Local problems and linear time

Régis Barbanchon and Etienne Grandjean

GREYC, Université de Caen, 14032 Caen Cedex, France

Abstract

In this paper, we define the LIN-LOCAL complexity-class — which captures problems linearly reducible to the problems defined by boolean local constraints — as well as its planar restriction LIN-PLAN-LOCAL. After showing that SAT and PLAN-SAT are respectively complete for these classes, we prove that some unexpected problems (involving seemingly global constraints about cardinality or connectivity of solutions) indeed belong to these classes and are complete for them. More precisely, we show that VERTEX-COVER and similar problems are LIN-LOCAL-complete. Our most striking result is that PLAN-HAMILTON is LIN-PLAN-LOCAL and moreover is LIN-PLAN-LOCAL-complete. Since all our linear-time reductions also turn out to be parsimonious, we derive DP-completeness results for UNIQUE-PLAN-HAMILTON and UNIQUE-PLAN-VERTEX-COVER.

Email address: regis.barbanchon,etienne.grandjean@info.unicaen.fr (Régis Barbanchon and Etienne Grandjean).

Preprint submitted to CCC 2002 *30 November 2001*

1 Introduction

This paper originates from the observation made by several authors [3,4,12,26] that a number of natural NP-complete problems (e.g. 3COL, 3DM, KERNEL) are linearly equivalent to the SAT problem — i.e., are reducible to SAT via linear-time reductions and conversely. Two related questions arise. First, what is linear-time computability? Secondly, what is the intrinsic complexity of SAT and of the problems linearly reducible to it?

Concerning the first question, linear time is an intuitive and informal notion of algorithm designers who have built many linear-time algorithms for various combinatorial problems (e.g., in graph theory: strong connectedness, planarity; in logic: Horn-formula satisfiability). In contrast, complexity theoreticians do not generally agree about what should be the "right" notion of linear time computability [8,15,22] (see also [23]). However, in a series of papers [10–12], (see also [14,24,25]), one of the authors defines the class DLIN as the class of problems that are computable in time $O(n)$ on deterministic RAMs that *use only integers of values $O(n)$ in all steps of computation*. He notices that this class includes the above-mentioned linear algorithms and argues that it exactly corresponds to the intuitive linear-time computability. The same author also defines the nondeterministic counterpart NLIN of the class DLIN and shows that these classes are robust from both computational and logical points of views [12,13]. Finally, he asserts that most "natural" NP-complete problems, including SAT and more generally the 21 problems exhibited by Karp in [17], belong to NLIN [10,12]. He also shows [9,10] (see also [14]) that a few of them are NLIN-complete via DLIN-reductions, in particular the problem of Reduction of Incompletely Specified Automaton, referenced as AL7 in [7]. In contrast, he argues [10] that it is unlikely that SAT be NLIN-complete, because this problem only uses a *sublinear* number of nondeterministic steps. To be more precise, for any input of size n (e.g. the size of a CNF propositional formula φ is the number of occurrences of literals in φ), a NLIN algorithm is allowed to perform $\Theta(n)$ nondeterministic guess instructions — i.e., can guess $\Theta(n)$ integers, each of value $\Theta(n)$, or equivalently $\Theta(n \log n)$ bits. However, the SAT problem is easily nondeterministically solved by guessing only $\Theta(n)$ bits (one bit per variable), or equivalently $\Theta(n/\log n)$ integers, each of value $\Theta(n)$. As a consequence, if SAT turned out to be NLIN-complete, then any linear algorithm using $O(n)$ "guess"-instructions could be simulated by a similar algorithm (so-called NSUBLIN algorithm in [10]) using only $O(n/\log n)$ such "guess"-instructions and $O(n)$ deterministic instructions.

Since it is unlikely that SAT be NLIN-complete, it is natural to ask what is the exact nondeterministic complexity of this problem. In this paper, we try to answer this question by introducing a new class of decision problems, denoted LIN-LOCAL, as well as its planar restriction, denoted LIN-PLAN-LOCAL.

The characterizations of these classes include computational and logical features. Let us recall that, in contrast to Fagin's Theorem stating that ESO — the Existential Second Order Logic — exactly characterizes the NP class [5], it is known that Monadic-ESO — the restriction of ESO to the Monadic Second Order quantifiers — can express SAT and similar hard problems (3COL, KERNEL, etc.) but *cannot* express the "easy" (DLIN) problem CONNECTIVITY [6], because Monadic-ESO can only express "local" properties as First-Order Logic does. Notice that Lautemann and Weinzinger have proved similar expressibility and non-expressibility results [18] for the more restrictive monadic logic (they called Monadic-NLIN) defined by formulas $\exists \mathcal{C}\ \forall x\ \psi$, where $\mathcal{C}$ is a list of monadic relation symbols, x is the only one first-order variable, and ψ is a quantifier-free first-order formula. We feel that any *robust* sequential (deterministic or nondeterministic) time-complexity class should be closed under linear-time — i.e., DLIN — reductions. As a consequence, such a class should trivially include DLIN, which is, in our opinion [14,24], the minimal robust complexity class for sequential time. In particular, the fact that a specific problem belongs to such a class is widely independent of the representation of the inputs because the sorting problem belongs to DLIN, as proved in [12].

The contributions of this paper are the following:

First, we define a very strict notion of local problem. An input for such a problem is a so-called *local structure*, which is essentially a bounded-degree graph with labelled vertices and labelled edges. A problem over local structures is a *local problem* if it is characterized by a *local formula* as above — i.e., of the form $\exists \mathcal{C}\ \forall x\ \psi$, where the $\mathcal{C}$ is a list of monadic relation symbols, x is the only one first-order variable, and ψ is a quantifier-free formula. The term "local" comes from the fact that, syntactically, ψ is a condition over one element x together with a constant number of elements at bounded distance from x. A problem is LIN-LOCAL if it is DLIN-reducible to some local problem. We prove that SAT is LIN-LOCAL-complete, i.e. the class of problems DLIN-reducible to SAT coincides with the class of problems DLIN-reducible to local problems. From a computational point of view, a LIN-LOCAL problem is recognized by the following very special NSUBLIN algorithm, which works in two phases:

(1) A DLIN computation which constructs a local structure S.
(2) A local and parallel nondeterministic computation of constant time: In parallel for each element a of S, guess a fixed number of bits $\mathcal{C}(a)$, and finally check "locally" — i.e., only by looking at a and its neighbors — that some fixed boolean condition $\psi(a)$ is satisfied.

The key point of the LIN-LOCAL class is its minimal way to use the nondeterminism, by restricting it to happen *at the end* of the algorithm in an amount of $O(1)$ guesses per element, so that all guesses are checked in parallel in time

$O(1)$. We show that the LIN-LOCAL class is *robust*, i.e., it is not modified for several restrictions or extensions of the definition, e.g., if we require both the degree of the underlying graph of the local structure to be bounded by 3 and only one bit $\mathcal{C}(a)$ to be guessed per element.

We also study the LIN-PLAN-LOCAL class — i.e., the restriction of LIN-LOCAL problems to planar local structures and obtain that PLAN-SAT — the satisfiability of planar formulas [16,19] — is LIN-PLAN-LOCAL-complete. The distinction between LIN-LOCAL and LIN-PLAN-LOCAL classes is motivated by the fact that algorithms running in subexponential time $2^{O(n^{1/2})})$ are known for PLAN-SAT [20,21] (and hence for any LIN-PLAN-LOCAL problem) whereas authors such as Hunt and Stearns [26] argue that SAT is unlikely to be solved in time $O(2^{n^{1-\varepsilon}})$ for some $\varepsilon > 0$. Hence, we suspect the following inclusions to be strict:

$$\text{DLIN} \subseteq \text{LIN-PLAN-LOCAL} \subseteq \text{LIN-LOCAL} \subseteq \text{NLIN}$$

Finally, we develop some techniques to prove that a number of seemingly non-local combinatorial problems indeed belong to the classes LIN-LOCAL or LIN-PLAN-LOCAL. E.g., we prove that a cardinality-constrained problem such as VERTEX-COVER is LIN-LOCAL-complete. Surprisingly, we also prove that a connectivity-constrained problem such as PLAN-HAMILTON — the planar version of HAMILTON — is LIN-PLAN-LOCAL-complete, though we conjecture that HAMILTON itself (known to be LIN-LOCAL-hard) is not LIN-LOCAL, since Lautemann and Weinzinger showed the weaker result that it is not local [18], even in the presence of an underlying linear order.

2 LIN-LOCAL problems and SAT

We first define the local structure which is the input of local problems. Each element of the universe $\mathcal{U}$ of such a structure has a list of neighbors. In order that the notion of locality makes sense, we want the length of these lists to be independent of the size of the input. Each element also holds a number of bits of information. This number too must be independent of the size of the input. Otherwise, this information could be used to locally check global properties (e.g., with only $O(\log n)$ bits per elements, a global information such as a linear order over $\mathcal{U}$ can be encoded by storing the rank of each element).

Definition 1 (Local structure) *A* local structure $S = (\mathcal{U}, \mathcal{F}, \mathcal{L})$ *is a finite unary structure of signature* $\sigma = (\mathcal{F}, \mathcal{L})$ *over a finite universe* $\mathcal{U}$*, i.e., a first-order structure equipped with a list* $\mathcal{F} = (f_0, \cdots, f_{k-1})$ *of* k *unary functions* f_i *and a list* $\mathcal{L} = (L_0, \cdots, L_{p-1})$ *of* p *unary relations (predicates)* L_i*. For* $x \in \mathcal{U}$*,* $f_i(x)$ $(i < k)$*, is said to be a* neighbor *of* x*, and the list* $\mathcal{F}(x) =$

$(f_0(x), \cdots, f_{k-1}(x))$ *is called the* neighborhood *of* x. *Also,* $L_i(x)$ $(i < p)$, *is called the* label-bit i *of* x, *and the list of bits* $\mathcal{L}(x) = (L_0(x), \cdots, L_{p-1}(x))$ *is called the* label *of* x. *Moreover, if* S *satisfies the additional properties* $|\mathcal{L}| = 1$ *and* $|\mathcal{F}| = 2$, *with* f_0 *and* f_1 *both bijective, then* S *is called a* strictly local *structure.*

A *local structure* $(\mathcal{U}, \mathcal{F}, \mathcal{L})$ is *planar* if its underlying graph $G(\mathcal{U}, E)$ is planar, with $E = \{(x, y) \subseteq \mathcal{U} \times \mathcal{U},\ \exists f_i \in \mathcal{F},\ y = f_i(x)\}$.

We now introduce the notion of coloring, and then define the class of local problems as the sets of local structures that are witnessed by a coloring whose restriction to an element and its neighborhood locally satisfies a simple boolean formula.

Definition 2 (Coloring) *Given a finite universe* $\mathcal{U}$, *a* coloring $\mathcal{C}$ *over* $\mathcal{U}$ *is a list* $(C_0, \cdots, C_{q-1})$ *of* q *unary relations over* $\mathcal{U}$. *For* $x \in \mathcal{U}$, *the list* $\mathcal{C}(x) = (C_0(x), \cdots, C_{q-1}(x))$ *is called the* color *of* x, *and* $C_i(x)$, $(i < q)$ *is called the color-bit* i *of* x.

Definition 3 (Local problem) *A decision problem* Q *over a set* $\mathcal{S}$ *of (planar) local* $(\mathcal{F}, \mathcal{L})$*-structures is called (planar)* local *and is denoted* $(\mathcal{S}, \mathcal{F}, \mathcal{L}, \mathcal{C}, \psi)$ *if there exists a formula of the monadic-ESO logic with one first-order variable,* $\varphi :\ \exists \mathcal{C}\ \forall x\ \psi(x, \mathcal{F}, \mathcal{L}, \mathcal{C})$ *such that for each* $S = (\mathcal{U}, \mathcal{F}, \mathcal{L}) \in \mathcal{S}$, $S \in Q$ *iff* $S \models \varphi$, *where* $\mathcal{C}$ *is a list of unary relation symbols,* x *is the unique first-order variable, and* ψ *is an arbitrary quantifier-free first-order formula involving only* x, $\mathcal{C}$ *and the signature* $(\mathcal{F}, \mathcal{L})$. *Both* φ *and* ψ *are called* local formulas. *Moreover,* Q *is called* strictly local *if* $\mathcal{S}$ *is a set of strictly local structures,* $|\mathcal{C}| = 1$, *and* ψ *does not contain terms that are functional compositions.*

As previously argued, local problems cannot represent any consistent time complexity class if they are not closed under DLIN reductions. Local problems are also too sensitive to the representation of the input. This justifies the following definition:

Definition 4 (LIN-LOCAL class and LIN-PLAN-LOCAL class) *A decision problem* Q *is* LIN-LOCAL *(resp. LIN-PLAN-LOCAL) if it is DLIN-reducible to a local (resp. planar local) problem* Q'.

The LIN-(PLAN-)LOCAL classes trivially contain DLIN, and the sensitiveness to the representation of the input is removed. Several things are worth to be noted concerning their robustness:

Theorem 5 • *SAT is LIN-LOCAL-complete under DLIN-reductions.*
• *PLAN-SAT is LIN-PLAN-LOCAL-complete under DLIN-reductions.*

Moreover, a refined proof of the LIN-(PLAN)-LOCALITY of (PLAN-)SAT leads to a strictly local (planar) problem, and hence:

Theorem 6 *Any LIN-(PLAN)-LOCAL problem is DLIN-reducible to a strictly local (planar) problem.*

We now give the elements of the proof of Theorem 5. We first show that SAT (resp. PLAN-SAT) is LIN-LOCAL (resp. LIN-PLAN-LOCAL). This proof is too weak for the aims of Theorem 6 because it uses two color-bit symbols and two label-bit symbols. A stronger proof, leading to Theorem 6, is presented in Appendix, due to the lack of space:

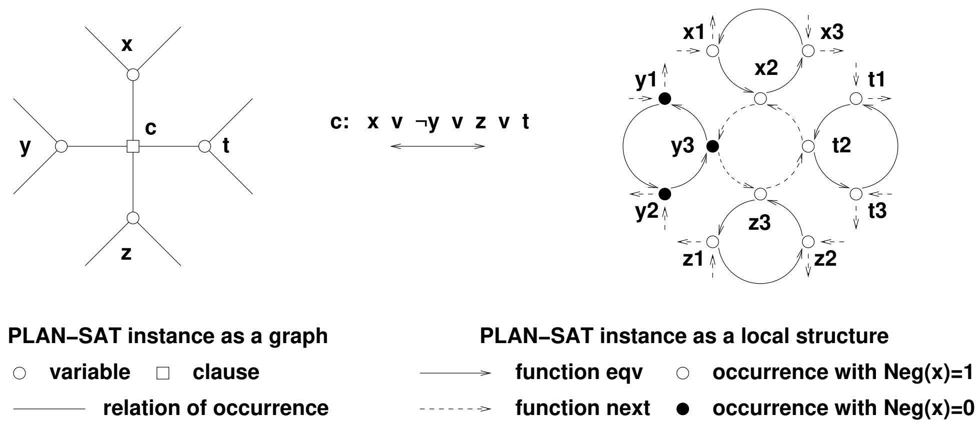

Fig. 1. The reduction from SAT to a local problem

Proof ((PLAN-)SAT is LIN-(PLAN-)LOCAL). We DLIN-reduce any SAT instance to a $(\mathcal{U}, \mathcal{F}, \mathcal{L})$-structure where $\mathcal{U}$ is the set of occurrences of variables, $\mathcal{L} = (Neg, First)$ and $\mathcal{F} = (eqv, next)$. Neg maps the negative occurrences to 1, and the others to 0. $First$ maps the first occurrences in clauses to 1, and the others to 0. eqv chains the occurrences of a same variable into a directed cycle. $next$ chains the occurrences of a same clause into a directed cycle. The construction is clearly DLIN and can be made planarity-preserving as shown in Figure 1. The local formula uses two colors $True$ and One to guess a satisfying assignment I: $True$ maps the occurrences to their truth-value in I. One maps the occurrences to the "or"-accumulated truth-values of all their predecessors in the same clause. A clause is satisfied iff One maps its last occurrence to 1. Note that the last occurrence of a clause is the one which satisfies $First(next(x))$, i.e. the one whose successor is the first occurrence in the clause. The local formula φ is $\exists\, True, One\ \forall x$:

$$
\begin{aligned}
& [True(x) = True(eqv(x)) \iff Neg(x) = Neg(eqv(x))] \\
\wedge\ & [First(x) \Longrightarrow One(x) = True(x)] \\
\wedge\ & [\neg First(next(x)) \Longrightarrow One(next(x)) = (True(next(x)) \vee One(x))] \\
\wedge\ & [First(next(x)) \Longrightarrow One(x)] \quad \square
\end{aligned}
$$

The LIN-LOCAL-hardness of SAT comes from a simple development of ψ over $\mathcal{U}$ and is not shown here. The proof of the LIN-PLAN-LOCAL-hardness of PLAN-SAT, more technically involved since planarity must be preserved, is presented in Appendix. It mainly needs the following lemma:

Lemma 7 *Let $(\mathcal{F}, \mathcal{L})$ be a unary signature, and $\varphi(\mathcal{F}, \mathcal{L}, \mathcal{C})$ be a local formula. Then, a new local formula $\varphi'(\mathcal{F}, \mathcal{L}, \mathcal{C}')$ can be computed such that, for any structure S of signature $(\mathcal{F}, \mathcal{L})$, $S \models \varphi$ iff $S \models \varphi'$, and its quantifier-free part ψ' is in CNF and contains no functional composition.*

Proof of Lemma 7. We first remove all functional compositions from φ, i.e. atomic formulas of the form $P(g_n \circ g_{n-1} \circ \cdots \circ g_1(x))$ with $n \geq 2$, $\{g_1, \cdots, g_n\} \subseteq \mathcal{F}$, and $P \in \mathcal{L} \cup \mathcal{C}$. The idea is to export all the needed bits to the immediate neighbors of x, by slightly augmenting the coloring $\mathcal{C}$ to a new coloring $\mathcal{C}'$. Start with $\psi' = \psi$, and $\mathcal{C}' = \mathcal{C}$. For each atomic formula α in ψ of the form $P(g_n \circ g_{n-1} \circ \cdots \circ g_1(x))$ with $n \geq 2$ and $P \in \mathcal{L} \cup \mathcal{C}$:

- add to $\mathcal{C}'$ the color-bits $P^{g_n}, P^{g_n \circ g_{n-1}}, \cdots, P^{g_n \circ \cdots \circ g_2}$;
- replace α in ψ' by $P^{g_n \circ \cdots \circ g_2}(g_1(x))$;
- add the conjunct $P^{g_n}(x) = P(g_n(x))$ to ψ';
- For each $2 \leq j < n$, add the conjunct $P^{g_n \circ \cdots \circ g_j}(x) = P^{g_n \circ \cdots \circ g_{j+1}}(g_j(x))$ to ψ'.

Now, ψ' has no functional composition, and it can be transformed into a CNF formula in the standard way: write the truth table of ψ' and generate one clause per non-satisfying assignment. $\square$

Note that since the size of φ is fixed, the size of the new formula φ' is also fixed and its computation takes constant time. The last part of the reduction consists in building a planar SAT-system of size $O(d(x))$ to simulate the constraint ψ around an element x of total degree $d(x) = d^-(x) + d^+(x)$. This makes use of Lichtenstein's planar crossover-box for SAT [19].

3 LIN-LOCAL problems and cardinality problems

A cardinality constraint on a unary relation C_i is a constraint of the form $(\#C_i \perp K)$ where $\#C_i$ is the cardinality of C_i, K is either a constant $\leq |\mathcal{U}|$ or the cardinality of another unary relation, and $\perp$ is a binary relation in $\{=, \neq, <, \leq, >, \geq\}$. A cardinality problem is a problem characterized by both local constraints and cardinality constraints, i.e. by some formula of the form $\exists \mathcal{C}\ (\forall x\ \psi_1) \wedge \psi_2$ where ψ_1 is a quantifier-free local formula and ψ_2 is a conjunction of cardinality constraints. Numerous natural NP-complete problems such as VERTEX-COVER, DOMINATING-SET, MAX-SAT, etc. [7] are cardinality problems. In this section, we give a simple argument to show that these problems are indeed LIN-LOCAL. In [16], Hunt et al. give a planarity-preserving and weakly parsimonious reduction from 1-EXACTLY-AMONG-MONOTONE-3SAT to VERTEX-COVER in order to show that #PLAN-VERTEX-COVER is #P-complete. Since this reduction is DLIN and 1-EX-MONO-3SAT is DLIN-equivalent to SAT, this also shows the LIN-LOCAL-completeness (resp. LIN-PLAN-LOCAL-hardness) of VERTEX-COVER (resp. PLAN-VERTEX-COVER). In this section we refine this reduction to make it parsimonious, and hence show the DP-completeness of UNIQUE-PLAN-VERTEX-COVER (under randomized polynomial reductions).

Theorem 8 *All the cardinality problems are LIN-LOCAL.*

Sketch of proof of Theorem 8. The main problem consists in building in linear time a gadget of size $O(|\mathcal{U}|)$ which outputs a list of $\ell = O(\log n)$ variables holding the cardinality of some unary predicate C_i. W.l.g., assume that $n = |\mathcal{U}| = 2^\ell$. Our additioner uses a divide-and-conquer strategy with $\ell+1$ levels (numbered from 0 to ℓ): The level 0 consists of a list of 2^ℓ 1-bits numbers $(X_0^0, \cdots, X_0^{2^\ell-1})$, namely the bits of $C_i(\mathcal{U})$ itself. For any $1 < k \leq \ell$, the level k is formed of $2^{\ell-k}$ numbers $(X_k^0, \cdots, X_k^{2^{\ell-k}-1})$, such that $X_k^j = X_{k-1}^{2j} + X_{k-1}^{2j+1}$. Since the sum of two numbers of b bits fits in $b+1$ bits, the numbers at level k all have $k+1$ bits. This way, the list of level $k = \ell$ consists of a single number X_k^0 of $\ell+1$ bits holding $\#C_i$. Encoding all the binary additions with a carry-propagation scheme takes space and time $O(t(n))$ where $t(n)$ is the total number of bits over all levels, and $t(n) = \sum_{k=0}^{\ell}(k+1)2^{\ell-k} = O(n)$ as required. Finally, it is straightforward to build gadgets of size $O(\ell)$ implementing arithmetic circuits to constrain any comparison $\perp$ between any output cardinalities or constants. $\square$

Theorem 9 • *VERTEX-COVER is LIN-LOCAL-complete.*
• *PLAN-VERTEX-COVER is LIN-PLAN-LOCAL-hard.*

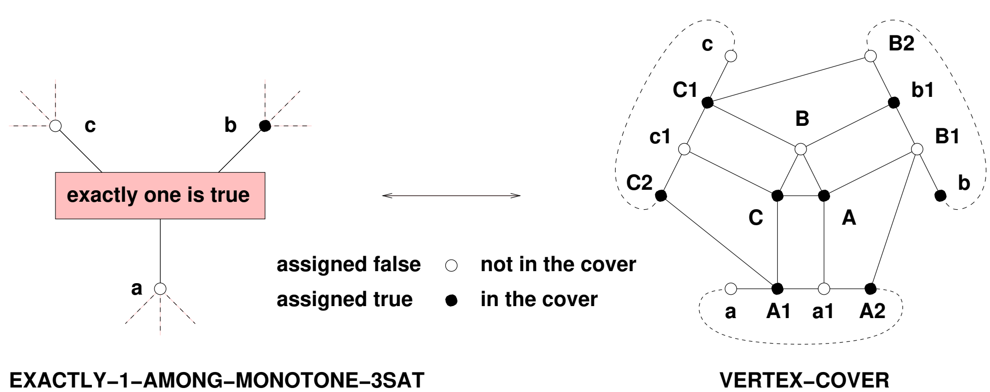

Fig. 2. The reduction from 1-EX-MONO-3SAT to VERTEX-COVER

Sketch of proof of Theorem 9. We establish the result by a planarity-preserving DLIN-reduction from 1-EX-MONO-3SAT to VERTEX-COVER. Let I be an input of 1-EX-MONO-3SAT with m 3-clauses (and hence $3m$ occurrences of variables). Our output-graph G for VERTEX-COVER has $15m$ vertices and we ask for a cover K of cardinality $\leq 8m$. Each variable x in I of degree d (i.e. occurring d times) has an associated even cycle e_x in G of length $4d$ (i.e. 4 vertices by occurrence), and each 3-clause r in I has an associated triangle t_r in G. Occurrence-vertices are connected to 3-clause-vertices according to Figure 2. The truth-values of the variables a, b, c in I are witnessed by the membership to K of the corresponding vertices a, b, c in G. Simple arguments of cardinality imply the correspondence between the configurations in I and G depicted in Figure 2. □

By lack of space, the end of the proof is detailed in Appendix. Since the transformation above turns out to be parsimonious, and since UNIQUE-PLAN-1-EX-MONO-3-SAT is DP-complete under randomized polynomial reductions [16], we conclude that:

Corollary 10 *UNIQUE-PLAN-VERTEX-COVER is DP-complete under randomized polynomial reductions.*

4 LIN-PLAN-LOCAL problems and PLAN-HAMILTON

In this section, we show that HAMILTON – which can be viewed as a DLIN-LOCAL problem plus an additional global constraint of connectivity on the solutions – becomes LIN-PLAN-LOCAL when restricted to planar instances. This is shown by a DLIN-reduction from PLAN-HAMILTON to PLAN-SAT. A converse reduction also exists and gives us the LIN-PLAN-LOCAL-completeness of PLAN-HAMILTON. We give a sketch of the proof of the LIN-PLAN-LOCALITY of PLAN-HAMILTON and omit the proof of its LIN-PLAN-LOCAL-hardness due to space-reasons.

Theorem 11 *PLAN-HAMILTON is LIN-PLAN-LOCAL-complete.*

Both reductions are detailed in [1,2] and turn out to be parsimonious. In particular, the later gives us the DP-completeness of UNIQUE-PLAN-HAMILTON and answers a question stated as open in [16]. Although we only deal with the undirected cycle version of PLAN-HAMILTON, all those results also hold for the directed versions, the path versions (with specified or unspecified ends), and the degree-bounded versions, because all these planar versions are equivalent under parsimonious DLIN reductions [1].

Corollary 12 *UNIQUE-PLAN-HAMILTON is DP-complete under randomized polynomial reductions.*

Sketch of proof (PLAN-HAMILTON is LIN-PLAN-LOCAL) The problem of the (planar) Hamiltonian partition into disjoint cycles is easily DLIN-reducible to (PLAN-)SAT. However, SAT does not seem to be able to detect if there is only one cycle in the general (i.e., non planar) case. We show that it is indeed possible in the plane. Let G be the embedding of a connected planar graph, and H be a Hamiltonian partition in G. Both G and H are viewed as sets of edges. Denote G' (resp. H') the set of dual edges of G (resp. of H). Define $D = G' - H'$. D and H satisfy the following lemma whose proof is omitted (see Figure 3):

Lemma 13 *H has exactly one cycle iff D is a forest of exactly two trees.*

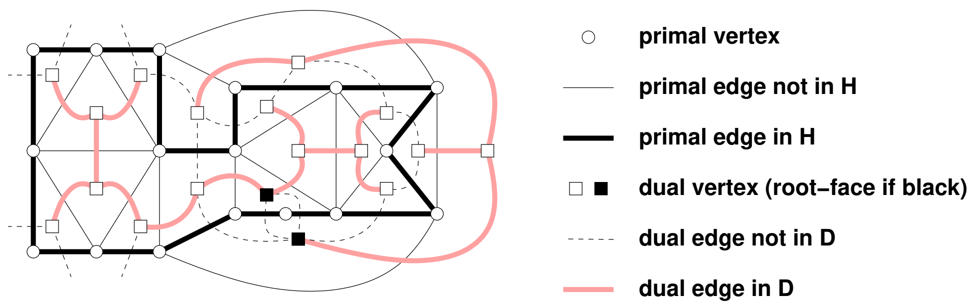

Fig. 3. Planar Hamiltonian cycle

Using local constraints in the dual graph of G, it is possible to force that D contains exactly two trees plus some optional non-tree components: the idea is to constrain each face (except for two root-faces that can be required to be adjacent) to have exactly one father among its adjacent faces in a same connected component of D. There will be exactly two trees in D because we only allow our two chosen faces to be roots. However, other non-tree components can occur in D that consist of trees whose roots are connected into a single circuit. These parasite components remain undetected because they do not have roots. However, it is easy to see that if such a component exists, then the roots of our two trees cannot be adjacent anymore (proof omitted: see Figure 4). Hence no parasite component can occur in D if our two root-faces are chosen so that they are adjacent.

We now show how to choose our two roots, i.e. two adjacent faces that are guaranteed to be in two distinct regions defined by any Hamiltonian partition H: if there exists a vertex x of degree 2 in G, then its two incident edges must be in H, which means that we can choose the two adjacent faces of x for our roots. Otherwise, choose any vertex of degree d, and explode it into the gadget of Figure 5. This produces d vertices of degree 2, and any of them can be chosen as above to separate our two root-faces. It is easy to show that there

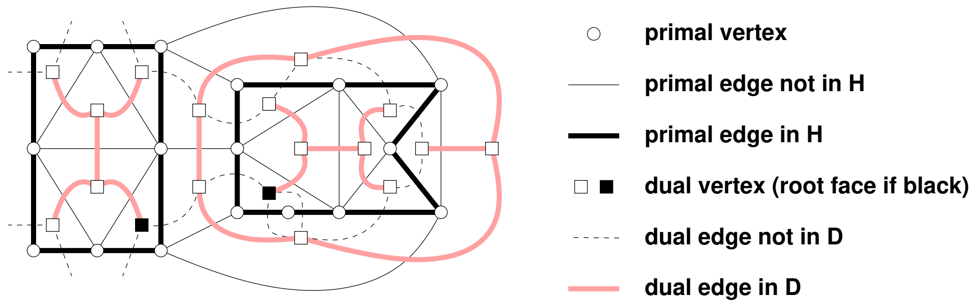

Fig. 4. Planar Hamiltonian partition into 2 disjoint cycles

is a bijection between the respective Hamiltonian cycles in G before and after the explosion, following the patterns (a) and (b) of Figure 5 (proof omitted). The precise transformation from PLAN-HAMILTON to PLAN-SAT is shown in Appendix.

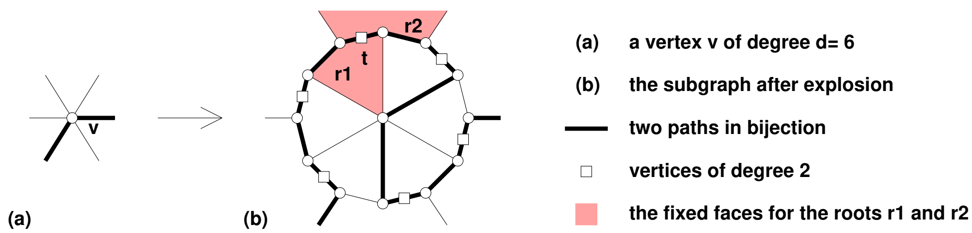

Fig. 5. Explosion of one vertex of degree d in order to produce vertices of degree 2

5 Conclusion

We have defined a new complexity class denoted LIN-LOCAL (resp. LIN-PLAN-LOCAL). It has been proved that this class is rather robust and is exactly the problems DLIN-reducible to SAT. (resp. PLAN-SAT). We argue that it is the minimal time complexity class that includes (planar) local problems, i.e., problems about (planar) local structures – roughly, (planar) bounded-degree labeled graphs – that are recognized by parallel algorithms using a constant number of one-bit nondeterministic steps. Surprisingly, we have proved that the class LIN-LOCAL (resp. LIN-PLAN-LOCAL) includes some seemingly non-local problems such as cardinality problems (e.g. VERTEX-COVER) (resp. connectivity problems such as PLAN-HAMILTON) which are also complete in this class via DLIN reductions. In contrast, we conjecture that the general (i.e., non-planar) problem HAMILTON is not LIN-LOCAL.

References

[1] R. Barbanchon. Planar hamiltonian problems and linear parsimonious reductions. Research report, Les Cahiers du GREYC 1, 2001. (postscript available at http://www.info.unicaen.fr/algo/publications).

[2] R. Barbanchon. The problems sat and hamilton are equivalent under linear parsimonious reductions. Research report, Les Cahiers du GREYC 4, 2001. (postscript available at http://www.info.unicaen.fr/algo/publications).

[3] Nadia Creignou. The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving NP-completeness. *Theoretical Computer Science*, 145(1–2):111–145, 1995.

[4] A. K. Dewdney. Linear time transformations between combinatorial problems. *Internat. J. Computer Math.*, 11:91–110, 1982.

[5] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity and Computation*, 7:43–73, 1974.

[6] R. Fagin. Monadic generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:89–96, 1975.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W.H. Freeman and Co., 1979.

[8] E. Grädel. On the notion of linear time computability. *International Journal of Foundations of Computer Science*, 1, 1990.

[9] E. Grandjean. A nontrivial lower bound for an NP problem on automata. *SIAM Journal on Computing*, 19:438–451, 1990.

[10] E. Grandjean. Invariance properties of RAMs and linear time. *Computational Complexity*, 4:62–106, 1994.

[11] E. Grandjean. Linear time algorithms and NP-complete problems. *SIAM Journal on Computing*, 23(3):573–597, 1994.

[12] E. Grandjean. Sorting, linear time and the satisfiability problem. *Annals of Mathematics and Artificial Intelligence*, 16:183–236, 1996.

[13] E. Grandjean and F. Olive. Monadic logical definability of nondeterministic linear time. *Computational Complexity*, 7(1):54–97, 1998.

[14] E. Grandjean and T. Schwentick. Machine-independent characterizations and complete problems for deterministic linear time. Research report, Les Cahiers du GREYC 10, 1999. submitted to SIAM Journal on Computing (postscript available at http://www.info.unicaen.fr/algo/publications).

[15] Y. Gurevich and S. Shelah. Nearly linear time. In *Proceedings of the Symposium on Logical Foundations of Computer Science*, volume 363 of *LNCS*, pages 108–118. Springer, 1989.

[16] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, and R. E. Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27(4):1142–1167, August 1998.

[17] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[18] C. L. Lautemann and B. Weinzinger. Monadic-NLIN and quantifier-free reductions. In *CSL, 8th annual conference of the EACSL, Lect. Notes Comput. Sci.*, volume 1683 of *LNCS*, pages 322–337, 1999.

[19] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[20] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

[21] S. S. Ravi and H. B. Hunt, III. An application of the planar separator theorem to counting problems. *Information Processing Letters*, 25(5):317–321, 1987.

[22] K. W. Regan. Machine models and linear time complexity. *SIGACT News*, 24(4):5–15, 1993.

[23] C. P. Schnorr. Satisfiability is quasilinear complete in NQL. *Journal of the ACM*, 25(1):136–145, 1978.

[24] T. Schwentick. Algebraic and logical characterizations of deterministic linear time classes. In *14th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1200 of *LNCS*, pages 463–474. Springer, 1997.

[25] T. Schwentick. Descriptive complexity, lower bounds and linear time. In *Lecture Notes in Computer Science*, volume 1584 of *LNCS*, pages 9–28. Springer, 1999.

[26] R. E. Stearns and H. B. Hunt III. Power indices and easier hard problems. *Mathematical Systems Theory*, 23(4):209–225, 1990.

A Appendix: Technical proofs

Proof of Theorem 5: (PLAN-)SAT is LIN-(PLAN)-LOCAL, strengthened version leading to Theorem 6. We DLIN-reduce any SAT-instance to a strictly local $(\mathcal{U}, \mathcal{F}, \mathcal{L})$-structure S where $\mathcal{F} = (next, link)$, $\mathcal{L} = (Occ)$, and each function $next$, $link$ is a permutation of $\mathcal{U}$. For each clause c, $\mathcal{U}$ contains two elements T_c, F_c which are meant to hold the true and false constants. For each occurrence of a variable v in a clause c , $\mathcal{U}$ contains an element $a_{v,c}$ (called accumulator) and two elements $p_{v,c}$ and $n_{v,c}$ which are meant to represent x and $\neg x$ (exactly one of them will be indeed used by c). Occ is primarily the label for occurrences: it maps all the $p_{v,c}$ and $n_{v,c}$ to 1 and all the $a_{v,c}$ to 0. The trick is that it also maps all the T_c to 0 and all the F_c to 1. For each variable v, $next$ alternately chains: all the $p_{v,c}$ and $n_{v,c}$ in a directed cycle, and for each clause c, $next$ chains: T_c, all the accumulators $a_{v,c}$ and F_c in this order in a directed cycle. $link$ primarily binds occurrences to accumulators: if a variable v occurs positively in a clause c then we define $link(p_{v,c}) = a_{v,c}$, $link(a_{v,c}) = p_{v,c}$, and the self-loop $link(n_{v,c}) = n_{v,c}$ (the symmetric case happens if v occurs negatively in c). Another trick is that for each clause c, we define the 2-cycle $link(T_c) = F_c$, $link(F_c) = T_c$. The construction is clearly DLIN and can be made planarity-preserving as shown in Figure A.1.

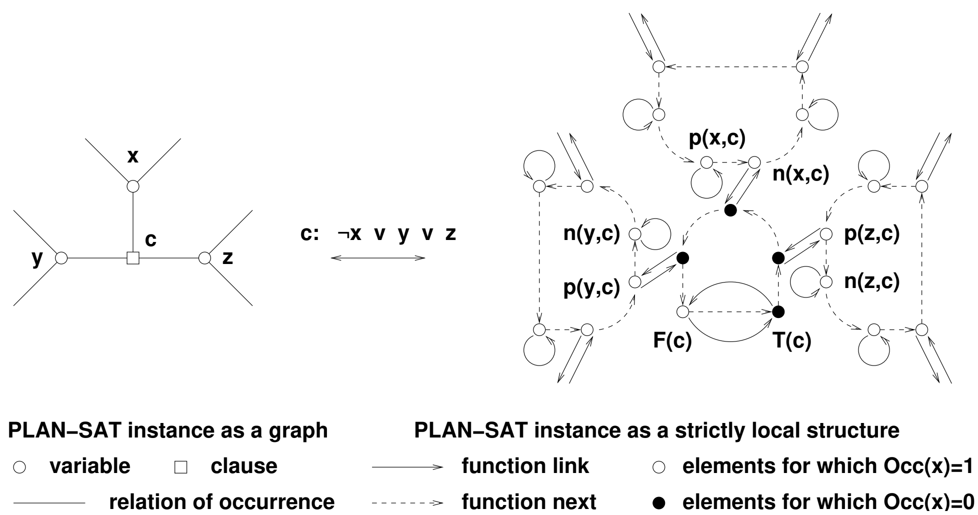

Fig. A.1. The enhanced reduction from SAT to a strictly local problem

The local formula uses one color $True$ which holds the truth values of all the $p_{v,c}$ and $n_{v,c}$. For all the T_c (resp. F_c) it will be shown to be 1 (resp. 0), and for any accumulator $a_{v,c}$ it will hold the accumulated truth-values of the occurrences linked to all its $next$-successors up to F_c. The local formula φ is $\exists\, True\ \forall x$:

$$[Occ(x) \Longrightarrow True(next(x)) = \neg True(x)]$$

$\wedge\ [\neg Occ(x) \implies (True(x) = True(next(x)) \vee True(link(x)))]$

The first constraint forces that all the $n_{v,c}$ and $p_{v,c}$ of a same variable v have opposite values. Also, since $Occ(F_c) = 1$ and $next(F_c) = T_c$ for any clause c, it forces that F_c and T_c have opposite values. The second constraint forces that for each clause c the bit $True$ is non-decreasing when following the arrows $next$ backwards from F_c to T_c (including $True(T_c)$ because $Occ(T_c) = 0$). Since $True(F_c) \neq True(T_c)$ because of the first constraint, this implies that $True(F_c) = 0$ and $True(T_c) = 1$. This also means that $True(T_c)$, which accumulates the truth-bit of the final occurrence and the truth-bit of F_c, indeed holds a copy of the truth-bit of the final accumulator. It follows that there is at least one $a_{v,c}$ such that $True(a_{v,c}) = 1$, i.e. such that the truth value of v (represented by $True(n_{v,c})$ and $True(p_{v,c})$) satisfies c. Furthermore, the reader can easily check that no constraint circulates through the arcs of type $link(F_c)$, $link(a_{v,c})$, and $link$ self-loops (i.e., the left-hand side of the implication of the second constraint always evalute to 0 in these cases). □

Proof of Theorem 5: (PLAN-)SAT is LIN-(PLAN-)LOCAL-hard. A planar crossover-box for PLAN-SAT is a standard device [16,19] which embeds 4 variables in the corners of a square, and constrains each opposite pair to hold the same Boolean value, independently of the other pair. Given a list of $r = O(1)$ boolean variables aligned along the bottom side of a square, it is straightforward to build a gadget (called DUP) of size $O(r^2) = O(1)$ which duplicates these variables along the two vertical sides of this square by using $O(r^2) = O(1)$ crossover-boxes, as shown in Figure A.2. Then, given a (non necessarily planar) CNF ψ of $s = O(1)$ clauses over these r variables, it is easy to build a gadget (called PSI) of size $O(1)$ embedding ψ in the plane using $O(s) = O(1)$ gadgets DUP, as shown in Figure A.3. Let $Q = (\mathcal{U}, \mathcal{F}, \mathcal{L}, \mathcal{C}, \psi)$ be the LIN-LOCAL instance for which we want a planarity-preserving DLIN reduction to SAT. We assume that ψ satisfies Lemma 7, i.e., ψ is a CNF without functional composition. For each element $x \in \mathcal{U}$ of total degree $d(x) = d^-(x) + d^+(x)$, let us build the gadget (called VAR) of size $O(d(x))$ shown in Figure A.4, which has d slots, each formed of $(p+q)(k+1) = O(1)$ variables, where $p = |\mathcal{L}|$, $q = |\mathcal{C}|$ and $k = |\mathcal{F}|$. These variables hold $\mathcal{C}(x)$, $\mathcal{L}(x)$ as well as $\mathcal{C}(\mathcal{F}(x))$ and $\mathcal{L}(\mathcal{F}(x))$. The bits of each slot are constrained to be copies of each other by $O(d(x))$ DUP gadgets and are forced to satisfy ψ by one PSI gadget. The reduction is completed by properly connecting the slots to import into a VAR gadget the bits of its neighbors. For each $x, y \in \mathcal{U}$ and $f_i \in \mathcal{F}$ such that $y = f_i(x)$, reserve one slot s_x (resp. s_y) in the VAR gadget of x (resp. y) and merge the $p+q$ variables reserved in s_x for the bits of $f_i(x)$ with the $p+q$ variables reserved in s_y for bits of y. The global construction is clearly DLIN and can be made planarity preserving as in Figure A.5. □

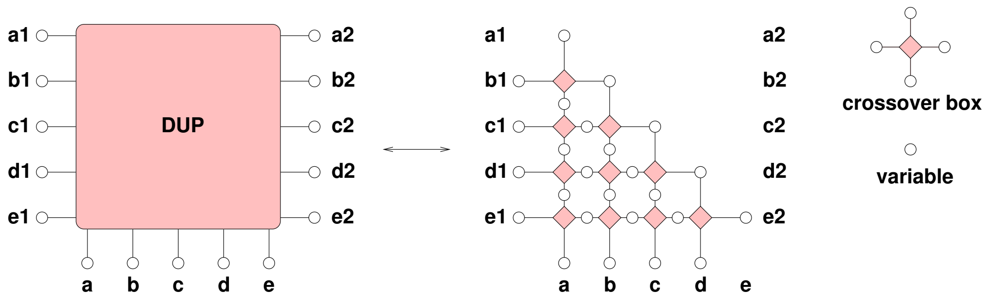

Fig. A.2. The PLAN-SAT gadget (called DUP) that duplicates a list of 5 Boolean variables $l = (a, b, c, d, e)$ into $l_1 = (a_1, b_1, c_1, d_1, e_1)$ and $l_2 = (a_2, b_2, c_2, d_2, e_2)$

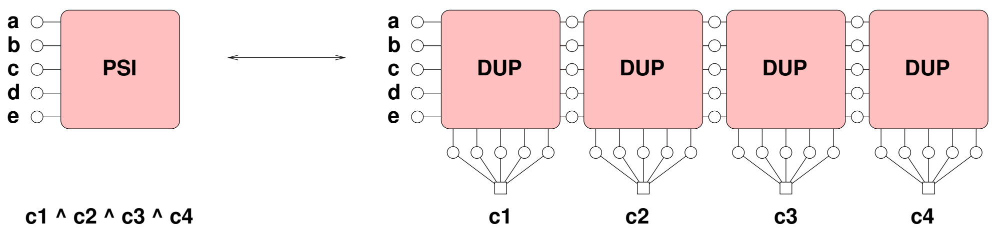

Fig. A.3. The PLAN-SAT gadget (called PSI) to simulate a conjunction of clauses $\psi : c_1 \wedge c_2 \wedge c_3 \wedge c_4$ with 5 variables a, b, c, d, e.

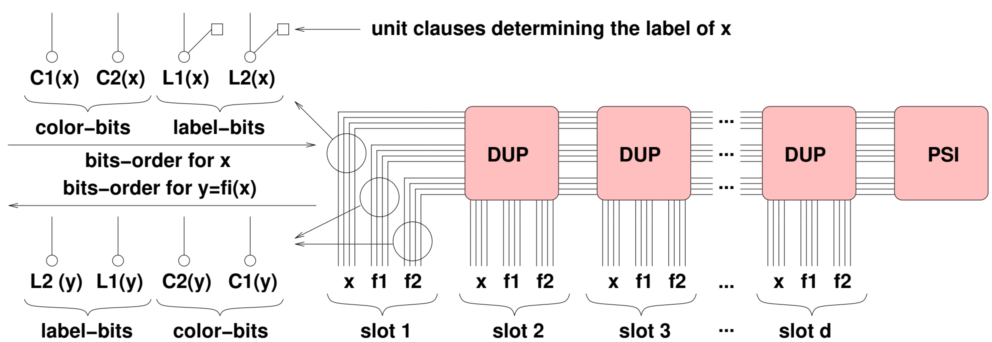

Fig. A.4. The PLAN-SAT gadget (called VAR) for an element $x \in \mathcal{U}$ of total degree d, in a local structure $(\mathcal{U}, (f_1, f_2), (L_1, L_2))$, and subject to a local formula $\psi(x, \mathcal{F}, \mathcal{L}, (C_1, C_2))$ in CNF without functional composition.

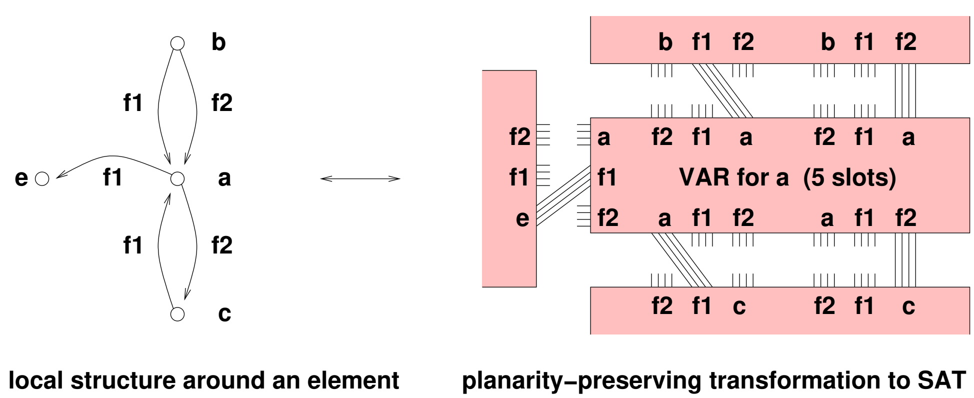

Fig. A.5. The use of the gadget VAR in the reduction from a local problem to SAT

Proof of Theorem 9: VERTEX-COVER is LIN-LOCAL-hard (continued). Note that, in any triangle t_r, at least two of its vertices must be in K. Also, in any even cycle e_x, at least half of its vertices must be in K, and this local optimum is reached only by the two alternating 2-colorings in such an even cycle. Thus, all the e_x together require that at least $4 \times 3m/2 = 6m$ of their vertices be in K, and the sum of all the t_r require at least $2m$ of their vertices to be in K. The maximum cardinality of $8m$ is reached, and all the e_x and t_r are disjoint. Hence for any K, the membership must be alternated along each e_x, and exactly 2 vertices per t_r must be in K. Following the notations of Figure 2: The alternating 2-coloring law in all the e_x ensures that all copies of a, b and c along their cycle get the same assignment. Without loss of generality, suppose that vertex $B \notin K$ and vertices $A, C \in K$ (the two other possible configurations in the triangle are symmetric). Then vertices $C_1, b_1 \in K$ since edges (B, C_1) and (B, b_1) must be covered. The alternating 2-coloring law in all the e_x then implies that vertices $c, c_1, B_1, B_2 \notin K$ and $C_1, C_2, b, b_1 \in K$. It follows that $A_2 \in K$ since the edge (A_2, B_1) must be covered. Finally, $A_1 \in K$ and $a, a_1 \notin K$ because of the alternating 2-coloring law in all the e_x. Uppercase and lowercase vertices end by having opposite membership statutes. This establishes the unique correspondence between the configuration $(a = 0, b = 1, c = 0)$ in I and the configuration $(a \notin K, b \in K, c \notin K)$ in G. □

Transformation from PLAN-HAMILTON to PLAN-SAT in the proof of Theorem 11. For the sake of readability, we assume that the special clauses $1/N(\ell_1, \cdots, \ell_d)$ and $2/N(\ell_1, \cdots, \ell_d)$ – which are satisfied iff exactly one (resp. two) literal among the ℓ_i $(1 \leq i \leq d)$ are assigned true – are available (these special clauses are easy to implement and embed in the plane using standard clauses). Let $G(V, E)$ be a planar embedding of the input graph, $G'(V', E')$ be the dual graph of G, H be a subset of E, H' be the dual of H and $D = E' - H'$. Denote r_1 and r_2, the two chosen adjacent root-faces. Here is the SAT-system satisfied iff G is Hamiltonian. (see also Figure 3):

(0) *Set of variables:* Each edge $e \in E \cup E'$ has an associated Boolean variable $thick_e$, asserting that "$e \in H \cup D$". Each face $f \in V'$ of degree d has d associated Boolean variables $father_f^{e'}$, one for each edge $e' = (f, g) \in E'$, asserting that "$e' \in D$ and g is the father of f in D".

(1) *H is a Hamiltonian partition of G:* For each vertex $v \in V$ of degree d, with incident edges $e_1, \cdots, e_d$, generate the constraint $2/N(thick_{e_1}, \cdots, thick_{e_d})$.

(2) *D equals $E' - H'$:* For each edge $e \in E$ and its dual edge e', generate the clauses $(thick_e \vee thick_{e'})$ and $(\neg thick_e \vee \neg thick_{e'})$.

(3) *Each face distinct from r_1 and r_2 has exactly one father:* For each face $f \in V'$ of degree d, $f \notin \{r_1, r_2\}$, with incident edges $e'_1, \cdots, e'_d$, generate the constraint $1/N(father_f^{e'_1}, \cdots, father_f^{e'_d})$.

(4) *Both adjacent roots r_1 and r_2 have no father:* For each edge $e' \in E'$ incident to a root $r \in \{r_1, r_2\}$, generate the unit clause $(\neg father_r^{e'})$.
(5) *D is consistently oriented:* For each edge $e' = (f, g) \in E'$, generate the constraints $(father_f^{e'} \implies thick_{e'})$, $(father_g^{e'} \implies thick_{e'})$, $(thick_{e'} \implies father_f^{e'} \vee father_g^{e'})$, and $(\neg father_g^{e'} \vee \neg father_f^{e'})$.

Figure A.6 shows how to embed our SAT-system in the plane for each face of G. Again, we use Lichtenstein's parsimonious crossover-box [19] to duplicate variables where dual edges cross their respective primal edges.

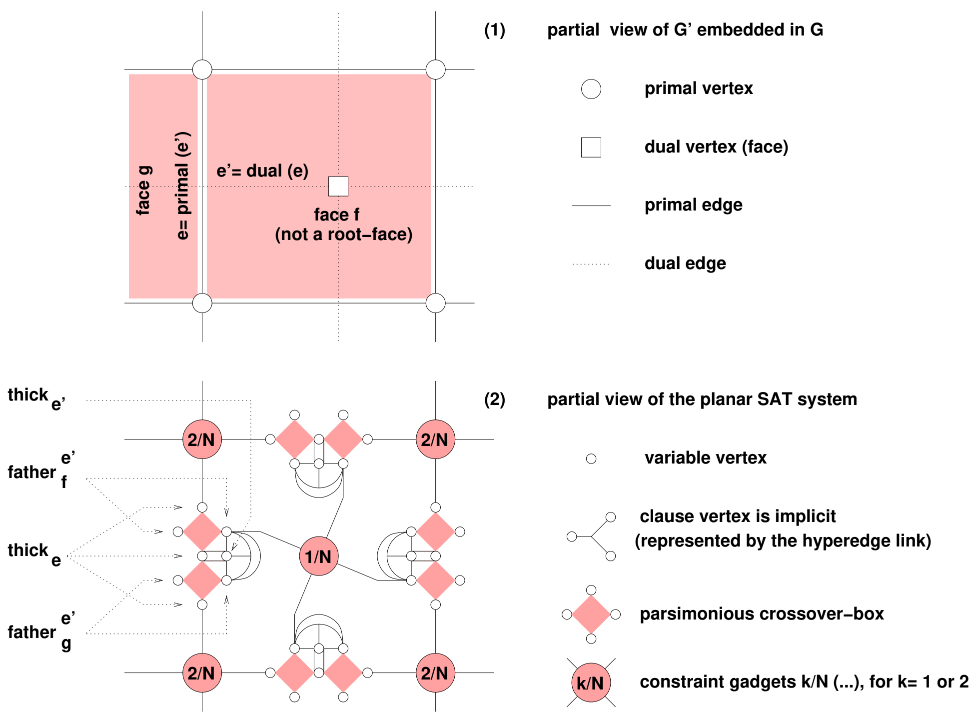

Fig. A.6. Partial view of the reduction from PLAN-HAMILTON to PLAN-SAT, here around a square face of the input graph.