclidean Four of Table 11.4. This set consists of random Euclidean instances of width 1000 units and four possible heights. The Spearman's non-parametric correlation coefficient is 0.947* (< 0.001)*. The plot suggests $\mu - n^{0.5}\sigma$ is predictive of the optimal tour cost.*

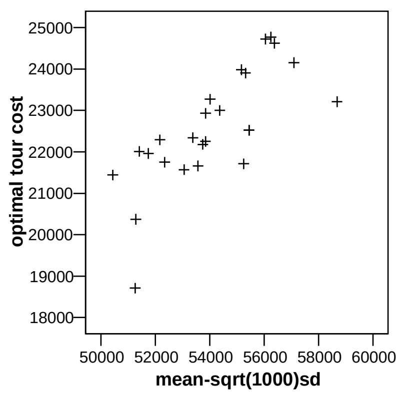


Figure 11.9: *The optimal tour cost versus the statistic $\mu - n^{0.5}\sigma$ in the set Index data of Table 11.4. This set consists of non-Euclidean TSPs which obey the triangular inequality. The Spearman's non-parametric correlation coefficient is 0.745* (< 0.001)*. The plot suggests $\mu - n^{0.5}\sigma$ is predictive of the optimal tour cost.*

Figure 11.10: *The optimal tour cost versus the* $\mu - n^{0.5}\sigma$ *in 200 fixed size TSPs. The problem set is the Random No Embedding set, of Table 11.4. The Spearman's non-parametric correlation coefficient is 0.373* (< 0.001)*. The plot suggests* $\mu - n^{0.5}\sigma$ *is weakly predictive of the optimal tour cost in this problem type.*

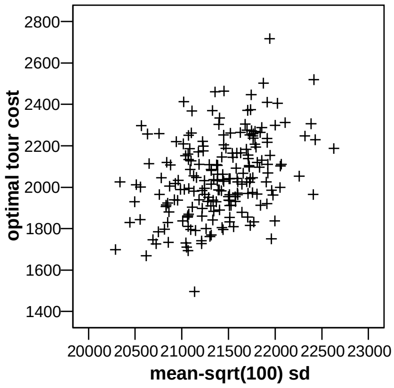


Figure 11.11: *The optimal tour cost versus the statistic* $\mu - n^{0.5}\sigma$*. The problem set is the RH data set, of Table 11.4. The instance sizes range from 68 to 588 cities. The Spearman's non-parametric correlation coefficient is 0.881* (< 0.001)*. The plot suggests* $\mu - n^{0.5}\sigma$ *is weakly predictive of the optimal tour cost.*

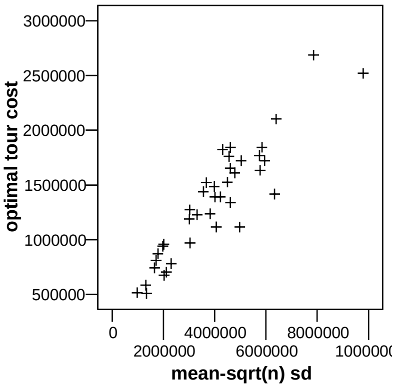
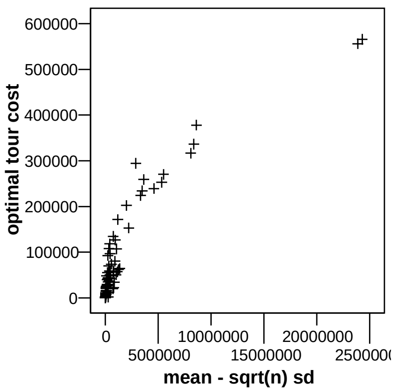


Figure 11.12: *The optimal tour cost versus the statistic* $\mu - n^{0.5}\sigma$*. The problem set is the TSPLIB set, of Table 11.4. The instance sizes range from 14 to 5934 cities. The Spearman's non-parametric correlation coefficient is 0.934* (< 0.001)*. The plot suggests* $\mu - n^{0.5}\sigma$ *is predictive of the optimal tour cost.*



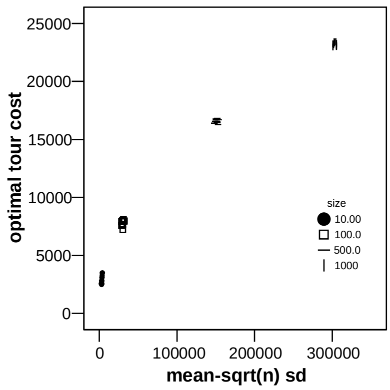


Figure 11.13: *The optimal tour cost versus the statistic $\mu - n^{0.5}\sigma$. The problem set consists of Random Euclidean instances with sizes from 10 to 1000 cities. The Spearman's non-parametric correlation coefficient is 0.963* (< 0.001). *The plot suggests $\mu - n^{0.5}\sigma$ is predictive of the optimal tour cost.*

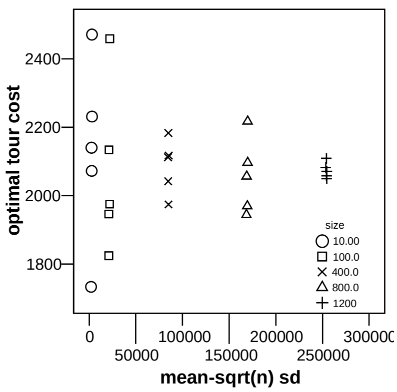


Figure 11.14: *The optimal tour cost versus the statistic $\mu - n^{0.5}\sigma$. The problem set consists of Random edge cost instances with sizes 100 to 1000 cities. No correlation is apparent, however the effect of an increase in problem size is clearly seen. As the number of cities increases the range of the optimal solution costs decreases. At each different instance size the range of the statistic $\mu - n^{0.5}\sigma$ is too small to be noticeable in this plot.*

Conclusions

In this section we have demonstrated that the optimal tour cost correlates in a predictive fashioned with the statistic $\mu - \sigma n^{\alpha}$ in a wide range of problem sets. Each of the correlations of Table 11.4 is stronger than both the correlation between the optimal tour cost and the mean and the optimal tour cost and the standard deviation. This indicates that the statistic $\mu - \sigma n^{\alpha}$ is combining information about the optimal cost from both μ and σ. This statistic is in fact not the best choice in every problem type, unsurprisingly, other linear combinations of μ and σ can be found that provide superior correlations in many of the problem sets we have considered. However our interest is in developing estimators of the optimal tour cost with general application. The rather pathological case of the Random No Embedding sets presents some difficulties in this task. However, we suspect that fast estimates of the degree of uniformity of edge costs and the degree to which the problem obeys the triangular inequality could be developed to detect these types of problems.

In the next section we provide a crude example of a statistical estimator for the optimal tour cost of a TSP.

11.2.2 An Example of Optimal Tour Cost Estimation using the Mean and Standard Deviation

It is instructive to consider just how much information about the optimal tour cost of a TSP is contained in the low order moments of its cost distribution. As before, let μ be the mean tour cost and σ the standard deviation of tour costs. In this section, we examine the predictive properties of a simple linear model to estimate the optimal tour cost of random edge cost instances of size $n = 100$ cities.

We recall from Figure 11.10 and Table 11.4 that this problem set had a poor correlation (0.375) between the optimal tour cost and the statistic $\mu - n^{0.5}\sigma$. We show, by consideration of a model provided by linear least squares regression, that even at this poor correlation, the mean and standard deviation can be predictive of the optimal tour cost. Table 11.5 and Figure 11.15 show the linear model used.

Table 11.5: *Curve fitting using linear least squares regression between the estimated optimal tour cost , $\bar{c}^*$, and the statistic $\mu - 10\sigma$, for 200 random integer edge cost instances each with 100 cities.*

Best fit	df.	r^2	F	Sig.
$\bar{c}^* = 0.164(\mu 10\sigma) - 1472.2$	198	0.139	31.97	< 0.001

This model was tested on a second, independently generated set of 100 random instances (with the same parameters as the model set). The Lin-Kernighan algorithm was applied to each instance in the test set to determine the probable optimal tour cost, this cost was compared to that predicted by the model. Figure 11.16 shows histograms of the estimation error between the cost predicted by the model and cost found by the Lin-Kernighan algorithm. The mean absolute error is 152, the maximum absolute error is 577. The mean relative error in the prediction is 7.5 %.

We contrast these results with the tour costs produced by the nearest neighbour algorithm (discussed in Section 3.1.4) and steepest descent from 100 random starts using the 2-opt move. Figure 11.17 shows the estimation error of these algorithms. The mean relative error in the prediction for the nearest neighbour algorithm is 132.8 % and for steepest descent this statistic is 38.9 %. For the nearest neighbour algorithm augmented by a single steepest descent search the results are identical to steepest descent from 100 random starts using the 2-opt move.

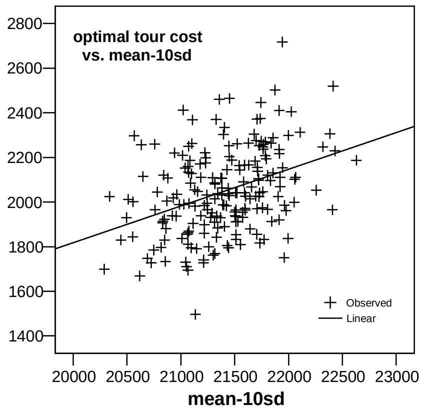


Figure 11.15: *A linear model based on 200 instances of the problem type Random No Embedding with 100 cities. The curve shows the model computed,* $\bar{c}^* = 0.164(\mu - 10\sigma) - 1472.2$, *where* $\bar{c}^*$ *is the estimated optimal tour cost* μ *is the mean tour cost and* σ *is the standard deviation of tour costs.*

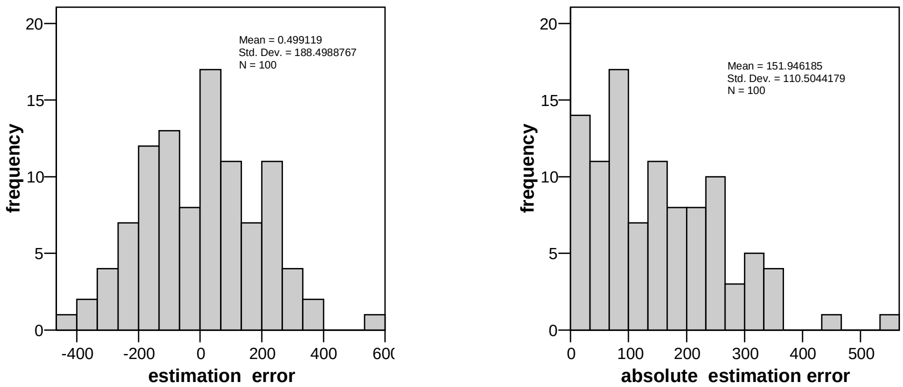


Figure 11.16: *Estimation errors for the linear model of Figure 11.15. (left) The histogram of estimation errors of the optimal cost of a random test set. (right) The histogram of the absolute value of estimation errors of the optimal cost of a random test set.*

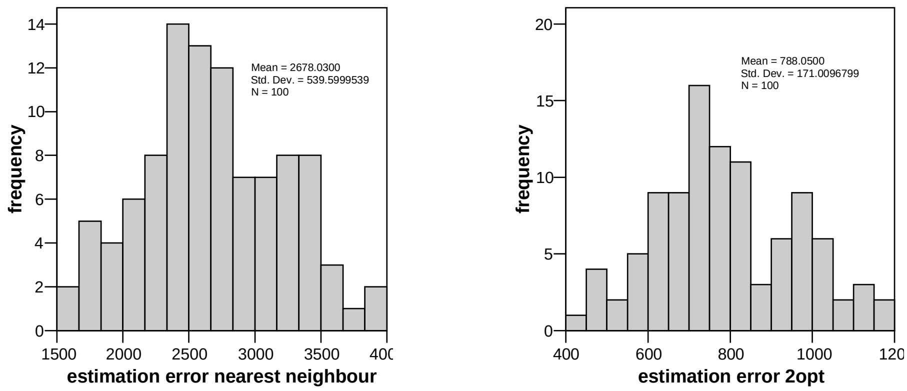


Figure 11.17: *Estimation errors for the nearest neighbour algorithm. The algorithm provides an upper bound for the optimal cost of each instance. (right) The histogram of the estimation errors for the nearest neighbour algorithm alone. (left) The histogram of the estimation errors for the nearest neighbour algorithm augmented by a 2-opt steepest descent phase. In this problem set, the results for the nearest neighbour algorithm augmented by a 2-opt steepest descent phase are identical to the 2-opt steepest descent from 100 random start tour.*

11.2.3 Conclusions

In this section we have demonstrated that the low order moments of the cost distribution of an instance of a TSP encapsulate sufficient information about the optimal cost of a solution to be useful in estimating this cost. Both the mean and standard deviation of tour costs of an n city problem can be computed in $\mathcal{O}(n^2)$. In addition, the results of Section 8.1.2 suggest that the third moment, and so the skewness of tour costs, can be approximated in $\mathcal{O}(n^2)$. The degree to which a problem obeys the triangular inequity and its other embedding properties also appear to be important factors. However, there are only $\mathcal{O}(n^2)$ triangles in the complete graph on n vertices, so the gross embedding properties of an instance can be characterised quickly if this proves necessary.

A fast predictor of the optimal tour cost of an instance would be beneficial to local search (and other methods) to aid the allocation of resources and the evaluation of candidate solutions. The existing $\mathcal{O}(n^2)$ algorithm, the nearest neighbour algorithm, provides a relatively poor estimate of the optimal tour cost of instances that do not obey the triangular inequality. Of course, the nearest neighbour algorithm produces a tour, and so an upper bound on the optimal tour cost. We make no claim to be able to do this with statistical methods alone. We envisage that an $\mathcal{O}(n^2)$ estimator would incorporate information provided by:

- the mean and standard deviation of tour costs.
- approximations of the higher moments of tour costs.
- Bender and Chekuri's [11] degree parameter, discussed in Section 2.3.2.
- upper bounds on the optimal tour cost provided by the nearest neighbour algorithm.
- the maximum and minimum edge costs.
- the size of the instance.

The development of such an estimator may be the subject of future research.

11.3 The Variance of Gains and Iterative Improvement in the 2-opt Landscape

We recall from Section 10.3, that the variance of gains $\mathrm{Var}(G)$ over the 2-opt landscape can be computed in $\mathcal{O}(n^2)$ for an n city instance of the TSP. In this section we show that this statistic is of interest in understanding the properties of iterative improvement on the 2-opt landscape. We view the results of this section as being useful both, to the understanding of *random walks* over those edges in the 2-opt landscape with positive gain, and to the practical matters of resource allocation and search strategy in methods such as Tabu search.

We begin with an obvious but very general result that relates the variance of gains over symmetric landscapes with the second moment about the origin of positive and negative gains over the landscapes.

Theorem 59 *Let the random variable G be the gain of an edge in a symmetric landscape of a COP. Let the random variables G_p be the gain of those edges with strictly positive gain. Similarly, let G_n, G_{np} and G_{nn} be respectively, the gain of those edges with, strictly negative, non-positive and non-negative gains. Then,*

$$\mathrm{Mo}_2(G_{np}) = \mathrm{Mo}_2(G_{nn}) \leq \mathrm{Var}(G) \leq \mathrm{Mo}_2(G_p) = \mathrm{Mo}_2(G_n). \tag{11.7}$$

Proof: Let SS be the sum of the squares of each gain over the landscape, similarly let the respective sums SS_p, SS_n, SS_{np} and SS_{nn} be the sum of the squares of the positive, negative, non-positive and non-negative gains over the landscape. Let m be the total number of edges in the 2-opt landscape and let m_p, m_n, m_{np} and m_{nn} be respectively the number of edges with positive, negative, non-positive and non-negative gains over the landscape. By the symmetry of the landscape, for each edge with a strictly positive gain, there is an edge with a strictly negative gain of the same absolute magnitude. Therefore we have, firstly $SS_p = SS_n = SS_{np} = SS_{nn} = \frac{1}{2}SS$, secondly $m_{np} = m_{nn} \geq \frac{1}{2}m \geq m_p = m_n$ and thirdly $\mathrm{Ex}(G) = 0$ which implies $\mathrm{Mo}_2(\mathrm{G}) = \mathrm{Var}(G)$. Which imply the result. $\square$

Corollary 60 *If a symmetric landscape of an instance of a COP has no moves with 0 gain, then,*

$$\mathrm{Mo}_2(G_{np}) = \mathrm{Mo}_2(G_{nn}) = \mathrm{Var}(G) = \mathrm{Mo}_2(G_p) = \mathrm{Mo}_2(G_n). \tag{11.8}$$

Proof: Obvious. □

We recall from Section 4.1.5 that the positive number $\sqrt{\mathrm{Mo}_2(X)}$ is termed the *root mean square* or *quadratic mean* and forms a biased measure of central tendency of the random variable $|X|$. Theorem 59 provides obvious bounds on the root mean square of G_p, G_n, G_{np} and G_{nn} in terms of the standard deviation of G.

Returning to the specific case of the TSP under the 2-opt landscape, the next section provides empirical evidence that $\mathrm{Var}(G)$ is predictive of the statistical properties of a descending walk from a randomly generated tour to a local minimum.

11.3.1 Iterative Improvement and the Variance of Gains

We recall from Section 3.3.1 that iterative improvement is a simple but effective search heuristic. In this section we provide the results of an empirical study of the relationship between the variance of gains in the 2-opt landscape and the mean squared gain encountered by iterative improvement.

Specifically, let the random variable D be the gain of the descending moves made by a single iterative improvement search from a randomly selected tour to a 2-opt local minima. At each move in the search, D is chosen as the first discovered improving move found in a randomized enumeration of the neighbourhood of the incumbent solution. (This strategy provides both lower cost local minima and faster run-time than iterative improvement using steepest descent.)

We examine the correlation between $\mathrm{Mo}_2(D)$ and $\mathrm{Var}(G)$ in five of the problem sets of Table 11.1. Details of the fifth set, Combined data set 2, are listed in Table 11.6. The Spearman's non-parametric correlation coefficient between these random variables is show in Table 11.7. Figures 11.18 to 11.20 illustrate the observed correlations in three of the problem sets. In all five cases the correlations are linear. The observed correlations suggest that the variance of the gains over the 2-opt landscape could be used as a predictor of the root mean square of the down hill gains encountered by iterative improvement.

Table 11.6: *Combined data set 2.*

Problem type	Cities n	Cases	Description
Random no embedding	1000	15	Uniform random edge costs
Index data	1000	25	Triangular inequality
Random Euclidean four	1000	40	10 instances of each height

Table 11.7: *Spearman's non-parametric correlation coefficient between* $\mathrm{Var}(G)$ *and* $\mathrm{Mo}_2(D)$. *Where* $\mathrm{Var}(G)$ *is the variance of gains over the 2-opt landscape and* $\mathrm{Mo}_2(D)$ *is the second moment about the origin of gains encountered on a single iterative improvement descent through the 2-opt landscape from a randomly generated tour to a local minima.*

Problem type	Instances	Cities n	$\mathrm{Var}(G)$ to $\mathrm{Mo}_2(D)$
TSPLIB	72	14-575	0.991
RH data	39	68-588	0.882
Random Euclidean four	40	1000	0.965
Index data	25	1000	0.579
Combined data set 2	80	1000	0.957

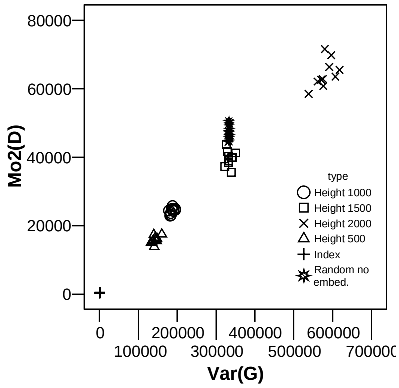


Figure 11.18: *The correlation between the variance of gains over the 2-opt landscape,* $\mathrm{Var}(G)$*, and the second moment about the origin of gains encountered on a single iterative improvement descent through the 2-opt landscape from a randomly generated tour to a local minima,* $\mathrm{Mo}_2(D)$*. The problem set is Combined data set 2 of Tables 11.6 and 11.7. The Spearman's non-parametric correlation coefficient is 0.957* (< 0.001).

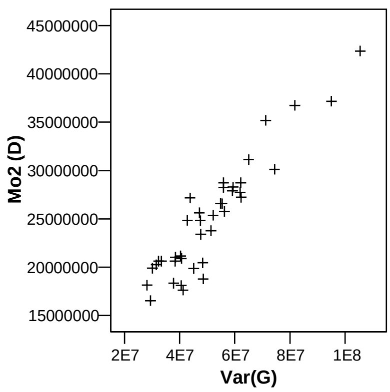


Figure 11.19: *The correlation between the variance of gains over the 2-opt landscape,* $\mathrm{Var}(G)$*, and the second moment about the origin of gains encountered on a single iterative improvement descent through the 2-opt landscape from a randomly generated tour to a local minima,* $\mathrm{Mo}_2(D)$*. The problem set is the RH data set of Table 11.7. The Spearman's non-parametric correlation coefficient is 0.882* (< 0.001).

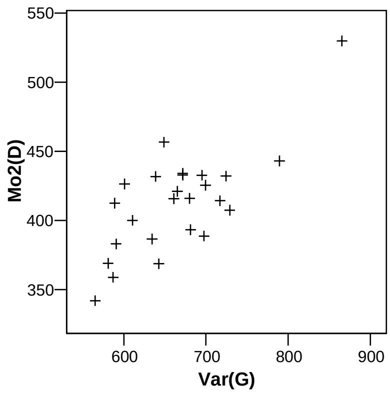


Figure 11.20: *The correlation between the variance of gains over the 2-opt landscape,* $\mathrm{Var}(G)$*, and the second moment about the origin of gains encountered on a single iterative improvement descent through the 2-opt landscape from a randomly generated tour to a local minima,* $\mathrm{Mo}_2(D)$*. The problem set is the Index Data set of Table 11.7. The Spearman's non-parametric correlation coefficient is 0.579* (< 0.001).

11.3.2 Conclusions

In this section we have shown that the variance of gains over a symmetric landscape of any COP closely relates to the mean squared positive and negative gains over the landscapes. This is not the case in asymmetric landscapes. In addition, in an empirical study of the properties of the 2-opt landscape of the TSP, we have demonstrated that the variance of gains over the landscape predictably relates to the mean squared gain of a realisation of a random walk from a randomly generated tour to a local minimum tour.

This result is useful to the understanding of random walks over the edges in the 2-opt landscape with positive gain, and to practical matters of resource allocation and search strategy in methods such as Tabu search. When coupled with an estimate of the expected number of moves of an iterate improvement search to a local 2-opt local minima the results could also provide a predictor of the expected cost of a 2-opt local minima.

11.4 The Covariance between Solution Spaces

Let X and Y be random variables with $\mathrm{Var}(X) > 0$ and $\mathrm{Var}(Y) > 0$. The covariance is

$$\mathrm{Cov}(X, Y) = \mathrm{Ex}((X - \mathrm{Ex}(X))(Y - \mathrm{Ex}(Y))). \tag{11.9}$$

In the case of two fixed edge cost n city TSPs, both normalised to have mean tour cost of 0, it seems reasonable that we *should* be able to compute the covariance of their tour costs, given that we can compute the variance of the tour costs. We envisage that a proof of this would simply follow that of the variance given in Section 7.1.

This would be an attractive result since the covariance is essentially a disguised inner product. So, such a result would facilitate the construction of inner product spaces, with the vector of tour costs of an instance of the TSP a point in the space [82].

Among other applications, this would be useful in characterising the effect of small changes in the edge costs of a instance on the minimum tour cost of the instance. It would also be valuable in understanding the properties of elementary landscapes, discussed in Section 4.5.1. These landscapes are of interest because of their strong algebraic properties.

11.5 Optimisation Algorithms

In Section 6.3 we noted that Theorem 31 can be employed as a tool to analyse certain construction heuristics. Here we exploit Theorem 31 to provide an optimisation method. Our approach is similar to the greedy expectation algorithm of Gutin and Yeo [48].

Algorithm 23 illustrates the procedure. It provides a nearest neighbour style heuristic with Theorem 31 used to compute the expected value of all tours containing a candidate path P'. A naive version of this arrangement has unacceptable computational complexity since $\mathrm{Ex}(\Theta|(P+e))$ would need to be computed for each candidate edge e. However it is not hard to see that an $\mathcal{O}(n^3)$ (for n cities) algorithm is achievable by avoiding certain repeated computations.

Algorithm 23 Nearest Neighbour Expected Value
Pseudo code nearest neighbour expected value heuristic.

Input: A graph $G = (V, E)$ of a TSP.
Output: A tour π

```
Let P ← ∅
for i = 1 to |V| − 2 do
    for all e adjacent to the end vertices of P do
        Find P′ ← P + e
        Such that Ex(Θ|P′) is minimum.
    end for
    Let P ← P′
end for
Add the final 2 edges to P and return it as π
```

We provide an example of the operation of the heuristic in Figures 11.21 and 11.22. The expected value of tour costs over the complete solution space of the problem shown in Figure 11.21 is 4975.2. The expected value of tour costs of those tours containing the black edges is 4338,4106.67,3825,3618.

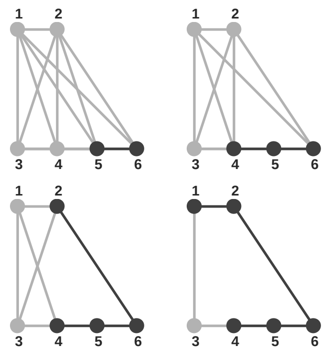


Figure 11.21: *The operation of the nearest neighbour expected value algorithm (Algorithm 23). The black edges are those added by the algorithm. The grey edges are those available to add. The algorithm operates by adding the edge to a partially constructed tour that will result in the lowest average cost of tours containing those edges. It is significant that the third edge added* $\{2, 6\}$ *is comparatively costly. The normal nearest neighbour algorithm would simply add a low cost edge at this point. Adding the edge* $\{2, 6\}$ *however results in the optimal tour.*

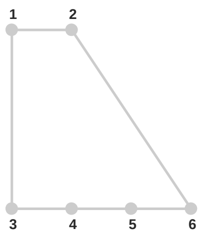


Figure 11.22: *The final tour produced by Algorithm 23. In this example the algorithm determines this tour after four iterations. The cost of the tour is 3618.*

In terms of tour cost, initial results suggest that Algorithm 23 provides superior results to the standard nearest neighbour approach and comparable results to the 2-opt iterative improvement heuristic. A greedy version of the approach, in which more than one path P is constructed, has also been tested. The approach, while giving better performance in terms of tour cost than the 2-opt iterative improvement heuristic has an uninviting computational complexity of $\mathcal{O}(n^5)$.

11.6 Generalisation to other Problems

It is easy to see that the out methods for computing the variance and other low order moments have generalisation to problems similar to the TSP such as the ATSP. We conjecture that the mean and variance of costs over the solution space of many other combinatorial optimisation problems can be found in polynomial time.

The $\mathcal{NP}$-complete optimisation problems have large solution spaces but small problem descriptions. This implies that, despite their size, the solution spaces of the problems have a poverty of information.

Our method to compute the variance of costs over the solution space of the TSP relied on firstly viewing the solution costs as *sums* of simpler entities, the edge costs. The proof of Theorem 33 followed by identifying three relationships between these smaller entities, the edges could be: equal, adjacent or non-adjacent. Counting the number of pairs of these simpler entities in each of the three classes provided the result. In the case of the TSP the distributive property of the arithmetic sum made this reasonable. We suspect that similar arguments can be made in the case of other problems.

11.7 Conclusion

In this thesis we have considered the statistical properties of the symmetric travelling salesman problem (TSP). We have proven, that in the case of the fixed edge cost problem on n cities, the variance of tour costs over the complete solution space can be computed in $\mathcal{O}(n^2)$. In addition the third central moment can be computed in $\mathcal{O}(n^4)$ and the fourth central moment can be computed in $\mathcal{O}(n^6)$.

We have demonstrated that the probability distribution of gains over the 2-opt landscape of an n city TSP can be computed in $\mathcal{O}(n^4 \log(n))$. This result provides a tractable algorithm to compute, among other statistics, the moments of gains over the landscape. The result also provides the 2-opt neutrality of an instance.

We have proven a relationship between the variance of tour costs over the solution space and the gains over the 2-opt landscape of a problem. This gives an $\mathcal{O}(n^2)$ method to compute the variance of gains over the landscape and shows that for a given sized problem, the variance of tour costs is directly proportional to the variance of gains. We have related the standard deviation of gains over any *symmetric* landscape of a COP to the root mean square of positive and negative gains over that landscapes.

In empirical studies we have shown that, firstly, the optimal tour cost predictively relates to the mean and standard deviation of tour costs over the solution space in a wide range of problem types. Secondly, the optimal tour cost correlates with the skewness of tour costs. Thirdly, the variance of gains over the 2-opt landscape is predictive of the mean squared gain encountered by iterative improvement from a randomly selected tour to a local minimum.

In the case of the stochastic TSP with *edge costs* defined as independently distributed random variables with (not necessarily identical) known mean and variance, we have provided an $\mathcal{O}(n^4)$ algorithm to compute the variance of tour costs. The closed form expression for the variance of tour costs is open to asymptotic analysis. In particular, it is of interest in understanding the behaviour of the variance of tour costs in comparison to the mean tour cost as the problem size approaches, in the limit, infinity.

Given a subgraph S of a tour in an n city TSP, we have provided an $\mathcal{O}(n^2)$ algorithm to compute the expected tour costs over the solution space of those tours containing S. This is useful in analysing and designing algorithms such as Gutin's greedy expectation heuristic and the well known branch and bound methods.

Finally, we have demonstrated, that the methods developed to compute the moments of a distribution of tour costs in this thesis can be applied to estimate the probability mass and cumulated distribution functions of tour costs.

Appendix TSPLIB Numeric results

Table 11.8: *TSPLIB problems a to f with size up to 2000 cities.*

Problem Name	Mean Tour Cost	Variance of Tour Costs	Variance of Gains in the 2-opt Landscape
a280.tsp	34106.16487	909104.7168	13080.98071
ali535.tsp	3531813.097	4401419540	33031523.21
att48.tsp	49882.08511	11713339.59	1019494.372
att532.tsp	512114.7119	111899300.2	844529.0221
bayg29.tsp	4736.642857	117032.8116	17384.18422
bays29.tsp	5975.428571	184060.8216	27340.5995
berlin52.tsp	29913.05882	2493999.522	199676.5708
bier127.tsp	628963.6508	383083764.7	12260237.33
brazil58.tsp	123636.7018	49295526.48	3523316.626
brg180.tsp	910175.1955	2254855700	50674095.46
burma14.tsp	6672.153846	503214.7199	169916.6587
ch130.tsp	46309.16279	2767774.852	86503.44177
ch150.tsp	53896.32215	3311592.292	89510.61251
d1291.tsp	1729066.38	388870534.1	1206737.011
d1655.tsp	2173702.831	513282566.9	1242063.986
d198.tsp	190630.132	52145267.96	1064244.267
d493.tsp	449566.8699	83717289.5	682020.2249
d657.tsp	854973.8872	197993781.5	1209127.958
dantzig42.tsp	3110.487805	45253.87668	4530.913172
dsj1000.tsp	556100178.5	6.211485335e+013	2.489578275e+011
eil101.tsp	3425.52	18993.39384	767.5649158
eil51.tsp	1652.2	7976.873469	651.7053488
eil76.tsp	2522.906667	12856.40606	695.1913877
fl1400.tsp	1688403.863	674410945.6	1929647.025
fl1577.tsp	1355219.937	202761429	514950.076
fl417.tsp	496026.6731	184302626.5	1776431.438
fri26.tsp	2693.2	43557.9	7283.929766

Table 11.9: *TSPLIB problems g to p with size up to 2000 cities.*

Problem Name	Mean Tour Cost	Variance of Tour Costs	Variance of Gains in the 2-opt Landscape
gil262.tsp	26703.44828	471674.941	7256.751429
gr120.tsp	52340.36975	5316724.742	180253.6309
gr137.tsp	645339.2206	768290299.2	22766637.04
gr17.tsp	4668.25	187134.8542	50321.97759
gr202.tsp	267163.5124	46759917.52	935244.8801
gr21.tsp	7641.6	422451.5137	89407.72776
gr229.tsp	1379534.535	1768458274	31163464.57
gr24.tsp	3542.521739	77071.11909	14068.53761
gr431.tsp	2480476.344	3942735781	36762503.76
gr48.tsp	21018.68085	1565662.384	136270.6149
gr666.tsp	5099224.087	7345916255	44252708
gr96.tsp	370711.5579	282957734.5	12043451.96
hk48.tsp	49097.19149	9432644.949	820989.4677
kroA100.tsp	171070.0404	67836155.2	2769393.552
kroA150.tsp	257602.953	102260630.2	2764051.501
kroA200.tsp	340233.9196	138784698.7	2803873.608
kroB100.tsp	168754.2222	65765191.11	2684846.977
kroB150.tsp	256742.1611	109007616.6	2946419.024
kroB200.tsp	332835.9598	126607390	2557854.885
kroC100.tsp	170055.3333	69323671.77	2830121.033
kroD100.tsp	163110.404	57435957.43	2344808.159
kroE100.tsp	173215.4343	71211824.85	2907204.396
lin105.tsp	123621.7308	35168344.77	1366016.006
lin318.tsp	587996.4858	200149017.6	2533582.453
nrw1379.tsp	1423590.894	312739255.7	908466.479
p654.tsp	2038124.021	2133478180	13088884.67
pa561.tsp	36591.225	347016.9157	2483.142274
pcb1173.tsp	1410049.52	321277145.9	1097447.017
pcb442.tsp	772610.7211	236382292.8	2148952.084
pr1002.tsp	6448477.083	7876277782	31505174.08
pr107.tsp	578253.5849	998324859.4	38038258.48
pr124.tsp	697295.3984	826264466.5	27094249.37
pr136.tsp	826062.6963	1019783073	30444651.68
pr144.tsp	812089.3287	922057159.7	25975999.57
pr152.tsp	1051054.464	1466057454	39098317.83
pr226.tsp	1695680.551	2684418431	47937945.7
pr264.tsp	1121589.848	1623698994	24790017.15
pr299.tsp	759641.4966	504710846.5	6797606.086
pr439.tsp	1904507.219	1651808226	15119684.08
pr76.tsp	574460.7467	696497526.3	37662086.86

Table 11.10: *TSPLIB problems r to w with size up to 2000 cities.*

Problem Name	Mean Tour Cost	Variance of Tour Costs	Variance of Gains in the 2-opt Landscape
rat195.tsp	22720.85567	620719.0878	12865.33152
rat575.tsp	113672.77	5194947.481	36265.1244
rat783.tsp	179477.8107	9437494.803	48335.59902
rat99.tsp	8414.591837	166754.6573	6877.927784
rd100.tsp	55566.16162	5519877.944	225347.5944
rd400.tsp	211549.3183	19562336.23	196608.8704
rl1304.tsp	9375921.383	1.25396553e+010	38524332.05
rl1323.tsp	9794930.638	1.392506405e+010	42165268.72
rl1889.tsp	14799934.22	2.385836207e+010	50574192.81
si1032.tsp	368483.5635	15085145.1	58583.1981
si175.tsp	48119.96552	572944.8263	13248.15944
si535.tsp	159017.0861	5192870.29	38971.15782
st70.tsp	3657.826087	32611.50684	1919.14198
swiss42.tsp	4835.073171	84849.55318	8495.315459
ts225.tsp	1593005.714	1926312492	34554073.92
tsp225.tsp	41304.94643	1401863.177	25146.53466
u1060.tsp	6758667.401	1.099356672e+010	41563653.49
u1432.tsp	3945929.064	2069348931	5788394.275
u159.tsp	449591.7468	287497255	7325361.439
u1817.tsp	2118825.983	526788251.7	1160966.548
u574.tsp	680713.7871	171789053.8	1201329.385
u724.tsp	872563.4053	225237458.1	1247857.734
ulysses16.tsp	13028.26667	1260810.748	363695.4081
ulysses22.tsp	16617.71429	1948653.833	391595.507
vm1084.tsp	8572042.547	1.461523474e+010	54030534.87
vm1748.tsp	14942296.09	2.470068653e+010	56588093.31
w100.56.tsp	43109.43434	3144355.623	128367.5079

Table 11.11: *TSPLIB problems a to k with size up to 600 cities.*

Problem Name	Mean Tour Cost	Variance of Tour Costs	Third Central Moment of Tour Costs
a280.tsp	34106.16487	909104.7168	-24168383.61
ali535.tsp	3531813.097	4401419540	-1.515114177e+013
att48.tsp	49882.08511	11713339.59	-4969221993
att532.tsp	512114.7119	111899300.2	-7.162195432e+010
bayg29.tsp	4736.642857	117032.8116	-4464513.888
bays29.tsp	5975.428571	184060.8216	-9073696.074
berlin52.tsp	29913.05882	2493999.522	-797981085.5
bier127.tsp	628963.6508	383083764.7	-8.593313184e+011
brazil58.tsp	123636.7018	49295526.48	-4.795489771e+010
brg180.tsp	910175.1955	2254855700	-4.926389246e+011
burma14.tsp	6672.153846	503214.7199	-22572220.71
ch130.tsp	46309.16279	2767774.852	-244948525.4
ch150.tsp	53896.32215	3311592.292	-312137320.1
d198.tsp	190630.132	52145267.96	-6.779137557e+010
d493.tsp	449566.8699	83717289.5	-3.198596505e+010
dantzig42.tsp	3110.487805	45253.87668	-1379208.83
eil101.tsp	3425.52	18993.39384	-154719.1972
eil51.tsp	1652.2	7976.873469	-61161.88616
eil76.tsp	2522.906667	12856.40606	-111577.7631
fl417.tsp	496026.6731	184302626.5	-5.230855078e+010
fri26.tsp	2693.2	43557.9	-1409740.736
gil262.tsp	26703.44828	471674.941	-12128404.79
gr120.tsp	52340.36975	5316724.742	-526014637.6
gr137.tsp	645339.2206	768290299.2	-9.448550321e+011
gr17.tsp	4668.25	187134.8542	-17047250.64
gr202.tsp	267163.5124	46759917.52	-2.553035682e+010
gr21.tsp	7641.6	422451.5137	-37836043.72
gr229.tsp	1379534.535	1768458274	-4.3104637e+012
gr24.tsp	3542.521739	77071.11909	-3075790.806
gr431.tsp	2480476.344	3942735781	-1.092038713e+013
gr48.tsp	21018.68085	1565662.384	-161736581.8
gr96.tsp	370711.5579	282957734.5	-2.506402028e+011
hk48.tsp	49097.19149	9432644.949	-3632175176
kroA100.tsp	171070.0404	67836155.2	-2.438844896e+010
kroA150.tsp	257602.953	102260630.2	-3.686544101e+010
kroA200.tsp	340233.9196	138784698.7	-4.637621433e+010
kroB100.tsp	168754.2222	65765191.11	-2.674870949e+010
kroB150.tsp	256742.1611	109007616.6	-3.368841886e+010
kroB200.tsp	332835.9598	126607390	-4.74606365e+010
kroC100.tsp	170055.3333	69323671.77	-2.232581552e+010
kroD100.tsp	163110.404	57435957.43	-2.298093115e+010
kroE100.tsp	173215.4343	71211824.85	-2.57529883e+010

Table 11.12: *TSPLIB problems l to w with size up to 600 cities.*

Problem Name	Mean Tour Cost	Variance of Tour Costs	Third Central Moment of Tour Costs
lin105.tsp	123621.7308	35168344.77	-8928949775
lin318.tsp	587996.4858	200149017.6	-9.403908355e+010
pa561.tsp	36591.225	347016.9157	-8146100.111
pcb442.tsp	772610.7211	236382292.8	-1.130045616e+011
pr107.tsp	578253.5849	998324859.4	-9.442151317e+011
pr124.tsp	697295.3984	826264466.5	-1.526776447e+012
pr136.tsp	826062.6963	1019783073	-1.386452627e+012
pr144.tsp	812089.3287	922057159.7	-1.47355593e+012
pr152.tsp	1051054.464	1466057454	-4.242807862e+012
pr226.tsp	1695680.551	2684418431	-6.259133081e+012
pr264.tsp	1121589.848	1623698994	-1.027298931e+012
pr299.tsp	759641.4966	504710846.5	-2.657661007e+011
pr439.tsp	1904507.219	1651808226	-3.239198807e+012
pr76.tsp	574460.7467	696497526.3	-1.422347645e+012
rat195.tsp	22720.85567	620719.0878	-15880308.91
rat575.tsp	113672.77	5194947.481	-233811810.6
rat99.tsp	8414.591837	166754.6573	-3163317.633
rd100.tsp	55566.16162	5519877.944	-768124659.8
rd400.tsp	211549.3183	19562336.23	-2635500504
si175.tsp	48119.96552	572944.8263	-31524519.43
si535.tsp	159017.0861	5192870.29	-179861709.7
st70.tsp	3657.826087	32611.50684	-426815.3812
swiss42.tsp	4835.073171	84849.55318	-3012708.664
ts225.tsp	1593005.714	1926312492	-3.51075631e+012
tsp225.tsp	41304.94643	1401863.177	-82459891.44
u159.tsp	449591.7468	287497255	-1.653035575e+011
u574.tsp	680713.7871	171789053.8	-4.996302927e+010
ulysses16.tsp	13028.26667	1260810.748	-284751981
ulysses22.tsp	16617.71429	1948653.833	-303523818.2
w100.56.tsp	43109.43434	3144355.623	-356163139.9

Table 11.13: *TSPLIB problems with size up to 60 cities.*

Problem Name	Variance of Tour Costs	Third Central Moment of Tour Costs	Fourth Central Moment of Tour Costs
att48.tsp	11713339.59	-4969221993	4.03367917e+014
bayg29.tsp	117032.8116	-4464513.888	4.019403518e+010
bays29.tsp	184060.8216	-9073696.074	9.961059323e+010
berlin52.tsp	2493999.522	-797981085.5	1.856171717e+013
brazil58.tsp	49295526.48	-4.795489771e+010	7.42351099e+015
burma14.tsp	503214.7199	-22572220.71	7.083099919e+011
dantzig42.tsp	45253.87668	-1379208.83	6019570451
eil51.tsp	7976.873469	-61161.88616	189186063.3
fri26.tsp	43557.9	-1409740.736	5525521976
gr17.tsp	187134.8542	-17047250.64	1.009148123e+011
gr21.tsp	422451.5137	-37836043.72	5.211806062e+011
gr24.tsp	77071.11909	-3075790.806	1.756583045e+010
gr48.tsp	1565662.384	-161736581.8	7.268064043e+012
hk48.tsp	9432644.949	-3632175176	2.624492603e+014
swiss42.tsp	84849.55318	-3012708.664	2.129530315e+010
ulysses16.tsp	1260810.748	-284751981	4.519302289e+012
ulysses22.tsp	1948653.833	-303523818.2	1.096111203e+013

Glossary

∇^2 The graph Laplacian operator.

$n!$ The factorial of a non-negative integer, n.

$\sim$ Equivalence elements a, b of a set X are written $a \sim b$.

$X/\sim$ The set of all equivalence classes in X under the equivalence relation $\sim$.

$\lceil x \rceil$ The smallest integer larger than or equal to x.

$*$ A solution s with global minimum cost is written s^*. The cost c of a global global minimum solution is written c^*.

$[a]$ The set of all elements of a set X equivalent to a.

$|S|$ The cardinality of a set S.

$\leq_{\mathcal{P}T}$ A polynomial time Turing reduction.

$\leq_{\mathcal{P}}$ A polynomial time reduction.

A_p The set of edges adjacent to edge e_p.

$\mathrm{b}(\pi, \pi')$ The bond distance between two tours π and π'.

c_0 For an edge e_p with cost $c(e_p)$ in an n city TSP $c_0(e_p) = c(e_p) - \mathrm{Ex}(C_\Theta)/n$.

c_Θ The cost of a tour in the solution space of a TSP.

$C, C_\Theta, \mathrm{Cost}, \mathrm{Cost}_\Theta$ The random variables, the cost of a tour.

$C(e)$ The random variable, the cost of a particular edge, e, in the set E.

C_E The random variable, the cost of an edge in the set E.

$\mathrm{Cov}(X, Y)$ The covariance of random variables X and Y.

$\mathrm{d}_k(\pi, \pi')$ The k-opt distance between two tours π and π'.

E_π The set of edges in a tour π.

$\mathrm{Ex}(X)$ The expected value of X.

G The gain as a random variable.

I_x The set of edges incident to vertex x.

$I1_P$ Given a set of paths, P, $I1_P$ is the set of edges each of which is incident to an end vertex of *one* path of P and not incident to any other path of P.

$I2_P$ Given a set of paths, P, $I2_P$ is the the set of edges incident to the end vertices of any *two* paths of P. So that an edge in $I2_P$ links two paths of P to form a single path.

K_n The complete graph on n vertices.

$\mathfrak{L}$ The combinatorial landscape of a problem.

$\mathrm{Mf_k}(X)$ The k th factorial moment of X.

$\mathrm{Mm_k}(X)$ The k th moment about the mean of X.

$\mathrm{Mo_k}(X)$ The k th moment about the origin of X.

$\mathfrak{N}(\pi)$ The neighbourhood of a tour π.

$N_{p,q,\dots}$ The set of edges neither adjacent to nor equal to edges $e_p, e_q, \dots,$

N_P Given a set of paths P, N_P is the set of edges non-adjacent nor equal to any edge in a path of P.

$\mathcal{O}(g(n))$ The class of all functions f to $\mathbb{R}$ such that there exist constants $c, n_o > 0$ with $0 \leq |f(n)| \leq c|g(n)|$ for all $n \geq n_o$.

$\theta(g(n))$ The class of all functions f to $\mathbb{R}$ such that there exist constants $c_1, c_2, n_o > 0$ with $0 \leq c_1|g(n)| \leq |f(n)| \leq c_2|g(n)|$ for all $n \geq n_o$.

Θ The solution space of a TSP.

$\mathcal{P}$ polynomial time An algorithm is polynomial time if its run time complexity is in $\mathcal{O}(P(n))$ for some polynomial $P(n)$, where n is the size of the input data to the algorithm.

polynomial bounded function A function $f(n)$ is polynomial bounded if it is in $\mathcal{O}(P(n))$ for some polynomial $P(n)$.

super-polynomial A function that is not polynomial bounded.

$\mathbb{Q}$ The set of all rational numbers.

$\mathbb{R}$ The set of all real numbers.

t_{mi} 1 if tour m contains edge i otherwise 0.

$\mathrm{Var}(X)$ The variance of X.

$\mathbb{Z}$ The set of all integer numbers.

Bibliography

[1] N. I. Akhiezer. *The Classical Moment Problem and Some Related Questions in Analysis*. Hafner Publishing Co., New York, NY ,USA, 1965.

[2] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics. Princeton University Press, NJ, USA, first edition, 2006.

[3] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[4] S. Arora. Approximation algorithms for geometric TSP. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and its Variations*. Kluwer Academic Publishers, New York, NY, USA, 2002.

[5] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag, Secaucus, NJ, USA, 1999.

[6] N. Balakrishnan and V. B Nevzorov. *A Primer on Statistical Distributions*. Wiley, NJ, USA, 2003.

[7] J. W. Barnes, B. Dimova, S. P. Dokov, and A. Solomon. The theory of elementary landscapes. *Applied Mathematics Letters*, 16(3):337–343, 2003.

[8] J. W. Barnes, S. P. Dokov, R. Acevedo, and A. Solomon. A note on distance matrices yielding elementary landscapes for the TSP. *Journal of Mathematical Chemistry*, 31(2):233–235, 2002.

[9] J. Basel III and T. R. Willemain. Random tours in the traveling salesman problem analysis and application. *Computational Optimization and Applications*, 20(2):211–217, 2001.

[10] J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.

[11] M. A. Bender and C. Chekuri. Performance guarantees for the TSP with a parameterized triangle inequality. In *Proceedings of the 6th Workshop on Discrete Algorithms (WADS):Vancouver, British Columbia, Canada*, pages 122–133, 1999.

[12] J. L. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing*, 4(4):387–411, 1992.

[13] K. Boese, A. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16(2):101–113, 1994.

[14] K. D. Boese. Cost versus distance in the traveling salesman problem. Technical Report TR-950018, University of California, Los Angeles, C.S. Department, University of California, Los Angeles USA, 2004.

[15] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. North-Holland, New York, NY ,USA, fifth edition, 1982.

[16] G. E. P Box. Evolutionary operation: a method of increasing industrial productivity. *Applied Statistics*, 46:81–101, 1957.

[17] R. M. Brady. Optimization strategies gleaned from biological evolution. *Nature*, 317:804–806, 1985.

[18] H. J. Bremermann and M. Rogson. An evolution-type search method for convex sets. Technical report, University of California, Berkeley, Department of Mathematics, 1964.

[19] T. N. Bui and M. Colpan. Solving geometric TSP with ants. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 271–272, New York, NY, USA, 2005. ACM.

[20] Y. N. Chong. *Heuristic algorithms for routing problems*. PhD thesis, Curtin University of Technology, 2001.

[21] K. Chung and M. J. Sobel. Linear programming solutions of the truncated moment problem. *Computers and operations research*, 18(5):477–485, 1991.

[22] K. L. Chung. *A Course in Probability Theory*. Academic Press, New York, NY, USA, third edition, 2001.

[23] M. I. Ciarleglio. *Modular Abstract Self-Learning Tabu Search (MASTS): Metaheuristics Theory and Practice*. PhD thesis, The University of Texas at Austin, 2008.

[24] G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[25] B. Codenotti and L. Margara. Traveling salesman problem and local search. *Applied Mathematics Letters*, 5(4):69–71, 1992.

[26] B. W. Colletti. *Group Theory and Metaheuristics*. PhD thesis, The University of Texas at Austin, 1999.

[27] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, second edition, 2001.

[28] G. A. Croes. A method for solving traveling salesman problems. *Operations Research*, 6(6):791–812, 1958.

[29] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large scale traveling salesman problem. *Operations Research*, 2:393–410, 1954.

[30] S. de Givry, M. Bouchez, P. Chabrier, D. Milan, and T. Schiex. Carthagene: multipopulation integrated genetic and radiated hybrid mapping. *Bioinformatics*, 21(8):1703–1704, 2005.

[31] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3(2):159–168, 1990.

[32] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, New York, USA, third edition, 2005.

[33] B. Dimova, J. W. Barnes, E. Popova, and B. Colletti. Some additional properties of elementary landscapes. *Applied Mathematics Letters*, 2008 (in press).

[34] T. Faraut, S. de Givry, P. Chabrier, T. Derrien, F. Galibert, C. Hitte, and T. Schiex. A comparative genome approach to marker ordering. *Bioinformatics*, 23:e50–e56, 2007.

[35] M. M. Flood. The traveling salesman problem. *Operations Research*, 4:61–75, 1956.

[36] A. Frieze. On random symmetric travelling salesman problems. *Mathematics of Operations Research*, 29(4):878–890, 2004.

[37] A. Frieze and J. E. Yukich. Probabilistic analysis of the TSP. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 256–307. Kluwer Academic Publishers, New York, NY, USA, 2002.

[38] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, 38(4):815–853, 1991.

[39] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[40] F. Glover. Tabu search part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[41] F. Glover. Tabu search - part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[42] F. Glover and F. Laguna. *Tabu Search.* Kluwer Academic Publisher, Norwell, MA, USA, 1997.

[43] B. L. Golden. Closed-form statistical estimates of optimal solution values to combinatorial problems. In *ACM 78: Proceedings of the 1978 annual conference*, pages 467–470, New York, NY, USA, 1978. ACM.

[44] O. Gómez and B. Barán. Relationship between genetic algorithms and ant colony optimization algorithms. In M. Solar, D. Fernández-Baca, and E. Cuadros-Vargas, editors, *Proceedings of CLEI2004. Latin-American Conference on Informatics*, pages 766–776, Arequipa, Perú, 2004. Sociedad Peruana de Computacin.

[45] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pages 160–168, Pittsburgh, PA, 1985.

[46] L. K. Grover. Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12:235–243, 1992.

[47] G. Gutin and A. Punnen. *The Traveling Salesman Problem and Its Variations.* Kluwer Academic Publishers, New York, NY, USA, 2002.

[48] G. Gutin and A. Yeo. Polynomial approximation algorithms for the TSP and the QAP with a factorial domination number. *Discrete Applied Mathematics*, 119(1-2):107–116, 2002.

[49] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, 1988.

[50] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operations Research*, 12:106–130, 2000.

[51] P. Henrici. *Applied and Computational Complex Analysis*, volume II. John Wiley & Sons, Inc., New York, NY, USA, 1977.

[52] F. S. Hillier. Quantitative tools for plant-layout analysis. *Journal of Industrial Engineering*, 14(1):33–44, 1963.

[53] C. Hitte, J. Madeoy, E. F. Kirkness, C. Priat, T. D. Lorentzen, F. Senger, D. Thomas, T. Derrien, C. Ramirez, C. Scott, G. Evanno, B. Pullar, E. Cadieu, V. Oza, K. Lourgant, D. B. Jaffe, S. Tacher, S. Dreano, N. Berkova, C. Andre, P. Deloukas, C. Fraser, K. Lindblad-Toh, E. A. Ostrander, and F. Galibert. Facilitating genome navigation, survey sequencing and dense radiation-hybrid gene mapping. *Nature reviews. Genetics*, 6(8):643–648, 2005.

[54] J. H. Holland. Outline for a logical theory of adaptive systems. *Journal of the ACM*, 9(3):297–314, 1962.

[55] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, USA, 1975.

[56] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:1–25, 1985.

[57] ICSC. Best known solutions for symmetric TSPs. `http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSPLIB95/STSP.html`, March 17 2006.

[58] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. MATLAB Curriculum Series. Prentice Hall, Upper Saddle River, NJ, USA, 1997.

[59] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume I, pages 67–161. MIT Press, Cambridge, MA, USA, 1990.

[60] D. S. Johnson, L. A., McGeoch, and E. E. Rothberg. Asymptotic experimental analysis for the Held-Karp traveling salesman bound. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 341–350, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[61] D. S. Johnson and L. A. McGeoch. The Traveling Salesman Problem: A Case Study. In E. H. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization.* Wiley and Sons, New York, NY, USA, 1997.

[62] D. S. Johnson and L. A. McGeoch. Experimental analysis of heuristics for the STSP. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 369–443. Kluwer Academic Publishers, New York, NY, USA, 2002.

[63] D. S. Johnson and C. H. Papadimitriou. Computational complexity. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, pages 37–85. John Wiley & Sons, New York, NY, USA, 1985.

[64] D. S. Johnson and C. H. Papadimitriou. Performance guarantees for heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, pages 145–180. John Wiley & Sons, New York, NY, USA, 1985.

[65] S. Karlin and H. M. Taylor. *A Second Course in Stochastic Processes.* Academic Press,San Diego, CA, USA, 1981.

[66] R. M. Karp. Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. *Mathematics of Operations Research*, 2(3):209–224, 1977.

[67] R. M. Karp and J. M. Steele. Probabilistic analysis of heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, pages 181–205. John Wiley & Sons, New York, NY, USA, 1985.

[68] J. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13:180–210, 1995.

[69] J. H. Kingston. *Algorithms and Data Structures: Design, Correctness, Analysis*. International Computer Science Series. Addison-Wesley, Boston, MA, USA, first edition, 1990.

[70] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[71] S. Kirkpatrick and G. Toulouse. Configuration space analysis of travelling salesman problems. *Journal de Physique*, 46:1277–1292, 1985.

[72] B. Korte and J. Vygen. *Combinatorial Optimization*. Springer, Berlin, Germany, second edition, 2002.

[73] W. Krauth and M. Mézard. The cavity method and the travelling-salesman problem. *Europhysics Letters*, 8(3):213–218, 1988.

[74] M. G. Krein and A. A. Nudelman, editors. *The Markov Moment Problem and Extremal Problems*, volume 50 of *Translation of Mathematical Monographs*. American Mathematical Society, Providence, RI, USA, 1977.

[75] A. H. Land and A. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[76] H. J. Landau, editor. *Moments in Mathematics: Proceedings of Symposia in Applied Mathematics*, volume 37 of *AMS Short Course Lecture Notes*. American Mathematical Society, Providence, RI, USA, 1987.

[77] P. Larrañaga, C. Kuijpers, R. Murga, I. Inza, and S. Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Articial Intelligence Review*, 13:129–170, 1999.

[78] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*. John Wiley & Sons Ltd., NY, USA, 1985.

[79] S. Lin. Computer solution to the traveling salesman problem. *Bell System Tecnical Journal*, 44:2245–2269, 1965.

[80] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

[81] J. D. Little, K. G. Murty, D. W. Sweeney, and C. Karel. An algorithm for the traveling salesman problem. *Operations Research*, 11(6):972–989, 1963.

[82] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., New York, NY, USA, 1997.

[83] O. Martin, S. W. Otto, and E. W. Felten. Large-step Markov chains for the traveling salesman problem. *Complex Systems*, 5:299–326, 1991.

[84] K. L. McRoberts. A search model for evaluating combinatorially explosive problems. *Operations Research*, 19(6):1331–1349, 1971.

[85] I. I. Melamed, S. I. Sergeev, and I. K. Sigal. The traveling salesman problem: Approximate algorithms. *Avtomat. Telemekh*, 11:3–26, 1998.

[86] M. Mézard and G. Parisi. A replica analysis of the travelling salesman problem. *Journal de Physique*, 47(8):1285–1296, 1986.

[87] M. Molloy. The probabilistic method. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 1–35. Springer-Verlag, Berlin, Germany, 1998.

[88] H. Muhlenbein, M. Gorges-Schleuter, and O. Kramer. Evolutionary algorithms in combinatorial optimization. *Parallel Computing*, 7:65–88, 1988.

[89] A. J. Orman and H. P. Williams. A survey of different integer programming formulations of the travelling salesman problem. In E. J. Kontoghiorghes and C. Gatu, editors, *Optimisation, Econometric and Financial Analysis (Advances in Computational Management Science)*, pages 91–104. Springer-Verlag, 2006.

[90] C. H. Papadimitriou. The complexity of the Lin-Kernighan heuristic for the traveling salesman problem. *SIAM Journal of Computing*, 21(3):450–465, 1992.

[91] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, chapter Algorithms and Complexity. Prentice-Hall, Englewood Cliffs, NJ, USA, 1982.

[92] J. Y. Potvin. The traveling salesman problem: A neural network perspective. *ORSA Journal on Computing*, 5:328–348, 1993.

[93] A. Punnen, F. Margot, and S. Kabadi. TSP heuristics: Domination analysis and complexity. *Algorithmica*, 35(2):111–127, 2003.

[94] S. S. Ray, S. A. Bandyopadhyay, and S. K. Pal. New operators of genetic algorithms for traveling salesman problem. In *ICPR'04: Proceedings of the Pattern Recognition, 17th International Conference*, volume 2, pages 497–500, Washington, DC, USA, 2004. IEEE.

[95] S. S. Ray, S. A. Bandyopadhyay, and S. K. Pal. Genetic operators for combinatorial optimization in TSP and microarray gene ordering. *Applied Intelligence*, 26(3):183–195, 2007.

[96] G. Reinelt. TSPLIB — a traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[97] G. Reinelt. *The traveling salesman: Computational solutions for TSP applications*. Springer Verlag, Berlin, Germany, 1994. LNCS 840.

[98] R.-D. Reiss and M. Thomas. *Statistical Analysis of Extreme Values*. Birkhäuser Verlag, Berlin, Germany, third edition, 2007.

[99] S. Reiter and D. B. Rice. Discrete optimizing solution procedures for linear and nonlinear integer programming problems. *Management Science*, 12(11):829–850, 1966.

[100] W. T. Rhee. On the travelling salesperson problem in many dimensions. *Random Structures and Algorithms*, 3(3):227–234, 1992.

[101] W. T. Rhee and M. Talgrand. A sharp deviation inequality for the stochastic traveling salesman problem. *Annals of Probability*, 17:1–8, 1989.

[102] V. I. Rublineckii. Estimates of the accuracy of procedures in the traveling salesman problem (in Russian). *Numerical Mathematics and Computer Technology*, 3:1823, 1973.

[103] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

[104] R. Shamir. The efficiency of the simplex method: a survey. *Management Science*, 33(3):301–334, 1987.

[105] A. N. Shiryaev. *Probability*. Graduate Texts in Mathematics. Springer-Verlag, Berlin, Germany, 2nd edition, 1996.

[106] J. A. Shohat and J. D. Tamarkin. *The Problem of Moments*. American Mathematical Society, Providence, RI, USA, 1943.

[107] S. D. Shtovba. Ant algorithms: Theory and applications. *Programming and Computing Software*, 31(4):167–178, 2005.

[108] B. G. Sidharth. Fluctuations in the quantum vacuum. *Chaos, Solitons and Fractals*, 14(1):167–169, 2002.

[109] G. F. Simmons. *Introduction to Topology and Modern Analysis*. McGraw-Hill Book Company Inc., New York, NY, USA, 1963.

[110] S. Smale. The problem of the average speed of the simplex method. In *Proceedings of the 11th International Symposium on Mathematical Programming*, pages 530–539, Bonn, Germany, 1982.

[111] K. Smith. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing*, 11:15–34, 1999.

[112] A. Solomon, P. Sutcliffe, and R. Lister. Sorting circular permutations by reversal. In *Algorithms and Data Structures, 8th International Workshop*, volume 2748 of *Lecture Notes in Computer Science*, pages 319–328, Berlin, Germany, 2003. Springer Verlag.

[113] P. F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1–45, 1996.

[114] J. M. Steele. *Probability Theory and Combinatorial Optimization*. SIAM, Philadelphia, PA, USA, 1997.

[115] T. Stutzle and H. Hoos. Improvements on the ant system: Introducing MAXMIN ant system. In *Proceedings of the ICANNGA97 International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 245–249, Berlin, Germany. Springer Verlag.

[116] A. Tari, M. Telek, and P. Buchholz. A simplified moment-based estimation method for extreme probabilities, infinite and positive cases. *International Journal of Simulation: Systems, Science Technology*, 7:15–28, 2006.

[117] E. E. Tyrtyshnikov. How bad are hankel matrices? *Numerische Mathematik*, 67:261–269, 1994.

[118] V. Vig and U. S. Palekar. On estimating the distribution of optimal traveling salesman tour lengths using heuristics. *European Journal of Operational Research*, 186(1):111–119, April 2008.

[119] J. Wästlund. The travelling salesman problem in the stochastic mean field model. `www.math.chalmers.se/~wastlund/`, 2006.

[120] E. D. Weinberger. Correlated and uncorrelated fitness landscapes how to tell the difference. *Biological Cyernetics*, (63):325–336, 1990.

[121] D. Whitley. An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and Software Technology*, 43(14):817–831, 2001.

[122] G. V. Wilson and G. S. Pawley. On the stability of the travelling salesman problem algorithm of Hopfield and Tank. *Biological Cybernetics*, 58(1):63–70, 1988.

[123] I. A. Wood and T. Downs. Fast optimization by demon algorithms. In T. Downs, M. Frean, and M. Gallagher, editors, *Proceedings of the Ninth Australian Conference on Neural Networks (ACNN98), Queensland, Australia*, pages 245–249, 1989.

[124] J. E. Yukich. *Probability Theory of Classical Euclidean Optimization Problems*, volume 1675 of *Lecture Notes in Mathematics*. Springer, Berlin, Germany, 1998.

[125] M. Zachariasen and M. Dam. Tabu search on the geometric traveling salesman problem. In I. H. Osman and J. P. Kelly, editors, *Metaheuristics: Theory and Applications*, pages 571–587. Kluwer Academic, New York, NY, USA.

List of Figures

List of Algorithms

Index

Colophon

This document was typeset by the author using $\LaTeX\,2_\varepsilon$. The typeface used is predominantly Computer Modern 12 point. The plots were produced by Mathematica®6 and SPSS 15®. Graphs were hand constructed in OpenOffice draw or programmatically produced using c++ code written by the author. It was printed and bound, on acid-free paper, by World of Print, Sydney.