Determining the number of all solutions, however, is an NP-hard problem; see [IrLe86].

The analogous problem for the complete graph $K_{2n}$ (known as the *stable roommates problem*) is more difficult; for example, it cannot be solved for each choice of $n$ and $w$. Irving [Irv85] gave an algorithm which decides with complexity $O(n^2)$ whether there exists a solution and, if this is the case, actually finds one; see also [Gus88]. We recommend the excellent monograph [GuIr89] for further study of this type of problems; see also [BaRa97] for a nice exposition.

# 15

# A Hard Problem: The TSP

*Which way are you goin'...*
JIM CROCE

Up to now, we have investigated only those optimization problems which allow an *efficient* – that is, polynomial – algorithm. In contrast, this final chapter will deal with a typical NP-complete problem: the travelling salesman problem already introduced in Chapter 1. We saw in Chapter 2 that no efficient algorithms are known for NP-complete problems, and that it is actually quite likely that no such algorithms can exist. Now we address the question of how such *hard* problems – which regularly occur in practical applications – might be approached: one uses, for instance, approximation techniques, heuristics, relaxations, post-optimization, local optima, and complete enumeration. We shall explain these methods only for the TSP, but they are typical for dealing with hard problems in general.

We will also briefly mention a further extremely important approach to solving hard problems: *polyhedral combinatorics*. A detailed discussion of this vast area of research would far exceed the limits of this book; as mentioned before, the reader can find an encyclopedic treatment of the polyhedral approach to combinatorial optimization in [Schr03].

## 15.1 Basic definitions

Let us recall the formal definition of the TSP given in Section 1.4:

**Problem 15.1.1 (travelling salesman problem, TSP).** Let $w\colon E \to \mathbb{R}^+$ be a weight function on the complete graph $K_n$. We seek a cyclic permutation $(1, \pi(1), \ldots, \pi^{n-1}(1))$ of the vertex set $\{1, \ldots, n\}$ such that

$$w(\pi) = \sum_{i=1}^{n} w(\{i, \pi(i)\})$$

is minimal. We call any cyclic permutation $\pi$ of $\{1, \ldots, n\}$ as well as the corresponding Hamiltonian cycle

$$1 — \pi(1) — \ldots — \pi^{n-1}(1) — 1$$

in $K_n$ a *tour*; if $w(\pi)$ is minimal among all tours, $\pi$ is called an *optimal tour*. The weights of the edges will be given via a matrix $W$, as explained in Section 1.4.

We shall use the following example also already introduced in Section 1.4 to illustrate the various methods for finding a good solution of the TSP, which are the subject matter of this chapter.

**Example 15.1.2.** Determine an optimal tour for

| | *Aa* | *Ba* | *Be* | *Du* | *Fr* | *Ha* | *Mu* | *Nu* | *St* |
|---|---|---|---|---|---|---|---|---|---|
| *Aa* | 0 | 57 | 64 | 8 | 26 | 49 | 64 | 47 | 46 |
| *Ba* | 57 | 0 | 88 | 54 | 34 | 83 | 37 | 43 | 27 |
| *Be* | 64 | 88 | 0 | 57 | 56 | 29 | 60 | 44 | 63 |
| *Du* | 8 | 54 | 57 | 0 | 23 | 43 | 63 | 44 | 41 |
| *Fr* | 26 | 34 | 56 | 23 | 0 | 50 | 40 | 22 | 20 |
| *Ha* | 49 | 83 | 29 | 43 | 50 | 0 | 80 | 63 | 70 |
| *Mu* | 64 | 37 | 60 | 63 | 40 | 80 | 0 | 17 | 22 |
| *Nu* | 47 | 43 | 44 | 44 | 22 | 63 | 17 | 0 | 19 |
| *St* | 46 | 27 | 63 | 41 | 20 | 70 | 22 | 19 | 0 |

We saw in Theorem 2.7.5 that the TSP is NP-complete, so that we cannot expect to find an efficient algorithm for solving it. Nevertheless, this problem is extremely important in practice, and techniques for solving – or at least approximately solving – instances of considerable size are essential.

Indeed, there are many applications of the TSP which bear little resemblance to the original *travelling salesman* interpretation. To mention a simple example, we might have to prepare the machines in a plant for $n$ successive production processes. Let $w_{ij}$ denote the setup cost arising if process $j$ is scheduled immediately after process $i$; then the problem of finding an ordering for the $n$ processes which minimizes the total setup cost can be viewed as a TSP. In [GrJR91] the reader can find an interesting practical case study, which demonstrates the relevance of approximation techniques for solving the TSP to some tasks arising in the production of computers. A further impressive example is described in [BkSh89]: applying the TSP in X-ray crystallography resulted in dramatic savings in the amount of time a measuring process takes. Several further applications are discussed in [LeRi75] and in [LaLRS85, Chapter 2].

Note that the instance given in Example 15.1.2 has a rather special structure: the weights satisfy the triangle inequality $w_{ik} \leq w_{ij} + w_{jk}$. Of course, this holds whenever the weights stand for distances in the plane, or in a graph, and (more generally) whenever $W$ corresponds to a metric space; see Section 3.2. Hence the following definition



**Problem 15.1.3 (metric travelling salesman problem, ΔTSP).** Let $W = (w_{ij})$ be a symmetric matrix describing a TSP, and assume that $W$ satisfies the triangle inequality:

$$w_{ik} \leq w_{ij} + w_{jk} \qquad \text{for } i, j, k = 1, \ldots, n.$$

Then one calls the given TSP *metric* or, for short, a ΔTSP.

Note that the TSP used in the proof of Theorem 2.7.5 is clearly metric. Hence we have the following result:

**Theorem 15.1.4.** *ΔTSP is NP-complete.* □

Nevertheless, the metric property does make a difference in the degree of complexity of a TSP: in the metric case, there always exists a good approximation algorithm; most likely, this does not hold for the general case, where the triangle inequality is not assumed; see Section 15.4.

Let us conclude this section with a brief discussion of three further variants of the TSP.

**Problem 15.1.5 (asymmetric travelling salesman problem, ATSP).** Instead of $K_n$, we consider the complete directed graph $\vec{K}_n$ on $n$ vertices: we allow the weight matrix $W$ to be non-symmetric (but still with entries 0 on the main diagonal). This *asymmetric* TSP contains the usual TSP as a special case, and hence it is likewise NP-hard.

**Example 15.1.6.** We drop the condition that the travelling salesman should visit each city exactly once, so that we now consider not only Hamiltonian cycles, but also closed walks containing each vertex of $K_n$ at least once. If the given TSP is metric, any optimal tour will still be an optimal solution. However, this does not hold in general, as the example given in Figure 15.1 shows: here $(w, x, y, z, x, w)$ is a shortest closed walk (of length 6), but the shortest tour $(w, x, y, z, w)$ has length 8.

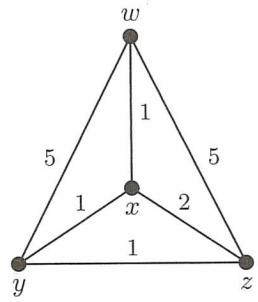


**Fig. 15.1.** A TSP for $n = 4$

Given a matrix $W = (w_{ij})$ not satisfying the triangle inequality, we may consider it as a matrix of lengths on $K_n$ and then calculate the corresponding distance matrix $D = (d_{ij})$. For example, we can use the algorithm of Floyd and Warshall for this purpose; see Section 3.8. Of course, $D$ satisfies the triangle inequality and, hence, defines a metric TSP. It is easy to see that the optimal closed walks with respect to $W$ correspond to the optimal tours with respect to $D$. Thus the seemingly more general problem described in Example 15.1.6 actually reduces to the metric TSP.

Finally, one may also consider an arbitrary connected graph $G$ with some length function $w$ instead of $K_n$. Then it is not at all clear whether any tours exist: we need to check first whether $G$ is Hamiltonian. As proved in Section 2.8, this feasibility question is already an NP-complete problem in itself.

## 15.2 Lower bounds: Relaxations

From a practical point of view, it will often be necessary (and also sufficient) to construct a reasonably good approximate solution instead of an optimal tour. For example, it will suffice for most practical applications if we can provide an efficient method for finding a solution which is at most 2% worse than the optimal tour: using a vast amount of resources for further improvement of the quality of the solution would not make any economic sense. In this context, note also that input data – distances, for example – always have a limited accuracy, so that it might not even mean much to have a truly optimal solution at our disposal.

In order to judge the quality of an approximate solution, we need lower bounds on the length of a tour, and these bounds should not only be strong but also easily computable – aims which are, of course, usually contradictory. A standard approach is the use of suitable *relaxations*: instead of the original problem $\mathbf{P}$, we consider a problem $\mathbf{P}'$ containing $\mathbf{P}$; this auxiliary (simpler) problem is obtained by a suitable weakening of the conditions defining $\mathbf{P}$. Then the weight $w(\mathbf{P}')$ of an optimal solution for $\mathbf{P}'$ is a lower bound for the weight $w(\mathbf{P})$ of an optimal solution for $\mathbf{P}$.[1]

Unfortunately, in many cases it is not possible to predict the quality of the approximation theoretically, so that we have to use empirical methods: for instance, comparing lower bounds found by relaxation with upper bounds given by solutions constructed by some heuristic. We shall consider various heuristics in Section 15.5; now we discuss several relaxations which have proved useful for dealing with TSP's. In this section, $\mathbf{P}$ is always a TSP on the complete graph $K_n$ on $V = \{1, \ldots, n\}$, given by a weight matrix $W = (w_{ij})$.

[1] Our discussion refers to the TSP, but applies to minimization problems in general. Of course, with appropriate adjustments, it can also be transferred to maximization problems.



### A. The assignment relaxation

One choice for $\mathbf{P}'$ is the assignment problem AP defined in Example 7.4.12 and studied in Chapter 14. Thus we seek a permutation $\pi$ of $\{1, \ldots, n\}$ for which $w_{1,\pi(1)} + \ldots + w_{n,\pi(n)}$ becomes minimal. In particular, we have to examine all cyclic permutations $\pi$ (each of which determines a tour); for these permutations, the sum in question equals the length of the associated tour. Therefore we can indeed relax TSP to AP.

Note that we ought to be a little more careful here, since we should not just use the given matrix $W$ to specify our AP: the diagonal entries $w_{ii} = 0$ would yield the identity as an optimal solution, which would result in a completely trivial lower bound: 0. As we are not interested in permutations with fixed points for the TSP anyway, we can avoid this problem by simply putting $w_{ii} = \infty$ for all $i$.[2] Clearly, this modification guarantees that an optimal solution of AP is a permutation without fixed points. If we should obtain a cyclic permutation as the optimal solution of AP, this permutation actually yields a solution of the TSP (by coincidence). Of course, in general, there is no reason why this should happen.

It is also comparatively easy to determine the weight $w(\mathrm{AP})$ of an optimal solution for the relaxed problem: the Hungarian algorithm of Section 14.2 will allow us to do so with complexity $O(n^3)$. Note that the Hungarian algorithm actually determines maximal weighted matchings, whereas we want to find a perfect matching of minimal weight for $K_{n,n}$ (with respect to the weights given by our modification of $W$). However, this merely requires a simple transformation, which was already discussed in Section 14.1.

It turns out that $w(\mathrm{AP})$ is usually a reasonably good approximation to $w(\mathrm{TSP})$ in practice – even though nobody has been able to prove this. Balas and Toth considered random instances for values of $n$ between 40 and 100 and got an average of 82% of $w(\mathrm{TSP})$ for $w(\mathrm{AP})$; see [LaLRS85, Chapter 10]. That the assignment relaxation has such good approximation properties is, perhaps, to be expected, since the cyclic permutations form quite a big part of all permutations without fixed points: the number of permutations without fixed points in $S_n$ is about $n!/e$, so that there is about one cyclic permutation among $n/e$ fixed point free permutations; see, for example, [Hal86].

Balas and Toth examined the assignment relaxation also for the ATSP, using 400 problems randomly chosen in the range $50 \leq n \leq 250$. Here $w(\mathrm{AP})$ was on average 99,2% of $w(\mathrm{ATSP})$.

**Example 15.2.1.** Consider the TSP of Example 15.1.2, where we replace the diagonal entries 0 in $W$ by 88 (the maximum of the $w_{ij}$) to obtain the matrix $W'$ for an associated AP. In order to reduce this AP to the determination of a maximal weighted matching, we consider the matrix $W'' = (88 - w'_{ij})$ instead of $W'$, as described in Section 14.1; note that $W''$ is the matrix given

[2] In practice, this is done by using a sufficiently large number $M$ instead of $\infty$: for instance, $M = \max\{w_{ij} : i, j = 1, \ldots, n\}$.