Noname manuscript No.
(will be inserted by the editor)

DIFFERENTIAL GEOMETRIC TREEWIDTH ESTIMATION IN ADIABATIC QUANTUM COMPUTATION

Chi Wang · Edmond Jonckheere · Todd Brun

Received: date / Accepted: date

Abstract The D-Wave adiabatic quantum computing platform is designed to solve a particular class of problems—the Quadratic Unconstrained Binary Optimization (QUBO) problems. Due to the particular "Chimera" physical architecture of the D-Wave chip, the logical problem graph at hand needs an extra process called minor embedding in order to be solvable on the D-Wave architecture. The latter problem is itself NP-hard. In this paper, we propose a novel polynomial-time approximation to the closely related treewidth based on the differential geometric concept of Ollivier-Ricci curvature. The latter runs in polynomial time and thus could significantly reduce the overall complexity of determining whether a QUBO problem is minor-embeddable, and thus solvable on the D-Wave architecture.
Keywords: Adiabatic Quantum Computation, Chimera architecture, minor embedding, treewidth, Ollivier-Ricci curvature

1 Introduction

1.1 Quantum annealing

Adiabatic Quantum Computation (AQC), first proposed in [17], was originally designed to solve such general optimization problems as 2-SAT and 3-SAT. In the more specific D-Wave implementation, it maps a Quadratic Unconstrained Binary Optimization (QUBO) problem, defined as

$$\min_X E(x_1, x_2, ..., x_n) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{i<j=1}^{n} c_{ij} x_i x_j, \quad x_i \in \{0, 1\}, \tag{1}$$

Dept. of Electrical Eng., University of Southern California, Los Angeles, CA 90089, USA

to the problem of finding the ground energy level of an Ising chain defined as

$$\begin{aligned} H_{\text{Ising}} &= \sum_{i=1}^{n} h_i \sigma_i^z + \sum_{1 \le i < j}^{n} J_{ij} \sigma_i^z \sigma_k^z, \\ \sigma_i^z &= I_{n\times n}^{\otimes i-1} \otimes \sigma^z \otimes I_{n\times n}^{\otimes n-i}, \end{aligned} \tag{2}$$

where σ^z is the usual Pauli operator. The annealer prepares an initial transverse magnetic field, an equal superposition of 2^n computational basis states, as

$$H_{\text{trans}} = -\sum_{k=1}^{n} \sigma_k^x, \tag{3}$$

where σ_i^x is defined as in (2) except for the Pauli operator σ^x being substituted for σ^z. During adiabatic evolution, the Hamiltonian evolves smoothly from H_{trans} to H_{Ising} with

$$H(t) = (1-s)H_{\text{trans}} + sH_{\text{Ising}}, \quad s \in [0,1]. \tag{4}$$

From the adiabatic theorem, if the evolution is "slow enough," the system would remain in its ground state; thus, the solution to the original QUBO problem could be obtained, with a certain success probability, from measurements on the D-Wave Ising system.

Adiabatic Quantum Computation is widely used in many research fields that feature optimization, graphical models, Bayesian networks, etc. One of the specific areas of applications is computer vision problems involving minimization of energy functions. Felzenszwalb [1] provides an insightful survey of the applications of QUBO in computer vision. QUBO has been proven to be NP-hard. There is some evidence that the D-Wave quantum computer gives a modest speedup over classical solvers for QUBO problems, and may provide a large speedup for *some* instances of QUBO problems [2]. Recently, on the D-Wave 2X chip with 1152 qubits, the speedup reached up to three orders of magnitude for a subset of scenarios in multiple query optimization problems [18].

1.2 D-Wave architecture and minor embedding

The general idea behind D-Wave is to embed a QUBO problem graph

$$G = (V = \{1, 2, ..., n\}, E = \{ij : c_{ij} \neq 0\})$$

into a hardware graph in order for the problem to be solvable by adiabatic evolution on the hardware. Due to physical architectural considerations, such as limited qubit fan-out, minimizing coupler strengths, 2D chip integration etc. [19], the current hardware architecture is designed using $K_{4,4}$ bipartite cells interconnected in a 2-dimensional lattice, referred to as "Chimera topology."

For a problem graph to be embedded into a hardware graph, it is required that the problem graph be a subgraph of the architecture graph. In most instances of general problems, this is a very strong requirement, since the hardware graph is fixed. In the D-Wave architecture, *minor embedding* instead of *subgraph embedding* is used to allow 1-to-many vertex mapping [3]. By properly adjusting the coupling strengths of particular edges and nodes [4], more than one physical qubits can represent the same logical qubit, thus greatly increasing the range of graphs that can be minor embedded into a fixed hardware graph, at the cost of using more resources (more physical qubits).

The definition of minor embedding is as follows. Let U be a fixed hardware graph. Given a problem graph G, the ***minor embedding*** of G is defined by $\phi : G \to U$ such that (i) each vertex v in $V(G)$ is mapped to a connected subtree T_v of U; (ii) there exists a map $V(G) \times V(G) \to E(U)$ such that for each $vw \in E(G)$, there are corresponding $i_v \in V(T_v)$ and $i_w \in V(T_w)$ such that $i_v i_w \in E(U)$.

Minor embedding relaxes the original requirement of subgraph embedding, provided that the resources (number of physical qubits) are adequate. Crucially related to minor embedding is the concept of ***tree decomposition*** T of G: Each vertex $i \in I$ of the tree T abstracts a subset $\mathcal{V}_i$ of vertices of G, called a "bag," such that (i) $\cup_{i \in I} \mathcal{V}_i = V(G)$; (ii) for any $vw \in E(G)$, there is a $i \in I$ such that $v, w \in \mathcal{V}_i$; (iii) for any $v \in V$, the set $\{i \in I : v \in \mathcal{V}_i\}$ forms a connected subtree of T. The width of a tree decomposition is $\max_i(|\mathcal{V}_i| - 1)$. The ***treewidth (*****tw*****)*** is the minimum width over all tree decompositions. The ***treelength*** is the minimum over all decompositions of the maximum of the diameters of the bags.

1.3 Paper outline

This paper is organized as follows: In section 2, the minor embeddability problem at hand will be reviewed and properly formulated in the present context, and the proposed approximate but fast solution will be introduced. In section 3, the mathematical background of our approach—the differential-geometric Ollivier-Ricci curvature—will be developed. In section 4, experimental results will be shown that prove that in practice our approach can very efficiently rule out cases where minor embedding fails.

The main contributions of this paper are **(1)** An embeddability preprocessor for adiabatic quantum computation; **(2)** A numerical connection (and a "weak" theoretical link) between graph curvature and treewidth; **(3)** A quick polynomial time curvature-based rule to reject cases where minor embedding is not possible, thereby greatly reducing the amount of time the heuristics might spent at attempting to construct an embedding.

2 Minor embeddability and problem formulation

2.1 Treewidth and minor embeddability

To determine whether a problem graph has a chance to be minor-embeddable into a hardware graph, one needs to check whether the treewidth of the problem graph is not larger than that of the hardware graph, following a well-known graph minor theorem as discussed in more detail in [5, 21, 23, 24]:

Theorem 1 *If G is a minor of H, then* $\mathrm{tw}(G) \leq \mathrm{tw}(H)$.

The problem is that the above condition is necessary but not sufficient for minor embeddability.

For regular graphs, the estimation of the treewidth is both standard and efficient. For $F(m, c)$, the graph of an array of $m \times m$ cells with each cell a bipartite graph $K_{c,c}$ (the D-Wave architecture), the treewidth of the graph has very tight lower and upper bounds represented as [5]

$$cm \leq \mathrm{tw}(F(m, c)) \leq cm + c - 1. \tag{5}$$

For the 128-qubit D-Wave "Chimera" architecture consisting of a 4×4 array of $K_{4,4}$ bipartite graph cells, the treewidth approximation has a range of $[16, 19]$, and it is exactly 17 if one needs the accurate treewidth.

However, as noted in [20], the biggest challenge to solve practical problems in quantum adiabatic computation is minor embedding of general graphs. The major problem here is not to find the treewidth of a fixed architecture graph U, but the treewidth of any arbitrary problem graph G. In order to rule out embeddability, that is, the case where $\mathrm{tw}(G) > \mathrm{tw}(U)$, a fast approximation of treewidth is still needed. In general, determining the treewidth of arbitrary graph is NP-hard [21, 22]. Although a linear-time algorithm to determine whether a graph has treewidth at most k is proposed in [21], the constants in the algorithm are extremely large and grow exponentially with k, which makes it impractical for most graphs [5]. Thus, heuristics are needed for practical considerations.

2.2 Problem formulation and related work

There exist several recent quantum annealing applications running on D-Wave, including solving database optimization [18], graph isomorphism [29], power system fault detection [28], Bayesian network structure learning [27], and operational planning [26]. However, in these applications, the problem size is relatively small and solvable by minor embedding heuristics proposed in [25] and implemented on D-Wave. Embeddability is not generally pre-determined, but decided on a trial-and-error basis. Also, in most cases, embedding only needs to be determined once and used throughout the entire annealing process. However, this is not practical if the topology of the QUBO (i.e., the topology of the original graph problem) is not fixed. The latter is precisely the case

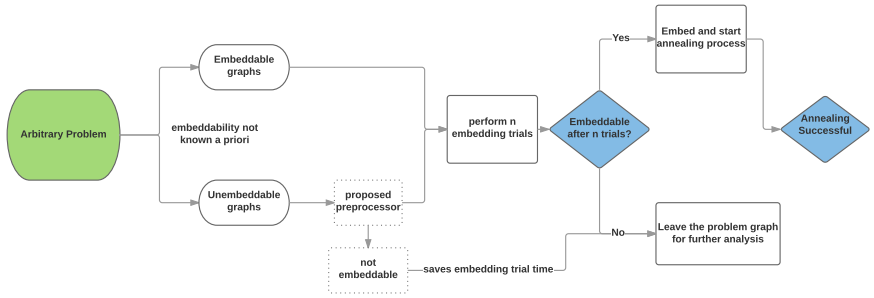

Fig. 1 Proposed problem pre-processor for the D-Wave adiabatic quantum computing platform. The arbitrary graph could be either "embeddable" or "unembeddable," but this is not known. The "preprocessor" acts only on the "unembeddable" graphs, as it only checks a necessary condition for embeddability, in polynomial-time in order to save embedding trial time. Note that even if the necessary condition test passes, the graph could still be unembeddable so that embedding trials are still necessary.

when solving the network scheduling problem in wireless networks [30], which we take as a prototype for more general cases of problems with time-varying QUBO topology.

The D-Wave heuristics of minor embedding $G \to H$ reduces the time complexity of the current best exact algorithm of $O(2^{(2k+1)\log k}|n_H|^{2k}2^{2|n_H|^2}|e_G|)$ [31] where k is the branchwidth of G and n_*, e_* the graph order, size, resp., to $O(n_H n_G e_H(e_G + n_G \log n_G))$, at the cost of the following:

- Success with a certain probability, and re-tries on failure;
- No attempt to prove minor exclusion.

We propose a polynomial-time preprocessor as shown in Figure 1, which could identify unembeddable problems by treewidth estimation, thus significantly reducing the embedding trial time for arbitrary problems. The embedding trial time cannot be ignored in time-sensitive problems. Figure 2 shows simulations on a set of random graphs, performed in MATLAB 2015b on Intel 4960x with 32GB DDR3 RAM running Ubuntu 14.04 without parallelization. The embedding API is provided by D-Wave. The testing environment in the following sections remains the same, unless otherwise specified. Note that Figure 2 only intends to show a typical process where embeddability is unknown, i.e. the order of the test input is at least larger than the treewidth of the D-Wave architecture. The computing time heavily depends on computing hardware, test set, and order of both the problem graph and the hardware graph; however, the time/cost breakdown would remain roughly the same for random graphs of arbitrary order—namely, the time spent on embedding trials using heuristics is much more significant than the actual annealing time. The latter is especially true if the problem graph is not minor embeddable in the Chimera architecture graph, as it may take many unsuccessful trial runs before it is "declared" that the graph is not embeddable. Clearly, a quick way to rule out

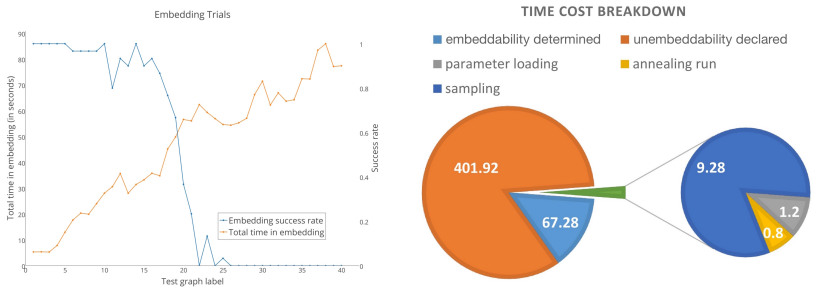

Fig. 2 Experimental results on wall-clock running time for the embedding heuristic. A set of 40 problems all with 40 nodes, generated in accordance with the Erdös-Rényi model, is tested. On the left plot, in order to test success probability, the embedding trials do not stop if a successful embedding is found, performed in 40 embedding trials. On the right graph, to simulate the real scenario, a default 10-trial process is run and stops once an embedding is found and the graph is positively determined to be embeddable. On the other hand, if after a great many runs no embedding is found, the graph is declared unembeddable (although it might be embeddable if the trial missed the embedding). The numbers in the pie chart are in seconds. Parameter loading and sampling time is provided in [41]. Determining embeddability takes much shorter time since the process would stop once an embedding is found, and is usually found within the first few trials.

embeddability would alleviate this situation, and this is the main motivation for this work.

There exist several methods for both bound estimation and exact solution of treewidth, such as polynomial-time bound estimations based on vertex degree (lower bound) [35], triangulation (lower bound) [32], vertex deletion (lower bound) [34], edge contraction (lower bound) [33], greedy degree (upper bound) [51], Breadth First Search (BFS, upper bound) [52], and exponential-time exact algorithms including QuickBB [11] using the branch and bound method and TreewidthDP [36] using dynamic programming. Most of these algorithms have issues with either 1) occasionally large error or 2) impractical time complexity. Note that some heuristics give very tight upper or lower bounds, including greedy, which slightly outperforms our Ollivier-Ricci algorithm in term of accuracy for varying order graphs (see Fig. 6). However, to the best of our knowledge, there is no algorithm that gives direct estimations instead of bounds on the treewidth. In Section 4, we compare our methods with the best heuristics and exact algorithms to demonstrate that our result is applicable in most scenarios.

3 Ollivier-Ricci curvature and treewidth

3.1 Ollivier-Ricci curvature

All fast approximation to treewidth algorithms are improvements of either lower or upper bounds. Here, besides bringing some theoretical insights, we propose a novel method of direct approximation based on a geometrical-

topological invariant associated with a graph: specifically, we propose to develop the connection between the *curvature* of a graph and the *treewidth* of a graph.

The Ollivier-Ricci curvature, recently developed by Yann Ollivier [7,8] and further developed by a group led by Jürgen Jost [9], is a combinatorial measure of how negatively/positively curved a graph is. It is inspired by a coarse version of the Ricci curvature in differential geometry. The ***Ollivier-Ricci curvature*** is conveniently defined in terms of the transport capabilities of the graph:

$$\kappa(x,y) = 1 - \frac{W_1(m_x, m_y)}{d(x,y)} \in (-2, 1),$$

where

W_1 : 1st Wasserstein distance,
m_x : distribution of 1-step non-lazy random walk from node x,
$d(x,y)$: distance between nodes x and y (in general hop-distance).

In the above, the first Wasserstein distance W_1 is defined as

$$W_1(m_i, m_j) = \inf_{\xi^{ij}} \sum_{k,l \in N(i) \times N(j)} d(k,l)\xi^{ij}(k,l), \tag{6}$$

with the infimum being extended over all coupling measures ξ^{ij} defined on $N(i) \times N(j)$, where $N(i) = \{j \in V : ij \in E\}$ is the neighborhood of i. $\xi^{ij}(k,l)$ is sometimes referred to as the *transference plan,* that is, it specifies the mass quantity initially at $k \in N(i)$ to be transferred to $l \in N(j)$. Formally,

$$\sum_{\ell \in N(j)} \xi^{ij}(k,\ell) = m_i(k), \qquad \sum_{k \in N(i)} \xi^{ij}(k,\ell) = m_j(\ell). \tag{7}$$

The reader is referred to [47,48] for a review of the Wasserstein distance and optimal transport. A more intuitive explanation of the Ollivier-Ricci curvature is shown in Figure 3. The Ollivier-Ricci curvature, though a local measure, can be given a global significance by averaging it over all edges, since there is currently no Ollivier-Ricci counterpart of the Gauss-Bonnet theorem.

The local aspect of the Ollivier-Ricci curvature makes it of relevance in anticipating how the Heat Diffusion protocol (or the Back Pressure protocol for that manner) will perform on a network, as both protocols use local information to forward packets, as described in more detail in [15,16].

At this stage, there is no theoretical proof that the Ollivier-Ricci curvature of a graph is related to its "tree-likeness." However, it is the main purpose of this paper to show that a strong correlation exists and develop a proof that a lower bound on treewidth is an increasing function of curvature. Clear agreement between curvature and treewidth at the limit of their ranges of values is shown in Table 1.

To better visualize the bigger picture of network topology and embeddability, the relationship among the most researched topics in network topology

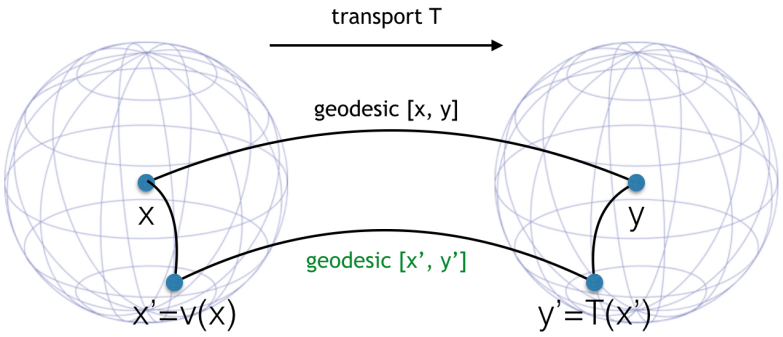

Fig. 3 Concept of optimal transport and curvature, where a mass distributed over a ball B_x in a Riemannian manifold is being transported to the ball centered at y. The comparison between (i) the piecemeal process of transporting every single mass element by parallel transport of xx' along the geodesic $[x, y]$ and (ii) the process of lumping the total mass of B_x at its center and transporting it in one move to the center of the other ball for redistribution across the ball is quantified by the fraction $\frac{\int_{B_x} d(x', T(x'))d\text{vol}(x')}{d(x,y)\text{vol}(B_x)}$. This measure is greater than 1 if and only if $\kappa_{\text{Ricci}}[x, y] < 0$, or less than 1 if and only if $\kappa_{\text{Ricci}}[x, y] > 0$.

Table 1 Ollivier-Ricci curvature versus treewidth agreement (n denotes the order of the graph)

	treewidth	Ollivier-Ricci curvature
range of possible values	$[1, n-1]$	$(-2, 1)$
lower bound reached for	tree	tree
upper bound reached for	complete graph	complete graph

is provided in Figure 4. There exist weak links between Gromov δ and embeddability [42], Gromov δ and treewidth [43], treewidth and treelength [44]; and strong links between treelength and Gromov δ [12], treewidth and graph minor and thus embeddability as discussed before. We will show strong correlation, if not a strong theoretical link, between Ollivier-Ricci curvature and treewidth; thus by calculating curvature using a linear programming interior point method, one would estimate treewidth in polynomial time.

As an aside, note that the Ollivier-Ricci curvature has been used to anticipate congestion in a wireless network under the specific Heat Diffusion communication protocol [15]. The theoretical fact that underpins this observation is that the Ricci curvature regulates the flow of heat on a Riemannian manifold, in somewhat the same way that the sectional curvature regulates geodesics. It has also seen recent applications including formulating Hamiltonian Monte Carlo [37], Internet topology [40], modeling robustness of cancer networks [38], and analyzing market fragility and systemic risk [39].

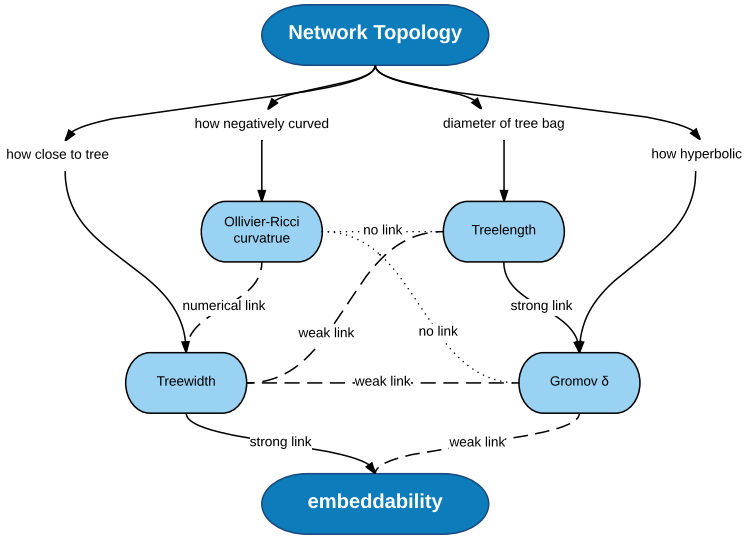

Fig. 4 The conceptual connection among four related invariants in graph theory. We provide strong numerical connections between Ollivier-Ricci curvature and treewidth in random arbitrary graphs; on the other hand, treewidth connects to graph minors, and hence curvature could determine unembeddability. It is worth noting that for engineering purposes, among the four topological invariants, only the Ollivier-Ricci curvature has polynomial-time complexity.

3.2 Ollivier-Ricci curvature computation

Clearly, (6)-(7) together with $\xi^{ij}(k,\ell) \geq 0$ make a linear programming problem. The most important motivation for the Ollivier-Ricci approach to graph curvature is that it is computable in polynomial-time via an Interior Point Method, while determining treewidth of a graph is NP-hard (shown in section 4). Note that in the actual computation throughout this paper, a more dedicated algorithm described in Rubner [50] is implemented in practice. The first Wasserstein distance in Eq. (6) is essentially Earth Mover's Distance (EMD) and has been proved to have polynomial-time complexity [50].

3.3 Proof of lower bound

We show that the Ollivier-Ricci curvature serves as a lower bound for the treewidth of an arbitrary graph G. This result is obtained by combining two known results: one that relates the lower bound on the spectrum of the graph Laplacian to the treewidth [45], and one that relates the spectrum lower bound to the Ollivier-Ricci curvature [46].

Proof It is known that

$$\mathrm{tw}(G) \geqslant \lfloor \frac{3n}{4} \frac{\lambda_1}{\Delta + 2\lambda_1} \rfloor - 1 =: f(\lambda_1), \tag{8}$$

where n and Δ denote the order and the maximum degree, resp., of the graph and λ_1 denotes the second smallest eigenvalue of the graph Laplacian. The other result to be utilized is

$$1-(1-k[t])^{\frac{1}{t}} \leq \lambda_1 \leq ... \leq \lambda_{N-1} \leq 1+(1-k[t])^{\frac{1}{t}}, \tag{9}$$

where $k[t]$ denotes the lower bound on the Ollivier-Ricci curvature of the t-fold neighborhood graph as defined in [46]. By convention, $t = 1$ corresponds to the original graph, so that $k[1] = k$ is the lower bound on the Ollivier-Ricci curvature of the original graph. It is easily seen that $f(\cdot)$ as defined by (8) is a monotonically increasing hyperbola in λ_1. Hence for $k \leq \lambda_1$, we have $f(\lambda_1) \geq f(k)$ and therefore

$$\mathrm{tw}(G) \geq \lfloor \frac{3n}{4} \frac{k}{\Delta + 2k} \rfloor - 1. \tag{10}$$

For the case considered in Fig. 5, left, this lower bound is not very useful as it is negative for some k's. However, its relevance lies in an important trend it reveals: Clearly, as the lower bound k on the Ollivier-Ricci curvature increases, the lower bound of the treewidth $f(k)$ increases.

Bauer *et al.* [46] showed that as t increases, the lower bound on λ_1 in (9) always gets nontrivially tighter. Therefore, a tighter bound would be

$$\mathrm{tw}(G) \geq \lfloor \frac{3n}{4} \frac{1-(1-k[t])^{\frac{1}{t}}}{\Delta + 2(1-(1-k[t])^{\frac{1}{t}})} \rfloor - 1. \tag{11}$$

Note that generating neighborhood graphs significantly increases computational cost. One could, however, construct $G[t]$ by simulating the random walk on the original graph G and by adding edge xy if the walk starting at $x \in V(G)$ reaches $y \in V(G)$ after t steps (see [46, Secs. 2.1 and 2.2]).

Even though (11) is an improvement over (10), it could still go negative for some networks. The reason is that, even though the bound (9) could be tight, the treewidth bound (8) could be very conservative and could even be negative. This especially happens when the maximum degree of the graph is high, as it is the case for scale-free graphs that have heavy tailed degree distribution. When the right-hand side of Equation (11) becomes negative, the bound unfortunately becomes useless in the preprocessor of Fig 1 for treewidth computation.

If, on the other hand, invoking Bauer's claim of tightness we assume that

$$\mathrm{tw}(G) \geq [\mathrm{RHS}(11)]_{\mathrm{problem}} \geq [\mathrm{RHS}(11)]_{\mathrm{architecture}},$$

then we may be able to conclude that

$$\mathrm{tw}(G) \geq \mathrm{tw}(\mathrm{architecture}),$$

in which case *we can positively conclude that G is not embeddable in the architecture while the heuristics would attempt to construct an embedding when there is no hope for.* This improved procedure is not considered here, as our results indicate that $t = 1$ already yields good results. An assessment of the extra effort needed to compute $k[t]$ versus the benefit of a tighter bound is left for further research.

4 Experimental results

We set up our simulation experiments on the hardware discussed in section 2. In the following, we examine three evaluation criteria of our curvature-based estimate of the treewidth: namely, **1)** the correlation between treewidth and Ollivier-Ricci curvature, **2)** the accuracy of the estimate, and **3)** the wall-clock time cost of the estimate compared to other estimation methods and the exact solution.

4.1 Correlation

In Figure 5, we show a strong correlation between Ollivier-Ricci curvature and treewidth of Erdös-Rényi random graphs. The curvature of the graphs is controlled by the probability of connection parameter in the Erdös-Rényi generator. In general, as the probability increases so does the curvature. We performed the experiment on two sets of graphs: those of the same order and those with varying order. Since for varying order, the treewidth could go as high as $|V(G)| - 1$, a normalization is needed. We propose

$$\mathrm{tw}_{\mathrm{norm}}(G) = \frac{\mathrm{tw}(G)}{|V(G)|}. \tag{12}$$

This is the normalization that was used in the simulation. Fig. 5 reveals a linear estimate of treewidth of the form $\mathrm{tw}(G) \approx a\kappa(G) + c$ where $\kappa(G)$ denotes the curvature of the graph. Even though Fig. 5 is restricted to Erdös-Rényi graphs, it is shown in our previous work [49] that data points of trees and scale-free graphs lie on, or close to, the same interpolation line. (More accurate results could be obtained by polynomial interpolation.)

4.2 Accuracy

We perform a linear interpolation-based treewidth estimation on randomly generated graphs of nontrivial treewidth computation. Since the exact solution has exponential complexity, the size of the test graphs is taken to be small enough to permit exact calculation. We estimate the treewidth based on several classical heuristics discussed in section 2 in addition to the nonclassical curvature estimate. We compare the results with the exact solution, as shown in Figure 6, where the exact solution is normalized to 1. Under such normalization, any other returned treewidth estimate is called *accuracy index*, with the idea that the closer the accuracy index is to 1, the more accurate the heuristic or Ollivier-Ricci curvature estimate is.

Out of several heuristics for upper and lower bound computation discussed in the previous section, we choose BFS and Greedy for upper bound estimation and Min Degree for lower bound estimation.

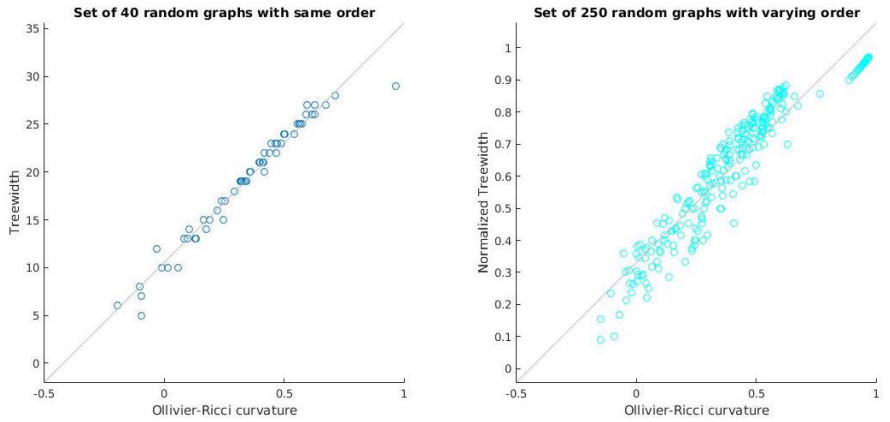

Fig. 5 Scatter plots of Ollivier-Ricci curvature versus treewidth (coefficient of determination R^2: 0.965 (left) and 0.889 (right)) The left plot is constructed on a set of 40 random graphs using the Erdös-Rényi random graph model with increasing probability of connection (hence increasing curvature). The right plot is constructed on a set of 250 random graphs, with the order of the graphs varying from 10 to 34. The treewidth of the right graphs is normalized by the order of the respective graphs; moreover, each order case contains random graphs with different probabilities of connection.

It turns out that the curvature estimate is of statistically best accuracy on sample graph sets of fixed order but with varying connectivity, with the greedy algorithm coming next (left panel of Fig. 6). Note, however, that on the varying order test set (right panel of Fig. 6), the accuracy of the curvature estimate is not as good as that of the greedy algorithm, but still better than the remaining two algorithms. This may be improved by a more carefully thought of definition of "normalized treewidth" or by higher order polynomial interpolation.

Student's t-test is utilized to test the significance of the observed Ollivier-Ricci curvature accuracy index being closer to the ideal value of 1 than the other estimates. The t-test would assume that the empirical distribution of the accuracy indexes returned by the various trials of the various methods (Greedy, Ollivier-Ricci, BFS, Min Degree) is Gaussian. It should be noted that the Greedy and BFS algorithms give only upper bounds on the accuracy index, while the Min Degree algorithm gives a lower bound. Consequently, the distribution of the accuracy indexes for the Greedy, Min Degree and BFS estimates is skewed and cannot be Gaussian. So, the t-test cannot be applied "as is" to prove that the various means differ. However, on theoretical grounds, the mean of the Min Degree index is less than 1, while the mean of the Greedy and BFS indexes is larger than 1. On the other hand, the sample distribution of the accuracy indexes returned by the Ollivier-Ricci curvature is symmetric, and appears to be Gaussian with mean 1. So, the only statistical test to be conducted is a t-test on the mean of the Ollivier-Ricci curvature to be 1. We used the Matlab t-test implementation with the null hypothesis that the mean of Ollivier-Ricci curvature is equal to 1 and the t-test does not reject the null

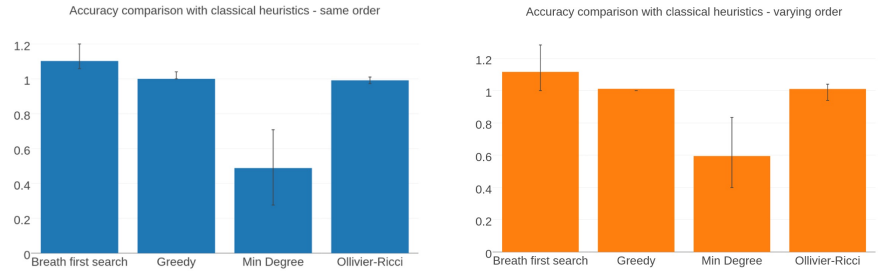

Fig. 6 Accuracy analysis of Ollivier-Ricci curvature estimation and other well-known classical heuristics. Accuracy is estimated relative to the exact solution, so that the exact solution has accuracy index of 1. Bars show $25th$ and $75th$ percentile of the estimation. The left graph shows the accuracy index of 60 graphs of the same order but with varying topology. The right plot shows the accuracy index for graphs with similar topology but with order varying from 10 to 40. Accuracy larger than one means that estimated treewidth is larger than actual treewidth. LibTW library [6] is used throughout the computation.

hypothesis with a *p-value* of 0.3979 for the fixed order case and 0.1415 for the varying order case.

4.3 Time cost

The classical treewidth heuristics used in accuracy comparison all have polynomial-time complexity and run smoothly in experiments; thus, their speed is not assessed in this section. We would mainly compare the curvature estimate with two best known exact treewidth algorithms: QuickBB and TreewidthDP. The former, discussed in detail in [36], treats the treewidth problem as a linear ordering problem, and finds a perfect elimination scheme of such ordering. The latter, discussed in detail in [11], performs a search over permutations of the vertices. TreewidthDP runs much more slowly in general than the latter; however, for memory constrained systems, QuickBB would utilize address space in a better way. The comparison is shown in Figure 7. As one would note from the left figure, curvature estimation takes more time as curvature and treewidth grow larger along the abscissa. Note that the time complexity of QuickBB scales as $O(n^{n-\mathrm{tw}(G)})$ [11, Sec. 8]. Therefore, QuickBB favors graphs of higher treewidth while curvature estimation favors graphs of lower treewidth. One may utilize this fact to develop "smart algorithms" that efficiently predetermine graph connectivity and decide which algorithm to choose. From the right plot of Figure 7, it follows that TreewidthDP scale exponentially with the order of the graph in general, and the curvature estimation appears subexponential as the curve is not quite a line. Note that it is not possible in practice to scale the problem to larger size, as the time cost for TreewidthDP is already approaching practical limits.

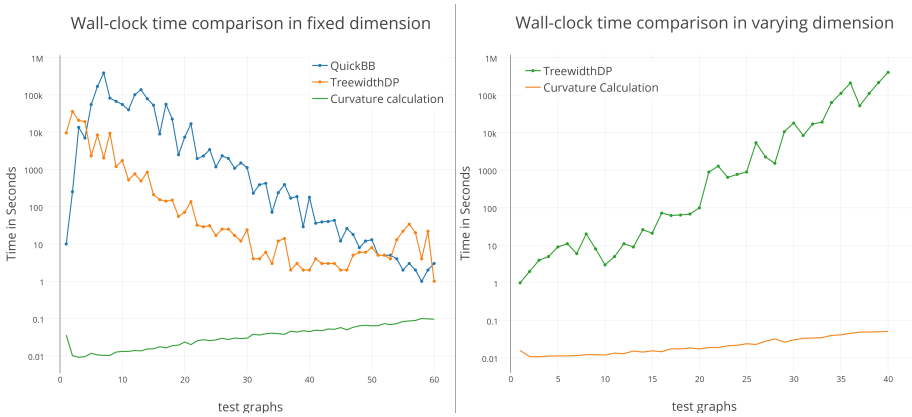

Fig. 7 Wall-clock time cost of classical exact treewidth algorithms and non classical curvature estimate. On the left panel, the abscissa is consistent with increasing treewidth on a set of 60 graphs with the same order. On the right panel, the abscissa is the order of the graph in a set of graphs with similar topological characteristics in the sense of normalized treewidth. Both simulations are performed in the same environment as before. No parallelization is added. Note that the time is in log scale.

5 Conclusion and future work

In conclusion, we reviewed the quantum adiabatic implementation of the QUBO problem—more specifically, the crucial minor embedding step that allows it to be solved on, say, the D-Wave quantum computer architecture. We proposed a fast estimation algorithm of the treewidth, the exact computation of which is NP-hard, based on the computationally less demanding differential-geometric Ollivier-Ricci curvature. Even though we derived analytically a lower bound on treewidth based on a lower bound of the curvature, this leaves open the theoretical problem of proving the existence of error bounds between the exact Ollivier-Ricci curvature and the exact treewidth of a graph.

It is also important to follow up on the impact of curvature-based treewidth estimation on real-world problems, as the D-Wave chip is being applied to more and more optimization problems. It is of interest to properly define a *time-sensitive,* time-varying QUBO problem that could benefit from fast treewidth estimation, thus possibly leading to a general quantum speedup in applications. Sensor coverage and, more specifically, optimization subject to interference constraints in wireless networking [30] would be potential problems of this nature.

An broader theoretical spinoff of this work, closely related to the treewidth versus treelength issue displayed in Figure 4, is the discrepancy between congestion in wireline networks under least length path routing [10] and congestion in wireless networks under the Heat Diffusion protocol [15]. Indeed, the former occurs over Gromov hyperbolic networks [10], while the latter occurs over Ollivier-Ricci negatively curved networks. Observing that the proof of congestion in Gromov-hyperbolic networks [10] *does not* rely on a "fattened

tree" interpretation of the Gromov property further corroborates our claim of Gromov-hyperbolicity versus tree-likeness discrepancy [49].

Acknowledgment

This research was supported by the Army Research Office (ARO) Multi University Research Initiative (MURI) contract W911NF-11-1-0268 and by National Science Foundation (NSF) grant CCF-1423624.

References

1. P. F. Felzenszwalb, *Dynamic programming and graph algorithms in computer vision,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 33, Issue 4, pp. 721-740, 2011.
2. Troels F. Rnnow et al, *Defining and detecting quantum speedup,* arXiv:1401.2910.
3. V. Choi, *Minor-embedding in adiabatic quantum computation: I. The parameter setting problem*, Quantum Information Processing, Volume 7, pp. 193–209, 2008.
4. V. Choi, *Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design*, Quantum Information Processing: Volume 10, Issue 3, pp. 343-353, 2011.
5. Ch. Klymko, B. D. Sullivan, T. S. Humble, *Adiabatic quantum programming: Minor embedding with hard faults*, arXiv:1210.8395 [quant-ph].
6. Th. van Dijk, J-P van den Heuvel, W. Slob, *Computing treewidth with LibTW.* (`http://www.treewidth.com/docs/LibTW.pdf`)
7. Y. Ollivier, *Ricci curvature of Markov chains on metric spaces*, arXiv:math/0701886 [math.PR].
8. Y. Ollivier, *A survey of Ricci curvature for metric spaces and Markov chains*, Probabilistic Approach to Geometry, 343-381, Adv. Stud. Pure Math., 57, Math. Soc. Japan, Tokyo, 2010.
9. J. Jost, S. Liu, *Ollivier's Ricci curvature, local clustering and curvature dimension inequalities on graphs*, arXiv:1103.4037 [math.CO].
10. E. Jonckheere, M. Lou, F. Bonahon, Y. Bryshnikov *Euclidean versus hyperbolic congestion in idealized versus experimental networks*, Internet Mathematics, Vol. 7, No.1, pp. 1-27, March 2011.
11. V. Gogate and R. Dechter, *A complete anytime algorithm for treewidth*, 20th Conference on Uncertainty in Artificial Intelligence (UAI), 2004.
12. V. Chepoi, F. Dragan, B. Estellon, M. Habib, Y. Vaxs, *Diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs.* In Proceedings of the Twenty-Fourth Annual Symposium on Computational Geometry (pp. 59-68). ACM.
13. H. L. Bodlaender, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM Journal on Computing, Vol. 25, Issue 6, pp. 13051317, 1996.
14. `http://strategic.mit.edu/downloads.php?page=matlab_networks.`
15. C. Wang, E. Jonckheere, R. Banirazi, *Wireless network capacity versus Ollivier-Ricci curvature under Heat Diffusion (HD) protocol,* American Control Conference (ACC), Portland, OR, June 04-06, 2014. (Available at `http://eudoxus2.usc.edu.`)
16. C. Wang, E. Jonckheere, R. Banirazi, *Interference constrained network performance control based on curvature control*, American Control Conference (ACC), Boston, MA, July 06-08, 2016, to appear. (Available at `http://eudoxus2.usc.edu.`)
17. E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, *Quantum Computation by Adiabatic Evolution*, arXiv:quant-ph/0001106.
18. I. Trummer, C. Koch, *Multiple Query Optimization on the D-Wave 2X Adiabatic Quantum Computer*, arXiv: 1501.06437.
19. P.I. Bunyk, E. Hoskinson, M.W. Johnson, E. Tolkacheva, F. Altomare, A.J. Berkley, R. Harris, J.P. Hilton, T. Lanting, J. Whittaker, *Architectural considerations in the design of a superconducting quantum annealing processor*, arXiv: 1401.5504.

20. K.J. Wu, *Solving practical problems with quantum computing hardware*, ASCR Workshop on Quantum Computing for Science, February, 2015.
21. H.L. Bodlaender, *A linear-time algorithm for finding tree-decompositions of small treewidth*. SIAM J. Comput. 25, 6 (1996), 1305-1317.
22. H.L. Bodlaender, A.M.C.A. Koster, *Treewidth Computation II. Lower Bounds*, Technical Report UU-CS-2010-22, Dept. of Information and Computing Sciences, Utrecht University (2010).
23. T. Kloks, *Treewidth: computations and approximations*. No. 842 in Lecture Notes in Computer Science. Springer Verlag, 1994.
24. N. Robertson, P. Seymour, *Graph minors - XIII: The disjoint paths problem*. J. Comb. Theory, Ser. B 63, 1 (1995), 65-110.
25. J. Cai, W.G. Macready, A. Roy, *A practical heuristic for finding graph minors*, arXiv:1406.2741, Jun. 2014.
26. E.G. Rieffel, D. Venturelli, B. O'Gorman, M.B. Do, E. Prystay, V.N. Smelyanskiy, *A case study in programming a quantum annealer for hard operational planning problems*, Quantum Information Processing, Vol. 14, Issue 1, pp 1-36, Jan. 2015.
27. B. O'Gorman, R. Babbush, A. Perdomo-Ortiz, A. Aspuru-Guzik, V. Smelyanskiy, *Bayesian network structure learning using quantum annealing*, The European Physical Journal Special Topics, Vol. 224, Issue 1, pp 163-188, Feb. 2015.
28. A. Perdomo-Ortiz, J. Fluegemann, S. Narasimhan, R. Biswas, V.N. Smelyanskiy, *A quantum annealing approach for fault detection and diagnosis of graph-based systems*, The European Physical Journal Special topics, Vol. 224, Issue 1, pp 131-148, Feb. 2015.
29. K.M. Zick, O. Shehab, M. French, *Experimental quantum annealing: Case study involving the graph isomorphism problem*, Scientific Reports, DOI: 10.1038/srep11168, Jun. 2015.
30. C. Wang, H. Chen, E. Jonckheere, *Quantum versus simulated annealing in wireless interference network optimization*, Scientific Reports, DOI: 10.1038/srep25797, May. 2016
31. I. Adler, F. Dorn, F.V. Fomin, I. Sau, D.M. Thilikos, *Faster parameterized algorithms for minor containment*. Theor. Comput. Sci, 412(50):7018-7028, 2011.
32. A. Berry, J.R.S. Blair, P. Heggernes, *Maximum cardinality search for computing minimal triangulations*, 28th International Workshop on Graph-Theoretic Concepts in Computer Science, pp 1-2, London, UK, 2002, Springer-Verlag.
33. V. Gogate, R. Dechter, *A complete anytime algorithm for treewidth*, Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pp 201-208, Arlington, Virginia, 2004. AUAI Press.
34. A.M.C.A. Koster, H.L. Bodlaender, S.P.M. van Hoesel. *Treewidth: computational experiments*. Research Memoranda 001, Maastricht: METEOR, 2002.
35. S. Ramachandramurthi, *The structure and number of obstructions to treewidth*, SIAM Journal on Discrete Mathematics, 10(1):146-157, 1997.
36. H.L. Bodlaender, F.V. Fomin, A.M.C.A. Koster, *On exact algorithms for treewidth*, Technical Report UU-CS-2006-032, Institute of Information and Computing Sciences, Utrecht University, 2006.
37. S. Holmes, S. Rubinstein-Salzedo, C. Seiler, *Curvature and concentration of Hamiltonian Monte Carlo in high dimensions*, arXiv:1407.1114, May. 2015.
38. A. Tannenbaum, C. Sander, R. Sandhu, L. Zhu, I. Kolesov, E. Reznik, Y. Senbabaoglu, T. Georgiou, *Graph curvature and the robustness of cancer networks*, Scientific Reports, doi:10.1038/srep12323, Jul. 2015.
39. R. Sandhu, T. Georgiou, A. Tannenbaum, *Market fragility, systemic risk, and Ricci curvature*, arXiv: 1505.05182, May 2015.
40. C.C. Ni, Y.Y. Lin, J. Gao, X.D. Gu, E. Saucan, *Ricci curvature of the Internet topology*, INFOCOM 2015, Hong Kong, Apr. 2015.
41. A.D. King, C.C. McGeoch, *Algorithm engineering for a quantum annealing platform*, arXiv: 1410.2628, Oct. 2014.
42. W. Carballosa, J. M. Rodrguez, O. Rosario, J. M. Sigarreta, *Gromov hyperbolicity of minor graphs*, arXiv:1506.06047, Jun. 2015.
43. F. Montgolfier, M. Soto, L. Viennot, *Treewidth and hyperbolicity of the Internet*, DOI: 10.1109/NCA.2011.11, Network Computing and Applications (NCA), 10th IEEE International Symposium on, 2011.

44. D. Lokshtanov, *On the complexity of computing treelength*, Mathematical Foundations of Computer Science, 2007, Vol. 4708, pp 276-287.
45. L.S. Chandrana, C.R. Subramanian, *A spectral lower bound for the treewidth of a graph and its consequences*, Information Processing Letters, Vol. 87, Issue 4, 31 Aug. 2003, pp 195200.
46. F. Bauer, J. Jost, S. Liu, *Ollivier-Ricci curvature and the spectrum of the normalized graph Laplace operator*, Math. Res. Lett. 19 (2012), No. 6, 1185-1205.
47. C. Villani, *Optimal transport, old and new*, ISBN 978-3-540-71050-9, Springer, 2009.
48. C. Villani, *Topics in Optimal Transportation*, ISBN 0-8218-3312, Graduate Studies in Mathematics, Volume: 58, 2003.
49. C. Wang, E. Jonckheere, T. Brun, *Ollivier-Ricci curvature and fast approximation to tree-width in embeddability of QUBO Problems,* International Symposium on Communications, Control, and Signal Processing (ISCCSP), Athens, Greece, May 21-23, 2014, pp. 639-642.
50. Y. Rubner, C. Tomasi, and L. J. Guibas. *A Metric for Distributions with Applications to Image Databases*, Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India, January 1998, pp. 59-66.
51. H.M. Markowitz, *The elimination form of the inverse and its application to linear programming*, Manage. Sci. 3, 255-269, 1957.
52. D.J. Rose, R.E. Tarjan, G.S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput. 5, 266-283, 1976.