

Travelling salesman problem

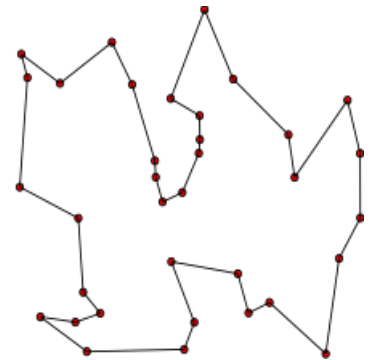
The **travelling salesman problem** (also called the **traveling salesperson problem**^[1] or **TSP**) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?". It is an NP-hard problem in combinatorial optimization, important in theoretical computer science and operations research.

The travelling purchaser problem and the vehicle routing problem are both generalizations of TSP.

In the theory of computational complexity, the decision version of the TSP (where given a length *L*, the task is to decide whether the graph has a tour of at most *L*) belongs to the class of NP-complete problems. Thus, it is possible that the worst-case running time for any algorithm for the TSP increases superpolynomially (but no more than exponentially) with the number of cities.

The problem was first formulated in 1930 and is one of the most intensively studied problems in optimization. It is used as a benchmark for many optimization methods. Even though the problem is computationally difficult, many heuristics and exact algorithms are known, so that some instances with tens of thousands of cities can be solved completely and even problems with millions of cities can be approximated within a small fraction of 1%.^[2]

The TSP has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, such as DNA sequencing. In these applications, the concept *city* represents, for example, customers, soldering points, or DNA fragments, and the concept *distance* represents travelling times or cost, or a similarity measure between DNA fragments. The TSP also appears in astronomy, as astronomers observing many sources will want to minimize the time spent moving the telescope between the sources; in such problems, the TSP can be imbedded inside an optimal control problem.^{[3][4]} In many applications, additional constraints such as limited resources or time windows may be imposed.



Solution of a travelling salesman problem: the black line shows the shortest possible loop that connects every red dot.

Contents

History

Description

As a graph problem

Asymmetric and symmetric

Related problems

Integer linear programming formulations

Miller–Tucker–Zemlin formulation

Dantzig–Fulkerson–Johnson formulation

Computing a solution

Exact algorithms

Heuristic and approximation algorithms

Special cases

Metric

Euclidean

Asymmetric

Analyst's problem

Path length for random sets of points in a square

Computational complexity

Complexity of approximation

Human and animal performance

Natural computation

Benchmarks

Popular culture

See also

[Notes](#)

[References](#)

[Further reading](#)

[External links](#)

History

The origins of the travelling salesman problem are unclear. A handbook for travelling salesmen from 1832 mentions the problem and includes example tours through [Germany](#) and [Switzerland](#), but contains no mathematical treatment.^[5]

The travelling salesman problem was mathematically formulated in the 1800s by the Irish mathematician [W.R. Hamilton](#) and by the British mathematician [Thomas Kirkman](#). Hamilton's [icosian game](#) was a recreational puzzle based on finding a [Hamiltonian cycle](#).^[6] The general form of the TSP appears to have been first studied by mathematicians during the 1930s in Vienna and at Harvard, notably by [Karl Menger](#), who defines the problem, considers the obvious brute-force algorithm, and observes the non-optimality of the nearest neighbour heuristic:

We denote by *messenger problem* (since in practice this question should be solved by each postman, anyway also by many travelers) the task to find, for finitely many points whose pairwise distances are known, the shortest route connecting the points. Of course, this problem is solvable by finitely many trials. Rules which would push the number of trials below the number of permutations of the given points, are not known. The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general does not yield the shortest route.^[7]



William Rowan Hamilton

It was first considered mathematically in the 1930s by [Merrill M. Flood](#) who was looking to solve a school bus routing problem.^[8] [Hassler Whitney](#) at [Princeton University](#) generated interest in the problem, which he called the "48 states problem". The earliest publication using the phrase "traveling salesman problem" was the 1949 [RAND Corporation](#) report by [Julia Robinson](#), "On the Hamiltonian game (a traveling salesman problem)."^{[9][10]}

In the 1950s and 1960s, the problem became increasingly popular in scientific circles in Europe and the US after the [RAND Corporation](#) in [Santa Monica](#) offered prizes for steps in solving the problem.^[8] Notable contributions were made by [George Dantzig](#), [Delbert Ray Fulkerson](#) and [Selmer M. Johnson](#) from the RAND Corporation, who expressed the problem as an [integer linear program](#) and developed the [cutting plane](#) method for its solution. They wrote what is considered the seminal paper on the subject in which with these new methods they solved an instance with 49 cities to optimality by constructing a tour and proving that no other tour could be shorter. Dantzig, Fulkerson and Johnson, however, speculated that given a near optimal solution we may be able to find optimality or prove optimality by adding a small number of extra inequalities (cuts). They used this idea to solve their initial 49 city problem using a string model. They found they only needed 26 cuts to come to a solution for their 49 city problem. While this paper did not give an algorithmic approach to TSP problems, the ideas that lay within it were indispensable to later creating exact solution methods for the TSP, though it would take 15 years to find an algorithmic approach in creating these cuts.^[8] As well as cutting plane methods, Dantzig, Fulkerson and Johnson used [branch and bound](#) algorithms perhaps for the first time.^[8]

In 1959, [Jillian Beardwood](#), [J.H. Halton](#) and [John Hammersley](#) published an article entitled "The Shortest Path Through Many Points" in the journal of the Cambridge Philosophical Society.^[11] The Beardwood–Halton–Hammersley theorem provides a practical solution to the traveling salesman problem. The authors derived an asymptotic formula to determine the length of the shortest route for a salesman who starts at a home or office and visits a fixed number of locations before returning to the start.

In the following decades, the problem was studied by many researchers from [mathematics](#), [computer science](#), [chemistry](#), [physics](#), and other sciences. In the 1960s, however, a new approach was created, that instead of seeking optimal solutions would produce a solution whose length is provably bounded by a multiple of the optimal length, and in doing so would create lower bounds for the problem; these lower bounds would then be used with branch and bound approaches. One method of doing this was to create a [minimum spanning tree](#) of the graph and then double all its edges, which produces the bound that the length of an optimal tour is at most twice the weight of a minimum spanning tree.^[8]

In 1976, [Christofides](#) and [Serdyukov](#) independently of each other made a big advance in this direction:^[12] the [Christofides-Serdyukov algorithm](#) yields a solution that, in the worst case, is at most 1.5 times longer than the optimal solution. As the algorithm was simple and quick, many hoped it would give way to a near optimal solution method. However, this hoped for improvement did not immediately materialize, and Christofides-Serdyukov remained the method with the best worst-case scenario until 2011, when a (very) slightly improved approximation algorithm was developed for the subset of "graphical" TSPs.^[13] In 2020 this tiny improvement was extended to the full (metric) TSP. ^{[14][15]}

Richard M. Karp showed in 1972 that the Hamiltonian cycle problem was NP-complete, which implies the NP-hardness of TSP. This supplied a mathematical explanation for the apparent computational difficulty of finding optimal tours.

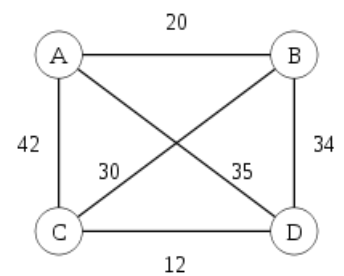
Great progress was made in the late 1970s and 1980, when Grötschel, Padberg, Rinaldi and others managed to exactly solve instances with up to 2,392 cities, using cutting planes and branch and bound.

In the 1990s, Applegate, Bixby, Chvátal, and Cook developed the program *Concorde* that has been used in many recent record solutions. Gerhard Reinelt published the TSPLIB in 1991, a collection of benchmark instances of varying difficulty, which has been used by many research groups for comparing results. In 2006, Cook and others computed an optimal tour through an 85,900-city instance given by a microchip layout problem, currently the largest solved TSPLIB instance. For many other instances with millions of cities, solutions can be found that are guaranteed to be within 2–3% of an optimal tour.^[16]

Description

As a graph problem

TSP can be modelled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's weight. It is a minimization problem starting and finishing at a specified vertex after having visited each other vertex exactly once. Often, the model is a complete graph (i.e., each pair of vertices is connected by an edge). If no path exists between two cities, adding a sufficiently long edge will complete the graph without affecting the optimal tour.



Symmetric TSP with four cities

Asymmetric and symmetric

In the *symmetric TSP*, the distance between two cities is the same in each opposite direction, forming an undirected graph. This symmetry halves the number of possible solutions. In the *asymmetric TSP*, paths may not exist in both directions or the distances might be different, forming a directed graph. Traffic collisions, one-way streets, and airfares for cities with different departure and arrival fees are examples of how this symmetry could break down.

Related problems

- An equivalent formulation in terms of graph theory is: Given a complete weighted graph (where the vertices would represent the cities, the edges would represent the roads, and the weights would be the cost or distance of that road), find a Hamiltonian cycle with the least weight.
- The requirement of returning to the starting city does not change the computational complexity of the problem, see Hamiltonian path problem.
- Another related problem is the Bottleneck traveling salesman problem (bottleneck TSP): Find a Hamiltonian cycle in a weighted graph with the minimal weight of the weightiest edge. For example, avoiding narrow streets with big buses.^[17] The problem is of considerable practical importance, apart from evident transportation and logistics areas. A classic example is in printed circuit manufacturing: scheduling of a route of the drill machine to drill holes in a PCB. In robotic machining or drilling applications, the "cities" are parts to machine or holes (of different sizes) to drill, and the "cost of travel" includes time for retooling the robot (single machine job sequencing problem).^[18]
- The generalized travelling salesman problem, also known as the "travelling politician problem", deals with "states" that have (one or more) "cities" and the salesman has to visit exactly one "city" from each "state". One application is encountered in ordering a solution to the cutting stock problem in order to minimize knife changes. Another is concerned with drilling in semiconductor manufacturing, see e.g., U.S. Patent 7,054,798 (<https://patents.google.com/patent/US7054798>). Noon and Bean demonstrated that the generalized travelling salesman problem can be transformed into a standard travelling salesman problem with the same number of cities, but a modified distance matrix.
- The sequential ordering problem deals with the problem of visiting a set of cities where precedence relations between the cities exist.
- A common interview question at Google is how to route data among data processing nodes; routes vary by time to transfer the data, but nodes also differ by their computing power and storage, compounding the problem of where to send data.
- The travelling purchaser problem deals with a purchaser who is charged with purchasing a set of products. He can purchase these products in several cities, but at different prices and not all cities offer the same products. The objective is to find a route between a subset of the cities, which minimizes total cost (travel cost + purchasing cost) and which enables the purchase of all required products.

Integer linear programming formulations

The TSP can be formulated as an integer linear program.^{[19][20][21]} Several formulations are known. Two notable formulations are the Miller–Tucker–Zemlin (MTZ) formulation and the Dantzig–Fulkerson–Johnson (DFJ) formulation. The DFJ formulation is stronger, though the MTZ formulation is still useful in certain settings.^{[22][23]}

Miller–Tucker–Zemlin formulation

Label the cities with the numbers $1, \dots, n$ and define:

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

For $i = 1, \dots, n$, let u_i be a dummy variable, and finally take $c_{ij} > 0$ to be the distance from city i to city j . Then TSP can be written as the following integer linear programming problem:

$$\begin{aligned} \min & \sum_{i=1}^n \sum_{j \neq i, j=1}^n c_{ij} x_{ij} \\ & x_{ij} \in \{0, 1\} & i, j = 1, \dots, n; \\ & u_i \in \mathbf{Z} & i = 2, \dots, n; \\ & \sum_{i=1, i \neq j}^n x_{ij} = 1 & j = 1, \dots, n; \\ & \sum_{j=1, j \neq i}^n x_{ij} = 1 & i = 1, \dots, n; \\ & u_i - u_j + nx_{ij} \leq n - 1 & 2 \leq i \neq j \leq n; \\ & 1 \leq u_i \leq n - 1 & 2 \leq i \leq n. \end{aligned}$$

The first set of equalities requires that each city is arrived at from exactly one other city, and the second set of equalities requires that from each city there is a departure to exactly one other city. The last constraints enforce that there is only a single tour covering all cities, and not two or more disjointed tours that only collectively cover all cities. To prove this, it is shown below (1) that every feasible solution contains only one closed sequence of cities, and (2) that for every single tour covering all cities, there are values for the dummy variables u_i that satisfy the constraints. (The dummy variables indicate tour ordering, such that $u_i < u_j$ implies city i is visited before city j . This may be accomplished by incrementing u_i each time it is visited.)

To prove that every feasible solution contains only one closed sequence of cities, it suffices to show that every subtour in a feasible solution passes through city 1 (noting that the equalities ensure there can only be one such tour). For if we sum all the inequalities corresponding to $x_{ij} = 1$ for any subtour of k steps not passing through city 1, we obtain:

$$nk \leq (n - 1)k,$$

which is a contradiction.

It now must be shown that for every single tour covering all cities, there are values for the dummy variables u_i that satisfy the constraints.

Without loss of generality, define the tour as originating (and ending) at city 1. Choose $u_i = t$ if city i is visited in step $t(i, t = 1, 2, \dots, n)$. Then

$$u_i - u_j \leq n - 1,$$

since u_i can be no greater than n and u_j can be no less than 1; hence the constraints are satisfied whenever $x_{ij} = 0$. For $x_{ij} = 1$, we have:

$$u_i - u_j + nx_{ij} = (t) - (t + 1) + n = n - 1,$$

satisfying the constraint.

Dantzig–Fulkerson–Johnson formulation

Label the cities with the numbers $1, \dots, n$ and define:

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

Take $c_{ij} > 0$ to be the distance from city i to city j . Then TSP can be written as the following integer linear programming problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j \neq i, j=1}^n c_{ij} x_{ij} \\ \sum_{i=1, i \neq j}^n x_{ij} &= 1 & j = 1, \dots, n; \\ \sum_{j=1, j \neq i}^n x_{ij} &= 1 & i = 1, \dots, n; \\ \sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{ij} &\leq |Q| - 1 & \forall Q \subsetneq \{1, \dots, n\}, |Q| \geq 2 \end{aligned}$$

The last constraint of the DFJ formulation ensures no proper subset Q can form a sub-tour, so the solution returned is a single tour and not the union of smaller tours. Because this leads to an exponential number of possible constraints, in practice it is solved with delayed column generation.

Computing a solution

The traditional lines of attack for the NP-hard problems are the following:

- Devising exact algorithms, which work reasonably fast only for small problem sizes.
- Devising "suboptimal" or heuristic algorithms, i.e., algorithms that deliver approximated solutions in a reasonable time.
- Finding special cases for the problem ("subproblems") for which either better or exact heuristics are possible.

Exact algorithms

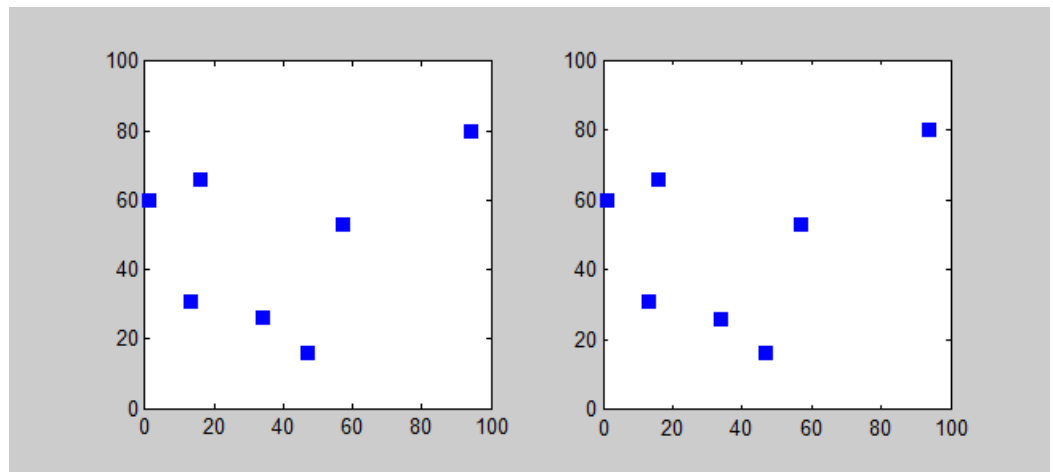
The most direct solution would be to try all permutations (ordered combinations) and see which one is cheapest (using brute-force search). The running time for this approach lies within a polynomial factor of $O(n!)$, the factorial of the number of cities, so this solution becomes impractical even for only 20 cities.

One of the earliest applications of dynamic programming is the Held–Karp algorithm that solves the problem in time $O(n^2 2^n)$.^[24] This bound has also been reached by Exclusion-Inclusion in an attempt preceding the dynamic programming approach.

Improving these time bounds seems to be difficult. For example, it has not been determined whether an exact algorithm for TSP that runs in time $O(1.9999^n)$ exists.^[25]

Other approaches include:

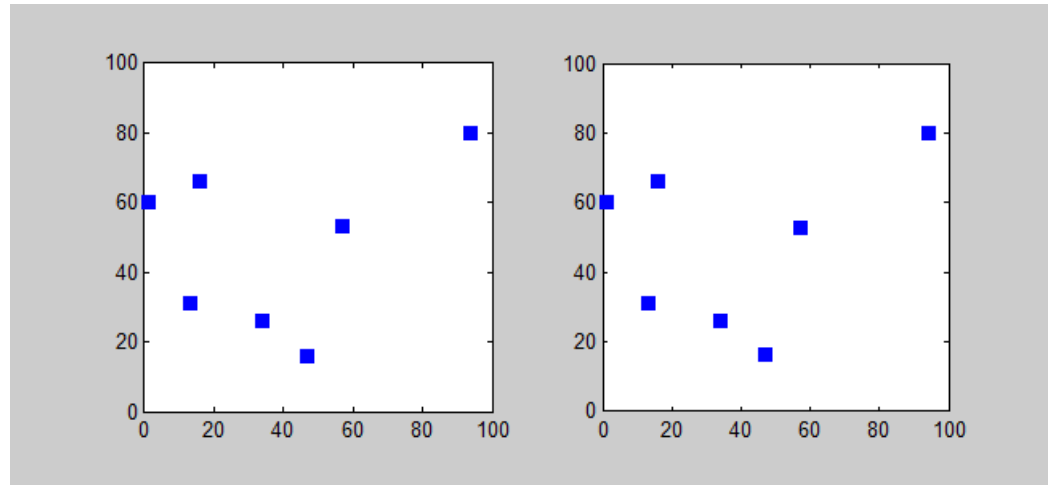
- Various branch-and-bound algorithms, which can be used to process TSPs containing 40–60 cities.
- Progressive improvement algorithms which use techniques reminiscent of linear programming. Works well for up to 200 cities.
- Implementations of branch-and-bound and problem-specific cut generation (branch-and-cut^[26]); this is the method of choice for solving large instances. This approach holds the current record, solving an instance with 85,900 cities, see Applegate et al. (2006).



Solution to a symmetric TSP with 7 cities using brute force search. Note: Number of permutations: $(7-1)!/2 = 360$

An exact solution for 15,112 German towns from TSPLIB was found in 2001 using the cutting-plane method proposed by George Dantzig, Ray Fulkerson, and Selmer M. Johnson in 1954, based on linear programming. The computations were performed on a network of 110 processors located at Rice University and Princeton University. The total computation time was equivalent to 22.6 years on a single 500 MHz Alpha processor. In May 2004, the travelling salesman problem of visiting all 24,978 towns in Sweden was solved: a tour of

length approximately 72,500 kilometres was found and it was proven that no shorter tour exists.^[27] In March 2005, the travelling salesman problem of visiting all 33,810 points in a circuit board was solved using *Concorde TSP Solver*: a tour of length 66,048,945 units was found and it was proven that no shorter tour exists. The computation took approximately 15.7 CPU-years (Cook et al. 2006). In April 2006 an instance with 85,900 points was solved using *Concorde TSP Solver*, taking over 136 CPU-years, see Applegate et al. (2006).



Solution of a TSP with 7 cities using a simple Branch and bound algorithm. Note: The number of permutations is much less than Brute force search

Heuristic and approximation algorithms

Various heuristics and approximation algorithms, which quickly yield good solutions, have been devised. These include the Multi-fragment algorithm. Modern methods can find solutions for extremely large problems (millions of cities) within a reasonable time which are with a high probability just 2–3% away from the optimal solution.^[16]

Several categories of heuristics are recognized.

Constructive heuristics

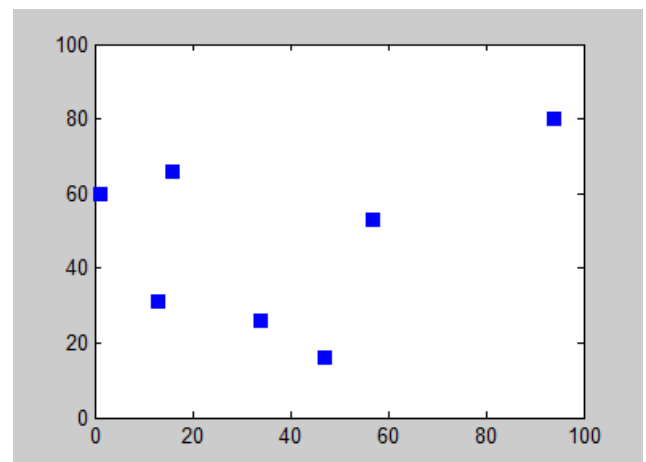
The nearest neighbour (NN) algorithm (a greedy algorithm) lets the salesman choose the nearest unvisited city as his next move. This algorithm quickly yields an effectively short route. For N cities randomly distributed on a plane, the algorithm on average yields a path 25% longer than the shortest possible path.^[28] However, there exist many specially arranged city distributions which make the NN algorithm give the worst route.^[29] This is true for both asymmetric and symmetric TSPs.^[30] Rosenkrantz et al.^[31] showed that the NN algorithm has the approximation factor $\Theta(\log |V|)$ for instances satisfying the triangle inequality. A variation of NN algorithm, called Nearest Fragment (NF) operator, which connects a group (fragment) of nearest unvisited cities, can find shorter routes with successive iterations.^[32] The NF operator can also be applied on an initial solution obtained by NN algorithm for further improvement in an elitist model, where only better solutions are accepted.

The bitonic tour of a set of points is the minimum-perimeter monotone polygon that has the points as its vertices; it can be computed efficiently by dynamic programming.

Another constructive heuristic, Match Twice and Stitch (MTS), performs two sequential matchings, where the second matching is executed after deleting all the edges of the first matching, to yield a set of cycles. The cycles are then stitched to produce the final tour.^[33]

The Algorithm of Christofides and Serdyukov

The algorithm of Christofides and Serdyukov follows a similar outline but combines the minimum spanning tree with a solution of another problem, minimum-weight perfect matching. This gives a TSP tour which is at most 1.5 times the optimal. It was one of the first approximation algorithms, and was in part responsible for drawing attention to approximation algorithms as a practical approach to intractable problems. As a matter of fact, the term "algorithm" was not commonly extended to approximation algorithms until later; the Christofides algorithm was initially referred to as the Christofides heuristic.^[12]



Nearest Neighbour algorithm for a TSP with 7 cities. The solution changes as the starting point is changed

This algorithm looks at things differently by using a result from graph theory which helps improve on the lower bound of the TSP which originated from doubling the cost of the minimum spanning tree. Given an Eulerian graph we can find an Eulerian tour in $O(n)$ time.^[8] So if we had an Eulerian graph with cities from a TSP as vertices then we can easily see that we could use such a method for finding an Eulerian tour to find a TSP solution. By triangular inequality we know that the TSP tour can be no longer than the Eulerian tour and as such we have a lower bound for the TSP. Such a method is described below.

1. Find a minimum spanning tree for the problem
2. Create duplicates for every edge to create an Eulerian graph
3. Find an Eulerian tour for this graph
4. Convert to TSP: if a city is visited twice, create a shortcut from the city before this in the tour to the one after this.

To improve the lower bound, a better way of creating an Eulerian graph is needed. By triangular inequality, the best Eulerian graph must have the same cost as the best travelling salesman tour, hence finding optimal Eulerian graphs is at least as hard as TSP. One way of doing this is by minimum weight matching using algorithms of $O(n^3)$.^[8]

Making a graph into an Eulerian graph starts with the minimum spanning tree. Then all the vertices of odd order must be made even. So a matching for the odd degree vertices must be added which increases the order of every odd degree vertex by one.^[8] This leaves us with a graph where every vertex is of even order which is thus Eulerian. Adapting the above method gives the algorithm of Christofides and Serdyukov.

1. Find a minimum spanning tree for the problem
2. Create a matching for the problem with the set of cities of odd order.
3. Find an Eulerian tour for this graph
4. Convert to TSP using shortcuts.

Pairwise exchange

The pairwise exchange or 2-opt technique involves iteratively removing two edges and replacing these with two different edges that reconnect the fragments created by edge removal into a new and shorter tour. Similarly, the 3-opt technique removes 3 edges and reconnects them to form a shorter tour. These are special cases of the k -opt method. The label Lin-Kernighan is an often heard misnomer for 2-opt. Lin-Kernighan is actually the more general k -opt method.

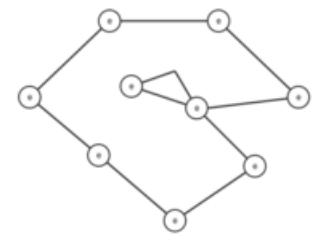
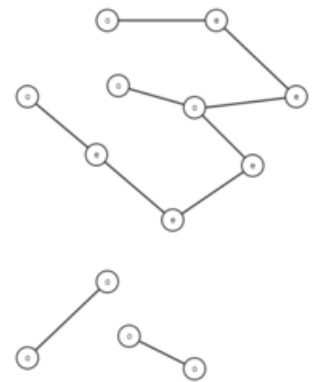
For Euclidean instances, 2-opt heuristics give on average solutions that are about 5% better than Christofides' algorithm. If we start with an initial solution made with a greedy algorithm, the average number of moves greatly decreases again and is $O(n)$. For random starts however, the average number of moves is $O(n \log(n))$. However whilst in order this is a small increase in size, the initial number of moves for small problems is 10 times as big for a random start compared to one made from a greedy heuristic. This is because such 2-opt heuristics exploit 'bad' parts of a solution such as crossings. These types of heuristics are often used within Vehicle routing problem heuristics to reoptimize route solutions.^[28]

k -opt heuristic, or Lin-Kernighan heuristics

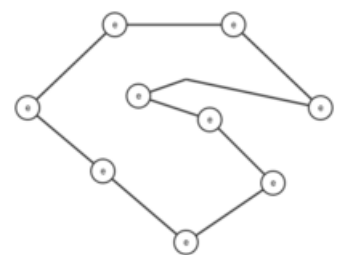
The Lin-Kernighan heuristic is a special case of the V -opt or variable-opt technique. It involves the following steps:

1. Given a tour, delete k mutually disjoint edges.
2. Reassemble the remaining fragments into a tour, leaving no disjoint subtours (that is, don't connect a fragment's endpoints together). This in effect simplifies the TSP under consideration into a much simpler problem.
3. Each fragment endpoint can be connected to $2k - 2$ other possibilities: of $2k$ total fragment endpoints available, the two endpoints of the fragment under consideration are disallowed. Such a constrained $2k$ -city TSP can then be solved with brute force methods to find the least-cost recombination of the original fragments.

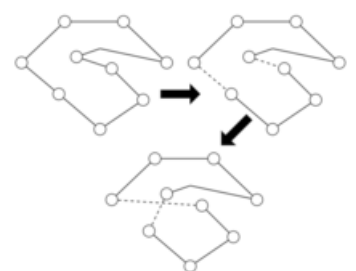
The most popular of the k -opt methods are 3-opt, as introduced by Shen Lin of Bell Labs in 1965. A special case of 3-opt is where the edges are not disjoint (two of the edges are adjacent to one another). In practice, it is often possible to achieve substantial improvement over 2-opt without the combinatorial cost of the general 3-opt by restricting the 3-changes to this special subset where two of the removed



Creating a matching



Using a shortcut heuristic on the graph created by the matching above



An example of a 2-opt iteration

edges are adjacent. This so-called two-and-a-half-opt typically falls roughly midway between 2-opt and 3-opt, both in terms of the quality of tours achieved and the time required to achieve those tours.

V-opt heuristic

The variable-opt method is related to, and a generalization of the k -opt method. Whereas the k -opt methods remove a fixed number (k) of edges from the original tour, the variable-opt methods do not fix the size of the edge set to remove. Instead they grow the set as the search process continues. The best known method in this family is the Lin–Kernighan method (mentioned above as a misnomer for 2-opt). Shen Lin and Brian Kernighan first published their method in 1972, and it was the most reliable heuristic for solving travelling salesman problems for nearly two decades. More advanced variable-opt methods were developed at Bell Labs in the late 1980s by David Johnson and his research team. These methods (sometimes called Lin–Kernighan–Johnson) build on the Lin–Kernighan method, adding ideas from tabu search and evolutionary computing. The basic Lin–Kernighan technique gives results that are guaranteed to be at least 3-opt. The Lin–Kernighan–Johnson methods compute a Lin–Kernighan tour, and then perturb the tour by what has been described as a mutation that removes at least four edges and reconnecting the tour in a different way, then V-opting the new tour. The mutation is often enough to move the tour from the local minimum identified by Lin–Kernighan. V-opt methods are widely considered the most powerful heuristics for the problem, and are able to address special cases, such as the Hamilton Cycle Problem and other non-metric TSPs that other heuristics fail on. For many years Lin–Kernighan–Johnson had identified optimal solutions for all TSPs where an optimal solution was known and had identified the best known solutions for all other TSPs on which the method had been tried.

Randomized improvement

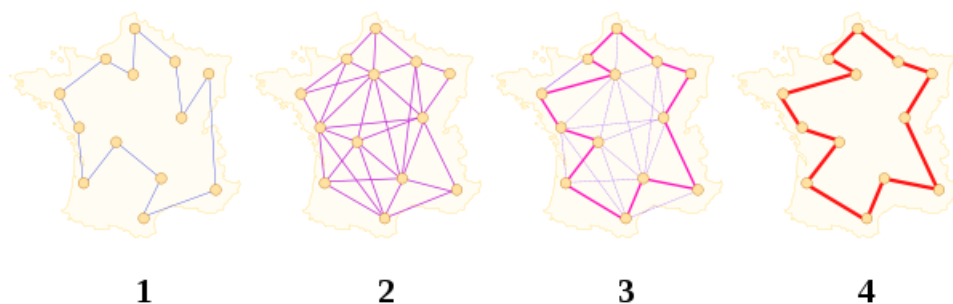
Optimized Markov chain algorithms which use local searching heuristic sub-algorithms can find a route extremely close to the optimal route for 700 to 800 cities.

TSP is a touchstone for many general heuristics devised for combinatorial optimization such as genetic algorithms, simulated annealing, tabu search, ant colony optimization, river formation dynamics (see swarm intelligence) and the cross entropy method.

Ant colony optimization

Artificial intelligence researcher Marco Dorigo described in 1993 a method of heuristically generating "good solutions" to the TSP using a simulation of an ant colony called ACS (*ant colony system*).^[34] It models behaviour observed in real ants to find short paths between food sources and their nest, an emergent behaviour resulting from each ant's preference to follow trail pheromones deposited by other ants.

ACS sends out a large number of virtual ant agents to explore many possible routes on the map. Each ant probabilistically chooses the next city to visit based on a heuristic combining the distance to the city and the amount of virtual pheromone deposited on the edge to the city. The ants explore, depositing pheromone on each edge that they cross, until they have all completed a tour. At this point the ant which completed the shortest tour deposits virtual pheromone along its complete tour route (*global trail updating*). The amount of pheromone deposited is inversely proportional to the tour length: the shorter the tour, the more it deposits.





Ant colony optimization algorithm for a TSP with 7 cities: Red and thick lines in the pheromone map indicate presence of more pheromone

Special cases

Metric

In the *metric TSP*, also known as *delta-TSP* or Δ -TSP, the intercity distances satisfy the triangle inequality.

A very natural restriction of the TSP is to require that the distances between cities form a metric to satisfy the triangle inequality; that is the direct connection from A to B is never farther than the route via intermediate C :

$$d_{AB} \leq d_{AC} + d_{CB}.$$

The edge spans then build a metric on the set of vertices. When the cities are viewed as points in the plane, many natural distance functions are metrics, and so many natural instances of TSP satisfy this constraint.

The following are some examples of metric TSPs for various metrics.

- In the Euclidean TSP (see below) the distance between two cities is the Euclidean distance between the corresponding points.
- In the rectilinear TSP the distance between two cities is the sum of the absolute values of the differences of their x - and y -coordinates. This metric is often called the Manhattan distance or city-block metric.
- In the maximum metric, the distance between two points is the maximum of the absolute values of differences of their x - and y -coordinates.

The last two metrics appear, for example, in routing a machine that drills a given set of holes in a printed circuit board. The Manhattan metric corresponds to a machine that adjusts first one co-ordinate, and then the other, so the time to move to a new point is the sum of both movements. The maximum metric corresponds to a machine that adjusts both co-ordinates simultaneously, so the time to move to a new point is the slower of the two movements.

In its definition, the TSP does not allow cities to be visited twice, but many applications do not need this constraint. In such cases, a symmetric, non-metric instance can be reduced to a metric one. This replaces the original graph with a complete graph in which the inter-city distance d_{AB} is replaced by the shortest path length between A and B in the original graph.

Euclidean

When the input numbers can be arbitrary real numbers, Euclidean TSP is a particular case of metric TSP, since distances in a plane obey the triangle inequality. When the input numbers must be integers, comparing lengths of tours involves comparing sums of square-roots.

Like the general TSP, Euclidean TSP is NP-hard in either case. With rational coordinates and discretized metric (distances rounded up to an integer), the problem is NP-complete.^[35] With rational coordinates and the actual Euclidean metric, Euclidean TSP is known to be in the Counting Hierarchy,^[36] a subclass of PSPACE. With arbitrary real coordinates, Euclidean TSP cannot be in such classes, since there are uncountably many possible inputs. However, Euclidean TSP is probably the easiest version for approximation.^[37] For example, the minimum spanning tree of the graph associated with an instance of the Euclidean TSP is a Euclidean minimum spanning tree, and so can be computed in expected $O(n \log n)$ time for n points (considerably less than the number of edges). This enables the simple 2-approximation algorithm for TSP with triangle inequality above to operate more quickly.

In general, for any $c > 0$, where d is the number of dimensions in the Euclidean space, there is a polynomial-time algorithm that finds a tour of length at most $(1 + 1/c)$ times the optimal for geometric instances of TSP in

$$O\left(n(\log n)^{O(c\sqrt{d})^{d-1}}\right),$$

time; this is called a polynomial-time approximation scheme (PTAS).^[38] Sanjeev Arora and Joseph S. B. Mitchell were awarded the Gödel Prize in 2010 for their concurrent discovery of a PTAS for the Euclidean TSP.

In practice, simpler heuristics with weaker guarantees continue to be used.

Asymmetric

In most cases, the distance between two nodes in the TSP network is the same in both directions. The case where the distance from A to B is not equal to the distance from B to A is called asymmetric TSP. A practical application of an asymmetric TSP is route optimization using street-level routing (which is made asymmetric by one-way streets, slip-roads, motorways, etc.).

Conversion to symmetric

Solving an asymmetric TSP graph can be somewhat complex. The following is a 3×3 matrix containing all possible path weights between the nodes A , B and C . One option is to turn an asymmetric matrix of size N into a symmetric matrix of size $2N$.^[39]

Asymmetric
path weights

	A	B	C
A		1	2
B	6		3
C	5	4	

To double the size, each of the nodes in the graph is duplicated, creating a second *ghost node*, linked to the original node with a "ghost" edge of very low (possibly negative) weight, here denoted $-w$. (Alternatively, the ghost edges have weight 0, and weight w is added to all other edges.) The original 3×3 matrix shown above is visible in the bottom left and the transpose of the original in the top-right. Both copies of the matrix have had their diagonals replaced by the low-cost hop paths, represented by $-w$. In the new graph, no edge directly links original nodes and no edge directly links ghost nodes.

Symmetric path weights

	A	B	C	A'	B'	C'
A				$-w$	6	5
B				1	$-w$	4
C				2	3	$-w$
A'	$-w$	1	2			
B'	6	$-w$	3			
C'	5	4	$-w$			

The weight $-w$ of the "ghost" edges linking the ghost nodes to the corresponding original nodes must be low enough to ensure that all ghost edges must belong to any optimal symmetric TSP solution on the new graph ($w=0$ is not always low enough). As a consequence, in the optimal symmetric tour, each original node appears next to its ghost node (e.g. a possible path is $A \rightarrow A' \rightarrow C \rightarrow C' \rightarrow B \rightarrow B' \rightarrow A$) and by merging the original and ghost nodes again we get an (optimal) solution of the original asymmetric problem (in our example, $A \rightarrow C \rightarrow B \rightarrow A$).

Analyst's problem

There is an analogous problem in geometric measure theory which asks the following: under what conditions may a subset E of Euclidean space be contained in a rectifiable curve (that is, when is there a curve with finite length that visits every point in E)? This problem is known as the analyst's travelling salesman problem.

Path length for random sets of points in a square

Suppose X_1, \dots, X_n are n independent random variables with uniform distribution in the square $[0, 1]^2$, and let L_n^* be the shortest path length (i.e. TSP solution) for this set of points, according to the usual Euclidean distance. It is known^[11] that, almost surely,

$$\frac{L_n^*}{\sqrt{n}} \rightarrow \beta \quad \text{when } n \rightarrow \infty,$$

where β is a positive constant that is not known explicitly. Since $L_n^* \leq 2\sqrt{n} + 2$ (see below), it follows from bounded convergence theorem that $\beta = \lim_{n \rightarrow \infty} \mathbb{E}[L_n^*]/\sqrt{n}$, hence lower and upper bounds on β follow from bounds on $\mathbb{E}[L_n^*]$.

The almost sure limit $\frac{L_n^*}{\sqrt{n}} \rightarrow \beta$ as $n \rightarrow \infty$ may not exist if the independent locations X_1, \dots, X_n are replaced with observations from a stationary ergodic process with uniform marginals.^[40]

Upper bound

- One has $L^* \leq 2\sqrt{n} + 2$, and therefore $\beta \leq 2$, by using a naive path which visits monotonically the points inside each of \sqrt{n} slices of width $1/\sqrt{n}$ in the square.
- Few^[41] proved $L_n^* \leq \sqrt{2n} + 1.75$, hence $\beta \leq \sqrt{2}$, later improved by Karloff (1987): $\beta \leq 0.984\sqrt{2}$.
- Some study reported^[42] an upper bound that $\beta \leq 0.73 \dots$

Lower bound

- By observing that $\mathbb{E}[L_n^*]$ is greater than n times the distance between X_0 and the closest point $X_i \neq X_0$, one gets (after a short computation)

$$\mathbb{E}[L_n^*] \geq \frac{1}{2}\sqrt{n}.$$

- A better lower bound is obtained^[11] by observing that $\mathbb{E}[L_n^*]$ is greater than $\frac{1}{2}n$ times the sum of the distances between X_0 and the closest and second closest points $X_i, X_j \neq X_0$, which gives

$$\mathbb{E}[L_n^*] \geq \left(\frac{1}{4} + \frac{3}{8}\right)\sqrt{n} = \frac{5}{8}\sqrt{n},$$

- The currently^[43] best lower bound is

$$\mathbb{E}[L_n^*] \geq \left(\frac{5}{8} + \frac{19}{5184}\right)\sqrt{n},$$

- Held and Karp^[44] gave a polynomial-time algorithm that provides numerical lower bounds for L_n^* , and thus for $\beta (\simeq L_n^*/\sqrt{n})$ which seem to be good up to more or less 1%.^[45] In particular, David S. Johnson^[46] obtained a lower bound by computer experiment:

$$L_n^* \gtrsim 0.7080\sqrt{n} + 0.522,$$

where 0.522 comes from the points near square boundary which have fewer neighbours, and Christine L. Valenzuela and Antonia J. Jones^[47] obtained the following other numerical lower bound:

$$L_n^* \gtrsim 0.7078\sqrt{n} + 0.551.$$

Computational complexity

The problem has been shown to be NP-hard (more precisely, it is complete for the complexity class FP^{NP} ; see function problem), and the decision problem version ("given the costs and a number x , decide whether there is a round-trip route cheaper than x ") is NP-complete. The bottleneck traveling salesman problem is also NP-hard. The problem remains NP-hard even for the case when the cities are in the plane with Euclidean distances, as well as in a number of other restrictive cases. Removing the condition of visiting each city "only once" does not remove the NP-hardness, since in the planar case there is an optimal tour that visits each city only once (otherwise, by the triangle inequality, a shortcut that skips a repeated visit would not increase the tour length).

Complexity of approximation

In the general case, finding a shortest travelling salesman tour is [NPO-complete](#).^[48] If the distance measure is a [metric](#) (and thus symmetric), the problem becomes [APX-complete](#)^[49] and the [algorithm of Christofides and Serdyukov](#) approximates it within 1.5.^{[50][51][12]} A 2020 [preprint](#) improves this bound to $1.5 - 10^{-36}$.^[52] The best known inapproximability bound is 123/122.^[53]

If the distances are restricted to 1 and 2 (but still are a metric) the approximation ratio becomes 8/7.^[54] In the asymmetric case with [triangle inequality](#), up until recently only logarithmic performance guarantees were known.^[55] In 2018, a constant factor approximation was developed by Svensson, Tarnawski and Végh.^[56] The best current algorithm, by Traub and Vygen, achieves performance ratio of $22 + \epsilon$.^[57] The best known inapproximability bound is 75/74.^[53]

The corresponding maximization problem of finding the *longest* travelling salesman tour is approximable within 63/38.^[58] If the distance function is symmetric, the longest tour can be approximated within 4/3 by a deterministic algorithm^[59] and within $\frac{1}{25}(33 + \epsilon)$ by a randomized algorithm.^[60]

Human and animal performance

The TSP, in particular the [Euclidean](#) variant of the problem, has attracted the attention of researchers in [cognitive psychology](#). It has been observed that humans are able to produce near-optimal solutions quickly, in a close-to-linear fashion, with performance that ranges from 1% less efficient for graphs with 10-20 nodes, and 11% less efficient for graphs with 120 nodes.^{[61][62]} The apparent ease with which humans accurately generate near-optimal solutions to the problem has led researchers to hypothesize that humans use one or more heuristics, with the two most popular theories arguably being the [convex-hull hypothesis](#) and the [crossing-avoidance heuristic](#).^{[63][64][65]} However, additional evidence suggests that human performance is quite varied, and individual differences as well as graph geometry appear to affect performance in the task.^{[66][67][68]} Nevertheless, results suggest that computer performance on the TSP may be improved by understanding and emulating the methods used by humans for these problems,^[69] and have also led to new insights into the mechanisms of human thought.^[70] The first issue of the *Journal of Problem Solving* was devoted to the topic of human performance on TSP,^[71] and a 2011 review listed dozens of papers on the subject.^[70]

A 2011 study in [animal cognition](#) titled "Let the Pigeon Drive the Bus," named after the children's book *Don't Let the Pigeon Drive the Bus!*, examined spatial cognition in pigeons by studying their flight patterns between multiple feeders in a laboratory in relation to the travelling salesman problem. In the first experiment, pigeons were placed in the corner of a lab room and allowed to fly to nearby feeders containing peas. The researchers found that pigeons largely used proximity to determine which feeder they would select next. In the second experiment, the feeders were arranged in such a way that flying to the nearest feeder at every opportunity would be largely inefficient if the pigeons needed to visit every feeder. The results of the second experiment indicate that pigeons, while still favoring proximity-based solutions, "can plan several steps ahead along the route when the differences in travel costs between efficient and less efficient routes based on proximity become larger."^[72] These results are consistent with other experiments done with non-primates, which have proven that some non-primates were able to plan complex travel routes. This suggests non-primates may possess a relatively sophisticated spatial cognitive ability.

Natural computation

When presented with a spatial configuration of food sources, the [amoeboid Physarum polycephalum](#) adapts its morphology to create an efficient path between the food sources which can also be viewed as an approximate solution to TSP.^[73]

Benchmarks

For benchmarking of TSP algorithms, [TSPLIB](#)^[74] is a library of sample instances of the TSP and related problems is maintained, see the [TSPLIB external reference](#). Many of them are lists of actual cities and layouts of actual [printed circuits](#).

Popular culture

- *Travelling Salesman*, by director Timothy Lanzone, is the story of four mathematicians hired by the U.S. government to solve the most elusive problem in computer-science history: [P vs. NP](#).^[75]

See also

- [Canadian traveller problem](#)
- [Exact algorithm](#)
- [Route inspection problem](#) (also known as "Chinese postman problem")
- [Set TSP problem](#)
- [Seven Bridges of Königsberg](#)
- [Steiner travelling salesman problem](#)
- [Subway Challenge](#)

- Tube Challenge
- Vehicle routing problem
- Graph exploration

Notes

1. "Search for "Traveling Salesperson Problem" " (<https://scholar.google.com/scholar?q=%22traveling+salesperson+problem%22>). Google Scholar. Retrieved 23 November 2019.
2. See the TSP world tour problem which has already been solved to within 0.05% of the optimal solution. [1] (<http://www.math.uwaterloo.ca/tsp/world/>)
3. Ross, I. M.; Proulx, R. J.; Karpenko, M. (6 May 2020). "An Optimal Control Theory for the Traveling Salesman Problem and Its Variants". *arXiv:2005.03186* (<https://arxiv.org/abs/2005.03186>) [math.OC (<https://arxiv.org/archive/math>.OC)].
4. Ross, I. M.; Proulx, R. J.; Karpenko, M. (July 2019). "Autonomous UAV Sensor Planning, Scheduling and Maneuvering: An Obstacle Engagement Technique" (<https://dx.doi.org/10.23919/acc.2019.8814474>). *2019 American Control Conference (ACC)*. IEEE: 65–70. doi:10.23919/acc.2019.8814474 (<https://doi.org/10.23919%2Facc.2019.8814474>). ISBN 978-1-5386-7926-5. S2CID 201811463 (<https://api.semanticscholar.org/CorpusID:201811463>).
5. "Der Handlungsreisende – wie er sein soll und was er zu tun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein – von einem alten Commis-Voyageur" (https://zs.thulb.uni-jena.de/receive/jportal_jp_article_00248075) (The travelling salesman – how he must be and what he should do in order to get commissions and be sure of the happy success in his business – by an old *commis-voyageur*)
6. A discussion of the early work of Hamilton and Kirkman can be found in *Graph Theory, 1736–1936* by Biggs, Lloyd, and Wilson (Clarendon Press, 1986).
7. Cited and English translation in Schrijver (2005). Original German: "Wir bezeichnen als *Botenproblem* (weil diese Frage in der Praxis von jedem Postboten, übrigens auch von vielen Reisenden zu lösen ist) die Aufgabe, für endlich viele Punkte, deren paarweise Abstände bekannt sind, den kürzesten die Punkte verbindenden Weg zu finden. Dieses Problem ist natürlich stets durch endlich viele Versuche lösbar. Regeln, welche die Anzahl der Versuche unter die Anzahl der Permutationen der gegebenen Punkte herunterdrücken würden, sind nicht bekannt. Die Regel, man solle vom Ausgangspunkt erst zum nächstgelegenen Punkt, dann zu dem diesem nächstgelegenen Punkt gehen usw., liefert im allgemeinen nicht den kürzesten Weg."
8. Lawler, E. L. (1985). *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization* (<https://books.google.com/books?id=qbFIMwEACAAJ>) (Repr. with corrections. ed.). John Wiley & sons. ISBN 978-0471904137.
9. Robinson, Julia (5 December 1949). "On the Hamiltonian game (a traveling salesman problem)" (<http://www.dtic.mil/gettr-doc/pdf?AD=AD0204961>). *Project Rand*. Santa Monica, CA: The Rand Corporation (RM-303). Retrieved 2 May 2020.
10. A detailed treatment of the connection between Menger and Whitney as well as the growth in the study of TSP can be found in Schrijver (2005).
11. Beardwood, Halton & Hammersley (1959).
12. van Bevern, René; Slugina, Viktoriia A. (2020). "A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem". *Historia Mathematica*. **53**: 118–127. arXiv:2004.02437 (<https://arxiv.org/abs/2004.02437>). doi:10.1016/j.hm.2020.04.003 (<https://doi.org/10.1016%2Fj.hm.2020.04.003>). S2CID 214803097 (<https://api.semanticscholar.org/CorpusID:214803097>).
13. Klarreich, Erica (30 January 2013). "Computer Scientists Find New Shortcuts for Infamous Traveling Salesman Problem" (<https://www.wired.com/2013/01/traveling-salesman-problem/>). *WIRED*. Retrieved 14 June 2015.
14. Klarreich, Erica (8 October 2020). "Computer Scientists Break Traveling Salesperson Record" (<https://www.quantamagazine.org/computer-scientists-break-traveling-salesperson-record-20201008/>). *Quanta Magazine*. Retrieved 13 October 2020.
15. Karlin, Anna R.; Klein, Nathan; Gharan, Shayan Oveis (30 August 2020). "A (Slightly) Improved Approximation Algorithm for Metric TSP". *arXiv:2007.01409* (<https://arxiv.org/abs/2007.01409>) [cs.DS (<https://arxiv.org/archive/cs>.DS)].
16. Rego, César; Gamboa, Dorabela; Glover, Fred; Osterman, Colin (2011), "Traveling salesman problem heuristics: leading methods, implementations and latest advances", *European Journal of Operational Research*, **211** (3): 427–441, doi:10.1016/j.ejor.2010.09.010 (<https://doi.org/10.1016%2Fj.ejor.2010.09.010>), MR 2774420 (<https://www.ams.org/mathscinet-getitem?mr=2774420>).
17. "How Do You Fix School Bus Routes? Call MIT in Wall street Journal" (http://online.wsj.com/public/resources/documents/printWSJ_-A002-20170812.pdf) (PDF).
18. Behzad, Arash; Modarres, Mohammad (2002), "New Efficient Transformation of the Generalized Traveling Salesman Problem into Traveling Salesman Problem", *Proceedings of the 15th International Conference of Systems Engineering (Las Vegas)*
19. Papadimitriou, C.H.; Steiglitz, K. (1998), *Combinatorial optimization: algorithms and complexity*, Mineola, NY: Dover, pp.308-309.
20. Tucker, A. W. (1960), "On Directed Graphs and Integer Programs", IBM Mathematical research Project (Princeton University)
21. Dantzig, George B. (1963), *Linear Programming and Extensions*, Princeton, NJ: PrincetonUP, pp. 545–7, ISBN 0-691-08000-3, sixth printing, 1974.

22. Velednitsky, Mark (2017). "Short combinatorial proof that the DFJ polytope is contained in the MTZ polytope for the Asymmetric Traveling Salesman Problem". *Operations Research Letters*. **45** (4): 323–324. arXiv:1805.06997 (<https://arxiv.org/abs/1805.06997>). doi:10.1016/j.orl.2017.04.010 (<https://doi.org/10.1016%2Fj.orl.2017.04.010>). S2CID 6941484 (<https://api.semanticscholar.org/CorpusID:6941484>).
23. Bektaş, Tolga; Gouveia, Luis (2014). "Requiem for the Miller–Tucker–Zemlin subtour elimination constraints?". *European Journal of Operational Research*. **236** (3): 820–832. doi:10.1016/j.ejor.2013.07.038 (<https://doi.org/10.1016%2Fj.ejor.2013.07.038>).
24. Bellman (1960), Bellman (1962), Held & Karp (1962)
25. Woeginger (2003).
26. Padberg & Rinaldi (1991).
27. Applegate, David; Bixby, Robert; Chvátal, Vašek; Cook, William; Helsgaun, Keld (June 2004). "Optimal Tour of Sweden" (<http://www.math.uwaterloo.ca/tsp/sweden/>). Retrieved 11 November 2020.
28. Johnson, D. S.; McGeoch, L. A. (1997). "The Traveling Salesman Problem: A Case Study in Local Optimization" (<https://www.cs.ubc.ca/~hutter/previous-earg/EmpAlgReadingGroup/TSP-JohMcg97.pdf>) (PDF). In Aarts, E. H. L.; Lenstra, J. K. (eds.). *Local Search in Combinatorial Optimisation*. London: John Wiley and Sons Ltd. pp. 215–310.
29. Gutina, Gregory; Yeob, Anders; Zverovich, Alexey (15 March 2002). "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP". *Discrete Applied Mathematics*. **117** (1–3): 81–86. doi:10.1016/S0166-218X(01)00195-0 (<https://doi.org/10.1016%2FS0166-218X%2801%2900195-0>).>
30. Zverovitch, Alexei; Zhang, Weixiong; Yeo, Anders; McGeoch, Lyle A.; Gutin, Gregory; Johnson, David S. (2007), "Experimental Analysis of Heuristics for the ATSP", *The Traveling Salesman Problem and Its Variations*, Combinatorial Optimization, Springer, Boston, MA, pp. 445–487, CiteSeerX 10.1.1.24.2386 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.2386>), doi:10.1007/0-306-48213-4_10 (https://doi.org/10.1007%2F0-306-48213-4_10), ISBN 978-0-387-44459-8
31. Rosenkrantz, D. J.; Stearns, R. E.; Lewis, P. M. (14–16 October 1974). *Approximate algorithms for the traveling salesman problem*. 15th Annual Symposium on Switching and Automata Theory (swat 1974). doi:10.1109/SWAT.1974.4 (<https://doi.org/10.1109%2FSWAT.1974.4>).
32. Ray, S. S.; Bandyopadhyay, S.; Pal, S. K. (2007). "Genetic Operators for Combinatorial Optimization in TSP and Microarray Gene Ordering". *Applied Intelligence*. **26** (3): 183–195. CiteSeerX 10.1.1.151.132 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.151.132>). doi:10.1007/s10489-006-0018-y (<https://doi.org/10.1007%2Fs10489-006-0018-y>). S2CID 8130854 (<https://api.semanticscholar.org/CorpusID:8130854>).
33. Kahng, A. B.; Reda, S. (2004). "Match Twice and Stitch: A New TSP Tour Construction Heuristic". *Operations Research Letters*. **32** (6): 499–509. doi:10.1016/j.orl.2004.04.001 (<https://doi.org/10.1016%2Fj.orl.2004.04.001>).
34. Dorigo, Marco; Gambardella, Luca Maria (1997). "Ant Colonies for the Traveling Salesman Problem". *Biosystems*. **43** (2): 73–81. CiteSeerX 10.1.1.54.7734 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.7734>). doi:10.1016/S0303-2647(97)01708-5 (<https://doi.org/10.1016%2FS0303-2647%2897%2901708-5>). PMID 9231906 (<http://pubmed.ncbi.nlm.nih.gov/9231906>). S2CID 8243011 (<https://api.semanticscholar.org/CorpusID:8243011>).
35. Papadimitriou (1977).
36. Allender et al. (2007).
37. Larson & Odoni (1981).
38. Arora (1998).
39. Jonker, Roy; Volgenant, Ton (1983). "Transforming asymmetric into symmetric traveling salesman problems". *Operations Research Letters*. **2** (161–163): 1983. doi:10.1016/0167-6377(83)90048-2 (<https://doi.org/10.1016%2F0167-6377%2883%2990048-2>).
40. Arlotto, Alessandro; Steele, J. Michael (2016), "Beardwood–Halton–Hammersley theorem for stationary ergodic sequences: a counterexample", *The Annals of Applied Probability*, **26** (4): 2141–2168, arXiv:1307.0221 (<https://arxiv.org/abs/1307.0221>), doi:10.1214/15-AAP1142 (<https://doi.org/10.1214%2F15-AAP1142>), S2CID 8904077 (<https://api.semanticscholar.org/CorpusID:8904077>)
41. Few, L. (1955). "The shortest path and the shortest road through n points". *Mathematika*. **2** (2): 141–144. doi:10.1112/s0025579300000784 (<https://doi.org/10.1112%2Fs0025579300000784>).
42. Fiechter, C.-N. (1994). "A parallel tabu search algorithm for large traveling salesman problems". *Disc. Applied Math*. **51** (3): 243–267. doi:10.1016/0166-218X(92)00033-I (<https://doi.org/10.1016%2F0166-218X%2892%2900033-I>).
43. Steinerberger (2015).
44. Held, M.; Karp, R.M. (1970). "The Traveling Salesman Problem and Minimum Spanning Trees". *Operations Research*. **18** (6): 1138–1162. doi:10.1287/opre.18.6.1138 (<https://doi.org/10.1287%2Fopre.18.6.1138>).
45. Goemans, M.; Bertsimas, D. (1991). "Probabilistic analysis of the Held and Karp lower bound for the Euclidean traveling salesman problem". *Mathematics of Operations Research*. **16** (1): 72–89. doi:10.1287/moor.16.1.72 (<https://doi.org/10.1287%2Fmoor.16.1.72>).
46. "error" (<https://about.att.com/error.html>). *about.att.com*.
47. Christine L. Valenzuela and Antonia J. Jones (<http://users.cs.cf.ac.uk/Antonia.J.Jones/Papers/EJORHeldKarp/HeldKarp.pdf>) Archived (<https://web.archive.org/web/20071025205411/http://users.cs.cf.ac.uk/Antonia.J.Jones/Papers/EJORHeldKarp/HeldKarp.pdf>) 25 October 2007 at the Wayback Machine
48. Orponen, P.; Mannila, H. (1987). On approximation preserving reductions: Complete problems and robust measures' (Report). Department of Computer Science, University of Helsinki. Technical Report C-1987–28.

49. Papadimitriou & Yannakakis (1993).
50. Christofides (1976).
51. Serdyukov, Anatoliy I. (1978), "О некоторых экстремальных обходах в графах" (http://nas1.math.nsc.ru/aim/journals/us/us17/us17_007.pdf) [On some extremal walks in graphs] (PDF), *Upravlyaemye Sistemy* (in Russian), **17**: 76–79
52. Karlin, Anna R.; Klein, Nathan; Gharan, Shayan Oveis (30 August 2020). "A (Slightly) Improved Approximation Algorithm for Metric TSP". *arXiv:2007.01409* (<https://arxiv.org/abs/2007.01409>) [cs (<https://arxiv.org/archive/cs>)].
53. Karpinski, Lampis & Schmied (2015).
54. Berman & Karpinski (2006).
55. Kaplan et al. (2004).
56. Svensson, Ola; Tarnawski, Jakub; Végh, László A. (2018). "A constant-factor approximation algorithm for the asymmetric traveling salesman problem" (<http://dl.acm.org/citation.cfm?doid=3188745.3188824>). *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing - STOC 2018*. Los Angeles, CA, USA: ACM Press: 204–213. doi:10.1145/3188745.3188824 (<https://doi.org/10.1145/3188745.3188824>). ISBN 978-1-4503-5559-9. S2CID 12391033 (<https://api.semanticscholar.org/CorpusID:12391033>).
57. Traub, Vera; Vygen, Jens (8 June 2020). "An improved approximation algorithm for ATSP" (<https://dl.acm.org/doi/10.1145/3357713.3384233>). *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. Chicago, IL: ACM: 1–13. arXiv:1912.00670 (<https://arxiv.org/abs/1912.00670>). doi:10.1145/3357713.3384233 (<https://doi.org/10.1145/3357713.3384233>). ISBN 978-1-4503-6979-4. S2CID 208527125 (<https://api.semanticscholar.org/CorpusID:208527125>).
58. Kosaraju, Park & Stein (1994).
59. Serdyukov (1984).
60. Hassin & Rubinstein (2000).
61. Macgregor, J. N.; Ormerod, T. (June 1996), "Human performance on the traveling salesman problem", *Perception & Psychophysics*, **58** (4): 527–539, doi:10.3758/BF03213088 (<https://doi.org/10.3758/BF03213088>), PMID 8934685 (<https://pubmed.ncbi.nlm.nih.gov/8934685>).
62. Dry, Matthew; Lee, Michael D.; Vickers, Douglas; Hughes, Peter (2006). "Human Performance on Visually Presented Traveling Salesperson Problems with Varying Numbers of Nodes". *The Journal of Problem Solving*. **1** (1). CiteSeerX 10.1.1.360.9763 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.360.9763>). doi:10.7771/1932-6246.1004 (<https://doi.org/10.7771/1932-6246.1004>). ISSN 1932-6246 (<https://www.worldcat.org/issn/1932-6246>).
63. Rooij, Iris Van; Stege, Ulrike; Schactman, Alissa (1 March 2003). "Convex hull and tour crossings in the Euclidean traveling salesperson problem: Implications for human performance studies". *Memory & Cognition*. **31** (2): 215–220. CiteSeerX 10.1.1.12.6117 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.6117>). doi:10.3758/bf03194380 (<https://doi.org/10.3758/bf03194380>). ISSN 0090-502X (<https://www.worldcat.org/issn/0090-502X>). PMID 12749463 (<https://pubmed.ncbi.nlm.nih.gov/12749463>). S2CID 18989303 (<https://api.semanticscholar.org/CorpusID:18989303>).
64. MacGregor, James N.; Chu, Yun (2011). "Human Performance on the Traveling Salesman and Related Problems: A Review" (<https://doi.org/10.7771/1932-6246.1090>). *The Journal of Problem Solving*. **3** (2). doi:10.7771/1932-6246.1090 (<https://doi.org/10.7771/1932-6246.1090>). ISSN 1932-6246 (<https://www.worldcat.org/issn/1932-6246>).
65. MacGregor, James N.; Chronicle, Edward P.; Ormerod, Thomas C. (1 March 2004). "Convex hull or crossing avoidance? Solution heuristics in the traveling salesperson problem" (<https://doi.org/10.3758/bf03196857>). *Memory & Cognition*. **32** (2): 260–270. doi:10.3758/bf03196857 (<https://doi.org/10.3758/bf03196857>). ISSN 0090-502X (<https://www.worldcat.org/issn/0090-502X>). PMID 15190718 (<https://pubmed.ncbi.nlm.nih.gov/15190718>).
66. Vickers, Douglas; Mayo, Therese; Heitmann, Megan; Lee, Michael D; Hughes, Peter (2004). "Intelligence and individual differences in performance on three types of visually presented optimisation problems". *Personality and Individual Differences*. **36** (5): 1059–1071. doi:10.1016/s0191-8869(03)00200-9 ([https://doi.org/10.1016/s0191-8869\(03\)00200-9](https://doi.org/10.1016/s0191-8869(03)00200-9)).
67. Kyritsis, Markos; Gulliver, Stephen R.; Feredoes, Eva (12 June 2017). "Acknowledging crossing-avoidance heuristic violations when solving the Euclidean travelling salesperson problem". *Psychological Research*. **82** (5): 997–1009. doi:10.1007/s00426-017-0881-7 (<https://doi.org/10.1007/s00426-017-0881-7>). ISSN 0340-0727 (<https://www.worldcat.org/issn/0340-0727>). PMID 28608230 (<https://pubmed.ncbi.nlm.nih.gov/28608230>). S2CID 3959429 (<https://api.semanticscholar.org/CorpusID:3959429>).
68. Kyritsis, Markos; Blathras, George; Gulliver, Stephen; Varela, Vasiliki-Alexia (11 January 2017). "Sense of direction and conscientiousness as predictors of performance in the Euclidean travelling salesman problem" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5727545>). *Heliyon*. **3** (11): e00461. doi:10.1016/j.heliyon.2017.e00461 (<https://doi.org/10.1016/j.heliyon.2017.e00461>) (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5727545>). PMID 29264418 (<https://pubmed.ncbi.nlm.nih.gov/29264418>).
69. Kyritsis, Markos; Gulliver, Stephen R.; Feredoes, Eva; Din, Shahab Ud (December 2018). "Human behaviour in the Euclidean Travelling Salesperson Problem: Computational modelling of heuristics and figural effects". *Cognitive Systems Research*. **52**: 387–399. doi:10.1016/j.cogsys.2018.07.027 (<https://doi.org/10.1016/j.cogsys.2018.07.027>). S2CID 53761995 (<https://api.semanticscholar.org/CorpusID:53761995>).
70. MacGregor, James N.; Chu, Yun (2011), "Human performance on the traveling salesman and related problems: A review" (<https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1090&context=jps>), *Journal of Problem Solving*, **3** (2), doi:10.7771/1932-6246.1090 (<https://doi.org/10.7771/1932-6246.1090>).
71. *Journal of Problem Solving* 1(1) (<https://docs.lib.purdue.edu/jps/vol1/iss1/>), 2006, retrieved 2014-06-06.

72. Gibson, Brett; Wilkinson, Matthew; Kelly, Debbie (1 May 2012). "Let the pigeon drive the bus: pigeons can plan future routes in a room". *Animal Cognition*. **15** (3): 379–391. doi:10.1007/s10071-011-0463-9 (<https://doi.org/10.1007/s10071-011-0463-9>). ISSN 1435-9456 (<https://www.worldcat.org/issn/1435-9456>). PMID 21965161 (<https://pubmed.ncbi.nlm.nih.gov/21965161/>). S2CID 14994429 (<https://api.semanticscholar.org/CorpusID:14994429>).
73. Jones, Jeff; Adamatzky, Andrew (2014), "Computation of the travelling salesman problem by a shrinking blob" (<http://www.phychip.eu/wp-content/uploads/2013/03/Computation-of-the-travelling-salesman-problem-by-a-shrinking-blob.pdf>) (PDF), *Natural Computing*: 2, 13, arXiv:1303.4969 (<https://arxiv.org/abs/1303.4969>), Bibcode:2013arXiv1303.4969J (<https://ui.adsabs.harvard.edu/abs/2013arXiv1303.4969J>)
74. "TSPLIB" (<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>). *comopt.ifl.uni-heidelberg.de*. Retrieved 10 October 2020.
75. Geere, Duncan (26 April 2012). "'Travelling Salesman' movie considers the repercussions if P equals NP" (<https://www.wired.co.uk/news/archive/2012-04/26/travelling-salesman>). *Wired UK*. Retrieved 26 April 2012.

References

- Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. (2006), *The Traveling Salesman Problem*, ISBN 978-0-691-12993-8.
- Allender, Eric; Bürgisser, Peter; Kjeldgaard-Pedersen, Johan; Mitersen, Peter Bro (2007), "On the Complexity of Numerical Analysis" (<https://www3.math.tu-berlin.de/algebra/work/focs7.pdf>) (PDF), *SIAM J. Comput.*, **38** (5): 1987–2006, CiteSeerX 10.1.1.167.5495 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.167.5495>), doi:10.1137/070697926 (<https://doi.org/10.1137/070697926>).
- Arora, Sanjeev (1998), "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems" (<http://graphics.stanford.edu/courses/cs468-06-winter/Papers/arora-tsp.pdf>) (PDF), *Journal of the ACM*, **45** (5): 753–782, doi:10.1145/290179.290180 (<https://doi.org/10.1145/290179.290180>), MR 1668147 (<https://www.ams.org/mathscinet-getitem?mr=1668147>), S2CID 3023351 (<https://api.semanticscholar.org/CorpusID:3023351>).
- Beardwood, J.; Halton, J.H.; Hammersley, J.M. (October 1959), "The Shortest Path Through Many Points" (https://www.cambridge.org/core/product/identifier/S0305004100034095/type/journal_article), *Proceedings of the Cambridge Philosophical Society*, **55** (4): 299–327, Bibcode:1959PCPS...55..299B (<https://ui.adsabs.harvard.edu/abs/1959PCPS...55..299B>), doi:10.1017/s0305004100034095 (<https://doi.org/10.1017/s0305004100034095>), ISSN 0305-0041 (<http://www.worldcat.org/issn/0305-0041>).
- Bellman, R. (1960), "Combinatorial Processes and Dynamic Programming", in Bellman, R.; Hall, M. Jr. (eds.), *Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics 10*, American Mathematical Society, pp. 217–249.
- Bellman, R. (1962), "Dynamic Programming Treatment of the Travelling Salesman Problem", *J. Assoc. Comput. Mach.*, **9**: 61–63, doi:10.1145/321105.321111 (<https://doi.org/10.1145/321105.321111>), S2CID 15649582 (<https://api.semanticscholar.org/CorpusID:15649582>).
- Berman, Piotr; Karpinski, Marek (2006), "8/7-approximation algorithm for (1,2)-TSP", *Proc. 17th ACM-SIAM Symposium on Discrete Algorithms (SODA '06)* (<http://eccc.hpi-web.de/report/2005/069/revision/2/download/>), pp. 641–648, CiteSeerX 10.1.1.430.2224 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.430.2224>), doi:10.1145/1109557.1109627 (<https://doi.org/10.1145/1109557.1109627>), ISBN 978-0898716054, S2CID 9054176 (<https://api.semanticscholar.org/CorpusID:9054176>), ECCC TR05-069 (<https://eccc.weizmann.ac.il/report/2005/069/>).
- Christofides, N. (1976), *Worst-case analysis of a new heuristic for the travelling salesman problem*, Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh.
- Hassin, R.; Rubinstein, S. (2000), "Better approximations for max TSP", *Information Processing Letters*, **75** (4): 181–186, CiteSeerX 10.1.1.35.7209 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.7209>), doi:10.1016/S0020-0190(00)00097-1 ([https://doi.org/10.1016/S0020-0190\(00\)00097-1](https://doi.org/10.1016/S0020-0190(00)00097-1)).
- Held, M.; Karp, R. M. (1962), "A Dynamic Programming Approach to Sequencing Problems", *Journal of the Society for Industrial and Applied Mathematics*, **10** (1): 196–210, doi:10.1137/0110015 (<https://doi.org/10.1137/0110015>), hdl:10338.dmlcz/103900 (<https://hdl.handle.net/10338.dmlcz/103900>).
- Kaplan, H.; Lewenstein, L.; Shafrir, N.; Sviridenko, M. (2004), "Approximation Algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs", in *Proc. 44th IEEE Symp. on Foundations of Comput. Sci.*, pp. 56–65.
- Karpinski, M.; Lampis, M.; Schmied, R. (2015), "New Inapproximability bounds for TSP", *Journal of Computer and System Sciences*, **81** (8): 1665–1677, arXiv:1303.6437 (<https://arxiv.org/abs/1303.6437>), doi:10.1016/j.jcss.2015.06.003 (<https://doi.org/10.1016/j.jcss.2015.06.003>).
- Kosaraju, S. R.; Park, J. K.; Stein, C. (1994), "Long tours and short superstrings", *Proc. 35th Ann. IEEE Symp. on Foundations of Comput. Sci.*, IEEE Computer Society, pp. 166–177.
- Larson, Richard C.; Odoni, Amedeo R. (1981), "6.4.7: Applications of Network Models § Routing Problems §§ Euclidean TSP" (http://web.mit.edu/urban_or_book/www/book/chapter6/6.4.7.html), *Urban Operations Research*, Prentice-Hall, ISBN 9780139394478, OCLC 6331426 (<https://www.worldcat.org/oclc/6331426>).
- Padberg, M.; Rinaldi, G. (1991), "A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems" (<https://semanticscholar.org/paper/966580e91b06f55605869b7155784c2fc0c444bb>), *SIAM Review*, **33**: 60–100, doi:10.1137/1033004 (<https://doi.org/10.1137/1033004>), S2CID 18516138 (<https://api.semanticscholar.org/CorpusID:18516138>).

- Papadimitriou, Christos H. (1977), "The Euclidean traveling salesman problem is NP-complete", *Theoretical Computer Science*, 4 (3): 237–244, doi:10.1016/0304-3975(77)90012-3 (<https://doi.org/10.1016%2F0304-3975%2877%2990012-3>), MR 0455550 (<https://www.ams.org/mathscinet-getitem?mr=0455550>).
- Papadimitriou, C. H.; Yannakakis, M. (1993), "The traveling salesman problem with distances one and two", *Math. Oper. Res.*, 18: 1–11, doi:10.1287/moor.18.1.1 (<https://doi.org/10.1287%2Fmoor.18.1.1>).
- Schrijver, Alexander (2005). "On the history of combinatorial optimization (till 1960)". In K. Aardal; G.L. Nemhauser; R. Weismantel (eds.). *Handbook of Discrete Optimization* (<http://homepages.cwi.nl/~lex/files/histco.pdf>) (PDF). Amsterdam: Elsevier. pp. 1–68.
- Serdyukov, A. I. (1984), "An algorithm with an estimate for the traveling salesman problem of the maximum'", *Upravlyaemye Sistemy*, 25: 80–86.
- Steinerberger, Stefan (2015), "New Bounds for the Traveling Salesman Constant", *Advances in Applied Probability*, 47 (1): 27–36, arXiv:1311.6338 (<https://arxiv.org/abs/1311.6338>), Bibcode:2013arXiv1311.6338S (<https://ui.adsabs.harvard.edu/abs/2013arXiv1311.6338S>), doi:10.1239/aap/1427814579 (<https://doi.org/10.1239%2Faap%2F1427814579>), S2CID 119293287 (<https://api.semanticscholar.org/CorpusID:119293287>).
- Woeginger, G.J. (2003), "Exact Algorithms for NP-Hard Problems: A Survey", *Combinatorial Optimization – Eureka, You Shrink! Lecture notes in computer science*, vol. 2570, Springer, pp. 185–207.

Further reading

- Adleman, Leonard (1994), "Molecular Computation of Solutions To Combinatorial Problems" (<https://web.archive.org/web/20050206144827/http://www.usc.edu/dept/molecular-science/papers/fp-sci94.pdf>) (PDF), *Science*, 266 (5187): 1021–4, Bibcode:1994Sci...266.1021A (<https://ui.adsabs.harvard.edu/abs/1994Sci...266.1021A>), CiteSeerX 10.1.1.54.2565 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.2565>), doi:10.1126/science.7973651 (<https://doi.org/10.1126%2Fscience.7973651>), PMID 7973651 (<https://pubmed.ncbi.nlm.nih.gov/7973651>), archived from the original (<http://www.usc.edu/dept/molecular-science/papers/fp-sci94.pdf>) (PDF) on 6 February 2005
- Babin, Gilbert; Deneault, Stéphanie; Laportey, Gilbert (2005), "Improvements to the Or-opt Heuristic for the Symmetric Traveling Salesman Problem", *The Journal of the Operational Research Society*, Cahiers du GERAD, Montreal: Group for Research in Decision Analysis, G-2005-02 (3): 402–407, CiteSeerX 10.1.1.89.9953 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.9953>), JSTOR 4622707 (<https://www.jstor.org/stable/4622707>)
- Cook, William (2012). *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press. ISBN 9780691152707.
- Cook, William; Espinoza, Daniel; Goycoolea, Marcos (2007), "Computing with domino-parity inequalities for the TSP", *INFORMS Journal on Computing*, 19 (3): 356–365, doi:10.1287/ijoc.1060.0204 (<https://doi.org/10.1287%2Fijoc.1060.0204>)
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (31 July 2009). "35.2: The traveling-salesman problem" (<https://books.google.com/books?id=i-bUBQAAQBAJ>). *Introduction to Algorithms* (2nd ed.). MIT Press. pp. 1027–1033. ISBN 9780262033848.
- Dantzig, G. B.; Fulkerson, R.; Johnson, S. M. (1954), "Solution of a large-scale traveling salesman problem" (<https://semanticscholar.org/paper/0fa37740fe865bcac0d9687c0e5f65131f4492c8>), *Operations Research*, 2 (4): 393–410, doi:10.1287/opre.2.4.393 (<https://doi.org/10.1287%2Fopre.2.4.393>), JSTOR 166695 (<https://www.jstor.org/stable/166695>), S2CID 44960960 (<https://api.semanticscholar.org/CorpusID:44960960>)
- Garey, Michael R.; Johnson, David S. (1979). "A2.3: ND22–24". *Computers and Intractability: A Guide to the Theory of NP-completeness* (<https://books.google.com/books?id=fjxGAQAIAAJ>). W. H. Freeman. pp. 211–212. ISBN 9780716710448.
- Goldberg, D. E. (1989), "Genetic Algorithms in Search, Optimization & Machine Learning", *Reading: Addison-Wesley*, New York: Addison-Wesley, Bibcode:1989gaso.book.....G (<https://ui.adsabs.harvard.edu/abs/1989gaso.book.....G>), ISBN 978-0-201-15767-3
- Gutin, G.; Yeo, A.; Zverovich, A. (15 March 2002). "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP". *Discrete Applied Mathematics*. 117 (1–3): 81–86. doi:10.1016/S0166-218X(01)00195-0 (<https://doi.org/10.1016%2FS0166-218X%2801%2900195-0>). ISSN 0166-218X (<https://www.worldcat.org/issn/0166-218X>).
- Gutin, G.; Punnen, A. P. (18 May 2007). *The Traveling Salesman Problem and Its Variations* (<https://books.google.com/books?id=pfRSPwAACAAJ>). Springer US. ISBN 9780387444598.
- Johnson, D. S.; McGeoch, L. A. (1997), "The Traveling Salesman Problem: A Case Study in Local Optimization", in Aarts, E. H. L.; Lenstra, J. K. (eds.), *Local Search in Combinatorial Optimisation* (<https://www.cs.ubc.ca/~hutter/previous-earg/EmpAlgReadingGroup/TSP-JohMcg97.pdf>) (PDF), John Wiley and Sons Ltd., pp. 215–310
- Lawler, E. L.; Shmoys, D. B.; Kan, A. H. G. Rinnooy; Lenstra, J. K. (1985). *The Traveling Salesman Problem* (<https://books.google.com/books?id=BXBGAAAAYAAJ>). John Wiley & Sons, Incorporated. ISBN 9780471904137.
- MacGregor, J. N.; Ormerod, T. (1996), "Human performance on the traveling salesman problem", *Perception & Psychophysics*, 58 (4): 527–539, doi:10.3758/BF03213088 (<https://doi.org/10.3758%2FBF03213088>), PMID 8934685 (<https://pubmed.ncbi.nlm.nih.gov/8934685>), S2CID 38355042 (<https://api.semanticscholar.org/CorpusID:38355042>)

- Medvedev, Andrei; Lee, Michael; Butavicius, Marcus; Vickers, Douglas (1 February 2001). "Human performance on visually presented Traveling Salesman problems". *Psychological Research*. **65** (1): 34–45. doi:10.1007/s004260000031 (https://doi.org/10.1007%2Fs004260000031). ISSN 1430-2772 (https://www.worldcat.org/issn/1430-2772). PMID 11505612 (https://pubmed.ncbi.nlm.nih.gov/11505612). S2CID 8986203 (https://api.semanticscholar.org/CorpusID:8986203).
- Mitchell, J. S. B. (1999), "Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, *k*-MST, and related problems", *SIAM Journal on Computing*, **28** (4): 1298–1309, doi:10.1137/S0097539796309764 (https://doi.org/10.1137%2FS0097539796309764)
- Rao, S.; Smith, W. (1998), "Approximating geometrical graphs via 'spanners' and 'banyans' ", *Proceedings*, pp. 540–550, CiteSeerX 10.1.1.51.8676 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.8676) Unknown parameter |conference= ignored (help)
- Rosenkrantz, Daniel J.; Stearns, Richard E.; Lewis, Philip M., II (1977). "An Analysis of Several Heuristics for the Traveling Salesman Problem" (https://semanticscholar.org/paper/ac5e93895ab03bf47070dab04f62f58717442f0d). *SIAM Journal on Computing*. SIAM (Society for Industrial and Applied Mathematics). **6** (5): 563–581. doi:10.1137/0206041 (https://doi.org/10.1137%2F0206041). S2CID 14764079 (https://api.semanticscholar.org/CorpusID:14764079).
- Ross, I. M., Proulx, R. J., Karpenko, M. (2020). *An optimal control theory for the traveling salesman problem and its variants*. arXiv 2005.03186.
- Walshaw, Chris (2000), *A Multilevel Approach to the Travelling Salesman Problem*, CMS Press
- Walshaw, Chris (2001), *A Multilevel Lin-Kernighan-Helsgaun Algorithm for the Travelling Salesman Problem* (http://dimacs.rutgers.edu/Challenges/TSP/WalshawTR8001.ps), CMS Press

External links

- *Traveling Salesman Problem* (https://web.archive.org/web/20131217224319/http://www.math.uwaterloo.ca/tsp/index.html) at the *Wayback Machine* (archived 17 December 2013) at *University of Waterloo*
- *TSPLIB* (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/) at the *University of Heidelberg*
- *Traveling Salesman Problem* (http://demonstrations.wolfram.com/TravelingSalesmanProblem/) by Jon McLoone at the *Wolfram Demonstrations Project*
- *TSP visualization tool* (https://tspvis.com/)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Travelling_salesman_problem&oldid=1030677047"

This page was last edited on 27 June 2021, at 12:01 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.