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

(Extended abstract) *

Cristina Bazgan[1] Miklos Santha[2] Zsolt Tuza[3]

[1] Université Paris-Sud, LRI, bât.490, F–91405 Orsay, France, `bazgan@lri.fr`
[2] CNRS, URA 410, Université Paris-Sud, LRI, F–91405 Orsay, France, `santha@lri.fr`
[3] Computer and Automation Institute, Hungarian Academy of Sciences, H–1111 Budapest, Kende u.13–17, Hungary, `tuza@sztaki.hu`

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. 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 another 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 [7]. The approximability of the associated optimization problems is very much open despite considerable efforts in recent years.

Monien [10] gave an algorithm to find a path of length k in time $O(k! \cdot n \cdot m)$ where n and m are respectively the number of vertices and the number of edges of the graph. Karger, Motwani and Ramkumar [8] 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 [4]. Since 1-tough graphs include Hamiltonian graphs, these algorithms can be used in particular to find such paths in graphs which contain 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 which contains 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 [8] have proved that unless $P=NP$, LONGEST PATH is not constant approximable in polynomial time. Their proof consists of two parts. First, they have shown that LONGEST PATH doesn't have a polynomial time approximation scheme, unless $P=NP$. They were able to show this even when the input instances are restricted to Hamiltonian graphs. Then they gave a self-improving scheme for the problem, showing that a polynomial time approximating algorithm for *some* constant can be transformed into a polynomial time approximating 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. But their self-improving scheme didn't conserve Hamiltonicity, and they asked if it can be proven also for Hamiltonian graphs that they are not constant approximable in polynomial time, unless $P=NP$.

In this paper we will prove an even stronger negative result. It turns out that we will be able to give 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 [6]. In addition 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.

The LONGEST CYCLE problem has an interesting variant in cubic Hamiltonian graphs. It is not hard to show [11] that 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 will call this problem SECOND HAMILTONIAN CYCLE. It is a well known instance of what Meggido and Papadimitriou [9] 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 $NP=co\text{-}NP$, no polynomial time algorithm is known for them. We consider here (for the first time up to our knowledge) approximating a problem in $TFNP$. In particular, we show that in striking opposition with 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 prove that it is not constant approximable either. In section 5 we describe a ptas for SECOND HAMILTONIAN CYCLE.

2 Preliminaries

In this paper by optimization problem we always mean an NP-optimization problem. Let us recall a few notions about their approximability. 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 with $opt_A(x)$ 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 set of problems which are constant approximable is denoted by APX. An optimization problem has a *polynomial time approximation scheme* (in short a *ptas*) if for every constant $\varepsilon > 0$, there exists a polynomial time $(1 + \varepsilon)$-approximation for it.

The notion of L-reduction was introduced by Papadimitriou and Yannakakis in [13]. Let A and B be two optimization problems. A is *L-reducible* to B if there are two constants $\alpha, \beta > 0$ such that

1. there exists a polynomial time computable function which transforms an 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 polynomial time computable function which transforms any 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 also a ptas.

Let $G = (V, E)$ 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 $0 \leq i \leq k - 1$, 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 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 there is an edge between s and t.

Our negative result is that there is no constant approximation for the longest path (cycle) problem in cubic Hamiltonian graphs, problems we now define 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. For technical reasons it is easier to show it 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 [16]) 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 also constant approximable.*

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 it very unlikely that Max 3Sat could be L-reduced to them, 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.
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.
Let us finally state the variant of Longest Cycle for which we will be able to give 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 by Arora et al [1] of Cook's theorem on the NP-hardness of 3Sat.

Theorem 2. *Let L be a language in NP. There exists a polynomial time algorithm and a constant $0 < \varepsilon < 1$ such that, given an input x, the algorithm constructs an instance F_x of 3SAT which satisfies the following properties:*
1. If $x \in L$ then F_x is satisfiable.
2. If $x \notin L$ then no assignment satisfies more than 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 can not 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 [9] showing that an FNP-complete function can not 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.*

Using Theorem 2 we can prove that SATISFIABLE MAX 3SAT has no ptas.

Lemma 4. SATISFIABLE MAX 3SAT *has no ptas, unless $P{=}NP$.*

Using now the L-reduction of [13] 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 $P{=}NP$.*

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 [12] from 3SAT to the Hamiltonian cycle problem. In [14] 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 later problem and the Hamiltonian cycle problem. Although we will give an L-reduction which is more constraining than a polynomial time reduction, we basically can avoid the complications in the construction of Papadimitriou and Yannakakis. The reason for that is that (here) we are concerned only with satisfiable instances of MAX 3SAT$(4, \bar{4})$. On the other hand, we have additional difficulties since the graph we construct must be cubic and Hamiltonian. In particular, similarly to both [12] and [14] we will use in our construction so-called variable and clause devices. The variable device will be taken from [12] (which is simpler than the one used in [14]), but for the clause device we will use additional features.

A basic ingredient for both is the modification of the ex-or device from [12] which is shown in Fig.1, where only the edges e_1, e_2, e_3, e_4 are joined with the rest of the graph. The only difference with respect to the original ex-or device is
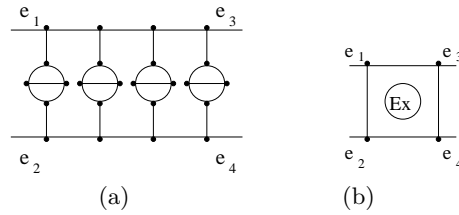
Fig. 1. The Ex-or device and its shorthand representation
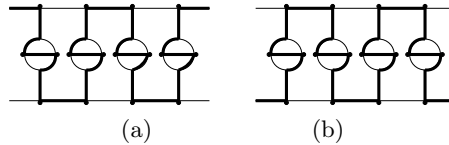

Fig. 2.

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 Fig. 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 Fig.3(a).
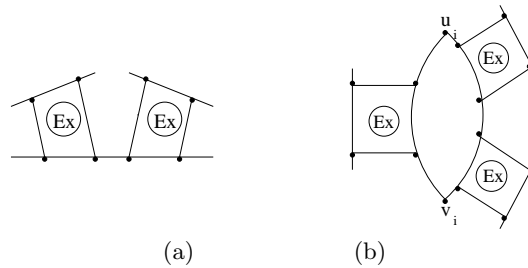

Fig. 3.

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 positives occurrences of x_i in F and let r_i be the number of its negatives occurrences. For every i, the ith variable device is the following: 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 like in Fig.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 Fig.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.
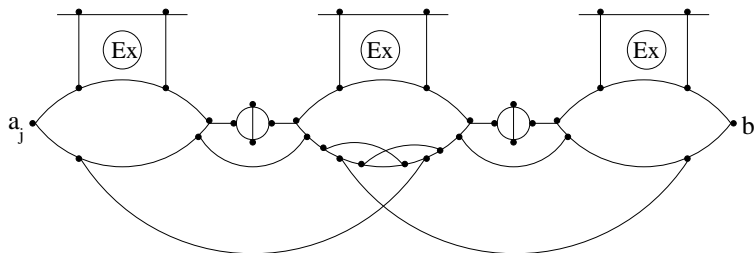
Fig. 4. The clause device C_j

Lemma 7. *For any subset $S \neq \emptyset$ 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 S. On the other hand, there is no path from a_j to b_j which contains all the vertices 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 $1 \leq i \leq n-1$, between v_n and a_1, between b_j and a_{j+1} for $1 \leq j \leq m-1$, 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. G is also cubic except for vertices s and t which have degree two. We show now that the reduction is indeed an L-reduction.

Let N be the number of vertices in G, then the size of the longest s-t path is $N-1$. The number of clauses m 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 the clause devices have a constant number of vertices, $N = \Theta(m+n) = \Theta(n)$, which shows 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 all the vertices not 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, in which 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 together at most 8 times in F, the statement follows. $\square$

To finish the proof of Theorem 6 we now show that the second condition in the definition of an 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 us given an s-t path P of length $N - 1 - \ell$. Then there are ℓ missing vertices in the graph. 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 $\beta = 8$. $\square$

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 every vertex v by a copy G_v of G and by a *connector* device C_v. The copy of the connector device for v is shown in Fig.5. This device will connect G_v with the rest of G^2 through the vertices a_v, b_v, c_v which we call *exterior* vertices. The 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.*

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 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.
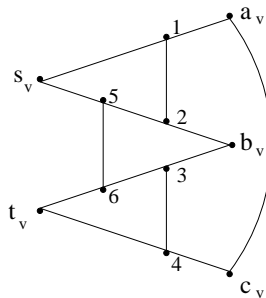
Fig. 5. The connector device C_v

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 $\sqrt{L} - 10$.*

This self-improving scheme with Theorem 6 gives using standard arguments

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

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 a ptas

In this section we prove that SECOND HAMILTONIAN CYCLE has a ptas in cubic Hamiltonian graphs, which is to our best knowledge the first-ever approximation

scheme for a problem in the complexity class TFNP. Actually we are going to prove this result in a much stronger form.

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.*

We will need the following terminology and notation.

Definitions. 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 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. 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. 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 $(n - k)/(k - 1) < k$ apart on the path $P' := v_{k+1} v_{k+2} \ldots v_n$. 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 $e_0 := e_{i_1}$, $e_0' := e_{i_2}$, and assume that $e_0 = v_a v_b$, $e_0' = v_{a'} v_{b'}$. It may be the case that e_0 and e_0' are incomparable (i.e., neither of them is smaller than the other), but we may assume without loss of generality (by renumbering the vertices if necessary) that $P_{e_0} = v_a v_{a+1} \cdots v_{b-1} v_b$ and that $P_{e_0'} \not\subset P_{e_0}$. 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_0 and f_0' the other endpoints of which are at distance less than k apart. If f_0 and f_0' are crossing, then a cycle of length at least $n - 2\sqrt{n}$ is easily found as above, therefore we may assume that f_0 and f_0' do not cross.

If both f_0 and f_0' are smaller than e_0, and say f_0 is smaller than f_0', then we rename $e_0 := f_0$, $e_0' := f_0'$, and do the previous step again. Note that this situation cannot occur more than $O(n)$ times.

Suppose next that f_0 or f_0' crosses e_0 but it does not cross e_0'. In this situation again, e_0 and the crossing chord create a cycle of length at least $n - 2\sqrt{n}$.

Similarly, if f_0 is smaller than e_0 but f'_0 crosses both e_0 and e'_0, then f'_0 with any one of e_0, e'_0 is a suitable choice to construct a cycle of required length.

Finally, suppose that f_0 and f'_0 are parallel and they cross both e_0 and e'_0. Remove the two pairs of short arcs (of lengths $< k$) joining the parallel chords (i.e., remove the subpaths of C that join P_{e_0} with $P_{e'_0}$ and also those between P_{f_0} and $P_{f'_0}$) 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_0, e'_0, f_0, f'_0. □

Remark. By very similar techniques, we can show that if $P \neq NP$ then the traveling salesman problem with weights one and two, restricted to instances where the graph formed by the edges of weight one is cubic and Hamiltonian, has no ptas. On the other hand, when a Hamiltonian cycle is given in the input, the problem has a ptas.

References

1. S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, *Proof verification and hardness of approximation problems*, Proc. of 33rd FOCS, pages 14-23, 1992.
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, pages 326-335, 1994.
3. R. P. Dilworth, *A decomposition theorem for partially ordered sets*, Ann. Math. (2) 51 (1950), 161–166.
4. M. Fürer, B. Raghavachari, *Approximating the Minimum-Degree Steiner Tree to within One of Optimal*, Journal of Algorithms, 17, pages 409-423, 1994.
5. G. Galbiati, A. Morzenti, F. Maffioli, *On the Approximability of some Maximum Spanning Tree Problems*, to appear in Theoretical Computer Science.
6. M. R. Garey, D. S. Johnson, R. E. Tarjan, *The planar Hamiltonian circuit problem is NP-complete*, SIAM J. Comput. 5:4, pages 704-714, 1976.
7. R. M. Karp, *Reducibility among combinatorial problems.* In R. E. Miller and J. W. Thatcher editors, Complexity of Computer Computations, pages 85-103, 1972.
8. D. Karger, R. Motwani, G. Ramkumar, *On Approximating the Longest Path in a Graph*, Proc. of 3rd Workshop on Algorithms and Data Structures, LNCS 709, pages 421-432, 1993.
9. N. Megiddo, C. Papadimitriou, *On total functions, existence theorems and computational complexity*, Theoretical Computer Science, 81, pages 317-324, 1991.
10. B. Monien, *How to find long paths efficiently*, Annals of Discrete Mathematics, 25, pages 239-254, 1984.
11. C. Papadimitriou, *On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence*, Journal of Computer and System Science 48, pages 498-532, 1994.
12. C. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
13. C. Papadimitriou, M. Yannakakis, *Optimization, Approximation and Complexity Classes*, Journal of Computer and System Science 43, pages 425-440, 1991.
14. C. Papadimitriou, M. Yannakakis, *The traveling salesman problem with distances one and two*, Mathematics of Operations Research, vol. 18, No. 1, pages 1-11, 1993.
15. A. Thomason, *Hamilton cycles and uniquely edge colourable graphs*, Ann. Discrete Math. 3, pages 259-268, 1978.
16. M. Yannakakis, personal communication, 1996.