The Traveling Salesman Problem in Bounded Degree Graphs

ANDREAS BJÖRKLUND, Lund University
THORE HUSFELDT, IT University of Copenhagen and Lund University
PETTERI KASKI and MIKKO KOIVISTO, Helsinki Institute for Information Technology and University of Helsinki

We show that the traveling salesman problem in bounded-degree graphs can be solved in time $O((2-\epsilon)^n)$, where $\epsilon > 0$ depends only on the degree bound but not on the number of cities, n. The algorithm is a variant of the classical dynamic programming solution due to Bellman, and, independently, Held and Karp. In the case of bounded integer weights on the edges, we also give a polynomial-space algorithm with running time $O((2-\epsilon)^n)$ on bounded-degree graphs. In addition, we present an analogous analysis of Ryser's algorithm for the permanent of matrices with a bounded number of nonzero entries in each column.

Categories and Subject Descriptors: F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.1 [**Discrete Mathematics**]: Combinatorics; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Counting, dynamic programming, inclusion–exclusion, permanent, Shearer's entropy lemma, traveling salesman problem, trimming

ACM Reference Format:
Björklund, A., Husfeldt, T., Kaski, P., and Koivisto, M. 2012. The traveling salesman problem in bounded degree graphs. ACM Trans. Algor. 8, 2, Article 18 (April 2012), 13 pages.
DOI = 10.1145/2151171.2151181 http://doi.acm.org/10.1145/2151171.2151181

1. INTRODUCTION

There is no faster algorithm known for the traveling salesman problem than the classical dynamic programming solution from the early 1960s, discovered by Bellman [1960; 1962], and, independently, Held and Karp [1962]. It runs in time within a polynomial factor of 2^n, where n is the number of cities. Despite the half century of algorithmic development that has followed, it remains an open problem whether the traveling salesman problem can be solved in time $O(1.999^n)$ [Woeginger 2003].

A preliminary version of this article appeared in ICALP 2008, Part I, L. Aceto et al. Eds., Lecture Notes in Computer Science, vol 5125, Springer, 198–209.
This research was supported in part by the Academy of Finland, grants 117499 (P.K.) and 109101 (M.K.) and by the Swedish Research Council, project "Exact Algorithms" (A.B., T.H.).
Authors' addresses: A. Björklund, Lund University, Department of Computer Science, P.O. Box 118, SE-22100 Lund, Sweden; email: andreas.bjorklund@yahoo.se; T. Husfeldt, IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark; email: thore.husfeldt@cs.lu.se; P. Kaski and M. Koivisto, Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, P.O. Box 68, FI-00014 University of Helsinki, Finland; email: petteri.kaski@cs.helsinki.fi, mikko.koivisto@cs.helsinki.fi.

DOI 10.1145/2151171.2151181 http://doi.acm.org/10.1145/2151171.2151181

In this article we provide such an upper bound for graphs with bounded maximum vertex degree. For this restricted graph class, previous attemps have succeeded to prove such bounds when the degree bound, Δ, is three or four. Indeed, Eppstein [2007] presents a sophisticated branching algorithm that solves the problem in time $2^{n/3}n^{O(1)} = O(1.260^n)$ on cubic graphs ($\Delta = 3$) and in time $O(1.890^n)$ for $\Delta = 4$. Recently, Iwama and Nakashima [2007] improved the former bound to $O(1.251^n)$. These algorithms run in space polynomial in n. Very recently, Gebauer [2008] gave an exponential-space algorithm that runs in time $(\Delta-1)^{n/2}n^{O(1)}$ and can also list the Hamiltonian cycles, improving the time bound for $\Delta = 4$ to $O(1.733^n)$. However, for $\Delta > 4$, none of these techniques seems to improve upon $O(2^n)$.

We show that, perhaps somewhat surprisingly, with minor modifications the classical Bellman–Held–Karp algorithm can be made to run in time $O\big((2-\epsilon)^n\big)$, where $\epsilon > 0$ depends only on the degree bound.

THEOREM 1.1. *The traveling salesman problem for an n-vertex graph with maximum degree $\Delta = O(1)$ can be solved in time $\xi_\Delta^n n^{O(1)}$ with*

$$\xi_\Delta = (2^{(\Delta+1)} - 2\Delta - 2)^{1/(\Delta+1)}.$$

Our main contribution is indeed more analytical than algorithmic, and largely relies on exploiting variants of a beautiful lemma due to Shearer [Chung et al. 1986] (Shearer's entropy lemma) that in a combinatorial context enables us to derive upper bounds for the size of a set family based on the sizes of its projections. We used this lemma recently in connection with analyzing expedited versions of the FFT-like algorithm of Yates to solve covering problems for bounded-degree graphs via Moebius inversion [Björklund et al. 2008b]. In the present article we use the same analytical tools on classical algorithms for the traveling salesman problem.

In general, this approach seems to be new and quite versatile for bounding the running time of dynamic programming algorithms on restricted graph classes. To illustrate this, we show how the technique can be adapted to more involved settings by proving a stronger bound for regular triangle-free graphs.

THEOREM 1.2. *The traveling salesman problem for a triangle-free n-vertex graph where every vertex has degree $\Delta = O(1)$ can be solved in time $\eta_\Delta^n n^{O(1)}$ with*

$$\eta_\Delta = (2^{2\Delta} - (\Delta+1)2^{\Delta+1} + 2(\Delta^2+1))^{1/(2\Delta)}.$$

To motivate a further discussion, we observe that the algorithms in Theorems 1.1 and 1.2 both require exponential space, which immediately prompts the question whether there exists a polynomial-space algorithm with running time $O((2-\epsilon)^n)$ on bounded-degree graphs. This turns out to be the case if the edge weights are bounded integers.

Indeed, a classical polynomial-space algorithm due to Karp [1981] and, independently, Kohn et al. [1977], which actually counts suitably weighted Hamiltonian paths, can be made to run in time $O((2-\epsilon)^n)$ on bounded-degree graphs, again with only minor tailoring.

Somewhat perplexingly, we characterize the running time of the polynomial-space algorithm in terms of the *connected dominating sets* of the input graph. To properly state the result, we recall the definitions here. For a graph G and a set $W \subseteq V$ of vertices, the set W is a *connected set* if the induced subgraph $G[W]$ is connected, and is a *dominating set* if every vertex $v \in V$ is in W or adjacent to a vertex in W. Denote by $\mathcal{C}$ the family of connected sets of G, and by $\mathcal{D}$ the family of dominating sets of G.

THEOREM 1.3. *The traveling salesman problem for an n-vertex graph with bounded integer weights can be solved in time $|\mathcal{C} \cap \mathcal{D}|n^{O(1)}$ and in space $n^{O(1)}$. In particular, for*

Table I. Constants in Theorems 1.1, 1.2, and 1.3 for Small Values of Δ

Δ	3	4	5	6	7	8	$\cdots$
β_Δ	1.9680	1.9874	1.9948	1.9978	1.9991	1.9999	$\cdots$
γ_Δ	1.9343	1.9744	1.9894	1.9955	1.9980	1.9991	$\cdots$
ξ_Δ	1.6818	1.8557	1.9320	1.9672	1.9840	1.9921	$\cdots$
η_Δ	1.6475	1.8376	1.9231	1.9630	1.9820	1.9912	$\cdots$

maximum degree Δ it holds that $|\mathcal{C} \cap \mathcal{D}| \le \gamma_\Delta^n + n$, where

$$\gamma_\Delta = (2^{\Delta+1} - 2)^{1/(\Delta+1)}.$$

Table I displays the constants in Theorems 1.1, 1.2, and 1.3 for small values of Δ. We expect there to be room for improvement in each of the derived bounds. In particular, in this regard we would like to highlight the question of asymptotically tight upper bounds for $|\mathcal{C}|$, $|\mathcal{D}|$, and $|\mathcal{C} \cap \mathcal{D}|$ on bounded-degree graphs (cf., Lemma 2.4). Such bounds should be of independent combinatorial interest, and we fully expect better bounds to occur in the literature, even if we were unable to find.

Finally, we also consider a close relative of the traveling salesman problem: the problem of computing a matrix permanent; see below. (This extends our preliminary work [Björklund et al. 2008a] dedicated to the traveling salesman problem.)

1.1. Related Results on the Permanent of Sparse Matrices

The problem of computing the permanent of a given $n \times n$ matrix closely resembles that of counting (weighted) Hamiltonian paths in an n-vertex graph. Both are "permutation problems" which can be solved by dynamic programming across the subsets of an n-element ground set in time and space $2^n n^{O(1)}$. And, not surprisingly, the polynomial-space algorithms for counting Hamiltonian paths [Karp 1981; Kohn et al. 1977] follow Ryser's [1963] inclusion–exclusion algorithm for computing the matrix permanent. Because of this similarity, it is natural to ask whether the presented results for the traveling salesman problem and for counting Hamiltonian paths in sparse graphs transfer to computing the permanent of sparse matrices; bounding the maximum vertex degree of a graph corresponds to bounding the number of nonzero entries in the columns or rows of a matrix.

In this direction, the only previous work we are aware of is due to Servedio and Wan [2005]. They give an algorithm which, for any constant $C > 0$, computes the permanent of any $n \times n$ matrix with at most Cn nonzero entries in time $O((2 - \epsilon)^n)$, where $\epsilon > 0$ depends only on C. In fact, the analysis by Servedio and Wan [2005] yields a concrete running time bound $\psi_C^n n^{O(1)}$ with $\psi_C = 2(1 - 1/4^C)^{1/(8C)}$. The algorithm is in essence a derandomized version of an earlier algorithm by Bax and Franklin [2002], which in turn is a randomized variant of Ryser's original algorithm.

Here, we apply our analysis technique to Ryser's algorithm on $n \times n$ matrices with at most C nonzero entries in each column, a natural subclass of sparse matrices. (Equivalently, we could bound the number of nonzero entries in rows instead of columns, since the permanents of a matrix and its transpose are equal.)

THEOREM 1.4. *The permanent of any $n \times n$ matrix with at most C nonzero entries in each column can be computed in space $n^{O(1)}$ and in time $\phi_C^n n^{O(1)}$ with $\phi_C = (2^C - 1)^{1/C}$.*

Comparing the two bounds above, we observe that ϕ_C is strictly less than ψ_C for all $C > 1$. For example, for $C = 2, 3, 4$, we have, respectively, $\phi_C \approx 1.7321, 1.9130, 1.9680$, while $\psi_C \approx 1.9920, 1.9987, 1.9998$ (rounding upwards). This suggests that for the subclass of sparse matrices in question, our analysis technique is stronger than that of Servedio and Wan. Unfortunately, as we will discuss, our technique does not seem to extend to arbitrary sparse matrices.

1.2. Organization

We begin by establishing the combinatorial analysis tools in Section 2. This is followed by four sections devoted to the traveling salesman problem: In Section 3 we provide a precursor to Theorem 1.1 by using a simple argument that illustrates the main ideas of our approach, but leads to a weaker running time bound $\beta_\Delta^n n^{O(1)}$ with $\beta_\Delta = (2^{\Delta+1} - 1)^{1/(\Delta+1)}$. Theorems 1.1, 1.2, and 1.3 are then proved in Sections 4, 5, and 6, respectively. The analysis of a polynomial-space algorithm for counting Hamiltonian paths in Section 6 is finally followed by an analysis of Ryser's related algorithm for matrix permanent in Section 7, which proves Theorem 1.4.

1.3. Conventions

We consider the directed, asymmetric variant of the traveling salesman problem. A problem instance consists of an n-element ground set V and a *weight* $d(u, v) \in \{0, 1, \ldots\} \cup \{\infty\}$ for all distinct $u, v \in V$. A *tour* is a permutation $(v_1, v_2, \ldots, v_n)$ of V. The *weight* of a tour is $d(v_1, v_2) + d(v_2, v_3) + \cdots + d(v_{n-1}, v_n) + d(v_n, v_1)$. Given a problem instance, the task is to find the minimum weight of a tour. For further background on the traveling salesman problem, we refer to Applegate et al. [2006]; Gutin and Punnen [2002]; and Lawler et al. [1985].

We associate with each problem instance an *undirected* graph G with vertex set V and edge set E such that any two distinct $u, v \in V$ are joined by an edge $\{u, v\}$ if and only if $d(u, v) < \infty$ or $d(v, u) < \infty$. Unless explicitly indicated otherwise, all graph-theoretic terminology refers to the graph G. For standard graph-theoretic terminology, we refer to West [2001].

2. COMBINATORIAL PRELIMINARIES

We are interested in upper bounds for the sizes of certain set families associated with a graph with maximum degree Δ. Our starting point is an entropy lemma due to Shearer (see Chung et al. [1986]). However, our present proof, which we present for completeness and convenience of exposition, is, apparently, folklore; it was rediscovered by Llewellyn and Radhakrishnan, and is reproduced in Radhakrishnan [2001] (J. Radhakrishnan, personal discussion).

We require some preliminaries on entropy. Let X, Y, Z be random variables that take values in a finite set S. The *entropy* of X is

$$\mathrm{H}(X) = -\sum_{x \in S} \Pr(X = x) \log \Pr(X = x).$$

In particular, the entropy of the random variable obtained by conditioning X on the outcome $Y = y$ is

$$\mathrm{H}(X|Y = y) = -\sum_{x} \Pr(X = x|Y = y) \log \Pr(X = x|Y = y).$$

The *conditional entropy* of X given Y is

$$\mathrm{H}(X|Y) = \sum_{y \in S} \mathrm{H}(X|Y = y) \Pr(Y = y).$$

We recall the following three elementary facts about entropy. First, the entropy of the joint random variable (X, Y) decomposes, via the *Bayes rule*, as

$$\mathrm{H}(X, Y) = \mathrm{H}(X|Y) + \mathrm{H}(Y).$$

Second, conditional entropy is *monotone*, that is,

$$\mathrm{H}(X|Y, Z) \leq \mathrm{H}(X|Y).$$

Third,

$$0 \leq \mathrm{H}(X) \leq \log |S|,$$

with equality in the upper bound if and only if X is uniformly distributed on S.

Let V be a finite set, and associate a random variable X_v with each $v \in V$. Shearer's entropy lemma gives an upper bound on the entropy of the joint random variable $X = (X_v)_{v \in V}$ based on the entropies of projections of X. For a subset $A \subseteq V$, define the *projection* of X to A by $X_A = (X_v : v \in A)$. The key assumption in the lemma is that each random variable X_v occurs in sufficiently many projectors.

LEMMA 2.1 (SHEARER'S ENTROPY LEMMA [CHUNG ET AL. 1986]). *Let $A_1, A_2, \ldots, A_r$ be subsets of a finite set V such that every $v \in V$ occurs in at least δ of the sets $A_1, A_2, \ldots, A_r$. Then,*

$$\delta \cdot \mathrm{H}(X) \leq \sum_{i=1}^{r} \mathrm{H}(X_{A_i}).$$

PROOF (RADHAKRISHNAN 2001). Order the elements of V arbitrarily. For a logical proposition P, we use Iverson's bracket notation $[P]$ to denote a 1 if P is true and 0 if P is false. We have

$$\begin{aligned}
\sum_{i=1}^{r} \mathrm{H}(X_{A_i}) &= \sum_{i=1}^{r} \sum_{v \in A_i} \mathrm{H}(X_v | X_u : u \in A_i,\ u > v) && \text{(Bayes)} \\
&\geq \sum_{i=1}^{r} \sum_{v \in A_i} \mathrm{H}(X_v | X_u : u > v) && \text{(monotonicity)} \\
&= \sum_{v \in V} \sum_{i=1}^{r} [v \in A_i] \mathrm{H}(X_v | X_u : u > v) && \text{(summation order)} \\
&\geq \sum_{v \in V} \delta \cdot \mathrm{H}(X_v | X_u : u > v) && \text{(assumption)} \\
&= \delta \cdot \mathrm{H}(X). && \text{(Bayes)}
\end{aligned}$$

□

Let us now state and prove a combinatorial consequence of the entropy lemma.

LEMMA 2.2 (CHUNG, FRANKL, GRAHAM, AND SHEARER [1986]). *Let V be a finite set with subsets $A_1, A_2, \ldots, A_r$ such that every $v \in V$ is occurs in at least δ subsets. Let $\mathcal{F}$ be a family of subsets of V. For each $1 \leq i \leq r$, define the projections $\mathcal{F}_i = \{F \cap A_i : F \in \mathcal{F}\}$. Then,*

$$|\mathcal{F}|^{\delta} \leq \prod_{i=1}^{r} |\mathcal{F}_i|.$$

PROOF. Let X be a random variable that is uniformly distributed on $\mathcal{F}$. In particular, $\mathrm{H}(X) = \log |\mathcal{F}|$. We may view X as a joint random variable $X = (X_v : v \in V)$ over $\{0, 1\}$-valued components X_v, indicating whether $v \in V$ occurs in an outcome of X. Thus, we may view each projection X_{A_i} as a random variable taking values in $\mathcal{F}_i$. In particular, $\mathrm{H}(X_{A_i}) \leq \log |\mathcal{F}_i|$. By Lemma 2.1, we conclude that

$$\delta \log |\mathcal{F}| = \delta \cdot \mathrm{H}(X) \leq \sum_{i=1}^{r} \mathrm{H}(X_{A_i}) \leq \sum_{i=1}^{r} \log |\mathcal{F}_i|.$$

The claim follows by taking exponentials on both sides. □

(a) (b)

Fig. 1. (a) The region A_v of a vertex v in a graph with $\Delta = 5$; (b) impossible projection for a connected set $C \in \mathcal{C}$, $|C| \geq 2$; if only the black vertex v belongs to C, then C cannot be connected because all of v's neighbors belong to A_v.

We now tailor the previous lemma into a form that is more useful for our present purposes, thereby abstracting and somewhat generalizing an analysis we have presented earlier [Björklund et al. 2008b; Theorem 3.2]. In particular, we find it handy to leave out a constant number s of special subsets. In the next lemma, we apply the notational convention that any empty product evaluates to 1.

LEMMA 2.3. *Let V be a finite set with r elements and with subsets $A_1, A_2, \ldots, A_r$ such that every $v \in V$ occurs in exactly δ subsets. Let $\mathcal{F}$ be a family of subsets of V and assume that there is a log-concave function $f \geq 1$ and an $0 \leq s \leq r$ such that the projections $\mathcal{F}_i = \{F \cap A_i : F \in \mathcal{F}\}$ satisfy $|\mathcal{F}_i| \leq f(|A_i|)$ for each $s+1 \leq i \leq r$. Then,*

$$|\mathcal{F}| \leq f(\delta)^{r/\delta} \prod_{i=1}^{s} 2^{|A_i|/\delta}.$$

PROOF. Let $a_i = |A_i|$ and note that $a_1 + a_2 + \cdots + a_r = \delta r$. By Lemma 2.2, we have

$$|\mathcal{F}|^\delta \leq \prod_{i=1}^{s} 2^{a_i} \prod_{i=s+1}^{r} f(a_i) \leq \prod_{i=1}^{s} 2^{a_i} \prod_{i=1}^{r} f(a_i). \tag{1}$$

Since f is log-concave, Jensen's inequality gives

$$\frac{1}{r}\sum_{i=1}^{r} \log f(a_i) \leq \log f((a_1 + a_2 + \cdots + a_r)/r) = \log f(\delta).$$

Taking exponentials and combining with (1) gives

$$|\mathcal{F}|^\delta \leq f(\delta)^r \prod_{i=1}^{s} 2^{a_i},$$

which yields the claimed bound. □

For Theorem 1.1 it suffices to consider the special case where the A_i are defined in terms of neighborhoods of the vertices of G. For each $v \in V$, define the closed neighborhood $N(v)$ by

$$N(v) = \{v\} \cup \{u \in V : u \text{ and } v \text{ are adjacent in } G\}.$$

Begin by defining the subsets A_v for $v \in V$ as $A_v = N(v)$. Then, for each $u \in V$ with degree $d(u) < \Delta$, add u to $\Delta - d(u)$ of the sets A_v not already containing it (it does not matter which). This ensures that every vertex $u \in V$ is contained in exactly $\Delta + 1$ sets A_v. Figure 1(a) shows an example. For each $v \in V$, call the set A_v so obtained as the *region* of v.

LEMMA 2.4. *An n-vertex graph with maximum vertex degree Δ has at most $\beta_\Delta^n + n$ connected sets and at most $\gamma_\Delta^n + n$ connected dominating sets, where*

$$\beta_\Delta = (2^{\Delta+1} - 1)^{1/(\Delta+1)}, \quad \gamma_\Delta = (2^{\Delta+1} - 2)^{1/(\Delta+1)}.$$

PROOF. Recall that by $\mathcal{C}$ we denote the family of connected sets and by $\mathcal{D}$ the family of dominating sets. Let $\mathcal{C}' = \mathcal{C} \setminus \{\{v\} : v \in V\}$. Then, for every $C' \in \mathcal{C}'$ and every region A_v, $C' \cap A_v \neq \{v\}$; see Figure 1(b). Thus the number of sets in the projection $\mathcal{C}'_v = \{F \cap A_v : F \in \mathcal{C}\}$ is at most $2^{|A_v|} - 1$. To obtain the bound on connected sets, apply Lemma 2.3 with the log-concave function $f(a) = 2^a - 1$ and $s = 0$. To obtain the upper bound for $|\mathcal{C} \cap \mathcal{D}|$, observe that, in addition to the singleton projection excluded for a connected set, the empty projection is also excluded for each region in the case of a connected dominating set. □

3. CONNECTED SETS

This section establishes Theorem 1.1, but with a weaker bound; the purpose is to show a very straightforward argument for an $O((2-\epsilon)^n)$ upper bound.

Our starting point is the dynamic programming solution, which we proceed to recall. Select an arbitrary reference vertex $s \in V$. For $T \subseteq V$ and $v \in T$, denote by $D(T, v)$ the minimum weight of a directed path (in the complete directed graph with vertex set V and edge weights given by d) from s to v that consists of the vertices in T. The minimum weight of a tour is then solved by computing

$$\min_{v \in V} D(V, v) + d(v, s).$$

To construct $D(T, v)$ for all $s \in T \subseteq V$ and all $v \in T$, the algorithm starts with $D(\{s\}, s) = 0$, and evaluates the recurrence

$$D(T, v) = \min_{u \in T \setminus \{v\}} D(T \setminus \{v\}, u) + d(u, v). \tag{2}$$

The values $D(T, v)$ are stored in a table when they are computed to avoid redundant recomputation, an idea sometimes called *memoisation*. The space and time requirements are within a polynomial factor of 2^n, the number of subsets $T \subseteq V$.

Our idea to expedite this will restrict the family of subsets for which (2) is ever evaluated. To this end, consider any prefix $(v_1, v_2, \ldots, v_k)$ of a finite-weight tour with $v_1 = s$. The set of vertices $T = \{v_1, v_2, \ldots, v_k\}$ satisfies certain connectivity properties that we want to exploit. In this section, we merely use the trivial observation that T must be a connected set. Put otherwise, $D(T, v) = \infty$, unless T is a connected set. Thus, it suffices to evaluate (2) not over all subsets of V, but only over the family of connected sets $\mathcal{C}$. A bottom-up evaluation of (2) with memoisation gives an algorithm for solving the traveling salesman problem within time $|\mathcal{C}|$ up to polynomial factors. (Indeed, whether $T \in \mathcal{C}$ can be tested in polynomial time by, e.g., depth-first search; furthermore, for every $T \in \mathcal{C}$ with $|T| > 1$, there exists at least one $v \in T$ with $T \setminus \{v\} \in \mathcal{C}$—consider the leaves of a spanning tree of $G[T]$—which enables T to be discovered from $T \setminus \{v\}$.) With Lemma 2.4, this gives $O((2-\epsilon)^n)$ running time when G has maximum degree $O(1)$.

4. TRANSIENT SETS

This section establishes Theorem 1.1, which amounts to a more careful analysis of sets of vertices T occurring in prefixes of a tour with finite weight. The key insight is that a tour from the start vertex s to a vertex u cannot contain the entire neighborhood of any yet unvisited vertex v that is not a neighbor of s or u because then the tour could not be extended to v or it would get stuck at v without ever returning to s. In fact, we may slightly strengthen this observation.

In precise terms, we call a vertex set $T \subseteq V$ *transient with endpoint* $u \in T$ if it is connected, $s \in T$, and the following holds for every vertex $v \notin N(s) \cup N(u)$:

(1) if v belongs to T, then so do at least two of its adjacent vertices;
(2) if v does not belong to T, then neither do at least two of its adjacent vertices.

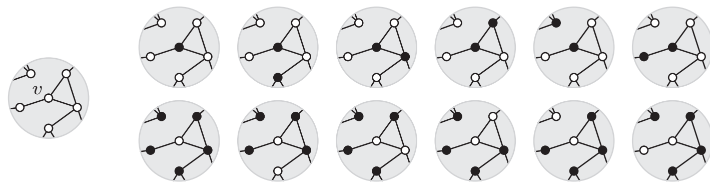

Fig. 2. A nonspecial region A_v (left) and the impossible intersections of A_v with a (black) transient set.

Note that testing if a vertex set is transient is a polynomial time task, we merely need to run a depth-first-search and checking each vertex neighborhood for the two properties above.

Let $\mathcal{T}_u$ denote the family of vertex sets that are transient with endpoint u.

Observe that any prefix $(v_1, v_2, \dots, v_k)$ of a finite-weight tour with $v_1 = s$ and $v_k = u$ has the property that $\{v_1, v_2, \dots, v_k\} \in \mathcal{T}_u$. It thus suffices to consider the recurrence

$$D(T, v) = \min_{\substack{u \in T \setminus \{v\} \\ T \setminus \{v\} \in \mathcal{T}_u}} D(T \setminus \{v\}, u) + d(u, v), \tag{3}$$

where we tacitly assume that the minimum of the empty set is ∞.

A top-down evaluation of (3) with memoisation leads to running time bounded by, up to polynomial factors,

$$\sum_{u \in V} |\mathcal{T}_u| \le n \max_{u \in V} |\mathcal{T}_u|. \tag{4}$$

To derive an upper bound for the size of $\mathcal{T}_u$, consider an arbitrary $u \in V$ and set $\delta = \Delta + 1$. Call a vertex $v \in V$ *special* if $N(v) \cap \big(N(s) \cup N(u)\big) \neq \emptyset$, and observe that there are at most $2(1 + \Delta^2) < 2\delta^2$ special vertices.

Now consider a nonspecial $v \in V$ and an arbitrary $T \in \mathcal{T}_u$. Let $a_v = |A_v|$. We can rule out the following projections $A_v \cap T$; see Figure 2 for an example.

(1) $v \in T$ and $|A_v \cap T| = 1$, so v has no neighbors in T. The tour never enters or leaves v. This can happen only if v is special.
(2) $v \in T$ but $|A_v \cap T| = 2$, so v has at most one neighbor in T. The tour never leaves v. This can happen only if v is special. There are at least $a_v - 1$ such cases (more if A_v contains vertices not connected to v).
(3) $v \notin T$ but $A_v \setminus \{v\} \subseteq T$, so all of v's neighbors are in T. When the tour arrives in v it cannot leave. This can happen only if v is special.
(4) $v \notin T$ but $|A_v \cap T| = a_v - 2$, so v has at most one neighbour also not in T. When the tour arrives in v it cannot leave. This can happen only if v is special. There are $a_v - 1$ such cases (more if A_v contains vertices not connected to v).

In total, we can rule out $2a_v$ of the 2^{a_v} potential projections. We now want to apply Lemma 2.3. To this end, we have to be slightly more careful as regards the arbitrary construction of the regions A_v (recall Section 2). In particular, whenever v is special, we want $|A_v| \le \delta$. For all large enough n and $\delta = O(1)$, this is easily arranged by not inserting additional vertices into a special A_v when $|A_v| = \delta$. Thus, we can apply Lemma 2.3 with $f(a) = 2^a - 2a$ and at most $2\delta^2$ special projectors A_v, each of size at most δ. We conclude that

$$|\mathcal{T}_u| \le \left(2^\delta - 2\delta\right)^{n/\delta} 2^{2\delta^2}. \tag{5}$$

Theorem 1.1 follows, with the asymptotic notation absorbing a factor n from (4) and a constant factor from (5).

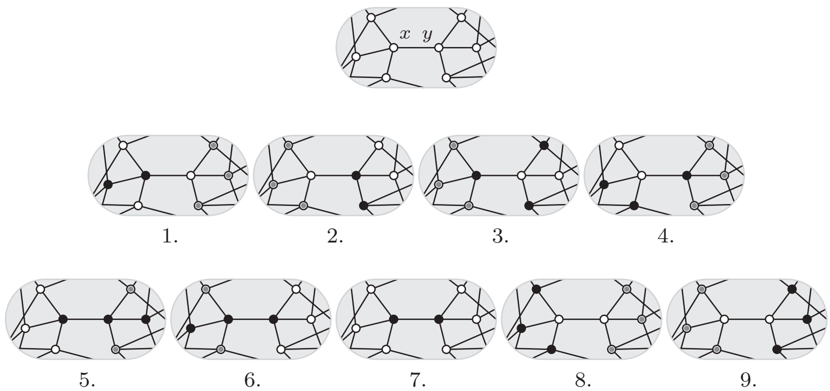

Fig. 3. Some impossible projections for regular triangle-free graphs. The vertex subset B_e is shown at the top. The black vertices are in T, the grey vertices can be in T or not. For the definitions of B_e and T, see the text.

5. TRIANGLE-FREE GRAPHS

To prove Theorem 1.2, we analyze the vertex sets of tour prefixes using a family of subsets B_e centered around every edge e. For each edge e in G, define B_e as the union of the closed neighborhoods of its endpoints,

$$B_e = N(x) \cup N(y), \quad e \text{ joins } x \text{ and } y.$$

The argument is somewhat more involved, but the bound becomes slightly better. We assume that G is regular with degree $\Delta = O(1)$ and contains no triangles; thus, each vertex $v \in V$ belongs to exactly $\delta = \Delta^2$ sets B_e.

Consider again the vertices $T = \{v_1, v_2, \ldots, v_k\}$ on a prefix of a finite-weight tour, $v_1 = s$, $v_k = u$. Suppose that e is an edge joining two vertices, x and y. Then, provided that e is again nonspecial, that is, sufficiently far from both s and u, we can again rule out certain projections of T to B_e.

(1) If both x and y belong to T, then either the tour travels along e, in which case x and y must each have another neighbor in T, or the edge e is not on the tour, in which case x and y must each have two other neighbors in T.
(2) If only one of the vertices, say x, belongs to T, then it must have two other neighbors in T. Moreover, the other vertex y cannot be completely surrounded by neighbors in T.

There are a number of symmetrical cases to these, all of which are checked in constant time around every edge. (Figure 3 is an example); a detailed enumeration of the cases appears as part of the analysis below.

We now turn to a detailed analysis of the projections $B_e \cap T$. To this end, partition B_e into $B_e = \{x\} \cup \{y\} \cup M(x) \cup M(y)$, where $M(x) = N(x) \setminus \{x, y\}$ and $M(y) = N(y) \setminus \{x, y\}$. We have $|M(x)| = |M(y)| = \Delta - 1$ because G is triangle-free. Call an edge e *special* if $B_e \cap \big(N(s) \cup N(u)\big) \neq \emptyset$. Because $\Delta = O(1)$, there are $O(1)$ special edges.

For a nonspecial e, we can rule out the following (nondisjoint) types of intersections $B_e \cap T$, exemplified in Figure 3.

(1) $x \in T$, $y \notin T$, $|M(x) \cap T| \le 1$. The tour would never leave x. There are $\Delta 2^{\Delta-1}$ such cases.
(2) Symmetrically, $y \in T$, $x \notin T$, $|M(y) \cap T| \le 1$. There are $\Delta 2^{\Delta-1}$ such cases.
(3) $x \in T$, $y \notin T$, $|M(y) \cap T| \ge \Delta - 2$. The tour would never reach and leave y. There are $\Delta 2^{\Delta-1}$ such cases.

(4) Symmetrically, $y \in T$, $x \notin T$, $|M(x) \cap T| \geq \Delta - 2$. There are $\Delta 2^{\Delta-1}$ such cases.
(5) $x \in T$, $y \in T$, $M(x) \cap T = \emptyset$, and $M(y) \cap T \neq \emptyset$. The tour never leaves x. There are $2^{\Delta-1} - 1$ such cases.
(6) Symmetrically, $x \in T$, $y \in T$, $M(y) \cap T = \emptyset$, and $M(x) \cap T \neq \emptyset$. There are $2^{\Delta-1} - 1$ such cases.
(7) $x \in T$, $y \in T$, $M(x) \cap T = M(y) \cap T = \emptyset$. The tour cannot leave $\{x, y\}$. There is 1 such case.
(8) $x \notin T$, $y \notin T$, $M(x) \subseteq T$. The tour cannot leave x. There are $2^{\Delta-1}$ such cases.
(9) Symmetrically, $x \notin T$, $y \notin T$, $M(y) \subseteq T$. There are $2^{\Delta-1}$ such cases.

In calculating the total number of forbidden intersections, observe that Types 1 and 3 are not disjoint (symmetrically, Types 2 and 4 are not disjoint). Both pairs of types have Δ^2 cases in common. Also, Types 8 and 9 are not disjoint; there is 1 case in common. Thus, in total we can rule out

$$4\Delta 2^{\Delta-1} + 2(2^{\Delta-1} - 1) + 1 + 2 \cdot 2^{\Delta-1} - 2\Delta^2 - 1 = (\Delta + 1)2^{\Delta+1} - 2(\Delta^2 + 1)$$

projections, so the number of projections is bounded by

$$2^{2\Delta} - (\Delta + 1)2^{\Delta+1} + 2(\Delta^2 + 1).$$

We can apply Lemma 2.3 with $\delta = \Delta^2$, $r = |E| = \Delta n/2$, the resulting bound is

$$(2^{2\Delta} - (\Delta + 1)2^{\Delta+1} + 2(\Delta^2 + 1))^{r/\delta} \cdot O(1),$$

which establishes Theorem 1.2 with (4) and (5).

6. POLYNOMIAL SPACE

For Theorem 1.3 our starting point is an algorithm of Karp [1981], and, independently, Kohn et al. [1977]. We assume that the weights $d(u, v)$ are bounded, that is, $d(u, v) \in \{0, 1, \ldots, B\} \cup \{\infty\}$, $B = O(1)$.

The algorithm is most conveniently described in terms of generating polynomials. Select an arbitrary reference vertex, $s \in V$, and let $U = V \setminus \{s\}$. For each $X \subseteq U$, denote by $q(X)$ the polynomial over the indeterminate z for which the coefficient of each monomial z^w counts the directed closed walks (in the complete directed graph with vertex set V and edge weights given by d) through s that (i) avoid the vertices in X; (ii) have length n; and (iii) have finite weight w.

We can compute $q(X)$ for a given $X \subseteq U$ in time polynomial in n by solving the following recurrence and setting $q(X) = p(n, s)$. Initialize the recurrence for each vertex $u \in V \setminus X$ with

$$p(0, u) = \begin{cases} 1 & \text{if } u = s; \\ 0 & \text{otherwise.} \end{cases}$$

For convenience, define $z^\infty = 0$. For each length $\ell = 1, 2, \ldots, n$ and each vertex $u \in V \setminus X$, let

$$p(\ell, u) = \sum_{v \in V \setminus X} p(\ell - 1, v) z^{d(v,u)}.$$

Note that, due to our assumption on bounded weights, each $p(\ell, u)$ has at most a polynomial number of monomials with nonzero coefficients.

By the principle of inclusion–exclusion, the monomials of the polynomial

$$Q = \sum_{X \subseteq U} (-1)^{|X|} q(X) \tag{6}$$

count, by weight, the number of directed closed walks through s that (i) visit each vertex in U at least once; and (ii) have length n. Put otherwise, what is counted by weight are the directed Hamilton cycles. It follows immediately that the traveling salesman problem can be solved in space polynomial in n and in time $2^n n^{O(1)}$. This completes the description of the algorithm.

Let us now analyze (6) in more detail, with the objective of obtaining an algorithm with better running time on bounded-degree graphs. It is convenient to work with a complemented form of (6), that is, for each $S \subseteq U$, let

$$r(S) = q(U \setminus S),$$

and rewrite (6) in the form of

$$Q = (-1)^n \sum_{S \subseteq U} (-1)^{|S|} r(S). \tag{7}$$

We want to reduce the number of $S \subseteq U$ that need to be considered in (7). To this end, observe that the induced subgraph $G[\{s\} \cup S]$ need not be connected. Associate with each $S \subseteq U$ the unique $f(S) \subseteq U$ such that $G[\{s\} \cup f(S)]$ is the connected component of $G[\{s\} \cup S]$ that contains s. Observe that $r(S) = r\big(f(S)\big)$ for all $S \subseteq U$. This observation enables the following partition of the subsets of U into f-preimages of constant r-value. For each $T \subseteq U$, let

$$f^{-1}(T) = \{S \subseteq U : f(S) = T\},$$

and rewrite (7) in the partitioned form of

$$Q = (-1)^n \sum_{T \subseteq U} r(T) \sum_{S \in f^{-1}(T)} (-1)^{|S|}. \tag{8}$$

The inner sum in (8) turns out to be determined by the connected dominating sets of G.

Lemma 6.1. *For every $T \subseteq U$ it holds that*

$$\sum_{S \in f^{-1}(T)} (-1)^{|S|} = \begin{cases} (-1)^{|T|} & \text{if } \{s\} \cup T \text{ is a connected dominating set of } G; \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Consider an arbitrary $T \subseteq U$. The preimage $f^{-1}(T)$ is clearly empty if $G[\{s\} \cup T]$ is not connected. Thus, in what follows, we can assume that $G[\{s\} \cup T]$ is connected. For a set $W \subseteq V$, denote by $\bar{N}(W)$ the set of vertices in W or adjacent to at least one vertex in W. Observe that $f(S) = T$ holds for an $S \subseteq U$ if and only if $S \supseteq T$ and $S \cap \bar{N}(\{s\} \cup T) = T$. In particular, if $V \setminus \bar{N}(\{s\} \cup T)$ is nonempty, then $f^{-1}(T)$ contains equally many even- and odd-sized subsets. Conversely, if $V \setminus \bar{N}(\{s\} \cup T)$ is empty (that is, $\{s\} \cup T$ is a dominating set of G), then $f^{-1}(T) = \{T\}$. □

Using Lemma 6.1 to simplify (8), we have

$$Q = (-1)^n \sum_{\substack{T \subseteq U \\ \{s\} \cup T \in \mathcal{C} \cap \mathcal{D}}} (-1)^{|T|} r(T). \tag{9}$$

To arrive at an algorithm with running time $|\mathcal{C} \cap \mathcal{D}| n^{O(1)}$ and space usage $n^{O(1)}$, it now suffices to list the elements of $\mathcal{C} \cap \mathcal{D}$ in space $n^{O(1)}$ and with delay bounded by $n^{O(1)}$.

The following listing strategy can be considered folklore, and is sketched here for interests of self-containment only. Observe that $\mathcal{C} \cap \mathcal{D}$ is an up-closed family of subsets of V; that is, if a set is in the family, then so are all of its supersets. Furthermore, whether a given $W \subseteq V$ is in $\mathcal{C} \cap \mathcal{D}$ can be decided in time $n^{O(1)}$. These observations enable the following top-down, depth-first listing algorithm for the sets in $\mathcal{C} \cap \mathcal{D}$. Initially, we visit the set V if and only if G is connected; otherwise $\mathcal{C} \cap \mathcal{D}$ is empty. Whenever we visit a set $Y \subseteq V$, we first list it, and then consider each of its maximal proper subsets $Y \setminus \{y\}$, $y \in Y$, in turn. We recursively visit $Y \setminus \{y\}$ if both (i) $Y \setminus \{y\} \in \mathcal{C} \cap \mathcal{D}$; and (ii) Y is the maximum (say, w.r.t. lexicographic order of subsets of V) minimal proper superset of $Y \setminus \{y\}$ in $\mathcal{C} \cap \mathcal{D}$.

Theorem 1.3 now follows from Lemma 2.4.

7. RYSER'S ALGORITHM FOR THE PERMANENT OF SPARSE MATRICES

Let A be an $n \times n$ matrix with entry $a_{ij} \in R$ in the ith row and jth column; here R is any algebraic ring (e.g., the real numbers), with neutral element 0 (zero) with respect to addition. The permanent of A is defined as

$$\mathrm{per}A = \sum_{\sigma} a_{1\sigma(1)} a_{2\sigma(2)} \cdots a_{n\sigma(n)},$$

where the sum is over all permutations σ of $[n] = \{1, 2, \ldots, n\}$.

Ryser's [1963] algorithm for computing per$A$ is based on the inclusion–exclusion formula

$$\mathrm{per}A = \sum_{S \subseteq [n]} (-1)^{n-|S|} \prod_{i=1}^{n} \sum_{j \in S} a_{ij}, \tag{10}$$

which can clearly be evaluated using $2^n n^{O(1)}$ additions and multiplications.

We analyze the number of additions and multiplications needed for evaluating the inclusion–exclusion expression under the assumption that every column of A contains at most C nonzero elements for some constant $C \leq n$. To this end, let $\mathcal{F}$ be the family of subsets S of $[n]$ such that for every $i = 1, 2, \ldots, n$ there is at least one $j \in S$ with $a_{ij} \neq 0$. Clearly, only such sets S can contribute a nonzero term to the sum (10). Furthermore, the family $\mathcal{F}$ is upwards-closed with respect to inclusion: It is possible to traverse through the members of $\mathcal{F}$ in time proportional to $|\mathcal{F}|n^{O(1)}$ using space polynomial in n. So it remains to bound the size $|\mathcal{F}|$.

To apply Lemma 2.3, we begin by defining the subsets A_i for $i = 1, 2, \ldots, n$ as $A_i = \{j \in [n] : a_{ij} \neq 0\}$. For $j \in [n]$, let $d(j)$ denote the number of sets A_i containing j. Then, for each $j \in [n]$ with $d(j) < C$, add j to $C - d(j)$ of the sets A_i not already containing it (it does not matter which). This ensures that every $j \in [n]$ is contained in exactly C sets A_i. Next, we consider the projections $\mathcal{F}_i = \{S \cap A_i : S \in \mathcal{F}\}$. Since every $S \in \mathcal{F}$ contains some $j \in S$ with $a_{ij} \neq 0$, the intersection $S \cap A_i$ cannot be empty. Thus $|\mathcal{F}_i| \leq 2^{|A_i|} - 1$, a log-concave function of $|A_i|$. Now, by Lemma 2.3, we obtain

$$|\mathcal{F}| \leq (2^C - 1)^{n/C}.$$

Thus, we have proved Theorem 1.4.

Here the assumption that every column (or every row) contains at most C nonzero entries is crucial. Indeed, it is easy to construct a matrix with only $2n-1$ nonzero entries such that 2^{n-1} out of the 2^n terms in the inclusion–exclusion sum (10) are nonzero: let $a_{ij} = 1$ if $i = 1$ or $j = 1$, else let $a_{ij} = 0$.

REFERENCES

APPLEGATE, D. L., BIXBY, R. E., CHVATAL, V., AND COOK, W. J. 2006. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, NJ.

BAX, E. AND FRANKLIN, J. 2002. A permanent algorithm with exp[(n1/3/(2 ln n))] expected speedup for 0–1 matrices. *Algorithmica 32*, 1, 157–62.

BELLMAN, R. 1960. Combinatorial processes and dynamic programming. In *Combinatorial Analysis*, R. Bellman and M. Hall, Jr. Eds., American Mathematical Society, Providence, RI, 217–249.

BELLMAN, R. 1962. Dynamic programming treatment of the travelling salesman problem. *J. ACM 9*, 61–63.

BJÖRKLUND, A., HUSFELDT, T., KASKI, P., AND KOIVISTO, M. 2008a. The travelling salesman problem in bounded degree graphs. In *Automata, Languages and Programming,* Lecture Notes in Computer Science, vol. 5125, Springer, Berlin, 198–209.

BJÖRKLUND, A., HUSFELDT, T., KASKI, P., AND KOIVISTO, M. 2008b. Trimmed Moebius inversion and graphs of bounded degree. In *Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science (STAC)*, S. Albers and P. Weil Eds., 85–96.

CHUNG, F. R. K., FRANKL, P., GRAHAM, R. L., AND SHEARER, J. B. 1986. Some intersection theorems for ordered sets and graphs. *J. Combinatorial Theory Ser. A,* 43, 23–37.

EPPSTEIN, D. 2007. The traveling salesman problem for cubic graphs. *J. Graph Alg. Appl. 11*, 61–81.

GEBAUER, H. 2008. On the number of Hamilton cycles in bounded degree graphs. In *Proceedings of the 4th Workshop on Analytic Algorithmics and Combinatorics*. SIAM, Philadelphia, PA.

GUTIN, G. AND PUNNEN, A. P. 2002. *The Traveling Salesman Problem and its Variations.* Kluwer, Amsterdam.

HELD, M. AND KARP, R. M. 1962. A dynamic programming approach to sequencing problems. *J. Soc. Ind. Appl. Math. 10*, 196–210.

IWAMA, K. AND NAKASHIMA, T. 2007. An improved exact algorithm for cubic graph TSP. In *Proceedings of the Computing and Combinatorics, 13th Annual International Conference (COCOON).* G. Lin Ed., Lecture Notes in Computer Science, vol. 459, Springer, Berlin, 108–117.

KARP, R. M. 1981. Dynamic programming meets the principle of inclusion and exclusion. *Oper. Res. Lett. 1*, 2, 49–51.

KOHN, S., GOTTLIEB, A., AND KOHN, M. 1977. A generating function approach to the traveling salesman problem. In *Proceedings of the ACM Annual Conference*. ACM, New York, 294–300.

LAWLER, E. L., LENSTRA, J. K., KAN, A. H. G. R., AND SHMOYS, D. B. 1985. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley.

RADHAKRISHNAN, J. 2001. Entropy and counting. In *Computational Mathematics, Modelling and Algorithms*, J. C. Mishra Ed., Narosa Publishers.

RYSER, H. J. 1963. *Combinatorial Mathematics*. The Mathematical Association of America, Washington, D.C.

SERVEDIO, R. A. AND WAN, A. 2005. Computing sparse permanents faster. *Inf. Proc. Lett. 96*, 89–92.

WEST, D. E. 2001. *Introduction to Graph Theory* 2nd Ed. Prentice–Hall, Englewood Cliffs, NJ.

WOEGINGER, G. J. 2003. Exact algorithms for NP-hard problems: A survey. In *Combinatorial Optimization—Eureka, You Shrink*! Lecture Notes in Computer Science, vol. 2570, Springer, Berlin, 185–207.

Received February 2009; revised March 2010; accepted March 2010