1: Challenges

It grew out of the trio's efforts to find solutions for a classic mathematical problem—the "Traveling Salesman" problem—which has long defied solution by man, or by the fastest computers he uses.

—IBM Press Release, 1964.[1]

An advertising campaign by Procter & Gamble caused a stir among applied mathematicians in the spring of 1962. The campaign featured a contest with a $10,000 prize. Enough to purchase a house at the time. From the official rules:

> Imagine that Toody and Muldoon want to drive around the country and visit each of the 33 locations represented by dots on the contest map, and that in doing so, they want to travel the shortest possible route. You should plan a route for them from location to location which will result in the shortest total mileage from Chicago, Illinois back to Chicago, Illinois.

Police officers Toody and Muldoon navigated *Car 54* in a popular American television series. Their 33-city task is an instance of the *traveling salesman problem*, or *TSP* for short. In its general form, we are given a collection of cities and the distance to travel between each pair of them. The problem is to find the shortest route to visit each city and to return to the starting point.

Is the general problem easy, hard, or impossible? The short answer is that no one really knows. This is both the mystery and attraction of this famous challenge in computational mathematics. And much more than a struggling salesman is at stake. The TSP is the focal point of a larger debate on the nature of complexity and possible limits to human knowledge. If you are ready for action, then a sharp pencil and a clean piece of paper are all you may need to give a helping hand to the salesman and possibly to make a quantum leap in our understanding of the world in which he or she travels.

Figure 1.1
Car 54 contest. Image courtesy of Procter & Gamble.

Tour of the United States

Despite its nasty reputation, the TSP is an easy enough task from one perspective: there are only finitely many possible routes through a given set of cities. So a 1962-era mathematician could have checked each possible Toody-Muldoon tour, recorded the shortest, sent the solution to Procter & Gamble, and waited for the $10,000 check to arrive in the mail. A simple and flawless strategy. With one possible catch. The number of distinct tours is exceedingly large to consider checking one by one.

This difficulty was noticed in 1930 by the Austrian mathematician and economist Karl Menger, who first brought the challenge of the TSP to the attention of the mathematics community. "This problem is of course solvable by finitely many trials. Rules that give a number of trials below the number of permutations of the given points are not known."[2] A tour can be specified by announcing the order in which the cities are to be visited. For example, if we label the 33 destinations of Toody and Muldoon as *A* through *Z* and *1* though *7*, that is, *A* for Chicago, *B* for Wichita, etc., then we can record a possible tour as

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567

or any other arrangement of the 33 symbols. Each such arrangement is a *permutation* of the symbols. The ordering implied by the arrangement is circular, in that we travel from the last city back to the first. So we can record the same tour in 33 ways, depending on which city we put in the first position. To avoid such overcounting, we may as well always start with city *A*. This leaves 32 choices for the second city, 31 choices for the third city, and so on. Altogether, we have $32 \times 31 \times 30 \times \cdots \times 3 \times 2 \times 1$ tours to consider. This is the total number of permutations of 32 objects. It is written as 32! and spoken as 32 *factorial*.

In the Procter & Gamble contest we can save effort by noting that the distance to travel between Chicago and Wichita is the same as the distance between Wichita and Chicago, and this is true also for every other pair of cities. With such symmetry it does not matter in which direction we travel around a tour, so an ordering

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567

is the same as its reverse

7654321ZYXWVUTSRQPONMLKJIHGFEDCBA.

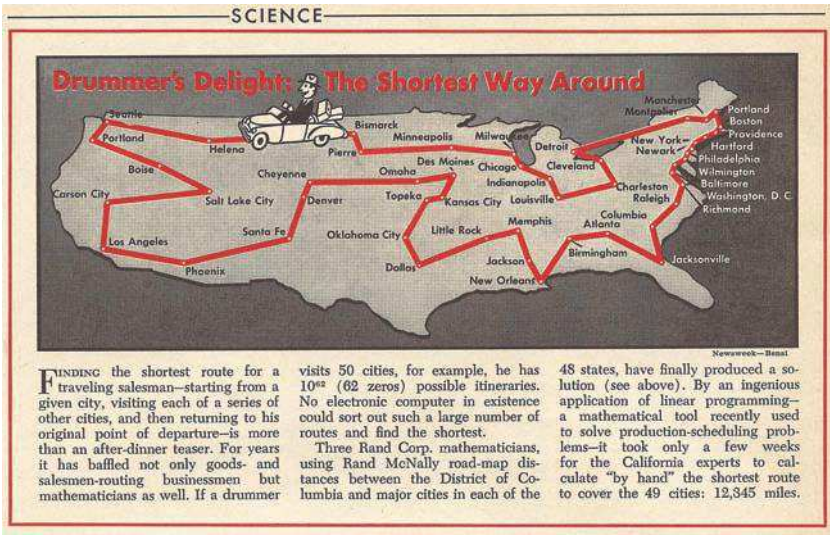
We can therefore cut down by half our count of the 33-city tours, leaving only 32!/2 orderings to check. Before you go ahead and get out your Ticonderoga #2 pencil, note that this is

131,565,418,466,846,765,083,609,006,080,000,000

distinct tours that we must examine.

These days we would of course employ a computer to run through the list. So let's choose a big one, the $133,000,000 IBM Roadrunner Cluster of the United States Department of Energy. This 129,600-core machine topped the 2009 ranking of the 500 world's fastest supercomputers, delivering up to 1,457 trillion arithmetic operations per second.[3] Let's assume we can arrange the search for tours such that examining each new one requires only a single arithmetic operation. We would then need roughly 28 trillion years to solve the 33-city TSP on the Roadrunner, an uncomfortable amount of time, given that the universe is estimated to be only 14 billion years old. No wonder Menger was unsatisfied with the brute-force solution to the problem.

When considering the implications of this quick analysis, we must keep in mind that Menger writes only that faster rules for solving the


SCIENCE

Drummer's Delight: The Shortest Way Around

Seattle
Portland
Helena
Bismarck
Pierre
Minneapolis
Milwaukee
Detroit
Chicago
Cleveland
Des Moines
Omaha
Indianapolis
Boise
Cheyenne
Carson City
Salt Lake City
Denver
Topeka
Kansas City
Louisville
Santa Fe
Oklahoma City
Little Rock
Memphis
Los Angeles
Phoenix
Dallas
Jackson
New Orleans
Birmingham
Atlanta
Columbia
Charleston
Raleigh
Manchester
Montpelier
Portland
Boston
Providence
Hartford
New York–Newark
Philadelphia
Wilmington
Baltimore
Washington, D. C.
Richmond
Jacksonville

Newsweek—Bemat

FINDING the shortest route for a traveling salesman—starting from a given city, visiting each of a series of other cities, and then returning to his original point of departure—is more than an after-dinner teaser. For years it has baffled not only goods- and salesmen-routing businessmen but mathematicians as well. If a drummer visits 50 cities, for example, he has 10^{62} (62 zeros) possible itineraries. No electronic computer in existence could sort out such a large number of routes and find the shortest.

Three Rand Corp. mathematicians, using Rand McNally road-map distances between the District of Columbia and major cities in each of the 48 states, have finally produced a solution (see above). By an ingenious application of linear programming—a mathematical tool recently used to solve production-scheduling problems—it took only a few weeks for the California experts to calculate "by hand" the shortest route to cover the 49 cities: 12,345 miles.

Figure 1.2
Drummer's Delight. *Newsweek*, July 26, 1954, page 74.

salesman problem are unknown, not that such rules are out of the question. John Little and coauthors sum this up nicely in the following comment on the Procter & Gamble contest. "A number of people, perhaps a little over-educated, wrote the company that the problem was impossible—an interesting misinterpretation of the state of the art."[4] Little et al. went on to describe a breakthrough in TSP solution methods, but they could not push their computer codes far enough to actually solve the 33-city challenge. It appears that no one in the country was able to produce a route that could be guaranteed to be the shortest of all possible tours for Toody and Muldoon.

The 33-city problem was definitely a tough nut to crack, but if we turn back the clock to 1954, then we find a team that almost certainly would be able to deliver the optimal route, together with a written guarantee that their solution is the shortest. The team tackled a larger touring problem through the United States, visiting a city in each of the 48 states, as well as Washington, D.C. This particular challenge had been circulating through the mathematics community since the mid-1930s. Its solution was reported in *Newsweek*.[5]

> Finding the shortest route for a traveling salesman—starting from a given city, visiting each of a series of other cities, and then returning to his original point of departure—is more than an

> after-dinner teaser. For years it has baffled not only goods- and salesman-routing businessmen but mathematicians as well. If a drummer visits 50 cities, for example, he has 10^{62} (62 zeros) possible itineraries. No electronic computer in existence could sort out such a large number of routes and find the shortest.
>
> Three Rand Corp. mathematicians, using Rand McNally road-map distances between the District of Columbia and major cities in each of the 48 states, have finally produced a solution. By an ingenious application of linear programming—a mathematical tool recently used to solve production-scheduling problems—it took only a few weeks for the California experts to calculate "by hand" the shortest route to cover the 49 cities: 12,345 miles.

The California experts were George Dantzig, Ray Fulkerson, and Selmer Johnson, part of an exceptionally strong and influential center for the new field of mathematical programming, housed at the RAND Corporation in Santa Monica.

The RAND team's guarantee involves some pretty mathematics that we take up later in the book. For now it is best to think of the guarantee as a proof, like those we learned in geometry class. The Dantzig et al. proof establishes that no tour through the 49 cities can have length less than 12,345 miles. Matching the proof with their tour of precisely this length shows that this particular instance of the TSP has been settled, once and for all.

Dantzig and company missed out on the $10,000 contest, but we can report that a computer implementation of their ideas makes easy work of the 33-city TSP. A shortest route for Toody and Muldoon is depicted in Figure 1.3. Although no one in 1962 knew for certain that this was the shortest possible tour, a number of contestants did find and report this

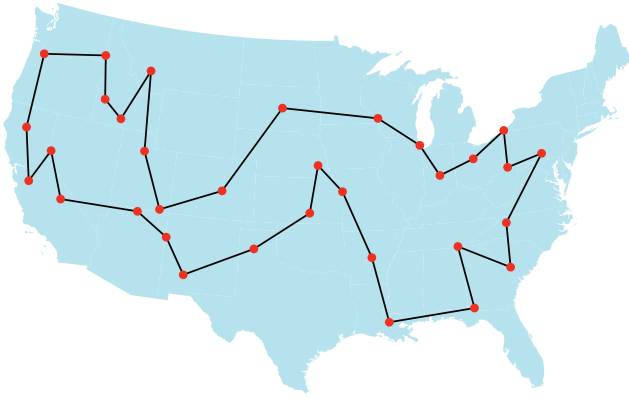


Figure 1.3
Optimal tour for *Car 54* contest.

same ordering. Among the people tied for first place in the contest were mathematicians Robert Karg and Gerald Thompson, who created a hit-or-miss heuristic strategy that produced the winning solution.[6] And the story has a happy ending, at least for the mathematics community. As a tiebreaker, contestants were asked to write a short essay on the virtues of one of Procter & Gamble's products. Thompson's prose on soaps took a grand prize.

An Impossible Task?

The RAND team's work put an end to the 48-states challenge, but it did not finish off the TSP. One big success did not imply the team could handle other, possibly larger, instances of the problem. In fact, if Las Vegas were taking bets on the outcome, the odds-on favorite among mathematicians would be that we will never fully solve the TSP. We must be careful here. By a solution we mean an *algorithm*, that is, a step-by-step recipe for producing an optimal tour for any example we may throw at it. Just finding the best route through the United States or any other country does not do the job.

Picking up on the expected difficulty of the general TSP challenge, the science-fiction story "Antibodies", by Charles Stross, chronicles doomsday events following the discovery of an efficient solution method for the salesman.[7] One can hope that a brilliant insight into the TSP will not signal the end of the world as we know it, but it will certainly turn the planet upside down and give it a good shake. To see why, let's start with a series of quotes.

> 'It seems very likely that quite a different approach from any yet used may be required for successful treatment of the problem. In fact, there may well be no general method for treating the problem and impossibility results would also be valuable.'
>
> —Merrill Flood, 1956.[8]

> 'I conjecture that there is no good algorithm for the traveling salesman problem.'
>
> —Jack Edmonds, 1967.[9]

> 'In this paper we give theorems which strongly suggest, but do not imply, that these problems, as well as many others, will remain intractable perpetually.'
>
> —Richard Karp, 1972.[10]

The authors of these remarks are three giants of traveling-salesman research. Merrill Flood rallied support for the problem in the 1940s; more than anyone else, Flood is responsible for the emergence of the TSP as a fundamental topic of study. Discussing the state of the problem in 1956, Flood first raised the possibility that efficient methods may simply never exist. This point was hammered home by Jack Edmonds a decade later in what amounts to a mathematical bet against the hope for a general solution method. Edmonds was modest in describing the support for his bet: "My reasons are the same as for any mathematical conjecture: (1) It is a legitimate mathematical possibility, and (2) I do not know." But he is teasing us with these words: Edmonds is one of the profound thinkers in twentieth-century mathematics and he certainly had something deep in mind when placing money against the TSP. Five years later, the true nature of the bet was made clear in a publication by the great computer scientist Richard Karp, connecting the TSP with a host of other computational problems. We save the details of Karp's theory for chapter 9, but a quick account will be enough to understand why the characters of "Antibodies" shuddered at the announcement of a fast TSP algorithm.

Good and Bad Algorithms

When Edmonds writes "good algorithm," he uses the word good in the same way as you and I: an algorithm is good if it can solve problems in an amount of time we find acceptable. For this to make sense in mathematics, however, he had to make "good" into a formal notion. Clearly, we cannot expect every example of the TSP to be solved, say, in under a minute by a human or by one of our machines. We must at least be willing to allow for the solution time to grow as the number of cities grows. The point to be decided is what rate of growth is acceptable.[11]

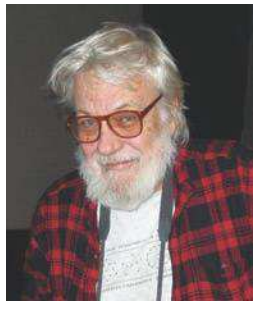


Figure 1.4
Jack Edmonds, 2009. Photograph courtesy of Marc Uetz.

Table 1.1
Running time on a 10^9-operations-per-second computer.

	$n = 10$	$n = 25$	$n = 50$	$n = 100$
n^3	0.000001 seconds	0.00002 seconds	0.0001 seconds	0.001 seconds
2^n	0.000001 seconds	0.03 seconds	13 days	40 trillion years

Let's use the symbol n to indicate the size of a problem; for the TSP this is the number of cities. Reading a list of locations to visit takes time proportional to n, so a tough manager might demand that we produce an optimal tour also in time proportional to n. Such a manager would be wildly optimistic. Edmonds himself allows for faster rates of growth in the running time, but with an insightful break between good and bad. A *good algorithm* is one that comes with a guarantee to complete its work in time at most proportional to n^k for some power k. The power k can be any value, such as 2, 3, or more, but it must be a fixed number—it cannot increase as n gets larger. Thus, a growth rate of n^3 is good, but growth rates of n^n and 2^n are bad. To give you a feeling for this, in table 1.1 we have calculated the running times for a few values of n, assuming a computer can handle 10^9 instructions per second. If $n = 10$, the bad algorithm is fine. But you don't want to be stuck behind a 2^n algorithm if n gets up to 100 or so.

Edmonds's formal notion of "good" might not always agree with our intuition. An algorithm that requires n^{1000} steps is not appealing if we need to solve an instance of the TSP with 100 cities. Nonetheless, his idea revolutionized the study of computing. The precise good/bad dichotomy creates real targets for mathematicians, fueling great interest in computational issues. And on the practical side, once a problem is shown to have a good algorithm, researchers pull out all stops in a race to decrease the value of the power k, typically getting down to running-time bounds proportional to n^2, n^3, or n^4, and computer codes capable of handling large instances.

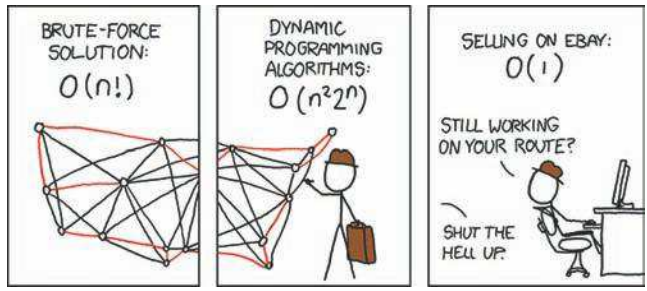


Figure 1.5
Travelling Salesman Problem. Image courtesy of Randall Munroe, xkcd.com.

Unfortunately for TSP fans, no good algorithm is known for the problem. The best result thus far is a solution method, discovered in 1962, that runs in time proportional to $n^2 2^n$. Although not good, this growth rate is much smaller than the total number of tours through n points, which we know is $(n-1)!/2$, perhaps satisfying the curiosity of Menger.

The Complexity Classes $\mathcal{P}$ and $\mathcal{NP}$

Edmonds's dichotomy carries over to computational problems, dividing them into those for which good algorithms exist and those for which they do not. The former problems are the ones we like, and they are known collectively as the class $\mathcal{P}$.

Why $\mathcal{P}$ and not $\mathcal{G}$? Well, researchers were not entirely comfortable with the emotional charge that comes with the word "good," and it became standard to use the term *polynomial-time algorithm*. So $\mathcal{P}$ for polynomial.

The definition of $\mathcal{P}$ is clear-cut, but it can be tricky to tell whether or not a given problem belongs to this "good" class. It may well be that the TSP is in $\mathcal{P}$ and we just haven't yet discovered the good algorithm to prove its membership. A glimmer of hope is that at least we know a good tour when we see one. Indeed, suppose our challenge is to find a tour, say, of length less than 100 miles. If someone hands us such a solution, then we can check easily that it does indeed beat the 100-mile target. This property makes the TSP a member of the class known as $\mathcal{NP}$, consisting of all problems for which we can check the correctness of a solution in polynomial time. The pair of letters stands for *non-deterministic polynomial*. The unusual name aside, this is a natural class of problems: when we make a computational request, we ought to be able to check that the result meets our specifications.

The Big Question

Could it be that $\mathcal{P}$ and $\mathcal{NP}$ are two names for the same class of problems? It is possible. An approach for proving this was laid out in a breakthrough result by Stephen Cook in 1971. (No relation to me, although I have enjoyed a number of free dinners due to mistaken identity.) Cook's Theorem states that there exists a problem in $\mathcal{NP}$ such that if we have a good algorithm for this single problem, then there is a good algorithm for every problem in $\mathcal{NP}$. In fact, Cook, Karp, and others have shown that there are many such *$\mathcal{NP}$-complete* problems, the most prominent being the TSP itself.

Finding a good algorithm for an $\mathcal{NP}$-complete problem would show that $\mathcal{P}$ is equal to $\mathcal{NP}$. Thus, the first person to discover a general method

for the TSP will bring home considerably more cash than the winner of the Procter & Gamble contest: the Clay Mathematics Institute has offered a $1,000,000 prize for either a proof or disproof that $\mathcal{P} = \mathcal{NP}$.

The betting line is that the two problem classes are not equal, but there is no great theoretical reason for thinking this is the case. It is simply a feeling that equality is too much to ask: any problem we can formulate in a verifiable manner would immediately have an efficient method of solution. In fact, current encryption systems make use of the assumption that certain $\mathcal{NP}$ problems are difficult to solve. Internet commerce would grind to a halt if there were quick algorithms for these members of $\mathcal{NP}$; this would be like handing code breakers and hackers a Swiss Army knife for snooping data.

The downfall of society in "Antibodies" was more insidious, however, than simply failures in encryption—artificial intelligence programs suddenly became greatly more effective and took over their biological masters. It seems probable we could deal with such pesky machines, and it is likely the good consequences of $\mathcal{P} = \mathcal{NP}$ would greatly outweigh the bad. In a 2009 survey article, Lance Fortnow wrote: "Many focus on the negative, that if $\mathcal{P} = \mathcal{NP}$ then public-key cryptography becomes impossible. True, but what we will gain from $\mathcal{P} = \mathcal{NP}$ will make the whole Internet look like a footnote in history."[12] His argument is that optimization becomes easy, thus salesmen can find their shortest routes, factories can run at peak capacity, airlines can manage their schedules without delays, and so on. Simply put, we will better utilize the resources available in our world. Vastly more powerful tools would also be available in science, economics, and engineering, providing a steady flow of breakthroughs to keep Nobel Prize committees busy for years to come. A rosy world, but the bets are against it.

The resolution of $\mathcal{P}$ versus $\mathcal{NP}$ is clearly one of the great challenges of our time. In approaching an $\mathcal{NP}$-complete problem like the TSP, however, it is important not to get too caught up in possible ramifications of a good solution method. The lofty implications aside, the problem comes down to a simple routing of a salesman. An ingenious idea could turn the scales.

One Problem at a Time

Until someone steps forward with a possibly earth-shattering result on the general complexity question, what is to be done with the TSP? Well, facing the salesman head on, the clear target is the solution of larger and more difficult instances of the problem.

The TSP is the standard bearer of a pragmatic school of research known as *algorithm engineering*.[13] The motto here is to not take no for an answer. Theoretical considerations may suggest that once we reach a certain size there exist instances of the TSP that necessarily take an exorbitant amount of computation, but this does not imply that whenever we see a specific large example we must give up and resort to a rough guess for a tour. Indeed, this take-no-prisoners attitude has led the community to techniques and computer codes capable of solving examples of almost unbelievable complexity.

Knocking off a previously unsolved challenge instance is a heralded event among researchers, akin to scaling a new Himalayan peak or running the 100-meter dash in record time. It is not that we have a desperate thirst for the details of particular optimal tours, but rather a desperate need to know that the TSP can be pushed back just a bit further. The salesman may defeat us in the end, but not without a good fight.

From 49 to 85,900

The heroes of the field are Dantzig, Fulkerson, and Johnson. Despite the dawning of the computer age and a steady onslaught of new researchers tackling the TSP, the 49-city example that Dantzig et al. solved by hand stood as an unapproachable record for seventeen years. Algorithms were developed, computer codes written, and research reports published, but year after year their record held its ground. The long run was finally snapped in 1971 by IBM researchers Michael Held and Richard Karp; the same Karp who studied TSP impossibility results, clearly not satisfied with theory alone. The test instance in this case consisted of 64 points dropped at random into a square region, with travel costs set to the straight-line distances between pairs of points.

The algorithm of Held and Karp reigned supreme for several years, with a number of teams tweaking the method in attempts to squeeze out greater performance. But the Dantzig et al. approach struck back in 1975, when Panagiotis Miliotis eclipsed the Held-Karp record by employing a variant of the original RAND idea to compute the shortest route through 80 random points.

The Miliotis work hinted at the fact that the Dantzig et al. approach might offer possibilities to push well beyond the expected limits of TSP computation. This was reinforced shortly thereafter by theoretical studies by Martin Grötschel and Manfred Padberg, who laid foundations for a great expansion of the basic methodology. This pair of mathematicians went

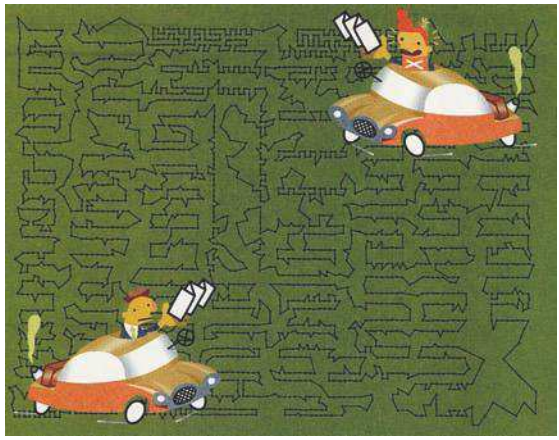


Figure 1.6
A new TSP record, 3,038 cities. *Discover*, January 1993.

on to dominate the TSP scene for the next fifteen years. Their successes began with Grötschel's construction of an optimal 120-city tour through Germany, published in his 1977 doctoral thesis. Padberg then teamed up with IBM researcher Harlan Crowder, computing the optimal solution for a 318-city example that arose in a circuit-board drilling application. These two results, although great in their own right, turned out to be only preliminary steps toward a series of startling announcements in 1987, a banner year for the TSP. Working independently on opposite sides of the Atlantic, Grötschel and Padberg led teams that solved in rapid succession instances consisting of 532 cities in the United States, 666 locations in the world, and 1,002-city and 2,392-city drilling problems; Grötschel worked with doctoral student Olaf Holland at the University of Bonn, and Padberg worked with Italian mathematician Giovanni Rinaldi at New York University.

Riding this wave of excitement, Vašek Chvátal and I decided to join the TSP-computation race in early 1988. We were in the unenviable position of trying to catch up to the fantastic efforts of Grötschel-Holland and Padberg-Rinaldi, but we had the luxury of working alongside a broad and active worldwide community delving ever deeper into the theoretical side of the problem. Sifting through the growing body of research on the TSP would provide powerful tools for use in a computational attack. Before getting into the process, however, we made the single most important step in the overall effort, recruiting to our team David Applegate and Robert Bixby, two of the strongest computational mathematicians of our time. Things started slowly and we had several false starts, but in 1992 we solved a record 3,038-city drilling problem, utilizing a large network of computers working in parallel. With the pieces now in place, the team computed an

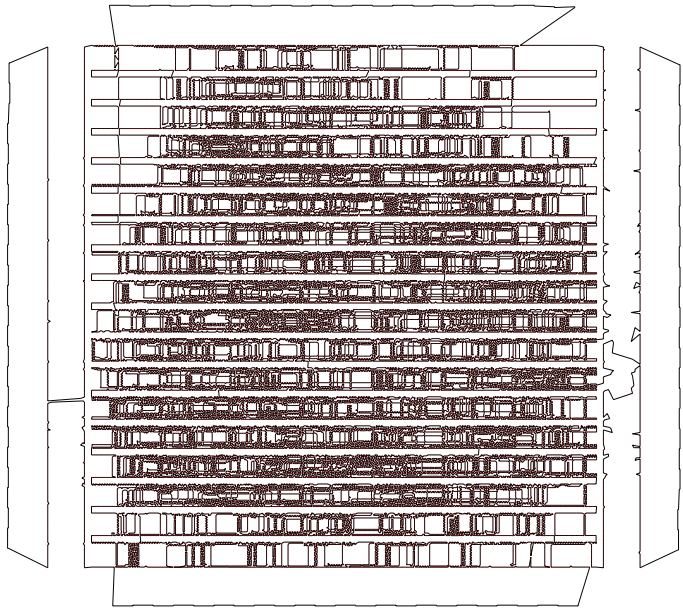


Figure 1.7
Solution of an 85,900-city TSP arising in a computer-chip application.

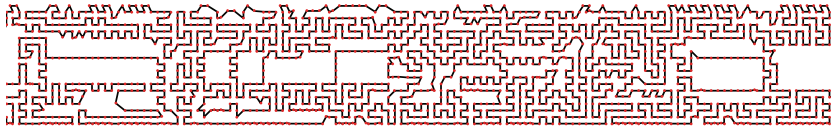


Figure 1.8
Close-up view of a portion of the 85,900-city tour.

optimal 13,509-city tour through the United States in 1998, an optimal 24,978-city tour of Sweden in 2004, and, finally, an optimal tour for an 85,900-city applied instance in 2006. The computer code used in these solutions is called *Concorde* and it is available over the internet.

The 85,900 cities in the record problem represent locations of connections that must be cut by a laser to create a customized computer chip. The TSP in this case models the movement of the laser from location to location. Although movements are measured in fractions of an inch, the total travel time was a major contributor to the chip's production cost. The optimal route for the laser is illustrated in figure 1.7, with a close-up view of a small region in figure 1.8.

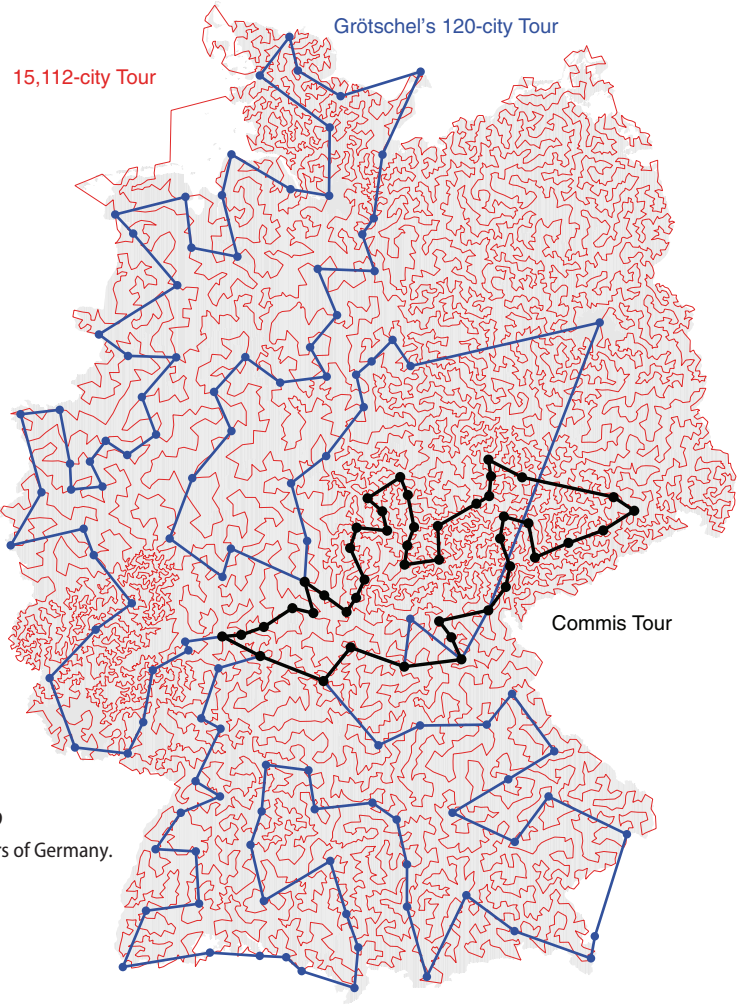


Figure 1.9
Three tours of Germany.

The World TSP

The grid-like distribution of points evident in the 85,900-city example, and in some of the drilling problems, does not really capture the traveling spirit of the 48-states tour that started the long TSP research program. But it is easy to appreciate the increased complexity of modern solutions by examining the three tours through Germany illustrated in figure 1.9. The small 33-city *Commis* tour was described in an 1832 book on tips for salesmen; the blue tour is Grötschel's 120-city record; and the tour in the background is an optimal route through 15,112 cities, computed with Concorde in 2001.

The 15,112-city route may be the final tour of Germany, but for an ultimate traveling challenge we put together a 1,904,711-city problem consisting of every city, town, and village in the world, including several research bases in Antarctica. Since 2001, this problem has withstood attacks by Concorde and by computer codes from around the globe. If the million-dollar Clay Prize is not to your taste, perhaps you would like to take on this World TSP Challenge. At the time of publication of this book, the best-known tour for the problem was produced by Danish computer scientist Keld Helsgaun. His tour of length 7,515,790,345 meters was found on October 10, 2010. This is almost certainly not the best-possible result, but we do know that no tour can be of length less than 7,512,218,268 meters, a bound computed with the Concorde code. Thus Helsgaun's tour is no more than 0.0476% longer than an optimal tour. That is close, but there is plenty of room for shortcuts.

Drawing the *Mona Lisa*

An optimal tour for the World TSP would be fantastic, but we are very likely more than a decade away from having the tools needed to make a serious attempt at its solution. Fortunately, there is no shortage of interesting problems to tackle along the way to conquering the world. A pretty example is the 100,000-city *Mona Lisa* TSP displayed in figure 1.10.

Figure 1.10
Leonardo da Vinci's *Mona Lisa* as a TSP. Tour found by Yuichi Nagata.

This data set was developed in February 2009 by Robert Bosch, to create a continuous-line drawing of da Vinci's famous portrait. The current best Mona Lisa tour was found by Yuichi Nagata of the Japan Advanced Institute of Science and Technology. His tour is known to be at most 0.003% longer than an optimal solution. This is tantalizingly close, but we are not yet home. As an incentive to anyone who might want to weigh in on this problem, there is a $1,000 prize offered to the first person who can improve on Nagata's tour. A nice trophy, but keep in mind that the real goal of problem-by-problem challenges is to gather ideas for use in general solution methods for the salesman, and beyond to application areas well outside the TSP. New avenues of attack are the name of the game.

Road Map of the Book

The T-shirt displayed in figure 1.11, with artwork by Jessie Brainerd, a 2007 Budapest Semester in Mathematics student, would be interpreted immediately as the TSP by every recent graduate of applied mathematics or computer science who is worth his or her salt.[14] Study of the salesman is a rite of passage in many university programs, and short descriptions have even worked their way into recent texts for middle school students.

With the existing wide coverage of the problem, what am I trying to accomplish with this book? The answer is simple: I plan to take the reader on a path that goes well beyond basic familiarity of the TSP, moving right up to current theory and state-of-the-art solution machinery. The ultimate goal is to encourage readers to take up their own pursuit of the salesman, with the hope that a knockout blow will come from an as yet unknown corner.

Figure 1.11
The TSP on Halloween 2007. Photograph courtesy of Jessie Brainerd.

We begin in chapter 2 by examining the roots of the salesman problem from both the mathematical and applied perspectives; the presentation of TSP history allows us to introduce basic themes picked up in later chapters. This is followed, in chapter 3, by a selection of the many applications of the TSP, including trip planning, genome sequencing, planet finding, and music arranging.

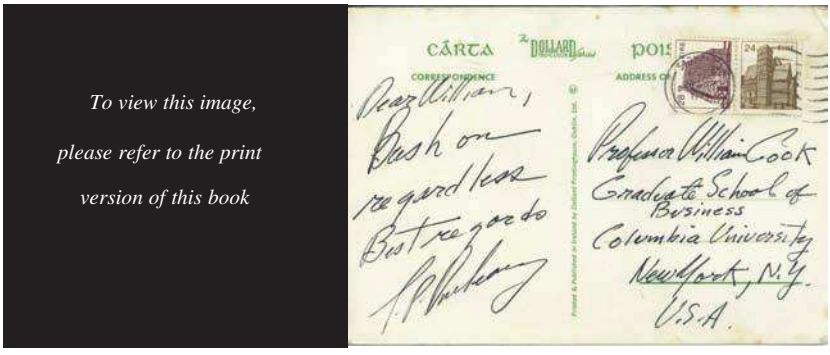
The heart of our technical treatment of the problem is the material presented in chapters 4 through 7, followed by a discussion of how TSP computer codes stack up to the task of solving large examples in chapter 8.

The $1,000,000 theoretical issue of a polynomial-time general method for the TSP is presented in chapter 9. If cold cash is what you desire, this is the chapter for you. I do not, however, recommend jumping ahead, even if your bank account is in desperate need of deposits. Indeed, the seeds of a successful theoretical attack may well be in methods that have proved themselves in the computational field of play. And if you are going for an impossibility result, you will need to handle the successful practical techniques in your proof.

Moving away from direct mathematics, in chapter 10 we cover studies on how humans, unaided by computers, go about solving the TSP; this area brings the problem into the realm of psychologists and neuroscientists. In chapter 11 we turn to the adoption of TSP tours in works of art, from the beautiful abstract paintings of Julian Lethbridge to the Jordan curves of Robert Bosch. Finally, chapter 12 wraps things up with a call for readers to take up the TSP challenge.

Figure 1.12

Left: W. Cook, far left, and V. Chvátal, far right, presenting author J. P. Donleavy a chamber pot, 1987. Photograph by Adrian Bondy. All rights reserved. Right: Postcard from J. P. Donleavy, 1987.



Bashing on Regardless

A bit of advice. When faced with an overwhelming number of slings and arrows, Irish writer J. P. Donleavy's character Rashers Ronald would vow to "Bash on regardless."[15] This became the rallying cry of the computational study of the TSP by Applegate et al. I recommend the reader, too, adopt this attitude when approaching the problem. We will cover work of numerous experts who have made huge advances, but the TSP remains essentially open. A new point of view could be just what is needed to dramatically alter our ability to tackle the salesman.

2: Origins of the Problem

It appears to have been discussed informally by mathematicians at mathematics meetings for many years.

—George Dantzig, Ray Fulkerson, and Selmer Johnson, 1954.[1]

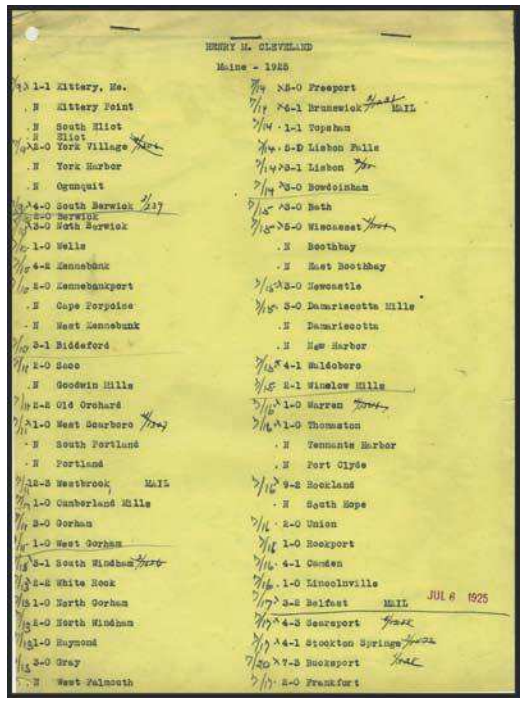
The traveling salesman problem is known far and wide, but the path it has taken to such mathematical prominence is somewhat obscure. For example, we cannot say for certain when the problem's lively name first came into use. Nevertheless, most of the story can be told, albeit with the help of an educated guess here and there. Its telling serves the useful side purpose of getting our TSP feet wet before jumping in with details of current attempts to crack the notorious problem.

Before the Mathematicians

As a practical matter, the TSP was tackled by humans long before it became a fashionable topic of study in mathematics. Our cave-dwelling elders no doubt solved small versions while out hunting and gathering, but likely without the aid of much in the way of long-term planning. In recent centuries, however, members of certain professions clearly did take advantage of carefully planned routes. An examination of their tours is a good place to begin our discussion.

Salesmen

Foremost among the route planers is the namesake of the TSP. Consider the list of cities given in the sheet displayed in figure 2.1. This item is part of the correspondence of salesman H. M. Cleveland in the year 1925.[2] Mr. Cleveland worked for the Page Seed Company, gathering orders for



HENRY M. CLEVELAND

Maine - 1925

1-1	Kittery, Me.	8-0	Freeport
.N	Kittery Point	6-1	Brunswick MAIL
.N	South Eliot	1-1	Topsham
8-0	York Village	8-D	Lisbon Falls
.N	York Harbor	8-1	Lisbon
.N	Ogunquit	8-0	Bowdoinham
4-0	South Berwick	8-0	Bath
8-0	Berwick	6-0	Wiscasset
8-0	Both Berwick	.N	Boothbay
1-0	Wells	.N	East Boothbay
4-R	Kennebunk	8-0	Newcastle
8-0	Kennebunkport	8-0	Damariscotta Mills
.N	Cape Porpoise	.N	Damariscotta
.N	West Kennebunk	.N	New Harbor
8-1	Biddeford	4-1	Waldoboro
8-0	Saco	8-1	Winslow Mills
8-8	Old Orchard	1-0	Warren
1-0	West Scarboro	1-0	Thomaston
.N	South Portland	.N	Tenants Harbor
.N	Portland	.N	Port Clyde
12-8	Westbrook MAIL	9-8	Rockland
1-0	Cumberland Mills	.N	South Hope
8-0	Gorham	8-0	Union
1-0	West Gorham	1-0	Rockport
8-1	South Windham	4-1	Camden
8-8	White Rock	1-0	Lincolnville
1-0	North Gorham	8-8	Belfast MAIL
8-0	North Windham	4-8	Searsport
1-0	Raymond	4-1	Stockton Springs
8-0	Gray	7-8	Bucksport
.N	West Falmouth	8-0	Frankfort

JUL 8 1925

Figure 2.1
Page Seed Company salesman list for Maine, 1925. One of five sheets.

corn and other products. His list of cities is one of five sheets outlining a tour of Maine. The full trip ran from July 9 through August 24, covering an amazing 350 stops.

Two observations make it clear that Mr. Cleveland and the Page Seed Company were interested in minimizing time spent on the road. First, the drawing of the tour, displayed in figure 2.2, reveals a remarkable efficiency in the itinerary; the portions where the tour appears to backtrack are all due to the available road network, where one town can only be reached by traveling to and from another town. Second, examine the following letter from Mr. Cleveland to his employer.

> July 15, 1925
>
> Dear Sirs
>
> My route list is balled up the worst I ever saw. Will take half as long again to work it as last year. I have changed it some beginning with Stockton Springs, Frankfort, Winterport, Hampden Highlands, Bangor, Stillwater, Orono, Oldtown, Millford, Bradley,

Brewer, So Brewer, Orrington, So Orrington, Bucksport, then to original at Orland.

I wish you would send me my old list 1924 from Dexter on as it is much better than this. I don't see how you could break it out as you did especially from Albion to Madison would be jumping all over the map. This section I changed.

The river from Bangor down has no bridge and you have those towns down as if I could cross it anywhere. Last season's list was made out the best of any one and I can't see the object of changing it over. I think I have made myself plain.

—Yours truly, H. M. Cleveland

Mr. Cleveland was most unhappy with part of the tour and went ahead with his own improvements in its design.

Maine was just one of the destinations of Mr. Cleveland in 1925. He also traveled through Connecticut, Massachusetts, New York, and Vermont, making over 1,000 stops in total. And he was far from being the only

Figure 2.2
Page Seed Company salesman tour of Maine, 1925. The tour starts at Kittery and ends at nearby Springvale, both in the south of the state.

person making the rounds. Timothy Spears's book *100 Years on the Road: The Traveling Salesman in American Culture* cites an 1883 estimate by *Commercial Travelers Magazine* of 200,000 traveling salesmen working in the United States, and a further estimate of 350,000 by the turn of the century. This number continued to grow through the early 1900s, and in Mr. Cleveland's day the salesman was a familiar site in most American towns and villages.

Spears describes how these salesmen used aids such as L. P. Brockett's *Commercial Traveller's Guide Book* to map out routes through their regions. Often, however, tours were planned in a central office, such as was done in the Page Seed Company. The images in figure 2.3 indicate one way such tours were optimized, using pins and strings to plot potential routes on a map.

An important reference in this discussion is the 1832 German handbook *Der Handlungsreisende—Von einem alten Commis-Voyageur*.[3] The *Commis-Voyageur* describes the need for good tours.[4]

> Business leads the traveling salesman here and there, and there is not a good tour for all occurring cases; but through an expedient choice and division of the tour so much time can be won that we feel compelled to give guidelines about this. Everyone should use as much of the advice as he thinks useful for his application. We believe we can ensure as much that it will not be possible to plan the tours through Germany in consideration of the distances and the traveling

Figure 2.3
Rand McNally map cabinet and pin map. *Secretarial Studies*, 1922.

Figure 2.4
1832 German salesman book.

back and forth, which deserves the traveler's special attention, with more economy. The main thing to remember is always to visit as many localities as possible without having to touch them twice.

This is an explicit description of the TSP, made by a traveling salesman himself!

The *Commis-Voyageur* book presents five routes through regions of Germany and Switzerland. Four of these routes include return visits to an earlier city that serves as a base for that part of the trip. The fifth route, however, is indeed a traveling salesman tour, indicated in figure 2.5. (The position of the route within Germany can be seen in the three-tours map displayed in figure 1.9.) As the *Commis-Voyageur* suggests, the tour is very good, perhaps even optimal, given road conditions at the time.

Numerous volumes written later in the century describe well-chosen routes in the United States, Britain, and other countries. The romantic image of the traveling salesman is captured, too, in stage, film, literature,

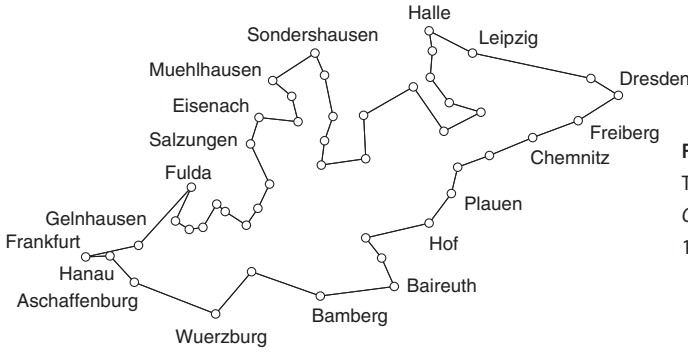


Figure 2.5
Tour by the alten *Commis-Voyageur*, 1832.

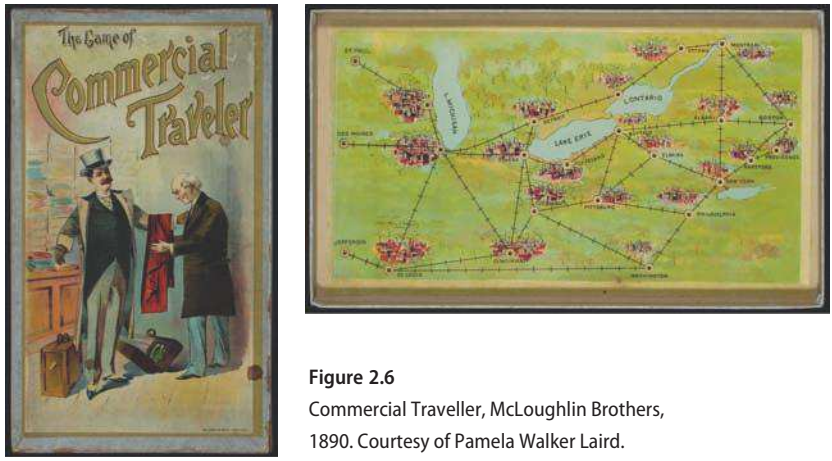


Figure 2.6
Commercial Traveller, McLoughlin Brothers, 1890. Courtesy of Pamela Walker Laird.

and song. The following is a typical turn-of-the-century salesman poem, taken from a compilation published in 1892.[5]

> Those who think a Drummer's life
> Is free from hardship, toil and strife,
> Are mistaken, for he has to go
> Through mud and rain, through sleet and snow.
> He sallies forth, gripsack in hand
> To seek for custom through the land.

The struggling drummer and his route-finding task were even featured in a board game, Commercial Traveller, created by McLoughlin Brothers in 1890, that asked players to build their own tours through a rail system. The choice of the salesman as the representative for the TSP is definitely well founded.

Lawyers

The salesman may have top billing, but other groups also traveled the land. The *Oxford English Dictionary* cites examples of the use of the word "circuit" as far back as the fifteenth century, concerning judicial districts in the United Kingdom. Traveling judges and lawyers served their districts by riding a circuit of the towns and villages, where court was held during specified times of the year. This practice was later adopted in the United States, where regional courts are still referred to as circuit courts, even though judges no longer take to the road.

Easily the best-known circuit rider in the history of the United States is the young Abraham Lincoln, who practiced law before becoming the country's sixteenth president. Lincoln worked the Eighth Judicial Circuit in the state of Illinois, covering fourteen county courthouses. His travel is described by Guy Fraker in the following passage.[6]

> Each spring and fall, court was held in consecutive weeks in each of the 14 counties, a week or less in each. The exception was Springfield, the state capital and the seat of Sangamon County. The fall term opened there for a period of two weeks. Then the lawyers traveled the fifty-five miles to Pekin, which replaced Tremont as the Tazewell County seat in 1850. After a week, they traveled the thirty-five miles to Metamora, where they spent three days. The next stop, thirty miles to the southeast, was Bloomington, the second-largest town in the circuit. Because of its size, it would generate more business, so they would probably stay there several days longer. From there they would travel to Mt. Pulaski, seat of Logan County, a distance of thirty-five miles; it had replaced Postville as county seat in 1848 and would soon lose out to the new city of Lincoln, to be named for one of the men in this entourage. The travelers would then continue to another county and then another and another until they had completed the entire circuit, taking a total of eleven weeks and traveling a distance of more than four hundred miles.

Fraker writes that Lincoln was one of the few court officials who regularly rode the entire circuit. A drawing of the route used by Lincoln and company in 1850 is given in figure 2.7. The tour is not quite the shortest possible (at least as the crow flies), but it is clear that it was constructed with an eye toward minimizing the travel of court personnel.

Preachers

The word circuit may have originated with the travel of judges and lawyers, but as a group they are rivaled in fame by the circuit-riding Christian preachers of the eighteenth and nineteenth centuries. John Hampson wrote the following passage in his 1791 biography of John Wesley, the founder of the Methodist church. "Every part of Britain and America is divided into regular portions, called circuits; and each circuit, containing twenty or thirty places, is supplied by a certain number of travelling preachers, from two to three or four, who go around it in a month or six weeks."[7] The conditions under which these men traveled is folklore in Britain, Canada,

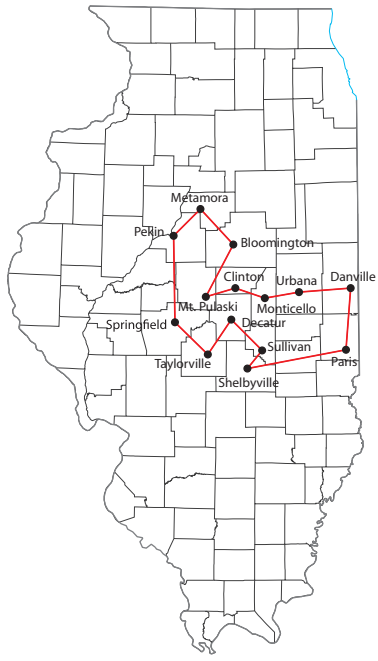


Figure 2.7
Eighth Judicial Circuit traveled by Lincoln in 1850.

and the United States. A feeling for the extent of their tours can be gathered from the following quotes.

> I travelled about five thousand miles, preached about five hundred sermons, visited most of the circuits in Virginia and North Carolina.
>
> —Freeborn Garrettson, 1781.[8]

> Our circuit at that time, was five hundred miles around it, and for me to preach as I did sixty-three sermons in four weeks, and travel five hundred miles, was too hard. But I cried unto the Lord and he heard me; for as my day was, so was my strength.
>
> —Billy Hibbard, 1825.[9]

I have not been able to obtain detailed itineraries of any of the longer circuits traveled by these Methodist preachers, but it is safe to assume that some planning went into the selection of the routes. A goal of their work was to reach as many church members as possible, so minimizing time on the trail would have been an important consideration.

Euler and Hamilton

Back in the realm of mathematics, the plight of salesmen, lawyers, and preachers did not capture the attention of busy researchers, who had their hands full, laying down fundamentals for the rapidly expanding fields of the physical sciences. Two of the leading figures of the era did, however, explore aspects of the TSP, and they are rightly viewed as the grandfathers of traveling-salesman research.

Graph Theory and the Bridges of Königsberg

The great Leonhard Euler wrote the most important of all early mathematical papers describing touring problems. The Euler Archive cites an estimate by historian Clifford Truesdell that "in a listing of all of the mathematics, physics, mechanics, astronomy, and navigation work produced during the 18th Century, a full 25% would have been written by Leonhard Euler." History's most prolific mathematician studied a vast array of topics, including a puzzle that was a longstanding challenge to the residents of the town of Königsberg in East Prussia.

A satellite image of Königsberg, now called Kaliningrad, reveals the elaborate waterway formed by the River Pregel. The rectangular island created by the splitting of the river is called the Kneiphof; the large island to the east of the Kneiphof is called Lomse; the region north of the river is the Altstadt; and the region south of the river is the Vorstadt.[10]

In Euler's day, the Pregel was crossed by seven walkways: the Green and Köttel bridges joined the Kneiphof to the Altstadt, the Krämer and

Figure 2.8
Königsberg and the River Pregel, TerraServer.com, 2011.

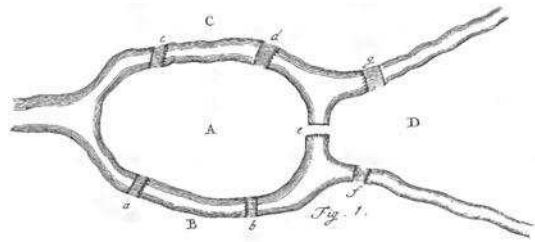


Figure 2.9
Euler's drawing of the Königsberg bridges.

Schmiede bridges joined the Kneiphof to the Vorstadt, the Honey bridge joined the Kneiphof and the Lomse, the High bridge joined the Lomse and the Altstadt, and the Wood bridge joined the Lomse and the Vorstadt. The good citizens of Königsberg enjoyed strolls through their town, crisscrossing the Pregel via the Green, Köttel, Krämer, Schmiede, Honey, Lomse, and Wood. The tale is that the Königsbergers had a standing challenge of crossing each of the seven bridges exactly once on a single walk through the town.

Euler weighed in on the Königsberg problem with a paper presented to the Academy of Sciences in Saint Petersburg on August 26, 1735.[11] His treatment follows a classic mathematical line of abstracting just the necessary information to capture the essence of the problem, and in so doing he laid the foundation for an important new branch of mathematics known as *graph theory*.[12]

To begin, Euler removed the physical nature of the challenge, sketching the town, river, and bridges, as displayed in figure 2.9. (This is a cleaned-up version of a computer scan taken from a copy of Euler's original published paper.) Euler labeled the regions of Königsberg as A, B, C, and D, and the seven bridges as a through g. These labels are enough to describe any route through the town, such as A to C via the c bridge, C to D via the g bridge, D to B via the f bridge, and B to A via the b bridge. A shorthand for this route would be $AcCgDfBbA$. Euler's arguments are based entirely on manipulating the routes as strings of symbols, rather than as walkers crossing the town.

The size of the land regions does not play a role in Euler's work, so the arguments can be visualized by a simple diagram, where A, B, C, and D are drawn as points, and a through g are drawn as lines joining pairs of these points, as in figure 2.10. The interpretation of the drawing is not influenced by the shape or length of the points and lines, but only by which pairs of points are joined. An object such as this is called a *graph*. The points of the graph are called *vertices*, the lines are called *edges*, and each edge has as its *ends* two of the vertices.

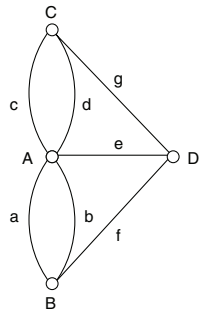


Figure 2.10
Graph representation of the Königsberg bridges.

In this stripped-down setting, a walk through Königsberg translates to movement from vertex to vertex in the graph, traveling along the graph's edges. A possible walk, starting at B and ending at D, is $B\,a\,A\,c\,C\,g\,D\,e\,A\,b\,B\,f\,D$. In this walk, three edges meet vertex B, namely a, b, and f; four edges meet vertex A, namely a, c, e, and b; two edges meet vertex C, namely c and g; and three edges meet vertex D, namely g, e, and f. The key observation of Euler is that the odd-even-even-odd pattern to these numbers is no accident: in any walk between two distinct points, the starting and ending vertices meet an odd number of edges and all other vertices meet an even number of edges. Furthermore, if we have a closed walk, that is, we start and end at the same point, then every vertex meets an even number of edges. So we have either all "even" vertices or exactly two "odd" vertices.

This is bad news for the Königsbergers. All four vertices of their bridge graph meet an odd number of edges, thus there can be no walk using each edge exactly once. Euler's short argument put an end to the Königsberg debate.

The Knight's Tour

Several years after settling the Königsberg puzzle, Euler wrote on a second touring challenge, known as the *knight's tour* problem in chess.[13] The task here is to find a sequence of knight's moves that take the piece from a starting square on a chessboard, through every other square exactly once, and then back to the starting square. Euler's solution is depicted in figure 2.11, where the order of moves is indicated by numbers on the squares.

The idea of a traveling knight appealed to Euler, who also laid out routes for boards of nonstandard size. These problems can be framed nicely using the language of graphs. In this case, we have a vertex for each square

42	57	44	9	40	21	46	7
55	10	41	58	45	8	39	20
12	43	56	61	22	59	6	47
63	54	11	30	25	28	19	38
32	13	62	27	60	23	48	5
53	64	31	24	29	26	37	18
14	33	2	51	16	35	4	49
1	52	15	34	3	50	17	36

Figure 2.11
Euler's solution to the knight's tour problem.

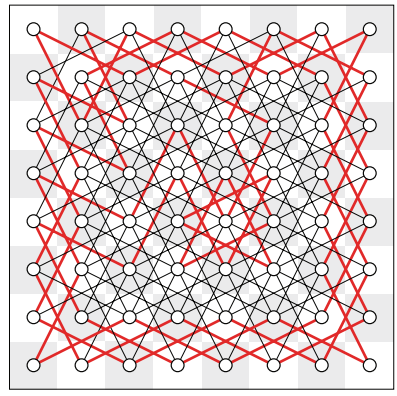


Figure 2.12
Knight's tour in the chessboard graph.

on the board, with two vertices joined by an edge if a knight can travel between the squares in a single move. A knight's tour is a closed walk that visits each vertex exactly once. (Note the similarity with the Königsberg problem, where we sought a walk traversing each edge exactly once.) The particular graph for the full chessboard is displayed in figure 2.12, together with Euler's route for the knight.

The Icosian

Ireland's Sir William Rowan Hamilton was also drawn to a question involving tours in a particular graph. A century after Euler, Hamilton studied ways to visit all twenty corner points of the dodecahedron, the twelve-sided Platonic solid. Hamilton made use of an abstract drawing he dubbed the *Icosian*, displayed in figure 2.13. The lines of the Icosian represent the dodecahedron's geometric edges and the circles represent its corners.

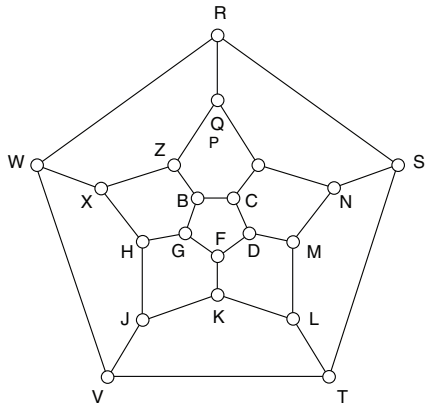


Figure 2.13
The Icosian.

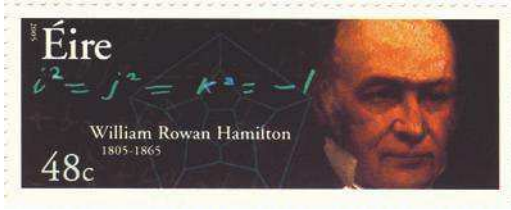


Figure 2.14
W. R. Hamilton commemorative stamp, Irish An Post, 2005. Portrait of Hamilton courtesy of the Royal Irish Academy.

The Icosian is a graph, and Hamilton's tours also proceed from vertex to vertex, always traveling along the graph's edges. Interestingly, Hamilton used an algebraic system to view this travel, in an approach similar in spirit to his defining equations for quaternions. He describes this in a formal letter to his friend John T. Graves in 1856.[14]

> As in the little paper which I lately sent you, let me continue to assume three symbols, i, κ, λ, which shall satisfy the four following equations:
>
> $$i^2 = 1,\ \kappa^3 = 1,\ \lambda^5 = 1,\ \lambda = i\kappa.$$
>
> What I have first to show, by one or two examples, is that the symbols so defined have curious but determinate properties, making them the legitimate instrument of a calculus: every symbolic result of which, so far as I can judge, and I have examined a great number of them, admits of easy and often interesting interpretation, with reference to the passage from face to face, or from corner to corner, of one or other of the solids considered in the ancient geometry.

The three symbols correspond to operations in the Icosian; when symbols are multiplied, the operations are made one after the other.[15] Through his calculus on these symbols, Hamilton showed that no matter what path

Figure 2.15
Left: The Icosian Game. Right: The Traveller's Dodecahedron.

of five vertices is chosen as a start, it is always possible to complete a tour through the remaining vertices of the Icosian. Fascinated with this structure, Hamilton concluded his letter to Graves with a description of a game to be played on the Icosian graph.

> I have found that some young persons have been much amused by trying a new mathematical game which the Icosian furnishes, one person sticking five pins in any five consecutive points, such as $abcde$, or $abcde'$, and the other player then aiming to insert, which by the theory in this letter can always be done, fifteen other pins, in cyclical succession, so as to cover all the other points, and to end in immediate proximity to the pin wherewith his antagonist had begun.

Two versions of the game were later marketed by a British toy merchant. One variant, called the Icosian Game, consists of a wooden board with ivory pegs to mark visited points. The second variant, called the Traveller's Dodecahedron: A Voyage Round the World, is a handheld device, shaped as a partially flattened dodecahedron, with pegs for the points and a string to trace out the tour.[16]

Despite Hamilton's enthusiasm, the games were a flop in the commercial market. If you try a few rounds of play you will see quickly the problem: finding tours in the Icosian graph is too easy. Hamilton was quite defensive about this point, stating that the puzzles were not at all easy for him. This odd status, where the game is simple for children but challenging for Ireland's greatest mathematician, may have been due to Hamilton's algebraic view of things. Perhaps Hamilton was solving the puzzles through mental manipulation of i, κ, and λ, rather than tracing the tours visually.

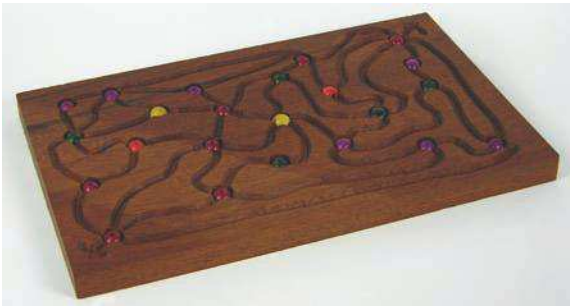


Figure 2.16
Worried Woodworm.
© 2009 Hordern-Dalgety Collection, puzzlemuseum.com.

On a happier note, a twentieth-century variant of Hamilton's game did manage to bring in a significant number of sales. James Dalgety's Worried Woodworm puzzle, from 1975, asks for walks in a particular graph, but in this case the routes are tricky to spot. Dalgety's wooden board is displayed in figure 2.16. The main goal is to discover a path starting in the bottom left, ending at the top right, and visiting every hole along the way.

The Concorde code was used recently to settle additional Worried Woodworm challenges posed by Dalgety, but fair-minded players would, no doubt, frown on employing a state-of-the-art TSP solver and a high-powered computer to plot the worm's path through the twenty-three points.

Hamiltonian circuits

Euler's knights and Hamilton's game-playing children both search for tours in graphs, but what about a general question? Not all graphs possess a tour through their vertices and a challenge is to decide which do and which do not. Hamilton's fame added considerable luster to this challenge at a time when graph theory was just beginning to find its place within the mathematical world. This explains why his name gets top billing when describing the problem. But do not jump in alarm at the snub of Euler. Graph theorists reserve Euler's name for closed walks that model the sought-after trip through Königsberg. Thus, a *Hamiltonian circuit* in a graph is a closed walk that visits each vertex exactly once, while a *Eulerian walk* is a closed walk that travels along each edge exactly once. Both walks are fundamental concepts in graph theory, but there is a world of difference between the two, despite the obvious similarities.

Deciding whether or not a graph has a Hamiltonian circuit is an $\mathcal{NP}$-complete problem, capturing much of the complexity of the general TSP. On the other hand, there is a simple rule for determining if a graph has an

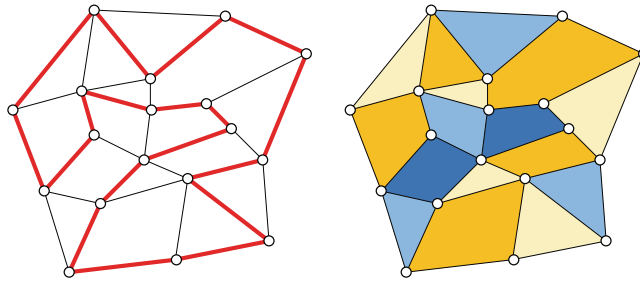


Figure 2.17
Coloring a map via a Hamiltonian circuit.

Eulerian walk, namely, except for vertices meeting no edges at all, the graph must be connected, that is, it consists of a single piece, and each vertex must be the end of an even number of edges.

So we understand Euler, but not Hamilton. Indeed, year after year, brave mathematicians have suggested conditions guaranteeing Hamiltonian circuits, only to see their conjectures fail. A famous example is due to P. G. Tait in the 1880s. Tait was caught up in the excitement of Alfred Kempe's announced proof of the four-color theorem. This result states that the regions (countries) of any map can be colored with at most four colors in such a way that any two regions sharing a border receive different colors. Looking for an alternative proof that four colors suffice, Tait conjectured that a certain type of graph always has a Hamiltonian circuit.

To see the connection between traveling and map coloring, think of the boundaries of a map's regions as the edges of a graph, with the intersection points as vertices. A Hamiltonian circuit through this boundary graph gives a way to color the map, as illustrated in figure 2.17, where the red edges form a Hamiltonian circuit. Such a circuit does not cross itself, so it has an inside and an outside. Moreover, the border edges on the inside, that are not part of the circuit, cut across the inner area. We can thus color these inner regions with two colors, switching every time we cross one of the non-circuit edges. The same trick allows us to two-color the regions lying outside of the Hamiltonian circuit, yielding altogether a four-coloring of the map. In the example, the inner regions are colored dark yellow and light yellow, and the outer regions are colored dark blue and light blue.

Tait knew that not all maps have Hamiltonian circuits through their borders (the map of the continental United States is a ready example), but available tricks allowed the four-color problem to be restricted to maps such that each vertex of the border graph meets exactly three edges. Furthermore, the border graph could be assumed to be three-connected, that is, it is impossible to break the graph into two parts by deleting one or

two vertices. Restricted to these three-regular, three-connected maps, Tait expected Hamiltonian circuits would always be available.

William Tutte, the great graph theorist and Bletchley Park code breaker, eventually showed Tait's conjecture to be false in 1946. That is too bad, but at least the circuit problem stood its ground longer than Kempe's four-color proof, which was shown to be incorrect by P. J. Heawood in 1890.

A historical footnote is that the first recorded description of the four-color problem is in a letter to Hamilton, written by Augustus De Morgan in 1852. Hamilton was not impressed with the problem, replying, "I am not likely to attempt your 'quaternion of colours' very soon."[17]

Mathematical Genealogy

Mathematicians enjoy tracing their academic heritage, following their Ph.D. thesis adviser to their adviser's adviser, and so on back through time. The Mathematics Genealogy Project Web site run by North Dakota State University contains over 130,000 records of Ph.D. advisers, with a goal to compile information on all the world's mathematicians. I am proud to trace my own roots back to Victorian-era mathematician Arthur Cayley, with an informal leap over to Sir Hamilton himself.

The path to Cayley is direct: W. Cook to U.S.R. Murty to C. R. Rao to Ronald Fisher to James Jeans to Edmund Whittaker to Andrew Forsyth to Arthur Cayley. The formal path stops here, since Cayley was trained in the law and did not obtain a Ph.D. degree. Cayley had great interest in mathematics, however, and in 1848 he traveled to Dublin to attend Hamilton's lectures on quaternions at Trinity College. Influenced by Hamilton, Cayley went on to write several hundred mathematical papers while practicing law, leading up to his appointment to the Sadleirian chair of mathematics at Cambridge in 1863. Cayley did not pick up Hamilton's interest in TSP-related problems, but he is a well-known figure in graph theory, introducing the notion of "trees" that we cover later in the book.

Vienna to Harvard to Princeton

Euler and Hamilton studied tours, but chessboards and dodecahedrons are a far cry from a salesman out on the road. A salesman is not satisfied with any old tour, she wants one of shortest possible length.

To bring in travel costs, we must jump ahead another century to Karl Menger and his work in Vienna. One of Menger's favorite topics in the 1920s was the study of techniques to measure lengths of curves in space.

This esoteric research likely provided inspiration for his announcement of a close relative of the TSP, made at a colloquium held on February 5, 1930.[18]

> We use the term *Messenger problem* (because this question is faced in practice by every postman, and, by the way, also by many travelers) for the task, given a finite number of points with known pairwise distances, to find the shortest path connecting the points.

The problem is to find a path through the points, without a return trip to the start. This is easily converted to the TSP by adding an extra "dummy" city that serves to link the ends of the path. The cost of travel between the dummy and each of the real cities can be set to zero, so that visiting the extra city will not influence the choice of the path's starting point or ending point.

Menger's "messenger problem" is recorded, in German, as part of the published documentation of the Vienna Mathematics Colloquium. The announcement is of clear historical importance, but it does not appear to have been the direct source of interest in the TSP among researchers in the United States. This honor goes to a lecture presented by prominent Harvard mathematician Hassler Whitney, cited in the following passage from Dantzig, Fulkerson, and Johnson's classic paper.[19]

> Merrill Flood (Columbia University) should be credited with stimulating interest in the traveling-salesman problem in many quarters. As early as 1937, he tried to obtain near optimal solutions in reference to routing of school buses. Both Flood and A. W. Tucker (Princeton University) recall that they heard the problem first in a seminar talk by Hassler Whitney at Princeton in 1934 (although Whitney, recently queried, does not seem to recall the problem).

Merrill Flood himself also credits Whitney's lecture when describing the history of the TSP in a 1956 research paper. "The problem was posed, in 1934, by Hassler Whitney in a seminar talk at Princeton University."[20] Even well after the fact, Flood refers to the TSP as the "48-states problem of Hassler Whitney" in an interview with Albert Tucker in 1984.[21]

It is natural to speculate on a possible connection between Menger's Vienna colloquium and Whitney's Princeton seminar. Support for such a connection was found by Alexander Schrijver, who notes that Menger and Whitney met at Harvard University in 1930–31, during a semester-long visit by Menger.[22] It is unclear, however, if the two actually exchanged information directly related to the salesman/messenger problem.

It also remains a question whether Whitney did in fact discuss the TSP at Princeton. There unfortunately is not an accessible record at

Figure 2.18
Hamiltonian circuit through Africa.

Princeton University covering the seminars delivered in the Department of Mathematics in the 1930s. The Pusey Library at Harvard University does, however, contain an archive of 3.9 cubic feet of Whitney's papers, and within the archive there is a set of handwritten notes that appear to be preparation for a seminar by Whitney, written sometime in the years shortly after 1930. The notes present an introduction to graph theory, including the following paragraph.

> A similar, but much more difficult problem is the following. Can we trace a simple closed curve in a graph through each vertex exactly once? This corresponds to the following problem. Given a set of countries, is it possible to travel through them in such a way that at the end of the trip we have visited each country exactly once?

In Whitney's problem, a graph is formed by placing a single vertex in each country, and joining two vertices by an edge if the countries share a border. A trip through the countries is a Hamiltonian circuit in the graph. This is an unusual choice as an example to describe the Hamiltonian-circuit problem and it is clearly not a far step from the TSP.

The geographic aspect of this example matches Flood's recollection of the "48-states problem." Indeed, Whitney's illustration of Hamiltonian

circuits may well be the starting point of TSP research in the United States. In the words of Alan Hoffman and Philip Wolfe, Whitney served "possibly as a messenger from Menger" in bringing the salesman to the mathematics community.[23]

And on to the RAND Corporation

There is not a record of the study of the salesman problem, under the TSP name, in the late 1930s and into the 1940s, but by the end of the 1940s it had become a known challenge. At this point the center of TSP action had moved from Princeton to RAND, coinciding with Flood's relocation to California.

Princeton University's Harold Kuhn writes the following in a December 2008 e-mail letter.

> The traveling salesman problem was known by name around Fine Hall by 1949. For instance, it was one of a number of problems for which the RAND corporation offered a money prize. I believe that the list was posted on a bulletin board in Fine Hall in the academic year 1948–49.

The RAND prize list! The TSP literature is peppered with mention of these prizes, but it is no longer easy to track down a copy of the original RAND document. Hoffman and Wolfe describe the RAND prize as one "for a significant theorem bearing on the TSP." The list, together with the great reputation of the RAND research group, played an important role in spreading the news of the TSP, although the prize itself was never awarded.

Within RAND, a prize is mentioned by famed mathematician Julia Robinson, in remarks concerning her research into the theory of games. "And RAND was offering a $200 prize for its solution. In my paper, 'An iterative method of solving a game,' I showed that the procedure did indeed converge, but I didn't get the prize, because I was a RAND employee."[24] Likely inspired by another problem on the list, Robinson took up the study of the TSP in 1949. Her work on the salesman is in tune with a general approach to mathematics she describes in handwritten notes from this period. "I prefer working on problems whose statement is comparatively simple but where nothing is known about what sort of methods might lead to a solution, to working on those requiring extensions of existing methods."[25] The TSP certainly fit the bill—no progress on the problem was reported in the nearly twenty years since Menger's colloquium. As

we will see in chapter 5, her contributions to the salesman provided the background for the RAND breakthrough several years later.

Whence the TSP?

In a 1949 research paper, Robinson uses "traveling salesman problem" in an offhand way, suggesting it was a familiar concept at the time. In fact, until a copy of the RAND prize list is uncovered from its likely hiding place in some archive or other, Robinson's report is the earliest known reference to the TSP by name. Robinson formulates the problem as finding "the shortest route for a salesman starting from Washington, visiting all the state capitals and then returning to Washington," matching both Flood's description and the data set used by Dantzig et al.

Robinson's language connects the TSP and the "48-states problem," but we do not know when and where the salesman name first came into play. Merrill Flood would seem to be the person to have this information, but unfortunately he does not, as he explained to Albert Tucker. "I don't know who coined the peppier name Traveling Salesman Problem for Whitney's problem, but that name certainly caught on, and the problem has turned out to be of very fundamental importance." Whatever the origin, except for small variations in spelling and punctuation, "traveling" versus "travelling," "salesman" versus "salesman's," etc., by the mid-1950s the TSP name was in wide use, and the problem was beginning to pick up its notorious reputation. The table was set for Dantzig et al.

A Statistical View

Many important problems in mathematics are attacked from all sides, sometimes without the attacking teams knowing others have joined the fray. Such is the case with the salesman problem. At about the time Flood and company were struggling with the TSP in the United States, on the other side of the world, statistician P. C. Mahalanobis took on the problem from a different mathematical point of view and with a far different application in mind.

Bengali Jute Farms

Mahalanobis is known as the Father of Statistics in India, founding both the Indian Statistical Institute and the *Sankhya* journal of statistics. One of his

Figure 2.19
Prasanta Chandra Mahalanobis. Photograph on right taken while on a farm sample survey. Courtesy Mahalanobis Museum, Indian Statistical Institute, Kolkata, India.

main interests was the development of techniques for carrying out sample surveys, and it is here he made a connection to the TSP.

A major source of revenue in India during the 1930s was obtained from its jute crop, accounting for roughly one quarter of total exports. The majority of India's jute was grown in the Bengal region and an important practical question was how to collect data to make accurate forecasts of the crop.

A complete survey of Bengali land used in jute production was impractical, owing to the fact that jute was grown on roughly six million small farms. Mahalanobis proposed instead to make a random sample survey, dividing the country into zones comprising land of similar characteristics, and within each zone selecting a random number of points to inspect for jute cultivation. A major component in the cost of making the survey would be the time spent in moving men and equipment from one sample area to the next. This is the TSP aspect of the application, to find efficient routes between the selected sites in the field. Concerning this, Mahalanobis writes the following in a 1940 research paper.[26]

> It is also easy to see in a general way how the journey is likely to behave. Let's suppose that n sampling units are scattered at random within any area; and let's suppose that we may treat each such sample as a geometrical point. We may also assume that arrangements will usually be made to move from one sample point to another in such a way as to keep the total distance travelled as small as possible; that is, we may assume that the path traversed in going from one sample point to another will follow a straight line. In this case it is easy to see that the mathematical expectation of the total length of the path travelled in moving from one sample point to another will be $(\sqrt{n} - 1/\sqrt{n})$. The cost of the journey from sample to sample will therefore be roughly proportional to $(\sqrt{n} - 1/\sqrt{n})$. When n is large, that is, when we consider a sufficiently large area,

> we may expect the time required for moving from sample to sample will be roughly proportional to $\sqrt{n}$, where n is the total number of samples in the given area.

The term *expectation* refers to the average length of the optimal tours we would see if we repeated many times the experiment of taking n random points and solving the TSP. Perhaps owing to his research interests as a statistician, Mahalanobis does not discuss the operational task of actually finding tours for specific data. He focuses instead on making statistical estimates of the lengths of optimal routes. This is quite a different angle on the problem than that taken up by researchers at Princeton and RAND.

Mahalanobis's estimates were included in the projected costs of carrying out sample surveys in Bengal, and these projections were an important consideration in the decision to implement a small test in 1937 and a large survey in 1938.

Verifying the Tour Estimates

Mahalanobis did not give a precise analysis of his TSP formula, but his research set up a nice target for further work by the statistics community. The object of this work was to learn more about tours that arise when city locations are chosen at random in a unit square, that is, each point (x, y) with both x and y between 0 and 1 is equally likely to be chosen as a sample location. In particular, what can be said about the lengths of optimal tours through such point sets?

Researchers approached the problem from two directions. Eli Marks showed, in 1948, that the expected length of an optimal tour through a random set of points is at least

$$\frac{1}{\sqrt{2}}\left(\sqrt{n}-\frac{1}{\sqrt{n}}\right)$$

and M. N. Ghosh showed, in 1949, that the expected length is at most $1.27\sqrt{n}$. For large n, these results combine to prove that Mahalanobis's intuition was correct; the expected length of the tour is indeed proportional to $\sqrt{n}$.

In his research paper on the upper bound, Ghosh made a point to comment on the operational task of producing results for specific data. "After locating the n random points in a map of the region, it is very difficult to find out *actually* the shortest path connecting the points, unless the number n is very small, which is seldom the case for a large-scale survey."[27]

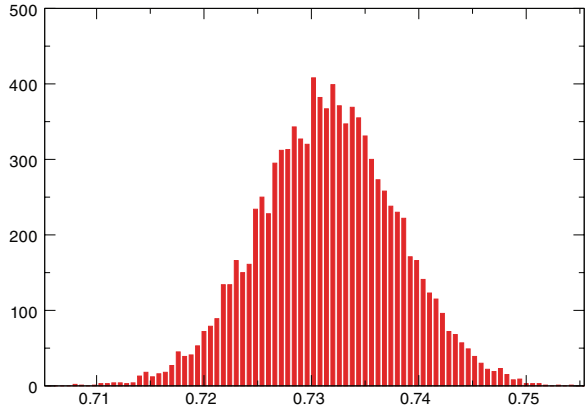


Figure 2.20
Distribution of tour lengths for 10,000 random geometric 1,000-city instances.

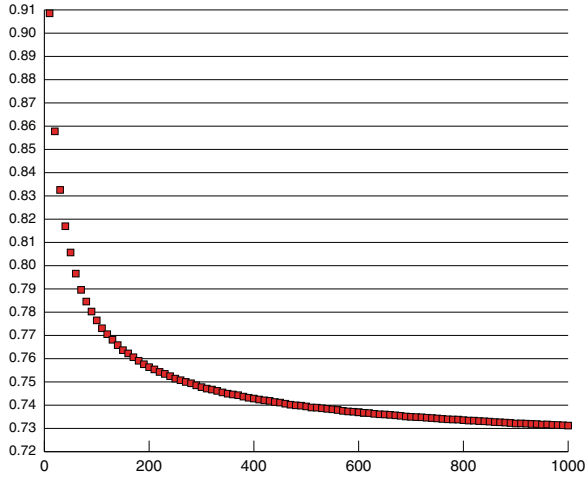


Figure 2.21
Average tour length; 10,000 samples for each n.

It is interesting that he observed the heart of the TSP challenge of finding optimal tours, apparently without connection to Menger, Whitney, and Flood.

The TSP Constant

The Mahalanobis-Marks-Ghosh result gives an estimate for the average tour length, but it does not say anything about the range of lengths we are likely to see in a series of experiments: some random point sets might have long optimal tours, while others could have tours that are quite short. This in fact does not happen, if n is reasonably large. To understand this point, examine the histogram given in figure 2.20, displaying the optimal

tour lengths divided by $\sqrt{1,000}$ for 10,000 random geometric instances, each with 1,000 cities. The results form a nice bell curve around the mean 0.7313. With only 1,000 cities there is still some variance in the tour values, but a famous theorem of Beardwood, Halton, and Hammersley, published in 1959, implies that as n gets large, the distribution of tour lengths will spike around a particular number called β, the TSP constant.[28]

An intriguing question is to determine the value of β. Its investigation has led to an important subfield of probability, but proven estimates do not come close to pinning it down. So we have a natural constant whose actual value is unknown.

In an ongoing study of β with David Applegate, David Johnson, and Neil Sloane, we have solved over 600,000,000 geometric instances of the TSP. This has given Concorde quite a workout, but the mountain of computation alone cannot prove any definitive results. Nonetheless, plots such as the one displayed in figure 2.21 strongly suggest a steady decrease in the average tour length divided by $\sqrt{n}$ as n increases, pointing toward an ultimate value of approximately 0.712 for β.[29]

3: The Salesman in Action

Because my mathematics has its origin in a real problem doesn't make it less interesting to me—just the other way around.

—George Dantzig, 1986.[1]

The name itself announces the applied nature of the traveling salesman problem. This has surely contributed to a focus on computational issues, keeping the research topic well away from perils famously described in John von Neumann's essay "The Mathematician". "In other words, at a great distance from its empirical source, or after much 'abstract' inbreeding, a mathematical subject is in danger of degeneration". Indeed, a strength of TSP research is the steady stream of practical applications that breathe new life into the area.

Road Trips

In our roundup of TSP applications, let's begin with a sample of tours taken by humans, including the namesake of the problem.

Salesmen in the Digital Age

An automobile equipped with a global positioning system (GPS) device is the mode of transportation typically chosen by local traveling salesmen. Mapping software running on the GPS unit often includes a TSP solver for small instances having a dozen or so cities, and this is usually adequate for daily trips. Detailed maps stored in the unit can be used to deliver accurate estimates of the time to travel from point to point, allowing TSP solutions to reflect actual driving conditions faced by travelers.

Alain Kornhauser of Princeton University, an expert on the application of mapping technology, described an interesting, reverse, use of GPS equipment. When a user specifies a destination with a latitude and longitude, it is sometimes impossible to project the point onto the known grid of roads and highways—there just isn't a way to get to the location. But if a package must be delivered, then a local trucker will often find a route, perhaps using a small lane that is not on the grid. In such a case, the GPS system reports back to a central server and a link is added into the grid, tracing the path traveled by the vehicle. Next time a delivery is requested for the location, the mapping software makes use of the newly inserted road.

Pick-ups and Deliveries

A common use of small-scale TSP models is the routing of buses and vans to pick up and deliver people and packages. Merrill Flood wrote that a school bus–routing problem provided his initiation into the study of the TSP. Another early team, George Morton and Ailsa Land of the London School of Economics, was drawn to the problem by a laundry-van application. In a more recent example, the firm Rapidis employed Concorde to plot routes for their customer Forbruger-Kontakt, a distributor of advertising material and samples, operating in Denmark and several other countries. The image in figure 3.1 is a drawing made from a screen dump of the routing software created by Rapidis. The route in the drawing obeys one-way streets and other travel restrictions, making the cost to travel between two points depend on the direction that is chosen.

Figure 3.1
TSP tour for deliveries by Forbruger-Kontakt. Courtesy of Thomas Isrealsen.

Meals on Wheels

A team from Georgia Tech described a successful application of a fast TSP heuristic algorithm for constructing the routes of aid workers in a "Meals on Wheels" program in Atlanta.[2] Each driver in the program delivers meals to 30 to 40 locations out of a total of 200 or so that are served daily. To construct routes for the drivers, all 200 locations are placed in a tour that is divided into segments of the appropriate lengths. The overall tour is found with the aid of the spacefilling curve illustrated in figure 3.2. Ever-finer versions of the curve will eventually include any point in the city, and the heuristic tour through the 200 locations is obtained by taking the order in which the locations appear on the curve.

The simplicity of the tour-finding method allowed the manager of the program to easily update the tour by hand as new clients joined the system and existing clients left the system. The process runs as follows. The position of a point in the tour depends only on its relative position θ on the spacefilling curve. The Georgia Tech team precomputed the value of θ for a fine grid of (x, y) locations from a standard map of Atlanta. The list of active clients was stored on two sets of index cards, one sorted alphabetically and the other stored in the tour order, that is, by increasing value of θ. To delete a client, his two cards are simply removed. To insert

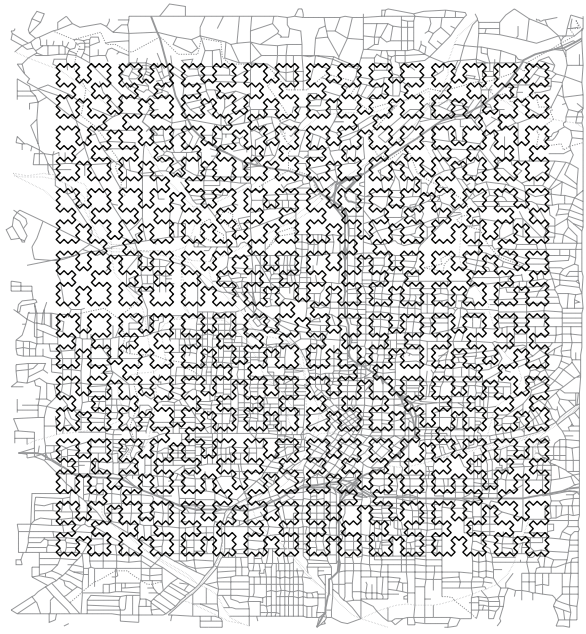


Figure 3.2
Spacefilling curve for Atlanta region. Image courtesy of John Bartholdi.

a new client, the map is used to determine the (x, y) coordinates of the client's location, the table is used to look up the corresponding value of θ, and θ is used to insert the client's card into the tour order. An ingenious, low-tech solution for a practical TSP application.

Farms, Oil Platforms, and Blue-claw Crabs

The farming study of Mahalanobis in the 1930s is an early example of the use of the TSP in planning inspections of remote sites. This type of logistical application occurs in many other contexts as well. For example, William Pulleyblank reports the use of TSP software to plan routes for an oil firm to visit a set of 47 platforms off the coast of Nigeria. In this instance, the platforms are visited via a helicopter flying from an onshore base. In another example, a group at the University of Maryland modeled the problem of scheduling boat-crew visits to approximately 200 stations in the Chesapeake Bay. The purpose of the boat trips was to monitor the blue-claw crab population in the bay; the researchers turned to the TSP after having difficulty completing trips quickly enough to permit frequent monitoring of all sites.

Book Tours

Manil Suri, the author of the novel *The Death of Vishnu* and a professor of mathematics, made the following remark in *SIAM News*.[3]

> The initial U.S. book tour, which starts January 24, 2001, will cover 13 cities in three weeks. When my publisher gave me the list of cities, I realized something amazing. I was actually going to live the Traveling Salesman Problem! I tried conveying my excitement to the publicity department, tried explaining to them the mathematical significance of all this, and how we could perhaps come up with an optimal solution, etc., etc. They were quite uneasy about my enthusiasm and assured me that they had lots of experience in planning itineraries, and would get back to me if they required mathematical assistance. So far, they haven't.

Despite the reluctance of Suri's publishers, book touring is a natural setting for the TSP.

Extra Miler Club

The motto of the Extra Miler Club is "because the shortest distance between two points is no fun." Nonetheless, members do like to plan their tours, aiming to visit all 3,100+ counties in the United States. This is not exactly a TSP, since crossing any point of a county line is sufficient, although some members prefer to visit each county seat of government.

The *Wall Street Journal* reported that one Extra Miler proposed to eat a Big Mac in each of the over 13,000 McDonald's in North America.[4] That would be a nice application of the TSP, but the club Web site reports the member "has now set forth upon a less gastronomically challenging goal."

The Iron Butt Rally

While the Extra Milers typically travel by automobile, the vehicle of choice for the 35,000-member strong Iron Butt Association is the motorcycle. One of their many challenges is the 48 States in 10 Days ride, where riders must visit all 48 continental states in the United States. Any route through the states is acceptable, but riders must obtain printed documentation, such as a gasoline receipt, verifying each state on their trip. Rider Maura Gatensby sent an e-mail letter in February 2009, asking about the city locations used in the Dantzig-Fulkerson-Johnson TSP tour.

> Most of us work on the problem by taking existing routes and trying to trim them somehow, but after reading about the existence of this problem in mathematics, I would like to make the Dantzig route my base route, and then perhaps try and reduce distance by moving some of the Dantzig locations. If the Dantzig route is not too long, I would like to just ride this route, because of its historical significance. Sometimes the shortest route isn't the "best" route, there is more poetry in walking in the footsteps of giants.

This is certainly a great use of their optimal solution.

Ms. Gatensby writes that the shortest distance known for the 48/10 ride is 6,967 miles. Although 48 cities is easy for today's TSP solvers, the problem is complicated by the fact that there are many choices for potential stops in each state. It would be an interesting challenge to find the optimal route with some fixed constraint, such as requiring each state visit to be among a list of known gasoline stations.

Figure 3.3
En route with the *Miss Izzy*. Photograph courtesy of Ron Schreck.

Flight Times

For record speed, it is hard to beat Ron Schreck, who uses the RV-8 airplane *Miss Izzy* for his tours. In 2007, Schreck had the idea to visit in a single day all 109 public airports in his home state of North Carolina. Concorde provided an optimal tour that Ron modified slightly to reach before sunrise several airports having lighted runways. His trip was made on July 4, a public holiday in the United States, which helped in avoiding delays. Schreck's total flight covered 1,991 miles in seventeen hours, with the time between landings averaging only nine and a half minutes. Landing here typically meant touching the wheels on the ground and bouncing back into the air.

Mapping Genomes

Turning away from the movement of people and vehicles we find surprising uses of the TSP model. One of the most interesting of these arises in genetics research, where a focus over the past decade has been the accurate placement of *markers* that serve as landmarks for genome maps.

A genome map has for each chromosome a sequence of markers with estimates of the distances between adjacent markers. The markers in these maps are segments of DNA that appear exactly once in the genome and can be reliably detected in laboratory work. The ability to recognize these unique segments allows researchers to use them to verify, compare, and combine physical maps created across different laboratories. It is particularly useful to have accurate information on the order in which the markers appear on the genome, and this is where the TSP comes into play.

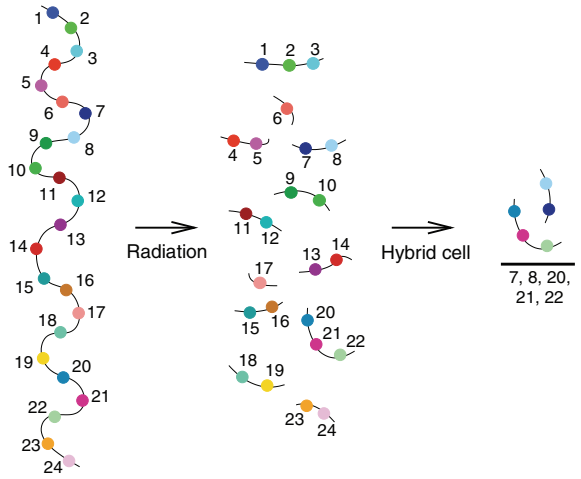


Figure 3.4
Radiation hybrid mapping.

One of the primary techniques for obtaining laboratory data on the relative position of markers is known as *radiation hybrid* (RH) *mapping*. This process exposes a genome to high levels of X-rays to break it into fragments. The fragments are then combined with genetic material, taken from rodents, to form hybrid cell lines that can be analyzed for the presence of markers. A simple illustration of the two steps is given in figure 3.4.

The central theme in RH mapping is that positional information can be gleaned from an analysis of which pairs of markers appear together in cell lines. If two markers A and B are close on the genome, then they are unlikely to be split apart in the radiation step. Thus, in this case, if A is present in a cell line, it is likely that B is present as well. On the other hand, if A and B are far apart on the genome, then we can expect to have cell lines that contain just A or just B, and only rarely a cell line containing both A and B. This positional reasoning can be crafted into a notion of an experimental distance between two markers.

Using the experimental distances, the problem of finding the genome order can be modeled as a TSP. Indeed, a genome ordering can be viewed as a path traveling through each marker in the collection. As usual, such a Hamiltonian path is readily converted to a tour by adding an extra city to permit the ends of the path to be joined.

A group at the National Institutes of Health (NIH), led by Richa Agarwala and Alejandro Schäffer, has developed methods and software for handling these genome TSP problems in practical settings, including procedures for dealing with erroneous data (a common occurrence in laboratories).[5] The NIH package uses Concorde to permit the software to

find optimal tours; it has been adopted in a number of important studies, including the construction of human, macaque, horse, dog, cat, mouse, rat, cow, sheep, and river buffalo maps.

Aiming Telescopes, X-rays, and Lasers

Although we normally associate the TSP with applications that require physical visits to remote locations, the problem also arises when sites can be observed from afar, without actual travel. A natural example is when the sites are planets, stars, and galaxies, and the observations are to be made with some form of telescope.

The process of rotating equipment into position to make an observation is called *slewing*. For large-scale telescopes, slewing is a complicated and time-consuming procedure, handled by computer-driven motors. In this setting, a TSP tour that minimizes the total slewing time for a set of observations can be implemented as part of an overall scheduling process. The cities in the TSP are the objects to be imaged and the travel costs are estimates of the slewing times to move from one object to the next.

In a *Scientific American* article, Shawn Carlson describes how a TSP heuristic came to his aid in scheduling a fragile, older telescope to image approximately 200 galaxies per night. Concerning the need for good TSP tours, Carlson writes the following. "Because large excursions from horizon to horizon sent the telescope's 40-year-old drive system into shock, it was vital that the feeble old veteran be moved as little as possible".[6] Modern telescope installations are certainly not feeble, but good solutions to the TSP are vital for the efficient use of very costly equipment.

Finding Planets

Interesting examples of the TSP have been considered in planning work for space-based telescope missions by NASA. Martin Lu of the Jet Propulsion Laboratory calls this study the "traveling planet-finder problem" since a major goal is the discovery of Earth-like planets in orbit around nearby stars.

As in the case of ground-based equipment, the TSP is used to determine the sequence of observations to be made by the telescopes. In this setting, however, the sequencing of observations is made well in advance of the mission, rather than on a nightly basis. This preplanning is due to the great amount of fuel consumed in slewing operations and to the length of time needed to study each star. Martin Lu estimates that approximately fifty stars would be observed in a three-year mission.

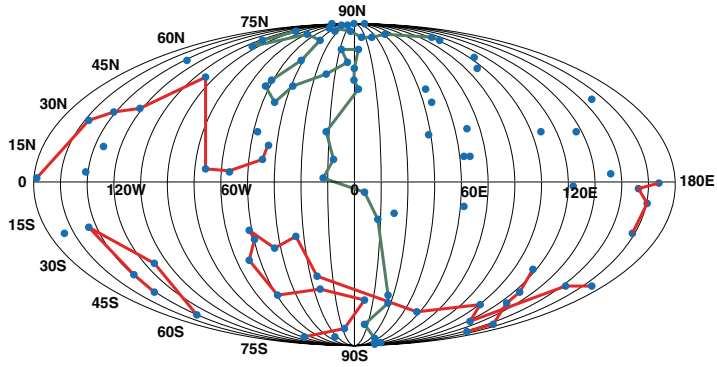


Figure 3.5
Two-occulter formation visiting 80 stars, by E. Kolemen and N. J. Kasdin.

A difficulty with observing a possible Earth-like planet is that light directly from its star washes out any image of the planet itself. A proposed solution consists of using a space telescope together with a large occulter stationed 50,000 to 100,000 kilometers away. Robert Vanderbei of Princeton University describes this as holding a giant thumb in front of the telescope's eye to block out starlight. The telescope remains in a fixed orbit, while the occulter moves from one position to another, setting up the observations.

A detailed study of the sequencing of occulter-based telescopes has been carried out at Princeton University by Egemen Kolemen and Jeremy Kasdin.[7] They use a sequence of optimization models to estimate fuel costs in moving the occulter from star to star. The image in figure 3.5 depicts their solution when a single telescope works with two occulters, alternating the observations from one to another; the colored paths represent the tours taken by the occulters. Note that sub-paths that appear to be isolated in the figure are actually connected around the back of the sphere. In this test, the solution picks out 80 of the top 100 candidate stars in a NASA target list.

X-ray Crystallography

The ground-based telescope TSP is similar to a study by Robert Bland and David Shallcross in a different domain.[8] Working with a team at Cornell University in the mid-1980s, Bland and Shallcross used the TSP

Figure 3.6
Crystal image drawn by laser.

to guide a diffractometer in X-ray crystallography. The travel costs in this case are estimates of the time for computer-driven motors to reposition sample crystals and to aim the X-ray equipment; experiments can consist of up to 30,000 observations per crystal. Bland and Shallcross reported improvements of up to 46% in total slewing time with the help of TSP methods.

Lasers for Crystal Art

The use of pulsed lasers in manufacturing settings provides another opportunity for this type of "aiming" TSP. A nice example is in the production of models and artwork burned into clear solid crystals, such as the pla85900 object produced by Mark Dickens of Precision Laser Art, displayed in figure 3.6. The focal point of a laser beam is used to create fractures at specified three-dimensional locations in the crystal, creating tiny points that are visible in the clear material. The TSP is to guide the laser through the points to minimize production time.

Dickens has adopted heuristic methods from Concorde to handle very large sets of points needed to obtain high-quality reproductions of elaborate images. This application holds a place of honor as having generated the largest industrial instances of the TSP we have encountered to date, with some examples exceeding one million cities.

Guiding Industrial Machines

In modern manufacturing, machines are often adopted to perform repeated tasks, such as drilling holes or attaching items. This is a common setting for TSP applications.

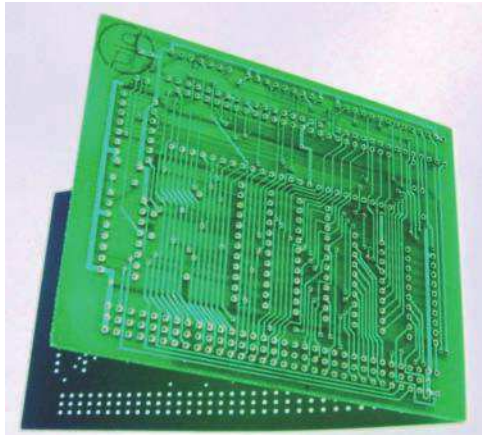


Figure 3.7
Printed circuit board with 441 holes. Photograph courtesy of Martin Grötschel.

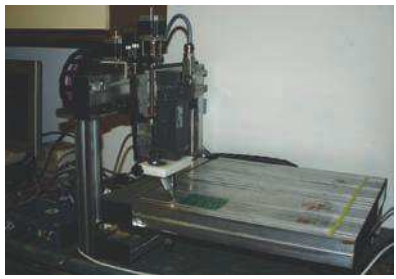
Drilling Circuit Boards

Printed circuit boards contained in common electronic devices often have numerous holes for mounting computer chips or for making connections between layers. The holes are produced by automated drilling machines that move between specified locations to create one hole after another, and a classic application of the TSP is to minimize the travel time of the drill head during the production process. Gerhard Reinelt's TSPLIB test set contains a number of examples of this type, including an instance based on the board displayed in figure 3.7.

The use of TSP algorithms has led to improvements of approximately 10% in the overall throughput of circuit-board production lines.[9] Typical problems in this class range in size from several hundred cities up to several thousand cities.

Soldering a Printed Circuit Board

Wladimir Nickel, an electronics engineer in Germany, wrote that he has adopted Concorde in a follow-up step in circuit-board production, where items are soldered onto the surface of the board. He uses a computer numerical controlled (CNC) machine, equipped with a solder paste dispenser, to print solder at specified locations. His machine is displayed in the photographs given in figure 3.8; the board being created has 256 solder locations and the TSP solution provides the quickest way to move the dispenser through the full set of points.



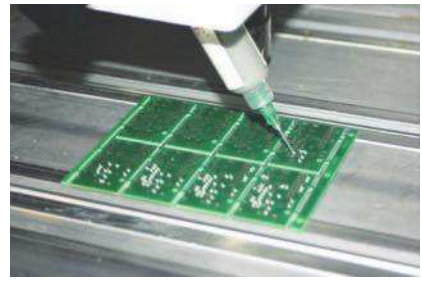


Figure 3.8
Applying solder to a printed circuit board. Courtesy of Wladimir Nickel.

Engraving Brass

Brass dies are used in printing raised images, such as those found on boxes of chocolates. The dies were once made by hand, but now they are typically engraved with heavy-duty CNC milling machines. When a CNC machine has completed the cutting of a letter or design element, the spindle is raised and the device moves to the next letter or element. Additional flexibility in this case comes from the fact that elements to be cut are not single points, so the machine can be guided to any location above the element. CNC engraver Bartosz Wucke wrote in 2008 that the application of the TSP reduced the working time by half in cases where dies have significant amounts of text or where there are abstract patterns of many points.

Customized Computer Chips

The same class of application, on a much smaller physical scale, arose in work at Bell Laboratories in the mid-1980s. Bell researchers developed a technique for the quick production of customized computer chips. The process starts with a basic chip having a network of simple building blocks, called logic gates. Portions of the network are then cut with a laser to create individual groups of gates that allow the chip to perform some described function. In this case, the TSP is to guide the laser through the locations that need to be cut. Jon Bentley and David Johnson provided fast TSP heuristic methods that lowered the slewing time by over 50% on typical examples, providing an important speedup in the production process.

This application also holds a place of honor as the source of the record 85,900-city TSP instance displayed in figure 1.7.

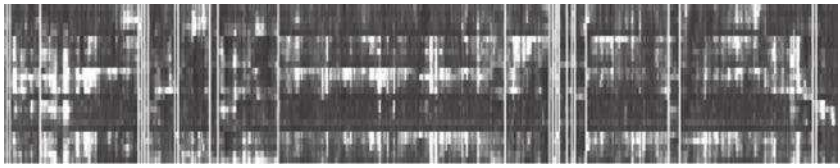


Figure 3.9
Gene expression data. Image courtesy of Sharlee Climer and Weixiong Zhang.

Cleaning Silicon Wafers

Another TSP application arises earlier in the production of computer chips. Standard chips are etched into large circular wafers of silicon and these wafers must be free of all impurities. The nanomanufacturing firm Applied Materials has a technique for cleaning defects on wafers and they have used Concorde to guide the machinery from one defect to another.

Organizing Data

Organizing information into groups of elements with similar properties is a basic tool in data mining, the process of extracting patterns from data. The TSP has been adopted in such efforts when there is a good measure of the similarity between pairs of data points. Using the similarity values as travel costs, a Hamiltonian path of maximum cost places similar points near to one another (since closely related points have high similarity measures), and thus segments in the path can be used as candidates for clusters. The final splitting into segments is typically done by hand, selecting natural breakpoints in the ordering.[10]

An elegant alternative to this two-stage method was proposed by researchers Sharlee Climer and Weixiong Zhang.[11] In their approach, $k + 1$ dummy cities are added when creating the TSP, rather than just a single city. Each of the dummy cities is assigned a travel cost of zero to all other cities. The additional cities serve as breakpoints to identify k clusters, since a good tour will use the zero-cost connections to dummy cities to replace large travel costs between clusters of points.

Climer and Zhang use their TSP+k method as a tool for clustering gene expression data, adopting Concorde to compute optimal tours and varying k to study the impact of different cluster counts. The image in figure 3.9 was produced with their software. The data set displayed in the figure consists of 499 genes from the plant *Arabidopis* under five different environmental

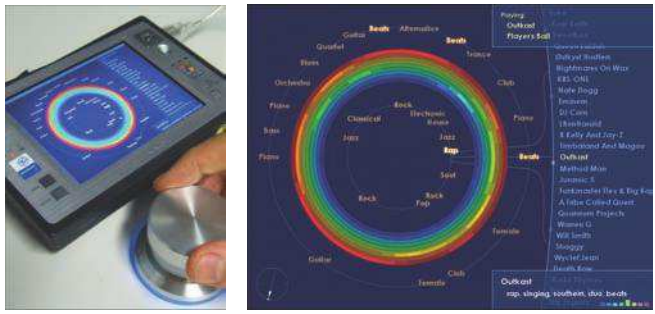


Figure 3.10
The MusicRainbow device. Courtesy of Elias Pampalk.

conditions; the shades of gray represent gene expression values; the clusters are indicated by solid white lines.

Musical Tours

The TSP has also been used to make sense out of vast collections of computer-encoded music. Elias Pampalk and Masataka Goto, working out of the National Institute of Advanced Industrial Science and Technology in Japan, created the *MusicRainbow* system to support users in discovering new artists that may appeal to their musical tastes. Pampalk and Goto took a collection of 15,336 tracks from 558 artists and developed a similarity measure between each pair of artists, computed by comparing audio properties of the tracks in the collection. The TSP was then used to arrange the artists in a circular order, such that similar artists are near to one another. In this application the cities are the musicians and the travel costs are the similarity measures.

Using the circular ordering, the music collection can be navigated by turning a knob, with artist information displayed on a computer screen. Various identifiers associated with the artists are indicated via a set of concentric colored rings corresponding to high-level classifications, such as rock and jazz. A nice feature of MusicRainbow is that all identifier information is obtained automatically via a search for Web pages, allowing the system to be easily deployed on any music collection.

Elias Pampalk was involved in a second music-related TSP application, together with colleagues Tim Pohle and Gerhard Widmer from the University of Linz in Austria. The idea this time is to organize a collection of music tracks into a circular list, such that similar pieces are near to one another. Such an arrangement allows a user to spin a wheel to pick a piece suiting their current mood, and the player follows this with a sequence of similar tracks. In their *Traveller's Sound Player* demonstration, the team used timbral similarities to measure the distance between pairs of tracks.

A test case included over 3,000 tracks, and the TSP was used to minimize the total distance in the circular order.

On a more local level, New York University's Drew Krause adopts the TSP as an aid in creating individual compositions. Smooth transitions in music are called conjunct melodies, and they are associated with pleasing sound. In Krause's process, Concorde is used to build arrangements with minimum transitions from one chord to the next; the cities are a collection of chords and the travel costs are defined as the sum of the half-step distances between the corresponding notes.

Speeding Up Video Games

Modern video games use large amounts of data to give objects in their displays an appearance of physical material, such as wood or metal. The basic components of this display data are called *textures* and libraries of thousands of textures are available, ranging from bricks to rust. Any scene in a game requires a specific set of textures to render the displayed objects, and a challenge is to get the texture data onto the video monitor as quickly as possible, to give smooth transitions from scene to scene. This is where the TSP can help.

A basic property of data access on digital video disks (DVD) is that reading items stored sequentially is much faster than accessing items from random locations. It follows that the layout of texture data on a disk can have a large impact on the time needed to render a game scene. It is highly desirable to have sets of textures used in the game residing sequentially, but this is typically not possible unless textures are duplicated on the disk, greatly increasing the storage requirement. As an alternative, the layout can be chosen such that the total number of breaks is minimized, where a break occurs when a set of textures needed in the game is stored in more than one location on the disk. If a texture set is split into k intervals, then it contributes $k - 1$ breaks to the layout. This is the same measure used in the genome-mapping application, where texture sets correspond to cell lines. In this TSP setting, the cities are the textures and the cost of travel between two textures is the number of sets that contain one of the textures but not the other. Like the genome problem, this application calls for a Hamiltonian path rather than a tour, which is handled in the usual manner via the addition of an extra city.

This application was described by Glen Miner of the Canadian firm Digital Extremes. Digital Extremes has experimented with the Concorde code for producing texture layouts, reporting significant improvements through the use of the TSP.

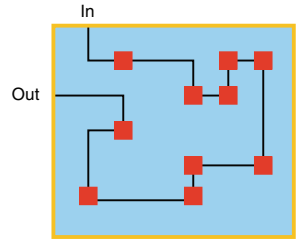


Figure 3.11
Scan chain.

Tests for Microprocessors

The computing technology firm NVIDIA recently adopted Concorde in optimizing the on-chip circuitry used to test their graphics processors. This is a common use of the TSP in the design of modern computer chips, where post-manufacturing testing is a critical step in the production process.

To facilitate such testing, *scan chains* were introduced in the 1980s to link components, or scan points, of a computer chip in a path having input and output connections on the chip's boundary, as illustrated in figure 3.11. A scan chain permits test data to be loaded into the scan points through the input end, and after the chip performs a series of test operations the data can be read and evaluated at the output end.

The TSP is used to determine the ordering of scan points to make the chain as short as possible. Minimizing the chain length helps to meet a number of goals, including saving valuable wiring space on the chip and saving time in the testing phase by allowing signals to be sent more quickly.

In most cases chip manufacturing technology allows only horizontal and vertical connections, thus the distance between two points in a scan-chain TSP is measured using paths that travel only horizontally or vertically, such as walking the streets of Manhattan. A drawing of an optimal path for a 764-city scan-chain problem is given in figure 3.12. This example

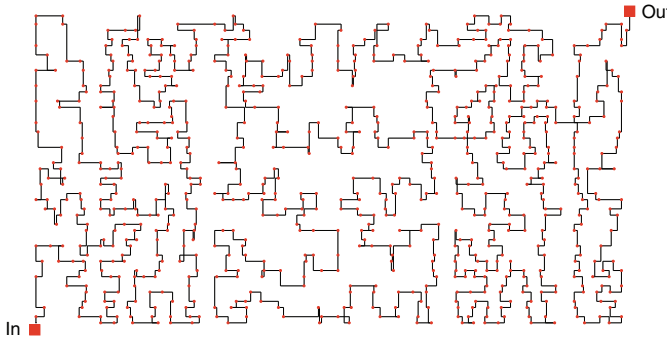


Figure 3.12
Scan-chain TSP with 764 cities.

was provided by Michael Jacobs and Andre Rohe of Sun Microsystems and it was solved using the Concorde code. To reduce the time required for testing, a modern computer chip will typically have multiple scan chains; the 764-city example was one of twenty-five chains on the given chip.

Scheduling Jobs

The German firm BÖWE CARDTEC delivers hardware and software products for managing the production of smart cards, such as credit cards and identification cards. Their customers typically produce many types of cards on the same hardware and this requires reconfiguration steps between production runs, such as a change in the ribbon color and the insertion of the correct blank cards. The setup time between different jobs is significant and reduces the overall daily production. To address this, BÖWE CARDTEC software uses the TSP to sequence jobs in an order that minimizes the total setup time: the cities are the jobs and the travel cost between jobs i and j is the time it takes to reconfigure the machine for job j after it has completed job i. The firm reports that using tours obtained with Concorde reduced the total setup time by up to 65% in typical applications, resulting in significant gains in the overall rate of production.

This type of scheduling application was first described by Merrill Flood in a lecture given in 1954. In typical examples, the setup time to move from job i to job j is different than the time to move from job j back to job i. The TSP thus takes the asymmetrical form, where the cost of a tour depends on the direction of travel.

And More

The areas of application we have described by no means exhaust the reach of the traveling salesman. Indeed, intriguing new uses for the model appear regularly in the applied mathematics literature. Successful projects that have been reported include the following:[12]

- planning hiking paths in a nature park
- minimizing wallpaper waste
- picking items in a rectangular warehouse
- cutting patterns in the glass industry
- constructing universal DNA linkers
- estimating the trenching costs for connecting a telescope array

- studying problems in evolutionary change
- assembling a genome map from a library of known subsequences
- gathering geophysical seismic data
- compressing large data sets of zero-one-valued arrays

Some of these settings are wildly distant from actual salesmen planning their tours.

4: Searching for a Tour

We do not claim that our program is infallible, but rather that it gives good answers in a computationally feasible amount of computer time.

—Robert Karg and Gerald Thompson, 1964.[1]

A salesman on the road will not be impressed by a claim of TSP unsolvability. She will nonetheless start up the car and get on with the task of visiting customers. This practical mind-set argues for an alternative approach to the problem: let's give up for now the notion that only the absolute best solution will do, and focus on delivering, as quickly as possible, a near-optimal route. Such a view opens the door to all sorts of creative ideas for getting the salesman home in time for dinner. Indeed, some of the techniques developed and employed in this branch of TSP research are now workhorses in computational science, such as simulated annealing, genetic algorithms, and local search. Tour finding serves as a sandbox for testing methods that aim to select a good solution from a large population. It is the playground of TSP studies, albeit one with serious consequences for numerous disciplines.

The 48-States Problem

The challenge of the 1940s was to route a salesman from Washington, D.C., through each of the 48 states in the United States, and back to Washington. Julia Robinson narrowed this down by proposing the salesman visit each of the state capitals, but it does not appear that anyone took the step of writing out a table of travel distances to specify completely the problem, most likely because a solution for such a large instance of the TSP appeared well out of reach.

Dantzig, Fulkerson, and Johnson clearly had a different opinion of the solvability of the challenge, and, without access to a standard set of travel

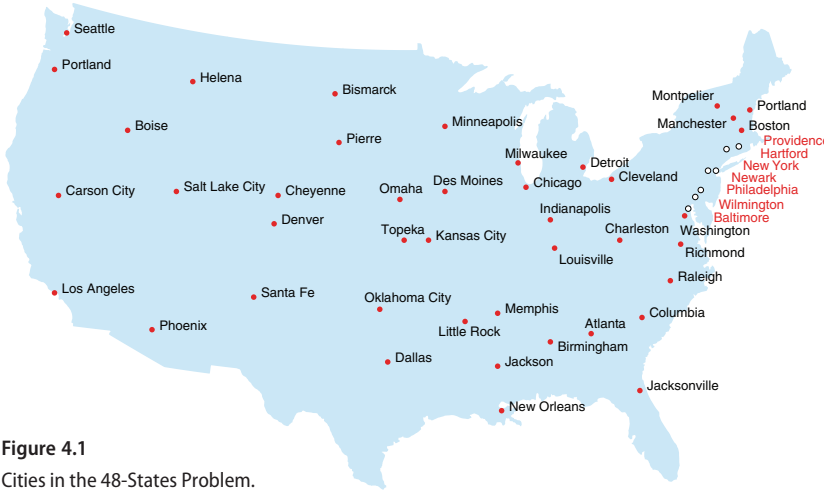


Figure 4.1
Cities in the 48-States Problem.

distances, went ahead and created their own version of the data. They opted for the quite different selection of cities displayed in figure 4.1; they hit every state, but only twenty of the locations are capitals. Despite this deviation, there is no mystery to their choice. "The reason for picking this particular set was that most of the road distances were easy to get from an atlas."[2] Fair enough, but the selection did give the researchers an immediate head start in the TSP computation: in the standard Rand McNally atlas they consulted, the shortest drive from Washington to Boston passed through seven other cities on the salesman's list. In a bit of a gamble, Dantzig et al. decided to drop these seven northeastern locations. Their reasoning is as follows. If the optimal tour through the remaining cities includes the direct link from Washington to Boston, then the RAND team could solve the original problem by rolling down the window and waving at Baltimore, Wilmington, Philadelphia, Newark, New York, Hartford, and Providence as they drove by. On the other hand, the 42-city tour might reach Washington by some other route, in which case it would have been back to the drawing board.

As you can guess by looking at the map, the optimal tour does indeed use the Washington-Boston link, and thus Dantzig et al. were justified in working with the reduced set of locations. We should point out that things are not so convenient today; using Google Maps, the direct route from Washington to Boston is 451 miles, while adding the remaining seven stops brings the trip to 491 miles. Much of the savings, however, comes from using Interstate 84 through Connecticut and into Massachusetts, and this section of the highway first opened for traffic in 1967.

The data collected from Dantzig's Rand McNally atlas is symmetric, giving distances that do not depend on the direction of travel. (Throughout the chapter we will assume that travel costs are symmetric.[3]) Dantzig et al. adjusted these values by subtracting 10 from each number, then dividing by 17 and rounding the result to the nearest integer. "This particular transformation was chosen to make the d_{ij} of the original table less than 256, which would permit compact storage of the distance table in binary representation; however, no use was made of this."[4] The full table of adjusted distances is contained in their research paper, making precise the problem that had been solved.

Pegs and String

Dantzig et al.'s data distorts somewhat the natural geometry of the problem, but the Euclidean version, where straight-line distances are used, can nonetheless be an effective tool in comparing potential tours. Indeed, the tour-finding approach adopted by the team is based entirely on straight-line approximations.

No hint as to how the USA tour was originally obtained is given in the famous Dantzig et al. paper, but in subsequent lectures Dantzig revealed that a physical device was used. The team constructed a wooden model of the problem, placing pegs at each of the 49 locations, and used a string, tied to a starting city, to wrap around the pegs and trace out a tour. Dantzig described this as a great aid in working with problems by hand; the taut string quickly measures possible routes and identifies likely continuations of subpaths. The model does not provide a solution algorithm in any sense, but with its help Dantzig et al. managed to locate the tour that later proved

Figure 4.2
A peg-and-string tour through Germany. Courtesy of Konrad-Zuse-Zentrum für Informationstechnik Berlin.

to be the optimal route through the 49 cities. Their solution weighed in at 699 units, as measured by the table. Translating this back to atlas distances, the tour covered the United States in 12,345 miles.

Growing Trees and Tours

There is something refreshing about taking a clean sheet of paper, or perhaps a wooden model, and attempting to lay out a good tour. The inclination is to pick a starting point and grow a path, adding one city after another, or, from another point of view, adding one road segment after another. Dantzig et al. relied on intuition to stretch their string from city to city, but simple algorithms can perform the task fairly well.

Nearest Neighbor

If you want to construct a tour, the simplest idea is to always drive to the closest city among those not yet visited. This *nearest-neighbor* algorithm is sensible, although it only rarely finds a shortest-possible solution.

The drawings in figure 4.3 illustrate nearest neighbor in action on the 42-city version of the USA problem, using the distances provided by Dantzig et al. The tour starts in Phoenix and spreads quickly across the southern part of the country. It looks very good for many steps, but when we arrive in the Pacific North West we have no place to go other than to travel all the way back to the East Coast to pick up cities carelessly skipped over during the first pass through the region. This is typical of the algorithm, where we paint ourselves into a corner by not looking ahead when moving from city to city. The final tour in figure 4.3 measures 1,013 units, compared with Dantzig et al.'s optimum of 699 units.

Now, if you are a devil's advocate, you can easily create a TSP instance where nearest neighbor returns a tour that is as bad as you can imagine in comparison to an optimal solution. The point to note is that the algorithm will be forced to take the last leg of the journey, back to the starting city, regardless of its travel cost. So if we increase by 1,000,000 the cost of travel between Montpelier and Phoenix, then poor nearest neighbor will still select the same tour, this time at a total cost of 1,001,013, while the optimal solution remains at 699.

This nasty modification produces a legitimate instance of the TSP, but it does not resemble the types of travel distances we see in road versions of the problem. Indeed, any reasonable instance will satisfy the *triangle inequality*: for any three cities A, B, and C, the cost to travel from A to B plus the cost

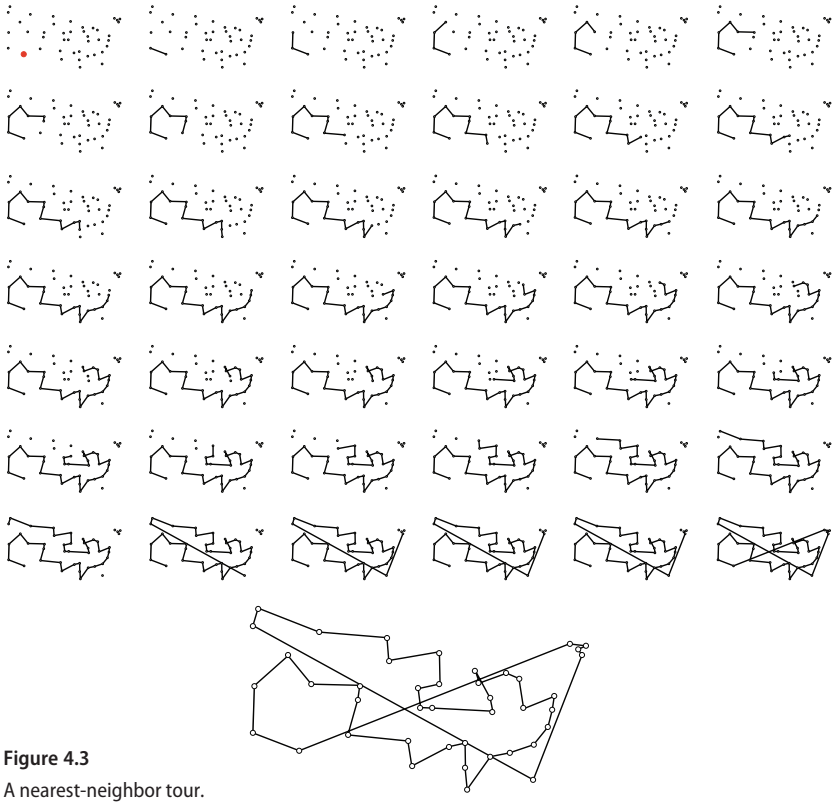


Figure 4.3
A nearest-neighbor tour.

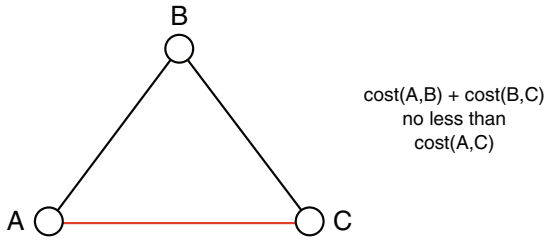


Figure 4.4
The triangle inequality.

to travel from B to C must not be less than the cost to travel from A directly to C. This condition rules out our nasty case. In fact, it can be shown that with the triangle inequality, and, as usual, symmetric travel costs, nearest neighbor will never do worse than $1 + \log(n)/2$ times the cost of an optimal tour for an n-city TSP.[5] So a fifty-city nearest-neighbor tour is guaranteed to be no longer than four times an optimal route, and a million-city tour no worse than eleven times optimal. Perhaps not a great comfort if you

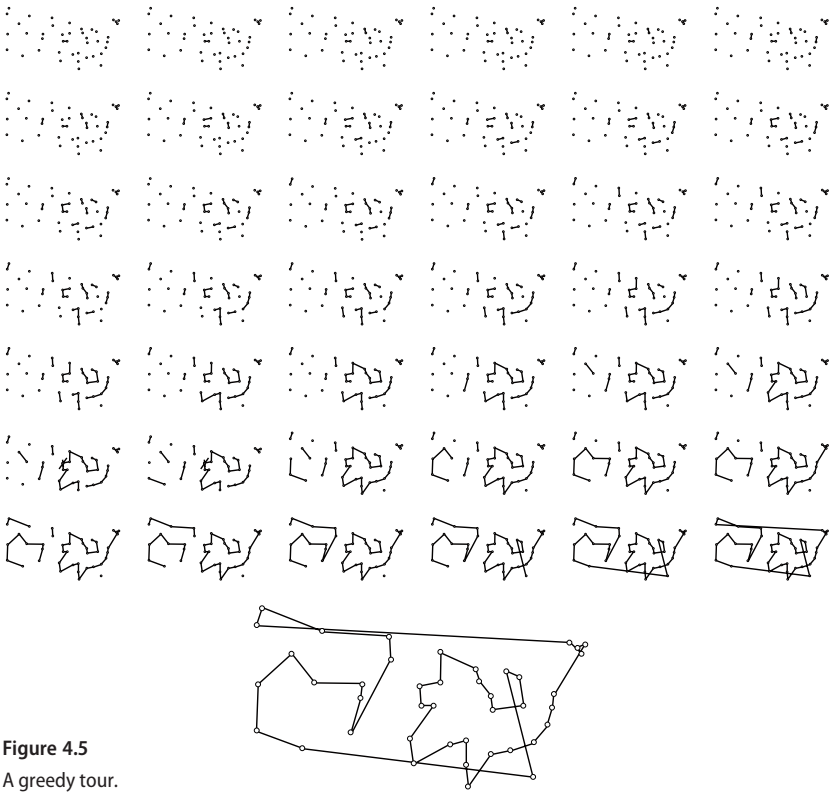


Figure 4.5
A greedy tour.

are depending on the algorithm for your travel plans, but we will see soon methods with better guarantees.

The Greedy Algorithm

Nearest neighbor grows a single path that eventually snakes around and visits every city. The method is a greedy one, extending the path in the shortest possible manner at each step. The name *greedy* is, however, reserved for an alternative algorithm that grows many subpaths simultaneously, adding shortest available road segments wherever they may be found. The operation of the algorithm on the USA problem is illustrated in figure 4.5; the subpaths grow across the map and eventually link up into a tour.

When describing TSP methods such as greedy, it is convenient to adopt graph-theory terminology, with cities being the vertices of the graph and city-to-city road segments the edges. A tour is a Hamiltonian circuit,

consisting of a selection of edges corresponding to road segments traveled by the salesman.

The greedy algorithm considers edges in a shortest-first order, adding an edge to the solution only if it joins two subpaths into a longer subpath. The progress of the algorithm looks fantastic early on; the first twenty edges or so in the USA example are very short indeed. The difficulty arises late in the process, when we are forced to accept several very long edges to make the final connections, bringing the tour length up to 995 units.

On large test instances greedy almost always significantly outperforms nearest neighbor. For example, if we drop cities randomly into a square and take straight-line travel distances, then greedy regularly finds tours of length no more than 1.15 times the optimal value, while nearest neighbor produces results in the range of 1.25 times optimal. Unfortunately, this is only an empirical observation. As far as worst-case guarantees go, greedy is known only to do no worse than $1/2 + \log(n)/2$ times optimal on instances satisfying the triangle inequality. So just a tiny bit better than the guarantee for nearest neighbor.

Inserting Cities Into a Partial Tour

An immediate question in 1954 was to determine to what extent the Dantzig et al. success relied on the fact that the peg and string model provided an optimal tour, something that could not be counted upon in further studies. In reply, young RAND associate John Robacker jumped in with a series of tests the following summer, solving several 9-city instances with the Dantzig et al. method, starting with random tours. The small examples in his study were not very convincing, but Robacker also described a general tour-finding method that could be automated when attacking large data sets.[6]

> In connection with these experiments, A. W. Boldyreff suggested an approximation procedure, the merit of which lies in its inherent simplicity and in the rapidity with which it may be applied. An application of this approximation method to the 49-city problem of [1] gave a tour of 851 units as compared with the optimal of 699 units, an error of 20%.

The idea is to start with a subtour through a small number of cities and stretch it out, like a rubber band, to enclose one additional city after another.

The Boldyreff/Robacker technique suggests a class of methods called *insertion* algorithms. The algorithms come in different flavors, *cheapest*,

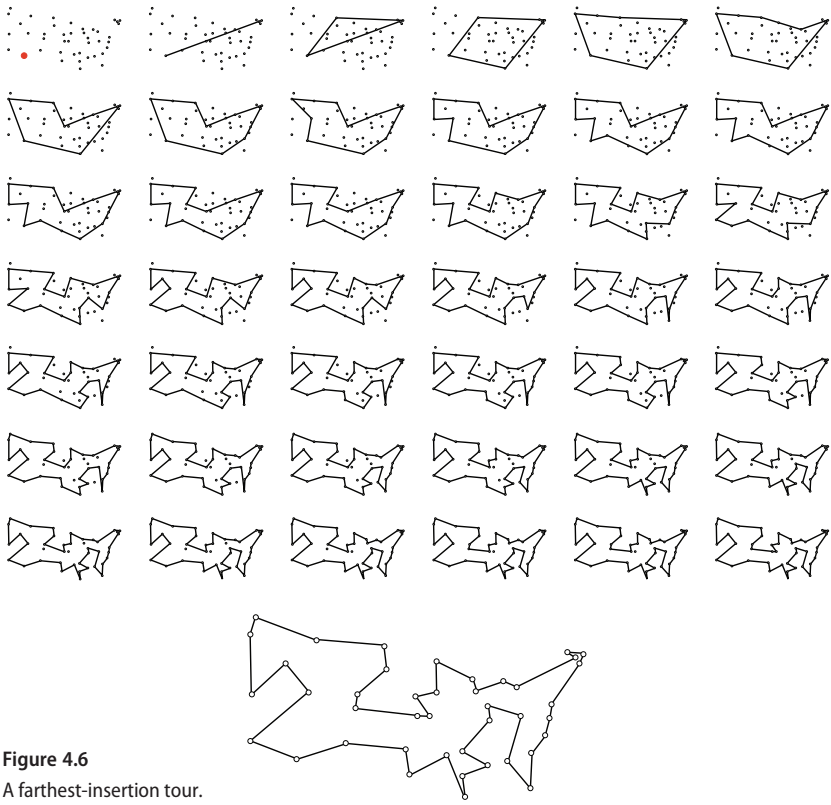


Figure 4.6
A farthest-insertion tour.

nearest, *farthest*, and *random*, depending on the rule for selecting the next city to add to the growing subtour. In each of these methods the new city is inserted into the spot that causes the smallest increase in the subtour's length.

Robacker described and tested cheapest insertion, where each new city is chosen to be the one that keeps the subtour as short as possible. Nearest insertion chooses the city that gives the shortest distance to any city currently in the subtour; farthest insertion chooses the city that is farthest from the subtour cities; and random insertion selects the next city at random from among those not yet in the subtour.

My favorite among these algorithms is farthest insertion; it obtains a good overall shape for a tour early on, and then completes the details as the last cities are added. The growth process for this variant is illustrated on the USA problem in figure 4.6, starting at Phoenix, expanding out to New Orleans, Minneapolis, and the two Portlands in stage five, and gradually building a tour of length 778.

Cheapest and nearest insertion have both been shown to produce tours no worse than twice the length of optimal solutions when the triangle inequality holds.[7] This is quite nice, but it is curious that farthest insertion comes with only a $\log(n)$ guarantee, even though it is generally the best-performing variant in practice.

Mathematical Trees

Nearest neighbor and greedy typically end up with disappointing tours, despite their beautiful-looking early selections. Greed does not pay in routing the salesman. Surprisingly, a greedy method does produce guaranteed optimal solutions to the related problem of selecting a minimum-cost set of roads to connect a group of cities. Such a minimum-cost structure for the USA data set is displayed in figure 4.7. It has length 591 units, and is thus a good bit shorter than an optimal tour.

My academic great-great-great-great-great-grandfather, Arthur Cayley, studied graphs such as that in figure 4.7. Note that the structure is connected and contains no circuits. Cayley used the wholesome name *trees* for such graphs. His mathematics writing has a nice botanical flavor, referring to vertices as "knots." "In a tree of N knots, selecting any knot at pleasure as a root, the tree may be regarded as springing from this root, and it is then called a root-tree."[8] Rather than springing from a root, we will use the structure to fashion a TSP solution, also guaranteed to be no longer than twice the length of an optimal tour. Trees, by the way, were the subject of the mysterious mathematical problem solved by Matt Damon's character in the film *Good Will Hunting*. The notes drawn by Damon in the scene displayed in figure 4.8 describe Cayley's formula for the number of trees with n vertices, together with several small examples.

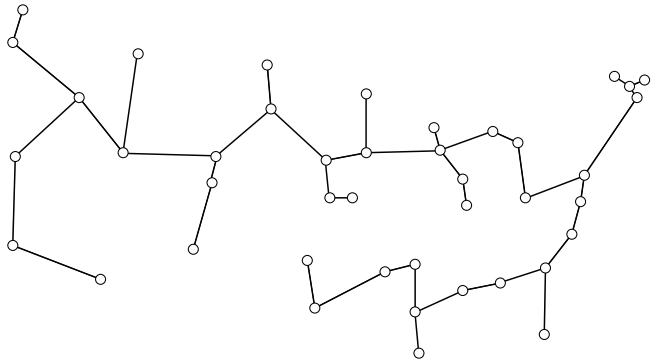


Figure 4.7
Optimal tree.

Figure 4.8
Matt Damon in *Good Will Hunting*. Copyright Miramax Films.

You likely have already convinced yourself that an optimal solution to the connection problem will indeed be a tree. The point is that we should never complete a circuit when building a network, since the ends of the final edge are already connected. The greedy algorithm in this case, working in a shortest-first order, includes an edge in the solution only if it is not possible to travel from one of its ends to the other using previously selected edges. The algorithm grows larger and larger connected components until a tree spanning the entire set of cities is produced. It is remarkable, and not too difficult to prove, that this simple method always produces a spanning tree of minimum cost.[9]

A tree is not a tour, but it does give a means to travel from city to city. One way to arrange this is as follows. Whenever we reach a new city, check if it is an end of an unexplored tree edge and, if so, choose such an edge and move along it to reach another city. If, on the other hand, we have already traveled along each of the tree edges meeting the new city, then backtrack until we return to a city that meets unexplored edges. Such a trip is called a *depth-first-search* traversal of the tree. It eventually reaches all cities and backtracks to the start.

The operation of depth-first search is illustrated on a 6-city tree in figure 4.9; the doubled edges are the ones along which we have backtracked. Notice that when the process ends we have traveled along each edge exactly

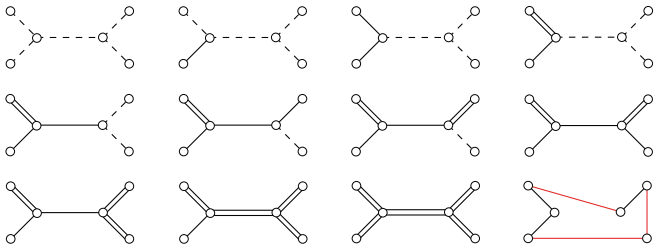


Figure 4.9
Walking along a tree to build a tour.

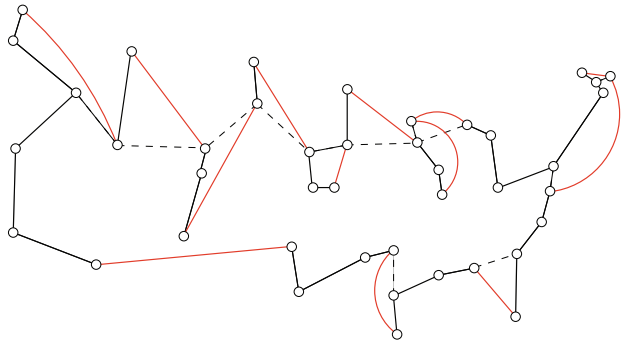


Figure 4.10
Tour created from an optimal tree.

two times, implying the cost of the trip is twice the cost of the tree. This is good news, since the cost of an optimal tree cannot be more than the cost of an optimal tour. Now, to obtain a tour from the traversal, we simply shortcut over the backtracking steps. These shortcuts are drawn in red in the final tour in figure 4.9.

Applying the algorithm to the USA problem produced the tour of length 823 units displayed in figure 4.10. The depth-first-search traversal in this case started in Phoenix; whenever there was more than one choice for a tree edge to explore, the edge leading to the subtree having the smallest number of cities was taken.

Christofides' Algorithm

Growing a tree to guide a salesman is a nice idea, but to realize its full power we need to step back and view things from the perspective of Leonhard Euler. A depth-first-search traversal of the tree is in fact a Eulerian walk through the graph obtained by duplicating the tree's edges. The duplication step ensures that each vertex of the graph meets an even number of edges, the condition unfortunately violated by the bridges of Königsberg.

Rather than duplicating the tree, we can instead add a set of edges that meets every odd vertex exactly once, where we call a vertex *odd* if it is the end of an odd number of tree edges. The resulting graph has no odd vertices, and therefore admits a Eulerian walk that can be shortcutted into a tour.

To illustrate the idea, figure 4.11 displays the twenty-six odd vertices in the USA tree, and a set of thirteen edges, in red, that meet each of these vertices exactly once. Such a set of edges is called a *perfect matching*, and Jack Edmonds showed how to compute, in polynomial time, a perfect matching of minimum cost. Edmonds's result is a milestone in the field

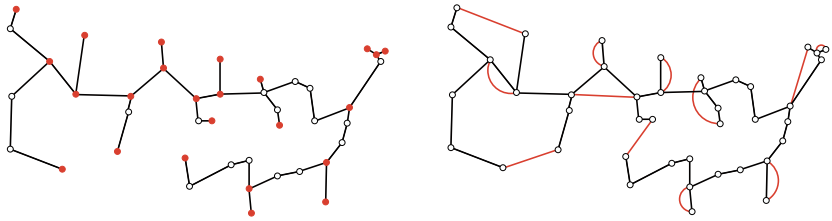


Figure 4.11
A minimum-cost perfect matching of the odd-degree vertices.

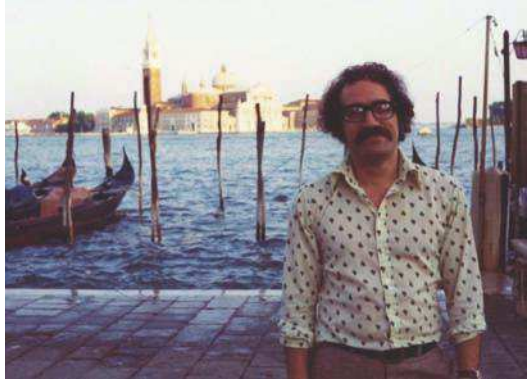


Figure 4.12
Nicos Christofides, 1976.

of optimization discussed in chapter 6. For now we note only that this is what the doctor ordered, since, as we argue below, the cost of such an optimal matching can be at most half the cost of an optimal tour. Adding the matching to the tree and shortcutting a Eulerian walk in the resulting graph, we obtain a tour of cost no more than one-and-a-half times that of an optimal solution to the TSP. This is a nice guaranteed performance, and in practice the algorithm typically produces even better solutions. Its operation on the USA problem is displayed in figure 4.13, resulting in a final tour of length 759 units.

Now, to estimate the cost of the optimal matching in general, note first that walking around a TSP tour will take us from odd vertex to odd vertex, with a few even vertices in-between. Shortcutting the even vertices results in a circuit through the odd vertices only, and such a circuit is the union of two perfect matchings, taking every other edge, starting with either the first edge or the second. One of these two matchings must have cost no greater than half the cost of the tour, and Edmonds's optimal matching can only be cheaper still. Voila! This three-step argument is illustrated in figure 4.14,

Figure 4.13
A Christofides tour.

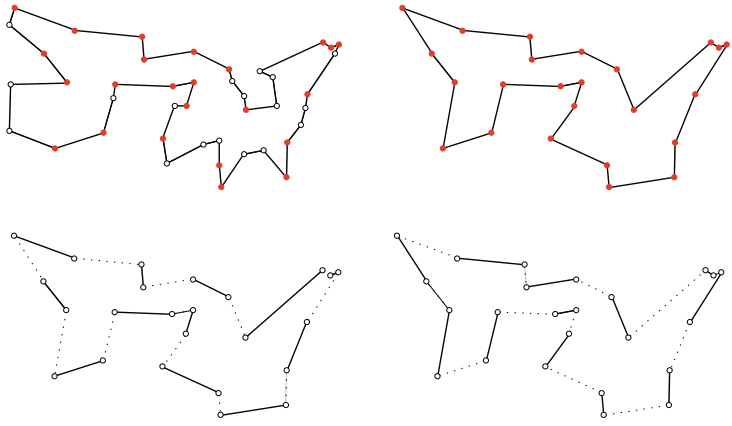


Figure 4.14
Two perfect matchings from an optimal 42-city tour.

where we start with an optimal USA tour, shortcut it to a circuit through the odd vertices, and split the circuit into two matchings.

The full process of combining Euler and Edmonds was strung together by Nicos Christofides in 1976, and it holds a place of honor in the pantheon of the TSP: no polynomial-time algorithm is known to have a better worst-case guarantee than Christofides' method.[10]

New Ideas?

The purity of laying down a tour, piece by piece, is what often attracts people and ideas to the TSP. And it is certainly a good place to gain firsthand experience with the complexity of the problem.

If you want to take a shot at the problem, then improving the performance guarantee of Christofides is a clear target. I must, however, warn you that it may be difficult to beat the factor of one-and-a-half times optimal, as we discuss in chapter 9. On the other hand, it would not be surprising to see new methods that fare well in practical competitions with existing tour-growing algorithms. Alongside the well-known methods we have discussed, TSP fans and researchers have proposed numerous alternatives, including clustering techniques, partitioning methods, spacefilling curves, and more. To date, none of these tour-growing algorithms can beat in practical computation the tour-improvement techniques we treat in the next section, but new ideas could certainly narrow the performance gap.

Alterations While You Wait

A spiffy drawing of a USA tour, displayed in figure 4.15, accompanied Martin Gardner's TSP article in *Discover*, April 1985. The combination of a popular journal and renowned problem solver brought considerable attention to the salesman, but it also stirred up trouble with readers. A close look at the drawing reveals the source of the hubbub: there are obvious shortcuts in the route through the cities!

In a phone conversation with IBM mathematician Ellis Johnson shortly after the article appeared, Gardner described that the tour was in fact obtained from the work of Dantzig et al. The problem did not lie with the tour, but rather with an overzealous editor who went ahead and shifted the locations of the cities over to the 48 state capitals. The *Discover* caption is as follows. "The traveling salesman problem is one of math's most enduring unsolved puzzles. Here's the shortest route for a salesman—or

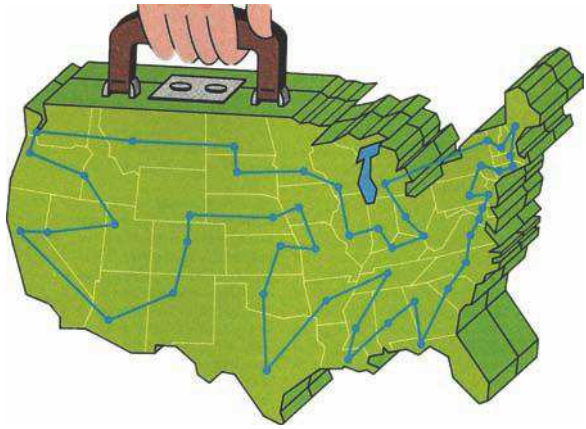


Figure 4.15
United States tour. Nina Wallace, illustrator, *Discover*, April 1985, page 87.

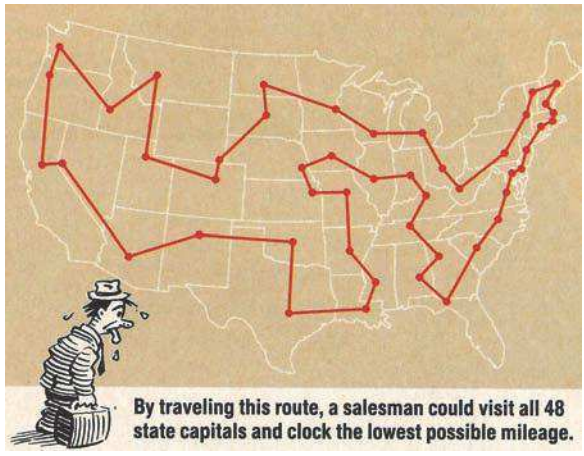


Figure 4.16
Optimal United States tour. Ron Barrett, illustrator, *Discover*, July 1985, page 16.

salesperson—visiting 48 state capitals." Bad luck. Dantzig et al.'s choice of convenience in 1954 left Gardner scrambling for a correction to his publication. This is what led to the phone conversation with Johnson, who directed Gardner to TSP star Manfred Padberg.

Padberg would certainly have been able to solve the 48-capitals problem, but he presumably could not be reached. In the end it was Shen Lin, of Bell Labs, who stepped up with a new tour, published in *Discover* four months after the original. Lin did not have an exact-solution procedure, but he was a master of tour-improvement methods.

The journal was careful in describing the tour this second time around. "Is he right? Lin is sure of it. So convinced of his results is he that he's personally offering a prize of $100 to anyone who can find a route for the salesman, using his distances between capitals, shorter than 10,628 miles."

The editors sent a table of travel distances to anyone interested in taking up Lin's challenge, but his money was safe. The tour is in fact optimal.

Exchanging Edges

Tour-improvement methods, championed by Lin, do exactly what the name implies. They take as input a tour, search for flaws, and correct them if possible. For example, the spike in the initial *Discover* tour reaching into Tennessee suggests something is wrong with that portion of the route, and the steps outlined in figure 4.18 show how to correct it. We first delete the two edges in the spike and a third edge just to the north, breaking the tour into three segments, one of which is the isolated capital of Tennessee. The segments are rejoined using three new edges, indicated in red. Since the three new edges are together much shorter than the three deleted edges, this *3-opt* move improves the tour.

Figure 4.17
Shen Lin, 1985.
Photograph courtesy
of David Johnson.

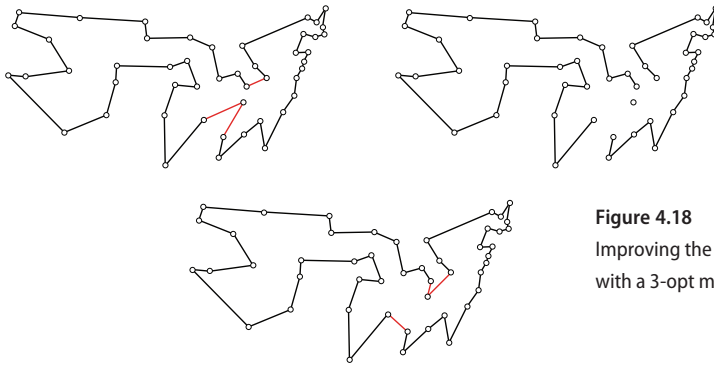


Figure 4.18
Improving the *Discover* tour
with a 3-opt move.

Lin's computation for *Discover* involved an extensive search for tour improvements, including *2-opt* moves, where two edges in a tour are deleted and the tour is reconnected with two shorter edges, 3-opt moves, and more. To explore the ideas he brought to bear on the problem, let's return to the nearest-neighbor tour constructed in our first attempt at the 42-city USA example, a fine candidate for improvement.

Perhaps the oldest theorem concerning the TSP is the fact that for Euclidean instances of the problem an optimal tour will never cross itself. The way to prove this is with a 2-opt move: replacing a crossing pair of edges will always shorten a tour. An obvious move of this type is indicated in figure 4.19. This exchange saves 31 units, bringing the total cost of the tour down to 982 units. And many more such exchanges are available.

By repeatedly making improving 2-opt moves (27 of them altogether), we arrive at the tour of cost 758 displayed in figure 4.20. At this point there exist no further improving moves with just two edges, but this simple process has brought our faulty nearest-neighbor tour to within 8% of the optimal route for the salesman.

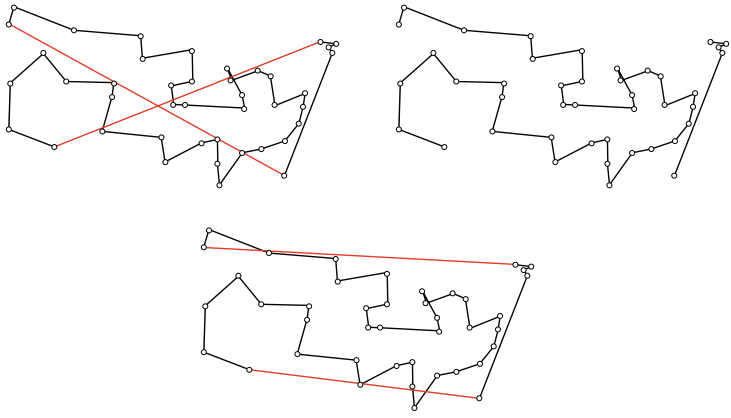


Figure 4.19
An improving 2-opt move for the nearest-neighbor tour.

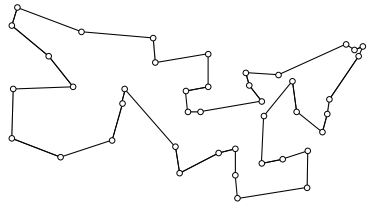


Figure 4.20
Tour with no further improving 2-opt moves.

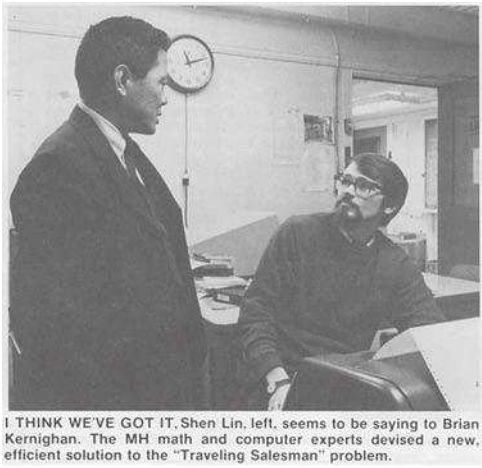

I THINK WE'VE GOT IT. Shen Lin, left, seems to be saying to Brian Kernighan. The MH math and computer experts devised a new, efficient solution to the "Traveling Salesman" problem.

Figure 4.21
Shen Lin and Brian Kernighan, *Bell Labs News*, January 3, 1977. Image courtesy of Brian Kernighan. Reprinted with permission of Alcatel-Lucent USA Inc.

Lin-Kernighan

Bashing on, we could now consider all possible 3-opt moves, checking if any might lead to further improvements. Then 4-opt moves, 5-opt moves, and so on up the line. Success with 3-opt was indeed reported by Lin in the mid-1960s, but the computational burden of searching directly for improving k-opt moves makes the process impractical for k much larger than 2 or 3. Nonetheless, Lin and computer-science pioneer Brian Kernighan accomplished this in a beautifully constructed algorithm.[11] Their work is one of the great achievements of TSP research.

The Lin-Kernighan method is elaborate, but the main idea can be gathered from the sketches in figure 4.22. In the display, the initial tour is laid out as a circle; this makes the process easier to follow, but be aware that the lengths of edges in the sketches are not meant to indicate travel costs.

The search begins by selecting a home city, as well as a tour edge meeting the selected city and a non-tour edge meeting the selected edge's other end. These are indicated by the red city, red edge, and blue edge in the second sketch. Such a triple is considered only if the travel cost of the blue edge is less than the travel cost of the red edge, with the plan being to remove reds and add blues. In the first step we can accomplish such a red-blue exchange by removing also an appropriate tour edge at the far blue end and adding the return segment to the home city, as indicated in the sketch. If this 2-opt move improves the tour, then great, we record how much it saves, but we continue the search in the hope of finding a greater improvement later on.

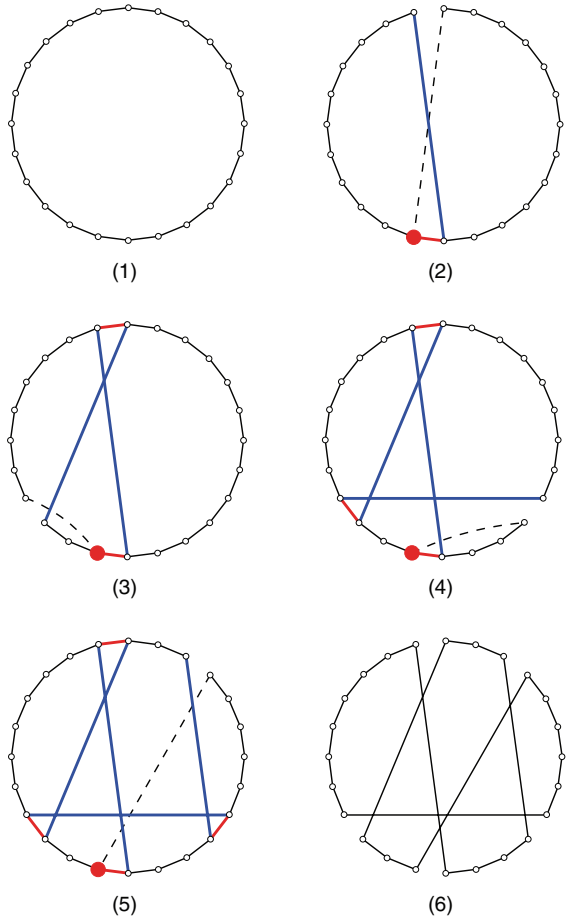


Figure 4.22
Lin-Kernighan search for a k-opt move.

The next step, illustrated in the third sketch, is to paint the second tour edge red (the one we just tried to delete), and to consider a blue alternative to the direct route home. This extension is explored only if the two blue edges together have cost less than the two red edges. Again in this case, it is possible to come home by removing a tour edge at the far blue end and adding the indicated return segment. We record this potential 3-opt move if it gives the biggest savings thus far.

The search continues to further red-blue pairs, as long as the sum of blue costs is less than the sum of red costs. If we reach the end of the line, where it is no longer possible to add another pair of edges, then we backtrack and explore alternative blue candidates at earlier levels. Eventually we halt the process, either due to time considerations or by running out of edges to consider.

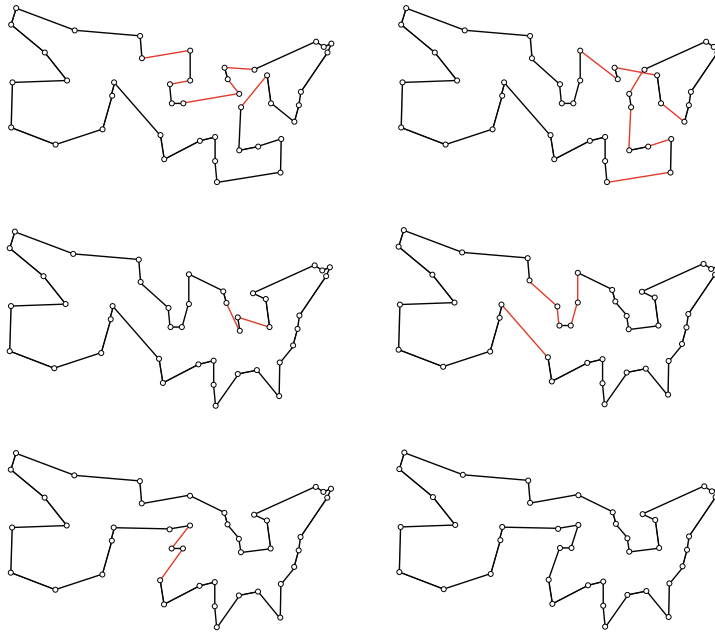


Figure 4.23
Five iterations of Lin-Kernighan.

At the end of the search we take the recorded move yielding the biggest savings, apply it to our tour, and begin again from the newly improved solution. If we failed to find any improving moves, then we return to our starting tour, select a new home city, and attempt another search.

Red-blue, check the home route. Red-blue, check the home route. Sounds easy enough, but there are plenty of devils in the details. Fortunately, together with great computational results, Lin and Kernighan layed out a crystal-clear exposition of their many ideas for implementing and enhancing the search algorithm. With their original paper as a guide, over the past forty years Lin-Kernighan has been engineered to precision, with current implementations capable of producing very good tours to huge TSP instances, having ten million cities and more.

The operation of Lin-Kernighan on the USA data set is illustrated in figure 4.23, starting with the tour obtained from repeated 2-opt moves. The algorithm finds an optimal solution in five iterations; in each step the edges colored red are those that are deleted from the tour.

It should not come as a surprise that the original computer code of Lin and Kernighan also makes short work of the USA example, using random starting tours. "The probability of obtaining optimum solutions in a single

trial is close to 1 for small-to-medium problems, say up to the 42-city problem."[12] It is remarkable, however, that their basic method, designed for instances with only several hundred cities or fewer, has served as the cornerstone for the majority of the most successful TSP heuristic methods developed in the past several decades, even as much larger examples have been tackled.

We must note that the great practical performance of k-opt methods is unfortunately not accompanied by great worst-case guarantees. For example, repeatedly making improving 2-opt moves is only guaranteed to produce a solution no worse than $4\sqrt{n}$ times longer than an optimal tour on instances satisfying the triangle inequality.[13] This is the dark side of Lin-Kernighan, but don't be overly concerned when applying the algorithm: this king of methods typically produces very good solutions indeed.

Lin-Kernighan-Helsgaun: LKH

The long reign of Lin-Kernighan in practical computation has been aided by a steady stream of enhancements supplied by the research community. Most of these are tweaks of the original ideas, but computer scientist Keld Helsgaun came along with a bombshell in 1998.

Helsgaun's main contribution was a reworked version of the core search engine, something that had remained basically intact for twenty-five years. Whereas standard Lin-Kernighan can be viewed as a search for a sequence of 2-opt exchanges that taken together result in an improving k-opt move, the new method searches for a sequence of 5-opt exchanges. That is, rather than adopting the step-by-step red-blue search, Helsgaun devised a scheme to consider ten edges at a time, five reds and five blues.

Ten edges. The first thing you should think when you see this is "that's a lot of edges." Indeed, looking at every possibility for five reds and five blues would slow the algorithm to a crawl. To get around this, Helsgaun limits his search to those sets of reds and blues that could potentially be created by a step-by-step red-blue search, if we ignored the condition that the blues must have cost less than the reds at each step. By considering any such *sequential* 5-opt exchange in a stroke, Helsgaun's method can explore improving moves that simply cannot be found by the standard algorithm.

The 5-opt moves, combined with a bag of assorted tricks, allowed LKH to set a new standard in tour finding. "For a typical 100-city problem the optimal solution is found in less than a second, and for a typical 1000-city problem [the] optimum is found in less than a minute."[14] This was an amazing jump in practical performance, in a field of study considered to be quite mature at the time.

Pancake Flipping, Bill Gates, and Big LKH Steps

When news broke that Helsgaun was putting up improved tours for a number of well-known challenge problems, there was plenty of speculation as to how he was able to successfully employ the 5-opt strategy in practical computation. To understand this, I must point out that in the twenty-five years between Lin and Kernighan's research paper and the announcement of LKH, there had been only a small handful of efficient computer codes implementing the standard algorithm. The Lin-Kernighan search method, although well described, is difficult to convert into software that can be run on large data sets.

Lin-Kernighan may be difficult, but LKH would appear to be impossible. Indeed, a great feature of working with a red-blue sequence is that at each step there is only one way to come home. In other words, if we remove two edges from a tour, then there is a unique way to hook up the resulting subpaths to obtain a new tour. A quick calculation shows that LKH, on the other hand, must handle 148 possibilities for joining up the five subpaths involved in a sequential 5-opt move. So 1 versus 148, or difficult versus very, very difficult.

Helsgaun's secret was revealed when he made his entire computer code available to researchers. Going through his files, Dave Applegate and I realized that in fact there was no stealthy method: the code contained a full listing of the 148 cases, independently covering each possibility. Helsgaun had put in a Herculean effort to write a correct and efficient code to implement an extremely complex algorithm.

Helsgaun's code and the performance of LKH were exciting, but it left one wondering if moving up to 6-opt exchanges might be better yet. Dave wrote a small computer code and calculated that sequential 6-opt moves created 1,358 possibilities for reconnecting a tour. That would be daunting enough, but why stop at 6-opt? Well, by the time we get to 9-opt there are a whopping 2,998,656 cases that must be treated. That would be a job indeed.

Not all was lost, however. Dave's code was able to list the reconnection tasks that must be handled, one by one. And an examination of LKH showed a regular pattern in the instructions needed to reconnect the tour. Combining these, we were able to create a computer program that could produce the actual computer code to handle a k-opt move, for any value of k. A computer code building a computer code.

This sounds good, but it resulted in lots of code: 6-opt, 120,228 lines; 7-opt, 1,259,863 lines; and 8-opt, 17,919,296 lines. This was all in the C programming language. Although difficult to compile into a machine workable form, the codes did run and produce interesting results. But 8-opt

as a limit was no more satisfying intellectually than 5-opt, and generating the full list for 9-opt was out of the question.

Dave kept up his courage. He had the idea that if we could make the generating code more efficient, then there would be no need to write out the full list of cases. The code-generation method could instead produce the steps needed to handle each case on-the-fly during an execution of a *k*-opt search. The method would still be limited by the computing time required to execute the search steps, but it potentially permitted the use of much larger moves.

Speeding up these code-generation calculations is closely related to the pancake-flipping problems famously studied by Microsoft's William Gates and TSP expert Christos Papadimitriou, while Gates was an undergraduate student at Harvard University. A flip of a top portion of a stack of pancakes corresponds to a reversal of a subpath in a tour, which is what happens in a 2-opt move. An implementation of a *k*-opt code generator calls for an algorithm to find a minimum number of flips to rearrange a tour in the order produced by a *k*-opt move, and this is a variant of the Gates-Papadimitriou work.[15] We managed to get this running, resulting in an efficient on-the-fly search mechanism for sequential *k*-opt.

Helsgaun incorporated similar ideas into a powerful upgrade to his LKH code, allowing users to specify the size of moves that will be strung together. Demonstrating the reach of the new software, Helsgaun employed 10-opt moves in a computation on a 24,978-city Sweden data in 2003, producing a tour that was shown to be optimal in the following year.

Borrowing from Physics and Biology

Taking a big picture of tour finding, viewing the TSP as just one example of a general search problem, proves to be useful both in finding good tours and in devising multipurpose techniques. The idea is to produce *metaheuristics*, that is, heuristic methods for the design of heuristic methods. The general nature of this work has brought in researchers from fields of science to join in the hunt for good tours.

Local Search and Hill Climbing

A useful analog in this arena is to think of tours as lying on a landscape, with the elevation of each tour corresponding to its quality. The type of picture to have in mind is one like the Gasherbrum group of mountains displayed in figure 4.24: good tours correspond to peaks of the mountains,

Figure 4.24
Gasherbrum group. Image by Florian Ederer.

with the optimal tour lying on top of mighty Gasherbrum II. A heuristic algorithm can be viewed as moving through the landscape in search of high land.

For this picture to make sense there should be a notion of when two tours are located near to one another. This is typically handled by creating *neighborhoods* around each tour. For example, two tours can be defined to be neighbors if one can be reached from the other via a 2-opt exchange, or via an exchange found by Lin-Kernighan. Large neighborhoods are useful for navigating around a landscape, but they should be constructed so that algorithms can view and evaluate neighbors.

Tour-improvement methods, such as repeatedly making improving 2-opt moves, are often called *hill-climbing* algorithms, since they can be viewed as walking up a sequence of neighboring tours, always moving to higher ground. At each step we do a *local search* for a nearby higher point. If we are thorough in our search, then the algorithm will terminate at a peak, or at least a plateau, since at this point all local moves will be either downhill or flat. A full run of the algorithm begins at the point corresponding to the starting tour and then scoots up a slope to reach a local peak.

Note that the choice of a starting tour can determine the fate of a hill-climbing approach: if the starting tour lands midway up a small hill, then the algorithm will be limited to reaching the modest-quality tour associated with the hill's peak. For this reason Lin and Kernighan proposed to carry out repeated runs of their algorithm from random starting solutions. The idea is to throw darts into the landscape. If we toss enough darts, then there is a decent chance of hitting a slope leading to a peak of good height.

Random tours provide a nice distribution of darts, but they have the disadvantage of typically starting far down in valleys, due to their poor tour

quality. For a large TSP instance it can take a long time to walk from a valley to a peak. A compromise approach is to use nearest-neighbor darts, gaining randomization from the selection of the starting city.

Simulated Annealing

In *simulated annealing* heuristics, the hill-climbing strategy is relaxed to allow the algorithm to accept with a certain probability a neighbor that is worse than the current solution. At the start the probability of acceptance is high, but it is gradually decreased as the run progresses. The idea is to allow the algorithm to jump over to a better hill before switching to a steady climb.

The paper of Scott Kirkpatrick, Daniel Gelatt, and Mario Vecchi that introduced the simulated-annealing paradigm studied the TSP, reporting on heuristic tours for a 400-city problem.[16] The authors of the paper write that the motivation for the method comes from a connection with statistical mechanics, where annealing is the process of heating a material and then allowing it to slowly cool to refine its structure.

For the salesman, the achievements of the paradigm have to date been rather modest. But as a general search tool simulated annealing has been a spectacular success. Google Scholar lists over 18,700 citations to the original research paper, an almost unheard of number.

Chained Local Optimization

The greatest impact of simulated annealing on current tour-finding methods is perhaps not the technique itself, but rather the fact that it brought the thinking of physics into the TSP arena. Indeed, it was a second major contribution from physicists that first pushed computational results beyond the limits of repeated Lin-Kernighan.

In the late 1980s, Olivier Martin, Steve Otto, and Edward Felten, from the physics department at Caltech, proposed an alternative to the dart-throwing strategy. The idea is to take advantage of the fact that a strong local-search algorithm, such as Lin-Kernighan, will typically take us up into the high-elevation region of the tour landscape. Rather than starting a second run of the algorithm from a random location, Martin et al. suggest we first look around our current peak to see if there might not be a way to jump over a few local barriers to reach a new slope to take us to an even better location.

The specific proposal is to *kick* the Lin-Kernighan solution to obtain a new starting tour, rather than throwing a dart. The overall process repeats

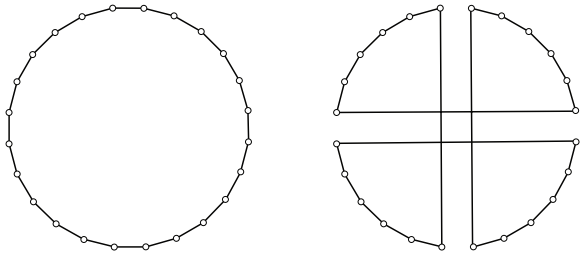


Figure 4.25
A double-bridge kick.

this many times, replacing the solution whenever we reach a better tour. For the method to work, a kick must take the solution out of its neighborhood, so it should be a modification that Lin-Kernighan cannot easily undo. Martin et al. found that a random 4-opt exchange of the type indicated in figure 4.25 does the job nicely.

The resulting algorithm is dubbed *Chained Lin-Kernighan* and its performance is outstanding. The stars were aligned for this idea. First, Martin et al.'s intuition was correct: visiting the nearby region via the kicking mechanism is a better way to sample the peaks in the landscape; we use Lin-Kernighan itself to guide us to the highest elevations. Second, the reapplication of Lin-Kernighan to a kicked tour runs much more quickly than an application to a random tour. This is simply due to the fact that much of a kicked tour remains in good shape, so the algorithm does not need many iterations to reach a locally optimal result.

For most of the 1990s, implementations of Chained Lin-Kernighan ruled the world of tour finding. The version included in the Concorde code routinely finds, in one or two seconds, solutions within 1% of the cost of optimal tours for instances with up to 100,000 cities. For even better solutions, one can turn to LKH, but Chained Lin-Kernighan remains dominant on very large data sets. For example, the plot in figure 4.26 shows the results of a run on a 25,000,000-city Euclidean instance, with city locations having integer coordinates drawn at random from a 25,000,000 × 25,000,000 square. In eight days, on a computer from the year 2000, a tour that is approximately 0.3% greater than optimal was found.[17]

Genetic Algorithms

An alternative to the landscape view is to consider a salesman's route as a living organism, mutating and evolving over time. This way of thinking is taken up in a class of methods known as *genetic algorithms*, inspired by John Holland's landmark book *Adaptation in Natural and Artificial*

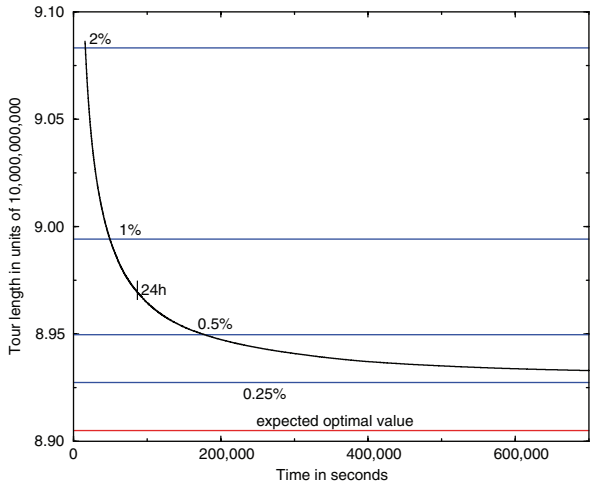


Figure 4.26
Chained Lin-Kernighan on a 25,000,000-city Euclidean instance.

Systems published in 1975.[18] Holland did not treat the TSP, but his ideas quickly made their way into the tour-finding literature.

A general outline of a genetic algorithm, as applied to the salesman, is the following. We begin by generating a starting population of tours, say by repeatedly applying nearest neighbor with random starting cities. In a general step, we select some pairs of members of the population and *mate* them to produce *child* tours.[19] A new population of tours is then selected from the old population and the children. The process is repeated a large number of times and the best tour in the population is chosen as the winner.

The spirit of genetic algorithms is to mimic evolutionary processes found in nature. The analogy is fun, but keep in mind that merely adopting the language of Darwin does not imply we end up with a good tour. Indeed, early genetic algorithms for the TSP were not especially successful, even while restricted to very small instances of the problem. But the idea of maintaining a population of tours has considerable merit and the general approach can be crafted into very strong heuristics, particularly in combination with local-search procedures.

The genetic-algorithm outline leaves plenty of freedom for selecting methods to evolve a tour population. Besides the mating process, we also get to choose a *fitness measure* for selecting the next population. Some cool ideas have been developed for such measures, seeking to balance the quality of solutions with the need for a diverse population.

For mating itself, early schemes attempted to find subpaths in one parent tour that could be substituted for subpaths in the other parent. This

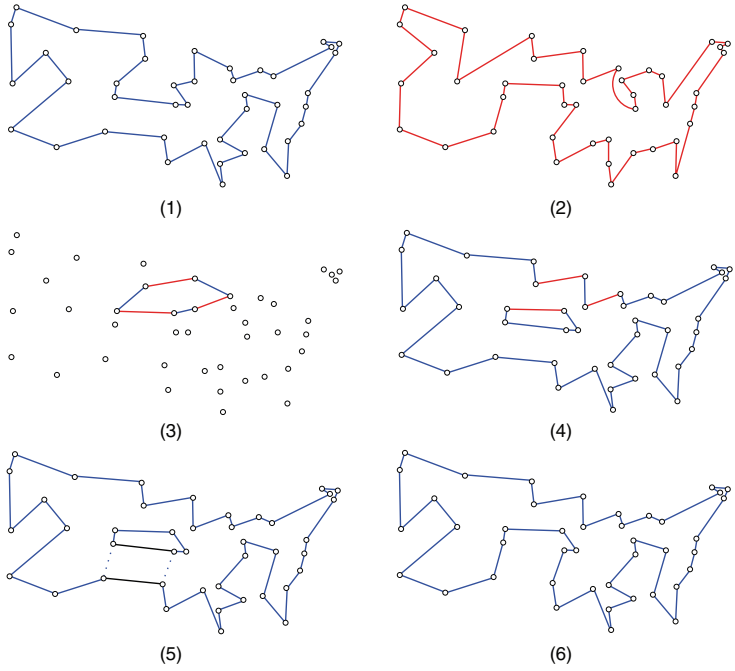


Figure 4.27
Mating two tours.

was rather restrictive, particularly in larger instances. A more successful approach is to create a new tour by choosing a subpath in parent A and extending it to a tour using, when possible, edges of parent B or edges of parent A, with preference given to the edges of B. Another mating technique, known as *edge-assembly crossover (EAX)*, is illustrated on a pair of USA tours in figure 4.27. To combine the blue and red solutions in the example, we form the graph consisting of the union of their edges and select a circuit, displayed in the third sketch in figure 4.27, that alternates between blue and red. We then delete from the blue tour each of the circuit's blue edges and add to the blue tour each of the circuit's red edges, as illustrated in the fourth sketch. The process creates subtours that are combined into a tour via a 2-opt move, displayed in the fifth and sixth sketches in the figure.

The EAX mating scheme was adopted by Yuichi Nagata in one of the most successful tour-finding procedures proposed to date.[20] His implementation relies on a very fast implementation of EAX, allowing the algorithm to proceed through many generations of tours. Among Nagata's

achievements is his discovery of the best-known tour for the 100,000-city Mona Lisa TSP.

Ant Colonies

At some point in your life you likely had the misfortune of losing food to a group of hungry ants. Typically the pests arrive in your home or garden via a long thin train of individuals, constantly moving back and forth in a nearly straight line. A single ant moves haphazardly, but the entire group, communicating via pheromone trails, finds an efficient route. This collective behavior is the inspiration for a class of TSP heuristics known as *ant-colony optimization* (ACO).

The leader of ACO research is Belgium's Marco Dorigo, who developed the ideas in his 1992 Ph.D. thesis.[21] His algorithms work with a small army of ant agents moving along the edges of a graph. Each agent traces out a tour, selecting at each new vertex an edge chosen among those leading to vertices not yet visited. The key to the process is the selection rule, which makes use of a *pheromone value* associated with each edge; if an edge has a high pheromone value then it has a high probability of being selected. After the agents have all completed tours, the pheromone values are adjusted using a rule that adds values proportional to the lengths of the computed tours; edges in good tours get their values increased more than those in poor tours.

The approach is both intuitive and appealing, but thus far ACO has not proved to be competitive with Lin-Kernighan-based methods. In recent years, however, the paradigm has been applied effectively to problems in

Figure 4.28
Ants working on the TSP. Image by Günter Wallner. Originally appeared in the book *Bilder der Mathematik* by Georg Glaeser and Konrad Polthier.

other areas, such as scheduling, graph coloring, classification, and protein folding. The active research topic is a good example of how a focus on the salesman can lead researchers to interesting general-purpose methods for attacking optimization problems arising in diverse applications.

And Many More

We have touched on only the best-performing applications of metaheuristic ideas for the TSP. Other schemes include neural networks, tabu search, and honeybee models, to name just a few. If you have a general search mechanism in mind, the TSP is a great place to develop, polish, test, and compare your strategy, even if your planned domain of application is far away from the humble routing of a salesman.

The DIMACS Challenge

The breadth of activity in tour finding is a strength of the area, but it has in the past led to misunderstandings concerning the state of the art. Indeed, in the 1980s research papers appeared in premier scientific journals, such as *Nature*, describing computations on TSP instances having 30 or 50 cities. The reported results were typically weak approximations, at a time when Lin-Kernighan could reliably deliver optimal solutions in a blink of an eye, and Martin Grötschel and Manfred Padberg were tackling instances with hundreds of cities via exact methods.

This difficulty was addressed by two important events in the following decade. The first of these was the TSP 90 conference held at Rice University's Center for Research in Parallel Computing. The organizers brought together exact-solution experts such as Grötschel and Padberg, together with tour-finding teams from around the world. An important outgrowth of the meeting was the establishment of the TSPLIB collection of test problems by Gerhard Reinelt from the University of Heidelberg. Reinelt's library was published in 1991, containing over 100 challenge instances of the TSP gathered from academic and industrial sources. The TSPLIB collection provides a common test bed for researchers around the world and across academic disciplines.[22]

The second event was the DIMACS TSP Challenge, led by David S. Johnson of AT&T Research. DIMACS is the short name for the Center for Discrete Mathematics and Theoretical Computer Science, housed at Rutgers University. In the 1990s DIMACS ran a series of implementation challenges, the best known of which is the TSP Challenge.[23]


Figure 4.29
Left: Martin Grötschel, Gerhard Reinelt, and Manfred Padberg.
Right: Robert Tarjan, Dorothy Johnson, Al Aho, and David Johnson.

> One goal of this Challenge is to create a reproducible picture of the state of the art in the area of TSP heuristics (their effectiveness, their robustness, their scalability, etc.), so that future algorithm designers can quickly tell on their own how their approaches compare with already existing TSP heuristics.

DIMACS made a call to the world's tour finders, and the world responded with 130 different algorithms and implementations. A great outcome of the challenge is a Web site that allows for direct comparisons between methods. The results are also gathered together in a very nice survey paper by Johnson and co-organizer Lyle McGeogh.[24]

Johnson's efforts in organizing the challenge, as well as his own detailed computational studies of tour-finding methods, have been a great force in shaping the current area of algorithm engineering. In 2010 he received the Knuth Prize from the Association for Computing Machinery, cited for his contributions to the theoretical and experimental analysis of algorithms. A well-deserved recognition for one of the world's leaders in TSP research.

Tour Champions

Heuristic methods must strike a balance between running time and tour quality. At the highest end of the scale we are willing to spend enormous amounts of time to deliver the best solution that is practically possible. This is Formula One racing, with participants in a no-holds-barred contest to push down the lengths of best-known tours through challenge data sets.

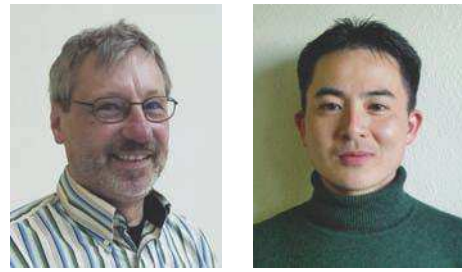


Figure 4.30
Left: Keld Helsgaun.
Right: Yuichi Nagata.

The world champions in this area are without a doubt Keld Helsgaun of Denmark and Yuichi Nagata of Japan. Helsgaun's LKH code has been the gold standard in tour finding since its introduction in 1998, and he has continued to extend and improve his algorithm with many new ideas. Helsgaun is the current holder of the best-known tour in the World TSP challenge, he provided the optimal tour for the record 85,900-city TSP, and his name peppers the leader board for the *VLSI Test Collection*.[25] Not to be outdone, Nagata's implementation of a genetic algorithm for the TSP has produced the best-known tour in the Mona Lisa TSP challenge as well as record solutions for the two largest examples in the *National TSP Collection*.[26] If you want a good solution to a large problem, these are the people to call.

5: Linear Programming

The development of linear programming is—in my opinion—the most important contribution of the mathematics of the 20th century to the solution of practical problems arising in industry and commerce.

—Martin Grötschel, 2006.[1]

Selecting the best tour through a set of points and knowing it is the best is the full challenge of the TSP. Users of a brute-force algorithm that sorts through all permutations can be certain they have met the challenge, but such an approach lacks both subtlety and, as we know, practical efficiency. What is needed is a means to guarantee the quality of a tour, short of inspecting each permutation individually. In this context, the tool of choice is *linear programming*, an amazingly effective method for combining a large number of simple rules, satisfied by all tours, to obtain a single rule of the form "no tour through this point set can be shorter than X." The number X gives an immediate quality measure: if we can also produce a tour of length X then we can be sure that it is optimal.

Sounds like magic, but linear programming is indeed the method adopted in Concorde and in all of the most successful exact TSP approaches proposed to date. Moreover, its application to problems beyond the TSP has made it one of the great success stories of modern mathematics.

General-Purpose Model

The tale of linear programming has a nice start, with a young George Dantzig arriving late for a class given by Jerzy Neyman at the University of California at Berkeley in 1939. The first-year graduate student hurriedly copied down two problems he found written on the board and turned in solutions several days later. "To make a long story short, the problems on