On the Approximation of Finding A(nother) Hamiltonian Cycle in Cubic Hamiltonian Graphs *

Cristina Bazgan[†] Miklos Santha[‡] Zsolt Tuza[§]

Abstract

It is a simple fact that cubic Hamiltonian graphs have at least two Hamiltonian cycles. Finding such a cycle is NP-hard in general, and no polynomial-time algorithm is known for the problem of finding a second Hamiltonian cycle when one such cycle is given as part of the input. We investigate the complexity of approximating this problem where by a feasible solution we mean a(nother) cycle in the graph, and the quality of the solution is measured by cycle length. First we prove a negative result showing that the LONGEST PATH problem is not constant approximable in cubic Hamiltonian graphs unless $P = NP$. No such negative result was previously known for this problem in Hamiltonian graphs. In strong opposition with this result we show that there is a polynomial-time approximation scheme for finding a second cycle in cubic Hamiltonian graphs if a Hamiltonian cycle is given in the input.

1 Introduction

LONGEST PATH and LONGEST CYCLE are well known problems in graph theory which were shown to be NP-complete in 1972 by Karp [6]. The approximability of the associated optimization problems is very much open, despite considerable efforts in recent years.

Monien [9] gave an algorithm to find a path of length k in time $O(k! \cdot n \cdot m)$, where n and m are the number of vertices and the number of edges of the graph, respectively. Karger, Motwani and Ramkumar [7] gave a polynomial-time algorithm which finds a path of length $\Omega(\log n)$ in any 1-tough graph. A similar result was obtained also by Fürer and Raghavachari [3]. Since 1-tough graphs include Hamiltonian graphs, these algorithms can be used in particular to find such paths in graphs which contain

*This research was supported by the ESPRIT Working Group RAND2 n^o 21726 and by the bilateral project Balaton, grant numbers 97140 (APAPE, France) and F-36/96 (TéT Alapítvány, Hungary)

[†]Université Paris-Sud, LRI, bât 490, F–91405 Orsay, France, bazgan@lri.fr

[‡]CNRS, URA 410, Université Paris-Sud, LRI, bât 490, F–91405 Orsay, France, santha@lri.fr

[§]Computer and Automation Institute, Hungarian Academy of Sciences, H–1111 Budapest, Kende u.13–17, Hungary, tuza@sztaki.hu

a Hamiltonian cycle. Alon, Yuster and Zwick [2] generalized this result by giving a polynomial-time algorithm which, for any $c > 0$, finds a path of length $c \log n$, in a graph containing such a path. Finding paths of length $\omega(\log n)$ in polynomial time is an open problem even for Hamiltonian graphs.

On the negative side, Karger, Motwani and Ramkumar [7] have proved that LONGEST PATH is not constant approximable in polynomial time unless P=NP. Their proof consists of two parts. First, they show that LONGEST PATH doesn't have a polynomial-time approximation scheme, unless P=NP; and that this property holds even when the input instances are restricted to Hamiltonian graphs. Then they give a self-improving scheme for the problem, showing that a polynomial-time approximation algorithm for *some* constant can be transformed into a polynomial-time approximation algorithm for *any* constant. These results remain valid also when the maximum degree of the input graph is bounded by a constant at least four. The self-improving scheme in [7] doesn't conserve hamiltonicity, however, and hence it is raised there as an open problem whether polynomial-time constant non-approximability can be proven also for Hamiltonian graphs (assuming $P \neq NP$).

In this paper we will answer this question in the affirmative, by proving an even stronger negative result. It will turn out that there exists a self-improving scheme for LONGEST PATH which preserves Hamiltonicity when the input graphs are further restricted to be also cubic. That LONGEST PATH remains NP-complete even for cubic graphs was shown by Garey, Johnson and Tarjan [5]. In addition to their theorem, we also prove that this problem doesn't have a polynomial-time approximation scheme in cubic Hamiltonian graphs, unless P=NP. These two results imply that LONGEST PATH is not constant approximable for any constant in cubic Hamiltonian graphs, unless P=NP. A similar result follows immediately for LONGEST CYCLE.

Galbiati, Morzenti and Maffioli [4] used this last result to prove that the maximum sum depth problem and the maximum sum distance problem are not constant approximable.

The LONGEST CYCLE problem has an interesting variant in cubic Hamiltonian graphs. According to a theorem of Smith (see e.g. [10, 14]), any such graph has at least two Hamiltonian cycles. Therefore, if some Hamiltonian cycle is given as part of the input, one can ask to find another Hamiltonian cycle in the graph. We shall call this problem SECOND HAMILTONIAN CYCLE. It is a well known instance of what Megiddo and Papadimitriou [8] call the class $TFNP$ of total functions. This class contains function problems associated with languages in NP where, for every instance of the problem, a solution is guaranteed to exist. Other examples in the class are FACTORING and the HAPPYNET problem.

Many functions in $TFNP$ (like the examples quoted above) have a challenging intermediate status between FP and FNP, the function classes associated with P and NP. Although these problems are not NP-hard unless NP=*co*-NP, no polynomial-time algorithm is known for them so far. We investigate here, up to our knowledge, for the first time in the literature, the approximability of a problem in $TFNP$. In particular, we show that — in striking opposition to the above negative result — SECOND HAMILTONIAN CYCLE admits a polynomial-time approximating scheme, where a feasible solution for this problem is a cycle different from the one given in the input.

The paper is organized as follows. In Section 2 we give the necessary definitions and reduce LONGEST PATH to approximating the longest path between two fixed vertices in cubic Hamiltonian graphs. In Section 3 we prove that this latter problem has no polynomial-time approximation scheme, and in Section 4 we deduce that it is not constant approximable either. In Section 5 we describe a polynomial-time approximation scheme for SECOND HAMILTONIAN CYCLE.

2 Preliminaries

We begin with some basic definitions, summarized in three groups.

Approximability. In this paper, by optimization problem we always mean an NP-optimization problem. Given an instance x of an optimization problem A and a feasible solution y of x, we denote by $m(x, y)$ the value of the solution y, and by $opt_A(x)$ the value of an optimum solution of x. The *performance ratio* of y is

$$R(x,y) = \max\left\{\frac{m(x,y)}{opt_A(x)}, \frac{opt_A(x)}{m(x,y)}\right\}.$$

For a constant $c > 1$, an algorithm is a *c-approximation* if for any instance x of the problem it returns a solution y such that $R(x, y) \leq c$. We say that an optimization problem is *constant approximable* if, for some $c > 1$, there exists a polynomial-time c-approximation for it. The class of problems which are constant approximable is denoted by APX. An optimization problem has a *polynomial-time approximation scheme* (a *ptas*, for short) if, for every constant $\varepsilon > 0$, there exists a polynomial-time $(1 + \varepsilon)$-approximation for it. Moreover, an optimization problem has an *efficient polynomial-time approximation scheme* (an *eptas*, for short) if, for every constant $\varepsilon > 0$, there exists an $(1 + \varepsilon)$-approximation algorithm for it with running time $f(1/\varepsilon) * n^c$ where c is a constant independent of ε.

Reductions. The notion of L-reduction was introduced by Papadimitriou and Yannakakis in [12]. Let A and B be two optimization problems. Then A is said to be *L-reducible* to B if there are two constants $\alpha, \beta > 0$ such that

1. there exists a function, computable in polynomial time, which transforms each instance x of A into an instance x' of B such that $opt_B(x') \leq \alpha \cdot opt_A(x)$,

2. there exists a function, computable in polynomial time, which transforms each solution y' of x' into a solution y of x such that $|m(x, y) - opt_A(x)| \leq \beta \cdot |m(x', y') - opt_B(x')|$.

For us the important property of this reduction is that it preserves ptas; that is, if A is L-reducible to B and B has a ptas then A has a ptas as well.

Paths, cycles, degree. Let $G = (V, E)$ be an undirected graph. A *path* of length k in G is a sequence of distinct vertices $v_0, v_1, \ldots, v_k$ such that, for all $0 \leq i \leq k - 1$, there is an edge between v_i and v_{i+1}. For two vertices s and t, an *s-t path* is a path whose first vertex is s and last vertex is t. A path of length at least three whose first and last vertices coincide is called a *cycle*. A path *covers* a subgraph H if it contains all the vertices of H. A path or a cycle is *Hamiltonian* if it covers G. The graph itself is called *Hamiltonian* if it has a Hamiltonian cycle, and it is called *cubic* if the degree of all its vertices is three. Finally, it is called *cubic with distinguished vertices* s and t if all its vertices have degree three except s and t which have degree two, and furthermore s and t are adjacent.

Our negative result is that no constant approximation exists for the longest path (cycle) problem in cubic Hamiltonian graphs. We now define the problems in question formally.

CH LONGEST PATH (CYCLE)

Input: A cubic Hamiltonian graph G.

Solution: A path (cycle).

Value: The length of the path (cycle).

Since CH LONGEST PATH is trivially L-reducible to CH LONGEST CYCLE, we will prove our non-approximability result for CH LONGEST PATH. In fact, for technical reasons, the assertion will be easier to prove for the following variant of the problem.

CH LONGEST s-t PATH

Input: A cubic Hamiltonian graph G with distinguished vertices s and t.

Solution: An s-t path.

Value: The length of the path.

It is probably standard knowledge (and it was pointed out to us by M. Yannakakis [15]) that these two problems have the same difficulty of approximation. We state here the exact reduction we need.

Lemma 1 *If* CH LONGEST PATH *is constant approximable, then* CH LONGEST s-t PATH *is constant approximable, too.*

Proof: Let us suppose that A is a polynomial-time algorithm which c-approximates CH LONGEST PATH for some $c > 1$. We will construct a polynomial-time algorithm B which $2c$-approximates CH LONGEST s-t PATH. Let $G = (V, E)$ be an instance of CH LONGEST s-t PATH with distinguished vertices s and t. We define the graph H which contains n copies $G_1, \ldots, G_n$ of the graph G, where the distinguished vertices of G_i are s_i and t_i. In addition, H contains the edges $\{t_i, s_{i+1}\}$ for $i = 1, \ldots, n-1$ and the edge $\{t_n, s_1\}$. Clearly, H is cubic and Hamiltonian.

Let the algorithm B run A on H, which then finds a path P of length $L \geq (n^2 - 1)/c$. This path starts in some copy of G, and then visits a series of copies which are all distinct, except maybe the last one which might be the same as the first one.

Let us suppose that P visits k copies. Inside each copy but maybe the first and the last, the subpath of P is an s-t path in G. Let ℓ be the length of the longest among these s-t paths. Then we have

$$L \leq (k-2)\ell + 2(n-1) + k \leq n(\ell+3) - 2\,,$$

which implies that $\ell > n/c - 3$. Thus, if $n \geq 6c$, a path of length ℓ is a $2c$-approximation for the longest s-t path in G. Finally, if $n < 6c$, we can find an s-t Hamiltonian path by brute force. $\square$

What is particular in these instances of the longest path problem is that the value of the optimum solution is known in advance. Although they remain hard to approximate, this property makes that Max 3Sat could not be L-reduced to them, unless P=NP, as we will show it in the next section. Therefore, to prove that they still don't have a ptas, we will reduce to them the special case of Max 3Sat where the value of an optimum solution is also known. Let us define it formally.

Satisfiable Max 3Sat

Input: A formula F with variables $x_1, \ldots, x_n$ and with clauses $C_1, \ldots, C_m$, where F is satisfiable.

Solution: A truth assignment for the variables.smallskip

Value: The number of clauses satisfied.

Satisfiable Max 3Sat$(4, \bar{4})$ is the restriction of Satisfiable Max 3Sat in which each variable and its negation appear at most four times in F.

Finally, let us state a variant of Longest Cycle for which we will design a ptas.

Second Hamiltonian Cycle

Input: A cubic Hamiltonian graph G and a Hamiltonian cycle C.

Solution: A cycle different from C.

Value: The length of the cycle.

3 CH Longest s-t Path has no ptas

The basis of our non-approximability result is the following refinement, due to Arora et al. [1], of Cook's theorem on the NP-hardness of 3Sat.

Theorem 2 *Let $\mathcal{L}$ be a language in NP. There exists a polynomial-time algorithm and a constant $0 < \varepsilon < 1$ such that, given any input x, the algorithm constructs an instance F_x of* 3Sat *which satisfies the following properties.*

1. *If $x \in \mathcal{L}$, then F_x is satisfiable.*
2. *If $x \notin \mathcal{L}$, then no assignment satisfies more than a fraction $(1-\varepsilon)$ of the clauses.*

The standard way for showing that an optimization problem has no ptas is to show the stronger result that it is hard for APX under L-reduction. But we cannot proceed here this way, since if $NP \neq co\text{-}NP$, then this stronger result doesn't hold for problems where the value of an optimum solution is known. This is somewhat analogous to the result of Megiddo and Papadimitriou [8] showing that an FNP-complete function cannot be total unless $NP = co\text{-}NP$.

Theorem 3 *If $NP \neq co\text{-}NP$, then an optimization problem where the value of an optimum solution is known can not be APX-hard under L-reduction.*

Proof: Let us suppose on the contrary that there is an APX-hard optimization problem A where the value of an optimum solution is known, and let $\mathcal{L}$ be any language in NP. We will show that $\bar{\mathcal{L}}$ is also in NP. For any input x, let F_x and the constant ε be given by Theorem 2. Since Max 3Sat is constant approximable, by our assumption there is an L-reduction from Max 3Sat to A. Let us suppose that this reduction transforms F_x into an instance x' of A.

The NP algorithm for $\bar{\mathcal{L}}$ works as follows. On input x, we can guess an optimum solution y' for x', since the value of such a solution is known. Via the L-reduction, we transform y' into an assignment for F_x. By the definition of the L-reduction, this has to be an optimum assignment. We accept x if it satisfies no more than a fraction $(1-\varepsilon)$ of the clauses. $\square$

Using Theorem 2 again, we can also prove that Satisfiable Max 3Sat has no ptas.

Lemma 4 Satisfiable Max 3Sat *has no ptas, unless $P = NP$.*

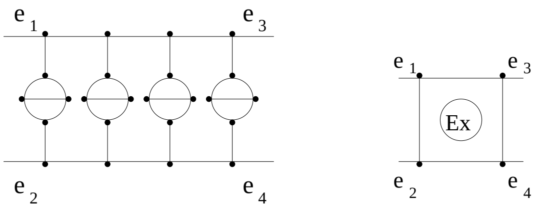
Proof: Suppose $P \neq NP$ and that Satisfiable Max 3Sat has a ptas denoted A. Let $\mathcal{L}$ be any NP-complete language; we will show the contradiction that $\mathcal{L}$ can be recognized in polynomial time. For an input x for $\mathcal{L}$, let the instance F_x and the constant $\varepsilon > 0$ be given by Theorem 2, and let m be the number of clauses in F_x. We choose $\delta < \varepsilon$ and apply the algorithm A to (F_x, δ). If $x \in \mathcal{L}$, then A returns in polynomial time an assignment satisfying at least $(1-\delta)m > (1-\varepsilon)m$ clauses. If $x \notin \mathcal{L}$, then A cannot find an assignment satisfying more than $(1-\varepsilon)m$ clauses. Therefore, we can decide in polynomial time whether x is in $\mathcal{L}$ or not. $\square$

Using now the L-reduction of [12] from Max 3Sat to Max 3Sat$(4, \bar{4})$, and observing that satisfiable instances are mapped into satisfiable instances, we get the following corollary.

Corollary 5 Satisfiable Max 3Sat$(4, \bar{4})$ *has no ptas, unless $P = NP$.*

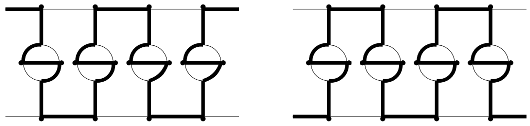
We now prove the main result of this section.

Theorem 6 CH Longest *s-t* Path *has no ptas, unless $P = NP$.*



(a) (b)

Figure 1: The ex-or device and its shorthand representation



(a) (b)

Figure 2:

Proof: We construct an L-reduction from Satisfiable Max 3Sat$(4, \bar{4})$ to CH Longest s-t Path. The outline of our construction follows the polynomial-time reduction given by Papadimitriou and Steiglitz [11] from 3Sat to the Hamiltonian cycle problem. In [13], Papadimitriou and Yannakakis gave an L-reduction from Max 3Sat$(4, \bar{4})$ to the traveling salesman problem with edges of weight one and two, by exploiting the strong connection between this latter problem and the Hamiltonian cycle problem. Although we will give an L-reduction which is more constrained than a polynomial-time reduction, basically we can avoid the complications in the construction of Papadimitriou and Yannakakis. The reason is that here we are concerned only with satisfiable instances of Max 3Sat$(4, \bar{4})$. On the other hand, additional difficulties will occur, since the graph to be constructed must be cubic and Hamiltonian.

In particular, similarly to both [11] and [13], we will use in our construction so-called variable and clause devices. The variable device will be taken from [11] (which is simpler than the one used in [13]), but for the clause device we will require additional features.

A basic ingredient for both is the modification of the ex-or device from [11] which is shown in Figure 1, where only the edges e_1, e_2, e_3, e_4 are joined with the rest of the graph. The difference with respect to the original ex-or device is that here all vertices have degree three. The ex-or device has the property that any covering path for the device, which starts and ends outside it, uses either the edge set $\{e_1, e_3\}$ or the edge set $\{e_2, e_4\}$ as connection with the rest of the graph, like in Figures 2(a) and 2(b). Also, it is impossible to have two disjoint paths starting and ending outside the device such that they both contain some vertices of the device and together they cover it. Ex-or devices can be connected in series, like in Figure 3(a).

Let F be an instance of Satisfiable Max 3Sat$(4, \bar{4})$ with n variables and m clauses. For each variable we will construct a variable device, and for each clause a clause device. For $1 \leq i \leq n$, let p_i be the number of positive occurrences of x_i in F, and let r_i be the number of its negative occurrences. For every i, the ith variable

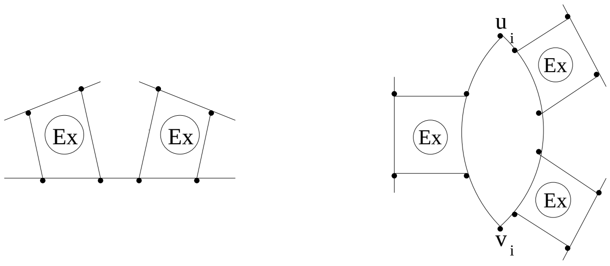


Figure 3:

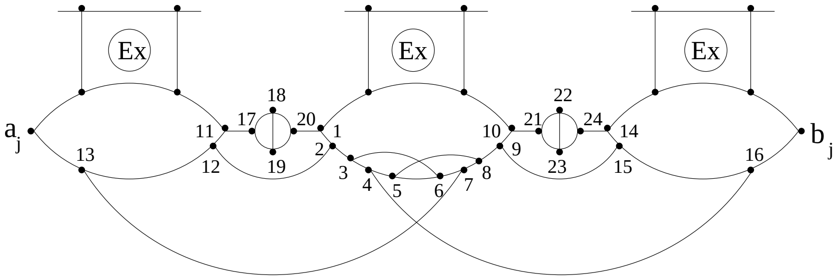


Figure 4: The clause device C_j

device is built as follows. For two specific vertices u_i and v_i, there are two paths between u_i and v_i. To one of these paths are attached p_i ex-or devices connected as in Figure 3(a), and we say that they are *standing for* x_i. To the other path are attached r_i ex-or devices in series which are standing for $\bar{x}_i$. If $p_i = 0$ or $r_i = 0$, then the corresponding path consists of just an edge. Figure 3(b) shows the variable device corresponding to a variable with $p_i = 1$ and $r_i = 2$.

The jth clause device corresponding to the clause C_j is shown in Figure 4, where the three ex-or devices stand for the three literals appearing in that clause. If C_j contains the literal x_i, then the jth clause device and the ith variable device will share an ex-or device which will stand in the latter for x_i. If C_j contains $\bar{x}_i$, then the same devices share again an ex-or device now standing for $\bar{x}_i$ in the variable device. The specific property satisfied by the clause devices is stated in the next lemma.

Lemma 7 *For any subset $D \neq \emptyset$ of the three ex-or devices in the jth clause device, there is a path from a_j to b_j which contains exactly those vertices of the clause device which are not in D. On the other hand, there is no path from a_j to b_j which contains all the vertices of the clause device.*

Proof: We describe, for every nonempty D, the corresponding Hamiltonian path. For simplicity, we refer to the three ex-or devices in the clause device as F (First), S (Second), and T (Third), and list the vertices of the paths as numbered in Figure 4.

$D = \{F, S, T\}$: a_j, 13, 12, 11, 17, 18, 19, 20, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 21, 22, 23, 24, 14, 15, 16, b_j.

$D = \{S,T\}:$ a_j, Ex, 11, 17, 18, 19, 20, 1, 2, 12, 13, 7, 6, 3, 4, 5, 8, 9, 10, 21, 22, 23, 24, 14, 15, 16, b_j.

$D = \{F,S\}:$ symmetrical to $D = \{S,T\}$.

$D = \{F,T\}:$ a_j, 13, 7, 6, 3, 2, 12, 11, 17, 18, 19, 20, 1, Ex, 10, 21, 22, 23, 24, 14, 15, 9, 8, 5, 4, 16, b_j.

$D = \{T\}:$ a_j, Ex, 11, 17, 18, 19, 20, 1, Ex, 10, 21, 22, 23, 24, 14, 15, 9, 8, 7, 13, 12, 2, 3, 6, 5, 4, 16, b_j.

$D = \{F\}:$ symmetrical to $D = \{T\}$.

$D = \{S\}:$ a_j, Ex, 11, 17, 18, 19, 20, 1, 2, 12, 13, 7, 8, 5, 6, 3, 4, 16, 15, 9, 10, 21, 22, 23, 24, 14, Ex, b_j.

Finally, if an a_j–b_j path covers all the three ex-or devices and contains the vertices $17, \ldots, 24$, then it cannot contain any other vertex of the clause device. □

The graph G contains all the variable and clause devices, and two additional vertices s and t. Beside the edges of the devices, there is an edge between s and u_1, between v_i and u_{i+1} for all $1 \leq i \leq n-1$, between v_n and a_1, between b_j and a_{j+1} for all $1 \leq j \leq m-1$, between b_m and t, and finally between s and t. If there is a satisfying assignment A for F, then the path which picks up in each variable device the ex-or devices standing for the literal satisfied by A, and which crosses the clause devices according to Lemma 7, is Hamiltonian. Moreover, G is cubic with distinguished vertices s and t which are adjacent and have degree two. We are going to show that this reduction is indeed an L-reduction.

Let N denote the number of vertices in G. Then the size of the longest s-t path is $N-1$. The number m of clauses in F is also the value of an optimum assignment for an instance of Satisfiable Max 3Sat$(4,\bar{4})$. Clearly, $m = \Theta(n)$, since every literal appears only a constant number of times in the formula. Since the variable and clause devices have a bounded number of vertices, $N = \Theta(m+n) = \Theta(n)$ holds, showing that the first condition of the L-reduction is satisfied.

For the second condition, let us consider an arbitrary s-t path P in G. We will call the vertices not appearing in this path *missing*.

We construct now from P a partial assignment A_P for the formula F, which will give a value to all variables whose corresponding variable device is "correctly traversed" by P for x_i or $\bar{x}_i$. We say that P *correctly traverses* the ith variable device *for* x_i if it covers all the ex-or devices standing for x_i, these ex-or devices are entered from the variable device, and none of the ex-or devices standing for $\bar{x}_i$ is entered from the variable device. In that case A_P assigns the value **true** for x_i. The definition for correctly traversing the ith variable device for $\bar{x}_i$ is analogous, but in this case A_P assigns the value **false** for x_i.

Lemma 8 *If the path P has k missing vertices, then the partial assignment A_P satisfies at least $m - 8k$ clauses.*

Proof: Let us suppose that a clause C_j is unsatisfied by A_P. Then either its three literals are made false by A_P or at least one of its literals didn't receive a truth value. In the former case, by the definition of A_P, the variable device of each literal was correctly traversed for the negation of that literal. Therefore, the only vertices where P can enter and leave the jth clause device are a_j and b_j, and there must be a missing vertex in that device by Lemma 7. In the latter case, there must be a missing vertex in the variable device corresponding to the variable without truth value. Since every variable and its negation appear at most 8 times altogether in F, the statement follows. □

To finish the proof of Theorem 6, we now show that the second condition in the definition of L-reduction is also satisfied. Since F is satisfiable, its optimum is m; and since G has a Hamiltonian cycle, its optimum is $N-1$. Let an s-t path P of length $N-1-\ell$ be given. Then there are ℓ missing vertices in the graph with respect to P. Let A be an assignment which extends A_P. By Lemma 8, A satisfies at least $m-8\ell$ clauses of F. Therefore, the second condition is satisfied with $\beta = 8$. □

Remark: A very similar proof shows that, if $P \neq NP$, the traveling salesman problem admits no ptas for the following restricted class of instances: each weight is equal to one or two, and the graph formed by the edges of weight one is cubic and Hamiltonian.

4 CH s-t Longest Path is not in APX

Given an instance $G=(V,E)$ of CH Longest s-t Path with distinguished vertices s and t, we now define the *vertex square* graph G^2 of G, which will be an instance of the same problem. The basic idea is to replace in G each vertex v by a copy G_v of G and by a *connector* device C_v. The connector device for v is shown in Figure 5. This device will, in a way to be specified later, connect G_v with the rest of G^2 through the vertices a_v, b_v, c_v which we call *exterior* vertices. One important property of the connector device is stated in the following lemma.

Lemma 9 *For every set $\{x,y\} \subseteq \{a_v, b_v, c_v\}$ there exist two paths, P_x starting from x and P_y starting from y, such that they are disjoint, together they contain all the vertices of the device, and the other two endpoints of the paths are s_v and t_v in some order.*

Proof: We describe the paths P_x and P_y for the three possible subsets. If $\{x,y\} = \{a_v, b_v\}$, we can take $P_x = a_v, c_v, 4, t_v$ and $P_y = b_v, 3, 6, 5, 2, 1, s_v$. The paths are analogous for the set $\{b_v, c_v\}$. Finally, if $\{x,y\} = \{a_v, c_v\}$, we can take $P_x = a_v, 1, s_v$ and $P_y = c_v, 4, 3, b_v, 2, 5, 6, t_v$. □

The graph G^2 will contain a copy G_v of G and a copy C_v of the connector device for every vertex v except s and t. It will also have two distinguished vertices S and T. For every v, we identify the distinguished vertices of G_v with the vertices s_v and t_v of C_v, and we delete the edge $\{s_v, t_v\}$. We denote the resulting graph by H_v, and call it the *component* corresponding to v. The components are connected by the following

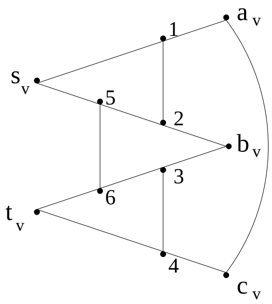


Figure 5: The connector device C_v

so-called *exterior* edges. For every edge $\{v, w\} \in E$, we put an edge between an exterior vertex of C_v and an exterior vertex of C_w. Let s' (respectively t') be the neighbor of s (t) in G different from t (s). We add an edge between S and an exterior vertex of $C_{s'}$ and an edge between T and an exterior vertex of $C_{t'}$. Finally, we add an edge between S and T.

Since there is a Hamiltonian s-t path in G, Lemma 9 implies that there is a Hamiltonian S-T path in G^2.

Lemma 10 *Any S-T path of length L in G^2 can be transformed in polynomial time into an s-t path in G of length $\sqrt{L} - 10$.*

Proof: Let P be a path of size L in G^2 between S and T. For every vertex v, P can enter (and leave) H_v via the exterior vertices of the connector device. Since there are three exterior vertices, this can be done at most once. Therefore, P induces a unique path P_v in every H_v. Also, this path is decomposed into two subpaths R_v^1 and R_v^2 in C_v and a subpath P'_v in G_v, since G_v and the connector are connected only via the vertices s_v and t_v. Clearly, P'_v is an clearly path in G. The total length of R_v^1 and R_v^2 is at most 9.

Let us suppose that P has k exterior edges. These edges naturally induce an s-t path of length k in G. Also, P visits $k - 1$ components. Let ℓ be the maximum length of P'_v over all the components which are visited. We claim that k or ℓ is at least $\sqrt{L} - 10$. Let us suppose the contrary. Then the length of P is at most $k + (k - 1)(\ell + 9) < L$, a contradiction. □

Using standard arguments, this self-improving scheme together with Theorem 6 yields

Theorem 11 CH LONGEST s-t PATH *is not constant approximable, unless P=NP.*

Proof: Let us suppose that there exists a polynomial-time c-approximation algorithm A for some constant $c > 1$. We will prove that, for any constant $\varepsilon > 0$, a polynomial-time $(1+\varepsilon)$-approximation algorithm can be constructed, in contradiction with Theorem 6.

Let G be a cubic Hamiltonian graph with distinguished vertices s and t, and let n be the number of vertices. If $n < 10/(c^{-1/2} - c^{-2/3})$, then we find an $(1+\varepsilon)$-approximation by brute force. Otherwise, let k be the smallest integer greater than $\log_{2/3} \log_c (1+\varepsilon)$.

For $i = 1, \ldots, k$ we construct G^{2^i} where, by definition, G^{2^i} is the vertex square graph of $G^{2^{i-1}}$. Then we apply the algorithm A to G^{2^k} to find a c-approximation P_k of the s-t Hamiltonian path in G^{2^k}, and then repeatedly the algorithm of Lemma 10 to find an approximation P_i of the s-t Hamiltonian path in G^{2^i}, for $i = k-1, \ldots, 0$. By a simple calculation, Lemma 10 and the lower bound on n imply that P_i is a $c^{(2/3)^{k-i}}$-approximation. Moreover, because of the choice of k, P_0 is easily checked to be an $(1+\varepsilon)$-approximation of an optimum path in G. For ε constant, the size of G^{2^k} and the running time of the algorithm are polynomial in n. $\square$

Our main negative results follow immediately from Lemma 1 and Theorem 11.

Theorem 12 CH LONGEST PATH *and* CH LONGEST CYCLE *are not in APX, unless P=NP.*

We can show a stronger non-approximability result under a stronger hypothesis.

Theorem 13 *For any $\varepsilon > 0$,* CH LONGEST PATH *and* CH LONGEST CYCLE *are not $2^{O(log^{1-\varepsilon} n)}$-approximable, unless $NP \subseteq DTIME(2^{O(log^{1/\varepsilon} n)})$.*

5 SECOND HAMILTONIAN CYCLE has an eptas

In this section we give an efficient approximation scheme for SECOND HAMILTONIAN CYCLE, that seems to us to be the first-ever approximation scheme for a problem in the complexity class $TFNP$. This result is obtained as a consequence of the following theorem:

Theorem 14 *Let $G = (V, E)$ be a cubic graph of order n with Hamiltonian cycle $C = v_1 v_2 \cdots v_n$. There is an algorithm that finds a cycle $C' \neq C$ of length at least $n - 4\sqrt{n}$ in $O(n^{3/2} \log n)$ steps.*

Corollary 15 SECOND HAMILTONIAN CYCLE *has an eptas in cubic Hamiltonian graphs.*

Afterwards, we will design a linear-time approximation scheme for SECOND HAMILTONIAN CYCLE.

Theorem 16 *For every $\varepsilon > 0$, a cycle $C' \neq C$ of length at least $(1-\varepsilon)n$ can be found in $2^{O(1/\varepsilon^2)} * n$ steps.*

In the proofs we shall apply the following terminology and notation.

Definitions For a given subgraph $H \subseteq G$, $E(H)$ denotes the edge set of H; and for a specified edge set $E' \subseteq E(G)$, the 'graph E'' is meant to be the graph with edge set E' and with those vertices of G which are incident to at least one edge belonging to E'.

We assume throughout that the vertices $v_1, v_2, \ldots, v_n$ follow each other in this order along the given Hamiltonian cycle C of G. The *length* of a chord $e = v_i v_j \in E(G) \setminus E(C)$ $(i < j)$ is defined as $||e|| := \min\{j-i, n+i-j\}$. We denote by P_e the shorter subpath of C with endpoints v_i and v_j if $||e|| < n/2$, and set $P_e := v_i v_{i+1} \cdots v_j$ if $||e|| = n/2$. Two distinct chords e, e' are said to be

crossing if $P_e \cap P_{e'} \neq \emptyset$, $P_e \not\subset P_{e'}$, and $P_{e'} \not\subset P_e$;

incomparable if $P_e \cap P_{e'} = \emptyset$;

parallel if they do not cross, i.e., either they are incomparable, or $P_e \subset P_{e'}$, or $P_{e'} \subset P_e$.

If $P_e \subset P_{e'}$, we also say that e is *smaller* than e'. The chord e is *minimal* if there is no chord smaller than e.

Proof of Theorem 14: Let $k := \lfloor \sqrt{n} \rfloor + 1$. First, we check in $n/2$ steps whether C has a chord of length at most k. If such a chord e exists, then $(E(C) \cup \{e\}) \setminus E(P_e)$ is a cycle of required length (even longer, actually).

Suppose that all chords are longer than k. We now consider k consecutive chords, say the ones starting from $v_1, \ldots, v_k$. Denoting by z_i the other endpoint of the chord e_i incident to v_i, we can find two subscripts i_1, i_2 such that z_{i_1} and z_{i_2} are at distance less than

$$\frac{n-k}{k-1} < k$$

apart on the path $P' := v_{k+1} v_{k+2} \cdots v_n$. Note that the order of the k vertices z_i on P' can be determined in at most $O(k \log k) = O(n^{1/2} \log n)$ steps by any standard sorting algorithm, and then the closest pair can be selected in k steps. If e_{i_1} and e_{i_2} are crossing chords, and say $i_1 < i_2$, then $v_{i_2} v_{i_2+1} \cdots z_{i_1-1} z_{i_1} v_{i_1} v_{i_1-1} \cdots z_{i_2+1} z_{i_2}$ is a cycle of length at least $n - 2k + 2 > n - 2\sqrt{n}$.

Otherwise, if e_{i_1} and e_{i_2} are parallel, we keep them as a starting configuration.

To simplify notation, denote $e := e_{i_1}$, $e' := e_{i_2}$, and assume that $e = v_a v_b$, $e' = v_{a'} v_{b'}$. It may be the case that e and e' are incomparable (i.e., neither of them is smaller than the other), but we may assume in any case, without loss of generality (by renumbering the vertices if necessary), that $P_e = v_a v_{a+1} \cdots v_{b-1} v_b$ and that $P_{e'} \not\subset P_e$. We then consider the next k chords $e'_1, \ldots, e'_k$, starting from the vertices $v_{a+1}, \ldots, v_{a+k}$, and select from them two chords f and f' the other endpoints of which are at distance less than k apart.

If both f and f' are smaller than e, and they are comparable, say f is smaller than f', then we rename $e := f$, $e' := f'$, and do the previous step again. Note that this situation cannot occur more than $O(n)$ times, since $||e||$ decreases in each step.

If f and f' are crossing, then a cycle of length at least $n - 2\sqrt{n}$ is easily found as above, therefore we may assume that f and f' do not cross.

Suppose next that f and f' are comparable and f or f' crosses e but it does not cross e'. In this situation again, e and the crossing chord create a cycle of length at least $n - 2\sqrt{n}$. Similarly, if f is smaller than e but f' crosses both e and e', then f' with any one of e, e' is a suitable choice to construct a cycle of required length.

Finally, suppose that f and f' are parallel and they cross both e and e'. Remove the two pairs of short arcs (of lengths $< k$) joining the parallel chords (i.e., remove those subpaths of C which join P_e with $P_{e'}$, and also those between P_f and $P_{f'}$) to create four paths of total length at least $n - 4k$. We then obtain a cycle longer than $n - 4\sqrt{n}$ by adjoining the four edges e, e', f, f'. □

In order to design linear-time approximations, we shall need the following Ramsey-type lemma.

Lemma 17 *Let e be a chord of the Hamiltonian cycle C, and Q a set of $3k^2 - 3k + 1$ chords crossing e. Then Q contains either k mutually crossing chords or $3k + 1$ parallel chords.*

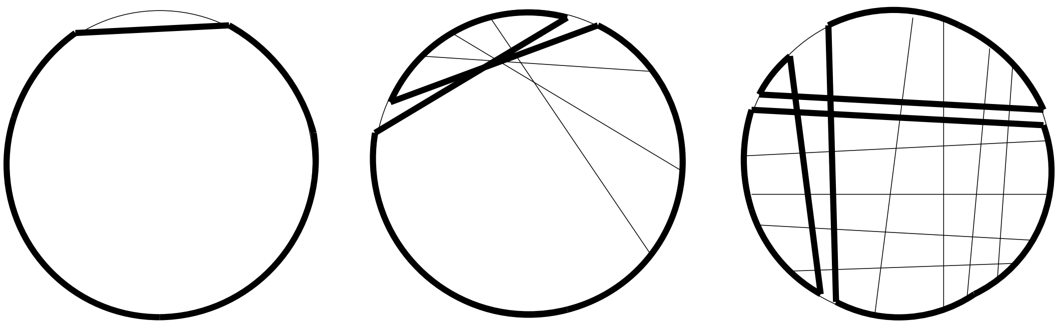
Proof: Let $e = v_i v_j$, $i < j$. For any two chords $e', e'' \in Q$, we write $e' <_e e''$ if and only if on both paths P_e and $E(C) \setminus E(P_e)$, v_i is closer to the endpoint of e' than to that of e''. Clearly, '$<_e$' defines a partial order, and two chords of Q are incomparable under $<_e$ precisely when they cross each other.

Removing all minimal elements from the partially ordered set $(Q, <_e)$, the length of each maximal chain decreases by precisely one. Therefore, it is easily seen by induction that $(Q, <_e)$ either contains a chain Q' of length $3k + 1$, or can be decomposed into at most $3k$ antichains. In the former case, Q' is a sufficiently large set of mutually parallel chords; while in the latter, some antichain Q'' of the decomposition has cardinality $|Q''| \geq \lceil |Q|/(3k) \rceil = k$. This Q'' provides us with k mutually crossing chords. □

Note that, for any fixed natural number k, it requires just a constant number of steps to find the subset Q' or Q'' of chords from any Q under the assumptions above.

Proof of Theorem 16: Given $\varepsilon > 0$, define $k := \lceil 1/\varepsilon \rceil$. We assume $n > 3k^3$, without loss of generality. (For $16k^2 \leq n < 3k^3$ we can find an $(1 + \varepsilon)$-approximation using the Theorem 14, and for $n < 16k^2$ a second *Hamiltonian* cycle $C' \neq C$ can be found in time $2^{O(1/\varepsilon^2)}$ by exhaustive search or as in the theorem of Smith [10, 14].) Our goal is to prove that at least one of the following three situations occurs for any cubic graph $G = (V, E)$ of order n, with Hamiltonian cycle $v_1 v_2 \cdots v_n$.

1. There is a chord of length at most $3k^2$ in G.
2. There are k mutually crossing chords in G.
3. There are two sets S_1, S_2 of chords such that



(1) (2) (3)

Figure 6: Three ways to find a long cycle

(a) $|S_1| = |S_2| = k+1$;

(b) S_i consists of mutually parallel chords, for $i = 1, 2$;

(c) each $e_1 \in S_1$ crosses each $e_2 \in S_2$.

Let us show first, how any one of these three possibilities implies the validity of the theorem. (See Figure 6 for illustration.)

Case 1. If e is a short chord, $||e|| \le 3k^2$, then $(E(C) \setminus E(P_e)) \cup \{e\}$ is a cycle longer than $n - 3k^2 > (1-\varepsilon)\,n$.

Case 2. If G contains k crossing chords, then there is a sequence $1 \le i_1 < i_2 < \cdots < i_{2k}$ such that $e_\ell := v_{i_\ell} v_{i_{\ell+k}} \in E(G)$ is a chord for all $1 \le \ell \le k$. Defining $d_\ell := (i_{\ell+1} - i_\ell) + (i_{\ell+k+1} - i_{\ell+k})$ for $1 \le \ell < k$ and $d_k := (i_{k+1} - i_k) + (n + i_1 - i_{2k})$, we have $d_1 + d_2 + \ldots + d_k = n$, therefore we may assume (after renumbering the vertices if necessary) that $d_1 \le n/k \le \varepsilon\, n$ holds. Deleting the vertices v_j for all $i_1 < j < i_2$ and all $i_{k+1} < j < i_{k+2}$ from C, the two remaining subpaths together with e_1 and e_2 form a sufficiently long cycle.

Case 3. With an appropriate choice of the starting vertex of the Hamiltonian cycle, the two subsets S_1, S_2 yield now a sequence

$$1 \le i_1 < i_2 < \cdots < i_{k+1} < j_1 < j_2 < \cdots < j_{k+1} < i_{k+2} < i_{k+3} < \cdots$$

$$\cdots < i_{2k+2} < j_{k+2} < j_{k+3} < \cdots < j_{2k+2} \le n\,,$$

corresponding to the chords $e_\ell := v_{i_\ell} v_{i_{2k+3-\ell}} \in S_1$ and $f_\ell := v_{j_\ell} v_{j_{2k+3-\ell}} \in S_2$. We consider, for $1 \le \ell \le k$, the sums $d_\ell := (i_{\ell+1} - i_\ell) + (j_{\ell+1} - j_\ell) + (i_{2k+3-\ell} - i_{2k+2-\ell}) + (j_{2k+3-\ell} - j_{2k+2-\ell})$. Now $d_1 + \ldots + d_k \le n - 4$, therefore at least one of those d_ℓ, say d_1 is smaller than $n/k \le \varepsilon n$. Then

$$v_{i_1} v_{i_{2k+2}} v_{i_{2k+2}+1} \cdots v_{j_{2k+1}-1} v_{j_{2k+1}} v_{j_2} v_{j_2+1} \cdots v_{i_{2k+1}-1} v_{i_{2k+1}} v_{i_2} v_{i_2+1} \cdots$$

$$\cdots v_{j_1-1} v_{j_1} v_{j_{2k+2}} v_{j_{2k+2}+1} \cdots v_{i_1-1}$$

is a cycle of required length.

It remains to show that one of the above cases must occur, and that the corresponding configuration can be found in linear time. If some of the $n/2$ chords has length at most $3k^2$, then it can be found in $O(n)$ steps and the proof is done. Therefore, from now we assume that every chord is longer. In this case, too, we can select a chord e of minimum length, in $O(n)$ steps. Say, $P_e = v_a v_{a+1} \cdots v_b$.

By the minimality of e, each of the $3k^2 - 3k + 1$ chords incident to the vertices $v_{a+1}, v_{a+2}, \ldots, v_{a+3k^2-3k+1}$ crosses e. Moreover, it follows from Lemma 17 that we can find among them either k mutually crossing chords (settled in Case 2), or $2k + 1$ parallel chords crossing e (in fact more, but $2k + 1$ are enough for the moment). The condition 'crossing e' implies that at least $k + 1$ of those $2k + 1$ chords form a 'chain,' i.e., we can choose a $(k+1)$-tuple $S := (e_0, e_1, \ldots, e_k)$ of chords, as a starting configuration, such that $P_{e_0} \subset P_{e_1} \subset \cdots \subset P_{e_k}$.

We are going to perform a constant number of steps (constant provided that k is fixed) after which either the pair S_1, S_2 is obtained or S is modified in such a way that $||e_k||$ decreases. The decrease in length cannot occur more than $n/2$ times, therefore eventually S_1, S_2 must occur, after at most $O(n)$ steps.

Let X be any set of $3k^2 - 3k + 1$ consecutive internal vertices of P_{e_0}, and Q the set of chords incident to X. As a matter of fact, $|Q| = |X|$ holds by the exclusion of short chords (Case 1). We first check in $3k^2 - 3k + 1$ steps whether all $f \in Q$ cross e_0. If this is not the case, say $f_0 \in Q$ is smaller than e_0, then we redefine $e_i := e_{i-1}$ for $i = k, k-1, \ldots, 1$ in this reverse order, and then put $e_0 := f_0$. Hence, $||e_k||$ has been decreased, still keeping S to be a chain of parallel chords.

On the other hand, if all $f \in Q$ cross e_0, then we apply Lemma 17 to find $3k + 1$ parallel chords $f_0, f_1, \ldots, f_{3k}$ in Q. (By what has been said concerning Case 2, we may assume that no k members of Q are mutually crossing.) Assuming that the endpoints of the f_i follow each other in the increase of their subscripts along P_{e_0}, we form three groups from those chords,

$$\begin{aligned} S' &:= \{f_i \mid 0 \le i \le k\} \\ S'' &:= \{f_i \mid k \le i \le 2k\} \\ S''' &:= \{f_i \mid 2k \le i \le 3k\} \end{aligned}$$

i.e., f_k and f_{2k} appear twice.

If some $f \in S''$ is parallel to e_k, then P_f entirely contains $P_{f'}$ for all $f' \in S'$ or all $f' \in S'''$. Thus, redefining $S := S'$ or $S := S'''$, the value of $||e_k||$ gets again decreased. On the other hand, if all $f \in S''$ cross e_k, then $S_1 := S$ and $S_2 := S''$ satisfy the requirements. □

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *Proof verification and hardness of approximation problems*, Proc. of 33rd FOCS, 1992, 14–23.

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

[3] M. Fürer, B. Raghavachari, *Approximating the Minimum-Degree Steiner Tree to within One of Optimal*, Journal of Algorithms 17 (1994), 409–423.

[4] G. Galbiati, A. Morzenti, F. Maffioli, *On the Approximability of some Maximum Spanning Tree Problems*, Theoretical Computer Science 181(1), 1997, 107–118.

[5] M. R. Garey, D. S. Johnson, R. E. Tarjan, *The planar Hamiltonian circuit problem is NP-complete*, SIAM J. Comput. 5:4 (1976), 704–714.

[6] R. M. Karp, *Reducibility among combinatorial problems.* In R. E. Miller and J. W. Thatcher editors, Complexity of Computer Computations, 1972, 85–103.

[7] D. Karger, R. Motwani, G. Ramkumar, *On Approximating the Longest Path in a Graph*, Algoritmica 18 (1997), 82–98.

[8] N. Megiddo and C. Papadimitriou, *On total functions, existence theorems and computational complexity*, Theoretical Computer Science 81 (1991), 317–324.

[9] B. Monien, *How to find long paths efficiently*, Annals of Discrete Mathematics 25 (1984), 239–254.

[10] C. Papadimitriou, *On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence*, Journal of Computer and System Science 48 (1994), 498–532.

[11] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[12] C. Papadimitriou and M. Yannakakis, *Optimization, Approximation and Complexity Classes*, Journal of Computer and System Science 43 (1991), 425–440.

[13] C. Papadimitriou and M. Yannakakis, *The traveling salesman problem with distances one and two*, Mathematics of Operations Research 18(1), 1993, 1–11.

[14] A. Thomason, *Hamilton cycles and uniquely edge colourable graphs*, Ann. Discrete Math. 3 (1978), 259–268.

[15] M. Yannakakis, personal communication, 1996.