Approximability of Dense and Sparse Instances of Minimum 2-Connectivity, TSP and Path Problems

Béla Csaba* Marek Karpinski† Piotr Krysta‡

Abstract

We study the approximability of dense and sparse instances of the following problems: the minimum 2-edge-connected (2-EC) and 2-vertex-connected (2-VC) spanning subgraph, metric TSP with distances 1 and 2 (TSP(1,2)), maximum path packing, and the longest path (cycle) problems. The approximability of dense instances of these problems was left open in Arora *et al.* [3]. We characterize the approximability of all these problems by proving tight upper (approximation algorithms) and lower bounds (inapproximability). We prove that 2-EC, 2-VC and TSP(1,2) are Max SNP-hard even on 3-regular graphs, and provide explicit hardness constants, under $\mathsf{P} \neq \mathsf{NP}$. We also improve the approximation ratio for 2-EC and 2-VC on graphs with maximum degree 3. These are the first explicit hardness results on sparse and dense graphs for these problems. We apply our results to prove bounds on the integrality gaps of LP relaxations for dense and sparse 2-EC and TSP(1,2) problems, related to the famous metric TSP conjecture, due to Goemans [18].

1 Introduction

Recently, considerable efforts of researchers were put into approximating optimization problems on special instances. It turned out that even when one restricts the input, most of the known problems still remain NP-hard. Of particular interest are the problems on dense and sparse instances, see recent surveys [23, 24] for references. We study the following fundamental combinatorial optimization problems.

2-EC: 2-edge-connected (or 2-EC) spanning subgraph problem, where given a 2-EC graph, the goal is to find a 2-EC spanning subgraph with minimum number of edges. A graph is 2-EC if for any pair of its vertices, there are at least two edge-disjoint paths between them.

2-VC: 2-vertex-connected (2-VC) spanning subgraph problem, where the definition is analogous to 2-EC, but the paths are assumed to be internally vertex-disjoint.

TSP(1,2): The traveling salesman problem on a complete graph with weight 1 or 2 on each edge. For an instance of TSP(1,2) problem, the graph induced by all weight 1 edges is called the input graph.

Longest path problem: Given a graph, find a simple path with maximum number of edges.

Path packing problem: Given a graph, find a set of vertex disjoint paths such that the number of edges in all the paths is maximized. Single vertices are treated as paths with zero edges.

We study the approximability of these problems on dense and sparse graphs. Dense instances of the above problems constitute a list of problems that were left open from the approximability point of view, in the paper of Arora *et al.* [3] (see also [14]). In this paper Arora *et al.* show a general technique that provides polynomial time approximation schemes for dense instances of many optimization problems. It seems that one cannot use their methods to give better approximation ratios to the above problems, mostly due to a non-local nature of these problems. We resolve the problem of approximability of dense versions of all these problems by proving tight upper (approximation algorithms[1]) and lower bounds (inapproximability).

Dual instances to dense give sparse graphs. Many graph optimization problems are Max SNP-hard[2] already on very restricted sparse instances – graphs with maximum degree bounded by a constant. Examples are the vertex cover, maximum independent set and max-

*Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, D-66123 Saarbrücken, Germany. E-mail: csaba@mpi-sb.mpg.de. Partially supported by the IST Program of the EU under contract number IST-1999-14186 (ALCOM-FT).

†Department of Computer Science, University of Bonn, Römerstrasse 164, D-53117 Bonn, Germany. E-mail: marek@cs.uni-bonn.de. Supported in part by DFG grant, DIMACS, and IST grant 14036 (RAND-APX).

‡Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85 D-66123 Saarbrücken, Germany. E-mail: krysta@mpi-sb.mpg.de. Partially supported by the DFG project no. 1126 on "Algorithmik großer und komplexer Netzwerke," and by the IST Program of the EU under contract number IST-1999-14186 (ALCOM-FT).

[1] For a minimization (maximization, resp.) problem, a polynomial time algorithm is called an α-*approximation algorithm*, if it finds a solution of cost at most (at least, resp.) α times the cost of an optimal solution. α is called an *approximation ratio (factor)*, and the problem is said to be *approximable within* α.

[2] Max SNP-hardness implies that the problem cannot be approximated in polynomial time within constant ratios that are arbitrarily close to 1, unless $\mathsf{P} = \mathsf{NP}$.

imum cut problems on maximum degree 3 graphs, see [24]. We give the first known explicit hardness factors for the 2-EC, 2-VC and TSP(1,2) problems on maximum degree 3 graphs, and on 3-regular graphs.

Suppose we are given a graph G with n vertices, and the minimum degree at least cn, $c \in [0,1]$ is a fixed constant. We call such a graph *c-dense*. The classical theorem of Dirac [20] says that if $c \geq \frac{1}{2}$, then G has a Hamilton cycle which can be found in polynomial time (even in the NC class [9]). Observe that the Hamilton cycle constitutes an optimal solution to all of the considered problems. That is why we assume $c < \frac{1}{2}$.

Previous results.

2-EC & 2-VC: This is the simplest non-trivial version of the connectivity problem and has been studied for a long time, but tight approximation guarantees and inapproximability results are not fully understood yet [7, 17, 25, 33, 29, 13, 8]. For 2-EC, Khuller and Vishkin [25] gave a $\frac{3}{2}$-approximation, improved by Cheriyan *et al.* [7] to $\frac{17}{12}$, and to $\frac{4}{3}$ by Vempala and Vetta [33]. The best known result, due to Krysta and Kumar [29], is a $(\frac{4}{3} - \frac{1}{1344})$-approximation. For 2-VC, Khuller and Vishkin [25] gave a $\frac{5}{3}$-approximation, improved to $\frac{3}{2}$ by Garg *et al.* [17], and to $\frac{4}{3}$ by Vempala and Vetta [33].

Both 2-VC and 2-EC problems are NP-hard even on 3-regular planar graphs. Fernandes [13] proved Max SNP-hardness on arbitrary graphs; Czumaj & Lingas [8] show Max SNP-hardness on bounded degree 6 graphs. These results do not give explicit hardness constants.[3]

TSP(1,2): For this version of the TSP, Karp has shown NP-completeness in his seminal paper [22]. Papadimitriou and Yannakakis [31] prove Max SNP-hardness of this problem, when the input graph has maximum degree 6. They also show a $\frac{7}{6}$-approximation algorithm for TSP(1,2). The first explicit hardness factor for TSP(1,2) was 5381/5380, due to Engebretsen [11]. This was improved to 743/742 by Engebretsen & Karpinski [12]. Let us fix any $c \in (0, 1/2)$. In [14], de la Vega & Karpinski show, that TSP(1,2) is Max SNP-hard, when the input graph is c-dense (implicit in their work is a parametrization of the hardness factor by c).

Longest path problem: Given a graph, let n be the number of its vertices. Karger *et al.* [21] have given a polynomial algorithm that finds a path of length $\Omega(\log n)$ in a 1-tough graph, i.e. an $\Omega(\frac{\log n}{n})$-approximation (Hamiltonian graphs are also 1-tough). Alon *et al.* [2] give a polynomial algorithm that for any constant $p > 0$, finds a path of length $p \log n$, if there is such a path. Vishwanathan [35] has improved this bound for Hamiltonian graphs, by showing a $\Omega(\frac{(\log n)^2}{n(\log \log n)^2})$-approximation. The problem is very hard, as it has no constant approximation for *any* constant, unless P = NP, even for graphs with maximum degree 4 [21]. Bazgan *et al.* [4] have proved the same hardness result on 3-regular Hamiltonian graphs. The same holds for the longest cycle problem.

For dense graphs, Karger *et al.* [21] gave a polynomial algorithm that finds a path of length at least m/n in a graph with n vertices and m edges. This is a $\frac{c}{2}$-approximation algorithm for the longest path problem on c-dense graphs. F. de la Vega & Karpinski [14] prove, that the problem is Max SNP-hard on c-dense instances, for any fixed $c \in (0, 1/2)$.

Path packing problem: This problem finds many applications, see [34]. Vishwanathan [34] shows that from the approximation view point, the problem is equivalent to TSP(1,2). This, and the algorithm of Papadimitriou and Yannakakis [31] imply a $\frac{5}{6}$-approximation algorithm for the path packing problem (see [34] for details). The path packing problem is also Max SNP-hard [34].

Our contributions.

We give a new, general and uniform technique that provides approximation algorithms for dense instances of all the problems above. Our technique provides the first approximation algorithms for the mentioned problems that are parametrized with the density c, where the parametrization is tight. This means that in each case the approximation ratio approaches 1 when the density c approaches $\frac{1}{2}$, which by Dirac's result is the threshold above which the problems become polynomially solvable. Our algorithms and analyses rely on deep results from graph theory, for instance the Regularity Lemma of Szemerédi [32], the Blow-up Lemma of Komlós, Sárközy and Szemerédi [26], or a generalization of Dirac's Theorem due to Bollobás and Brightwell [20]. The algorithms are very efficient, as they can be implemented in parallel in the NC class (details omitted). For NC implementations of the Blow-up and Regularity lemmas, see [1, 27].

We prove that our parametrized approximation algorithms are close to best possible by showing explicit lower bounds (also parametrized by c) on the approximation ratios, under the usual P $\neq$ NP conjecture. These lower bounds show that there is an explosion of difficulty in approximating our problems when $c < \frac{1}{2}$.

We also prove the first explicit hardness ratios and improve the approximation ratios for some of these problems on graphs with maximum degree 3. The precise list of our results appears below. Let the input graph $G = (V, E)$ have minimum degree $\geq c|V|$, where $c \in [0, \frac{1}{2}]$ is any fixed constant, and $\beta > 0$ be any fixed

[3] In this paper, saying that a minimization problem is *hard to approximate* or *inapproximable* within a factor of f means that there is no $(f - \epsilon)$-approximation algorithm for any $\epsilon > 0$, unless P = NP. f is called a *hardness factor (or constant)*. Similarly for a maximization problem.

arbitrarily small constant, and $\varepsilon_0 = 1/742$.

2-EC & 2-VC: We give a $(2-2c+\beta)$-approximation algorithm for 2-EC and 2-VC problems on G. This improves on the $\frac{4}{3}$-approximations in [33], and on $(\frac{4}{3} - \frac{1}{1344})$-approximation in [29], for almost any $c > \frac{1}{3}$. We prove that the problems are Max SNP-hard for any fixed density $c \in (0, 1/2)$, and an explicit hardness factor is $1 + (\frac{1}{2} - c)\varepsilon_0$. If c tends to $\frac{1}{2}$, the algorithms are essentially 1-approximation, and the approximation and hardness factors are arbitrarily close to each other. This is also true for the other dense results. 2-EC and 2-VC are proved NP-hard to approximate within: 1573/1572 on maximum degree 3 graphs, and 2581/2580 on 3-regular graphs. We give a $(\frac{5}{4}+\epsilon)$-approximation for 2-EC and 2-VC on maximum degree 3 graphs. A $(\frac{21}{16}+\epsilon)$-approximation for 2-EC and 2-VC on such graphs was previously known [29]. To our knowledge, no results were known for dense 2-EC and 2-VC problems, except the ones on arbitrary graphs cited above. Max SNP-hardness on bounded degree 3 and on dense graphs was not known before. Our results significantly improve on Czumaj & Lingas [8], since they do not give explicit hardness ratios and their bound on degree is 6.

TSP(1,2): We give a $(2-2c+\beta)$-approximation, improving on the $\frac{7}{6}$-approximation of Papadimitriou and Yannakakis [31] when $c > \frac{5}{12}$. Our algorithm can be viewed as a generalization of the mentioned Dirac's theorem. We give a hardness factor of $1 + (1-2c)\varepsilon_0$ for any fixed $c \in (0, 1/2)$. If the input graph has maximum degree 3, we show a hardness of 787/786, and of 1291/1290 for 3-regular input graphs. This improves on the results of Engebretsen and Karpinski [12], since they need graphs of maximum degree 4.

Path packing problem: We show a $(2c-\beta)$-approximation, and a hardness factor of $1-(1-2c)\varepsilon_0$ on c-dense graphs. This improves on the $\frac{5}{6}$-approximation due to Papadimitriou and Yannakakis [31] and Vishwanathan [34] when $c > \frac{5}{12}$.

Longest path problem: We show a $(\frac{c}{1-c} - \beta)$-approximation algorithm, and a hardness factor of $1-(1-2c)\varepsilon_0$ for c-dense instances. This improves significantly on $\frac{c}{2}$-approximation algorithm of Karger *et al.* [21], for *all* values of c.

The Linear Programming (LP) relaxation for 2-EC problem and the subtour LP relaxation for TSP are closely related [7]. The integrality gap[4] of the LP relaxation for 2-EC is not well understood. The best known upper bound is $\frac{17}{12}$ [7]. It has been conjectured that the integrality gaps of both LPs are $\frac{4}{3}$. We give stronger bounds than $\frac{4}{3}$ for some versions of these conjectures for dense and sparse 2-EC and TSP(1,2).

Related work. The Regularity Lemma was used in a context of approximating dense problems by Frieze and Kannan [16] to speed-up some algorithms.

Organization of paper. Sec. 2: preliminaries; Sec. 3: the technique and algorithms for dense problems; Sec. 4: algorithms for bounded degree 2-EC & 2-VC; Sec. 5: applications to integrality gaps; Sec. 6: hardness results. Missing material is deferred to the full paper version.

2 Preliminaries

Given an (undirected) graph $G = (V, E)$, we write $V(G) = V$, $E(G) = E$, and $v(G) = |V|$, $e(G) = |E|$. The elements of V are *vertices*, and elements of E are *edges*. A closed path of length l is a cycle, denoted C_l, and a *simple* path means that the vertices are distinct. A $u-v$ *path* is a path with end vertices u, v. A vertex v is a *cut vertex* if its removal disconnects the graph. If v is a cut vertex of a graph G, and some two vertices x, y are in distinct components of $G \setminus v$, then v *separates* x and y. For a given non-empty set $S \subset V$, $(S, \bar{S})$ denotes an *edge cut*, i.e. the set of the edges in E with exactly one end vertex in S ($\bar{S} = V \setminus S$). An edge is a *bridge* if its removal disconnects the graph. $deg_G(v)$ denotes the degree of vertex v in G. Let $\delta(G)$ be the minimum degree of G. The *density* of graph G is $\delta(G)/|V(G)|$. If density $\geq c$, then G is *c-dense*.

An *ear decomposition* $\mathcal{E}$ of a graph G is a partition of the edge set into open or closed paths, $\mathcal{E} = \{Q_0, Q_1, \ldots, Q_k\}$, such that Q_0 is the trivial path with one vertex, and each Q_i ($i = 1, \ldots, k$) is a path that has both end vertices in $V_{i-1} = V(Q_0) \cup \cdots \cup V(Q_{i-1})$ but has no internal vertex in V_{i-1}. A (closed or open) *ear* means one of the (closed or open) paths $Q_0, Q_1, \ldots, Q_k$ in $\mathcal{E}$. In the ear decomposition $\mathcal{E} = \{Q_0, Q_1, \ldots, Q_k\}$, we say that ear Q_i is *earlier* than ear Q_j, and Q_j is *later* than Q_i, when $i < j$. Given a positive integer ℓ, *ℓ-ear* is an ear with ℓ edges. An ear decomposition $\{Q_0, Q_1, \ldots, Q_k\}$ is *open* if all ears $Q_2, \ldots, Q_k$ are open. If a graph is 2-vertex(edge)-connected, then we say it is 2-VC(EC). *opt*(G) or *opt* denotes the value of an optimal solution on G to the considered problem.

PROPOSITION 2.1. ([20]) *A graph is 2-EC iff it has an ear decomposition. Also, a graph is 2-VC iff it has an open ear decomposition. An (open) ear decomposition can be found in polynomial time.*

3 Approximation Technique on Dense Graphs

We use tools from the Extremal Graph Theory to give a technique for approximating dense problems. We first

[4] Definition of the LP relaxation of the unweighted 2-EC appears in Section 5. The *integrality gap* of the LP relaxation is defined as $\sup_I \frac{OPT_{\mathrm{INT}}(I)}{OPT_{\mathrm{LP}}(I)}$, where $OPT_{\mathrm{INT}}(I)$ is the value of an optimum integral solution on a problem instance I, and $OPT_{\mathrm{LP}}(I)$ is the value of an optimum LP solution on instance I.

give an overview based on [28] (see also Diestel [10]).

Regularity and Blow-up Lemmas. Let $G=(V,E)$ be a graph, and $deg(x,Y)$ be the number of neighbors of vertex $x\in V$ in set $Y\subseteq V$. Let $X,Y\subset V$, $X\cap Y=\emptyset$, then $e(X,Y)$ denotes the number of edges between X and Y. Let $G=(A,B,E)$ denote a bipartite graph with color classes A and B, and the set of edges E. For disjoint X,Y we define a *density* $d(X,Y)=\frac{e(X,Y)}{|X|\cdot|Y|}$. The density of a bipartite graph $G=(A,B,E)$ is $d(G)=d(A,B)=\frac{|E|}{|A|\cdot|B|}$. Given two graphs G and H, we say that G has a subgraph *isomorphic* to H, or H is *embeddable* into G if and only if there is a one-to-one map (injection) $\varphi: V(H)\longrightarrow V(G)$ s.t. for each (x,y), $(x,y)\in E(H)$ implies $(\varphi(x),\varphi(y))\in E(G)$.
Regularity Condition. Let $\varepsilon>0$. Given a graph $G=(V,E)$ and two disjoint sets $A,B\subset V$, we say that the pair (A,B) is ε*-regular* if for every $X\subseteq A$ and $Y\subseteq B$ such that $|X|>\varepsilon|A|$ and $|Y|>\varepsilon|B|$, we have $|d(X,Y)-d(A,B)|<\varepsilon$.

THEOREM 3.1. (REGULARITY LEMMA, [32, 28]) *For every $\varepsilon>0$, there is an $M=M(\varepsilon)$ such that if $G=(V,E)$ is any graph and $d\in[0,1]$ is any real number, then there is a partition of the vertex set V into $k+1$ clusters $V_0,V_1,\dots,V_k$, and there is a subgraph G' of G with the following properties:* **(i)** *$k\le M$, $|V_0|\le\varepsilon|V|$;* **(ii)** *all clusters V_i, $i\ge1$, are of the same size $m\le\lceil\varepsilon|V|\rceil$;* **(iii)** *$deg_{G'}(v)>deg_G(v)-(d+\varepsilon)|V|$ for all $v\in V$;* **(iv)** *$e(G'(V_i))=0$ for all $i\ge1$;* **(v)** all *pairs $G'(V_i,V_j)$ $(1\le i<j\le k)$ are ε-regular with density either 0 or greater than d.*

LEMMA 3.1. (FACT 1.3 IN [28]) *Let (A,B) be an ε-regular pair with density d. Then for any $Y\subseteq B$, with $|Y|>\varepsilon|B|$, we have $|\{x\in A: deg(x,Y)\le(d-\varepsilon)|Y|\}|\le\varepsilon|A|$.*

LEMMA 3.2. (FACT 1.5 IN [28]) *Let (A,B) be an ε-regular pair with density d, and, for some $\gamma>\varepsilon$, let $A'\subseteq A$, $|A'|\ge\gamma|A|$, $B'\subseteq B$, $|B'|\ge\gamma|B|$. Then (A',B') is an ε'-regular pair with $\varepsilon'=\max(\varepsilon/\gamma,2\varepsilon)$, and $|d(A',B')-d|<\varepsilon$.*

Super-Regularity Condition. Given a graph $G=(V,E)$ and $A,B\subset V$ $(A\cap B=\emptyset)$, we say that pair (A,B) is (ε,δ)*-super-regular* if for every $X\subseteq A$ and $Y\subseteq B$ s.t. $|X|>\varepsilon|A|$, $|Y|>\varepsilon|B|$, we have $d(X,Y)>\delta$, and $deg(a)>\delta|B|$ for all $a\in A$, $deg(b)>\delta|A|$ for all $b\in B$.

THEOREM 3.2. (BLOW-UP LEMMA, [26]) *Given a graph R with $v(R)=r$ and positive parameters δ, Δ, there exists an $\varepsilon>0$ such that the following holds. Let $n_1,n_2,\dots,n_r$ be arbitrary positive integers, and let us replace the vertices of R with pairwise disjoint sets $V_1,V_2,\dots,V_r$ of sizes $n_1,n_2,\dots,n_r$ (blowing-up). We construct two graphs on the same vertex-set $V=\cup_iV_i$. The first graph $\tilde R$ is obtained by replacing each edge (v_i,v_j) of R with the complete bipartite graph between the corresponding vertex-sets V_i and V_j. The graph G is constructed by replacing each edge (v_i,v_j) of R with an (ε,δ)-super-regular pair between V_i and V_j. If a graph H with maximum degree bounded by Δ is embeddable into $\tilde R$, then it is also embeddable into G.*

The Generic Algorithm. Let $G=(V,E)$ be a given graph, $|V|=n$, with minimum degree at least cn, where $c=\frac13+\alpha$ and $\alpha\in(0,\frac16)$ (else $c\ge\frac12$). Let us fix $\beta>0$ to be very small and much smaller than α, i.e. $\beta\ll\alpha$.
Step 1. We apply the Regularity Lemma to G with parameters ε and d, s.t. $\varepsilon<\frac12$, $\varepsilon\ll d$ and $\frac{2\varepsilon+d}{1-d}\le\beta$. Note: when ε and d are arbitrarily small, then so is β. Also, $d+\varepsilon\le\beta$. Based on the output from the Regularity Lemma, we define a *reduced* graph R as follows. The vertices of R are the clusters $V_1,V_2,\dots,V_k$ (we skip the cluster V_0 here), and we put an edge between V_i and V_j in R if (V_i,V_j) is ε-regular with density $\ge d$. From now on we mostly deal with graph R. In particular, we will treat cluster V_0 in the end, and will also discard some vertices from clusters V_i, $i\ge1$, and place them into V_0. ε was a fixed constant, $k\le M(\varepsilon)$, $k=|V(R)|$, so k is also a fixed constant.

LEMMA 3.3. *The degree of each vertex in R is at least $(\frac13+\alpha-\beta)k$.*

Proof. Let us fix a vertex $v_i\in V(R)$. Let V_i be the cluster corresponding to vertex v_i. Assume towards a contradiction that $deg_R(v_i)<(\frac13+\alpha-\beta)k$. Consider all the clusters $V_j\ne V_i$ such that $d(V_i,V_j)<d$. The number of such clusters V_j is $k-deg_R(v_i)-1$, and for each such cluster V_j there are less than dm^2 edges between V_i and V_j (by $d(V_i,V_j)<d$). The overall number of edges running between V_i and such $k-deg_R(v_i)-1$ clusters V_j is less than $dm^2(k-deg_R(v_i)-1)$. Therefore, there is a vertex in cluster V_i, say $u\in V_i$, such that $deg(u,W)<dm(k-deg_R(v_i)-1)$, where W is the union of $k-deg_R(v_i)-1$ clusters V_j. Finally, the degree of u in G can be bounded as

$$deg_G(u)<deg(u,W)+deg_R(v_i)m+deg(u,V_i)+$$

$$deg(u,V_0)\le dm(k-deg_R(v_i)-1)+deg_R(v_i)m+$$

$$m+\varepsilon n=deg_R(v_i)m(1-d)+dmk+m(1-d)+$$

$$+\varepsilon n<\left(\frac13+\alpha-\beta\right)(1-d)mk+dmk+2\varepsilon n\le$$

$$((1/3+\alpha-\beta)(1-d)+d+2\varepsilon)n\le(1/3+\alpha)n.$$

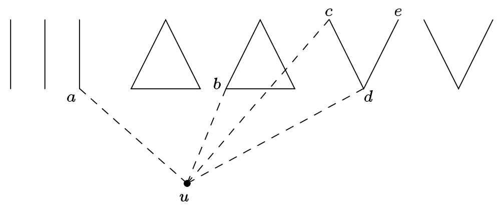

Figure 1: An illustration for proofs of Lemma 3.4 & 3.5.

The last estimate follows by using our assumption, that $\frac{2\varepsilon+d}{1-d} \leq \beta$. And thus we have derived a contradiction, which proves the claim. $\square$

Step 2. We call a path on 3 vertices a *v-shape*, and a cycle of length 3 a *triangle*. Compute a decomposition of R into a maximal collection of edges, triangles and v-shapes, s.t. these subgraphs are pairwise vertex-disjoint in R. Since R has constant size, we can use brute force.

LEMMA 3.4. *Each vertex of R is included in this decomposition.*

Proof. Otherwise, there is a vertex $u \in V(R)$ not included in any edge, triangle or v-shapes of the decomposition. Since the degree of u is, by Lemma 3.3, greater than $\frac{1}{3}k$, u cannot be just adjacent to all center vertices, like vertex d in Fig. 1, of all v-shapes. Thus, if u is adjacent to an end vertex – vertex c in Fig. 1 – of a v-shape, then we could produce two new edges (u, c) and (d, e) from u and the v-shape. A contradiction. If u is adjacent to a vertex of a triangle – vertex b in Fig. 1 – then we can produce two new edges from u and that triangle, which again gives a contradiction. If u is adjacent to a vertex, say a of an edge in Fig. 1, then we obtain a contradiction by producing a new v-shape. $\square$

Step 3. Let us fix an edge (V_i, V_j) of the decomposition of R. Recall, that pair (V_i, V_j) was ε-regular with density $\geq d$. We first make the pair (V_i, V_j) super-regular. We know: $|V_i| = |V_j| = m$. By Lemma 3.1, the number of vertices $v \in V_i$ with small degree $deg(v, V_j) \leq (d-\varepsilon)|V_j|$ is at most $\varepsilon|V_i| = \varepsilon m$. We delete these εm low degree vertices from V_i and put them into V_0. Similarly, we delete εm low degree vertices from V_j and put them into V_0. After that, for any $v \in V_i$, $deg(v, V_j) > (d - 2\varepsilon)|V_j|$, and for any $w \in V_j$, $deg(w, V_i) > (d-2\varepsilon)|V_i|$. By Lemma 3.2, with $\gamma = 1-\varepsilon$, we get $\varepsilon' = 2\varepsilon$ (since $\varepsilon \leq \frac{1}{2}$). And by that lemma, the new pair (V_i, V_j) is 2ε-regular, with density $\geq d - \varepsilon$. Now easily, pair (V_i, V_j) is $(2\varepsilon, d - 3\varepsilon)$-super-regular, and $|V_i| = |V_j|$.

We can now apply the Blow-up Lemma to pair (V_i, V_j), with $\delta = d-3\varepsilon$, $\Delta = 2$ (i.e. we are looking for a Hamilton cycle), $r = 2$, $n_1 = n_2 = |V_i| = |V_j|$. By that lemma, there is a Hamilton cycle of length $2n_1$ spanning (V_i, V_j) (this cycle can be found efficiently [27]).

We sketch that the same can be done for any v-shape (V_i, V_j, V_l) of the decomposition. First, we make the pairs (V_i, V_j) and (V_j, V_l) super-regular exactly in the same way as before. Then, the new clusters V_i and V_j have sizes greater than the size of the new V_l. To make the sizes equal, we use Lemma 3.2 once to pair (V_i, V_j) and once to pair (V_j, V_l), by deleting arbitrary εm vertices in V_i and arbitrary εm vertices in V_j. Then, we use the Blow-up Lemma and find: **(i).** Two Hamilton cycles: one in (V_i, V_j) and the other in (V_j, V_l) (for 2-VC and 2-EC problems); or **(ii).** A set of $|V_i|$ vertex disjoint paths, each going between three vertices – one in V_i, one in V_j, and one in V_l (for maximum path packing and TSP(1,2) problems). In this case we delete from each cluster $2\varepsilon m$ vertices and place them into V_0.

Similar arguments can be applied to any triangle (V_i, V_j, V_l) of our decomposition, to find one Hamilton cycle for the subgraph induced by the vertex-set $V_i \cup V_j \cup V_l$. In this case we need to delete up to $4\varepsilon m$ vertices from each cluster, and place them into V_0.

Finally, in the worst case for each cluster V_i, we have deleted at most $4\varepsilon m$ vertices from V_i and placed them into V_0. Thus, the size of V_0 increased by at most $4\varepsilon mk \leq 4\varepsilon n$, and so the new V_0 has size at most $5\varepsilon n$.

We use the computed structures in the decomposition subgraphs to built the final solution to a problem in mind. The rest of the algorithm is problem-specific. We have to specify how to put the structures together, and how to deal with cluster V_0. The lower/upper bound used to relate the size of the solution to the optimum, will always be n – the number of vertices. We upper/lower bound the sizes of the computed structures in the decomposition by charging the vertices in the clusters, using them as a "local" lower/upper bound.

Generic Analysis. The next lemma is crucial.

LEMMA 3.5. *If $\frac{1}{2}p_2 k$ is the number of all edges in the decomposition of R ($p_2 \in [0,1]$), then $p_2 \geq 6(\alpha - \beta)$.*

Proof. Recall, that k is the number of clusters (vertices) of the reduced graph R. Let $p_1 \cdot k$ be the number of vertices (clusters) in $V(R)$ in all the triangles of the decomposition, and let $p_2 \cdot k$ be the number of vertices in $V(R)$ in all the edges of the decomposition, for some $p_1, p_2 \in [0,1]$. Then $(1 - p_1 - p_2) \cdot k$ is the number of vertices in all the v-shapes. If there is no v-shape, then by Lemma 3.6, the local approximation factor is one. Assume thus that there is at least one v-shape, say P, in the decomposition. Consider an end vertex, say c, of P (see Fig. 1). There is no edge in $E(R)$ between c and any triangle, since otherwise we could replace that triangle and the v-shape P by three new edges in the decomposition (this is impossible by the maximality of our decomposition). Now, if c is adjacent to some edge of the decomposition, then it cannot be adjacent simultaneously to its two end vertices. Otherwise, we could replace that edge and the v-shape P by a new triangle and a new edge in the decomposition (this is again impossible by the maximality). Finally, we observe that if c is adjacent to an end vertex of any other v-shape, then we could replace the two v-shapes by three new edges, which contradicts the maximality. Thus, the maximum possible degree of c is $\frac{p_2}{2}k + \frac{(1-p_1-p_2)}{3}k$, and since R has minimum degree at least $(\frac{1}{3} + \alpha - \beta)k$ by Lemma 3.3, we obtain: $\frac{p_2}{2}k + \frac{(1-p_1-p_2)}{3}k \geq (1/3 + \alpha - \beta)\,k$, which gives $p_2 \geq 6(\alpha - \beta)$. $\square$

Application to 2-Connectivity. We run Steps 1,2 and 3 of the generic algorithm. In this way, we connect each edge and each triangle in the decomposition by one Hamilton cycle, and each v-shape by two Hamilton cycles. Thus, charging the edges used in the Hamilton cycles to the vertices of the original graph (as a "local" lower bound of 2-VC), the "local" approximation factor for each edge or triangle is one (Hamilton cycle has the number of edges equal to the number of vertices in these subgraphs), and for each v-shape it is $\frac{4}{3}$ (Hamilton cycle has $4|V_i|$ edges, and we have $3|V_i|$ vertices).

LEMMA 3.6. *The worst case local ratio for 2-connecting within any edge or triangle in the decomposition is one, and the ratio for 2-connecting within any v-shape is $\frac{4}{3}$.*

Step 4. So far we have 2-vertex-connected each subgraph of the decomposition. Let us contract each such subgraph into a single super-vertex, and delete all resulting self-loops. Consider now a graph, say $\tilde{G}$, with vertices being the union of the new super-vertices and the vertices in V_0. This graph is clearly 2-EC, and it has at most $k + 5\varepsilon n$ vertices. Compute an ear decomposition of $\tilde{G}$, and discard all 1-ears from it. The resulting graph is a spanning 2-EC subgraph of $\tilde{G}$. It is easy to check that the ear decomposition has at most $2(k+5\varepsilon n)$ edges. To make the graph 2-VC, it clearly suffices to add one additional edge for each block. Since the number of blocks is at most $k + 5\varepsilon n$, the overall additive error is at most $3(k + 5\varepsilon n) = 15\varepsilon n + 3k$.

LEMMA 3.7. *The overall additive error, i.e. the number of edges added to 2-connect all the structures of the decomposition and the vertices of cluster V_0 together, is at most $15\varepsilon n + 3k$.*

Finally, by Lemma 3.6 and 3.7, the size of the output solution can be upper-bounded by: $1 \cdot \frac{p_1}{3}k \cdot 3 \cdot (m - 4\varepsilon m) + 1 \cdot \frac{p_2}{2}k \cdot 2 \cdot (m - \varepsilon m) + \frac{4}{3} \cdot \frac{1-p_1-p_2}{3}k \cdot 3 \cdot (m - 2\varepsilon m) + 15\varepsilon n + 3k \leq p_1 km + p_2 km + \frac{4}{3} \cdot (1 - p_1 - p_2)km + 15\varepsilon n + 3k - \varepsilon mk$.

Assume that $3k \leq \varepsilon mk$. Then, since $mk \leq n$, the size of the solution is at most: $p_1 n + p_2 n + \frac{4}{3} \cdot (1 - p_1 - p_2)n + 15\varepsilon n = \left(\frac{4}{3} - \frac{p_1+p_2}{3} + 15\varepsilon\right) n$. This, by Lemma 3.5, and by the fact that $\varepsilon \leq d + \varepsilon \leq \beta$, and $n \leq$ *opt*, is upper-bounded by: $\left(\frac{4}{3} - 2(\alpha - \beta) + 15\varepsilon\right) n \leq \left(\frac{4}{3} - 2\alpha + 17\beta\right)$ *opt*. By choosing $\beta/17$ instead of β, we can get a bound of $(\frac{4}{3} - 2\alpha + \beta)$*opt*.

Assume now that $3k > \varepsilon mk$. Then $m < 3/\varepsilon$, and since $|V_0| \leq 5\varepsilon n$, we must have that the number of the rest of the vertices inside clusters $V_1, \ldots, V_k$ is at least $(1 - 5\varepsilon)n$. But $mk \geq |V_1| + \ldots + |V_k| \geq (1 - 5\varepsilon)n$. This, by $m < 3/\varepsilon$, gives $n \leq \frac{3k}{\varepsilon(1-5\varepsilon)} \leq \frac{3M(\varepsilon)}{\varepsilon(1-5\varepsilon)}$. Thus, the input graph has a fixed size, and we can solve the 2-connectivity problem on it exactly by enumeration. The polynomial running time of the overall algorithm follows basically by the algorithmic versions of the Regularity and Blow-up lemmas [1, 27]. Finally, we have proved the following theorem (implementation omitted).

THEOREM 3.3. *Let $G = (V, E)$ be a given 2-EC (or 2-VC) graph, with $|V| = n$, and degree of each vertex being at least $(\frac{1}{3} + \alpha)n$, where $\alpha \in [0, \frac{1}{6}]$ is any fixed constant. Let $\beta > 0$ be any fixed constant. Then there is a polynomial time $(\frac{4}{3} - 2\alpha + \beta)$-approximation algorithm for the unweighted 2-EC (and 2-VC) problem on G. The algorithm can be implemented in the* NC *class.*

Remark. If density is smaller than $\frac{1}{3}$ ($\alpha \leq 0$), then we can use known $\frac{4}{3}$-approximation algorithms.

Applications to path packing and TSP.

THEOREM 3.4. *Let $G = (V, E)$ be a given graph, with $|V| = n$, and degree of each vertex at least $(\frac{1}{3} + \alpha)n$, where $\alpha \in [0, \frac{1}{6}]$ is any fixed constant. Let $\beta > 0$ be any fixed constant. Then there is a deterministic polynomial time $(\frac{2}{3} + 2\alpha - \beta)$-approximation algorithm for the maximum path packing problem on G.*

THEOREM 3.5. *Let $G = (V, E)$ be a given complete graph, with $|V| = n$, with weights 1 or 2 on its edges. Let H be a subgraph of G induced by all the edges of weight 1. Assume that the minimum degree of H is at least $(\frac{1}{3} + \alpha)n$, where $\alpha \in [0, \frac{1}{6}]$ is any fixed constant. Then G defines a dense instance of the TSP(1,2), and there is a deterministic polynomial time $(\frac{4}{3} - 2\alpha + \beta)$-approximation algorithm for the TSP(1,2) defined by G for any fixed $\beta > 0$.*

Remark. If $\alpha \geq \frac{1}{6}$, H has a Hamilton cycle, which can be found in poly-time (Dirac's theorem). In this regard, Theorem 3.5 is a generalization of Dirac's theorem.

Application to longest path problem. To apply here the technique, we need the following result due to Bollobás and Brightwell.

PROPOSITION 3.1. (THM. 2.14, P. 27 IN [20]) *Let $p \in \mathbb{N}$ be positive and G be a simple graph with n vertices and of minimum degree at least $\frac{n}{p+1}$, where $n \geq 3$. Then G contains a simple cycle of length $\geq \frac{n}{p}$.*

THEOREM 3.6. *Let $G = (V, E)$ be a given graph, with $|V| = n$, and degree of each vertex at least cn, where $c \in [0, \frac{1}{2}]$ is any fixed constant. Let $\beta > 0$ be any fixed constant. Then there is a deterministic polynomial time $(\frac{c}{1-c} - \beta)$-approximation algorithm for the longest path problem on G. More exactly, the algorithm produces a path of length at least $(\frac{c}{1-c} - \beta)n$.*

Remark. We can obtain a similar result for the dense version of the longest cycle problem.

4 2-EC & 2-VC on Bounded Degree 3 Graphs

It is easy to see that in a graph with maximum degree 3, any ear decomposition is open. Thus 2-EC and 2-VC problems are here equivalent.

Local Optimization Heuristics. Let Π be a minimization problem on $G = (V, E)$, s.t. we want to find a subgraph of G feasible w.r.t. Π, with minimum number of edges. Given $j \in \mathbb{N}$, the *j-opt heuristic* is the algorithm which given any feasible solution $H \subseteq G$ to Π, repeats, until possible, the *j-opt exchange* operation: if there are sets $E_0 \subseteq E \setminus E(H), E_1 \subseteq E(H)$ with $|E_0| = j$, $|E_1| > j$, and $(H \setminus E_1) \cup E_0$ is feasible w.r.t. Π, then set $H \leftarrow (H \setminus E_1) \cup E_0$.

The Algorithm. Let $G = (V, E)$ be a given 2-EC graph, with $|V| = n$. W.l.o.g. we can assume that G is 2-VC. Otherwise we can solve the 2-EC problem separately on each 2-VC component.

The 1st step of the algorithm finds an ear decomposition H of G with minimum number ϕ of even ears, using the algorithm of Frank [15, 7] (delete all 1-ears, since they are redundant). In the 2nd step, the algorithm performs all possible 1-opt exchanges on H w.r.t. 2-EC. The resulting ear decomposition, say H', is the output.

LEMMA 4.1. ([7]) *$n + \phi - 1$ is a lower bound on the optimum 2-EC solution in G. An ear decomposition with ϕ even ears can be computed in $O(|V| \cdot |E|)$ time.*

Analysis. For the purpose of our analysis, we analyse a slightly different algorithm that produces a solution of a size lower bounded by the size of H' – the size of the original solution. Let a j-opt exchange that does not increase the number of even ears in H be called a *parity-preserving* j-opt exchange. More precisely, a parity-preserving j-opt exchange is a j-opt exchange which given H, produces a new feasible graph, say $\hat{H}$, such that $\hat{H}$ has an ear decomposition with no more than ϕ even ears.

The modified algorithm has the same first step as the previous one, producing the ear decomposition H. In the second step, the algorithm uses only parity-preserving 1-opt exchanges w.r.t. 2-EC, producing the final solution, say H''. It is clear that the size of H' is at most the size of H''. (We can just perform the second step of the original algorithm skipping all the 1-opt exchanges that are not parity-preserving.)

Let p_ℓ be the number of internal vertices in all ℓ-ears of H''. Then, $p_\ell/(\ell - 1)$ is the number of ℓ-ears. Let us fix a positive integer $k \leq \frac{n}{2}$. Then:

$$(1) \quad |E(H'')| \leq \sum_{i=2}^{2k} \frac{i}{i-1} p_i + \frac{2k+1}{2k} \left(n - \sum_{i=2}^{2k} p_i \right).$$

The first summation in the right-hand-side of (1) is the number of all edges in ℓ-ears for $\ell = 2, 3, \ldots, 2k$. Note, $n - \sum_{i=2}^{2k} p_i$ is the number of the internal vertices in all ℓ-ears, for $\ell > 2k$. We can rewrite (1) as follows.

$$(2) \quad \begin{aligned} |E(H'')| \leq & \left(\tfrac{2k+1}{2k} n + \textstyle\sum_{\substack{i=2 \\ 2|i}}^{2k} \left(\tfrac{i}{i-1} - \tfrac{2k+1}{2k} \right) p_i \right) \\ & + \left(\textstyle\sum_{\substack{i=3 \\ 2\nmid i}}^{2k-1} \left(\tfrac{i}{i-1} - \tfrac{2k+1}{2k} \right) p_i \right). \end{aligned}$$

Since in the modified algorithm we only use parity-preserving exchanges, the bound ϕ on the number of even ears still applies. Thus, we have: $n + \sum_{\substack{i=2 \\ 2|i}}^{2k} \frac{p_i}{i-1} - 1 \leq n + \phi - 1$, which gives

$$(3) \quad \begin{aligned} & \tfrac{2k+1}{2k} \left(n + \textstyle\sum_{\substack{i=2 \\ 2|i}}^{2k} \tfrac{p_i}{i-1} \right) \leq \\ & \tfrac{2k+1}{2k} (n + \phi - 1) + \tfrac{2k+1}{2k}. \end{aligned}$$

LEMMA 4.2. *For any $i \geq 2$, $\frac{i}{i-1} - \frac{2k+1}{2k} \leq \frac{2k+1}{2k} \frac{1}{i-1}$.*

The first term in the brackets in (2) can be upper bounded by $\frac{2k+1}{2k} opt + \frac{2k+1}{2k}$. Bounding the second term is harder. Given any ear S in an ear decomposition $\mathcal{E}$, we say that an internal vertex in S is *free* if its degree in $\mathcal{E}$ is exactly two. To prove the next lemma we heavily use the properties of a locally optimal solution.

LEMMA 4.3. *Each odd ear in the ear decomposition H'' has at least 2 free internal vertices.*

Assumption (∗). $\exists a > 0 : \sum_{\substack{i=3 \\ 2\nmid i}}^{2k-1} \left(\frac{a}{i-1} p_i\right) \leq n.$

If assumption (∗) holds, then we have.

$$(4)\quad \sum_{\substack{i=3 \\ 2\nmid i}}^{2k-1} \frac{2}{a}\left(\frac{3}{2} - \frac{2k+1}{2k}\right)\frac{ap_i}{i-1} \leq \frac{2}{a}\left(\frac{3}{2} - \frac{2k+1}{2k}\right)n.$$

LEMMA 4.4. *For $i \geq 3$, $(\frac{3}{2} - \frac{2k+1}{2k})\frac{2}{i-1} \geq \frac{i}{i-1} - \frac{2k+1}{2k}$.*

By Lemma 4.4, and (4), we can upper bound the second term in the brackets in (2) by $\frac{2}{a}(\frac{3}{2} - \frac{2k+1}{2k})n$. Since $n \leq opt$, and using (3), we bound our solution from (2) by: $|E(H'')| \leq \left(\frac{2k+1}{2k} + \frac{2}{a}\left(\frac{3}{2} - \frac{2k+1}{2k}\right)\right) opt + \frac{2k+1}{2k}$, and so $|E(H'')| \leq \left(1 + \frac{1}{a} + \frac{a-2}{2ak}\right) opt + \frac{2k+1}{2k}$. We can plug $n/2$ in k, to finally get.

$$(5)\qquad |E(H'')| \leq \left(1 + \frac{1}{a} + \frac{a-2}{an}\right) opt + \frac{n+1}{n}.$$

LEMMA 4.5. *If the input graph is of maximum degree 3, then assumption (∗) holds with $a = 4$.*

Proof. Note that $p_i/(i-1)$ is the number of all i-ears. For each odd ear, we assign to that ear its 2 free internal vertices (they exist by Lemma 4.3) and its 2 end vertices. Since the input graph has maximum degree 3, no vertex is assigned simultaneously to two different ears. □

Using Lemma 4.5, and the fact that in such a graph $opt \leq \frac{3}{2}n$, we obtain: $|E(H'')| \leq \left(\frac{5}{4} + \frac{1}{2n}\right) opt + \frac{n+1}{n} \leq \frac{5}{4}opt + \frac{3}{4} + \frac{n+1}{n} \leq \frac{5}{4}opt + 2$. The last estimate holds if $n \geq 4$. Now, if $2/opt > \epsilon$, where $\epsilon > 0$ is a fixed constant, since $n \leq opt \leq 2/\epsilon$, the input graph is of constant size. The problem can be solved exactly by enumeration. Otherwise, when $n \geq 4$ and $2/opt \leq \epsilon$, we get a $(\frac{5}{4} + \epsilon)$-approximation. Our analysis is tight with respect to the lower bound we use. This follows from the work of Cheriyan *et al.* [7], who show an infinite family of maximum degree 3 graphs, where the ratio of the size of optimum 2-EC and 2-VC subgraph to $n + \phi - 1$, is asymptotically $\frac{5}{4}$. Therefore, we obtain the following theorem.

THEOREM 4.1. *The local search is a $(\frac{5}{4} + \epsilon)$-approximation algorithm for the 2-EC problem on maximum degree 3 graphs (for any $\epsilon > 0$). The approximation ratio is asymptotically tight with respect to the lower bound.*

5 Related Conjectures and Integrality Gaps

This section describes applications of our results in the polyhedral combinatorics. Consider the standard cut LP relaxation for the unweighted 2-EC problem.

$$(6)\qquad \begin{array}{lll} \min & \sum_{e\in E} x_e & \\ \text{s.t.} & \sum_{e\in\delta(S)} x_e \geq 2 & \forall S \subset V, S \neq \emptyset \\ & x_e \geq 0 & \forall e \in E \end{array}$$

$\delta(S)$ denotes the set of all edges with exactly one end vertex in S. The optimum value of the LP is a lower bound on the optimal integral solution to 2-EC problem. If we add to this LP the constraints $\sum_{e\in\delta(\{v\})} x_e = 2, \forall v \in V$, then the new LP is the famous *subtour relaxation* of the TSP. It was proved that the optimal solution to LP (6) is equal to the optimum of the subtour relaxation, if one assumes metric costs on the edges [30].[5] The famous metric $\frac{4}{3}$ TSP conjecture due to Goemans [18] is as follows.

Conjecture 1. The integrality gap of the subtour relaxation of TSP with metric edge costs is at most $\frac{4}{3}$.

A related conjecture, see Carr and Ravi [6], reads.

Conjecture 2. The integrality gap of the LP (6) of 2-EC problem with metric edge costs is at most $\frac{4}{3}$.

Conjecture 1 implies the second one. Both are as now unsettled. Carr and Ravi [6] give a proof of a special case of Conjecture 2, where they restrict the LP (6) to *half-integral* solutions, i.e. with all $x_e \in \{0, \frac{1}{2}, 1\}$.

FACT 5.1. (CHERIYAN *et al.* [7]) *If Conjecture 2 (and thus also Conjecture 1) holds, then the integrality gap of LP (6) for unweighted 2-EC problem is at most $\frac{4}{3}$.*

The considerations in [7] and Fact 5.1 allow us to formulate the following.

Conjecture 3. The integrality gap of LP (6) for the unweighted 2-EC problem is at most $\frac{4}{3}$.

Conjecture 3 is implicit in Cheriyan *et al.* [7], and they prove it with $\frac{4}{3}$ replaced by $\frac{17}{12}$. Let LP be the optimum value of LP (6). Obviously $n \leq LP$, and n is the lower bound we used to obtain our algorithms for dense problems. By the parsimonious property, the optimum value of the LP (6) is equal to the optimum value of the LP relaxation of TSP(1,2) (weights 1 and 2 define a metric). This, and our previous results give:

THEOREM 5.1. *Let G has the minimum degree at least $(\frac{1}{3} + \alpha)n$, where $\alpha \in [0, \frac{1}{6}]$ is a fixed constant. Let $\beta > 0$ be any fixed constant. Then the integrality gap of:* **(i).** *the LP relaxation for the unweighted 2-EC problem on such graphs G; and* **(ii).** *the subtour LP relaxation of*

[5] A generalization of this property is a so-called *parsimonious* property [19].

the TSP(1,2) where G is the graph induced by weight one edges, is at most $\frac{4}{3} - 2\alpha + \beta$. The integrality gap of the LP relaxation for the unweighted 2-EC on maximum degree 3 graphs is at most $\frac{5}{4} + \epsilon$, for any fixed $\epsilon > 0$.

Thus, our results prove stronger (density-parametrized) versions of Conjectures 1 and 3, and of Conjecture 3 on maximum degree 3 graphs. On the other hand, the worst known lower bound on the integrality gap of LP for unweighted 2-EC: **(i)** on maximum degree 3 graphs is $\frac{10}{9}$, **(ii)** on $\frac{3}{10}$-dense graphs is $\frac{11}{10}$ (Petersen graph).

6 Hardness of Approximation

Hardness of Dense TSP, 2-EC & 2-VC.

LEMMA 6.1. *Assume, that TSP(1,2) is* NP*-hard to approximate within $(1 + \varepsilon_0)$, for a fixed $\varepsilon_0 > 0$. Fix any d_0 s.t. $0 < d_0 < \frac{1}{2}$, and let δ be s.t. $d_0 = \frac{1-\delta}{2}$. Let G with $v(G) = n$ be an instance of TSP(1,2), where the input graph has minimum degree $d_0 n$. If we know that its minimum cost TSP tour is either of cost n or at least $(1+\varepsilon_0\delta)n$, it is* NP*-hard to decide which of the two cases holds. The claim holds for $\varepsilon_0 = 1/742$.*

The following simple lemma can easily be deduced from the proof of Lemma 5.1 in [8].

LEMMA 6.2. ([8]) *Let $G = (V, E)$ be a graph with n vertices, and with weights 1 or 2 on its edges. Let H be the subgraph of G induced by all edges of weight 1, and assume that H is a spanning 2-VC subgraph of G. Let moreover T be a spanning tree of H, and T having l vertices of degree one. Then, we can find in polynomial time a TSP tour in G of cost at most $n + l - 1$.*

Using Lemmas 6.1 and 6.2 we can prove:

THEOREM 6.1. *Let $G = (V, E)$ be a 2-VC graph, and d_0 be s.t. $0 < d_0 < 1/2$. Then the unweighted 2-VC problem is* Max SNP*-hard on instances G with density $\geq d_0$. Moreover, if the TSP(1,2) is* NP*-hard to approximate within a factor of $(1 + \varepsilon_0)$, then it is* NP*-hard to approximate the unweighted 2-VC problem on d_0-dense instances to within $(1 + \frac{\varepsilon_0\delta}{2})$, where $d_0 = \frac{1-\delta}{2}$. The claim holds for $\varepsilon_0 = 1/742$.*

Proof. Assume, that we are given an instance of the dense TSP(1,2) problem on G ($v(G) = n$), where the subgraph, say G_1, of G induced by edges of weight one has minimum degree $d_0 n$. Let the minimum cost TSP tour, say T^*, on G be either of cost n or at least $(1 + \varepsilon_0\delta)n$. We show that if the unweighted 2-VC problem could be approximated to within $(1 + \frac{\varepsilon_0\delta}{2})$, then we could decide in polynomial time which of the two cases holds.

If G_1 is not 2-VC, then the minimum cost TSP tour T^* on G has cost $cost(T^*) > n$, and so $cost(T^*) \geq (1 + \varepsilon_0\delta)n$. So assume now that G_1 is 2-VC. Notice, that this also means that G_1 is a spanning subgraph of G. Let H_1 be any 2-VC spanning subgraph of G_1, and let T be a spanning tree of H_1, having l vertices of degree one. Since each vertex in H_1 has degree at least two, we have that $|E(H_1)| \geq n-1+\lceil\frac{l}{2}\rceil$. By Lemma 6.2, we can find in polynomial time a TSP tour, say T', in G, such that $cost(T') \leq n+l-1 = 2(n-1+\frac{l}{2})-(n-1) \leq 2|E(H_1)| - n + 1$.

Let H_1 be a minimum size 2-VC spanning subgraph of G_1. If $cost(T^*) = n$, then of course $|E(H_1)| = n$. If $cost(T^*) \geq (1 + \varepsilon_0\delta)n$, then by the above argument, we obtain that $2|E(H_1)| - n + 1 \geq (1 + \varepsilon_0\delta)n$, and so $|E(H_1)| \geq (1 + \frac{\varepsilon_0\delta}{2})n - \frac{1}{2}$. Thus, if there is a polynomial time $(1 + \frac{\varepsilon_0\delta}{2})$-approximation algorithm for the unweighted 2-VC problem (with just a bit smaller constant than ε_0), then it can decide if $cost(T^*) = n$ or $cost(T^*) \geq (1+\varepsilon_0\delta)n$, which is NP-hard by Lemma 6.1. The best known hardness constant ε_0 is $1/742$ [12]. $\square$

Remark. The hardness result in Theorem 6.1 can be modified to hold for the dense 2-EC problem.

Hardness of Dense Path Problems. As corollaries to the methods used in the previous section and using [14] one can also show:

THEOREM 6.2. *Let us fix any d_0 such that $0 < d_0 < \frac{1}{2}$, and let δ be such that $d_0 = \frac{1-\delta}{2}$. The longest path problem and the path packing problem on d_0-dense graphs are both* NP*-hard to approximate within $(1-\varepsilon_0\delta)$, where $\varepsilon_0 = 1/742$.*

Hardness of Sparse TSP, 2-EC & 2-VC. We show here a similar result to this in Lemma 6.1.

LEMMA 6.3. *Assume, we are given an instance of TSP(1,2) on a graph G, s.t. subgraph of G ($v(G) = n$) induced by weight-1 edges has maximum degree 3. Assume, that we know that its minimum cost TSP tour is either of cost n or at least $(1 + \varepsilon_0)n$, for some fixed $\varepsilon_0 > 0$. Then there exists such a constant $\varepsilon_0 > 0$, for which it is* NP*-hard to decide which of the two cases holds. The claim holds for $\varepsilon_0 = 1/786$. If G is 3-regular, then the claim holds for $\varepsilon_0 = 1/1290$.*

THEOREM 6.3. *Let $G = (V, E)$ be a 2-VC (or 2-EC) graph, with maximum degree 3. Then the unweighted 2-VC (and 2-EC) problem is* Max SNP*-hard on instances G. Moreover, it is* NP*-hard to approximate the unweighted 2-VC (2-EC) problem on such graphs G within $1573/1572$, and within $2581/2580$ if G is 3-regular.*

Acknowledgment. We thank Piotr Berman for helpful remarks and discussions, and in particular for suggesting one of the gadgets used in Lemma 6.3.

References

[1] N. Alon, R. Duke, H. Leffman, V. Rödl and R. Yuster. The algorithmic aspects of the regularity lemma. *J. Algorithms*, **16**, 80–109, 1994. Prelim. ver. in *Proc. 33rd FOCS*, 1992.

[2] N. Alon, R. Yuster and U. Zwick. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs. In *Proc. 26th ACM STOC*, 326–335, 1994.

[3] S. Arora, D.R. Karger and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proc. ACM STOC*, 284-293, 1995.

[4] C. Bazgan, M. Santha and Z. Tuza. On the Approximation of Finding A(nother) Hamiltonian Cycle in Cubic Hamiltonian Graphs. In *Proc. 15th STACS*, LNCS, 1998.

[5] B. Bollobás. *Extremal Graph Theory.* Academic Press, London, 1978.

[6] R. Carr and R. Ravi. A new bound for the 2-edge connected subgraph problem. In *Proc. 6th IPCO*, LNCS **1412**, 1998.

[7] J. Cheriyan, A. Sebő and Z. Szigeti. An Improved Approximation Algorithm for Minimum Size 2-Edge Connected Spanning Subgraphs. In *Proc. 6th IPCO*, LNCS **1412**, 126–136, 1998.

[8] A. Czumaj and A. Lingas. On approximability of the minimum-cost k-connected spanning subgraph problem. In *Proc. 10th ACM-SIAM SODA*, 281–290, 1999.

[9] E. Dahlhaus, P. Hajnal and M. Karpinski. On the Parallel Complexity of Hamiltonian Cycle and Matching Problem on Dense Graphs. *J. Algorithms*, **15**, 367–384, 1993.

[10] R. Diestel. *Graph Theory.* Springer, N.Y., 1997.

[11] L. Engebretsen. An explicit lower bound for TSP with distances one and two. In *Proc. 16th STACS*, LNCS, **1563**, 373–382, 1999.

[12] L. Engebretsen and M. Karpinski. Approximation hardness of TSP with bounded metrics. In the *Proc. 28th ICALP*, LNCS, 2001.

[13] C.G. Fernandes. A better approximation ratio for the minimum size k-edge-connected spanning subgraph problem. *J. Algorithms*, **28**, 105–124, 1998. Prelim. ver. in *Proc. 8th SODA*, 1997.

[14] W. Fernandez de la Vega and M. Karpinski. On the approximation hardness of dense TSP and other path problems. *Information Processing Letters*, **70**, 53–55, 1999.

[15] A. Frank. Conservative weightings and ear-decompositions of graphs. *Combinatorica*, **13**, 65–81, 1993.

[16] A. Frieze and R. Kannan. The Regularity Lemma and Approximation Schemes for Dense Problems. in *Proc. 37th FOCS*, 12–20, 1996.

[17] N. Garg, A. Singla, and S. Vempala. Improved Approximation Algorithms for Biconnected Subgraphs via Better Lower Bounding Techniques. In *Proc. 4th SODA*, 103–111, 1993.

[18] M.X. Goemans. Worst-case Comparison of Valid Inequalities for the TSP. *Math. Programming*, **69**, 335–349, 1995.

[19] M.X. Goemans, D.J. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Math. Programming*, **60**, 145–166, 1993.

[20] R.L. Graham, M. Grötschel and L. Lovász (Eds.) *Handbook of Combinatorics. Vol. I.* North-Holland, 1995.

[21] D. Karger, R. Motwani and G. Ramkumar. On Approximating the Longest Path in a Graph. *Algorithmica*, **18**, 82–98, 1997.

[22] R.M. Karp. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, R.E. Miller & J.W. Thatcher (Eds.), Plenum, New York, 1972.

[23] M. Karpinski. Polynomial Time Approximation Schemes for Some Dense Instances of NP-Hard Problems. *Algorithmica*, **30**, 386–397, 2001.

[24] M. Karpinski. Approximating Bounded Degree Instances of NP-hard Problems. In *Proc. 13th Symp. Fundamentals of Comp. Theory (FCT)*, LNCS, **2138**, 2001.

[25] S. Khuller and U. Vishkin. Biconnectivity Approximations and Graph Carvings. *J. ACM*, **41**(2), 214–235, 1994. Prelim. ver. in *Proc. 24th ACM STOC*, 1992.

[26] J. Komlós, G.N. Sárközy and E. Szemerédi. Blow-up Lemma. *Combinatorica*, **17**(1), 109–123, 1997.

[27] J. Komlós, G.N. Sárközy and E. Szemerédi. An algorithmic version of the Blow-up Lemma. *Random Struct. & Algorithms*, **12**, 297–312, 1998.

[28] J. Komlós and M. Simonovits. Szemerédi's Regularity Lemma and its applications in graph theory. *DIMACS Tech. Rep. 96-10*, 1996.

[29] P. Krysta and V. S. Anil Kumar. Approximation Algorithms for Minimum Size 2-Connectivity Problems. In *Proc. 18th STACS*, LNCS, **2010**, 431–442, 2001.

[30] C.L. Monma, S.B. Munson, and W.R. Pulleyblank. Minimum-weight two-connected spanning networks. *Math. Programming*, **46**, 153–171, 1990.

[31] C.H. Papadimitriou and M. Yannakakis. The Traveling Salesman Problem With Distances One and Two. *Math. of Oper. Res.*, **18**, 1–11, 1993.

[32] E. Szemerédi. Regular partitions of graphs. In *Colloques Internationaux C.N.R.S. No. 260. Problèmes Combinatoires et Théorie des Graphes*, Orsay, 399–401, 1976.

[33] S. Vempala and A. Vetta. Factor 4/3 Approximations for Minimum 2-Connected Subgraphs. In *Proc. 3rd Workshop APPROX*, LNCS, **1913**, 262–273, 2000.

[34] S. Vishwanathan. An approximation algorithm for the asymmetric travelling salesman problem with distances one and two. *Inf. Proc. Lett.*, **44**, 297–302, 1992.

[35] S. Vishwanathan. An Approximation Algorithm for Finding a Long Path in Hamiltonian Graphs. In *Proc. 11th SODA*, 680–695, 2000.