Differential approximation(septembre 2005)

Giorgio Ausiello, Vangelis Th. Paschos

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision

CNRS UMR 7024

CAHIER DU LAMSADE

228

Septembre 2005

Differential approximation

Giorgio AUSIELLO, Vangelis Th. PASCHOS

Differential approximation*

Giorgio Ausiello[1]   Vangelis Th. Paschos[2]

[1] Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma "La Sapienza"
`ausiello@dis.uniroma1.it`

[2] LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine
`paschos@lamsade.dauphine.fr`

September 15, 2005

Abstract

In this paper, we survey the fundamentals of the differential approximation theory, as well as some operational and structural results.

1 Introduction

In this chapter, we use the notations introduced in chapter R-7. Also, given an approximation algorithm $\mathtt{A}$ for an **NPO** problem Π, we denote by $m_{\mathtt{A}}(x, y)$, the value of the solution y computed by $\mathtt{A}$ on instance x of Π. When clear from the context, reference to $\mathtt{A}$ will be omitted.

Very frequently, the commonly used approximation measure (called *standard approximation ratio* in what follows) may not be very meaningful, in particular if the ratio of $m(x, y_w)$, the worst solution's value for x, to $\mathrm{opt}(x)$ is already bounded (above, if $\mathrm{goal}(\Pi) = \min$, below, otherwise). Consider, for instance, the seminal maximal matching algorithm for MIN VERTEX COVER (the definitions of the most of the problems dealt in this chapter can be found in [4, 35]; also, for graph-theoretic notions, the interested reader can be referred to [13]), that achieves approximation ratio 2 for this problem. There, given a graph $G(V, E)$, a maximal[1] matching M of G is computed and the endpoints of the edges in M are added in the solution for MIN VERTEX COVER. If M is perfect (almost any graph, even relatively sparse, admits a perfect matching [14]), then the whole of V has been included in the cover built, while an optimal cover contains at least a half of V. So, the absolutely worst solution (that one could compute without using any algorithm) achieves in most cases approximation ratio 2.

The remark above is just one drawback of the standard approximation ratio. Several other drawbacks have been also observed, for instance, the artificial dissymmetry between "equivalent" minimization and maximization problems (for example, MAX CUT and MIN CLUSTERING, see [55]) introduced by the standard approximation ratio. The most blatant case of such dissymmetry is the one appearing when dealing with the approximation of MIN VERTEX COVER and MAX INDEPENDENT SET (given a graph, a vertex cover is the complement of an independent set with respect to the vertex set of the graph). In other words, using linear programming vocabulary, the objective function of the former is an affine transformation of the objective function of the latter.

---

*Preliminary version of Chapter 13 to appear in T. Gongalez (ed.), *Approximation algorithms and metaheuristics*, Taylor and Francis.

[1] With respect to the inclusion.

This equivalence under such simple affine transformation is not reflected to the approximability of these problems: the former is approximable within constant ratio, in other words it is in **APX** (see [4, 53] for definition of the approximability classes dealing with the standard paradigm; the ones of the differential paradigm are defined analogously), while the latter is inapproximable within ratio $\Omega(n^{\epsilon-1})$, for any $\epsilon > 0$ (see [41]). In other words, the standard approximation ratio is unstable under affine transformations of the objective function.

In order to remedy to these phenomena, several researchers have tried to adopt alternative approximation measures not suffering from these inconsistencies. One of them is the ratio $\delta(x, y) = (\omega(x) - m(x, y))/(\omega(x) - \mathrm{opt}(x))$, called differential ratio in the sequel, where $\omega(x)$ is the value of a worst solution for x, called *worst value*. It will be formally dealt in the next sections. It has been used rather punctually and without following a rigorous axiomatic approach until the paper [29] where such an approach is introduced. To our knowledge, differential ratio is introduced in [5] in 1977, and [6, 1, 60] are, to our knowledge, the most notable cases in which this approach has been applied. It is worth noting that in [60], a weak axiomatic approach is also presented.

Finally, let us note that several other authors that have also recognized the methodological problems implied by the standard ratio, have proposed other alternative ratios. It is interesting to remark that the main such ratios are very close, although with some small or less small differences, to the differential ratio. For instance, in [17], for studying MAX TSP, it is proposed the ratio $d(x, y, z_r) = |\mathrm{opt}(x) - m(x, y)|/|\mathrm{opt}(x) - z_r|$, where z_r is a positive value computable in polynomial time, called *reference-value*. It is smaller than the value of any feasible solution of x, hence smaller than $\omega(x)$ (for a maximization problem a worst solution is the one of the smallest feasible value). The quantities $|\mathrm{opt}(x) - m(x, y)|$ and $|\mathrm{opt}(x) - z_r|$ are called *deviation*, and *absolute deviation*, respectively. The approximation ratio $d(x, y, z_r)$ depends on both x and z_r, in other words, there exist a multitude of such ratios for an instance x of an **NPO** problem, one for any possible value of z_r. Consider a maximization problem Π and an instance x of Π. Then, $d(x, y, z_r)$ is increasing with z_r, so, $d(x, y, z_r) \leqslant d(x, y, \omega(x))$. In fact, in this case, for any reference value z_r: $r(x, y) \geqslant 1 - d(x, y, z_r) \geqslant 1 - d(x, y, \omega(x)) = \delta(x, y)$, where r denotes the standard-approximation ratio for Π. When $\omega(x)$ is computable in polynomial time, $d(x, y, \omega(x))$ is the smallest (tightest) over all the d-ratios for x. In any case, if for a given problem, one sets $z_r = \omega(x)$, then $d(x, y, \omega(x)) = 1 - \delta(x, y)$ and both ratios have the natural interpretation of estimating the relative position of the approximate solution-value in the interval worst solution-value – optimal value.

2 Towards a new approximation paradigm

2.1 The differential approximation ratio

In [29], it is undertaken the task of adopting, in an axiomatic way, an approximation measure founded on both intuitive and mathematical links between optimization and approximation. It is claimed there that a "consistent" ratio must be *order preserving* (i.e., the better the solution the better the approximation ratio achieved) and *stable under affine transformation of the objective function*. Furthermore, it is proved that no ratio function of two parameters – for example, m, opt – can fit this latter requirement. Hence it is proposed what will be called *differential approximation ratio*[2] in what follows. Problems related by affine transformations of their objective functions are called *affine equivalent*.

Consider an instance x of an **NPO** problem Π and a polynomial time approximation algorithm A for Π, the differential approximation ratio $\delta_{\mathtt{A}}(x, y)$ of a solution y computed by A in x is

---

[2]This notation is suggested in [29]; another notation drawing the same measure is *z-approximation* suggested in [39].

defined by:

$$\delta_{\mathtt{A}}(x, y) = \frac{(\omega(x) - m_{\mathtt{A}}(x, y))}{(\omega(x) - \mathrm{opt}(x))}$$

where $\omega(x)$ is the value of a worst solution for x, called *worst value*. Note that for any goal, $\delta_{\mathtt{A}}(x, y) \in [0, 1]$ and, moreover, the closer $\delta_{\mathtt{A}}(x, y)$ to 1, the closer $m_{\mathtt{A}}(x, y)$ to $\mathrm{opt}(x)$. By definition, when $\omega(x) = \mathrm{opt}(x)$, i.e., all the solutions of x have the same value, then the approximation ratio is 1. Notice that, $m_{\mathtt{A}}(x, y)) = \delta_{\mathtt{A}}(x, y)\mathrm{opt}(x) + (1 - \delta_{\mathtt{A}}(x, y))\omega(x)$. So, differential approximation ratio measures how an approximate solution is placed in the interval between $\omega(x)$ and $\mathrm{opt}(x)$.

We note that the concept of the worst solution has a status similar to the one of the optimal solution. It depends on the problem itself and is defined in a non-constructive way, i.e., independently of any algorithm that could build it. The following definition for worst solution is proposed in [29].

Definition 1. Given an **NPO** problem $\Pi = (\mathcal{I}, \mathrm{Sol}, m, \mathrm{goal})$, a worst solution of an instance x of Π is defined as an optimal solution of a new problem $\bar{\Pi} = (\mathcal{I}, \mathrm{Sol}, m, \overline{\mathrm{goal}})$, i.e., of a **NPO** problem having the same sets of instances and of instances and of feasible solutions and the same value-function as Π but its goal is the inverse of the one of Π, i.e., $\overline{\mathrm{goal}} = \min$ if $\mathrm{goal} = \max$ and vice-versa. ∎

Example 1. The worst solution for an instance of MIN VERTEX COVER or of MIN COLORING is the whole vertex-set of the input-graph, while for an instance of MAX INDEPENDENT SET the worst solution is the empty set. On the other hand, if one deals with an MAX INDEPENDENT SET with the additional constraint that a feasible solution has to be maximal with respect to inclusion, the worst solution of an instance of this variant is a *minimum maximal independent set*, i.e., an optimal solution of a very well-known combinatorial problem, the MIN INDEPENDENT DOMINATING SET. Also, the worst solution for MIN TSP is a "heaviest" Hamiltonian cycle of the input-graph, i.e., an optimal solution of MAX TSP, while for MAX TSP the worst solution is the optimal solution of a MIN TSP. The same holds for the pair MAX SAT, MIN SAT. ∎

From Example 1, one can see that, although for some problems a worst solution corresponds to some trivial input-parameter and can be computed in polynomial time (this is, for instance, the case of MIN VERTEX COVER, MAX INDEPENDENT SET, MIN COLORING, etc.), there exist a lot of problems for which, determining a worst solution is as hard as determining an optimal one (as for MIN INDEPENDENT DOMINATING SET, MIN TSP, MAX TSP, MIN SAT, MAX SAT, etc.).

Remark 1. Consider the pair of affine equivalent problems MIN VERTEX COVER, MAX INDEPENDENT SET and an input-graph $G(V, E)$ of order n. Denote by $\tau(G)$ the cardinality of a minimum vertex cover of G and by $\alpha(G)$, the stability number of G. Obviously, $\tau(G) = n - \alpha(G)$. Based upon what has been discussed above, the differential ratio of some vertex cover C of G is $\delta(G, C) = (n - |C|)/(n - \tau(G))$. Since the set $S = V \setminus C$ is an independent set of G, its differential ratio is $\delta(G, S) = (|S| - 0)/(\alpha(G) - 0) = (n - |C|)/(n - \tau(G)) = \delta(G, C)$. ∎

As we have already mentioned, the differential ratio, although without systematic use or axiomatic approach, has been used at several times by many authors, before and after [29]. They use it in several contexts going from mathematical (linear or non-linear) programming [12, 49, 58] to pure combinatorial optimization [6, 7, 1, 39, 40], or they disguise differential approximation of a problem to standard approximation of affine transformations of it. For instance, in order to study differential approximation of BIN PACKING, one can deal with standard approximation of the problem of maximizing the number of unused bins; for MIN COLORING, the affinely equivalent problem is the one of maximizing the number of unused colors, for MIN SET COVER, this problem is the one of maximizing the number of unused sets, etc.

2.2 Asymptotic differential approximation ratio

In any approximation paradigm, the notion of asymptotic approximation (dealing, informally, with a class of "interesting" instances) is pertinent. In the standard paradigm, the asymptotic approximation ratio is defined on the hypothesis that the interesting (from an approximation point of view) instances of the simple problems are the ones whose values of the optimal solutions tend to ∞ (because, in the opposite case[3], these problems, called *simple* ([54]), are polynomial). In the differential approximation framework on the contrary, the size (or the value) of the optimal solution is not always a pertinent hardness criterion (see [31] for several examples about this claim). Henceforth, in [31], another hardness criterion, *the number $\sigma(x)$ of the feasible values of x*, has been used to introduce the *asymptotic differential approximation ratio*. Under this criterion, the asymptotic differential approximation ratio of an algorithm $\mathtt{A}$ is defined as

$$\delta_{\mathtt{A}}^{\infty}(x,y) = \lim_{k\to\infty} \inf_{\substack{x \\ \sigma(x)\geqslant k}} \left\{ \frac{\omega(x) - m(x,y)}{\omega(x) - \mathrm{opt}(I)} \right\} \tag{1}$$

Let us note that $\sigma(x)$ is motivated by, and generalizes, the notion of the *structure of the instance* introduced in [6]. We also notice that the condition $\sigma(x) \geqslant k$ characterizing "the sequence of unbounded instances" (or "limit instances") cannot be polynomially verified[4]. But in practice, for a given problem, it is possible to directly interpret condition $\sigma(x) \geqslant k$ by means of the parameters $\omega(x)$ and $\mathrm{opt}(x)$ (note that $\sigma(x)$ is not a function of these values). For example, for numerous cases of discrete problems, it is possible to determine, for any instance x, a step $\pi(x)$ defined as the least variation between two feasible values of x. For example, for BIN PACKING, $\pi(x) = 1$. Then,

$$\sigma(x) \leqslant \frac{(\omega(x) - \mathrm{opt}(x))}{\pi(x)} + 1$$

Therefore, from (1):

$$\delta_{\mathtt{A}}^{\infty}(x,y) \geqslant \lim_{k\to\infty} \inf_{\substack{x \\ \frac{\omega(x)-\mathrm{opt}(x)}{\pi(x)} \geqslant k-1}} \left\{ \frac{\omega(x) - m(x,y)}{\omega(x) - \mathrm{opt}(x)} \right\}$$

Whenever π can be determined, condition $(\omega(x) - \mathrm{opt}(x))/\pi(x) \geqslant k-1$ can be easier to evaluate than $\sigma(x) \geqslant k$, and in this case, the former condition is used (this is not senseless since we try to bound below the ratio).

The adoption of $\sigma(x)$ as hardness criterion can be motivated by considering a class of problems, called *radial problems* in [31], that includes many well-known combinatorial optimization problems, as BIN PACKING, MAX INDEPENDENT SET, MIN VERTEX COVER, MIN COLORING, etc. Informally, a problem Π is radial if, given an instance x of Π and a feasible solution y for x, one can, in polynomial time, on the one hand, deteriorate y as much as one wants (up to finally obtain a worst-value solution) and, on the other hand, greedily improve y in order to obtain (always in polynomial time) a sub-optimal solution (eventually the optimal one).

Definition 2. A problem $\Pi = (\mathcal{I}, \mathrm{Sol}, m, \mathrm{goal})$ is radial if there exist three polynomial algorithms ξ, ψ and φ such that, for any $x \in \mathcal{I}$:

1. ξ computes a feasible solution $y^{(0)}$ for x;

2. for any feasible solution y of x strictly better (in the sense of the value) than $y^{(0)}$, algorithm φ computes a feasible solution $\varphi(y)$ (if any) such that $m(x, \varphi(y))$ is strictly worse

---

[3]The case where optimal values are bounded by fixed constants.
[4]The same holds for the condition $\mathrm{opt}(x) \geqslant k$ induced by the hardness criterion in the standard paradigm.

than $m(x, y)$ (i.e., $m(x, \varphi(y)) > m(x, y)$, if $\text{goal}(\Pi) = \min$ and $m(x, \varphi(y)) < m(x, y)$, if $\text{goal}(\Pi) = \max$);

3. for any feasible solution y of x with value strictly better than $m(x, y^{(0)})$, there exists $k \in \mathbb{N}$ such that $\varphi^k(y) = y^{(0)}$ (where φ^k denotes the k-times iteration of φ);

4. for a solution y such that, either $y = y^{(0)}$, or y is any feasible solution of x with value strictly better than $m(x, y^{(0)})$, $\psi(y)$ computes the set of ancestors of y, defined by:

$$\psi(y) = \varphi^{-1}(\{y\}) = \{z : \varphi(z) = y\}$$

(this set being eventually empty) ▮.

Let us note that the class of radial problems includes in particular the well-known class of *hereditary* problems for which any subset of a feasible solution remains feasible. In fact, for an hereditary (maximization) problem, a feasible solutions y is a subset of the input-data, for any instances x, $y^{(0)} = \emptyset$ and for any other feasible solution y, $\varphi(y)$ is just obtained from y by removing a component of y. The hereditary notion deals with problems for which a feasible solution is a subset of the input-data, while the radial notion allows problems for which solutions are also second-order structures of the input-data.

Example 2. We show that BIN PACKING is radial. Consider a list L of n rational numbers; then,

- one can easily compute a solution $B^{(0)}$ for BIN PACKING consisting of putting an item per bin (so $B^{(0)}$ is a collection of n one-item bins); so, condition 1 is satisfied;

- given a feasible solutio B, one can deteriorate it by removing an item a from a bin of B containing at least two items and by putting a in a new (unused) bin; so, condition 2 is satisfied;

- one can continue this deterioration of B by repeatedly executing the above item until solution $B^{(0)}$ is obtained; so, condition 3 is satisfied;

- we show finally that condition 4 of definition 2 is also satisfied; consider either solution $B^{(0)}$ or any other feasible solution B

  - one can obtain better solutions by iteratively trying to empty some bins of $B^{(0)}$ (or B), i.e., by considering an single-item bin and by trying to place its item in another non-empty bin of $B^{(0)}$ (or B);

  - one can continue this procedure as far as it leads to smaller feasible BIN PACKING-solutions;

  - moreover, if one guesses successfully the single items to be moved, then one could obtain even an optimal BIN PACKING-solution;

In all, BIN PACKING satisfies all the items of Definition 2. ▮

Proposition 1. *([31]) Let κ be a fixed constant and consider a radial problem Π such that, for any instance x of Π of size n, $\sigma(x) \leqslant \kappa$. Then, Π is polynomial.*

3 Differential approximation results for some optimization problems

In general, no systematic way allows to link results obtained in standard and differential approximation paradigms when dealing with minimization problems. In other words, there is no evident transfer of positive or inapproximability results from one framework to the other one. Hence, a "good" differential approximation result does not signify anything for the behavior of the approximation algorithm studied, or of the problem itself, when dealing with the standard framework, and vice-versa. Things are somewhat different for maximization problems with positive solution-values. In fact, considering an instance x of a maximization problem Π and a solution $y \in \mathrm{Sol}(x)$ that is a δ-differential approximation, we immediately get:

$$\frac{m(x,y) - \omega(x)}{\mathrm{opt}(x) - \omega(x)} \geqslant \delta \implies \frac{m(x,y)}{\mathrm{opt}(x)} \geqslant \delta + (1-\delta)\frac{\omega(x)}{\mathrm{opt}(x)} \stackrel{\omega(x) \geqslant 0}{\implies} \frac{m(x,y)}{\mathrm{opt}(x)} \geqslant \delta$$

So, positive results are transferred from differential to standard approximation, while transfer of inapproximability thresholds is done in the opposite direction.

Fact 1. Approximation of a maximization **NPO** problem Π within differential-approximation ratio δ, implies its approximation within standard-approximation ratio δ.

Fact 1 has interesting applications. The most immediate of them deals with the case of maximization problems with worst-solution values 0. There, standard and approximation ratios coincide. In this case, the differential paradigm inherits the inapproximability thresholds of the standard one. For instance, the inapproximability of MAX INDEPENDENT SET within $n^{\epsilon-1}$, for any $\epsilon > 0$ ([41]), becomes a result shared from both of the paradigms.

Furthermore, since MAX INDEPENDENT SET and MIN VERTEX COVER are affine equivalent, henceforth differentially equi-approximable, the negative result for MAX INDEPENDENT SET is shared, in the differential paradigm, by MIN VERTEX COVER.

Corollary 1. *Both* MAX INDEPENDENT SET *and* MIN VERTEX COVER *are inapproximable within differential ratios $n^{\epsilon-1}$, for any $\epsilon > 0$, unless $\boldsymbol{P = NP}$.*

Notice that differential equi-approximability of MAX INDEPENDENT SET and MIN VERTEX COVER makes that, in this framework the latter problem is not constant approximable but inherits also the positive standard approximation results of the former one ([30, 28, 38]).

In what follows in this section, we mainly focus ourselves on three well-known **NPO** problems: MIN COLORING, BIN PACKING, TSP in both minimization and maximization variants, ant MIN MULTIPROCESSOR SCHEDULING. As we will see, approximabilities of MIN COLORING and MIN TSP are radically different from the standard paradigm (where these problems are very hard) to the differential one (where they become fairly well-approximable). For the two first of them, differential approximability will be introduced by means of more general problem that encompasses both MIN COLORING and BIN PACKING, namely, the MIN HEREDITARY COVERING. MIN HEREDITARY COVERING@MIN HEREDITARY COVERING

3.1 MIN HEREDITARY COVER

Let π be a non-trivial *hereditary* property[5] on sets and C a ground set. A π-covering of C is a collection $\mathcal{S} = \{S_1, S_2, S_q\}$ of subsets of C (i.e. a subset of 2^C), any of them verifying π and such that $\cup_{i=1}^q S_i = C$. Then, MIN HEREDITARY COVER consists, given a property π, a ground set C and a family $\mathcal{S}$ including *any* subset of C verifying π, of determining a π-covering of minimum

---

[5]A property is hereditary if whenever is true for some set, it is true for any of its subsets; it is non-trivial if it is true for infinitely many sets and false for infinitely many sets also.

size. Observe that, by definition of the instances of MIN HEREDITARY COVER singletons of the ground sets are included in any of them and are always sufficient to cover C. Henceforth, for any instance x of the problem, $\omega(x) = |C|$.

It is easy to see that, given a π-covering, one can yield a π-partition (i.e., a collection $\mathcal{S}$ where for any $S_i, S_j \in \mathcal{S}$, $S_i \cap S_j = \emptyset$) of the same size, by greedily removing duplications of elements of C. Henceforth, MIN HEREDITARY COVER or MIN HEREDITARY PARTITION are, in fact, the same problem. MIN HEREDITARY COVER has been introduced in [44] and revisited in [39] under the name MIN COVER BY INDEPENDENT SETS. Moreover, in the former paper, using a clever adaptation of the local improvement methods of [37], a differential ratio 3/4 for MIN HEREDITARY COVER has been proposed. Based upon [32], this ratio has been carried to 289/360 by [39].

A lot of well-known **NPO** problems are instantiations of MIN HEREDITARY COVER. For instance, MIN COLORING becomes an MIN HEREDITARY COVER-problem, considering as ground set the vertices of the input-graph and as set-system, the set of the independent sets[6] of this graph. The same holds for the partition of the covering of a graph by subgraphs that are planar, or by degree-bounded subgraphs, etc. Furthermore, if any element of C is associated with a weight and a subset S_i of C is in $\mathcal{S}$ if the total weight of its members is at most 1, then one recovers BIN PACKING.

In fact, an instance of MIN HEREDITARY COVER can be seen as a virtual instance of MIN SET COVER, even if there is no always need to explicit it. Furthermore, the following general result links MIN k-SET COVER (the restriction of MIN SET COVER to subsets of cardinality at most k) and MIN HEREDITARY COVER (see [52] for its proof in the case of MIN COLORING; it can be immediately seen that its extension to the general MIN HEREDITARY COVER is immediate).

Theorem 1. *If* MIN k-SET COVER *is approximable in polynomial time within differential approximation ratio δ, then* MIN HEREDITARY COVER *is approximable in polynomial time within differential-approximation ratio* $\min\{\delta, k/(k+1)\}$.

3.1.1 MIN COLORING

MIN COLORING has been systematically studied in the differential paradigm. Sub-sequent papers ([23, 24, 40, 36, 37, 57, 32]) have improved its differential approximation ratio from 1/2 to 289/360. This problem is also a typical example of a problem that behaves in completely different ways when dealing with the standard or the differential paradigms. Indeed, dealing with the former one, MIN COLORING is inapproximable within ratio $n^{1-\epsilon}$, for any $\epsilon > 0$, unless problems in **NP** can be solved by slightly super-polynomial deterministic algorithms (see [4]).

As we have seen just previously, given a graph $G(V, E)$, MIN COLORING can be seen as a MIN HEREDITARY COVER-problem considering $C = V$ and taking for $\mathcal{S}$ the set of the independent sets of G. According to Theorem 1 and to [32], where MIN 6-SET COVER is proved approximable within differential ratio 289/360, one can derive that also is approximable within differential ratio 289/360. Notice that any result for MIN COLORING holds also for the minimum vertex-partition (or covering) into cliques problem since an independent set in some graph G becomes a clique in the complement $\bar{G}$ of G (in other words this problem is also an instantiation of MIN HEREDITARY COVER). Furthermore, in [23, 24], a differential ratio preserving reduction is devised between minimum vertex-partition into cliques and minimum edge-partition (or covering) into cliques. So, as in the standard paradigm, all these three problems have identical differential approximation behavior.

Finally, it is proved in [9] that MIN COLORING is **DAPX**-complete (see also Section 4.3.1); consequently, unless **P** = **NP**, it cannot be solved by polynomial time differential-approximation

---

[6]It is well-known that the independence property is hereditary.

schemata. This derives immediately that neither MIN HEREDITARY COVER belongs to **DPTAS**, unless **P = NP**.

3.1.2 BIN PACKING

We now deal with another very well-known **NPO** problem, the BIN PACKING. According to what has been discussed just above, BIN PACKING being a particular case of MIN HEREDITARY COVER, it is approximable within differential ratio $289/360$. In what follows in this section, we refine this result by first presenting an approximation preserving reduction transforming any standard approximation ratio ρ into differential approximation ratio $\delta = 2 - \rho$. Then, based upon this reduction we show that BIN PACKING can be solved by a polynomial time differential approximation schema; in other words, BIN PACKING $\in$ **DPTAS**. This result draws another, although less dramatical than the one in Section 3.1.1, difference between standard and differential approximation. In the former paradigm, BIN PACKING is solved by an *asymptotical* polynomial time approximation schema, more precisely within standard-approximations ratio $1 + \epsilon + (1/\mathrm{opt}(L))$, for any $\epsilon > 0$ ([34]), but it is **NP**-hard to approximate it by a "real" polynomial time approximation schema ([35]).

Consider a list $L = \{x_1, \ldots, x_n\}$, instance of BIN PACKING, assume, without loss of generality, that items in L are rational numbers ranged in decreasing order and fix an optimal solution B^* of L. Observe that $\omega(L) = n$. For the purposes of this section, a bin i will be denoted either by b_i, or by explicit listing of the numbers placed in it; finally, any solution will be alternatively represented as union of its bins.

Lemma 1. *([25]) Let k^* be the number of bins in B^* that contain a single item. Then, there exists an optimal solution $\bar{B}^* = \{x_1\} \cup \ldots \cup \{x_{k^*}\} \cup \bar{B}_2^*$ for L, where any bin in $\bar{B}_2^*$ contains at least two items. Furthermore, for any optimal solution $\hat{B} = \{b_j : j = 1, \ldots, \mathrm{opt}(L)\}$ and for any set $J \subset \{1, \ldots, \mathrm{opt}(L)\}$, the solution $B_j = \{b_j \in B : j \in J\}$ is optimal for the sub-list $L_j = \cup_{j \in J} b_j$.*

Theorem 2. *From any algorithm achieving standard approximation ratio ρ for BIN PACKING, can be derived an algorithm achieving differential approximation ratio $\delta = 2 - \rho$.*

Sketch of proof. Consider Algorithm `SA` achieving standard approximation ratio ρ for BIN PACKING, denote by `SA`(L) the solution computed by it, when running on an instance L (recall that L is assumed ranged in decreasing order), and run the following algorithm, denoted by `DA` in the sequel, which uses `SA` as sub-procedure:

1. for $k = 1$ to n set: $L_k = \{x_{k+1}, \ldots, x_n\}$, $B_k = \{x_1\} \cup \ldots \cup \{x_k\} \cup$ `SA`(L_k);

2. output $B = \mathrm{argmin}\{|B_k| : k = 0, \ldots, n-1\}$.

Let $\bar{B}^*$ be the optimal solution claimed by Lemma 1. Then, $\bar{B}_2^*$ is an optimal solution for the sub-list L_{k^*}. Observe that Algorithm `SA` called by `DA` has also been executed on L_{k^*} and denote by B_{k^*} the solution so computed by `DA`. The solution returned in Step 2 verifies $|B| \leqslant |B_{k^*}|$. Finally, since any bin in $\bar{B}_2^*$ contains at least two items, $|L_{k^*}| = n - k^* \geqslant 2\mathrm{opt}(L_{k^*})$. Putting all this together, we get:

$$\delta_{\mathtt{DA}}(L, B) = \frac{n - |B|}{n - \mathrm{opt}(L)} \geqslant \frac{|L_{k^*}| - |B_{k^*}|}{|L_{k^*}| - \mathrm{opt}(L_{k^*})} \geqslant 2 - \rho$$

q.e.d. ∎

In what follows, denote by `EXHAUSTIVE` an exhaustive-search algorithm for BIN PACKING, by `SA` any polynomial algorithm approximately solving BIN PACKING within (fixed) constant

standard-approximation ratio $\rho_{\mathtt{SA}} \geqslant 1$, by $\mathtt{ASCHEMA}(\epsilon)$ the asymptotic polynomial time standard-approximation schema of [34], parameterized by $\epsilon > 0$, and consider the following algorithm, denoted by $\mathtt{EBP}(\mathtt{L}, \eta)$, parameterized by $\eta \in \{0, \dots, n\}$ (L is always assumed ranged in decreasing order):

1. for $k = n - \eta + 1, \dots, n$ build list L_{k-1} where L_{k-1} is as in Step 1 of Algorithm $\mathtt{DA}$;

2. for any list L_i computed in Step 1 above, set $B_i = \{\{x\} : x \in L \setminus L_i\} \cup \mathtt{EXHAUSTIVE}(L_i)$;

3. return B the smallest of the solutions computed in Step 2.

Algorithm $\mathtt{EBP}(\mathtt{L}, \eta)$ obviously computes a feasible BIN PACKING-solution for L in polynomial time whenever η is a fixed constant.

Lemma 2. ([25]) Assume a list L such that $|L_{k^*+1}| \leqslant \eta$. Then, algorithm $\mathtt{EBP}(\mathtt{L}, \eta)$ exactly solves BIN PACKING in L in polynomial time when μ is a fixed constant.

Lemma 3. ([25]) Fix a polynomial time approximation algorithm $\mathtt{SA}$ for BIN PACKING that guarantees standard-approximation ratio $\rho_{\mathtt{SA}}$, let ϵ be any fixed positive constant and L be an instance of BIN PACKING (ranged in decreasing order). Assume that L is such that $|L_{k^*+1}| \geqslant 2(\rho_{\mathtt{SA}} - 1 + \epsilon)/\epsilon^2$. Then, if $\mathrm{opt}(L_{k^*+1}) \leqslant \epsilon|L_{k^*+1}|/(\rho_{\mathtt{SA}} - 1 + \epsilon)$, the approximation ratio of algorithm $\mathtt{DA}$, when calling $\mathtt{SA}$ as sub-procedure, is $\delta \geqslant 1 - \epsilon$; on the other hand, if $\mathrm{opt}(L_{k^*+1}) \geqslant \epsilon|L_{k^*+1}|/(\rho_{\mathtt{SA}} - 1 + \epsilon)$, then the approximation ratio of algorithm $\mathtt{DA}$, when calling $\mathtt{ASCHEMA}(\epsilon/2)$ as sub-procedure, is $\delta \geqslant 1 - \epsilon$.

Fix any polynomial algorithm $\mathtt{A}$ for BIN PACKING with standard-approximation ratio ρ and consider the following algorithm for BIN PACKING, denoted by $\mathtt{DSCHEMA}$:

1. fix a constant $\epsilon > 0$ and set $\eta = \lfloor 2(\rho - 1 + \epsilon)/\epsilon^2 \rfloor$;

2. run $\mathtt{EBP}$ and $\mathtt{DA}$ both with $\mathtt{A}$ and $\mathtt{ASCHEMA}(\epsilon/2)$, respectively as sub-procedures on L;

3. output the best among the three solutions computed in Step 2.

Theorem 3. ([25]) Algorithm $\mathtt{DSCHEMA}$ is a polynomial time differential-approximation schema for BIN PACKING. So, BIN PACKING $\in$ **DPTAS**.

Sketch of proof. Since ρ and ϵ do not depend on n, neither does η, computed at Step 1. Consequently, by Lemma 2, execution of $\mathtt{EBP}$ at Step 2 can be performed in polynomial time and, if $|L_{k^*+1}| \leqslant \eta$, it provides some optimal solution for L. On the other hand, if $|L_{k^*+1}| \geqslant 2(\rho - 1 + \epsilon)/\epsilon^2$, Algorithm $\mathtt{PTDAS}$ achieves, by Lemma 3, differential-approximation ratio $1 - \epsilon$ for any possible value of $\mathrm{opt}(L_{k^*+1})$. ∎

Let us note that, as we will see in Section 4.4, BIN PACKING is **DPTAS**-complete; consequently, unless **P** = **NP** it is inapproximable by fully polynomial time differential-approximation schemata. Inapproximability of BIN PACKING by such schemata has independently been shown also in [30].

3.2 Travelling salesman problems

MIN TSP is one of the most paradigmatic problems in combinatorial optimization and one of the hardest one to approximate. Indeed, unless **P** = **NP**, no polynomial algorithm can guarantee, on an edge-weighted complete graph of size n when no restriction is imposed to the edge-weights, standard-approximation ratio $O(2^{p(n)})$, for any polynomial p. As we will see in this section things

is completely different when dealing with differential approximation where MIN TSP $\in$ **APX**. This result draws another notorious difference between the two paradigms.

Consider an edge-weighted complete graph of order n, denoted by K_n and observe that the worst MIN TSP-solution in K_n is an optimal solution for MAX TSP. Consider the following algorithm (originally proposed by [45] for MAX TSP) based upon a careful patching of the cycles of a minimum-weight 2-matching[7] of K_n:

- compute a $M = (C_1, C_2, \ldots, C_k)$; denote by $\{v_i^j : j = 1, \ldots, k, i = 1, \ldots, |C_j|\}$, the vertex-set of C_j; if $k = 1$, return M;

- for any C_j, pick arbitrarily four consecutive vertices v_i^j, $i = 1, \ldots, 4$; if $|C_j| = 3$, $v_4^j = v_1^j$; for C_k (the last cycle of M), pick also another vertex, denoted by u that is the other neighbor of v_1^k in C_k (hence, if $|C_k| = 3$, then $u = v_3^k$ while if $|C_k| = 4$, then $u = v_4^k$);

- if k is even (resp., odd), then set:

  - $R_1 = \cup_{j=1}^{k-1}\{(v_2^j, v_3^j)\} \cup \{(v_1^k, v_2^k)\}$
    $A_1 = \{(v_1^k, v_3^1), (v_2^1, v_2^2)\} \cup_{j=1}^{(k-2)/2} \{(v_3^{2j}, v_3^{2j+1}), (v_2^{2j+1}, v_2^{2j+2})\}$
    (resp., $R_1 = \cup_{j=1}^{k}\{(v_2^j, v_3^j)\}$, $A_1 = \{(v_2^k, v_3^1)\} \cup_{j=1}^{(k-1)/2} \{(v_2^{2j-1}, v_2^{2j}), (v_3^{2j}, v_3^{2j+1})\}$)
    $T_1 = (M \setminus R_1) \cup A_1$;
  - $R_2 = \cup_{j=1}^{k-1}\{(v_1^j, v_2^j)\} \cup \{(u, v_1^k)\}$
    $A_2 = \{(u, v_2^1), (v_1^1, v_1^2)\} \cup_{j=1}^{(k-2)/2} \{(v_2^{2j}, v_2^{2j+1}), (v_1^{2j+1}, v_1^{2j+2})\}$
    (resp., $R_2 = \cup_{j=1}^{k}\{(v_1^j, v_2^j)\}$, $A_2 = \{(v_1^k, v_2^1)\} \cup_{j=1}^{(k-1)/2} \{(v_1^{2j-1}, v_1^{2j}), (v_2^{2j}, v_2^{2j+1})\}$)
    $T_2 = (M \setminus R_2) \cup A_2$;
  - $R_3 = \cup_{j=1}^{k-1}\{(v_3^j, v_4^j)\} \cup \{(v_2^k, v_3^k)\}$
    $A_3 = \{(v_2^k, v_4^1), (v_3^1, v_3^2)\} \cup_{j=1}^{(k-2)/2} \{(v_4^{2j}, v_4^{2j+1}), (v_3^{2j+1}, v_3^{2j+2})\}$
    (resp., $R_3 = \cup_{j=1}^{k}\{(v_3^j, v_4^j)\}$, $A_3 = \{(v_3^k, v_4^1)\} \cup_{j=1}^{(k-1)/2} \{(v_3^{2j-1}, v_3^{2j}), (v_4^{2j}, v_4^{2j+1})\}$)
    $T_3 = (M \setminus R_3) \cup A_3$;

- output T the best among T_1, T_2 and T_3.

As it is proved in [45], the set $(M \setminus \cup_{i+1}^3 R_i) \cup_{i+1}^3 A_i$ is a feasible solution for MIN TSP, the value of which is a lower bound for $\omega(K_n)$; furthermore:

$$m(K_n, T) \leqslant \frac{\sum_{i=1}^{3} m(K_n, T_i)}{3}$$

Then, a smart analysis, leads to the following theorem (the same result has been obtained, by a different algorithm working also for negative edge-weights, in [39]).

Theorem 4. *([45])* MIN TSP *is differentially 2/3-approximable.*

Notice that MIN TSP, MAX TSP, MIN METRIC TSP and MAX METRIC TSP are all affine equivalent (see [47] for the proof; for the two former of them, just replace weight $d(i,j)$ of edge (v_i, v_j) by $M - d(i,j)$, where M is some number greater than the maximum edge weight). Hence, the following theorem holds.

---

[7]A minimum total weight partial subgraph of K_n any vertex of which has degree at most 2; this computation is polynomial, see, for example [16]; in other words a 2-matching is a collection of paths and cycles, but when dealing with complete graphs a 2-matching can be considered as a collection of cycles.

Theorem 5. MIN TSP, MAX TSP, MIN METRIC TSP *and* MAX METRIC TSP *are differentially 2/3-approximable.*

A very famous restrictive version of MIN METRIC TSP is the MIN TSP12, where edge-weights are all either 1, or 2. In [46], it is proved that this version (as well as, obviously, MAX TSP12) is approximable within differential ratio 3/4.

3.3 MIN MULTIPROCESSOR SCHEDULING

We now deal with a classical scheduling problem, the MIN MULTIPROCESSOR SCHEDULING ([42]), where we are given n tasks $t_1, \ldots, t_n$ with (execution) time lengths $l(t_j)$, $j = 1, \ldots, n$, polynomial with n, that have to be executed on m processors, and the objective is to partition these tasks on the processors in such a way that the occupancy of the busiest processor is minimized. Observe that the worst solution is the one where all the task are executed in the same processor; so, given an instance x of MIN MULTIPROCESSOR SCHEDULING, $\omega(x) = \sum_{j=1}^{n} l(t_j)$. A solution y of this problem will be represented as a vector in $\{0,1\}^{mn}$, the non-zero components y_j^i of which correspond to the assignment of task j to processor i.

Consider a simple local search algorithms that starts from some solution and improves it upon any change of the assignment of a single task from one processor to another one. Then the following result can be obtained ([48]).

Theorem 6. MIN MULTIPROCESSOR SCHEDULING *is approximable within differential ratio* $m/(m+1)$.

Sketch of proof. Assume that both tasks and processors are ranged with decreasing lengths and occupancies, respectively. Denote by $l(p_i)$, the total occupancy of processor p_i, $i = 1, \ldots, m$. Then:

$$
\begin{aligned}
\mathrm{opt}(x) &\geqslant l(t_1) \\
l(p_1) &= \sum_{j=1}^{n} y_j^1 l(t_j) = \max_{i=1,\ldots,m} \left\{ l(p_i) = \sum_{j=1}^{n} y_j^i l(t_j) \right\}
\end{aligned}
$$

Denote, w.l.o.g., by $1, \ldots, q$, the indices of the tasks assigned to p_1. Since y is a local optimal, it verifies, for $i = 2, \ldots, m$, $j = 1, \ldots q$: $l(t_j) + l(p_i) \geqslant l(p_1)$. We can assume $q \geqslant 2$ (on the contrary y is optimal). Then, adding the preceding expression for $j = 1, \ldots, q$, we get: $l(p_i) \geqslant l(p_1)/2$. Also, adding $l(p_1)$ with the preceding expression for $l(p_i)$, $i = 2, \ldots, m$, we obtain: $\omega(x) \geqslant (m+1)l(p_1)/2$. Putting all this together we finally get:

$$
m(x, y) = l(p_1) \leqslant \frac{m\mathrm{opt}(x)}{m+1} + \frac{\omega(x)}{m+1}
$$

q.e.d. ∎

4 Structure in differential approximation classes

What has been discussed in the previous sections makes clear that the entire theory of approximation, that tries to characterize and to classify problems with respect to their approximability hardness, can be redone in the differential paradigm. There exist problems having several differential-approximability levels and inapproximability bounds. What follows further confirms this claim. It will be shown that the approximation paradigm we deal with, allows to devise its proper tools and to use them in order to design an entire structure for the approximability classes involved.

4.1 Differential NPO-completeness

Obviously, the *strict* reduction of [50] (see also [3]), can be identically defined in the framework of the differential approximation; for clarity, we denote this derivation of the strict reduction by D-reduction. Two **NPO** problems will be called D-equivalent if there exist D-reductions from any of them to the other one.

Theorem 3.1 in [50] (where the differential approximation ratio is mentioned as possible way of estimating the performance of an algorithm), based upon an extension of Cook's proof ([15]) of SAT **NP**-completeness to optimization problems, works also when the differential ratio is dealt instead the standard one. Furthermore, solution *triv*, as defined in [50] is indeed a worst solution for MIN WSAT. On the other hand, the following proposition holds.

Proposition 2. *([3])* MAX WSAT *and* MIN WSAT *are* D-equivalent.

Sketch of proof. With any clause $\ell_1 \vee \ldots \vee \ell_t$ of an instance φ of MAX WSAT, we associate in the instance φ' of MIN WSAT the clause $\bar{\ell}_1 \vee \ldots \vee \bar{\ell}_t$. Then, if an assignment y satisfies the instance φ, the complement y' of y satisfies φ', and vice-versa. So, $m(\varphi, y) = \sum_{i=1}^{n} w(x_i) - m(\varphi', y')$, for any y'. Thus, $\delta(\varphi, y) = \delta(\varphi', y')$. The reduction from MIN WSAT to WSAT is completely analogous. ∎

In a completely analogous way, as in Proposition 2, it can be proved that MIN 0-1 INTEGER PROGRAMMING and MAX 0-1 INTEGER PROGRAMMING are also D-equivalent. Putting all the above together the following holds

Theorem 7. MAX WSAT, MIN WSAT, MIN 0-1 INTEGER PROGRAMMING *and* MAX 0-1 INTEGER PROGRAMMING *are* **NPO**-*complete under* D-*reducibility.*

4.2 The class 0-DAPX

Informally, class **0-DAPX** is the class of **NPO** problems for which the differential ratio of any polynomial algorithm is *equal* to 0. In other words, for any such algorithm, there exists an instance on which it will compute its worst solution. Such situation draws the worst case for the differential approximability of a problem. Class **0-DAPX** is defined in [3] by means of a reduction, called G-reduction. It can be seen as a particular kind of the GAP-reduction ([2, 4, 59]).

Definition 3. A problem Π is said to be G-reducible to a problem Π', if there exists a polynomial reduction that transforms any δ-differential approximation algorithm for Π', $\delta > 0$ into an optimal (exact) algorithm for Π. ∎

Let Π be an **NP**-complete decision problem and Π' an **NPO** problem. The underlying idea for $\Pi \leqslant_G \Pi'$ in definition 3 is, starting from an instance of Π, to construct instances for Π' that have only two distinct feasible values and to prove that any differential δ-approximation for Π', $\delta > 0$, could distinguish between positive instances and negative instances for Π. Note finally that the G-reduction generalizes both the D-reduction of Section 4.1 and the strict reduction of [50].

Definition 4. **0-DAPX** is the class of **NPO** problems Π' for which there exists an **NP**-complete problem Π G-reducible to Π'. A problem is said to **0-DAPX**-hard, if any problem in **0-DAPX** G-reduces to it. ∎

An obvious consequence of Definition 4 is that *0-DAPX is the class of* **NPO** *problems* Π *for which approximation within any differential approximation ratio* $\delta > 0$ *would entail* **P = NP**.

Proposition 3. *([11])* MIN INDEPENDENT DOMINATING SET $\in$ *0-DAPX*.

Sketch of proof. Given an instance φ of SAT with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$, construct a graph G, instance of MIN INDEPENDENT DOMINATING SET associating with any positive literal x_i a vertex u_i and with any negative literal $\bar{x}_i$ a vertex v_i. For $i = 1, \ldots, n$, draw edges (u_i, v_i). For any clause C_j, add in G a vertex w_j and an edge between w_j and any vertex corresponding to a literal contained in C_j. Finally, add edges in G in order to obtain a complete graph on $w_1, \ldots, w_m$. An independent set of G contains at most $n + 1$ vertices. An independent dominating set containing the vertices corresponding to true literals of a non satisfiable assignment and one vertex corresponding to a clause not satisfied by this assignment, is a worst solution of G of size $n + 1$. If φ is satisfiable then $\mathrm{opt}(G) = n$. If φ is not satisfiable then $\mathrm{opt}(G) = n + 1$. So, any independent dominating set of G has cardinality either n, or $n + 1$. ■

By analogous reductions, restricted versions of optimum weighted satisfiability problems are proved **0-DAPX** in [53].

Finally, the following relationship between **NPO** and **0-DAPX** holds.

Theorem 8. *([3]) Under D-reducibility, **NPO**-complete = **0-DAPX**-complete $\subseteq$ **0-DAPX**.*

If, instead D, a stronger reducibility is considered, for instance, by allowing f and/or g to be multivalued in the strict reduction, then, under this type of reducibility, it can be proved that **NPO**-complete = **0-DAPX** ([3]).

4.3 DAPX- and Poly-DAPX-completeness

In this section we address the problem of completeness in the classes **DAPX** and **Poly-DAPX**. For this purpose, we first introduce a differential approximation schemata preserving reducibility, originally presented in [3], called DPTAS-reducibility.

Definition 5. Given two **NPO** problems Π and Π', Π DPTAS-reduces to Π' if there exist a (possibly) multi-valued function $f = (f_1, f_2, \ldots, f_h)$, where h is bounded by a polynomial in the input-length, and two functions g and c, computable in polynomial time, such that:

- for any $x \in \mathcal{I}_\Pi$, for any $\epsilon \in (0, 1) \cap \mathbb{Q}$, $f(x, \epsilon) \subseteq \mathcal{I}_{\Pi'}$;

- for any $x \in \mathcal{I}_\Pi$, for any $\epsilon \in (0, 1) \cap \mathbb{Q}$, for any $x' \in f(x, \epsilon)$, for any $y \in \mathrm{sol}_{\Pi'}(x')$, $g(x, y, \epsilon) \in \mathrm{sol}_\Pi(x)$;

- $c : (0, 1) \cap \mathbb{Q} \to (0, 1) \cap \mathbb{Q}$;

- for any $x \in \mathcal{I}_\Pi$, for any $\epsilon \in (0, 1) \cap \mathbb{Q}$, for any $y \in \cup_{i=1}^h \mathrm{sol}_{\Pi'}(f_i(x, \epsilon))$, $\exists j \leqslant h$ such that $\delta_{\Pi'}(f_j(x, \epsilon), y) \geqslant 1 - c(\epsilon)$ implies $\delta_\Pi(x, g(x, y, \epsilon)) \geqslant 1 - \epsilon$. ■

It can be easily shown that given two **NPO** problems Π and Π', if $\Pi \leqslant_{\mathsf{DPTAS}} \Pi'$ and $\Pi' \in \mathbf{DAPX}$, then $\Pi \in \mathbf{DAPX}$.

4.3.1 DAPX-completeness

If one restricts her/himself to problems with polynomially computable worst solutions, then things are rather simple. Indeed, given such a problem $\Pi \in \mathbf{DAPX}$, it is affine equivalent to a problem Π' defined on the same set of instances and with the same set of solutions but, for any solution y of an instance x of Π, the measure for solution y with respect to Π' is defined as $m_{\Pi'}(x, y) = m_\Pi(x, y) - \omega(x)$. Affine equivalence of Π and Π' ensures that $\Pi' \in \mathbf{DAPX}$; furthermore, $\omega_{\Pi'}(x) = 0$. Since, for the latter problem, standard and differential approximation ratios coincide, it follows that $\Pi' \in \mathbf{APX}$. MAX INDEPENDENT SET is **APX**-complete under PTAS-reducibility ([19]), a particular kind of the AP-reducibility [4, 18, 20]. So, Π' PTAS-reduces to MAX

INDEPENDENT SET. Putting together affine equivalence between Π and Π', PTAS-reducibility between Π' and MAX INDEPENDENT SET, and taking into account that composition of these two reductions is an instantiation of DPTAS-reduction, we conclude the **DAPX**-completeness of MAX INDEPENDENT SET.

However, things become much more complicated, if one takes into account problems with non-polynomially computable worst solutions. In this case, one needs more sophisticated techniques and arguments. We informally describe here the basic ideas and the proof-schema in [3]. It is first shown that any **DAPX** problem Π is reducible to MAX WSAT-B by a reduction transforming a polynomial time approximations schema for MAX WSAT-B into a polynomial time differential-approximation schema for Π. For simplicity, denote this reduction by S-D. Next, a particular **APX**-complete problem Π' is considered, say MAX INDEPENDENT SET-B. MAX WSAT-B, that is in **APX**, is PTAS-reducible to MAX INDEPENDENT SET-B. MAX INDEPENDENT SET-B is both in **APX** and in **DAPX** and, moreover, standard and differential approximation ratios coincide for it; this coincidence draws a trivial reduction called ID-reduction. It trivially transforms a differential polynomial time approximation schema into a standard polynomial time approximation schema. The composition of the three reductions specified (i.e., the S-D-reduction from Π to MAX WSAT-B, the PTAS-reduction from MAX WSAT-B to MAX INDEPENDENT SET-B and the ID-reduction) is a DPTAS-reduction transforming a polynomial time differential-approximation schema for MAX INDEPENDENT SET-B into a polynomial time differential-approximation schema for Π, i.e., MAX INDEPENDENT SET-B is **DAPX**-complete under DPTAS-reducibility.

Also, by standard reductions that turn out to be DPTAS-reductions also, the following can be proved.

Theorem 9. *([3, 9])* MAX INDEPENDENT SET-B, MIN VERTEX COVER-B, *for fixed B,* MAX k-SET PACKING, MIN k-SET COVER, *for fixed k, and* MIN COLORING *are* **DAPX**-*complete under* DPTAS-*reducibility.*

4.3.2 Poly-DAPX-completeness

Let us notice first that, for reasons very cleverly explained in [43] for the standard paradigm, use of restrictive reductions as the E-reducibility introduced there, where the functions f and g do not depend on any parameter ϵ seems very unlikely to be able to handle **Poly-APX**-completeness. For this reason, in [43], only **Poly-APX**-completeness for polynomially bounded problems is handled. As it is shown in [9], in order to handle the whole **Poly-APX**-completeness, less restrictive reducibilities are needed. The same observation can be done also for **Poly-DAPX**-completeness and any translation of E-reducibility to the differential paradigm.

Fortunately, the scope of the DPTAS-reducibility is large enough to allow not only apprehension of **DAPX**-completeness but also of **Poly-DAPX**-completeness. Recall that a maximization problem $\Pi \in$ **NPO** is *canonically hard for* **Poly-APX** ([43]), if and only if there exist a polynomially computable transformation T from 3SAT to Π, two constants n_0 and c and a function F, hard for **Poly**[8], such that, given an instance x of 3SAT on $n \geqslant n_0$ variables and a number $N \geqslant n^c$, the instance $x' = T(x, N)$ belongs to $\mathcal{I}_\Pi$ and verifies the three following properties: (i) if x is satisfiable, then $\mathrm{opt}(x') = N$; (ii) if x is not satisfiable, then $\mathrm{opt}(x') = N/F(N)$; (iii) given a solution $y \in \mathrm{sol}_\Pi(x')$ such that $m(x', y) > N/F(N)$, one can polynomially determine a truth assignment satisfying x.

Based upon DPTAS-reducibility and the notion of canonical hardness, the following is proved in [9].

---

[8]The set of functions from $\mathbb{N}$ to $\mathbb{N}$ bounded by a polynomial ; a function $f \in$ **Poly** is hard for **Poly**, if and only if there exist three constants k, c and n_0 such that, for any $n \geqslant n_0$, $f(n) \leqslant kF(n^c)$.

Theorem 10. *If a (maximization) problem $\Pi \in$ **NPO** is canonically hard for **Poly-APX**, then any problem in **Poly-DAPX** DPTAS-reduces to Π.*

As it is shown in [43], MAX INDEPENDENT SET is canonically hard for **Poly-APX**. Furthermore, MIN VERTEX COVER is affine equivalent to MAX INDEPENDENT SET. Henceforth, use of Theorem 10, immediately derives the following result.

Theorem 11. MAX INDEPENDENT SET *and* MIN VERTEX COVER *are complete for **Poly-DAPX** under DPTAS-reducibility.*

Finally let us note that, as it is proved in [9], MAX INDEPENDENT SET is **Poly-APX**-complete under PTAS-reducibility.

4.4 DPTAS-completeness

Completeness in **DPTAS** is tackled by means of a kind of reducibility preserving membership in **DFPTAS**, which is called DFT-reducibility in [9].

Definition 6. ([9]) Let Π and Π' be two **NPO** problems. Let $\bigcirc_\alpha^{\Pi'}$ be an oracle for Π' producing, for any $\alpha \in (0, 1]$ and for any instance x' of Π', a feasible solution $\bigcirc_\alpha^{\Pi'}(x')$ of x' that is an $(1-\alpha)$-differential approximation. Then, Π DFT-reduces to Π' if and only if, for any $\epsilon > 0$, there exists an algorithm $\mathtt{A}_\epsilon(x, \bigcirc_\alpha^{\Pi'})$ such that:

- for any instance x of Π, $\mathtt{A}_\epsilon$ returns a feasible solution which is a $(1-\epsilon)$-differential approximation;

- if $\bigcirc_\alpha^{\Pi'}(x')$ runs in time polynomial with both $|x'|$ and $1/\alpha$, then $\mathtt{A}_\epsilon$ is polynomial with both $|x|$ and $1/\epsilon$. ∎

DPTAS-reduction under DFT-reducibility can be easily derived by two intermediate lemmata. The first one introduces some property of the seminal Turing-reducibility (see [4] for formal definition).

Lemma 4. *([9]) If an **NPO** problem Π' is **NP**-hard, then any **NPO** problem Turing-reduces to Π'.*

Before stating the second lemma, we need to introduce the class of diameter polynomially bounded problems that is a subclass of the radial problems seen in Section 2.2. An **NPO** problem Π is *diameter polynomially bounded* if and only if, for any $x \in \mathcal{I}_\Pi$, $|\mathrm{opt}(x) - \omega(x)| \leqslant q(|x|)$. The class of diameter polynomially bounded **NPO** problems will be denoted by **NPO-DPB**.

The second lemma claims that, under certain conditions, a Turing-reduction (that only preserves optimality) can be transformed into an DFT-reduction. Informally, starting from a Turing-reduction between two **NPO** problems Π and Π', where Π' is polynomially bounded, one can devise a DFT-reduction transforming a fully polynomial time differential-approximation schema for Π' into a fully polynomial time differential-approximation schema for Π.

Lemma 5. *([9]) Let $\Pi' \in$ **NPO-DPB**. Then, any **NPO** problem Turing-reducible to Π' is also DFT-reducible to Π'.*

Combination of Lemmata 4 and 5 immediately derives the following theorem.

Theorem 12. *([9]) Let Π' be an **NP**-hard problem **NPO-DPB**. Then, any problem in **NPO** is DFT-reducible to Π'.*

Theorem 12 immediately implies that, on the one hand, *the closure of **DPTAS** under DFT-reducibility is the whole **NPO*** and, on the other hand that *any **NP**-hard problem in **NPO-DPB**∩ **DPTAS** is **DPTAS**-complete under DFT-reducibility.*

Consider now MIN PLANAR VERTEX COVER, MAX PLANAR INDEPENDENT SET and BIN PACKING. They are all **NP**-hard and in **NPO-DPB**. Furthermore, they are all in **DPTAS** (for the first two problems, this is derived by the inclusion of MAX PLANAR INDEPENDENT SET in **PTAS** proved in [8]; for the third one, revisit Section 3.1.2). So, the following theorem holds and concludes this section.

Theorem 13. *([9])* MAX PLANAR INDEPENDENT SET, MIN PLANAR VERTEX COVER *and* BIN PACKING *are **DPTAS**-complete under DFT-reducibility.*

Finally, dealing with the scope of DFT-reducibility, it is proved in [9] that *if there exist **NPO**-intermediate problems under Turing-reducibility, then there exit problems that are **DPTAS**-intermediate, under DFT-reducibility.*

5   Discussion and final remarks

As we have already claimed in the beginning of Section 4, the entire theory of approximation can be reformulated in the differential paradigm. This paradigm has the diversity of the standard one, it has a non-empty scientific content and, to our opinion, it represents in some sense a kind of revival for the domain of the polynomial approximation.

Since the work in [29], a great number of paradigmatic combinatorial optimization problems has been studied in the framework of the differential approximation. For instance, KNAPSACK has been sudied in [29] and revisited in [39]. MAX CUT, MIN CLUSTER, STACKER CRANE, MIN DOMINATING SET, MIN DISJOINT CYCLE COVER, MAX ACYCLIC SUBGRAPH, MIN FEEDBACK ARC SET have been dealt in [39]. MIN VERTEX COVER and MAX INDEPENDENT SET are studied in [29] and in [39]. MIN COLORING is dealt in [23, 24, 40, 36, 37, 32, 9], while MIN WEIGHTED COLORING (where the input is a vertex-weighted graph and the weight of a color is the weight of the heaviest of its vertices) is studied in [22] (see also [21]). MIN INDEPENDENT DOMINATING SET is dealt in [11]. BIN PACKING is studied in [24, 31, 25, 26]. MIN SET COVER, under several assumptions on its worst value, is dealt in [29, 39, 10], while MIN WEIGHTED SET COVER is dealt in [24, 10]. MIN TSP and MAX TSP, as well as, several famous variants of them, MIN METRIC TSP, MAX METRIC TSP, MIN TSPab (the most famous restrictive case of this problem is MIN TSP12), MAX TSPab are studied in [45, 47, 46, 39, 56]. STEINER TREE problems under several assumptions on the form of the input-graph and on the edge-weights are dealt in [27]. Finally, several optimum satisfiability and constraint satisfaction problems (as MAX SAT, MAX E2SAT, MAX 3SAT, MAX E3SAT, MAX EkSAT, MIN SAT, MIN kSAT, MIN EkSAT, MIN 2SAT and their corresponding constraint satisfaction versions) are studied in [33].

Dealing with structural aspects of approximation, besides the existing approximability classes (defined rather upon combinatorial arguments) two logical classes have been very notorious in the standard paradigm. These are **Max-NP** and **Max-SNP**, originally introduced in [51] (see also [4, 53]). Their definitions, independent from any approximation ratio consideration, make that they can identically be considered also in differential approximation. In the standard paradigm, the following strict inclusions hold: **PTAS** ⊂ **Max-SNP** ⊂ **APX** and **MAX-NP** ⊂ **APX**. As it is proved in [33], MAX SAT ∉ **DAPX**, unless **P** = **NP**. This, draws an important structural difference in the landscape of approximation classes in the two paradigms, since an immediate corollary of this result is that **MAX-NP** ⊄ **DAPX**. Position of **Max-SNP** in the differential landscape is not known yet. It is conjectured, however, that **MAX-SNP** ⊄ **DAPX**. In any case, formal relationships of **Max-SNP** and **Max-NP** with the other differential approximability classes deserve further study.

References

[1] A. Aiello, E. Burattini, M. Furnari, A Massarotti, and F. Ventriglia. Computational complexity: the problem of approximation. In Colloquia Mathematica Societatis János Bolyai, editor, *Algebra, combinatorics, and logic in computer science*, volume I, pages 51–62, New York, 1986. North-Holland.

[2] S. Arora and C. Lund. Hardness of approximation. In D. S. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, chapter 10, pages 399–446. PWS, Boston, 1997.

[3] G. Ausiello, C. Bazgan, M. Demange, and V. Th. Paschos. Completeness in differential approximation classes. *International Journal of Foundations of Computer Science*. To appear.

[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer, Berlin, 1999.

[5] G. Ausiello, A. D'Atri, and M. Protasi. On the structure of combinatorial problems and structure preserving reductions. In *Proc. ICALP'77*, volume 52 of *Lecture Notes in Computer Science*, pages 45–60. Springer-Verlag, 1977.

[6] G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *J. Comput. System Sci.*, 21:136–153, 1980.

[7] G. Ausiello, A. D'Atri, and M. Protasi. Lattice-theoretical ordering properties for NP-complete optimization problems. *Fundamenta Informaticæ*, 4:83–94, 1981.

[8] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180, 1994.

[9] C. Bazgan, B. Escoffier, and V. Th. Paschos. Completeness in standard and differential approximation classes: Poly-(D)APX- and (D)PTAS-completeness. *Theoret. Comput. Sci.*, 339:272–292, 2005.

[10] C. Bazgan, J. Monnot, V. Th. Paschos, and F. Serrière. On the differential approximation of MIN SET COVER. *Theoret. Comput. Sci.*, 332:497–513, 2005.

[11] C. Bazgan and V. Th Paschos. Differential approximation for optimal satisfiability and related problems. *European J. Oper. Res.*, 147(2):397–404, 2003.

[12] M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. *Math. Programming*, 69:429–441, 1995.

[13] C. Berge. *Graphs and hypergraphs*. North Holland, Amsterdam, 1973.

[14] B. Bollobás. *Random graphs*. Academic Press, London, 1985.

[15] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. STOC'71*, pages 151–158, 1971.

[16] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. J. Wiley, 1998.

[17] 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(8):789–810, 1977.

[18] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. *SIAM J. Comput.*, 28(5):1759–1782, 1999.

[19] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. In *Foundations of Software Technology and Theoretical Computer Science, FST-TCS*, number 880 in Lecture Notes in Computer Science, pages 330–341. Springer-Verlag, 1994.

[20] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. *Theory of Computing Systems*, 33(1):1–16, 2000.

[21] M. Demange, D. de Werra, J. Monnot, and V. Th. Paschos. Time slot scheduling of compatible jobs. Cahier du LAMSADE 182, LAMSADE, Université Paris-Dauphine, 2001. Available on `http://www.lamsade.dauphine.fr/cahdoc.html#cahiers`.

[22] M. Demange, D. de Werra, J. Monnot, and V. Th. Paschos. Weighted node coloring: when stable sets are expensive. In L. Kučera, editor, *Proc. 28th International Workshop on Graph Theoretical Concepts in Computer Science, WG'02*, volume 2573 of *Lecture Notes in Computer Science*, pages 114–125. Springer-Verlag, 2002.

[23] M. Demange, P. Grisoni, and V. Th. Paschos. Approximation results for the minimum graph coloring problem. *Inform. Process. Lett.*, 50:19–23, 1994.

[24] M. Demange, P. Grisoni, and V. Th. Paschos. Differential approximation algorithms for some combinatorial optimization problems. *Theoret. Comput. Sci.*, 209:107–122, 1998.

[25] M. Demange, J. Monnot, and V. Th. Paschos. Bridging gap between standard and differential polynomial approximation: the case of bin-packing. *Appl. Math. Lett.*, 12:127–133, 1999.

[26] M. Demange, J. Monnot, and V. Th. Paschos. Maximizing the number of unused bins. *Found. Comput. Decision Sci.*, 26(2):169–186, 2001.

[27] M. Demange, J. Monnot, and V. Th. Paschos. Differential approximation results for the steiner tree problem. *Appl. Math. Lett.*, pages 733–739, 2003.

[28] M. Demange and V. Th. Paschos. Improved approximations for weighted and unweighted graph problems. *Theory of Computing Systems*. To appear, Preliminary version available at `http://l1.lamsade.dauphine.fr/~paschos/documents/c177.pdf`.

[29] M. Demange and V. Th. Paschos. On an approximation measure founded on the links between optimization and polynomial approximation theory. *Theoret. Comput. Sci.*, 158:117–141, 1996.

[30] M. Demange and V. Th. Paschos. Improved approximations for maximum independent set via approximation chains. *Appl. Math. Lett.*, 10(3):105–110, 1997.

[31] M. Demange and V. Th. Paschos. Asymptotic differential approximation ratio: definitions, motivations and application to some combinatorial problems. *RAIRO Oper. Res.*, 33:481–507, 1999.

[32] R. Duh and M. Fürer. Approximation of k-set cover by semi-local optimization. In *Proc. STOC'97*, pages 256–265, 1997.

[33] B. Escoffier and V. Th. Paschos. Differential approximation of MIN SAT, MAX SAT and related problems. *European J. Oper. Res.* To appear.

[34] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[35] M. R. Garey and D. S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* W. H. Freeman, San Francisco, 1979.

[36] M. M. Halldórsson. Approximating discrete collections via local improvements. In *Proc. Symposium on Discrete Algorithms, SODA*, pages 160–169, 1995.

[37] M. M. Halldórsson. Approximating k-set cover and complementary graph coloring. In *Proc. International Integer Programming and Combinatorial Optimization Conference*, number 1084 in Lecture Notes in Computer Science, pages 118–131. Springer Verlag, 1996.

[38] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appli.*, 4(1):1–16, 2000.

[39] R. Hassin and S. Khuller. z-approximations. *J. Algorithms*, 41:429–442, 2001.

[40] R. Hassin and S. Lahav. Maximizing the number of unused colors in the vertex coloring problem. *Inform. Process. Lett.*, 52:87–90, 1994.

[41] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[42] D. S. Hochbaum and D. B. Shmoys. Using dual approximation algorithms for scheduling problems: theoretical and practical results. *J. Assoc. Comput. Mach.*, 34:144–162, 1987.

[43] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28:164–191, 1998.

[44] J. Monnot. *Critical families of instances and polynomial approximation.* Phd thesis, LAMSADE, University Paris-Dauphine, 1998. In French.

[45] J. Monnot. Differential approximation results for the traveling salesman and related problems. *Inform. Process. Lett.*, 82(5):229–235, 2002.

[46] J. Monnot, V. Th. Paschos, and S. Toulouse. Differential approximation results for the traveling salesman problem with distances 1 and 2. *European J. Oper. Res.*, 145(3):557–568, 2002.

[47] J. Monnot, V. Th. Paschos, and S. Toulouse. Approximation algorithms for the traveling salesman problem. *Mathematical Methods of Operations Research*, 57(1):387–405, 2003.

[48] J. Monnot, V. Th. Paschos, and S. Toulouse. Optima locaux garantis pour l'approximation différentielle. *Technique et Science Informatiques*, 22(3):257–288, 2003.

[49] A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley, Chichester, 1983.

[50] P. Orponen and H. Mannila. On approximation preserving reductions: complete problems and robust measures. Technical Report C-1987-28, Dept. of Computer Science, University of Helsinki, Finland, 1987.

[51] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *J. Comput. System Sci.*, 43:425–440, 1991.

[52] V. Th. Paschos. Polynomial approximation and graph coloring. *Computing*, 70:41–86, 2003.

[53] V. Th. Paschos. *Complexité et approximation polynomiale*. Hermès, Paris, 2004.

[54] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoret. Comput. Sci.*, 15:251–277, 1981.

[55] S. Sahni and T. Gonzalez. P-complete approximation problems. *J. Assoc. Comput. Mach.*, 23:555–565, 1976.

[56] S. Toulouse. *Approximation polynomiale: optima locaux et rapport différentiel*. Thèse de doctorat, LAMSADE, Université Paris-Dauphine, 2001.

[57] X. D. Tzeng and G. H. King. Three-quarter approximation for the number of unused colors in graph coloring. *Inform. Sciences*, 114:105–126, 1999.

[58] S. A. Vavasis. Approximation algorithms for indefinite quadratic programming. *Math. Programming*, 57:279–311, 1992.

[59] V. Vazirani. *Approximation algorithms*. Springer, Berlin, 2001.

[60] E. Zemel. Measuring the quality of approximate solutions to zero-one programming problems. *Math. Oper. Res.*, 6:319–332, 1981.