



HAL
open science

Approximation algorithms for the maximum Hamiltonian Path Problem with specified endpoint(s)

Jérôme Monnot

► **To cite this version:**

Jérôme Monnot. Approximation algorithms for the maximum Hamiltonian Path Problem with specified endpoint(s). European Journal of Operational Research, 2005, 161, pp.721-735. hal-00004071

HAL Id: hal-00004071

<https://hal.science/hal-00004071>

Submitted on 26 Jan 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approximation algorithms for the maximum Hamiltonian Path Problem with specified endpoint(s)

Jérôme Monnot*

12 September 2003

Abstract

This paper deals with the problem of constructing Hamiltonian paths of optimal weight, called $\text{HPP}_{s,t}$ if the two endpoints are specified, HPP_s if only one endpoint is specified. We show that $\text{HPP}_{s,t}$ is $\frac{1}{2}$ -differential approximable and HPP_s is $\frac{2}{3}$ -differential approximable. Moreover, we observe that these problems can not be differential approximable better than $\frac{741}{742}$.

Based upon these results, we obtain new bounds for *standard ratio*: a $\frac{1}{2}$ -standard approximation for $\text{MAX HPP}_{s,t}$ and a $\frac{2}{3}$ for MAX HPP_s , which can be improved to $\frac{2}{3}$ for $\text{MAX HPP}_{s,t}[a, 2a]$ (all the edge weights are within an interval $[a, 2a]$), to $\frac{5}{6}$ for $\text{MAX HPP}_s[a, 2a]$ and to $\frac{2}{3}$ for $\text{MIN HPP}_{s,t}[a, 2a]$, to $\frac{3}{4}$ for $\text{MIN HPP}_s[a, 2a]$.

Keywords: Approximate algorithms; Differential ratio; Complexity theory; Combinatorial optimization; Performance ratio; Analysis of Algorithms; Hamiltonian paths.

1 Introduction

Routing design problems are of a major importance in combinatorial optimization, and the most important ideas of algorithmic have been applied to them during the last twenty years, see Christofides [6], Fisher et al. [15], Haimovich and Rinnooy Kan [19], Kosaraju et al. [25], Hassin and Rubinstein [21] and Bazgan et al. [5]. We will be concerned with some problems closely related to the Maximum Traveling Salesman problem, namely, the problem of finding a *Hamiltonian path* of maximum weight. We will study two variants depending on the number of specified endpoints (one or two) of the path. MAX HPP_s and $\text{MAX HPP}_{s,t}$ respectively denote the Hamiltonian path problem with one fixed endpoint $s \in V$ and two fixed endpoints $s, t \in V$. To our knowledge and from approximation point of view, these two latter problems have not been studied before, whereas their minimization versions have been studied by Hoogeveen [22] and Guttmann-Beck et al. [18] (in particular, it is well known that the minimization problems are **NP-hard**). We also deal with a variant called $\text{HPP}_{s,t}[a, 2a]$, where the edge-weights are in the set $\{a, a+1, \dots, b-1, b\}$. Both *Min*- and $\text{MAX HPP}_{s,t}$ are **NP-hard**, even in their restricted versions with $b > a$, since they are polynomial-time Karp-reducible [24] to each other.

*monnot@lamsade.dauphine.fr, CNRS-LAMSADE,UMR 7024, Université Paris-Dauphine, France

The maximum Hamiltonian path problem mainly has the same applications as the maximum traveling salesman problem since an optimum Hamiltonian path is easily converted to an optimum traveling salesman by adding one (or two) dummy vertex (vertices) with appropriate distance to all other vertices, where the specification of "appropriate" depends on the number of endpoints that have been fixed. Thus, for instance, it is known to be a relevant model for scheduling a single processor with setups arising in manufacturing, computing, VLSI design and many other applications, Lawler et al. [26]. However, this problem also has specific applications to *compression data*, Tarhio and Ukkonen [34] or *data array clustering* in DNA or marketing budgets, Hartigan [1]. For example, the *maximal compression problem* which arises in various compression data problems can be defined as follows: given a collection of strings s_1, \dots, s_n , we seek a string S such that every string in the collection is a substring of S and that maximizes $\sum_i |s_i| - |S|$. In the setting, the vertices represent strings and the weight of an edge between two "strings" is set to the amount of maximum overlap between these strings. The optimal compression is equivalent to the weight of a maximum Hamiltonian path. Another application to maximum Hamiltonian path with two specified endpoints is given by the following example, Garfinkel [17]: suppose we are given a data array in the form of an $m \times n$ matrix $A = (a_{i,j})$ that consists of elements that are either 1 if it exists a relation between row i and column j exists or 0 if it does not. We are interested in grouping rows and columns together in such a way that they show similar relations. For instance, consider a number of m marketing techniques and n products. If a marketing technique i works out successfully on a product j , then $a_{i,j}$ gets the value 1, and 0 otherwise. Similar marketing techniques are supposed to be successful on similar products. Therefore, clustering the techniques and the products gives insight in the relations of the marketing techniques and the products.

To formalize this, we introduce the measure of effectiveness $me_{i,j} = a_{i,j}(a_{i-1,j} + a_{i+1,j} + a_{i,j-1} + a_{i,j+1})$ for each element $a_{i,j}$. To ensure their existence, we add to the matrix A artificial rows of index 0 and $m+1$ and artificial columns of index 0 and $n+1$; these rows and columns contain zeroes only. The total measure of effectiveness of the matrix A , denoted by $tme(A)$, is computed by summing the measure of effectiveness over all elements of the matrix except for the artificial rows and columns. Thus, the goal is to find a matrix A' constructed from A by permuting some rows and some columns maximizing $tme(A')$.

For arbitrary permutations ρ and σ of the rows and columns (representing the matrix A'), the total measure of effectiveness of the matrix A' is $tme(A') = tme_1(A') + tme_2(A')$ where $tme_1(A') = \sum_{i=1}^m \sum_{j=1}^n (a_{\rho(i),\sigma(j)} \times a_{\rho(i),\sigma(j-1)} + a_{\rho(i),\sigma(j)} \times a_{\rho(i),\sigma(j+1)})$ and $tme_2(A') = \sum_{i=1}^m \sum_{j=1}^n (a_{\rho(i),\sigma(j)} \times a_{\rho(i-1),\sigma(j)} + a_{\rho(i),\sigma(j)} \times a_{\rho(i+1),\sigma(j+1)})$. Rewriting tme_1 yields: $tme_1(A') = \sum_{j=1}^n \sum_{k=1}^m 2a_{k,\sigma(j)} \times a_{k,\sigma(j+1)}$ since on the one hand, ρ is a permutation and on the other hand, $a_{k,\sigma(0)} = a_{k,\sigma(n+1)} = 0$. Thus, if we define the distance $d(i, j)$ between columns i and j as $d(i, j) = 2 \sum_{k=1}^m a_{k,i} \times a_{k,j}$, then we obtain the problem of finding a maximum Hamiltonian path from column 0 to column $n+1$. Similarly, we can by rewriting $tme_2(A')$ obtain the problem of finding a maximum Hamiltonian path from row 0 to row $m+1$, where this time the distances are defined by $d(i, j) = 2 \sum_{k=1}^m a_{i,k} \times a_{j,k}$. Finally, we see that problem of clustering a data array can be decomposed into two maximum Hamiltonian path problems, one defined on the rows and one defined on the columns.

We focus on the design of approximation algorithms with guaranteed performance ratios, that run within polynomial time and produce sub-optimal solutions. Usually, one compares the worst case ratio (called standard ratio) of the cost of the solution generated by the algorithm to the optimal cost, in the worst-case. However, we mainly refer in this article to another ratio called *differential ratio* which measures the worst ratio of, on the one hand, the difference between the cost of the solution generated by the algorithm and the worst cost, and on the other hand, the difference between the optimal cost and the worst cost. This measure, studied by Aiello et al. [2], Ausiello et al. [4], Cornuejols et al. [7], Vavasis [35] (in the context of non-linear programming), Zemel [36] and more recently by Demange and Paschos [13] and Hassin and Khuller [20], leads to new algorithms taking into account the extreme solutions of the instance, and provides the opportunity to better understand these problems. There are great differences between standard and differential approximation for the maximum Hamiltonian path problems. For instance, we can easily prove that the *Nearest Neighbor Heuristic* (see Fisher et al. [15] or Monnot [29]) is $\frac{1}{2}$ -standard approximable for MAX HPP_s and is $\frac{1}{3}$ -standard approximable for MAX HPP_{s,t} or that we have a trivial standard approximation scheme for MAX HPP_{s,t}[n; n + 1] whereas, the *Nearest Neighbor Heuristic* is not a differential approximation with any constant ratio for MAX HPP_s and MAX HPP_{s,t}[n; n + 1] is not differential approximate with ratio greater than $\frac{741}{742}$.

We now give some standard definitions:

Definition 1.1 An **NPO** problem π is a five-tuple $(\mathcal{I}, sol, m, Triv, goal)$ such that:

- (i) \mathcal{I} is the set of instances and is recognizable in polynomial-time.
- (ii) Given an instance $I \in \mathcal{I}$, $sol[I]$ is the set of feasible solutions of I ; moreover, there exists a polynomial P such that, for any $x \in sol[I]$, $|x| \leq P(|I|)$; furthermore, it is decidable in polynomial time whether $x \in sol[I]$ for any I and for any x such that $|x| \leq P(|I|)$. Finally, there is a feasible solution $Triv(I)$ ¹ computable in polynomial-time for any I .
- (iii) Given an instance I and a solution x of I , $m[I, x]$ denotes the non-negative integer value of x . The function m is computable in polynomial time and is also called the objective function.
- (iv) $goal \in \{Max, Min\}$. ◇

We call $\bar{\pi}$ the **NPO** problem $(\mathcal{I}, sol, m, Triv, \overline{goal})$ where \overline{goal} is defined as follows: if $goal = Max$, then $\overline{goal} = Min$ and $\overline{\overline{goal}} = Max$. The goal of an **NPO**-optimization problem with respect to an instance I is to find an *optimum solution* x^* such that $opt(I) = m[I, x^*] = goal\{m[I, x] : x \in sol[I]\}$. Another important solution of π is a *worst solution* x_* defined by: $wor(I) = m[I, x_*] = \overline{goal}\{m[I, x] : x \in sol[I]\}$. A worst solution for π is an optimal solution for $\bar{\pi}$ and vice versa. In Ausiello et al. [4], the term *trivial solution* referred to as *worst solution* and all the exposed examples have the property that a worst solution can be trivially computed in polynomial-time. For example, this is the case of the maximum Cut problem where, given a graph, the worst solution is the empty edge-set given by

¹The common definition of class **NPO** does not require the existence of a trivial solution.

the partition (V, \emptyset) , or the Bin-Packing problem where we can trivially put the items using a distinct bin per item. On the contrary, since a worst solution of the maximum weight Hamiltonian path from s to t is an optimal solution of the minimum weight Hamiltonian path from s to t , the computation of such a solution is **NP-hard**. Thus, computing a worst solution of HPP (or HPP_s or $\text{HPP}_{s,t}$ respectively) is as hard as computing an optimal one of HPP (or HPP_s or $\text{HPP}_{s,t}$ respectively). Note that the same property occurs for a large class of problems, Monnot [27].

1.1 Approximate algorithms and reductions

In order to study algorithm performances, there are two known measures: *standard ratio* [16], [3], [8] and *differential ratio* [13], [4], [20] and [7].

Definition 1.2 *Let π be an NPO problem and $x \in \text{sol}[I]$. We define the performance ratios of x with respect to the instance I as*

- **(standard ratio)** $\rho_\pi(I, x) = \text{Min} \left\{ \frac{m[I, x]}{\text{opt}(I)}, \frac{\text{opt}(I)}{m[I, x]} \right\}$
- **(differential ratio)** $\delta_\pi(I, x) = \frac{\text{wor}(I) - m[I, x]}{\text{wor}(I) - \text{opt}(I)}$ ◇

The performance ratio is a number less than or equal to 1, and is equal to 1 if and only if $m[I, x] = \text{opt}(I)$. Note that, compared to some definitions, we have inverted the standard performance ratio in the case of minimization problems so that the ratio value is always between 0 and 1. Let π be an NPO problem. For any instance I of π , a polynomial time algorithm A returns a feasible solution x^A . The performance of A with respect to $R \in \{\delta, \rho\}$ on the instance I is the quantity $R_A[\pi](I) = R_\pi(I, x^A)$. We say that A is an ε -approximation algorithm with respect to R if for any instance I , we have $R_A(I) \geq \varepsilon$.

Definition 1.3 *For any performance ratio $R \in \{\delta, \rho\}$,*

- *an NPO problem belongs to the class **APX**(R) if there exists an ε -approximation with respect to R for some constant $\varepsilon \in]0; 1[$.*
- *an NPO problem belongs to the class **PTAS**(R) if there exists an ε -approximation A_ε for any constant $\varepsilon \in]0; 1[$. The family $\{A_\varepsilon\}_{0 < \varepsilon < 1}$ is said to be a polynomial time approximation scheme.* ◇

Clearly, the following inclusion holds for any measure $R \in \{\delta, \rho\}$: $\text{PTAS}(R) \subseteq \text{APX}(R)$. As it is usually done, we will denote by **APX** and **PTAS**, respectively, the classes $\text{APX}(\rho)$ and $\text{PTAS}(\rho)$. We could argue whether the differential ratio is really pertinent: the authors of [13] and [4] answered positively to that question and concluded that this measure is complementary with the standard ratio. As shown in [11], many problems can have different behavior patterns depending on whether the differential or standard ratio is chosen: consider for instance Vertex Covering or Dominating Set problems. On the other hand, there are problems that establish some connections between the differential and the standard ratios, like Bin Packing [12] or maximum weight bounded-depth spanning tree [28] and see Zemel [36] for motivations and complementarity links between the two

measures. Besides, we show that there are tight links between both measures for the problems dealt with in the case where the edge-weights have lower and upper bounds.

Now, consider the following approximation preserving reductions between pairs (π, R) .

Definition 1.4 For $\pi_i \in NPO$ and $R_i \in \{\delta, \rho\}$, $i = 1, 2$,

- an A -reduction from (π_1, R_1) to (π_2, R_2) , denoted by $(\pi_1, R_1) \leq^A (\pi_2, R_2)$, is a triplet (α, f, c) such that:

(i) $\alpha: \mathcal{I}_{\pi_1} \mapsto \mathcal{I}_{\pi_2}$, transforms an instance of π_1 into an instance of π_2 in polynomial-time.

(ii) $f: sol_{\pi_2}[\alpha(I)] \mapsto sol_{\pi_1}[I]$, transforms solutions for π_2 into solutions for π_1 in polynomial-time.

(iii) $c: [0; 1] \mapsto [0; 1]$ (called expansion of the A -reduction) is a function satisfying $c^{-1}(0) \subseteq \{0\}$ and $\forall \varepsilon \in [0; 1], \forall I \in \mathcal{I}_{\pi_1}, \forall x \in sol_{\pi_2}[\alpha(I)]: R_2[\pi_2](\alpha(I), x) \geq \varepsilon \implies R_1[\pi_1](I, f(x)) \geq c(\varepsilon)$

- an A^*P -reduction from the pair (π_1, R_1) to the pair (π_2, R_2) , denoted by $(\pi_1, R_1) \leq^{A^*P} (\pi_2, R_2)$, is an A -reduction from (π_1, R_1) to (π_2, R_2) such that the restriction of function c to some interval $[a; 1]$ is bijective and $c(1) = 1$ ($c(0)$ may be non-zero). \diamond

An A -reduction preserves constant approximation while A^*P -reduction preserves approximation schemes. They are a natural generalization of those described by Orponen and Mannila [30] and Crescenzi and Panconesi [9].

Definition 1.5 If $(\pi_1, R_1) \leq^{A^*P} (\pi_2, R_2)$ and $(\pi_2, R_2) \leq^{A^*P} (\pi_1, R_1)$ with $c(\varepsilon) = \varepsilon$, we say that (π_1, R_1) is equivalent to (π_2, R_2) . \diamond

The differential ratio measures how the value of an approximate solution $m[I, x]$ is located in the interval between $opt(I)$ and $wor(I)$. More exactly it is equivalent for a maximization problem to prove $\delta_\pi(I, x) \geq \varepsilon$ and $m[I, x] \geq \varepsilon opt(I) + (1 - \varepsilon)wor(I)$. On the other hand, the standard ratio measures (for a maximization problem) how the value of an approximate solution is placed in the interval between 0 and $opt(I)$. Hence, we have an A^*P -reduction from the standard ratio to the differential ratio:

Lemma 1.6 If $\pi = (\mathcal{I}, sol, m, Triv, Max) \in NPO$, then $(\pi, \rho) \leq^{A^*P} (\pi, \delta)$ with $c(\varepsilon) = \varepsilon$.

Proof : Let I be an instance of π and x be a feasible solution. If $m[I, x] \geq \varepsilon opt(I) + (1 - \varepsilon)wor(I)$ then we have all the more so $m[I, x] \geq \varepsilon opt(I)$ since $wor(I) \geq 0$. \square

Note that, in general, there is no evident transfer of a positive or negative result from one framework to the other for a minimization problem. For instance, we have proved in Demange et al. [10] that a version of weighted minimum coloring admits a standard non-approximation threshold equal to $\frac{7}{8}$ in bipartite graphs whereas we have built a differential approximation scheme; in this coloring version, the cost of a stable set is given by the maximum of the vertex weights in this stable set.

2 The Hamiltonian path problem

The Hamiltonian path problem, also called the *Traveling Salesman Path problem*, is formally defined as follows.

Definition 2.1 Consider a complete graph K_n with non-negative costs $d(x, y)$ for each vertex pair. We want to find an optimal-cost Hamiltonian path, where the cost of a path is the sum of the weights on its edges. We refer this problem as HPP. When one endpoint s (resp. two endpoints s and t) of Hamiltonian path are specified, we use the notation HPP_s (resp. $\text{HPP}_{s,t}$).

If goal = Max, the problem is called MAX HPP, else MIN HPP. We use notation HPP, HPP_s or $\text{HPP}_{s,t}$ with no prefix when we consider without distinction the case goal = Max or goal = Min. \diamond

Standard ratio approximation results can be derived for HPP by using trivial reduction to TSP: the first negative approximation result (that we can deduce from [33]) states that it is not possible to approximate MIN HPP within $1/f(|I|)$ where f is any integer function computable within polynomial time unless $\mathbf{P}=\mathbf{NP}$. On the other hand, *metric*² MIN HPP is approximable within $2/3$ [6] and MIN HPP[1, 2] is **APX-complete** (deduced from Papadimitriou and Yannakakis [31]). For MAX HPP, the results are more optimistic since this problem is in **APX**. The best-known standard ratio is equal to $\frac{25}{33}$ and can be deduced from Hassin and Rubinfeld [21].

MIN METRIC $\text{HPP}_{s,t}$ is as hard to approximate as MIN METRIC HPP_s . Is MIN METRIC $\text{HPP}_{s,t}$ really much harder to approximate than MIN METRIC HPP_s ? This interesting question raised the first time by Johnson and Papadimitriou [23] on the relative hardness of the two specified endpoints version compared to the one specified endpoint is still open today. However, the positive results given on these problems lead to a positive answer to the question since the best-known standard ratios are $\frac{2}{3}$ for MIN METRIC HPP_s , Hoogeveen [22] and $\frac{3}{5}$ for MIN METRIC $\text{HPP}_{s,t}$ Hoogeveen [22], Guttmann-Beck et al. [18]. Finally, if we consider the case $a \leq d(e) \leq 2a$ there are no specific results. For example Christofides' modification algorithm [22] remains a $2/3$ -standard ratio for MIN $\text{HPP}_s[a; 2a]$. To our knowledge, no standard approximation result has been found for MAX HPP_s and MAX $\text{HPP}_{s,t}$.

We show that HPP_s is $\frac{2}{3}$ approximable and $\text{HPP}_{s,t}$ is $\frac{1}{2}$ approximable under the differential framework. We can deduce from Lemma 1.6 a $\frac{2}{3}$ -standard approximation for MAX HPP_s and a $\frac{1}{2}$ -standard approximation for MAX $\text{HPP}_{s,t}$. Moreover, our technique allows to handle the case where all the edge weights are within an interval $[a, 2a]$ for any positive a since from previous results, we deduce a $\frac{3}{4}$ (resp. $\frac{2}{3}$)-standard approximation for MIN $\text{HPP}_s[a, 2a]$ (resp. MIN $\text{HPP}_{s,t}[a, 2a]$) and a $\frac{5}{6}$ (resp. $\frac{3}{4}$)-standard approximation for MAX $\text{HPP}_s[a, 2a]$ (resp. MAX $\text{HPP}_{s,t}[a, 2a]$). Thus for these restrictions, we improve the best-known bounds of $\frac{2}{3}$ (resp. $\frac{3}{5}$) for minimization versions given by Hoogeveen [22] (resp. Guttmann-beck et al. [18] or [22]).

²Satisfying for all vertices x, y, z the inequality: $d(x, y) \leq d(x, z) + d(z, y)$.

3 Elementary properties

Let us first establish some relations between HPP, HPP_s, HPP_{s,t} and different subcases. We prove that HPP_{s,t} is the most general problem. As a second step, we establish for each problem some connected relations between differential and standard ratios. In the following paragraph, without specification, the properties that we present for HPP_{s,t} are also true for HPP_s and HPP.

HPP_{s,t} is as hard as HPP_s (which is itself as hard as HPP) to approximate for both performance ratios. Moreover, from a differential approximability point of view, these different versions are very close to the TSP, even if we consider the restriction $a \leq d(e) \leq b$.

Lemma 3.1 *For any goal $\in \{Min, Max\}$, we have:*

$$(i) \text{ (goal TSP}[a, b], \delta) \leq^{A*P} \text{ (goal HPP}_{s,t}[a, b], \delta) \text{ with } c(\varepsilon) = \varepsilon.$$

$$(ii) \text{ (goal HPP}[a, b], \delta) \leq^{A*P} \text{ (goal TSP}[a, b], \delta) \text{ with } c(\varepsilon) = \varepsilon.$$

Proof: We only show the case $goal = Max$.

• For (i): Let $I = (n, d)$ with $a \leq d(e) \leq b$ be an instance of MAX TSP $[a, b]$. Choose a vertex s in K_n and define $I_v = (n, s, v, d)$ an instance of MAX HPP_{s,v} $[a, b]$ for every $v \in V \setminus \{s\}$. Let μ_v be a Hamiltonian path from s to v of I_v which is an ε -differential approximation for MAX HPP_{s,v} $[a, b]$. So, for every $v \in V \setminus \{s\}$ we have:

$$m[I_v, \mu_v] \geq \varepsilon opt_{\text{MAX HPP}_{s,v}}(I_v) + (1 - \varepsilon) wor_{\text{MAX HPP}_{s,v}}(I_v) \quad (3.1)$$

From μ_v with $v \in V \setminus \{s\}$, we construct the Hamiltonian cycle $\Gamma = argmax\{m[I, \Gamma_v] : v \in V \setminus \{s\}\}$ where $\Gamma_v = \mu_v \cup \{(s, v)\}$.

Now, consider v^* such that an optimal Hamiltonian cycle of $I = (n, d)$ contains edge (s, v^*) ; thus, we have:

$$opt_{\text{MAX HPP}_{s,v^*}}(I_{v^*}) + d(s, v^*) = opt_{\text{MAX TSP}}(I) \quad (3.2)$$

Let μ_* be a worst Hamiltonian path from s to v^* ; $\mu_* \cup \{(s, v^*)\}$ is an Hamiltonian cycle and we deduce:

$$wor_{\text{MAX HPP}_{s,v^*}}(I_{v^*}) + d(s, v^*) \geq wor_{\text{MAX TSP}}(I) \quad (3.3)$$

Combining inequalities (3.1), (3.2) and (3.3), we obtain: $m[I, \Gamma] \geq m[I, \mu_{v^*}] + d(s, v^*) \geq \varepsilon opt_{\text{MAX TSP}}(I) + (1 - \varepsilon) wor_{\text{MAX TSP}}(I)$.

• For (ii): Let $I = (n, d)$ with $a \leq d(e) \leq b$ be an instance of MAX HPP $[a, b]$. We transform I into instance $\times(I) = (n+1, d')$ as follow: add a new vertex s to graph K_n and define $d'(s, v) = a, \forall v$, $d'(e) = d(e)$ for other edges. \square

Observe that the proof of item (ii) also holds for the standard ratio with $goal = Max$, but in this case, we might have $a = 0$. So, we deduce from the result of Hassin and Rubinfeld [21] for MAX

TSP that MAX HPP is $\frac{25}{33}$ -standard approximable. On the other hand, from the result of Sahni and Gonzalez [33] we know that MIN HPP_{s,t} is not in **APX** unless **P=NP**. This asymmetry in the approximability of both versions (MAX HPP_{s,t} is in **APX** as later proved) can be considered as somewhat strange given the structural symmetry existing between them. Since differential approximation is stable under affine transformation of the objective function (see for instance Hassin and Khuller [20] or Demange and Paschos [13]), MAX HPP_{s,t} and MIN HPP_{s,t} are differential-equivalent (see Definition 1.5).

Proposition 3.2 *The following assertions hold:*

- (i) MIN HPP_{s,t} is differential-equivalent to MAX HPP_{s,t}.
- (ii) MinHPP_{s,t}[a, b] is differential-equivalent to MaxHPP_{s,t}[a, b].
- (iii) HPP_{s,t} is differential-equivalent to metric HPP_{s,t}.
- (iv) HPP_{s,t}[a, b] is differential-equivalent to HPP_{s,t}[a + t, b + t], for any t.

Proof : Let $d_{max} = \max_{e \in E} d(e)$ and $d_{min} = \min_{e \in E} d(e)$. Given an instance with distance function d of the left problem in items (i) – (iv), we construct a distance function d' to an instance of the corresponding right problem in items (i) – (iv) as follows: (i) $d'(e) = d_{max} + d_{min} - d(e)$, (ii) $d'(e) = a + b - d(e)$, (iii) $d'(e) = d_{max} + d(e)$, (iv) $d'(e) = t + d(e)$. Since differential ratio is stable under affine transformation of the objective function (see Demange and Paschos [13] or Hassin and Khuller [20]), this concludes the proof. \square

Observe that the (iv) of this proposition allows to deal with the case where the distances are negative. The following easy theorem holds, thus giving a bridge between differential and standard ratios for *goal = Max* and *goal = Min*, in the case where edge weights belong to an interval $[a, b]$.

Theorem 3.3 (*goal* HPP_{s,t}[a, b], ρ) \leq^{A*P} (*goal* HPP_{s,t}[a, b], δ) with the expansion satisfying:

- $c_1(\varepsilon) = \frac{(b-a)\varepsilon}{b} + \frac{a}{b}$ if *goal = Max*
- $c_2(\varepsilon) = \frac{a}{b - (b-a)\varepsilon}$ if *goal = Min*

Proof : We only prove the *goal = Max* case. Let I be an instance and μ be a Hamiltonian path from s to t . If $m[I, \mu] \geq \varepsilon opt(I) + (1-\varepsilon)wor(I)$, then $m[I, \mu] \geq c_1(\varepsilon)opt(I)$ since $wor(I) \geq \frac{a}{b}opt(I)$. \square

The Proposition 3.2 and the Theorem 3.3 also hold for HPP_s and more generally, these results work for many specific optimization problems from graph theory, those for which all feasible solutions have an equal size that depends on the instance size (see Monnot [27]).

MAX HPP_{s,t}[a, b] and MIN HPP_{s,t}[a, b] (for a and b not depending on the instance) are trivially in **APX** when $a > 0$ since any solution is at least a a/b -standard approximation (take $\varepsilon = 0$ in

Theorem 3.3); in this case, the standard ratio may not be that meaningful since even a worst solution yields a constant standard approximation. Nevertheless, we can deduce from this theorem that the hardness thresholds for standard and differential framework are identical since $\text{MIN HPP}_{s,t}[a, b]$ is **APX-complete**.

Corollary 3.4 *For all $b > a \geq 0$, $\text{HPP}_{s,t}[a, b] \notin \text{PTAS}(\delta)$ unless $P=NP$.*

We can also establish a limit on its differential approximation for some values of a and b . Recall the negative result of Engebretsen and Karpinski [14] for $\text{MIN TSP}[1, 2]$: for any $\epsilon > 0$, no polynomial time algorithm can guarantee a standard approximation ratio greater than, or equal to, $\frac{740}{741} + \epsilon$. It is easy to observe that $\text{MIN HPP}_{s,t}[1, 2]$ and $\text{MIN HPP}_s[1, 2]$ are (asymptotically) equivalent to approximate $\text{MIN TSP}[1, 2]$. Thus, we can deduce that $\text{MIN HPP}_s[1, 2]$ and $\text{MIN HPP}_{s,t}[1, 2]$ are not standard approximable with ratio greater than $\frac{740}{741}$. Finally, using Theorem 3.3 and (iv) of Proposition 3.2, we obtain:

Proposition 3.5 *For all a , $\text{HPP}_{s,t}[a, a+1]$ and $\text{HPP}_s[a, a+1]$ are not approximable with differential ratio greater than $\frac{741}{742}$ unless $P=NP$.*

4 Approximate algorithms for these problems

In this section, we propose two types of algorithms which yield constant differential-ratio. For MAX HPP_s , the algorithm is obtained by getting several feasible solutions and by choosing the best one among them. Each of these individual solutions has a differential approximation ratio tending towards zero with the size of the instance. For $\text{MAX HPP}_{s,t}$, the algorithm is very different and takes into account the extreme solutions. So, on the one hand, the algorithm tries to be the nearest from the best solution value and on the other hand, tries to be the furthest from the worst solution value. In order to do that, it iteratively provides a solution of value greater than $(\text{wor}(I_j) + \text{opt}(I_j))/2$, where I_j is the sub-graph built at step j .

4.1 The algorithm for two specified endpoints version

$\text{MAX HPP}_{s,t}$ can also be regarded as the problem of determining a Hamiltonian cycle that contains edge (s, t) . The algorithm works by finding a maximum weight 2-matching among 2-matchings containing (s, t) and at each step, merging the cycles two by two. The main idea consists in pointing out that we could have lost much more by merging the two cycles in a different way. Thus, we will build dynamically another solution which approximate the worst solution; this solution will actually depends on the choices made by the algorithm at each iteration.

Consider two cycles C_i and two edges $(x_1, x_2) \in C_1$ and $(y_1, y_2) \in C_2$, we call localchange_i for $i = 1, 2$ the following process:

$$\text{localchange}_i[(C_1, (x_1, x_2)), (C_2, (y_1, y_2))] = \{(x_1, y_{3-i}), (x_2, y_i)\} \cup (C_1 \cup C_2 \setminus \{(x_1, x_2), (y_1, y_2)\})$$

These two processes merge the cycles C_1 and C_2 into a single cycle (see the Figure 1 for an illustration.). Note that the vertex order is important in the processes; thus, edges (x_1, x_2) or (y_1, y_2)

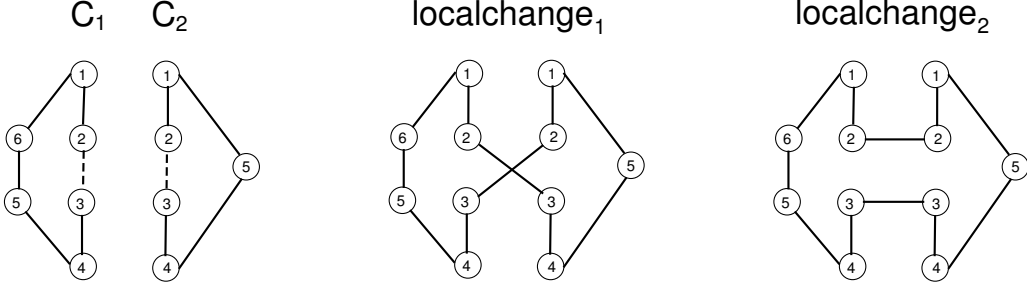


Figure 1: The $localchange_i$ processes between the edge $(2, 3)$ of C_1 and the edge $(2, 3)$ of C_2 .

are implicitly given as directed edges and we have: $localchange_1[(C_1, (x_2, x_1)), (C_2, (y_1, y_2))] = localchange_2[(C_1, (x_1, x_2)), (C_2, (y_1, y_2))]$. Moreover, when $C_1 = C_2$ and (x_1, x_2) is not adjacent to (y_1, y_2) , these processes simply amount to local edge swaps. We associate with $localchange_i$ a function $cost_i$ that represents the loss in merging two cycles:

$$cost_i[(x_1, x_2), (y_1, y_2)] = d(x_1, x_2) + d(y_1, y_2) - d(x_1, y_{3-i}) - d(x_2, y_i)$$

[LocalchangeHPP $_{s,t}$]

input: An instance (n, s, t, d) ;

output : A Hamilton path sol from s to t ;

Change the cost of (s, t) into $|V|d_{max} + 1$. Call this function d' ;

Compute a maximum weight 2-matching $M = \{C_i, i = 1, \dots, k\}$ of (n, d') ;

Suppose that $(s, t) \in C_1$

Choose 2 consecutive edges (x_1^1, x_2^1) and (x_2^1, x_3^1) in C_1 different from (s, t) ;

$sol_1 = C_1 \setminus \{(s, t)\}$, $e_1^1 = (x_1^1, x_2^1)$ and $e_2^1 = (x_2^1, x_3^1)$;

For $i=2$ to k do

Choose 2 consecutive edges (x_1^i, x_2^i) and (x_2^i, x_3^i) in C_i ;

If $cost_1[e_1^{i-1}, (x_1^i, x_2^i)] \leq cost_2[e_2^{i-1}, (x_2^i, x_3^i)]$ then

$sol_i = localchange_1[(sol_{i-1}, e_1^{i-1}), (C_i, (x_1^i, x_2^i))]$;

Suppose $e_1^{i-1} = (x, y)$, and x_0^i is the other neighbor of x_1^i in C_i

Set $e_1^i = (y, x_1^i)$ and $e_2^i = (x_1^i, x_0^i)$;

Else

$sol_i = localchange_2[(sol_{i-1}, e_2^{i-1}), (C_i, (x_2^i, x_3^i))]$;

Suppose $e_2^{i-1} = (x, y)$, and x_4^i is the other neighbor of x_3^i in C_i

Set $e_1^i = (y, x_3^i)$ and $e_2^i = (x_3^i, x_4^i)$;

End if ;

End for i ;

$sol = sol_k$;

As this algorithm is polynomial, let us then show that sol is an Hamiltonian path. Firstly, note that by construction, (s, t) belongs to every maximum weight 2-matching of (n, d') . Moreover, e_1^i and e_2^i obviously belong to sol_i for every iteration $i \leq k$ of the algorithm. These two facts lead to the result. A description of the algorithm is given in the Figure 2 when $M = \{C_i : i = 1, 2, 3\}$ with $|C_1| = 6$, $|C_2| = 4$ and $|C_3| = 5$.

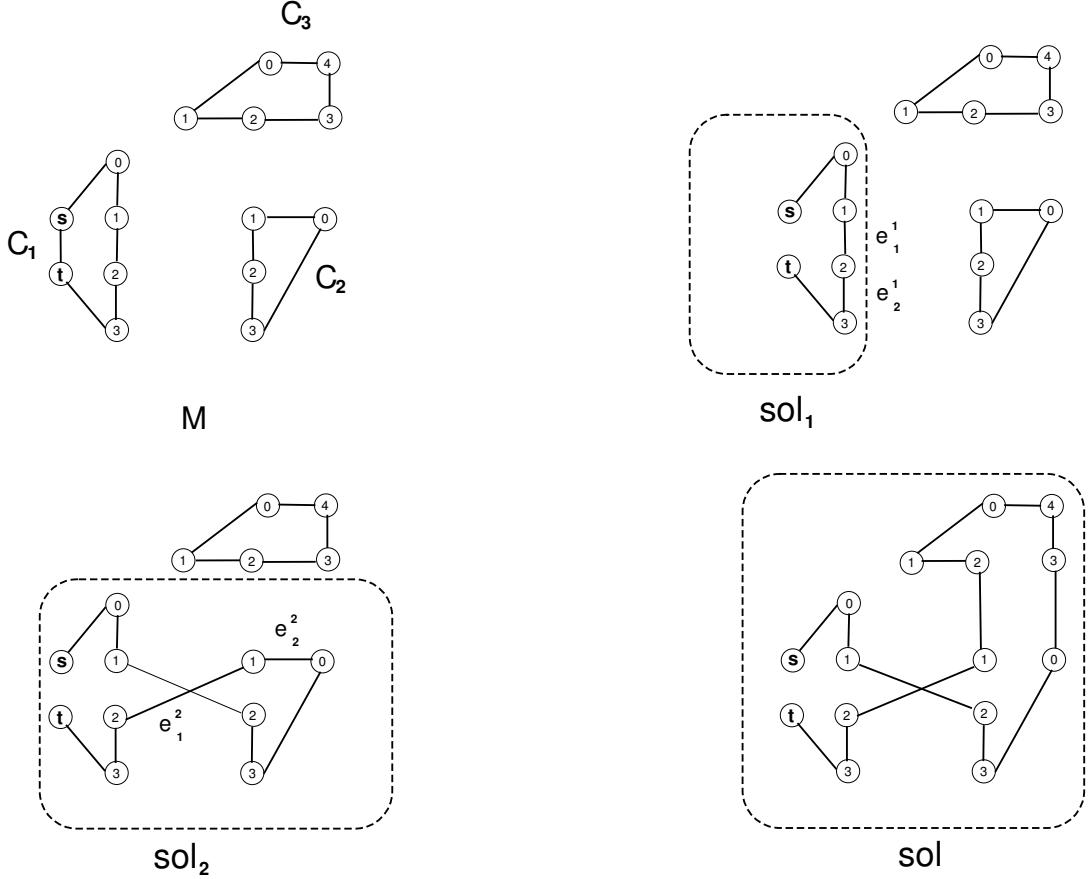


Figure 2: The 2-Matching M and the different iterations of algorithm when $k = 3$.

Theorem 4.1 *The algorithm $[LocalchangeHPP_{s,t}]$ is a $\frac{1}{2}$ -differential approximation for $\text{MAX HPP}_{s,t}$ and this ratio is tight.*

Proof: Given $I = (n, s, t, d)$, an instance of $\text{MAX HPP}_{s,t}$, we denote (i_2, \dots, i_k) with $i_j \in \{1, 2\}$ the sequence of choices produced by the algorithm such that, for $j \in \{2, \dots, k\}$:

$$sol_j = localchange_{i_j}[(sol_{j-1}, e_{i_j}^{j-1}), (C_j, (x_{i_j}^j, x_{i_j+1}^j))]$$

Thus, $d(sol_j) = d(sol_{j-1}) + d(C_j) - cost_{i_j}(j)$ with $cost_{i_j}(j) = cost_{i_j}[e_{i_j}^{j-1}, (x_{i_j}^j, x_{i_j+1}^j)]$. Summing

up these equalities for $j = 2$ to k , and since $d(sol_1) = d(C_1) - d(s, t)$ and $d(sol) = d(sol_k)$, we obtain:

$$d(sol) = d(M) - d(s, t) - \sum_{j=2}^k cost_{i_j}(j) \quad (4.1)$$

The main idea is to note that edge-subset $\{e_{3-i_j}^{j-1}, (x_{3-i_j}^j, x_{4-i_j}^j) : j = 2, \dots, k\}$ belongs to solution sol_k . Hence, we can "damage" the current solution by local edges-swap from this edge-subset. More formally, consider solutions sol'_j defined by $sol'_1 = sol_k$ and for $j = 2, \dots, k$,

$$sol'_j = localchange_{3-i_j}[(sol'_{j-1}, e_{3-i_j}^{j-1}), (sol'_{j-1}, (x_{3-i_j}^j, x_{4-i_j}^j))]$$

An illustration of solutions sol'_i with $i \leq k$ is depicted in the Figure 3 for the example described in Figure 2.

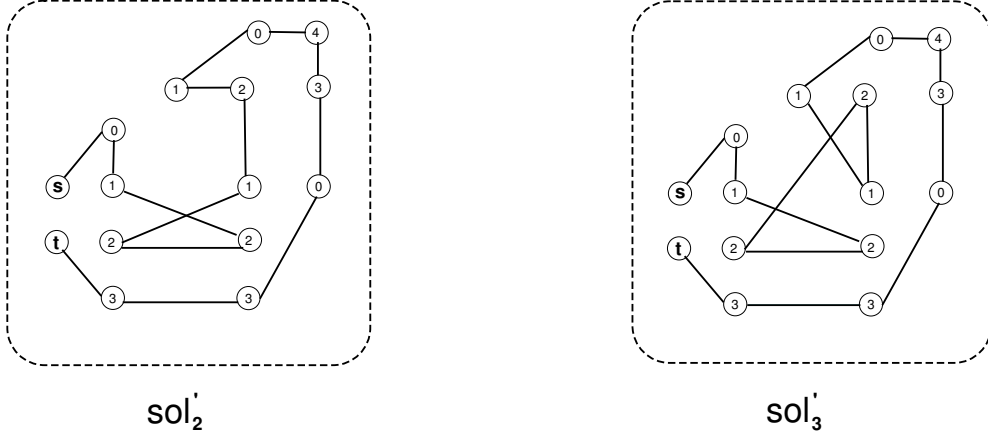


Figure 3: The solutions sol'_2 and sol'_3 .

Lastly, proceeding as previously, we obtain $d(sol'_k) = d(M) - d(s, t) - \sum_{j=2}^k (cost_{i_j}(j) + cost_{3-i_j}(j))$. By construction, $cost_{i_j}(j) + cost_{3-i_j}(j) \geq 2cost_{i_j}(j)$ and $wor(I) \leq d(sol'_k)$. Hence:

$$wor(I) \leq d(M) - d(s, t) - 2 \sum_{j=2}^k cost_{i_j}(j) \quad (4.2)$$

M is an optimal weight 2-matching among the 2-matching of (n, d) containing the edge (s, t) ; thus

$$opt(I) \leq d(M) - d(s, t) \quad (4.3)$$

By combining expressions (4.3), (4.2) and (4.1), we obtain:

$$d(sol) \geq \frac{1}{2}opt(I) + \frac{1}{2}wor(I)$$

We now show that this ratio is tight. Let $J_n = (n, s, t, d)$ be an instance defined by: $V = (\{x_i^j, 1 \leq i \leq 3, 2 \leq j \leq 2n + 1\} \cup \{s, u, t\})$, $d(x_1^j, x_1^{j+1}) = d(x_1^j, x_2^{j+1}) = 1 \forall j = 2, \dots, 2n$,

$d(x_1^j, x_2^{j+2}) = 1 \forall j = 2, \dots, 2n - 1$, $d_n(s, x_2^2) = d_n(u, x_1^2) = d(u, x_2^2) = d(t, x_3^2) = 1$ and let the cost of all other edges be two. The 2-matching is composed of $C_1 = \{s, u, t\}$ and $C_j = \{x_1^j, x_2^j, x_3^j\}$ $j = 2, \dots, 2n + 1$. The edges produced by the algorithm are: $e_1^1 = (s, u)$, $e_2^1 = (u, t)$, $e_1^2 = (u, x_1^2)$, $e_2^2 = (x_1^2, x_3^2)$, $e_1^j = (x_1^{j-1}, x_1^j)$, $e_2^j = (x_1^j, x_3^j)$ $j = 3, \dots, 2n + 1$ and $cost_1(2) = cost_2(2) = 2$, $cost_1(j) = cost_2(j) = 1$ $j = 3, \dots, 2n + 1$.

$$d(sol) = 10n + 4, \quad wor(J_{2n+1}) = 8n + 3, \quad opt(J_{2n+1}) = 12n + 4$$

Thus, we obtain that $\delta_{LocalchangeHPP_{s,t}}(J_{2n+1})$ approaches $\frac{1}{2}$ as n goes to infinity. \square

For the standard ratio, we deduce two new improved results by using Lemma 1.6 from the general case and Theorem 3.3 with $b = 2a$ for the case where the weights of the graph are bounded between the values a and $2a$.

Corollary 4.2 *We have the following results:*

- MAX HPP_{s,t} is $\frac{1}{2}$ -standard approximable and Max HPP_{s,t}[$a, 2a$] is $\frac{3}{4}$ -standard approximable.
- MIN HPP_{s,t}[$a, 2a$] is $\frac{2}{3}$ -standard approximable.

4.2 The algorithm for one specified endpoint version

We propose an algorithm which differs from the one previously studied since we explicitly compute several solutions. Our algorithm is based upon a simple idea and uses structural properties of solutions. It still works by finding a maximum weight 2-matching containing specified edges and then discarding some edges and arbitrarily connecting the resulting paths to form an Hamiltonian path from s . The principle of our algorithm is to generate not only one but several feasible solutions following this method.

Consider a maximum weight 2-matching M_r among those containing (s, r) , including elementary cycles C_i , $i = 1, \dots, k$. In order to do that, we substitute $|V|d_{max} + 1$ for the cost of (s, r) and we compute a maximum 2-matching in this new instance. Lastly, for each cycle C_i , we consider four consecutive vertices $x_1^i, x_2^i, x_3^i, x_4^i$. Note that we have numbered vertices such that $x_1^1 = r$ and $x_2^1 = s$. Moreover, if $|C_i| = 3$ then $x_4^i = x_1^i$. For the last cycle C_k , we consider an additional vertex y which is the other neighbor of x_1^k in C_k . Thus, if $|C_k| = 4$ then $y = x_4^k$ while y is a new vertex in the other case.

[Patching 2 – matching]

input: An instance (n, s, d) ;

output: A Hamiltonian path sol from s ;

For every $r \in V \setminus \{s\}$ do

Change the cost of (s, r) into $|V|d_{max} + 1$. Call this function d' ;

Compute a maximum weight 2-matching $M_r = \{C_i, i = 1, \dots, k\}$ of (n, d') ;

if $k = 1$ then $sol_r = M_r \setminus \{(s, r)\}$;

if k is even then

$S_1 = \cup_{j=1}^{k-1} \{(x_2^j, x_3^j)\} \cup \{(x_1^k, x_2^k), (s, r)\}$;
 Build $sol_1 = (M_r \setminus S_1) \cup \{(x_1^k, x_3^1), (x_2^1, x_2^2)\} \cup_{j=1}^{(k-2)/2} \{(x_3^{2j}, x_3^{2j+1}), (x_2^{2j+1}, x_2^{2j+2})\}$;
 (sol_1 is a Hamiltonian path from s to r)
 $S_2 = \cup_{j=2}^{k-1} \{(x_1^j, x_2^j)\} \cup \{(y, x_1^k), (s, r)\}$;
 Build $sol_2 = (M_r \setminus S_2) \cup \{(x_1^1, x_1^2)\} \cup_{j=1}^{(k-2)/2} \{(x_2^{2j}, x_2^{2j+1}), (x_1^{2j+1}, x_1^{2j+2})\}$;
 (sol_2 is a Hamiltonian path from s to y)
 $S_3 = \cup_{j=1}^{k-1} \{(x_3^j, x_4^j)\} \cup \{(x_2^k, x_3^k), (s, r)\}$;
 Build $sol_3 = (M_r \setminus S_3) \cup \{(x_2^k, x_4^1), (x_3^1, x_3^2)\} \cup_{j=1}^{(k-2)/2} \{(x_4^{2j}, x_4^{2j+1}), (x_3^{2j+1}, x_3^{2j+2})\}$;
 (sol_3 is an Hamiltonian path from s to r)

End if ;

if k is odd then

$S_1 = \cup_{j=1}^k \{(x_2^j, x_3^j)\} \cup \{(s, r)\}$;
 Build $sol_1 = (M_r \setminus S_1) \cup \{(x_2^k, x_3^1)\} \cup_{j=1}^{(k-1)/2} \{(x_2^{2j-1}, x_2^{2j})(x_3^{2j}, x_3^{2j+1})\}$;
 (sol_1 is an Hamiltonian path from s to r)
 $S_2 = \{(s, r)\} \cup_{j=2}^k \{(x_1^j, x_2^j)\}$;
 Build $sol_2 = (M_r \setminus S_2) \cup_{j=1}^{(k-1)/2} \{(x_1^{2j-1}, x_1^{2j}), (x_2^{2j}, x_2^{2j+1})\}$;
 (sol_2 is a Hamiltonian path from s to x_1^k)
 $S_3 = \cup_{j=1}^k \{(x_3^j, x_4^j)\} \cup \{(s, r)\}$;
 Build $sol_3 = (M_r \setminus S_3) \cup \{(x_3^k, x_4^1)\} \cup_{j=1}^{(k-1)/2} \{(x_3^{2j-1}, x_3^{2j}), (x_4^{2j}, x_4^{2j+1})\}$;
 (sol_3 is a Hamiltonian path from s to r)

End if ;

$sol_r = \operatorname{argmax}\{d(sol_1), d(sol_2), d(sol_3)\}$;

End for r ;

$sol = \operatorname{argmax}\{d(sol_r) : r \in V \setminus \{s\}\}$;

Observe that for every r , the solutions sol_1 , sol_2 and sol_3 are Hamiltonian paths (from s to different endpoints) since the additional edges are adjacent to the ones substituted. A description of solutions sol_1 , sol_2 , sol_3 is given in the Figure 4 when $M_r = \{C_i : i = 1, 2, 3\}$ with $|C_1| = |C_3| = 6$ and $|C_2| = 3$.

The time-complexity of this algorithm remains polynomial since the computation of the 2-matching problem is polynomial.

Theorem 4.3 *The algorithm [Patching 2 – matching] is a $\frac{2}{3}$ -differential approximation for MAX HPP $_s$ and this ratio is tight.*

Proof: Let $I = (n, s, d)$ be an instance and let sol^* be an optimal Hamiltonian path from s to r^* . We denote $loss_i$, $i = 1, 2, 3$, the quantity $d(sol_i) - d(M_{r^*}) + d(s, r^*)$. Obviously, $loss_i \leq 0$ and we have

$$d(sol) \geq d(sol_r^*) \geq d(M_{r^*}) - d(s, r^*) + \frac{1}{3}(loss_1 + loss_2 + loss_3) \quad (4.4)$$

Moreover, the following structural property holds:

$sol_* = \cup_{j=1,2,3} (sol_j \setminus M_{r^*}) \cup M_{r^*} \setminus (S_1 \cup S_2 \cup S_3)$ is a Hamiltonian path starting from s

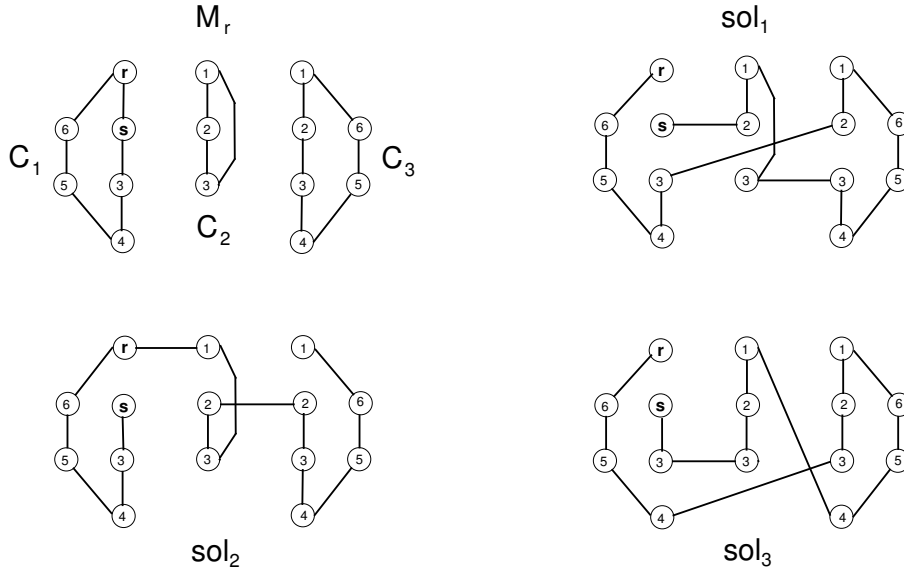


Figure 4: The 2-Matching M_r and the solutions sol_1 , sol_2 and sol_3 when $k = 3$.

A description of solution sol_* is depicted in the Figure 5 for the example described in Figure 4.

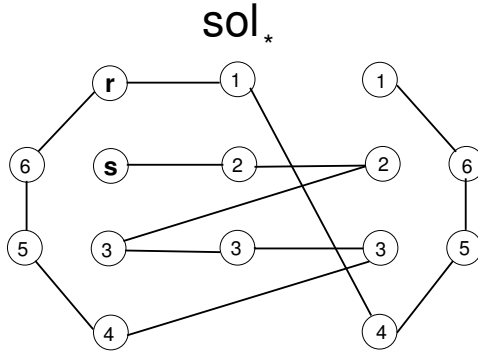


Figure 5: The solution sol_* .

$d(sol_*) = d(M_{r^*}) - d(s, r^*) + loss_1 + loss_2 + loss_3$ since $d(sol_j \setminus M_{r^*}) = loss_j + d(S_j) - d(s, r^*)$ and $d(M_{r^*} \setminus (S_1 \cup S_2 \cup S_3)) = d(M_{r^*}) - d(S_1) - d(S_2) - d(S_3) + 2d(s, r^*)$. Hence, we deduce

$$wor(I) \leq d(M_{r^*}) - d(s, r^*) + loss_1 + loss_2 + loss_3 \quad (4.5)$$

Since $sol^* \cup (s, r^*)$ is a particular 2-matching containing (s, r^*) , we have:

$$opt(I) \leq d(M_{r^*}) - d(s, r^*) \quad (4.6)$$

Lastly, combining (4.4),(4.5) and (4.6) we obtain:

$$d(sol) \geq \frac{1}{3}wor(I) + \frac{2}{3}opt(I)$$

To show that the bound is approachable, consider the following instances. Let $I_n = (n, s, d)$ be an instance defined by: $V = \{x_i^j : 1 \leq i \leq 3, 1 \leq j \leq 2n + 1\}$ with $x_2^1 = s$, $d(x_1^j, x_2^j) = d(x_3^j, x_2^j) = d(x_1^j, x_3^j) = 2, \forall j = 1, \dots, 2n + 1$, $d(x_1^j, x_3^{j+1}) = 2, \forall j = 1, \dots, 2n$ and $d(x_1^1, x_3^j) = d(x_1^1, x_2^j) = 2, \forall j = 2, \dots, 2n + 1$. Let the cost of all other edges be one. We have:

$$d(sol) \leq 10n + 4, \quad opt(I_n) = 12n + 4, \quad wor(I_n) = 6n + 2$$

leading to the conclusion that $\delta_{Patching \ 2-matching}(I_n)$ approaches $\frac{2}{3}$ as n goes to infinity. \square

As previously, we deduce two new improved standard approximation results by using Lemma 1.6 from the general case and Theorem 3.3 with $b = 2a$ when $a \leq d(e) \leq 2a$.

Corollary 4.4 *We have the following results:*

- MAX HPP_s is $\frac{2}{3}$ -standard approximable and MAX HPP_s[a, 2a] is $\frac{5}{6}$ -standard approximable.
- MIN HPP_s[a, 2a] is $\frac{3}{4}$ -standard approximable.

5 Conclusion and open problems

In this paper, we have mainly provided new results concerning the approximability of the Hamiltonian path problem in the case in which one or two endpoints are specified. Moreover, we have exposed some basic properties (mainly by using reductions preserving differential approximation) between these problems and some variants of them.

Although in introduction we have pointed out some great difference between the differential and standard approximability of these problems, when we use bounded metric and especially, when the weights in the graph are between the values a and $2a$, from differential approximation results we can derive new standard approximation results.

An interesting open problem under differential framework is to know if the two-specified endpoints version is really more difficult to approximate than the one-specified endpoint version (we only know that HPP_{s,t} is at least as hard as HPP_s). This question is still open under standard framework. The positive approximation results indicate a positive answer but it is not a formal proof. A formal proof would show that the differential non-approximation threshold for HPP_{s,t} is strictly better than the differential non-approximation threshold for HPP_s. In order to prove that, a useful technique is to prove that a problem is not *simple*. Recall that an **NPO** problem is called *simple* by Paz and Moran [32] if its restriction to instances satisfying for any fixed integer k , $opt(I) \leq k$, can be resolved within polynomial time. So, we can also prove a standard non-approximation threshold equal to $\frac{2}{3}$ for the Bin-Packing problem because its restriction to instances verifying $opt(I) \leq 2$ is still **NP-hard** problem. Similarly, we will say that π is δ -*simple* if its restriction π_k to instances verifying for any integer fixed k , $|wor(I) - opt(I)| \leq k$ can be solved in polynomial time. Thus, if the sub-problem verifying $|wor(I) - opt(I)| \leq k_0$ is **NP-hard** (in other words, π is not δ -*simple*)

then for any $\epsilon > 0$, no polynomial time algorithm can guarantee a differential approximation ratio greater than, or equal to $\frac{k_0}{k_0+1} + \epsilon$. We conjecture that $\text{HPP}_{s,t}$ and HPP_s are not δ -simple and the value k_0 found for $\text{HPP}_{s,t}$ is smaller than the k_0 found for HPP_s .

Acknowledgments. Many thanks to three anonymous referees for pertinent and useful comments and suggestions.

References

- [1] J. A. HARTIGAN. *Clustering algorithms*. New York, Wiley, 1975.
- [2] A. AIELLO, E. BURATTINI, M. MASSAROTTI, and F. VENTRIGLIA. A new evaluation function for approximation. *Sem. IRIA*, 1977.
- [3] G. AUSIELLO, P. CRESCENZI, and M. PROTASI. Approximate solutions of NP-optimization problems. *Theoretical Computer Science*, 150:1–55, 1995.
- [4] G. AUSIELLO, A. D’ATRI, and M. PROTASI. Structure preserving reductions among convex optimization problems. *J. of Computer and System Science*, 21:136–153, 1980.
- [5] C. BAZGAN, R. HASSIN, and J. MONNOT. Differential approximation for some routing problems (extended abstract). *Proc. CIAC 03, LNCS 2653*, pages 277–288, 2003.
- [6] N. CHRISTOFIDES. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report 338, Grad. School of Industrial Administration, CMU, 1976.
- [7] G. CORNUEJOLS, M. L. FISHER, and G. L. NEMHAUSER. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23:789–810, 1977.
- [8] P. CRESCENZI and V. KANN. A compendium of NP-optimization problems. Available on www address: <http://www.nada.kth.se/~viggo/problemlist/compendium.html>, 2000.
- [9] P. CRESCENZI and A. PANCONESI. Completeness in approximation classes. *Information and Computation*, 93:241–262, 1991.
- [10] M. DEMANGE, D. DE WERRA, J. MONNOT, and V. Th. PASCHOS. Weighted node coloring: when stable sets are expensive (extended abstract). *Proc. WG 02, LNCS 2573*, pages 114–125, 2002.
- [11] M. DEMANGE, P. GRISONI, and V. Th. PASCHOS. Differential approximation algorithms for some combinatorial optimization problems. *Theoretical Computer Science*, 209:107–122, 1998.
- [12] M. DEMANGE, J. MONNOT, and V. Th. PASCHOS. Bridging gap between standard and differential polynomial approximation: the case of bin-packing. *Applied Mathematics Letters*, 12:127–133, 1999.

- [13] M. DEMANGE and V. Th. PASCHOS. On an approximation measure founded on the links between optimization and polynomial approximation theory. *Theoretical Computer Science*, 158:117–141, 1996.
- [14] L. ENGBRETTSEN and M. KARPINSKI. Approximation hardness of TSP with bounded metrics. *Manuscript available on www address: <http://www.nada.kth.se/~enge/enge-bib/>*, 2002.
- [15] M. L. FISHER, G. L. NEMHAUSER, and L. A. WOLSEY. An analysis of approximations for finding a maximum weight hamiltonian circuit. *Operations Research*, 27(4):799–809, 1979.
- [16] M. R. GAREY and D. S. JOHNSON. *Computers and intractability. A guide to the theory of NP-completeness*. CA. Freeman, 1979.
- [17] R. S. GARFINKEL. Motivation and modeling. E. L LAWLER, J. K. LENSTRA, A. H. G. RINNOOY KAN AND D. B. SHMOYS (EDS.) *The Traveling Salesman Problem: a guided tour of Combinatorial Optimization Wiley Chichester*, pages 17–36, 1985.
- [18] N. GUTTMANN-BECK, R. HASSIN, S. KHULLER, and B. RAGHAVACHARI. Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica*, 28(4):422–437, 2000.
- [19] M. HAIMOVICH and A. H. G. RINNOOY KAN. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(5):527–542, 1985.
- [20] R. HASSIN and S. KHULLER. z -approximations. *J. of Algorithm*, 41(2):429–442, 2001.
- [21] R. HASSIN and S. RUBINSTEIN. Better approximation for max TSP. *Information Processing Letters*, 75, 2000.
- [22] J. A. HOOGEVEEN. Analysis of Christofides’ heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10:291–295, 1991.
- [23] D. S. JOHNSON and C. H. PAPADIMITRIOU. *Performance guarantees for heuristics*. E. L LAWLER, J. K. LENSTRA, A. H. G. RINNOOY KAN AND D. B. SHMOYS (EDS) *The Traveling Salesman Problem: a guided tour of Combinatorial Optimization WILEY CHICHESTER* 145–180, 1985.
- [24] R. M. KARP. *Reducibility among combinatorial problems*. R. E MILLER and J. W. THATCHER, Complexity of Computer Computations, 85-103, Plenum Press, NY, 1972.
- [25] R. KOSARAJU, J. PARK, and C. STEIN. tours and short superstrings. *Proc. FOCS*, 11(4):166–177, 1994.
- [26] E. L. LAWLER, J. K. LENSTRA, A. H. G. RINNOOY KAN, and D. B. SHMOYS (EDS.). *The traveling salesman problem: A guided tour of combinatorial optimization*. Wiley, New York, 1985.

- [27] J. MONNOT. Differential approximation of NP-hard problems with constant size feasible solutions. *RAIRO/Operations Research*, 36(4):279–297, 2002.
- [28] J. MONNOT. The maximum f -depth spanning tree problem. *Information Processing Letters*, 80:179–187, 2001.
- [29] J. MONNOT. Approximation results toward Nearest Neighbor heuristic. *Yugoslav Journal of Operations Research*, 12(1):11–16, 2002.
- [30] P. ORPONEN and H. MANNILA. On approximation preserving reductions: complete problems and robust measures. Technical report, Department of Computer Science, University of Helsinki, 1987.
- [31] C. H. PAPANITRIOU and M. YANNAKAKIS. The traveling salesman problem with distance one and two. *Mathematics of Operations Research*, 18:1–11, 1993.
- [32] A. PAZ and S. MORAN. Non deterministic polynomial optimisation problems and their approximation. *Theoretical Computer Science*, 95:251–277, 1981.
- [33] S. SAHNI and T. GONZALEZ. P-complete approximation problems. *J. of ACM*, 23:555–565, 1976.
- [34] J. TARHIO and E. UKKONEN. A greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Computer Science*, 57:131–145, 1988.
- [35] S. A. VAVASIS. Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, 57:279–311, 1972.
- [36] E. ZEMEL. Measuring the quality of approximate solutions to zero-one programming problems. *Mathematics of Operations Research*, 6:319–332, 1981.