Faster exponential-time algorithms in graphs of bounded average degree[☆]

Marek Cygan[a], Marcin Pilipczuk[a]

[a]*Institute of Informatics, University of Warsaw, Poland*

Abstract

We present a number of exponential-time algorithms for problems in sparse matrices and graphs of bounded average degree. First, we obtain a simple algorithm that computes a permanent of an $n \times n$ matrix over an arbitrary commutative ring with at most dn non-zero entries using $O^\star(2^{(1-1/(3.55d))n})$ time and ring operations[1], improving and simplifying the recent result of Izumi and Wadayama [FOCS 2012].

Second, we present a simple algorithm for counting perfect matchings in an n-vertex graph in $O^\star(2^{n/2})$ time and polynomial space; our algorithm matches the complexity bounds of the algorithm of Björklund [SODA 2012], but relies on inclusion-exclusion principle instead of algebraic transformations.

Third, we show a combinatorial lemma that bounds the number of "Hamiltonian-like" structures in a graph of bounded average degree. Using this result, we show that

1. a minimum weight Hamiltonian cycle in an n-vertex graph with average degree bounded by d can be found in $O^\star(2^{(1-\varepsilon_d)n})$ time and exponential space for a constant ε_d depending only on d;
2. the number of perfect matchings in an n-vertex graph with average degree bounded by d can be computed in $O^\star(2^{(1-\varepsilon'_d)n/2})$ time and exponential space, for a constant ε'_d depending only on d.

The algorithm for minimum weight Hamiltonian cycle generalizes the recent results of Björklund et al. [TALG 2012] on graphs of bounded degree.

Keywords: moderately-exponential algorithms, bounded average degree, counting perfect matchings, minimum weight Hamiltonian cycle

1. Introduction

Improving upon the 50-years old $O^\star(2^n)$-time dynamic programming algorithms for the Traveling Salesman Problem by Bellman [1] and Held and Karp [2] is a major open problem in the field of exponential-time algorithms [3]. A similar situation appears when we want to count perfect matchings in a graph: a half-century old $O^\star(2^{n/2})$-time algorithm of Ryser for bipartite graphs [4] has only recently been transferred to arbitrary graphs [5], and breaking these time complexity barriers seems like a very challenging task.

From a broader perspective, improving upon a trivial brute-force or a simple dynamic programming algorithm is one of the main goals of the field of exponential-time algorithms. The last few years brought a number of positive results in that direction, most notably the $O^\star(1.66^n)$ randomized algorithm for finding a Hamiltonian cycle in an undirected graph [6]. However, it is conjectured (the so-called Strong Exponential Time Hypothesis [7]) that the central problem of satisfying a general CNF-SAT formulae does not admit any exponentially better algorithm than the trivial brute-force one. A number of lower bounds were proven using this assumption [8, 9, 10].

Although the aforementioned 2^n or $2^{n/2}$-barriers may be difficult or even outright impossible to break, it seems reasonable to suspect that additional assumptions on the graph structure, such as bounded degree or bounded average

[1] The $O^\star$-notation suppresses factors polynomial in the input size.

[☆] A preliminary version of this work has been presented at ICALP 2013.

Email addresses: cygan@mimuw.edu.pl (Marek Cygan), malcin@mimuw.edu.pl (Marcin Pilipczuk)

degree, simplify the computational tasks significantly. In the case of the problem of counting perfect matchings in bipartite graphs, the classic algorithm of Ryser [4] the best known improvement in general graphs is an algorithm running in expected time $O^\star(2^{(1-O(n^{2/3}\log n))\cdot n/2})$ due to Bax and Franklin [11]. If one assumes bounded average degree, faster algorithms have been given by Servedio and Wan [12] and, very recently, by Izumi and Wadayama [13]. In Section 2 we continue this line of research and show the following.

Theorem 1. *For any commutative semiring R, given an $m \times n$ matrix M, $m \leq n$ with elements from R with at most dm non-zero entries for some $d \geq 2$, one can compute the permanent of M using $O^\star(2^{(1-1/(3.55d))m})$ time and performing $O^\star(2^{(1-1/(3.55d))m})$ operations over the semiring R. The algorithm may require to use exponential space and store an exponential number of elements from R.*

Note that the number of perfect matchings in a bipartite graph is equal to the permanent of the adjacency matrix of this graph (computed over $\mathbb{Z}$). Hence, we improve the running time of [13, 12] in terms of the dependency on d. We would like to emphasise that our proof of Theorem 1 is elementary and does not need the advanced techniques of coding theory used in [13].

Since the algorithm of Theorem 1 is able to handle the computation of the permanent of any matrix over commutative semiring, not only the special case of computing the number of perfect matchings, our result shows also that the running time of the algorithm of Björklund et al. [14] can be improved for sparse matrices.

In Section 3, we move to the problem of counting perfect matchings in general graphs. An algorithm solving this problem in $O^\star(2^{n/2})$ time, that is, in time matching the bound for bipartite graphs, has been discovered very recently, in 2012, by Björklund [5]. Björklund's result improved upon previous algorithms with running time $O^\star(1.732^n)$ due to Björklund and Husfeldt [15] and with running time $O^\star(1.619^n)$ due to Koivisto [16]. We remark that, in contrast, the corresponding algorithm of Ryser [4] for bipartite graphs is already 50-years old. In Section 3, we observe that the problem of counting perfect matchings in general graphs can be reduced to a problem of counting some special types of cycle covers, which, in turn, can be done in $O^\star(2^{n/2})$-time and polynomial space for an n-vertex graph, using the inclusion-exclusion principle. Thus, we obtain a new proof of the main result of [5], using the inclusion-exclusion principle instead of advanced algebraic transformations.

The problem of counting some special types of cycle covers, introduced in Section 3, moves us to the area of Hamiltonian-like problems. In 2008 Björklund et al. [17] observed that the classic dynamic programming algorithm for finding minimum weight Hamiltonian cycle can be trimmed to running time $O^\star(2^{(1-\varepsilon_\Delta)n})$ in graphs of maximum degree Δ. The cost of this improvement is the use of exponential space, as we can no longer easily translate the dynamic programming algorithm into an inclusion-exclusion formula. The ideas from [17] were also applied to the Fast Subset Convolution algorithm [18], yielding a similar improvements for the problem of computing the chromatic number in graphs of bounded degree [19].

In Section 4 we show a combinatorial lemma that bounds the number of "Hamiltonian-like" structures in a graph of bounded average degree. Using this result, in Section 5 we show the following.

Theorem 2. *For every $d \geq 1$ there exists a constant $\varepsilon_d > 0$ such that, given an n-vertex graph G of average degree bounded by d, in $O^\star(2^{(1-\varepsilon_d)n})$ time and exponential space one can find in G a minimum weight Hamiltonian cycle.*

Theorem 3. *For every $d \geq 1$ there exists a constant $\varepsilon'_d > 0$ such that, given an n-vertex graph G of average degree bounded by d, in $O^\star(2^{(1-\varepsilon'_d)n/2})$ time and exponential space one can count the number of perfect matchings in G.*

Theorem 2 generalizes the results of [17] to the graphs of bounded average degree. To the best of our knowledge, Theorem 3 is the first result that breaks the $2^{n/2}$-barrier for counting perfect matchings in not necessarily bipartite graphs of bounded (average) degree. We note that in Theorems 2 and 3 the constants ε_d and ε'_d depend on d in a doubly-exponential manner, which is worse than the single-exponential behaviour of [17] in graphs of bounded degree.

Let us now shortly elaborate on the techniques used to prove the above theorems. Following the same general approach as the results of [17] for graphs of bounded degree, we want to limit the number of states of the classic dynamic programming algorithm for minimum weight Hamiltonian cycle or of the algorithm developed in Section 3 for counting perfect matchings (written as a dynamic programming algorithm, instead of one based on the inclusion-exclusion principle). However, in order to deal with graphs of bounded average degree, we need to introduce new concepts and tools. Recall that, by a standard averaging argument, if the average degree of an n-vertex graph G is

bounded by d, for any $D \geq d$ there are at most dn/D vertices of degree at least D. However, it turns out that this bound cannot be tight for a large number of values of D at the same time. This simple observation lies at the heart of our combinatorial lemma, intuitively allowing us to afford a more expensive branching on vertices of degree more than D provided that there are significantly less than dn/D of them.

Notation. We use standard (multi)graph notation. For a graph $G = (V, E)$ and a vertex $v \in V$ the *neighbourhood* of v is defined as $N_G(v) = \{u : uv \in E\} \setminus \{v\}$ and the *closed neighbourhood* of v as $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of $v \in V$ is denoted $\deg_G(v)$ and equals the number of end-points of edges incident to v. In particular a self-loop contributes 2 to the degree of a vertex. We omit the subscript if the graph G is clear from the context. The *average degree* of an n-vertex graph $G = (V, E)$ is defined as $\frac{1}{n}\sum_{v \in V} \deg(v) = 2|E|/n$. A *cycle cover* in a multigraph $G = (V, E)$ is a subset of edges $C \subseteq E$, where each vertex is of degree exactly two if G is undirected or each vertex has exactly one outgoing and one ingoing arc, if G is directed. Note that this definition allows a cycle cover to contain cycles of length 1, i.e. self-loops, as well as taking two different parallel edges as a length 2 cycle (but does not allow using the same edge twice). A *walk* in a graph $G = (V, E)$ is sequence $v_0, e_1, v_1, e_2, v_2, e_3 \ldots, e_s, v_s$ where $v_i \in V$ for $0 \leq i \leq s$ and e_i is an edge of G that connects v_{i-1} and v_i, for $1 \leq i \leq s$. A walk is *closed* if $v_0 = v_s$. The *length* of a walk equals s, i.e., the number of edges in the sequence. We emphasize here that our definition of a closed walk distinguishes not only the direction of a walk, but also the starting point.

For a graph $G = (V, E)$ by $V_{\deg=c}, V_{\deg>c}, V_{\deg\geq c}$ let us denote the subsets of vertices of degree equal to c, greater than c and at least c respectively.

For a nonnegative integer i we define $i \oplus 1 = i + 1$ if i is even and $i \oplus 1 = i - 1$ if i is odd.

2. Counting perfect matchings in bipartite graphs

In this section we prove Theorem 1. Let R be an arbitrary commutative semiring, and $A = (a_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ be an $m \times n$ matrix, $m \leq n$, with elements from R, and with at most dm non-zero elements, for some $d \geq 2$. Recall that the permanent of A is defined as

$$\mathrm{per}A = \sum_{\substack{\sigma:\{1,2,\ldots,m\}\to\{1,2,\ldots,n\} \\ \text{injective}}} \prod_{i=1}^{m} a_{i,\sigma(i)}.$$

We are to compute the permanent of A.

Let $K \subseteq \{1, 2, \ldots, m\}$ be the set of rows of A containing $\lfloor m/(\alpha d) \rfloor$ rows with the smallest number of non-zero entries, where $\alpha > 2$ is a constant to be determined later. By the commutativity of R, we may assume that K contains the bottommost $|K|$ rows. Let $L = \{1 \leq j \leq n : \exists_{i \in K} : a_{i,j} \neq 0\}$. Observe that $|L| \leq m/\alpha$, as the average number of non-zero entries in the rows of K is at most d. Again by using the commutativity of R, assume that $L = \{n - |L| + 1, n - |L| + 2, \ldots, n\}$, that is, we order the columns of A such that the columns of L appear at the right of the matrix. In particular for any $1 \leq i \leq n - m/\alpha$ and $j \in K$ we have $a_{i,j} = 0$ (see Figure 1).

Consider the following standard dynamic programming approach. For a subset X of rows of A and integer $|X| \leq r \leq n$ define $t[X, r]$ as the permanent of $(a_{i,j})_{i \in X, 1 \leq j \leq r}$, a $|X| \times r$ submatrix of A. Note that we are to compute $t[\{1, 2, \ldots, m\}, n]$. Observe that the following recursive formula allows to compute the entries of the table t, where we sum over the element used in the permanent computation in the r-th column of A:

$$t[X, r] = t[X, r - 1] + \sum_{i \in X} a_{i,r} t[X \setminus \{i\}, r - 1]\,,$$

with the boundary conditions $t[\emptyset, r] = 1$ for any $0 \leq r \leq n$ and $t[X, r] = 0$ for any $r < |X|$. Moreover, for computing $t[\{1, 2, \ldots, m\}, n]$ we do not need to compute the values $t[X, r]$ where $n - r < m - |X|$.

Let us upper bound the number of pairs (X, r) for which $n - r \geq m - |X|$ and $t[X, r]$ is non-zero. We consider two cases, depending on r.

First, consider the case $r \leq n - m/\alpha$. Recall that $a_{i,j} = 0$ for any $i \in K$ and $j \leq n - m/\alpha$. Hence, if $t[X, r] \neq 0$, then $X \cap K = \emptyset$. Since $|K| = \lfloor m/(\alpha d) \rfloor$, there are at most $n2^{m - \lfloor m/(\alpha d) \rfloor} \leq n2^{1 + (1 - 1/(\alpha d))m}$ pairs (X, r) with $t[X, r] \neq 0$ and $r \leq n - m/\alpha$.

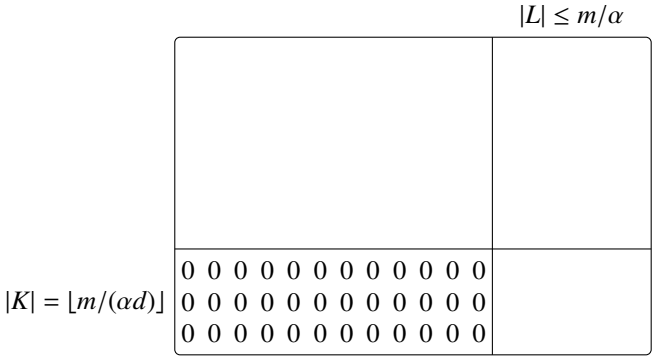

Figure 1: A block of zero entries in the binary matrix.

Second, consider the case $r > n - m/\alpha$. Recall that we are only interested in the values $t[X, r]$ for pairs (X, r) satisfying $n - r \geq m - |X|$. In our case this implies $|X| > m(1 - 1/\alpha)$ and, consequently, as $\alpha > 2$, there are only $m\binom{m}{\lfloor m/\alpha \rfloor}$ choices for the set X. By using the binary entropy function, we get $\binom{m}{\lfloor m/\alpha \rfloor} = O^\star(2^{H(1/\alpha)m})$, where $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$. For $d \geq 2$ and $\alpha = 3.55$ we have $2^{H(1/\alpha)} \leq 2^{1-1/(\alpha d)}$.

Consequently, to obtain the claimed running time of Theorem 1 it suffices to skip the computation of values $t[X, r]$ whenever $n - r < m - |X|$, or $r \leq n - m/\alpha$ and $X \cap K \neq \emptyset$. This finishes the proof of Theorem 1.

We remark here that the constant $\alpha = 3.55$ can be improved if we have a stronger lower bound on d. However, in our analysis it is crucial that $\alpha > 2$, so that the number of subsets $X \subseteq \{1, 2, \ldots, m\}$ of size $|X| > m(1 - 1/\alpha)$ is exponentially smaller than 2^m.

We would like also to observe that the assumption of commutativity of R is essential to perform the dynamic programming algorithm in the appropriate order of rows and columns (which, in the above proof, is disguised in the operation of reordering rows and columns of A). On the other hand, we do not need the additive inverse in R, hence the algorithm works for semirings, as claimed.

3. Counting perfect matchings in general graphs

In this section we study the problem of counting the number of perfect matchings in general graphs. We show an inclusion-exclusion based algorithm, which given an n-vertex graph computes the number of its perfect matchings in $O^\star(2^{n/2})$ time and polynomial space. This matches the time and space bounds of the algorithm of Björklund [5].

Theorem 4. *Given an n-vertex graph $G = (V, E)$ in $O^\star(2^{n/2})$ time and polynomial space one can count the number of perfect matchings in G.*

Proof. Clearly we can assume that n is even. Consider the edges of G being black and let $V = \{v_0, \ldots, v_{n-1}\}$. Now we add to the graph a perfect matching of red edges $E_R = \{v_{2i}v_{2i+1} : 0 \leq i < n/2\}$ obtaining a multigraph G'. Denote $e_i = v_{2i}v_{2i+1} \in E_R$.

We say that a cycle or a walk in G' is *alternating*, if, when we traverse the cycle (walk), the colours of the edges alternate; in particular, an alternating cycle or closed walk is of even length. Observe that for any perfect matching $M \subseteq E$ the multiset $M \cup E_R$ is a cycle cover (potentially with 2-cycles), where all the cycles are *alternating*. Moreover, for any cycle cover Y of G' composed of alternating cycles the set $Y \setminus E_R$ is a perfect matching in G. This leads us to the following observation.

Claim 5. *The number of perfect matchings in G equals the number of cycle covers in G' where each cycle is alternating.*

We emphasize here that our definition of cycle cover treats it as a subset of edges, and hence all the cycles are treated as undirected ones. This is in contrast with our definition of a closed walk, that not only is treated as a direct

one, but also has a designated starting point. However, note that the definition of *alternating* applies well to (closed) walks.

We are going to compute the number of cycle covers of G' with alternating cycles using the inclusion-exclusion principle over the set of red edges E_R.

For an edge $e_i \in E_R$, we say that a closed walk C is e_i-*nice* if it is alternating, starts with v_{2i}, e_i, v_{2i+1}, traverses e_i exactly once, and does not traverse any edge $e_j \in E_R$ for $j < i$. Note that for an alternating walk C, if C contains a vertex v_{2j} or v_{2j+1}, it needs to traverse the edge $e_j \in E_R$, as this is the only red edge incident to v_{2j} and v_{2j+1}. A closed walk is *nice* if it is e_i-nice for some $e_i \in E_R$; note that, in this case, the edge e_i is defined uniquely. Moreover, note that for any alternating cycle C there is exactly one way to define a nice closed walk with the same length and set of edges as C: one needs to start with the edge e_i traversed by C of lowest possible index, and traverse it in the direction from v_{2i} to v_{2i+1}.

For a positive integer r let us define the universe Ω_r as the set of r-tuples, where each of the r coordinates contains a nice closed walk in G' and the total length of all the walks equals n. For $0 \leq i < n/2$ let $A_{r,i} \subseteq \Omega_r$ be the set of r-tuples where at least one walk contains the arc e_i. We now claim the following.

Claim 6. *The number of perfect matchings in G equals $\sum_{1 \leq r \leq n/2} |\bigcap_{0 \leq i < n/2} A_{r,i}|/r!$.*

Proof. By Claim 5, it suffices to focus on cycle covers of G' where each cycle is alternating.

Consider first such a cycle cover Y. In the multigraph (V, Y) each connected component is an alternating cycle. Let r be the number of these cycles. Moreover, for each such cycle C, let $W(C)$ be the unique nice closed walk with the same length and edge set as C. Let $\tau = (W_1, W_2, \ldots, W_r)$ be any ordering of the walks $W(C)$, where C iterates over the set of cycles of Y. Observe that $\tau \in \Omega_r$, as each walk W_j is nice. As each cycle of Y is alternating, $E_R \subseteq Y$ and, consequently, for any $0 \leq i < n/2$ we have $\tau \in A_{r,i}$. Thus, each cycle cover Y gives rise to $r!$ elements of $\bigcap_{0 \leq i < n/2} A_{r,i}$ that differ on the choice of the order of walks W_j in the tuple τ. Furthermore, observe that the walks W_j are pairwise disjoint (and, hence, all these $r!$ tuples are pairwise different) and, moreover, different cycle covers Y yield different tuples τ, as their differ on the set of used edges.

In the other direction, consider a tuple $\tau = (W_1, W_2, \ldots, W_r) \in \bigcap_{0 \leq i < n/2} A_{r,i}$. By the definition of $A_{r,i}$, each edge of E_R needs to be contained in at least one of the walks W_j. On the other hand, the total length of the walks W_j is n, while $|E_R| = n/2$. Since the walks W_j are alternating, we infer that each edge E_R is traversed exactly once by exactly one walk W_j. Since each vertex of G' is incident to exactly one edge of E_R, we have that each W_j induces a cycle in G', and these cycles are pairwise vertex disjoint. Hence, the set of all edges of all walks in τ is a cycle cover in G' with each cycle being alternating. The claim follows. ⌟

By Claim 6, it suffices to compute $|\bigcap_{0 \leq i < n/2} A_{r,i}|$ for all $1 \leq r \leq n/2$. Let us now focus on a single value of r. By the inclusion-exclusion principle

$$\left| \bigcap_{0 \leq i < n/2} A_{r,i} \right| = \sum_{I \subseteq \{0, \ldots, n/2-1\}} (-1)^{|I|} \left| \bigcap_{i \in I} (\Omega_r \setminus A_{r,i}) \right|,$$

where we define $\bigcap_{i \in I} (\Omega_r \setminus A_{r,i})$ for $I = \emptyset$ as Ω_r. Hence to prove the theorem it is enough to compute the value $|\bigcap_{i \in I} (\Omega_r \setminus A_{r,i})|$ for a given $1 \leq r \leq n/2$ and $I \subseteq \{0, \ldots, n/2 - 1\}$ in polynomial time. This is done by a standard dynamic programing approach in the remainder of this proof.

Let G'_I be the graph G' with all the endpoints of edges e_i for $i \in I$ removed. For every $0 \leq a < n/2$ and every even $2 \leq q \leq n$, let $p_{a,q}$ be the number of e_a-nice closed walks in G'_I of length q. We first show how to compute each value $p_{a,q}$ in polynomial time.

To this end, for every $2a + 1 \leq i < n$ and every odd $0 < \hat{q} < q$ define $t_p[i, \hat{q}]$ as the number of alternating walks W of length $\hat{q}$ in G'_I, where each walk starts with v_{2a}, e_a, v_{2a+1}, ends with $v_{i \oplus 2}, e_{\lfloor i/2 \rfloor}, v_i$, visits e_a only once and does not visit any edge e_c for $c < a$. It is straightforward to verify that $t_p[i, \hat{q}]$ satisfies the following boundary conditions:

$$\begin{aligned} t_p[2a+1, 1] &= 1 \\ t_p[i, 1] &= 0 && \text{for } i \neq 2a+1 \\ t_p[2a+1, \hat{q}] &= 0 && \text{for } \hat{q} > 1 \end{aligned}$$

and the following recurence relation for every $2a+1 < i < n$ and every odd $3 \leq \hat{q} < q$:

$$t_p[i, \hat{q}] = \sum_{\substack{v_{i \oplus 1} v_j \in E(G'_I) \\ 2a+1 \leq j < n}} t_p[j, \hat{q} - 2].$$

Hence, the values $t_p[i, \hat{q}]$ can be computed in polynomial time. Finally, observe that

$$p_{a,q} = \sum_{\substack{v_a v_i \in E(G'_I) \\ 2a+1 \leq i < n}} t_p[i, q-1].$$

Having the values $p_{a,q}$, we now show how to compute $|\bigcap_{i \in I}(\Omega_r \setminus A_{r,i})|$ by the standard knapsack type dynamic programming. That is, for every $0 \leq \hat{r} \leq r$ and every even $0 \leq q \leq n$ we define $t[\hat{r}, q]$ to be the number of $\hat{r}$-tuples of nice closed walks in G'_I of total length q. It is straightforward to verify that $t[\hat{r}, q]$ satisfies the following boundary conditions:

$$\begin{aligned} t[0,0] &= 1 \\ t[\hat{r}, 0] = t[0, q] &= 0 \qquad \text{for } \hat{r}, q > 0 \end{aligned}$$

and the following recurence relation for every $0 < \hat{r} \leq r$ and every even $0 < q \leq n$:

$$t[\hat{r}, q] = \sum_{\substack{0 < \hat{q} < q \\ \hat{q} \text{ even}}} t[\hat{r} - 1, \hat{q}] \sum_{\substack{0 \leq a < n/2 \\ a \notin I}} p_{a, q - \hat{q}}.$$

Hence, the values $t[\hat{r}, q]$ can be computed in polynomial time. Finally, observe that

$$\left| \bigcap_{i \in I} (\Omega_r \setminus A_{r,i}) \right| = t[r, n].$$

This concludes the proof of Theorem 4. □

4. Properties of bounded average degree graphs

This section contains technical results concerning bounded average degree graphs. In particular we prove Lemma 12, which bounds the number of "Hamiltonian-like" structures in a graph and is needed to get the claimed running times in Theorems 2 and 3. However, as the proofs of this section are not needed to understand the algorithms in Section 5 the reader may decide to see only Definition 11 and the statement of Lemma 12.

We start with a few well-known bounds.

Lemma 7. *For any $n \geq k \geq 1$ it holds that*

$$\binom{n}{k} \leq \left(\frac{en}{k}\right)^k.$$

Moreover, the upper bounding function $x \mapsto (en/x)^x$ is nondecreasing on the domain $0 < x \leq n$.

Proof. For the first claim, observe that

$$\binom{n}{k} \leq \frac{n^k}{k!} \leq n^k \left(\frac{e}{k}\right)^k,$$

where the last inequality follows from the Stirling's approximation of the factorial function. For the second claim, substitute $y = n/x$ and observe that $y \mapsto y^{-1} \ln y$ is nonincreasing for $y \geq 1$. □

Lemma 8. *For any $n \geq 1$, it holds that $H_{n-1} \geq \ln n$, where $H_n = \sum_{i=1}^{n} \frac{1}{i}$.*

Proof. It is well-known that $\lim_{n\to\infty} H_n - \ln n = \gamma$ where $\gamma > 0.577$ is the Euler-Mascheroni constant and the sequence $H_n - \ln n$ is decreasing. Therefore $H_{n-1} = H_n - \frac{1}{n} \geq \ln n + \gamma - \frac{1}{n}$, hence the lemma is proven for $n \geq 2$ as $\gamma > \frac{1}{2}$. For $n = 1$, note that $H_{n-1} = \ln n = 0$. □

The following lemma helps us identify disjoint parts of the input graph where we can independently save on the number of "Hamiltonian-like" structures, provided that we have already bound the maximum degree of the graph.

Lemma 9. *Given an n-vertex graph $G = (V, E)$ of average degree at most d and maximum degree at most D one can in polynomial time find a set A containing $\lceil \frac{n}{2+4dD} \rceil$ vertices of degree at most $2d$, where for each $x, y \in A$, $x \neq y$ we have $N_G[x] \cap N_G[y] = \emptyset$.*

Proof. Note that $|V_{\deg \leq 2d}| \geq n/2$. We apply the following procedure. Initially we set $A := \emptyset$ and all the vertices are unmarked. Next, as long as there exists an unmarked vertex x in $V_{\deg \leq 2d}$, we add x to A and mark all the vertices $N_G[N_G[x]]$. Since the set $N_G[N_G[x]]$ contains at most $1 + 2d + 2d(D-1) = 1 + 2dD$ vertices, at the end of the process we have $|A| \geq \frac{n}{2+4dD}$. Clearly this routine can be implemented in polynomial time. □

The next lemma states that, although in an n-vertex graph of average degree at most d there may be up to $\frac{nd}{D}$ vertices of degree at least D, this bound cannot be tight for a large range of values of D at once. This simple observation is in fact the main source of the gain in bounds proven in Lemma 12.

Lemma 10. *For any $\alpha \geq 0$ and an n-vertex graph $G = (V, E)$ of average degree at most d there exists $D \leq e^\alpha$ such that $|V_{\deg > D}| \leq \frac{nd}{\alpha D}$.*

Proof. By standard counting arguments we have

$$\sum_{i=0}^{\infty} |V_{\deg > i}| = \sum_{i=0}^{\infty} i|V_{\deg = i}| \leq nd.$$

For the sake of contradiction assume that $|V_{\deg > i}| > \frac{nd}{\alpha i}$, for each $i \leq e^\alpha$. Then

$$\sum_{i=0}^{\infty} |V_{\deg > i}| \geq \sum_{i=1}^{\lfloor e^\alpha \rfloor} |V_{\deg > i}| > \frac{nd}{\alpha} \sum_{i=1}^{\lfloor e^\alpha \rfloor} 1/i = \frac{nd}{\alpha} H_{\lfloor e^\alpha \rfloor} \geq nd,$$

where the last inequality follows from Lemma 8. □

In the following definition we capture "Hamiltonian-like" structures in a graph. The family of all these structures contains the family of the sets used in the dynamic programming algorithms for TSP-like problems (that is, in Theorems 2 and 3).

Definition 11. *For an undirected multigraph $G = (V, E)$ and two vertices $s, t \in V$ by* deg2sets(G, s, t) *we define the set of all subsets $X \subseteq V \setminus \{s, t\}$, for which there exists a set of edges $F \subseteq E$ such that:*

- $\deg_F(v) = 0$ *for each* $v \in V \setminus (X \cup \{s, t\})$,
- $\deg_F(v) = 2$ *for each* $v \in X$,
- $\deg_F(v) \leq 1$ *for* $v \in \{s, t\}$.

We are now ready to state and prove the main result of this section.

Lemma 12. *For every $d \geq 1$ there exists a constant $\varepsilon_d > 0$, such that for an n-vertex multigraph $G = (V, E)$ without self-loops, of average degree at most d, for any $s, t \in V$ the cardinality of* deg2sets(G, s, t) *is at most $O^\star(2^{(1-\varepsilon_d)n})$.*

Proof. We begin with invoking Lemma 10 to get a partition of V into high- and low-degree vertices. Let c be a sufficiently large universal constant; it suffices to take $c = 20$. Lemma 10, invoked on G and $\alpha := e^{cd}$, provides us with an integer $D \leq e^{\alpha} = e^{e^{cd}}$ such that there are at most $\frac{nd}{\alpha D}$ vertices of degree greater than D in G. Recall that the standard averaging argument provides us with only a $\frac{nd}{D}$ upper bound on $|V_{\deg>D}|$; we are going to heavily rely on the fact that $|V_{\deg>D}|$ is in fact α times smaller than this bound.

The threshold D provided by Lemma 10 may turn out to be even smaller than d so, for technical reasons, we are going to use a slightly adjusted threshold $D' := \max(2d, D)$. Observe that $D' \leq e^{\alpha}$ as $d \geq 1$ implies $2d \leq e^{\alpha}$ for any $c \geq 1$. Let $H = G[V_{\deg\leq D'}]$ be the subgraph induced by the low-degree vertices and let $Y = V_{\deg>D'}$ be the (remaining) set of high-degree ones. Note that the bound of Lemma 10 implies that $|Y| \leq \frac{nd}{\alpha D}$, as $D' \geq D$ and $Y \subseteq V_{\deg>D}$.

The main reason why we have adjusted the threshold from D to D' is that $D' \geq 2d$ implies that H contains at least $n/2$ vertices. Moreover, as H contains all low-degree vertices of G, the average degree of H is not greater than the average degree of G, which in turn is bounded by d.

As H has bounded *maximum* degree by D', we may apply Lemma 9 to get a large family of pairwise disjoint closed neighbourhoods in H. Formally, we can in polynomial time construct $A \subseteq V(H)$ of $\lceil n/(4 + 8dD')\rceil$ vertices, each of degree at most $2d$, and with $N_H[v] \cap N_H[w] = \emptyset$ for any $v, w \in A, v \neq w$.

Let us now give an informal overview of the purpose of the set A. Assume for a moment that $Y = \emptyset$, that is, there are no high-degree vertices and $G = H$. Consider a set $X \in \text{deg2sets}(G, s, t)$ and a corresponding set $F \subseteq E$ from Definition 11. The crucial observation is that for any $v \in A \setminus \{s, t\}$, the set $X \cap N[v]$ cannot be equal to $\{v\}$, as v needs to be of degree 2 in F and G does not contain any self-loops by the assumptions of the lemma. Consequently, out of $2^{|N[v]|}$ choices for $X \cap N[v]$, at least one is invalid, and

$$\left|\text{deg2sets}(G, s, t)\right| \leq 2^n \cdot \prod_{v\in A} \frac{2^{|N[v]|} - 1}{2^{|N[v]|}} \leq 2^n \cdot \left(1 - \frac{1}{2^{2d+1}}\right)^{|A|}. \tag{1}$$

If $|A|$ is $\Omega(n)$, such a statement would finish the proof of the lemma. However, in the general case Y may not be empty, and $X \cap N_H[v] = \{v\}$ for some $v \in A \setminus \{s, t\}$ if both edges of F incident to v have their second endpoints in Y. Luckily, $|Y|$ is very small thanks to Lemma 10 and may "disturb" the argument of (1) only for a small part of A. Thus, to count sets $X \in \text{deg2sets}(G, s, t)$, for each choice of small subset of A that get "disturbed" by the elements of Y, we apply the argument of (1) to the remainder of A. Thanks to the bound on $|Y|$, the gain from the argument of (1) is larger than the overhead we get from enumerating all sets of vertices of A "disturbed" by Y.

Let us now proceed with a formal argumentation. First, we rephrase the bound on $|A|$ so that it is more convienient in the future. Since $d \geq 1$ and $D' \geq 2d$, we have:

$$|A| = \left\lceil \frac{n}{4 + 8dD'} \right\rceil \geq \frac{n}{4 + 8dD'} \geq \frac{n}{2dD' + 8dD'} = \frac{n}{10dD'}. \tag{2}$$

If $n \leq \frac{8edD'}{4-e}$, $n = O(1)$ and the claim of the lemma is trivial. Otherwise:

$$|A| = \left\lceil \frac{n}{4 + 8dD'} \right\rceil < \frac{n}{8dD'} + 1 < \frac{n}{2edD'}. \tag{3}$$

Recall that, by Lemma 10, $|Y|$ is very small, namely $|Y| \leq \frac{nd}{\alpha D}$. We now formally show that $|Y|$ is much smaller than $|A|$. As $d \geq 1$ and $D' = \max(2d, D) \leq 2dD$, for sufficiently large c we have:

$$|Y| \leq \frac{nd}{\alpha D} \leq \frac{n}{20dD'} \cdot \frac{40d^3}{e^{cd}} \leq \frac{|A|}{2} \cdot \frac{40d^3}{e^{cd}} < \frac{|A|}{2}. \tag{4}$$

For an arbitrary set $X \in \text{deg2sets}(G, s, t)$, and a corresponding set $F \subseteq E$ from Definition 11, define Z_X as the set of vertices $x \in (X \cap V(H)) \setminus \{s, t\}$ such that $N_H(x) \cap X = \emptyset$. Intuitively, these are the vertices that get "disturbed" by Y in the argument of (1). Formally, recall that F is a set of paths and cycles, where each vertex of Z_X is of degree two. Hence F contains at least $2|Z_X|$ edges between Z_X and Y, as any path/cycle of F visiting a vertex of Z_X has to enter from Y and leave to Y (G does not contain any self-loops). Hence by the upper bound of 2 on the degrees in F we have $|Z_X| \leq |Y|$: only a few vertices of A get "disturbed" by Y.

By the definition of Z_X, for each $x \in A \setminus (Z_X \cup \{s,t\})$ we have that $N_H[x] \cap X \neq \{x\}$ and $|N_H[x]| \leq 2d+1$. Recall that if $x \in A \cap Z_X$, we have $N_H[x] \cap X = \{x\}$. Therefore, following the lines of (1), we have that for fixed $A \cap Z_X$ there are at most

$$2^n \left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A\setminus(Z_X\cup\{s,t\})|} \left(\frac{1}{2}\right)^{|A\cap Z_X|} \leq 2^{n+2}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|} \tag{5}$$

choices for $X \in \mathrm{deg2sets}(G,s)$.

As $|Z_X| \leq |Y|$ and $|Y|$ is much smaller than $|A|$, we can bound the number of choices of $A \cap Z_X$ in a straightforward manner: there are at most $\sum_{i=0}^{|Y|}\binom{|A|}{i} \leq n\binom{|A|}{|Y|}$ choices for $A \cap Z_X$. Thus

$$|\mathrm{deg2sets}(G,s,t)| \leq 2^{n+2} \cdot \left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|} \cdot n\binom{|A|}{|Y|}. \tag{6}$$

It remains to estimate $\binom{|A|}{|Y|}$. To this end, we use Lemma 7 as follows:

$$\begin{aligned}
\binom{|A|}{|Y|} &\leq \left(\frac{e|A|}{|Y|}\right)^{|Y|} && \text{by Lemma 7}\\
&\leq \left(e \cdot \frac{n}{2edD'} \cdot \frac{1}{|Y|}\right)^{|Y|} && \text{by (3)}\\
&\leq \left(e \cdot \frac{n}{2edD'} \cdot \frac{\alpha D}{nd}\right)^{\frac{nd}{\alpha D}} && \text{by } |Y| \leq \frac{nd}{\alpha D} \leq \frac{n}{2edD'} \text{ (cf. (4)) and by Lemma 7}\\
&\leq \left(\frac{\alpha}{2d^2}\right)^{\frac{nd}{\alpha D}} && \text{since } D \leq D'\\
&< \alpha^{\frac{nd}{\alpha D}}. && (7)
\end{aligned}$$

We now proceed to the final calculations, putting all obtained bounds together. By the standard inequality $1-x \leq e^{-x}$ we have that

$$(2^{2d+1}-1)/2^{2d+1} = (1-1/2^{2d+1}) \leq e^{-1/2^{2d+1}}. \tag{8}$$

Using (2), (7) and (8) we obtain that

$$\binom{|A|}{|Y|}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2} \leq \exp\left(\frac{nd\ln\alpha}{\alpha D} - \frac{n}{20dD'2^{2d+1}}\right).$$

Plugging in $\alpha = e^{cd}$ and using the fact that $e^{10d} > 40d^3$ for $d \geq 1$ we obtain:

$$\binom{|A|}{|Y|}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2} \leq \exp\left(\frac{ncd}{e^{(c-10)d}20d\cdot 2dD} - \frac{n}{20dD'2^{2d+1}}\right).$$

Since $D' = \max(2d, D) \leq 2dD$ and $e^{5d} > d2^{2d+1}$ as $d \geq 1$, we get

$$\binom{|A|}{|Y|}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2} \leq \exp\left(\frac{n}{20dD'2^{2d+1}}\left(\frac{c}{e^{(c-15)d}} - 1\right)\right).$$

Finally, for sufficiently large c, as $d \geq 1$, we have $c < e^{(c-15)d}$ and

$$\binom{|A|}{|Y|}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2} < 1. \tag{9}$$

Consequently, we obtain:

$$\begin{aligned}
|\mathrm{deg2sets}(G,s,t)| &\leq 2^{n+2}\cdot\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|}\cdot n\binom{|A|}{|Y|} && \text{by (6)}\\
&= n2^{n+2}\cdot\left(\binom{|A|}{|Y|}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2}\right)\cdot\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2}\\
&< n2^{n+2}\left(\frac{2^{2d+1}-1}{2^{2d+1}}\right)^{|A|/2} && \text{by (9)}\\
&\leq n2^{n+2}\exp\left(-\frac{1}{2^{2d+1}}\cdot\frac{|A|}{2}\right) && \text{by (8)}\\
&\leq n2^{n+2}\exp\left(-\frac{n}{2^{2d+1}\cdot 20dD'}\right) && \text{by (2)}\\
&\leq n2^{n+2}\exp\left(-\frac{n}{2^{2d+1}\cdot 20d\cdot e^{e^{cd}}}\right) && \text{as } D'\leq e^{\alpha}.
\end{aligned}$$

This concludes the proof of the lemma. Note that the dependency on d in the final constant ε_d is doubly-exponential. □

5. Trimming dynamic programming algorithms in graphs of bounded average degree

5.1. Minimum weight Hamiltonian cycle

To prove Theorem 2, it suffices to solve in $O^\star(2^{(1-\varepsilon_d)n})$ time the following problem. We are given an undirected n-vertex graph $G=(V,E)$ of average degree at most d, vertices $a,b\in V$ and a cost function $c:E\to\mathbb{R}_+$. We are to find the cheapest Hamiltonian path between a and b in G, or verify that no Hamiltonian ab-path exists.

We solve the problem by the standard dynamic programing approach. That is for each $a\in X\subseteq V$ and $v\in X$ we compute $t[X,v]$, which is the cost of the cheapest path from a to v with the vertex set X. The entry $t[V,b]$ is the answer to our problem. Note that it is enough to consider only such pairs (X,v), for which there exists an av-path with the vertex set X.

We first set $t[\{a\},a]=0$. Then iteratively, for each $i=1,2,\ldots,n-1$, for each $u\in V$, for each $X\subseteq V$ such that $|X|=i$, $a,u\in X$ and $t[X,u]$ is defined, for each edge $uv\in E$ where $v\notin X$, if $t[X\cup\{v\},v]$ is undefined or $t[X\cup\{v\},v]>t[X,u]+c(uv)$, we set $t[X\cup\{v\},v]=t[X,u]+c(uv)$.

Finally, note that if $t[X,v]$ is defined then $X\setminus\{a,v\}\in\mathrm{deg2sets}(G,a,v)$. Hence, the complexity of the above algorithm is within a polynomial factor from $\sum_{v\in V}|\mathrm{deg2sets}(G,a,v)|$, which is bounded by $O^\star(2^{(1-\varepsilon_d)n})$ by Lemma 12.

5.2. Counting perfect matchings

In this section we show how the algorithm from Section 3 can be reformulated as a dynamic programming routine (using exponential space), which together with Lemma 12 will imply the running time claimed in Theorem 3. Note that this reformulation causes the space complexity to be exponential.

Assume that we are given an n-vertex undirected graph $G=(V,E)$, where n is even, and we are to count the number of perfect matchings in G. We start with a small technical detour to ensure that the complement of G has perfect matching; the motivation for this step will become clear later.

Lemma 13. *Given an n-vertex graph $G=(V,E)$ with average degree at most d, one can in polynomial time identify a set X of at most $4d+4$ vertices of G such that the complement of $G\setminus X$ contains a perfect matching.*

Proof. If $n\leq 4d+4$, we may output $X:=V$, so assume othewise. Construct a set $Y\subseteq V$ as follows: initiate $Y=\emptyset$ and, as long as $|Y|<2d$ and there exists a vertex $v\in V\setminus Y$ with $|N(v)\setminus Y|>\frac{|V\setminus Y|}{2}-1$, add v to Y.

We now show that $|Y|<2d$, that is, the construction of Y always stops by triggering the second condition. Assume otherwise, $2d\leq|Y|<2d+1$. At each step of the construction of Y, at least $(n-|Y|-1)/2$ edges are removed from $G\setminus Y$. Note that, by the assumptions on n and $|Y|$, we have $n-|Y|-1>n/2$. Thus, the number of edges incident to Y is strictly greater than $|Y|\frac{n}{4}\geq\frac{dn}{2}$. However, the total number of edges of G is bounded by $dn/2$, a contradiction.

As the construction of Y stopped by triggering the second condition, the complement of $G \setminus Y$ has minimum degree at least $|V \setminus Y|/2$ and, by Dirac's theorem, it contains a Hamiltonian cycle. Define $X = Y$ if $|V \setminus Y|$ is even and $X = Y \cup \{v\}$, where v is an arbitrary vertex of $V \setminus Y$, if $|V \setminus Y|$ is odd. In this manner $|X| < 2d + 1$ and the complement of $G \setminus X$ contains a Hamiltonian path and has even number of vertices. Note that selecting every other edge of a Hamiltonian path gives us a perfect matching in $G \setminus X$. This concludes the proof of the lemma. □

We preprocess the input graph $G = (V, E)$ as follows. We compute the set X using Lemma 13 and add $|X|/2$ connected components isomorphic to K_2 to the graph G. In this manner, the complement of G contains a perfect matching: the new vertices can be matched to the vertices of X and the complement of $G \setminus X$ contains a perfect matching by Lemma 13. Moreover, note that the number of perfect matchings of G does not change, the number of vertices of G increases only by an additive factor of at most $4d + 4$, which is a constant, and the new graph has still average degree bounded by d (recall $d \geq 1$).

Thus, by using the aforementioned preprocessing, we may henceforth focus only on the case when the complement of the input n-vertex graph $G = (V, E)$ contains a perfect matching. Order the vertices of G as $v_0, v_1, \ldots, v_{n-1}$, so that $v_{2i}v_{2i+1} \notin E$ for any $0 \leq i < n/2$.

We are going to construct an undirected multigraph G' having only $n/2$ vertices, where the edges of G' will be labeled with unordered pairs of vertices of G, that is, with edges of G. As the set of vertices of $G' = (V', E')$ we take $V' = \{v'_0, \ldots, v'_{n/2-1}\}$. For each edge v_av_b of G we add to G' exactly one edge: $v'_{\lfloor a/2 \rfloor}v'_{\lfloor b/2 \rfloor}$ labeled with $\{v_a, v_b\}$. For an edge $e' \in E'$ by $\ell(e')$ let us denote the label of e'. Note that G' may contain parallel edges but, due to the preprocessing step, G' does not contain self-loops. In other words, the preprocessing step allows us to use Lemma 12 on the graph G'.

Observe also that if the graph G is of average degree at most d, then the graph G' is of average degree at most $2d$.

In what follows we count the number of particular cycle covers of G', where we use the labels of edges to make sure that a cycle going through a vertex $v'_i \in V'$ never uses two edges of G' corresponding to two edges of G incident to the same vertex.

Lemma 14. *The number of perfect matchings in G equals the number of cycle covers $C \subseteq E'$ of G', where $\bigcup_{e \in C} \ell(e) = V$.*

Proof. We show a bijection between perfect matchings in G and cycle covers C of G' satisfying the condition $\bigcup_{e \in C} \ell(e) = V$.

Let M be a perfect matching in G. As $f(M)$ we define $f(M) = \{v'_{\lfloor a/2 \rfloor}v'_{\lfloor b/2 \rfloor} : v_av_b \in M\}$. Note that $f(M)$ is a cycle cover and moreover $\bigcup_{e \in f(M)} \ell(e) = V$. In the reverse direction, for a cycle cover $C \subseteq E'$ of G', consider a set of edges $h(C)$ defined as $h(C) = \{\ell(e) : e \in C\}$. Clearly the condition $\bigcup_{e \in C} \ell(e) = V$ implies that $h(C)$ is a perfect matching, and moreover $h = f^{-1}$. □

Observe, that if a cycle cover $C \subseteq E'$ of G' does not satisfy $\bigcup_{e \in C} \ell(e) = V$, then there is a vertex $v'_i \in V'$, such that the two edges of C incident to v'_i do not have disjoint labels. Intuitively this means we are able to verify the condition $\bigcup_{e \in C} \ell(e) = V$ locally, which is enough to derive the following dynamic programming routine.

Lemma 15. *In $O^\star(\sum_{s,t \in V} |\mathrm{deg2sets}(G', s, t)|)$ time and space, one can compute the number of cycle covers C of G' satisfying $\bigcup_{e \in C} \ell(e) = V$.*

Proof. An *ordered r-cycle cover* of a graph H is a tuple of r cycles in H, whose union is a cycle cover of H. As each cycle cover of H that contains exactly r cycles can be ordered into exactly $r!$ different ordered r-cycle covers, it is sufficient to count, for any $1 \leq r \leq n/2$, the number of ordered r-cycle covers C in G' such that each two edges in C have disjoint labels. In the rest of the proof, we focus on one fixed value of r.

For $0 \leq q \leq r$ and $X \subseteq V'$ as $t[q, X]$ let us define the number of ordered q-cycle covers in $G'[X]$ where each two edges have disjoint labels; note that $t[r, V']$ is exactly the value we need. Moreover for $0 \leq q < r$, $X \subseteq V'$, $v'_a, v'_b \in X$, $a < b$ and $\zeta \in \{2b, 2b + 1\}$ as $t_2[q, X, v'_a, v'_b, \zeta]$ we define the number of pairs (C, P) where all the following conditions hold:

- C is an ordered q-cycle cover of $G'[Y]$ for some $Y \subseteq X \setminus \{v'_a, v'_b\}$;
- P is a $v'_av'_b$-path with the vertex set $X \setminus Y$ that does not contain any vertex v'_c with $c < a$;

- any two edges of $C \cup P$ have disjoint labels;
- the label of the edge of P incident to v'_a contains v_{2a}; and
- the label of the edge of P incident to v'_b contains v_ζ.

Note that we have the following border values: $t[0, \emptyset] = 1$ and $t[0, X] = 0$ for $X \neq \emptyset$.

Consider an entry $t_2[q, X, v'_a, v'_b, \zeta]$, and let (C, P) be one of the pairs counted in it. We have two cases: either P is of length 1 or longer. The number of pairs (C, P) in the first case equals

$$t[q, X \setminus \{v'_a, v'_b\}] \cdot |\{v'_a v'_b \in E' : \ell(v'_a v'_b) = \{v_{2a}, v_\zeta\}\}|.$$

In the second case, let $v'_c v'_b$ be the last edge of P; note that $c > a$ by the assumptions on P. The label of $v'_c v'_b$ equals $\{v_{2c}, v_\zeta\}$ or $\{v_{2c+1}, v_\zeta\}$. Thus, the number of elements (C, P) in the second case equals

$$\sum_{\substack{v'_c \in X \setminus \{v'_a, v'_b\} \\ a<c}} \sum_{\eta \in \{2c, 2c+1\}} t_2[q, X \setminus \{v'_b\}, v'_a, v'_c, \eta \oplus 1] \cdot |\{v'_c v'_b \in E' : \ell(v'_c v'_b) = \{v_\eta, v_\zeta\}\}|.$$

Let us now move to the entry $t[q, X]$ and let C be an ordered q-cycle cover in $G'[X]$. Let W be the last cycle of C. Observe that, since G' does not contain any self-loops, W is of length at least two. Let v'_a be the lowest-numbered vertex on W and let $e = v'_a v'_b$ be the edge of W where $v_{2a+1} \in \ell(e)$. Note that both v'_a and e are defined uniquely; moreover, $a < b$ and no vertex v'_c with $c < a$ belongs to W. Thus

$$t[q, X] = \sum_{\substack{v'_a, v'_b \in X \\ a<b}} \sum_{\zeta \in \{2b, 2b+1\}} t_2[q-1, X, v'_a, v'_b, \zeta \oplus 1] \cdot |\{v'_a v'_b \in E' : \ell(v'_a v'_b) = \{v_{2a+1}, v_\zeta\}\}|.$$

So far we have given recursive formulas, that allow computing the entries of the tables t and t_2. However the values $t[q, X]$, $t_2[q, X, v'_a, v'_b, \zeta]$ for $X \notin \bigcup_{s,t \in V'} \mathrm{deg2sets}(G', s, t)$ are equal to zero. The last step of the proof is to show how to perform the dynamic programming computation in a time complexity within a polynomial factor from the number of non-zero entries of the table. We do that in a bottom-up manner, that is iteratively, for each $q = 0, 1, 2, \ldots, r$, for each $i = 1, 2, \ldots, n$, we want to compute the values of non-zero entries $t[q, X]$ for all sets X of cardinality i and then compute the values of non-zero entries $t_2[q, X, *, *, *]$ for all sets X of cardinality i.

Whenever we compute some value $t[q, X]$ or $t_2[q, X, *, *, *]$ and it turns out to be non-zero, we enqueue for further computation all values $t[q', X']$ or $t_2[q', X', *, *, *]$ for which the currently computed value contributes to in the recursive formulae. Observe that each value $t[q, X]$ of $t_2[q, X, *, *, *]$ contributes to a polynomial number of applications of the recursive formulae. We process the values $t[q, X]$ and $t_2[q, X, *, *, *]$ in the aforementioned order (i.e., in the increasing order of q and $|X|$, and, for fixed q and X, first $t[q, X]$ and then $t_2[q, X, *, *, *]$), but we only focus on the cells that were previously enqueued.

The single non-zero value that we start with is $t[0, \emptyset] = 1$. (Note that $t[0, X] = 0$ for $X \neq \emptyset$.) This finishes the proof of the lemma. □

Theorem 3 follows directly from the Lemma 12 together with Lemma 15.

6. Conclusions

In our work we have shown how the assumption of bounded average degree of a graph can provide exponential speed up in the classic dynamic programming routines. Very recently Golovnev et al. [20] proved that the dynamic programming algorithms of Theorems 2 and 3 can be turned into polynomial-space algorithms by using algebraic techniques. Moreover, they also showed how to use the gap obtained in Lemma 10 to compute the chromatic number of a graph of average degree d with exponential speedup over the $O^\star(2^n)$-time algorithm [21]. We believe our approach, in particular Lemma 10, may inspire to further improvements in the area of exponential-time algorithms.

Acknowledgements

We would like to thank anonymous referees for their helpful remarks and comments.
This research has been partially supported by NCN grant N206567140 and Foundation for Polish Science.

[1] R. Bellman, Dynamic programming treatment of the travelling salesman problem, J. ACM 9 (1962) 61–63.
[2] M. Held, R. M. Karp, A dynamic programming approach to sequencing problems, J. Soc. Ind. Appl. Math. 10 (1962) 196–210.
[3] G. J. Woeginger, Exact algorithms for NP-hard problems: A survey, in: M. Jünger, G. Reinelt, G. Rinaldi (Eds.), Combinatorial Optimization, Vol. 2570 of Lecture Notes in Computer Science, Springer, 2001, pp. 185–208.
[4] H. Ryser, Combinatorial Mathematics, The Carus Mathematical Monographs, Mathematical Association of America, 1963.
[5] A. Björklund, Counting perfect matchings as fast as Ryser, in: Y. Rabani (Ed.), SODA, SIAM, 2012, pp. 914–921.
[6] A. Björklund, Determinant sums for undirected hamiltonicity, in: FOCS, IEEE Computer Society, 2010, pp. 173–182.
[7] R. Impagliazzo, R. Paturi, On the complexity of k-SAT, J. Comput. Syst. Sci. 62 (2) (2001) 367–375.
[8] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, M. Wahlström, On problems as hard as CNF-SAT, in: IEEE Conference on Computational Complexity, IEEE Computer Society, 2012, pp. 74–84.
[9] D. Lokshtanov, D. Marx, S. Saurabh, Known algorithms on graphs on bounded treewidth are probably optimal, in: D. Randall (Ed.), SODA, SIAM, 2011, pp. 777–789.
[10] M. Pătraşcu, R. Williams, On the possibility of faster SAT algorithms, in: M. Charikar (Ed.), SODA, SIAM, 2010, pp. 1065–1075.
[11] E. T. Bax, J. Franklin, A permanent algorithm with $\exp(\Omega(n^{\frac{1}{3}})/2\ln n)$ expected speedup for 0-1 matrices, Algorithmica 32 (1) (2002) 157–162.
[12] R. A. Servedio, A. Wan, Computing sparse permanents faster, Inf. Process. Lett. 96 (3) (2005) 89–92.
[13] T. Izumi, T. Wadayama, A new direction for counting perfect matchings, in: FOCS, IEEE Computer Society, 2012, pp. 591–598.
[14] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Evaluation of permanents in rings and semirings, Inf. Process. Lett. 110 (20) (2010) 867–870.
[15] A. Björklund, T. Husfeldt, Exact algorithms for exact satisfiability and number of perfect matchings, Algorithmica 52 (2) (2008) 226–249.
[16] M. Koivisto, Partitioning into sets of bounded cardinality, in: J. Chen, F. V. Fomin (Eds.), IWPEC, Vol. 5917 of Lecture Notes in Computer Science, Springer, 2009, pp. 258–263.
[17] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, The traveling salesman problem in bounded degree graphs, ACM Transactions on Algorithms 8 (2) (2012) 18.
[18] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Fourier meets Möbius: fast subset convolution, in: D. S. Johnson, U. Feige (Eds.), STOC, ACM, 2007, pp. 67–74.
[19] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Trimmed Möbius inversion and graphs of bounded degree, Theory Comput. Syst. 47 (3) (2010) 637–654.
[20] A. Golovnev, A. S. Kulikov, I. Mihajlin, Families with infants: a general approach to solve hard partition problems, CoRR abs/1311.2456.
[21] A. Björklund, T. Husfeldt, M. Koivisto, Set partitioning via inclusion-exclusion, SIAM J. Comput. 39 (2) (2009) 546–563.