Copyrighted Material

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 officia 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 officer 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 fin   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.

Copyrighted Material

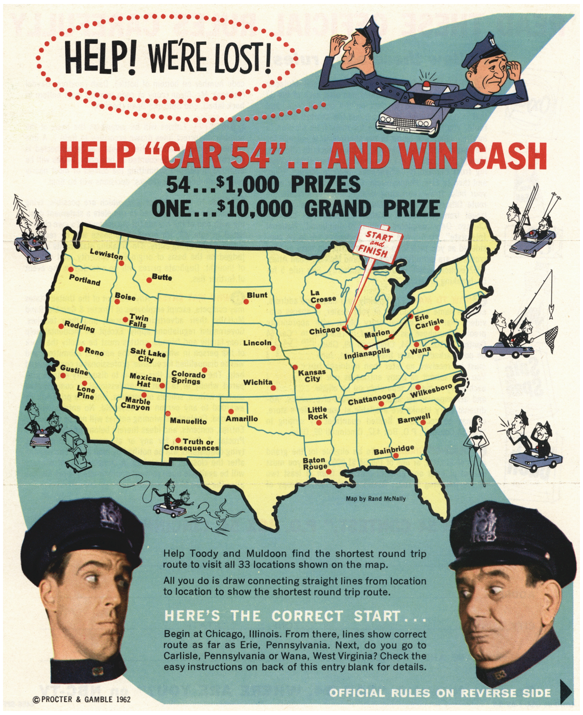

HELP! WE'RE LOST!
HELP "CAR 54"... AND WIN CASH
54...$1,000 PRIZES
ONE...$10,000 GRAND PRIZE
START and FINISH
Lewiston
Portland
Butte
Boise
Blunt
La Crosse
Twin Falls
Redding
Reno
Salt Lake City
Lincoln
Chicago
Marion
Erie
Carlisle
Indianapolis
Wana
Gustine
Lone Pine
Mexican Hat
Colorado Springs
Wichita
Kansas City
Marble Canyon
Manuelito
Amarillo
Little Rock
Chattanooga
Wilkesboro
Barnwell
Truth or Consequences
Baton Rouge
Bainbridge
Map by Rand McNally
Help Toody and Muldoon find the shortest round trip route to visit all 33 locations shown on the map.
All you do is draw connecting straight lines from location to location to show the shortest round trip route.
HERE'S THE CORRECT START...
Begin at Chicago, Illinois. From there, lines show correct route as far as Erie, Pennsylvania. Next, do you go to Carlisle, Pennsylvania or Wana, West Virginia? Check the easy instructions on back of this entry blank for details.
OFFICIAL RULES ON REVERSE SIDE
©PROCTER & GAMBLE 1962

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 finitel 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 flawles strategy. With one possible catch. The number of distinct tours is exceedingly large to consider checking one by one.

This difficult was noticed in 1930 by the Austrian mathematician and economist Karl Menger, who firs brought the challenge of the TSP to the attention of the mathematics community. "This problem is of course solvable by finitel 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 specifie 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

Copyrighted Material

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 firs 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

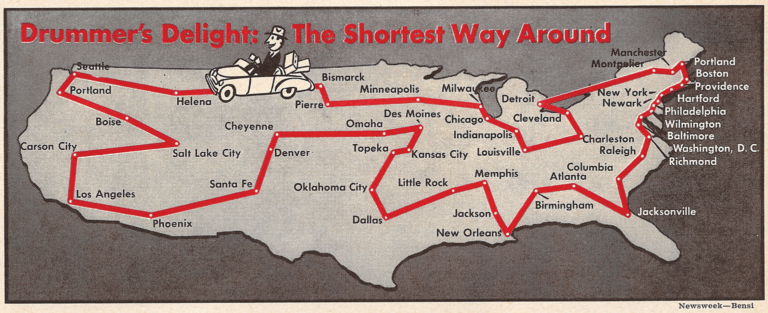
distinct tours that we must examine.

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 unsatisfie 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

Copyrighted Material

SCIENCE



Newsweek—Bensi

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 definitel a tough nut to crack, but if we turn back the clock to 1954, then we fin 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

Copyrighted Material

> after-dinner teaser. For years it has baffle 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 fin 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 finall 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 influentia center for the new fiel 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 fin and report this

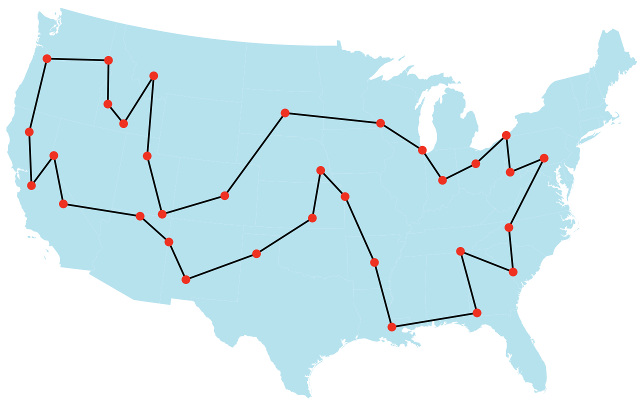


Figure 1.3
Optimal tour for *Car 54* contest.

Copyrighted Material

same ordering. Among the people tied for firs 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 finis 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 findin the best route through the United States or any other country does not do the job.

Picking up on the expected difficult of the general TSP challenge, the science-fictio story "Antibodies", by Charles Stross, chronicles doomsday events following the discovery of an efficien 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]

Copyrighted Material

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 firs raised the possibility that efficien 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 fin 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]

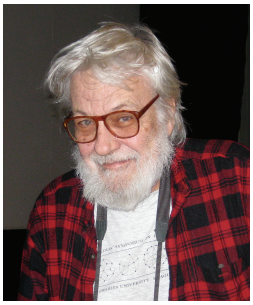


Figure 1.4
Jack Edmonds, 2009. Photograph courtesy of Marc Uetz.

Copyrighted Material

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 fixe 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.

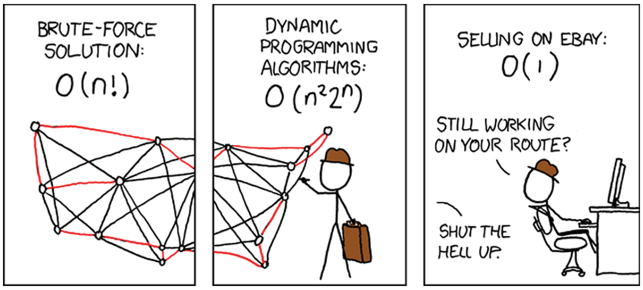


Figure 1.5
Travelling Salesman Problem. Image courtesy of Randall Munroe, xkcd.com.

Copyrighted Material

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 definitio 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 fin 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 firs person to discover a general method

Copyrighted Material

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 verifiabl manner would immediately have an efficien method of solution. In fact, current encryption systems make use of the assumption that certain $\mathcal{NP}$ problems are difficul 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—artificia 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 fin 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 flo 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 ramification 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 difficul instances of the problem.

Copyrighted Material

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 specifi 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 fiel 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 finall snapped in 1971 by IBM researchers Michael Held and Richard Karp; the same Karp who studied TSP impossibility results, clearly not satisfie 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

Copyrighted Material

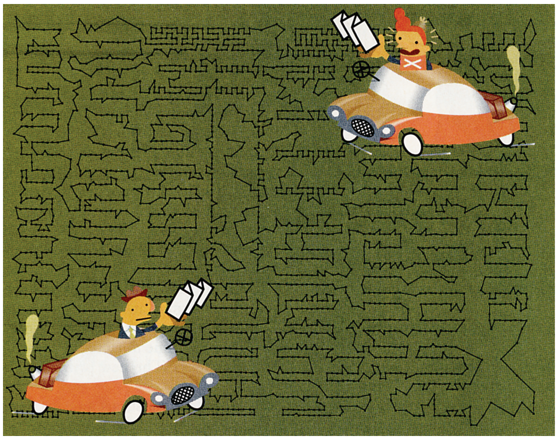


Figure 1.6
A new TSP record, 3,038 cities.
Discover, January 1993.

on to dominate the TSP scene for the next fiftee 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

Copyrighted Material

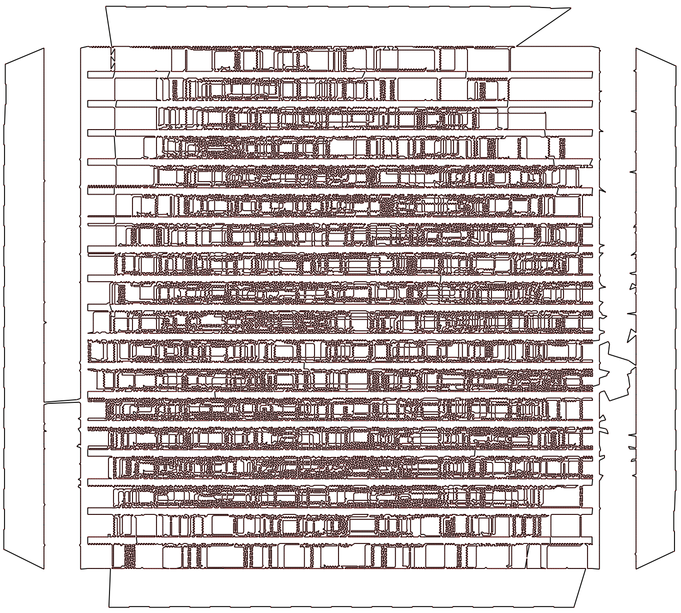


Figure 1.7
Solution of an 85,900-city TSP arising in a computer-chip application.

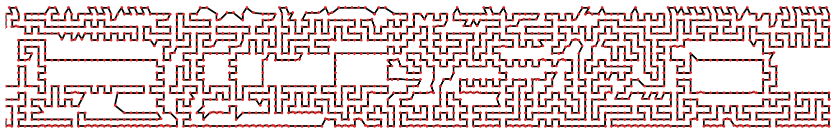


Figure 1.8
Close-up view of a portion of the 85,900-city tour.

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 figur 1.7, with a close-up view of a small region in figur 1.8.

Copyrighted Material

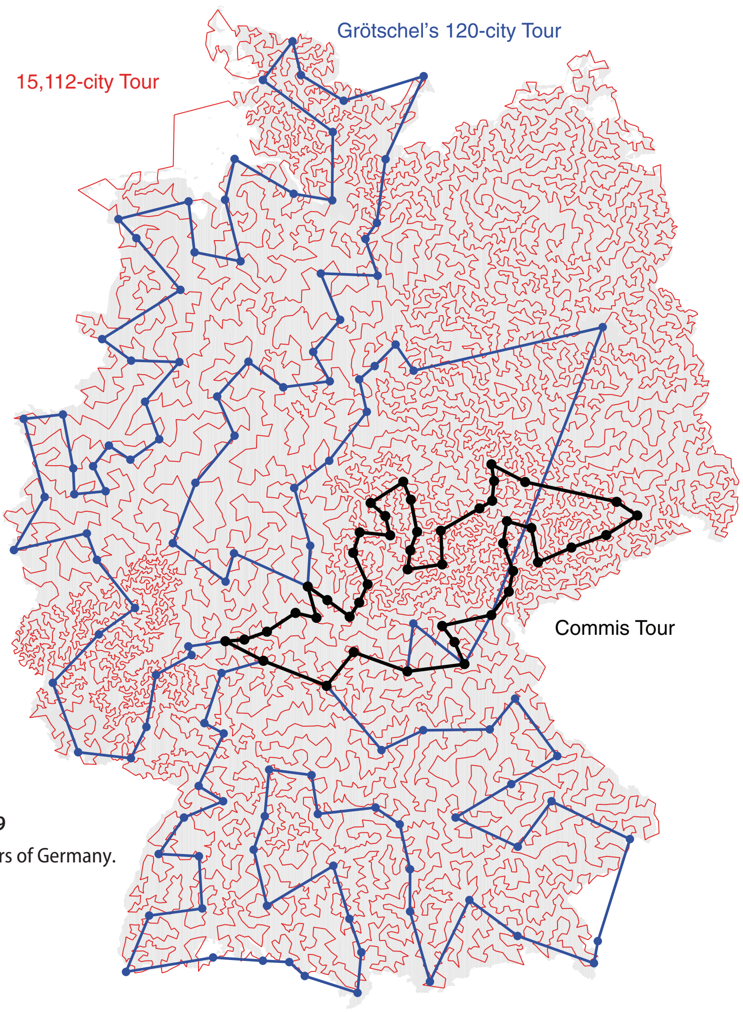


Figure 1.9
Three tours of Germany.

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 figur 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.

Copyrighted Material

The 15,112-city route may be the f nal 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 figur 1.10.

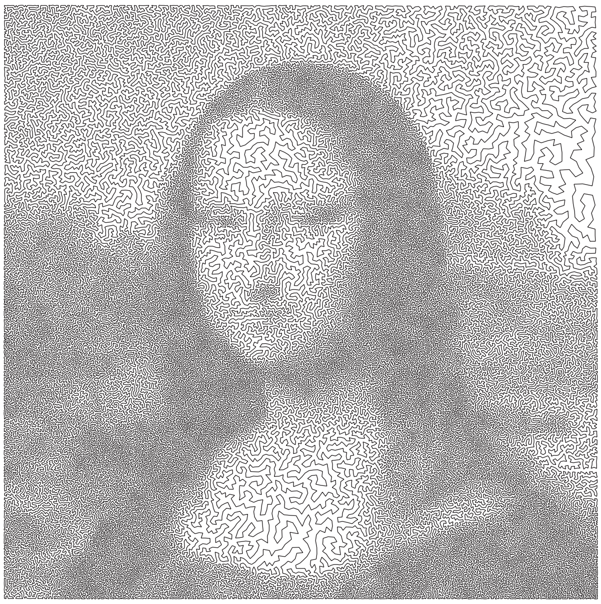


Figure 1.10 Leonardo da Vinci's *Mona Lisa* as a TSP. Tour found by Yuichi Nagata.

Copyrighted Material

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 firs 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 f gure 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.

Copyrighted Material

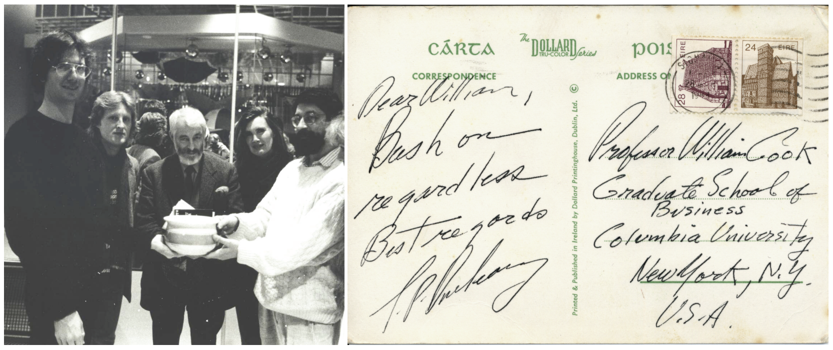
We begin in chapter 2 by examining the roots of the salesman problem from both the mathematical and applied perspectives; the presentation of TSP history allows us to introduce basic themes picked up in later chapters. This is followed, in chapter 3, by a selection of the many applications of the TSP, 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 f eld 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.


CÁRTA
The DOLLARD TRU-COLOR Series
POI
CORRESPONDENCE
ADDRESS O
Printed & Published in Ireland by Dollard Printinghouse, Dublin, Ltd. ©
28 EIRE
24 EIRE

Dear William,
Bash on
regardless
Best regards
[illegible]

Professor William Cook
Graduate School of
Business
Columbia University
New York, N.Y.
U.S.A.

Copyrighted Material

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.