The Travelling Salesman Problem in Bounded Degree Graphs*

Andreas Björklund[1], Thore Husfeldt[1], Petteri Kaski[2], and Mikko Koivisto[2]

[1] Lund University, Department of Computer Science, P.O.Box 118, SE-22100 Lund, Sweden, e-mail: andreas.bjorklund@logipard.com, thore.husfeldt@cs.lu.se
[2] Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, P.O.Box 68, FI-00014 University of Helsinki, Finland, e-mail: firstname.lastname@cs.helsinki.fi,

Abstract. We show that the travelling 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 present a polynomial-space algorithm with running time $O((2-\epsilon)^n)$ on bounded-degree graphs.

1 Introduction

There is no faster algorithm known for the travelling salesman problem than the classical dynamic programming solution from the early 1960s, discovered by Bellman [2, 3], and, independently, Held and Karp [9]. It runs in time within a polynomial factor of 2^n, where n is the number of cities. Despite the half a century of algorithmic development that has followed, it remains an open problem whether the travelling salesman problem can be solved in time $O(1.999^n)$ [15].

In this paper 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 [6] 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 [10] improved the former bound to $O(1.251^n)$. These algorithms run in space polynomial in n. Very recently, Gebauer [7] 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 $O((2-\epsilon)^n)$, where $\epsilon > 0$ depends only on the degree bound:

* This research was supported in part by the Academy of Finland, Grants 117499 (P.K.) and 109101 (M.K.).

Theorem 1 *The travelling 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 [5] ("Shearer's Entropy Lemma") that in a combinatorial context enables one 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 analysing expedited versions of the FFT-like algorithm of Yates to solve covering problems for bounded-degree graphs via Moebius inversion [4], realising only later that classical algorithms for the travelling salesman problem yield to the same analytical tools. 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 prove a stronger bound for regular triangle-free graphs:

Theorem 2 *The travelling 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 yet further illustration, we observe that the algorithms in Theorems 1 and 2 both require exponential space, which immediately prompts the question whether there exists a polynomial-space algorithm with running time $(2-\epsilon)^n$ 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 [11] and, independently, Kohn, Gottlieb, and Kohn [12], can be made to run in time $(2-\epsilon)^n$ on bounded-degree graphs, again with only minor tailoring.

Somewhat perplexingly, we characterise 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, 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 3 *The travelling 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 maximum degree Δ it holds that $|\mathcal{C} \cap \mathcal{D}| \le \gamma_\Delta^n + n$, where*

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

Remark. Table 1 displays the constants in Theorems 1, 2, and 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

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	$\cdots$

Fig. 1. The constants in Theorems 1, 2, and 3 for small values of Δ.

asymptotically tight upper bounds for $|\mathcal{C}|$, $|\mathcal{D}|$, and $|\mathcal{C} \cap D|$ on bounded-degree graphs (cf. Lemma 6). 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 these.

Organisation. The combinatorial analysis tools are established in Sect. 2. We establish a precursor to Theorem 1 in Sect. 3 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, 2, and 3 are proved in Sect. 4, 5, and 6, respectively.

1.1 Conventions

We consider the directed, asymmetric variant of the travelling 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 travelling salesman problem, we refer to [1, 8, 13].

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 [14].

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 the following lemma due to Shearer (see [5]).

Lemma 4 (Chung, Frankl, Graham, and Shearer [5]) *Let V be a finite set with subsets $A_1, A_2, \ldots, A_r$ such that every $v \in V$ is contained in at least δ subsets. Let $\mathcal{F}$ be a family of subsets of V. For each $1 \le i \le r$ define the projections $\mathcal{F}_i = \{F \cap A_i : F \in \mathcal{F}\}$. Then,*

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

First, we bring the lemma into a format that is more useful for our present purposes. For instance, we will find it handy to leave out a constant number s of special subsets. The following lemma abstracts and to a certain extent generalises an analysis we have presented earlier [4, Theorem 3.2].

Lemma 5 *Let V be a finite set with r elements and with subsets $A_1, A_2, \ldots, A_r$ such that every $v \in V$ is contained 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 4, we have

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

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.

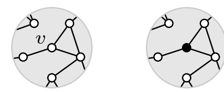
For Theorem 1 it suffices to consider the special case where the A_i are defined in terms of neighbourhoods of the vertices of G. For each $v \in V$, define the closed neighbourhood $N(v)$ by

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

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 2(a) shows an example. For each $v \in V$, call the set A_v so obtained the *region* of v.

Lemma 6 *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*

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



(a) (b)

Fig. 2. (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 belongs to C then C cannot be connected, because all of v's neighbours belong to A_v.

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 Fig. 2(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 5 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, also the empty projection is excluded for each region in the case of a connected dominating set.

3 Connected sets

This section establishes Theorem 1, but with a weaker bound; the purpose is to show a very straightforward argument for an $O\big((2-\epsilon)^n\big)$ 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

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

The values $D(T, v)$ are stored 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 the present section, we use merely 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 travelling 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 6 this gives $O\big((2-\epsilon)^n\big)$ running time when G has maximum degree $O(1)$.

4 Transient sets

This section establishes Theorem 1, which amounts to a more careful analysis of sets of vertices T occurring in prefixes of a tour with finite weight. For example, such a set T cannot contain all vertices adjacent to a vertex $v \notin T \cup N(s)$, because then the tour necessarily either avoids v or gets stuck at v without returning to s.

In precise terms, a vertex set $T \subseteq V$ is *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.

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 neighbourhood for the two properties above.

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

Observe that any prefix $(v_1, v_2, \ldots, v_k)$ of a finite-weight tour with $v_1 = s$ and $v_k = u$ has the property that $\{v_1, v_2, \ldots, v_k\} \in \mathcal{T}_u$. 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 an 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 $N(v) \cap (N(s) \cup N(u)) \neq \emptyset$, and observe that there are at most $2(1 + \Delta^2) < 2\delta^2$ special vertices.

Now consider a non-special $v \in V$ 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 Fig. 3 for an example.

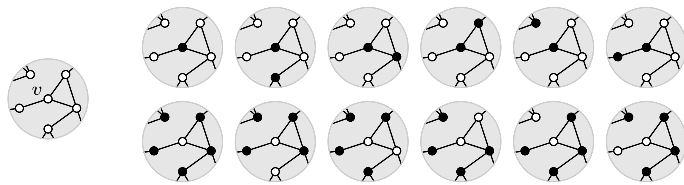


Fig. 3. A non-special region A_v (left) and the impossible intersections of A_v with a (black) transient set.

1. $v \in T$ and $|A_v \cap T| = 1$, so v has no neighbours 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 neighbour 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 neighbours 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 5. To this end, we have to be slightly more careful as regards the arbitrary construction of the regions A_v (recall Sect. 2). In particular, whenever v is special, we want $|A_v| \leq \delta$. 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 5 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

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

Theorem 1 follows, with the asymptotic notation absorbing a factor n from (4) and a constant factor from (5).

5 Triangle-free graphs

We now analyse the vertex sets of tour prefixes using a family of subsets B_e centered around every edge. 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.

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 non-special, that is, sufficiently far from both s and u, we can again rule out certain projections of T to B_e:

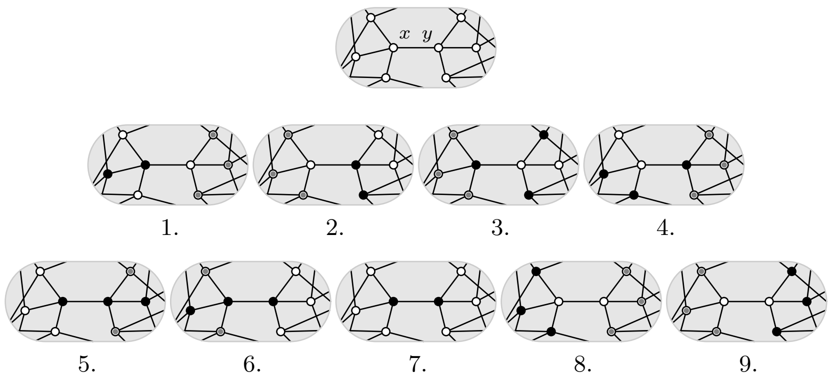


Fig. 4. Some impossible projections for regular triangle-free graphs. B_e is the vertex subset at the top. The black vertices are in T, the grey vertices can be in T or not.

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

There are a number of symmetrical cases to these, all of which are checked in constant time around every edge. See Fig. 4 for an example; a detailed enumeration of the cases appears as part of the analysis below.

For each edge e in G, define B_e as the union of the closed neighbourhoods of its endpoints,

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

Because G is triangle-free and Δ-regular, each vertex $v \in V$ belongs to exactly $\delta = \Delta^2$ sets B_e.

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 $B_e \cap (N(s) \cup N(u)) \neq \emptyset$. Because $\Delta = O(1)$, there are $O(1)$ special edges.

For a non-special e, we can rule out the following (non-disjoint) types of intersections $B_e \cap T$, exemplified in Fig. 4.

1. $x \in T$, $y \notin T$, $|M(x) \cap T| \leq 1$. The tour would never leave x. There are $\Delta 2^{\Delta - 1}$ such cases.
2. Symmetrically, $y \in T$, $x \notin T$, $|M(y) \cap T| \leq 1$. There are $\Delta 2^{\Delta - 1}$ such cases.
3. $x \in T$, $y \notin T$, $|M(y) \cap T| \geq \Delta - 2$. 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 5 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 2 with (4) and (5).

6 Polynomial space

Our starting point is an algorithm of Karp [11], and, independently, Kohn, Gottlieb, and Kohn [12]. 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 $q(X) = p(n, s)$. Initialise 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 travelling 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 analyse (6) in more detail, with the objective of obtaining an algorithm with better running time on bounded-degree graphs. It will be 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

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

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 $r(S) = r(f(S))$ 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

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

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

Lemma 7 *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 7 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 here sketched 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 3 now follows from Lemma 6.

References

1. Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W. J.: The Traveling Salesman Problem: A Computational Study, Princeton University Press, 2006.
2. Bellman, R.: Combinatorial processes and dynamic programming, Combinatorial Analysis (Bellman, R., Hall, M., Jr., Eds.), Proceedings of Symposia in Applied Mathematics 10, American Mathematical Society, 1960, pp. 217–249.
3. Bellman, R.: Dynamic programming treatment of the travelling salesman problem, J. Assoc. Comput. Mach. 9 (1962), 61–63.
4. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Trimmed Moebius inversion and graphs of bounded degree, Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science STACS 2008, Bordeaux, February 21–23, 2008, to appear.
5. Chung, F. R. K., Frankl, P., Graham, R. L., Shearer, J. B.: Some intersection theorems for ordered sets and graphs, J. Combinatorial Theory Ser. A 43 (1986), 23–37.

6. Eppstein, D.: The traveling salesman problem for cubic graphs. J. Graph Algorithms Appl. 11 (2007), 61–81.
7. Gebauer, H.: On the number of Hamilton cycles in bounded degree graphs, Proceedings of the Fourth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2008, San Francisco, California, USA, January 19, SIAM, 2008, to appear.
8. Gutin, G., Punnen, A. P., Eds.: The Traveling Salesman Problem and its Variations, Kluwer, 2002.
9. Held, M., Karp, R. M.: A dynamic programming approach to sequencing problems, J. Soc. Indust. Appl. Math. 10 (1962), 196–210.
10. Iwama, K., Nakashima, T.: An improved exact algorithm for cubic graph TSP, Computing and Combinatorics, 13th Annual International Conference, COCOON 2007, Banff, Canada, July 16–19, 2007, Lecture Notes in Computer Science 4598, Springer, 2007, pp. 108–117.
11. Karp, R. M.: Dynamic programming meets the principle of inclusion and exclusion, Oper. Res. Lett. 1 (1982) no. 2, 49–51.
12. Kohn, S., Gottlieb, A., Kohn, M.: A generating function approach to the traveling salesman problem, Proceedings of the 1977 Annual Conference (ACM'77), Association for Computing Machinery, 1977, pp. 294–300.
13. Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B., Eds.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Wiley, 1985.
14. West, D. B.: Introduction to Graph Theory, 2nd ed., Prentice–Hall, 2001.
15. Woeginger, G. J.: Exact algorithms for NP-hard problems: a survey, Combinatorial Optimization – Eureka, You Shrink! (Jünger, M., Reinelt, G., Rinaldi, G., Eds.), Lecture Notes in Computer Science 2570, Springer, 2003, pp. 185–207.