Available online at www.sciencedirect.com
ScienceDirect
Discrete Mathematics 307 (2007) 2030–2041
DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Can transitive orientation make sandwich problems easier?

Michel Habib[a], David Kelly[b], Emmanuelle Lebhar[c,1], Christophe Paul[a,*,1]

[a] CNRS, LIRMM, Université Montpellier II, 161 rue Ada, 34 392 Montpellier Cedex, France
[b] Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2
[c] Laboratoire de l'Informatique du Parallélisme, École Normale Supérieure de Lyon, 46 Allée d'Italie, 69364 Lyon Cedex 07, France

Received 8 September 2003; received in revised form 14 November 2004; accepted 12 December 2005
Available online 13 December 2006

Abstract

A graph $G_s = (V, E_s)$ is a sandwich for a pair of graphs $G_t = (V, E_t)$ and $G = (V, E)$ if $E_t \subseteq E_s \subseteq E$. A sandwich problem asks for the existence of a sandwich graph having an expected property. In a seminal paper, Golumbic et al. [Graph sandwich problems, J. Algorithms 19 (1995) 449–473] present many results on sub-families of perfect graphs. We are especially interested in comparability (resp., co-comparability) graphs because these graphs (resp., their complements) admit one or more transitive orientations (each orientation is a partially ordered set or poset). Thus, fixing the orientations of the edges of G_t and G restricts the number of possible sandwiches. We study whether adding an orientation can decrease the complexity of the problem. Two different types of problems should be considered depending on the transitivity of the orientation: the poset sandwich problems and the directed sandwich problems. The orientations added to both graphs G and G_s are transitive in the first type of problem but arbitrary for the second type.

Keywords: Graph sandwich; Comparability graphs; Partially ordered set

1. Introduction

A graph $G_t = (V, E_t)$ is a *spanning sub-graph* of $G = (V, E)$ if $E_t \subseteq E$. A graph $G_s = (V, E_s)$ is a *sandwich graph* for the pair (G_t, G) if $E_t \subseteq E_s \subseteq E$. Golumbic et al. [8] introduced the following decision problem:

Problem 1. GRAPH SANDWICH PROBLEM FOR PROPERTY Π.
Instance: Two graphs $G_t = (V, E_t)$ and $G = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist a sandwich graph $G_s = (V, E_s)$ for the pair (G_t, G) satisfying property Π?

In their paper, Golumbic et al. present as an example, the *directed Eulerian sandwich problem* where the input graphs are digraphs. Since a digraph is Eulerian iff the in-degree and the out-degree of any vertex are equal, it is possible to design a polynomial time algorithm to decide the existence of an Eulerian sandwich digraph. Unfortunately, many sandwich problems can be proved to be NP-complete. Graph sandwich problems can be thought of as the generalization

* Corresponding author.
E-mail address: paul@lirmm.fr (C. Paul).
[1] Authors supported in part by the European Research Training Network COMBSTRU (Combinatorial Structure of Intractable Problems).

0012-365X/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.disc.2005.12.048

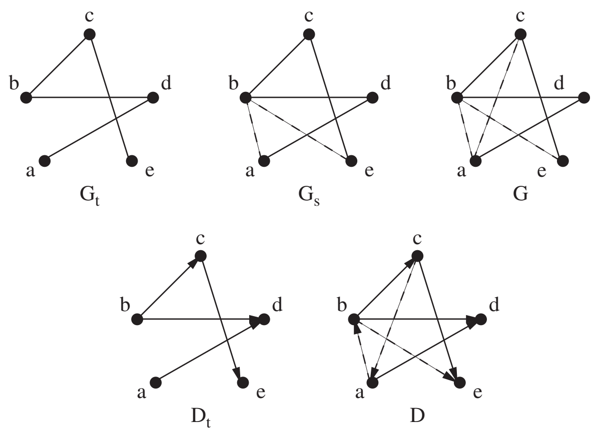

Fig. 1. For the undirected instance (G_t, G) there exists a sandwich with an Eulerian cycle while for the directed instance (D_t, D) admits no sandwich with an Eulerian circuit.

of various well-studied graph-theoretic problems. For instance, the case $E_t = E$ corresponds to the recognition problem while the case $E = V^2$ is the fill-in problem. In practice, sandwich problems arise in diverse areas such as biology [9,7], communication [13] and algebra [16]. Results concerning graph properties can also be found in [8,2,17,4,11].

A *complementary property* $\overline{\Pi}$ can be defined as follows: for any graph G, G satisfies $\overline{\Pi}$ iff the complement graph $\overline{G}$ satisfies Π. The following was proved in [8]:

Proposition 1.1 (*Golumbic et al. [8]*). *There exists a sandwich graph G_s satisfying property Π for the instance (G_t, G) iff there exists a sandwich graph H_s satisfying property $\overline{\Pi}$ for the instance $(\overline{G}, \overline{G_t})$.*

It follows that the Π sandwich problem is polynomially equivalent to the $\overline{\Pi}$ sandwich problem. Being a comparability graph is a complementary property. In [8], many results concern sub-families of comparability or co-comparability graphs. They are of special interest since a comparability graph admits one or more transitive orientations (each orientation defines a partially ordered set or poset). Let us defined a *directed property* $\overrightarrow{\Pi}$ as a property such that if a digraph $\overrightarrow{G}$ satisfies $\overrightarrow{\Pi}$, then the non-oriented graph G satisfies Π. For example, the directed Eulerian property fits this definition. Similarly, being a poset for a digraph corresponds, in the non-oriented version, to being a comparability graph.

Proposition 1.2. *Let D_t and D be arbitrary orientations of the graphs G_t and G. If there exists a sandwich digraph D_s satisfying the directed property $\overrightarrow{\Pi}$ for the instance (D_t, D), then there exists a sandwich graph G_s satisfying property Π for the instance (G_t, G).*

Since fixing the orientation of the edges of G_t and G restricts the number of possible sandwiches, it is possible that while a sandwich graph exists in the non-directed instance, no sandwich digraph exists in the directed instance. Fig. 1 shows such an example.

A property Π^1 is stronger than a property Π^2 iff any graph (or poset) that satisfies Π^2 also satisfies Π^1. Clearly a directed property $\overrightarrow{\Pi}$ is a weaker property than Π. Since, as observed in [8], there is no simple relationship between the complexities of the Π^1 and Π^2 sandwich problems, neither is there between the complexities of the $\overrightarrow{\Pi}$ and Π sandwich problems a priori.

A property $\overrightarrow{\Pi}$ defined on posets is a *comparability invariant* iff when a poset P satisfies $\overrightarrow{\Pi}$ then any transitive orientation of the comparability graph G of P also satisfies $\overrightarrow{\Pi}$. For example, being a poset of dimension k is a comparability invariant [10]. Comparability invariants are directed properties. Even if we restrict to comparability invariants, the phenomenon depicted in Fig. 1 can still occur. To check $\overrightarrow{\Pi}$, we only have to test one transitive orientation of the comparability graph. However, there is still no simple complexity relation (i.e., with respect to the Karp reduction, $\leqslant_K$). There may be no transitive orientation of a given non-oriented sandwich with the expected property (see Fig. 2). Therefore, one must check all possible non-oriented sandwiches, and there can be an exponential number of these.

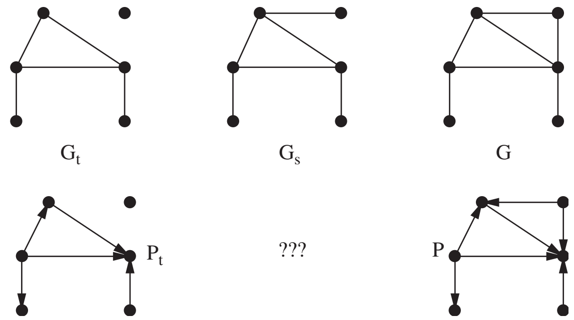

Fig. 2. In the above figure, the expected property for the sandwich is to have the degree sequence (3, 3, 3, 1, 1, 1). Notice that the degree sequence is a comparability invariant. Only one solution is possible in the non-oriented instance while there is no poset sandwich with that degree sequence.

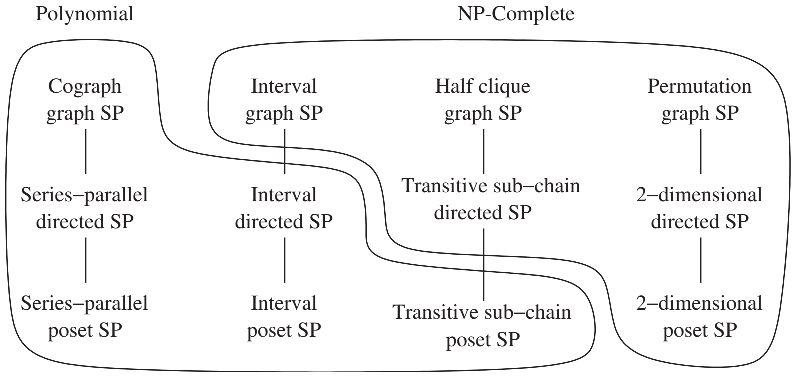

Fig. 3. A map of the results.

In the paper, we will focus on even stronger property than comparability invariant, since for those properties, if there exists a non-oriented sandwich satisfying Π, then there exists a directed sandwich satisfying $\overrightarrow{\Pi}$. We wonder whether for such properties adding orientations to the edges of G_t and G helps to solve the corresponding sandwich digraph problem. For example, the comparability graph sandwich problem has been proved to be NP-complete [8]. But testing the existence of a transitive sandwich digraph D_s for a pair of digraphs (D_t, D) can be completed in polynomial time. The algorithm just has to test whether the transitive closure of D_t is included in D which is a polynomial problem. We will focus on properties related to comparability graphs or co-comparability graphs (which are equivalent by Proposition 1.1) that can be naturally translated in terms of poset. Two slightly different problems are distinguished by the transitivity of the input digraphs.

Problem 2. DIRECTED SANDWICH PROBLEM FOR POSET PROPERTY $\overrightarrow{\Pi}$.
Instance: Two digraphs D_t and D such that $D_t \subseteq D$.
Question: Does there exist a sandwich digraph D_s for the pair (D_t, D) satisfying $\overrightarrow{\Pi}$?

Problem 3. POSET SANDWICH PROBLEM FOR POSET PROPERTY $\overrightarrow{\Pi}$.
Instance: Two posets $P_t = (V, E_t)$ and $P = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist a sandwich poset $P_s = (V, E_s)$ for the pair (P_t, P) satisfying $\overrightarrow{\Pi}$?

In the *directed sandwich problem*, digraphs D_t and D of the sandwich instance have an arbitrary orientation. If the required property $\overrightarrow{\Pi}$ deals with posets, we can assume that D_t is transitively oriented as its transitive closure is contained in any transitively oriented sandwich graph. Therefore, the instance will be written (P_t, D). Since the poset sandwich problem is a sub-problem of the directed sandwich problem, it follows that:

$$\text{Poset sandwich problem for } \overrightarrow{\Pi} \leqslant_K \text{ Directed sandwich problem for } \overrightarrow{\Pi}. \tag{1}$$

This inequality is helpful in complexity proofs: to show that a directed sandwich problem is NP-complete, we only need to show the NP-completeness of the poset sandwich problem, and conversely if the problem is polynomial (Fig. 3).

In this paper, we study poset properties inspired by the original results of [8]. We are interested in finding sandwiches that are series–parallel posets, interval posets or two-dimensional posets. These families correspond to co-graphs, interval graphs and permutation graphs, respectively (all of which are co-comparability graphs). Moreover, these three properties are comparability invariants. We prove the following results:

- the series–parallel poset sandwich problem is polynomial like the co-graph sandwich problem;
- the interval poset sandwich problem is polynomial while the interval graph sandwich problem in NP-complete;
- the two-dimensional directed sandwich problem is NP-complete like the permutation graph sandwich problem.

We also introduce a new problem, namely the *transitive sub-chain problem* that asks for the existence in a digraph of a transitive sub-chain of length at least $\lfloor |V|/2 \rfloor$. Reducing 3-SAT to this problem, we prove its NP-completeness. It follows that the associated directed sandwich problem is NP-complete, while the poset sandwich problem remains polynomial. Notice that in terms of non-oriented graphs, the transitive sub-chain problem corresponds to the "half-clique" problem (does there exist a clique containing at least one half of the vertices?). Clearly, the half-clique sandwich problem is NP-complete. Therefore, we show that any possible configuration with respect to (1) exists.

Notation: Let S_P be the set of sources of a poset P, $N_P(x)$ be the neighborhood of vertex x in P, $P_{|A}$ be the poset induced by P on a vertex set A. The notation $\mathrm{Succ}^P(x)$, denoting the set of out-vertices (successors) of x in P, is extended to sets: $\mathrm{Succ}^P(A) = \bigcup_{a \in A} \mathrm{Succ}^P(a)$.

2. Series–parallel posets

In certain scheduling problems, tasks are subject to a partial order. Although scheduling problems for an arbitrary partial order are NP-complete, they have efficient algorithms if the partial order is series–parallel [14]; these algorithms use a "divide-and-conquer" approach with the recursive structure of these posets. There is a linear-time algorithm to recognize a series–parallel poset due to Valdes et al. [19].

A series–parallel poset is obtained from the single-vertex poset by the application of two composition rules. The *parallel composition* of posets P_1 and P_2 is the poset $P_1 + P_2 = (V_1 \cup V_2, <_+)$ such that $u <_+ v$ if and only if $u, v \in V_1$ and $u <_1 v$ or $u, v \in V_2$ and $u <_2 v$. The *series composition* of posets P_1 and P_2 is the poset $P_1 * P_2 = (V_1 \cup V_2, <_*)$ such that $u <_* v$ if and only if $u, v \in V_1$ and $u <_1 v$ or $u, v \in V_2$ and $u <_2 v$ or $u \in V_1$ and $v \in V_2$. Therefore, series–parallel posets are organized in a tree structure (Fig. 4).

2.1. Series–parallel poset sandwich problem

The family of comparability graphs of the series–parallel posets is exactly the family of co-graphs [18], for which the sandwich problem has been proved to be polynomial [8]. The co-graph sandwich algorithm can be modified so that it applies to the series–parallel poset sandwich problem. By adding an argument about transitivity to its proof, we can prove that the poset sandwich problem is also polynomial like the co-graph sandwich problem is.

Problem 4. SERIES–PARALLEL POSET SANDWICH PROBLEM.
Instance: Two posets $P_t = (V, E_t)$ and $P = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist a series–parallel sandwich poset P_s for the pair (P_t, P)?

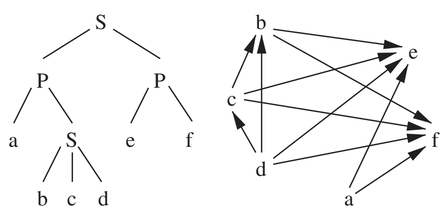

Fig. 4. A series–parallel poset and its canonical composition tree.

Let us briefly describe the principle of this algorithm. Proofs of Lemmas 2.1 and 2.2 are omitted since they can easily be deduced from [8]. Let us denote by $\overline{P}$ the incomparability graph of the poset P.

Lemma 2.1. *If both P_t and $\overline{P}$ are connected, then there is no series–parallel sandwich poset for the instance (P_t, P).*

Lemma 2.2. *Let $\{C_1, \ldots, C_k\}$ be the connected components of P_t. If $P_s^1, \ldots, P_s^k$ are, respectively, series–parallel sandwich posets for the instances $(P_t|_{C_1}, P|_{C_1}) \ldots (P_t|_{C_k}, P|_{C_k})$, then the parallel composition of $P_s^1, \ldots, P_s^k$ is a series–parallel sandwich poset for (P_t, P).*

Lemma 2.3. *Let $\{\overline{C}_1, \ldots, \overline{C}_k\}$ be the connected components of $\overline{P}$. If $P_s^1, \ldots, P_s^k$ are, respectively, series–parallel sandwich posets for the instances $(P_t|_{\overline{C}_1}, P|_{\overline{C}_1}) \ldots (P_t|_{\overline{C}_k}, P|_{\overline{C}_k})$, then the series composition of $P_s^1, \ldots, P_s^k$ is a series–parallel sandwich poset for (P_t, P).*

A similar lemma can be found in [8] for the non-oriented case of co-graphs. The only difference is that we have to confirm that transitivity can be ensured between the connected components $\{\overline{C}_1, \ldots, \overline{C}_k\}$.

Proof. Let x_1, x_2, x_3 be three vertices of different connected components $\overline{C}_1, \overline{C}_2, \overline{C}_3$ of $\overline{P}$. By definition of the connected component, the three possible arcs between these three vertices belong to E. Since by assumption P is a poset, these three edges are transitively oriented.

Let us now prove that the arcs between two connected components $\overline{C}, \overline{C}'$ are oriented in the same direction. Let $N^+(x)$ and $N^-(x)$ be, respectively, the out-neighborhood and in-neighborhood of x in $\overline{C}'$. Since P is transitive all possible arcs exist from $N^+(x)$ towards $N^-(x)$, which contradicts the fact that $\overline{C}'$ is connected in $\overline{P}$. □

Now it is straightforward to see the algorithm. If both P_t and $\overline{P}$ are connected, then there is no series–parallel sandwich poset. Otherwise if P_t is not connected, recurse on the instances induced by its connected components $C_1, \ldots, C_k$; if $\overline{P}$ is not connected, recurse on the instances induced by the connected components $\overline{C}_1, \ldots, \overline{C}_k$ of $\overline{P}$. This algorithm is exactly the polynomial time algorithm given in [8].

Theorem 2.1. *The series–parallel poset sandwich problem is polynomial.*

2.2. Series–parallel directed sandwich problem

However, the above method does not provide an algorithm for the directed sandwich problem. For this more general problem, we provide a polynomial-time algorithm based on an elimination ordering principle.

Problem 5. SERIES–PARALLEL DIRECTED SANDWICH PROBLEM.
Instance: A poset $P_t = (V, E_t)$ and a digraph $D = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist a series–parallel sandwich poset P_s for the pair (P_t, D)?

Let us first introduce a useful proposition. A property π is an *hereditary property* iff for any induced sub-graph (or induced sub-poset), the sub-graph satisfies the property π. Being a tree or a series–parallel poset is an hereditary property.

Proposition 2.2. *Let π be an hereditary property. If there exists a π-*sandwich *G_s for the instance (G_t, G) on vertex set V, then for any subset $V' \subset V$, the induced sub-sandwich $G_s|_{V'}$ is a π-sandwich for the induced instance $(G_t|_{V'}, G|_{V'})$.*

Proof. First of all, $G_s|_{V'}$ is a sandwich for $(G_t|_{V'}, G|_{V'})$. Since π is hereditary $G_s|_{V'}$ also verifies property π. □

Lemma 2.4. *If there is a series–parallel sandwich poset P_s for the instance (P_t, D) where P_t is connected, then there exists a non-empty set of vertices $\mathscr{A} \subsetneq V$ such that*

$$\forall x \in \mathscr{A}, \quad V \backslash \mathscr{A} \subseteq \mathrm{Succ}^D(x) \quad \textit{and} \quad \mathrm{Succ}^{P_t}(V \backslash \mathscr{A}) \cap \mathscr{A} = \emptyset.$$

SP-Alg(V, E_t, E): outputs a set E_s of edges
If V is a singleton vertex set **Then**
  Return $E_s = \emptyset$
For each connected component C of P_t **Do**
  Look for a set $\mathcal{A} \subsetneq V$ such that $\forall x \in \mathcal{A}$, $C \setminus \mathcal{A} \subseteq Succ^{D|_C}(x)$ and $Succ^{P_s}(C \setminus \mathcal{A}) \cap \mathcal{A} = \emptyset$
  If no such $\mathcal{A}$ exists **Then** EXIT
  Else
    Let $E^s_{\mathcal{A}}$ =**SP-Alg**$(\mathcal{A}, E_t \cap \mathcal{A}^2, E \cap \mathcal{A}^2)$
    Let $E^s_{C \setminus \mathcal{A}}$ =**SP-Alg**$(\mathcal{A}, E_t \cap (C \setminus \mathcal{A})^2, E \cap (C \setminus \mathcal{A})^2)$
    $E_C = E^s_{\mathcal{A}} \cup E^s_{C \setminus \mathcal{A}} \cup \mathcal{A} \times (V \setminus \mathcal{A})$
End of for
Return $E_s = \bigcup_C E_C$

Fig. 5. A polynomial time algorithm for the series–parallel directed sandwich problem.

Proof. Suppose there exists P_s, a series–parallel sandwich poset for the instance (P_t, D). Observe that such a set $\mathscr{A}$ must contain the source S_{P_s} of P_s. Since P_t is connected, P_s is also connected. Thus, it is the result of a series composition and its co-comparability graph $\overline{G}_s$ is not connected. Notice that S_{P_s} is contained in a connected component of $\overline{G}_s$ (since S_{P_s} is an antichain of P_s). Let $\mathscr{A}$ be that component. $\mathscr{A}$ is in series composition with $V \setminus \mathscr{A}$. Since $\mathscr{A}$ contains the sources of P_s, the arcs between $\mathscr{A}$ and $V \setminus \mathscr{A}$ are oriented towards $V \setminus \mathscr{A}$. It follows that $V \setminus \mathscr{A} \subseteq \mathrm{Succ}^D(x)$, $\forall x \in \mathscr{A}$.

Moreover, since P_s is a poset, there is no arc from $V \setminus \mathscr{A}$ towards $\mathscr{A}$. The inclusion $E_t \subseteq E_s$ implies that $\mathrm{Succ}^{P_t}(V \setminus \mathscr{A}) \cap \mathscr{A} = \emptyset$. $\square$

Lemma 2.5. *Let $\{C_1, \ldots, C_k\}$ be the set of connected components of P_t. The following statements are equivalent*:

(1) *There is a series–parallel sandwich poset for the instance (P_t, D).*
(2) *For any i, $1 \leqslant i \leqslant k$, there exist*:
  - *a non-empty set $\mathscr{A}_i$ such that $\forall x \in \mathscr{A}_i$, $C_i \setminus \mathscr{A}_i \subseteq \mathrm{Succ}^{D|C_i}(x)$ and $\mathrm{Succ}^{P_t}(C_i \setminus \mathscr{A}_i) \cap \mathscr{A}_i = \emptyset$;*
  - *a series–parallel sandwich poset $P^s_{\mathscr{A}_i}$ for the instance $(P_t|_{\mathscr{A}_i}, D|_{\mathscr{A}_i})$;*
  - *a series–parallel sandwich poset $P^s_{C_i \setminus \mathscr{A}_i}$ for the instance $(P_t|_{C_i \setminus \mathscr{A}_i}, D|_{C_i \setminus \mathscr{A}_i})$.*

Proof. Suppose P_s is a series–parallel sandwich poset for the instance (P_t, D). By Lemma 2.4, whenever $1 \leqslant i \leqslant k$, there exists an non-empty set $\mathscr{A}_i$. By definition, $\mathscr{A}_i$ can be composed in series with $C_i \setminus \mathscr{A}_i$ in both $D|_{C_i}$ and $P_s|_{C_i}$. Since being a series–parallel poset is an hereditary property, any induced subset of P_s is also a series–parallel poset. If follows that $P_s|_{C_i \setminus \mathscr{A}_i}$ and $P_s|_{\mathscr{A}_i}$ are series–parallel sandwiches for the instances $(P_t|_{C_i \setminus \mathscr{A}_i}, D|_{C_i \setminus \mathscr{A}_i})$ and $(P_t|_{\mathscr{A}_i}, D|_{\mathscr{A}_i})$, respectively.

Assume the converse. Since for any i, $1 \leqslant i \leqslant k$, $C_i \setminus \mathscr{A}_i \subseteq \mathrm{Succ}^{D|C_i}(x)$ and $\mathrm{Succ}^{P_t}(C_i \setminus \mathscr{A}_i) \cap \mathscr{A}_i = \emptyset$, $P^s_{C_i \setminus \mathscr{A}_i}$ and $P^s_{\mathscr{A}_i}$ can be composed in series to form a series–parallel poset $P^s_{C_i}$ for the instance $(P_t|_{C_i}, D_{C_i})$. It follows that a series–parallel sandwich poset for the whole instance (P_t, D) is obtained by the parallel composition of the $P^s_{C_i}$'s. $\square$

Assuming that we are able to compute a set $\mathscr{A}$ as described in Lemma 2.4, we are now able to design algorithm based on Lemma 2.5 for solving the series–parallel directed sandwich problem.

When solving the problem recursively in Fig. 5, Proposition 2.2 guarantees that we can find a set $\mathscr{A} \subsetneq V$ such that

$$\forall x \in \mathscr{A}, \quad V \setminus \mathscr{A} \subseteq \mathrm{Succ}^D(x) \quad \text{and} \quad \mathrm{Succ}^{P_t}(V \setminus \mathscr{A}) \cap \mathscr{A} = \emptyset. \tag{2}$$

To describe how to compute such a set, we shall use three simple facts.

Fact 2.6. *Let x and y be distinct vertices. If $x \in \mathscr{A}$ and $y \notin \mathrm{Succ}^D(x)$, then $y \in \mathscr{A}$.*

Proof. This directly follows from the definition of $\mathscr{A}$. $\square$

Fact 2.7. *Let x and y be distinct vertices. If* $x \in \mathscr{A}$ *and* $x \in \mathrm{Succ}^{P_t}(y)$, *then* $y \in \mathscr{A}$.

Proof. If $x \in \mathscr{A}$ and $y \notin \mathscr{A}$, then both arcs xy and yx would belong to E_s: yx because $x \in \mathrm{Succ}^{P_t}(y)$ and xy because $\mathscr{A}$ is composed in series with $V \backslash \mathscr{A}$ in P_s. This is a contradiction, since no pair of symmetric arcs can belong to a poset. $\square$

Fact 2.8. $\mathscr{A} \cap S_{P_t} \neq \emptyset$.

Proof. If $A \cap S_{P_t} = \emptyset$, then $\forall x \in \mathscr{A}$, $x \notin S_{P_t}$ and there exists $y \in V \backslash \mathscr{A}$ such that $yx \in P_t$ (as there are no cycle in P_t), this contradicts the definition of $\mathscr{A}$. $\square$

Assume for some source $x \in S_{P_t}$, the following algorithm returns a strict subset $\mathscr{A}$ of V. If $\mathscr{A}$ does not respect (2), then either there exists $y \in V \backslash A$ such that $y \notin \mathrm{Succ}^D(x)$ and it contradicts Fact 2.6, or there exists $y \in V \backslash A$ such that for an $x \in A$, $x \in \mathrm{Succ}^{P_t}(y)$ and it contradicts Fact 2.7. Then we just have to test Facts 2.6 and 2.7 to prove that $\mathscr{A}$ satisfies (1). It can therefore be used at line 4 of the algorithm depicted in Fig. 5 to recurse.

SP-Rec($x \in S_{P_t}$, V, P_t, D): outputs a subset $\mathscr{A}$ of vertices
1. $\mathscr{A} = \{x\}$
2. **While** there exists $y \notin \mathscr{A}$ and $z \in \mathscr{A}$ such that either
   $yz \in E_t$ or $zy \notin E$ Add y to $\mathscr{A}$
3. **Return** $\mathscr{A}$

Theorem 2.3. *The series–parallel directed sandwich problem is polynomial.*

Proof. The validity of the algoritm described in Fig. 5 follows from Lemmas 2.4, 2.5, Proposition 2.2 and Facts 2.6–2.8. The number of recursive calls in 5 is at most $n = |V|$. A brute force analysis shows that testing the existence of a set $\mathscr{A}$ satisfying (2) requires $\mathcal{O}(n^3)$ per source of P_t. It follows that testing the existence of a series–parallel sandwich is polynomial. $\square$

3. Interval posets

A poset P is an interval poset iff a real interval $I_v = [a_v, b_v]$ can be assigned to each element v in P, such that $v \leqslant w$ if and only if $b_v \leqslant a_w$.

Problem 6. INTERVAL DIRECTED SANDWICH PROBLEM.
Instance: A poset $P_t = (V, E_t)$ and a digraph $D = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist an interval sandwich poset P_s for the pair (P_t, D)?

We prove the interval directed sandwich problem is polynomial, in contrast to the undirected case: the interval graph sandwich problem is NP-complete [8]. This complexity gap is probably due to the fact that an interval graph is a co-comparability graph that can have many transitive orientations. Thus, fixing an arbitrary transitive orientation of the interval graph drastically simplifies the problem. The same phenomenon occurs for the recognition problem: although both interval graph and interval poset recognition problems have linear time complexity, the interval graph recognition is much harder [1,3,12]. The interval poset recognition algorithm [15] is based on the following characterization by Fishburn [5].

Theorem 3.1 (*Fishburn [5]*). *A poset P is an interval poset if and only if the set of successors* $\{\mathrm{Succ}^P(v) = \{u \in V, v \leqslant u\}\}_{v \in V}$ *is linearly ordered by inclusion.*

From the above characterization, we can deduce that a non-connected interval poset contains at most one connected component of more than one vertex. In other words, a non-connected interval poset consists in a connected interval poset plus some isolated vertices. This yields to another well-known characterization of interval posets in term of forbidden sub-poset: a poset is an interval poset iff it does not contain a $2 + 2$ as induced poset (see Fig. 6).

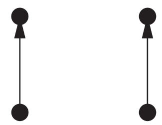

Fig. 6. The 2+2 poset.

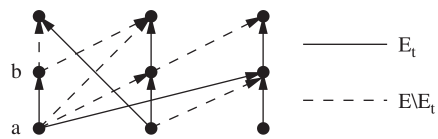

Fig. 7. In this instance $\mathscr{A} = \{a\}$. To simplify the drawing, we omit the transitive arcs of E_t. However, E is not necessarily transitive.

Lemma 3.1 is quite similar to Lemma 2.4, except that a smaller part of the vertices is this time successor of the set $\mathscr{A}$. This result is the basis of our algorithm and can be seen as a generalization of the recognition algorithm of [15].

Lemma 3.1. *If there is an interval sandwich poset P_s for the instance (P_t, D), then there exists a set $\mathscr{A} \subseteq S_{P_t}$ such that $V \backslash S_{P_t} \subseteq \mathrm{Succ}^D(x)$, $\forall x \in \mathscr{A}$.*

Proof. Let P_s be an interval sandwich poset for the instance (P_t, D). From Theorem 3.1 P_s has its sets of successors linearly ordered by inclusion. Therefore, there exists a set $\mathscr{A}$ of P_s sources such that every non-source vertex of P_s is a successor of any vertex $x \in \mathscr{A}$: $\mathrm{Succ}^{P_s}(x) = V \backslash S_{P_s}$. Since any source of P_s is a source of P_t, $\mathscr{A} \subseteq S_{P_t}$. And since $\forall x \in \mathscr{A}$, $\mathrm{Succ}^{P_s}(x) \subseteq \mathrm{Succ}^D(x)$, it follows that $V \backslash S_{P_t} \subseteq \mathrm{Succ}^D(x)$, $\forall x \in \mathscr{A}$. $\square$

Let us define the following set of vertices:

$$\mathscr{A} = \{x \in V | \mathrm{Succ}^D(x) \supseteq V \backslash S_{P_t}\}.$$

An example of set $\mathscr{A}$ in a directed sandwich instance is given in Fig. 7.

Lemma 3.2. *The following statements are equivalent*:

(1) *There is an interval sandwich poset for the instance (P_t, D).*
(2) *There exists a non-empty set $\mathscr{A}$ and an interval sandwich poset for the instance $(P_t, D)|_{V \backslash \mathscr{A}}$.*

Proof. First assume P_s is an interval sandwich poset for the instance (P_t, D). From Lemma 3.1, $\mathscr{A}$ is not empty. Since being an interval poset is an hereditary property, the poset $P_s|_{V \backslash \mathscr{A}}$ is an interval poset. Moreover, any arc of $P_t|_{V \backslash \mathscr{A}}$ is an arc of $P_s|_{V \backslash \mathscr{A}}$. It follows that $P_s|_{V \backslash \mathscr{A}}$ is an interval sandwich poset for $(P_t, D)|_{V \backslash \mathscr{A}}$.

Conversely, assume $\tilde{P}_s$ is an interval sandwich poset for the instance $(P_t, D)|_{V \backslash \mathscr{A}}$ and $\mathscr{A} \neq \emptyset$. By Theorem 3.1, the vertices of $V \backslash \mathscr{A}$ are linearly ordered by inclusion of successors. Among the set $S_{\tilde{P}_s}$ of sources of $\tilde{P}_s$, we can distinguish $S_{\tilde{P}_s} \backslash S_{P_t}$ and $S_{\tilde{P}_s} \cap S_{P_t}$. Notice that any arc xy, with $x \in \mathscr{A}$ and $y \in \backslash S_{P_t}$ belongs to E (remark that $(V \backslash \mathscr{A}) \backslash S_{P_t} = V \backslash S_{P_t}$ since $\mathscr{A} \subseteq S_{P_t}$). Therefore, adding those arcs to $\tilde{E}_s$ defines a sandwich $P_s = (V, E_s)$. Since for any $x \in \mathscr{A}$, we have $\mathrm{Succ}^{P_s}(x) = V \backslash S_{P_t}$, for any $y \in V \backslash \mathscr{A}$, we have $\mathrm{Succ}^{P_s}(y) \subset \mathrm{Succ}^{P_s}(x)$. Since the set of successors in P_s is linearly ordered by inclusion, P_s is an interval order. $\square$

Corollary 3.2. *If there exists an interval poset sandwich for the instance (P_t, D), then there exists one, say P_s, such that $S_{P_s} = S_{P_t}$.*

Proof. The proof follows from the construction described in the proof of Lemma 3.2. Notice that when constructing P_s from $\tilde{P}_s$, we never add arcs towards sources of P_t. Applying this argument recursively completes the proof. $\square$

```
Interval-Alg(V, E_t, E): outputs a set E_s of edges
    If E_t = ∅ Then Return E_s = ∅
    Else
        Look for a set 𝒜 among P_t sources such that ∀x ∈ 𝒜,
        V \ S_{P_t} ⊆ Succ^D(x)
        If no such 𝒜 exists Then STOP the algorithm
        Else
            Let Ẽ_t and Ẽ be the edge sets of P_t|_{V\𝒜} and D|_{V\𝒜}
            E_s = 𝒜 × (V \ S_{P_t}) ∪ Interval-Alg(Ṽ, Ẽ_t, Ẽ)
        Return E_s
    End of if
```

Fig. 8. A polynomial time algorithm for the interval directed sandwich problem.

Theorem 3.3. *The interval directed sandwich problem is polynomial.*

Proof. The correctness of the algorithm depicted in Fig. 8 is implied by Lemmas 3.1 and 3.2. Testing for the existence of the set $\mathscr{A}$ can clearly be done in $\mathcal{O}(n^2)$ where $n = |V|$. Since there are at most n recursive calls, the whole complexity is polynomial. $\square$

Corollary 3.4. *The interval poset sandwich problem is polynomial.*

4. The transitive sub-chain problem

In a directed graph $D = (V, E)$, a *transitive sub-chain* is a chain $[v_1, \ldots, v_k]$ such that any arcs v_i, v_j (with $1 \leqslant i < j \leqslant k$) belongs to E. This section introduces the transitive sub-chain problem that asks for the existence of a transitive sub-chain containing at least half of the vertices. Using a reduction from 3-SAT inspired from [6], we first prove that this problem is NP-complete. Then we show that the corresponding poset sandwich problem is polynomial while the directed sandwich problem is NP-complete.

Problem 7. TRANSITIVE SUB-CHAIN PROBLEM.
Instance: A digraph $D = (V, E)$.
Question: Does D contains a transitive sub-chain of length at least $\lfloor |V|/2 \rfloor$?

Theorem 4.1. *The transitive sub-chain problem is NP-complete.*

Proof. We reduce 3-SAT to the transitive sub-chain problem. Let $\mathscr{I}$ be an instance of k clauses $c_i = (x_i^1 \vee x_i^2 \vee x_i^3)$ $(1 \leqslant i \leqslant k)$. We transform $\mathscr{I}$ into a digraph $D = (V, E)$ with $V = \{c_0\} \bigcup_{1 \leqslant i \leqslant k} \{c_i, x_i^1, x_i^2, x_i^3\}$. The three literals of a clause c_i are independent. For any $1 \leqslant i \leqslant k$ and any $1 \leqslant j \leqslant 3$, $x_i^j c_i$ is an arc of E. Moreover for $1 \leqslant i < j \leqslant k$ and $1 \leqslant h, h' \leqslant 3$, $x_i^h x_j^{h'}$ belongs to E iff $x_i^h \neq \overline{x}_j^{h'}$. Finally for any $v \neq c_0$, we add the arcs $c_0 v$. Fig. 9 gives an example. Clearly D is polynomial time constructible.

We now prove that a 3-SAT instance $\mathscr{I}$ is satisfiable iff the associated digraph contains a transitive chain of length equal to $\lfloor |V|/2 \rfloor$.

Assume the instance $\mathscr{I}$ is satisfiable. Then at least one literal, denoted by l_i, per clause has been satisfied and for any $i \neq j$, $l_i \neq \overline{l_j}$. Then there exists a chain in D from c_0 to c_k through the l_i's such that the subgraph induced by the vertices of the chain is transitive. Such a chain contains $2k + 1$ vertices; therefore, its length is $2k$, which is equal to $\lfloor |V|/2 \rfloor$ since $|V| = 4k + 1$.

Conversely, if a transitive chain of length $\lfloor |V|/2 \rfloor$ exists in D, then it has to contain any c_i $(0 \leqslant i \leqslant k)$ plus one literal l_i per clause. The transitivity ensures that for no $i \neq j$, we have $l_i = \overline{l_j}$. It therefore defines an assignment that satisfies the instance. $\square$

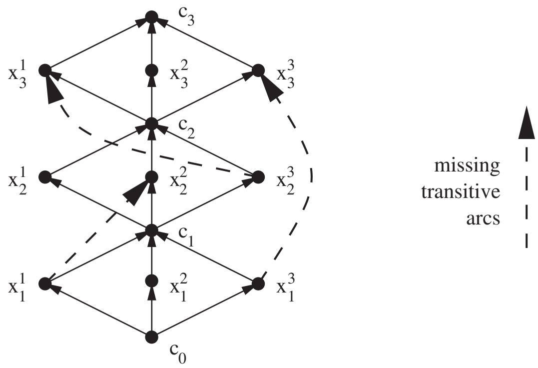

Fig. 9. The digraph associated to the 3-SAT instance $(x_1^1 \vee x_1^2 \vee x_1^3) \wedge (x_2^1 \vee x_2^2 \vee x_2^3) \wedge (x_3^1 \vee x_3^2 \vee x_3^3)$ where $x_1^1 = \overline{x}_2^2$, $x_1^3 = \overline{x}_3^3$ and $x_2^3 = \overline{x}_3^1$. The transitive arcs are omitted and dotted arcs do not belong to the digraphs.

4.1. Transitive sub-chain poset sandwich problem

In a poset any chain is transitive. The height $h(P)$ of a poset P is the length of the longest chain of P. It turns out the transitive sub-chain problem for poset is polynomially equivalent to the computation of the height, which is polynomial (it can be done using a DFS).

Problem 8. TRANSITIVE SUB-CHAIN POSET SANDWICH PROBLEM.
Instance: A poset $P_t = (V, E_t)$ and a poset $P = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist a poset $P_s = (V, E_s)$ such that $E_t \subseteq E_s \subseteq E$ and $h(P_s) \geqslant \lfloor |V|/2 \rfloor$?

It is well known that the height of a poset is a monotone increasing function of the set of edges. The following lemma states that property.

Lemma 4.1. *Let $Q = (V, E)$ and $Q' = (V, E')$ be two posets such that $E \subseteq E'$. Then $h(Q) \leqslant h(Q')$.*

Theorem 4.2. *The transitive sub-chain poset sandwich problem is polynomial.*

Proof. Lemma 4.1 shows that computing $h(P)$ suffices to test the existence of a sandwich poset P_s such that $h(P_s) \geqslant \lfloor |V|/2 \rfloor$. $\square$

4.2. Transitive sub-chain directed sandwich problem

Problem 9. TRANSITIVE SUB-CHAIN DIRECTED SANDWICH PROBLEM.
Instance: A poset $P_t = (V, E_t)$ and a digraph $D = (V, E)$ such that $E_t \subseteq E$.
Question: Does there exist a poset $P_s = (V, E_s)$ such that $E_t \subseteq E_s \subseteq E$ and $height(P_s) \geqslant \lfloor n/2 \rfloor$?

As for the transitive sub-chain problem, 3-SAT reduces to the above problem. Setting $P_t = (V, \emptyset)$ and D as defined in Theorem 4.1 is enough to prove its NP-completeness.

Theorem 4.3. *The transitive sub-chain directed sandwich problem is NP-complete.*

5. Two-dimensional posets

Let L and P be, respectively, a total order and a poset on the same vertex set. If $x <_P y$ implies $x <_L y$, then L is a linear extension of P. The dimension of a poset P is the minimum number k of linear extensions such that $x <_P y$ if and only if $x <_{L_i} y$ for any i, $1 \leqslant i \leqslant k$. The comparability graphs of two-dimensional posets are the permutation graphs.

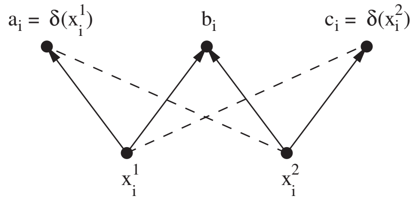

Fig. 10. The gadget associated to a triple $T_i = (a_i, b_i, c_i)$. We denote $a_i = \delta(x_i^2)$ and $c_i = \delta(x_i^1)$.

The permutation graph sandwich problem is known to be NP-complete [8]. Unlike the interval case where the directed problem is polynomial while the non-oriented problem is NP-complete, this section shows that for two-dimensional posets, the poset sandwich problem is also NP-complete. In this case, fixing the orientation of the edges does not help.

Problem 10. TWO-DIMENSIONAL POSET SANDWICH PROBLEM.
Instance: $P_t = (V, E_t)$ and $P = (V, E)$ two posets on the same ground vertex set such that $E_t \subseteq E$.
Question: Does there exist a sandwich poset $P_s = (V, E_s)$ of dimension 2?

Theorem 5.1. *The two-dimensional poset sandwich problem is NP-complete.*

Proof. We reduce to the same BETWEENNESS problem that was used in [8] for permutation sandwich graphs. Since we are dealing with posets, we need some additional arguments.

Problem 11. BETWEENNESS PROBLEM.
Instance: A ground set S and a set $\mathscr{T} = \{T_1, \dots, T_k\}$ of triples of S.
Question: Does there exist a linear ordering λ such that for any triple $T_i = (a_i, b_i, c_i)$, either $a_i <_\lambda b_i <_\lambda c_i$ or $c_i <_\lambda b_i <_\lambda a_i$?

Let $\mathscr{T} = \{T_1, \dots, T_k\}$ a set of triples on ground set S be an instance of BETWEENNESS. We associate a 5-vertex gadget to any triple $T_i = (a_i, b_i, c_i)$ (see Fig. 10): $\{a_i, b_i, c_i, x_i^1, x_i^2\}$ where $a_i = \delta(x_i^2)$ and $c_i = \delta(x_i^1)$. To any instance of BETWEENNESS we associate a pair of posets $P_t = (V, E_t)$ and $P = (V, E)$ with $E_t \subseteq E$, based on the gadget of Fig. 10 as follows:

$$V = S \cup X \quad \text{where } X = \{x_i^1 | 1 \leqslant i \leqslant k\} \cup \{x_i^2 | 1 \leqslant i \leqslant k\},$$

$$E_t = \bigcup_{1 \leqslant i \leqslant k} \{x_i^1 a_i, x_i^1 b_i, x_i^2 b_i, x_i^2 c_i\} \quad \text{and} \quad E = X \times S \backslash \{uv | v = \delta(u)\}.$$

Clearly, P_t and P are polynomial time constructible. First, suppose there is a two-dimensional sandwich poset P_s for (P_t, P). We write $u < v$ if uv is an arc of P_s, and $u \| v$ if u and v are incomparable. Let $L_s = (L_1, L_2)$ be a realizer of P_s. Then P_s is a sub-order of the planar lattice $L = L_1 \times L_2$ since product-dimension and intersection-dimension are equal. We can define an ordering λ on L by

$$u <_\lambda v \text{ if and only if } L_1(u) > L_1(v) \text{ and } L_2(u) < L_2(v).$$

Since S is an antichain in P_s, the restriction of λ to S is a linear ordering (as any two elements i, j of S satisfy by definition either $L_1(j) > L_1(i)$ and $L_2(j) < L_2(i)$, or $L_1(i) > L_1(j)$ and $L_2(i) < L_2(j)$). Let us consider a triple $T_i = (a_i, b_i, c_i)$. Without loss of generality, we can assume that $a_i <_\lambda c_i$. Since $x_i^1 < a_i$ and $x_i^1 \| c_i$, then $x_i^1 <_\lambda c_i$. Indeed if we suppose that $x_i^1 \|_\lambda c_i$, this means that x_i^1 and c_i are comparable in $L_1 \cap L_2$, which is a contradiction. And if we suppose $x_i^1 >_\lambda c_i$, then by transitivity of λ we have $a_i <_\lambda x_i^1$, and since $c_i <_\lambda a_i$, by transitivity $c_i <_\lambda x_i^1$, which is also a contradiction. Since $x_i^1 < b_i$ and $b_i \| c_i$, it follows that $b_i <_\lambda c_i$. Similarly, considering x_i^2 we can prove that $a_i <_\lambda b_i$. Therefore, b_i is between a_i and c_i in λ. Thus, a solution to the two-dimensional poset sandwich problem (P_t, P) implies a solution to the BETWEENNESS problem on S with triples $\mathscr{T}$.

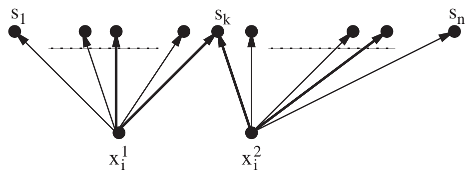

Fig. 11. Completion of P_t into a two-dimensional sandwich poset: $x_i^1 \in X_k^1$ implies that any arc xs_j with $j \leqslant k$ belongs to E_s; $x_i^2 \in X_k^2$ implies that any arc xs_j with $k \leqslant j$ belongs to E_s. The arcs drawn in bold belong to E_t.

For the converse, let λ be a linear ordering on S that solves the BETWEENNESS problem for the triples $\mathscr{T}$. By reversing some of the triples, we can assume that $a_i <_\lambda b_i <_\lambda c_i$ for every i. We shall define a two-dimensional poset P_s that is a sandwich for the pair (P_t, P).

Let $S = \{s_k \mid 1 \leqslant k \leqslant n\}$, where $s_k <_\lambda s_l$ if and only if $k < l$. For $2 \leqslant m \leqslant n-1$, let $X_m^1 = \{x_i^1 \mid b_i = s_m\}$ and define X_m^2 similarly. The poset P_s has the edges xs_k, where either $x \in X_m^1$ with $m \geqslant k$, or $x \in X_m^2$ with $m \leqslant k$ (Fig. 11).

Consider the following two listings of V.

$$L_1 : X_2^1, X_3^1, \ldots, X_{n-1}^1, s_1, X_2^2, s_2, X_3^2, s_3, \ldots, s_{n-2}, X_{n-1}^2, s_{n-1}, s_n,$$
$$L_2 : X_{n-1}^2, \ldots, X_3^2, X_2^2, s_n, X_{n-1}^1, s_{n-1}, \ldots, s_4, X_3^1, s_3, X_2^1, s_2, s_1.$$

Note that the listing used for each set X_m^j in L_1 is reversed in L_2.

If L_1 and L_2 are considered to be linear orders, then their intersection is P_s, proving that P_s is two-dimensional. □

References

[1] K.S. Booth, G.S. Lueker, Testing for the consecutive ones properties, interval graphs and graph planarity using PQ-tree algorithm, J. Comput. Systems Sci. 13 (1976) 335–379.
[2] M. Cerioli, H. Everett, C.H.M. de Figueiredo, S. Klein, The homogeneous set sandwich problem, Inform. Process. Lett. 67 (1) (1998) 31–35.
[3] D.G. Corneil, S. Olariu, L. Stewart, The ultimate interval graph recognition algorithm? in: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1998, pp. 175–180.
[4] C.M.H. de Figueiredo, S. Klein, K. Vušković, The graph sandwich problem for 1-join composition is NP-complete, Discrete Appl. Math. 121 (2002) 73–82.
[5] P.C. Fishburn, Intransitive indifference in preference theory: a survey, Oper. Res. 18 (1970) 207–228.
[6] H.N. Gabow, S.N. Maheshwari, L.J. Osterweil, On two problems in the generation of program test paths, IEEE Trans. Software Eng. 2 (3) (1976) 227–231.
[7] M.C. Golumbic, H. Kaplan, R. Shamir, On the complexity of DNA physical mapping, Adv. in Appl. Math. 15 (1994) 251–261.
[8] M.C. Golumbic, H. Kaplan, R. Shamir, Graph sandwich problems, J. Algorithms 19 (1995) 449–473.
[9] M.C. Golumbic, R. Shamir, Complexity and algorithms for reasoning about time: a graph-theoretic approach, J. ACM 40 (1993) 1108–1133.
[10] M. Habib, Comparability invariants, Ann. Discrete Math. 23 (1984) 371–386.
[11] M. Habib, E. Lebhar, C. Paul, A note on finding all homogeneous set sandwiches, Inform. Process. Lett. 87 (2003) 147–151.
[12] M. Habib, R. McConnell, C. Paul, L. Viennot, Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing, Theoret. Comput. Sci. 234 (2000) 59–84.
[13] P.B. Henderson, Y. Zalcstein, A graph-theoretic characterization of the pv chunk class of synchronizing primitives, SIAM J. Comput. 6 (1977) 88–108.
[14] E.L. Lawler, Sequencing jobs to minimize total weighted completion time subject to precedence constraints, Ann. Discrete Math. 2 (1978) 75–90.
[15] C.H. Papadimitriou, M. Yannakakis, Scheduling interval-ordered tasks, SIAM J. Comput. 8 (1979) 405–409.
[16] J.D. Rose, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, in: R.C. Reed (Ed.), Graph Theory and Computing, Academic Press, NY, 1972, pp. 183–217.
[17] S.-M. Tang, F.-L. Yeh, Y.-L. Wang, An efficient algorithm for solving the homogeneous set sandwich problem, Inform. Process. Lett. 77 (2001) 17–22.
[18] J. Valdes, Parsing flowcharts and series–parallel graphs, Technical Report STAN-CS-78-682, Computer science Department, Stanford University, Stanford, CA, 1978.
[19] J. Valdes, R.E. Tarjan, E.L. Lawler, The recognition of series parallel digraphs, SIAM J. Comput. 11 (2) (1982) 298–313.