Discrete Mathematics 213 (2000) 67–86
DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Performance guarantees and applications for Xia's algorithm ☆

Michelle A. Donalies *, Bernd S.W. Schröder [1]

Department of Mathematics, Hampton University, Hampton, VA 23668, USA

Abstract

Xia's algorithm consists of a reduction algorithm and a translation procedure both originally used to tackle the fixed point property for ordered sets. We present results that show that the translation procedure allows access to a much wider range of problems and results showing that the algorithm is very efficient when applied to the fixed point property in ordered sets or for order/graph isomorphism/rigidity. © 2000 Published by Elsevier Science B.V. All rights reserved.

MSC: 06A06; 05C85; 05C99; 68Q15; 68Q25

Keywords: Ordered set; Fixed point property; Algorithm; Isotone relation; Complexity; Graph; NP-completeness; Cutset property; Isomorphism problem

Xia's algorithm originally was conceived through the translation of the problem whether a given ordered set has the fixed point property into formal concept analysis (cf. [17]). It has now become clear (cf. [14]) that the algorithm is very efficient and that it has applications that reach far beyond the fixed point theory for ordered sets. In this paper we present Xia's algorithm as it would apply to the problem of finding an r-clique in a graph with a vertex r-coloring. We then show how many interesting decision problems can be efficiently translated into the problem of finding an r-clique in a graph with a vertex r-coloring. The performance guarantees proved in Section 3 — though limited to the realm of graph and order-homomorphisms — show the great potential of Xia's algorithm to be a versatile and effective tool in the treatment of many NP-complete problems.

☆ This work was sponsored by Office of Naval Research grant nr. N 00014-95-1-0660. Part of this work is the subject of M. Donalies' MS thesis [3]. Part of this work was presented at the ORDAL 96 conference in Ottawa, Canada, August 1996 and at the Banach Center Minisemester in Discrete Mathematics (week on ordered sets) in Warsaw, Poland, September 1996.

* Corresponding author.

[1] Current address is: Program of Mathematics and Statistics, Louisiana Tech University, Ruston, LA 71272, USA.

PII: S0012-365X(99)00326-X

1. The algorithm

We start by introducing the translation of Xia's algorithm (cf. [17]) to the problem of finding an r-clique in an r-vertex-colored graph. Since this problem is trivial for $r \leqslant 2$ we will assume throughout that $r \geqslant 3$. The underlying idea is to successively remove edges that cannot possibly be part of any r-clique (i.e., a 'branch and bound' approach). By Lemma 1.5, part 3 it follows that termination of Xia's algorithm with a graph with no edges implies that no r-clique exists. Example 1.7 shows that the other implication is not true in general.

Definition 1.1. Recall that a graph $G = (V,E)$ is *vertex r-colored by γ* iff γ is a surjective function of V onto the r-element set C such that if $\gamma(x)=\gamma(y)$, then x is not adjacent to y. We will denote a graph G that is vertex r-colored by γ by $(G=(V,E),\gamma)$. To avoid technicalities, in this paper we shall always assume that $r \geqslant 3$.

Definition 1.2. Let $(G=(V,E),\gamma)$ be a graph that is vertex r-colored by γ. A three-element-subset $\{x,y,z\}$ of V is called a *triangle* iff $\{x,y\}, \{x,z\}$ and $\{y,z\}$ are edges. Let $k \in \{1,\ldots,r\}$. An edge $\{x,y\} \in E$ *has a triangle with a k corner* iff $k \in C$ and there is a $z \in \gamma^{-1}(k)$ such that $\{x,y,z\}$ is a triangle.

Definition 1.3. Let $(G=(V,E),\gamma)$ be a graph that is vertex r-colored by γ. We define the removal step **REMOVE** (cf. [17], step II), which produces a new graph $G'=(V,E')$ with at most one less edge as follows:

REMOVE: Obtain G' from G by removing one edge $\{x,y\}$ that for some $k \in C \setminus \{\gamma(x),\gamma(y)\}$ does not have a triangle with a k-corner.

If this removal is not possible, let $G' = G$.

Clearly $(G'=(V,E'),\gamma)$ is again vertex r-colored by γ. *Xia's algorithm* keeps the vertex r-coloring of V by γ constant and iterates the removal step **REMOVE** until no more edges can be removed.

In particular, in the step from a graph G to a graph G', **REMOVE** does not remove any edges $\{x,y\}$ that for all $k \in C\setminus\{\gamma(x),\gamma(y)\}$ have a k corner in G. It can and does happen however that in G an edge $\{x,y\}$ has a k-corner for all $k \in C\setminus\{\gamma(x),\gamma(y)\}$, while in G' it does not anymore. Such higher-order effects are the strength of the algorithm.

Lemma 1.4. *Let $(G=(V,E),\gamma)$ be a finite graph that is vertex r-colored by γ. The graph G_{ret} that is returned by Xia's algorithm applied to $(G=(V,E),\gamma)$ is unique, i.e., it is independent of the choice of which edge to remove in every iteration of the step* **REMOVE**.

Proof (*For the idea cf.* [6]). Let $\mathscr{G}$ be the set of all possible graphs $G'=(V,E')$ that can be obtained by iterated application of **REMOVE** to $(G=(V,E),\gamma)$. Then $\mathscr{G}$ is ordered by $(V,E_1) \leqslant (V,E_2)$ iff $E_1 \subseteq E_2$. Note that (V,E_2) is an upper cover of (V,E_1) in $\mathcal{G}$ iff $E_2 = E_1 \cup \{\{x,y\}\}$ and there is a $k \in C \backslash \{\gamma(x), \gamma(y)\}$ such that $\{x,y\}$ has no triangle with a k corner in (V,E_2). Also note that G is the largest element of $\mathcal{G}$. We will show that any two elements of $\mathcal{G}$ that have a common upper cover also must have a common lower cover. This implies that $\mathcal{G}$ has a smallest element, which must be the unique graph obtained by applying Xia's algorithm to $(G=(V,E),\gamma)$.

Let (V,F) and (V,F') be distinct elements of $\mathcal{G}$ that have (V,U) as common upper cover in $\mathcal{G}$. Then $U = F \cup \{\{x,y\}\} = F' \cup \{\{x',y'\}\}$ and there are $k \in C \backslash \{\gamma(x), \gamma(y)\}, k' \in C \backslash \{\gamma(x'), \gamma(y')\}$ such that $\{x,y\}$ has no triangle with a k corner in (V,U) and $\{x',y'\}$ has no triangle with a k' corner in (V,U). But then $\{x,y\}$ has no triangle with a k corner in (V,F') and $\{x',y'\}$ has no triangle with a k' corner in (V,F). Thus (V,L) with $L := F \backslash \{\{x',y'\}\} = F' \backslash \{\{x,y\}\}$ is a common lower cover of (V,F) and (V,F'). $\square$

Lemma 1.5. *Let $(G=(V,E),\gamma)$ be a finite graph that is vertex r-colored by γ and let G_{ret} be the graph obtained by applying Xia's algorithm to $(G=(V,E),\gamma)$.*

1. *If $F \subseteq E$ is such that every edge $\{x,y\} \in F$ has for all $k \in C \backslash \{\gamma(x), \gamma(y)\}$ a triangle with a k corner in F, then G_{ret} will contain all the edges in F,*
2. *If G induces a complete graph on the r vertices $\{x_1, \ldots, x_r\}$, then G_{ret} also induces a complete graph on the r vertices $\{x_1, \ldots, x_r\}$,*
3. *If G_{ret} is a graph with no edges, then G does not contain a complete graph with r vertices,*
4. *Suppose $(G'=(V,E'),\gamma)$ is another graph that is vertex r-colored by γ and that $E \subseteq E'$. If Xia's algorithm applied to $(G'=(V,E'),\gamma)$ returns a graph with edge set F', then all edges of G_{ret} are contained in F'.*

Proof. For part 1 note that if $(V,\tilde{E})$ is obtained from G by applying **REMOVE** finitely many times and $F \subseteq \tilde{E}$, then **REMOVE** applied to $(V,\tilde{E})$ will not remove any of the edges in F. This proves that any graph $(V,\tilde{E})$ obtained from G by applying **REMOVE** finitely many times contains all edges in F. Part 2 follows from part 1 and part 3 follows from part 2.

Finally for part 4, let $G_j = (V,E_j)$, $j = 1, \ldots, n$ be a sequence of graphs such that $G_0 = G'$, G_{j+1} is obtained from G_j by applying **REMOVE** once and such that G_n has edge set F'. Define $H_0 := G$. If $H_0, \ldots, H_j$ such that every edge of H_i is an edge of G_i and such that H_i is obtained from H_{i-1} by applying **REMOVE** at most once have been defined, let $\{x,y\}$ be the edge that is removed from G_j to obtain G_{j+1}. If $\{x,y\}$ is not an edge of H_j, let $H_{j+1} := H_j$. Otherwise, there is a $k \in C \backslash \{\gamma(x), \gamma(y)\}$ such that $\{x,y\}$ has no triangle with a k corner in G_j. But then $\{x,y\}$ has no triangle with a k corner in H_j and one application of **REMOVE** turns H_j into $H_{j+1} := (V, E \backslash \{\{x,y\}\})$.

The sequence $H_0, \ldots, H_n$ starts with G and ends with a graph the edge set of which is contained in F', thus showing the claim. $\square$

Remark 1.6. Lemma 1.5, part 4 can also be used in cases where G has fewer vertices than G' as long as G contains a vertex of each color: Simply add the missing vertices as isolated vertices to G.

Example 1.7. If $r \geqslant l \geqslant 4$, then vertex r-colored graphs that do not contain an l-clique and for which Xia's algorithm does not remove a single edge can be constructed as follows: Let $V_{r,l} := \{(x,y)\colon x \in \{1,\dots,r\},\ y \in \{1,\dots,l-1\}\}$ and $E_{r,l} := \{\{(x_1,y_1),(x_2,y_2)\}\colon x_1 \neq x_2,\ y_1 \neq y_2\}$. Then $G_{r,l} = (V_{r,l}, E_{r,l})$ is vertex r-colored by $\gamma((x,y)) := x$. $G_{r,l}$ contains no l-clique, since if $\{(x_1,y_1),\dots,(x_l,y_l)\}$ were an l-clique, then $y_1,\dots,y_l \in \{1,\dots,l-1\}$ would be mutually distinct numbers, which is a contradiction. Finally if Xia's algorithm is applied to $(G_{r,l} = (V_{r,l}, E_{r,l}), \gamma)$, then no edge would ever be removed, as for each edge $\{(x_1,y_1),(x_2,y_2)\}$ and each $x_3 \notin \{x_1,x_2\}$, there is a $y_3 \notin \{y_1,y_2\}$ such that $\{(x_1,y_1),(x_2,y_2),(x_3,y_3)\}$ is a triangle of $G_{r,l}$.

Lemma 1.8. *Let $(G=(V,E),\gamma)$ be a finite graph that is vertex r-colored by γ.*

1. (*Cf. Step I in* [17]) *If x is a vertex such that there is a $k \in C$ such that no edge contains x and a $y \in \gamma^{-1}(k)$, then Xia's algorithm applied to $(G=(V,E),\gamma)$ returns a graph with no edge containing x,*
2. *If $k \neq k'$ and G is such that no edge contains an $x \in \gamma^{-1}(k)$ and a $y \in \gamma^{-1}(k')$, then Xia's algorithm applied to $(G=(V,E),\gamma)$ returns a graph with no edges.*

Proof. For part 1 note that no edge that contains x can have a triangle with a k corner. Thus all edges that contain x are removed.

For part 2, apply part 1 to all vertices in $\gamma^{-1}(k)$. Then Xia's algorithm applied to $(G=(V,E),\gamma)$ returns a graph with no edge containing an element of $\gamma^{-1}(k)$. But then no remaining edge could have a triangle with a k corner and thus Xia's algorithm must return a graph with no edges. $\square$

2. Maps satisfying conditions

Xia's algorithm becomes a very versatile tool through the translation process outlined in this section. The key insight is that many decision (and enumeration) problems can be formulated in the form 'Is there a function $f\colon P \to Q$ that satisfies the conditions $(\forall x \in P)[\sigma(x,f(x))]$ and $(\forall x \in P)(\forall y \in P\backslash\{x\})[\tau(x,f(x),y,f(y))]$?' (respectively, for enumeration, how many such maps are there) with the conditions written in a first-order language over an appropriate alphabet.

For basic logical terminology, cf. [9, Chapter I, Section 2]. In all logical constructions we will be using a first-order language (or predicate language) over an appropriate alphabet. $(\forall x \in X)[\cdots]$ will be used as an abbreviation for $(\forall x)[(x \in X) \Rightarrow \cdots]$.

Some problems that can be formulated in the form indicated above are:

1. (Cf. Xia's original article [17]) 'Does a given ordered set $(P, \leqslant)$ have a fixed point free order-preserving self map'? can be formulated as: Is there a function $f: P \to P$ such that

$$(\forall x \in P)[x \neq f(x)]$$

and

$$(\forall x \in P)(\forall y \in P \setminus \{x\})[(x \leqslant y) \Rightarrow (f(x) \leqslant f(y))]?$$

2. 'Is there an isomorphism from the graph $G = (V, E)$ to the graph $H = (W, F)$ with H such that $|V| = |W|$'? can be formulated as: Is there a function $f: V \to W$ such that

$$(\forall x \in V)(\forall y \in V \setminus \{x\})$$
$$[[f(x) \neq f(y)] \wedge [(\{x, y\} \in E) \Leftrightarrow (\{f(x), f(y)\} \in F)]]?$$

3. 'Does the finite graph $G = (V, E)$ have a Hamiltonian circuit'? can be formulated as follows: Is there a function $f: \{1, \ldots, |V|\} \to V$ such that

$$(\forall x \in \{1, \ldots, |V|\})(\forall y \in \{1, \ldots, |V|\} \setminus \{x\})$$
$$[[f(x) \neq f(y)] \wedge [(|x - y| \in \{1, |V| - 1\}) \Rightarrow (\{f(x), f(y)\} \in E)]]?$$

4. The solutions to the n-queen problem (place n queens on an $n \times n$ chessboard such that no two of them threaten each other) are functions $f: \{1, \ldots, n\} \to \{1, \ldots, n\}$ that satisfy

$$(\forall x \in \{1, \ldots, n\})(\forall y \in \{1, \ldots, n\} \setminus \{x\})$$
$$[[f(x) \neq f(y)] \wedge [|x - y| \neq |f(x) - f(y)|]].$$

In the following, lowercase greek letters will denote formulas in our predicate language and in parentheses after the letter we indicate the free (i.e., unquantified) variables or the intended replacement for the variable. Formulas $\sigma(p, q)$ and $\tau(p, q, p', q')$ (with p, q, p', q' variables) that can be used to formulate the above conditions as $(\forall x \in \langle \text{domain} \rangle)[\sigma(x, f(x))]$ and $(\forall x \in \langle \text{domain} \rangle)(\forall y \in \langle \text{domain} \rangle \setminus \{x\})[\tau(x, f(x), y, f(y))]$ (with x, $f(x)$, y, $f(y)$ replacing the variables p, q, p', q', respectively) are as follows:

1. $\sigma(p, q) := [p \neq q]$ and $\tau(p, q, p', q') := [(p \leqslant p') \Rightarrow (q \leqslant q')]$,
2. $\tau(p, q, p', q') := [[q \neq q'] \wedge [(\{p, p'\} \in E) \Leftrightarrow (\{q, q'\} \in F)]]$,
3. $\tau(p, q, p', q') := [[q \neq q'] \wedge [(|p - p'| \in \{1, |V| - 1\}) \Rightarrow (\{q, q'\} \in E)]]$,
4. $\tau(p, q, p', q') := [[q \neq q'] \wedge [|p - p'| \neq |q - q'|]]$.

Motivated by the above similarities between, on the surface, very different problems we define:

Definition 2.1. Let

1. P, Q be finite sets (possibly with additional structures such as relations or operations),

2. $\sigma(p,q)$ be a formula with two free variables over an appropriate alphabet,
3. $\tau(p,q,p',q')$ be a formula with four free variables over an appropriate alphabet.

We will call (P,Q,σ,τ) a *Xia configuration*. The *Xia graph* for the above Xia configuration is the graph $X(P,Q,\sigma,\tau)=(V,E)$ with vertices

$$V := \{(x,u) : x \in P,\ u \in Q,\ \sigma(x,u)\}$$

and edge set

$$E := \{\{(x,u),(y,v)\} : x \neq y,\ \tau(x,u,y,v) \wedge \tau(y,v,x,u)\}.$$

We define $\gamma((x,u)) := x$ for all $(x,u) \in V$ and we will from now on always assume that a Xia graph is vertex r-colored by the function γ.

This means that the vertices, edges and k-cliques of the Xia graph $X(P,Q,\sigma,\tau)$ are *partial* maps from P to Q whose domain has one, two or k points and which satisfy the conditions $(\forall x \in P)[\sigma(x,f(x))]$ and $(\forall x \in P)(\forall y \in P\backslash\{x\})[\tau(x,f(x),y,f(y))]$. This insight is the heart of the translation from given problems to finding $|P|$-cliques in Xia graphs $X(P,Q,\sigma,\tau)$, which we articulate once more in Theorem 2.2 (the proof is obvious). From now on in vertices of the Xia graph, the second component can always be considered as the image of the first component.

Theorem 2.2. *Let (P,Q,σ,τ) be a Xia configuration. Then $f : P \to Q$ is a map that satisfies*

$$(\forall x \in P)[\sigma(x,f(x))] \quad \textit{and} \quad (\forall x \in P)(\forall y \in P\backslash\{x\})[\tau(x,f(x),y,f(y))]$$

iff f is (as a subset of $P \times Q$) a $|P|$-clique in the Xia graph $X(P,Q,\sigma,\tau)$.

Corollary 2.3. *Let (P,Q,σ,τ) be a Xia configuration. If Xia's algorithm applied to the Xia graph $X(P,Q,\sigma,\tau)$ returns a graph with no edges, then there is no map $f : P \to Q$ such that*

$$(\forall x \in P)[\sigma(x,f(x))] \quad \textit{and} \quad (\forall x \in P)(\forall y \in P\backslash\{x\})[\tau(x,f(x),y,f(y))]$$

are satisfied.

Proof. By Lemma 1.5, part 1 and by Theorem 2.2 if such a map would exist, then Xia's algorithm would not terminate with a graph with no edges. □

Definition 2.4. Let P,Q be sets, let ρ_P be a binary relation on P and let ρ_Q be a binary relation on Q. Then $f : P \to Q$ is called a (ρ_P,ρ_Q)-*homomorphism* iff for all $x,y \in P$ we have that $x\rho_P y$ implies $f(x)\rho_Q f(y)$. When specification of the relations is not necessary we will refer to (ρ_P,ρ_Q)-homomorphisms as *relation-preserving* maps. We will also use the words endomorphism, isomorphism, etc., with their usual meanings in this context.

Example 2.5. The most important examples of Xia configurations to date are inspired by Xia's original article [17] on the fixed point property for ordered sets. The main object there are fixed-point-free order-preserving mappings that are modeled via Theorem 2.2 as cliques in a Xia graph. Underlying is the realization that any set of (ρ_P, ρ_Q)-homomorphisms $f : P \to Q$ can be modeled as the set of $|P|$-cliques of a Xia graph via

$$\sigma_{(\rho_P,\rho_Q)\text{-hom}}(p,q) := (p\rho_P p) \Rightarrow (q\rho_Q q)$$

and

$$\tau_{(\rho_P,\rho_Q)\text{-hom}}(p,q,p',q') := [(p\rho_P p') \Rightarrow (q\rho_Q q')].$$

In fact, the edges of the corresponding Xia graph would be the partial (ρ_P, ρ_Q)-homomorphisms with a domain of size 2, and the $|P|$-cliques f are exactly the (ρ_P, ρ_Q)-homomorphisms. Clearly by adding more conditions we can now model mappings that preserve several relations simultaneously and also satisfy other conditions. In the following, we list some frequently encountered conditions, which can be part of the σ and τ conditions in specific implementations.

1. *Injectivity*: $\tau_{\text{inj}}(p,q,p',q') := [q \neq q']$, causes the $|P|$-cliques to be injective mappings (doubles as a condition for surjectivity in case $|P| = |Q|$),
2. *Relational Isomorphism resp. Isomorphic Embedding*:

   $$\sigma_{(\rho_P,\rho_Q)\text{-iso}}(p,q) := [(p\rho_P p) \Leftrightarrow (q\rho_Q q)],$$

   together with

   $$\tau_{(\rho_P,\rho_Q)\text{-iso}}(p,q,p',q') := [(p\rho_P p') \Leftrightarrow (q\rho_Q q')]$$

   and $\tau_{\text{inj}}(p,q,p',q')$, causes the $|P|$-cliques to be isomorphisms between (P, ρ_P) and the image of P with the relation inherited from (Q, ρ_Q), (in particular these are isomorphisms between (P, ρ_P) and (Q, ρ_Q) in case $|P| = |Q|$),
3. *Retraction*: $\tau_{\text{retr}}(p,q,p',q') := [(p' = q) \Rightarrow (q' = p')]$, causes the $|P|$-cliques to be retractions, i.e, idempotent functions,
4. *Image restrictions*: For $A \subseteq P$ and $B \subseteq Q$

   $$\sigma_{A\text{ to }B}(p,q) := [(p \in A) \Rightarrow (q \in B)],$$

   causes the $|P|$-cliques to be maps that map A into B.

The associated Xia graphs and the interpretations of their $|P|$-cliques for some examples are:

1. From now on let '$x \sim y$' denote '$x \leqslant y$ or $x \geqslant y$'. $G_{\text{fpfree}}(P) := X(P, P, [p \nsim q], \tau_{(\leqslant,\leqslant)\text{-homo}})$ is the fixed-point-free endomorphism graph of a finite ordered set P with order $\leqslant$ (cf. [14, Definition 2.4], or originally [17, start of Section 2], the problem is also the first in our introduction here). The $|P|$-cliques are the fixed-point-free order-preserving self maps of P (such maps cannot have a single point x that is comparable to its image). In this fashion Xia's algorithm can be applied to the

NP-complete (cf. [4]) problem of whether P has a fixed-point-free order-preserving self-map.

2. $X(P,Q,\bigwedge_{i=1}^{n}\sigma_{(\rho_i,\rho_i')\text{-homo}},\bigwedge_{i=1}^{n}\tau_{(\rho_i,\rho_i')\text{-homo}})$, the joint homomorphism graph. $|P|$-cliques are the (jointly) relation-preserving maps from one set P with binary relations $\rho_1,\dots,\rho_n$ to another set Q with binary relations $\rho_1',\dots,\rho_n'$, i.e., the (ρ_i,ρ_i')-homomorphisms. (We gave the formulation in terms of preservation of multiple relations since the inspiration for this very general use of Xia's algorithm comes from a question about the enumeration of functions preserving multiple relations from a 216-element set to a 6-element set asked by H. Priestley.)

   (a) With the relations being order relations $P=Q$, $n=2$ and $\rho_i=\rho_i'$ $(i=1,2)$ we gain access to the problem if two given orders ρ_1 and ρ_2 are perpendicular (two orders are called perpendicular iff they only have the identity and the constant functions as joint endomorphisms, cf. [18]).
   (b) For $n=1$, Q a graph, P a $|Q|$-cycle/path and the injectivity condition τ_{inj} added, we obtain access to the question whether a given finite graph has a HAMILTONIAN CIRCUIT/PATH (problems [GT37] and [GT39] in [8]; from now on references of the preceeding form will refer to the list of problems in the appendix of [8]; HAMILTONIAN CIRCUIT was earlier shown to be accessible with a slightly different formulation).
   (c) More generally, if $n=1$, Q is a graph and P is another fixed graph, and if we add the injectivity condition τ_{inj} we obtain access to the SUBGRAPH ISOMORPHISM problem [GT48] and its various sub-variants such as for example 2b above or DIRECTED HAMILTONIAN CIRCUIT/PATH [GT38], CLIQUE [GT19] (not useful when applied to Xia graphs), INDEPENDENT SET [GT20], BALANCED COMPLETE BIPARTITE SUBGRAPH [GT24], ISOMORPHIC SPANNING TREE [ND8], LONGEST CIRCUIT with $l(e)=1$ for all e [ND28], LONGEST PATH with $l(e)=1$ for all e [ND29], etc.
   (d) For $n=1$ and ρ_1 and ρ_1' being order relations and with the appropriate image restrictions, we gain access to the NP-complete problem OPEXT of [4] of whether there exists an order-preserving map subject to certain constraints. This naturally can also be used for the problem of existence of maps that preserve (multiple) relations and satisfy certain constraints.
   (e) With $n=1$, P an ordered set, Q a $|P|$-chain, $\tau_{(\leqslant_P,\leqslant_Q)\text{-hom}}$ and injectivity τ_{inj} added we obtain a graph that has all linear extensions of P as its $|P|$-cliques.

3. $X(P,Q,\bigwedge_{i=1}^{n}\sigma_{(\rho_i,\rho_i')\text{-iso}},\bigwedge_{i=1}^{n}\tau_{(\rho_i,\rho_i')\text{-iso}})$, the joint isomorphism graph. The $|P|$-cliques are the (joint) relation-isomorphisms from one set P with binary relations $\rho_1,\dots,\rho_n$ to another set Q with binary relations $\rho_1',\dots,\rho_n'$, i.e., the maps that are (ρ_i,ρ_i')-homomorphisms with inverses being (ρ_i',ρ_i)-homomorphisms.

   (a) With $n=1$ and P,Q graphs this gives access to the problem GRAPH ISOMORPHISM [OPEN1] (already mentioned in the introduction) or more generally to the problem of when (P,ρ_P) and (Q,ρ_Q) have a relation-isomorphism. (The complexity status of this problem is still unclear even in the case of ordered sets or graphs.)

(b) Again, we could also add constraints and ask if there is an isomorphism that satisfies these constraints.

Example 2.6. Xia configurations can be brought to bear on SAT, resp. 3SAT also. This is done via a construction in [4, Theorem 4.1]. Let an instance of SAT with clauses $C_1,\dots,C_m$ and literals $x_1,\dots,x_n$ be given. Order the literals in each clause in a fixed fashion (for example sort them such that the indices increase). Let k_i be the number of literals in clause C_i. Define graphs $G=(V_G,E_G)$ and $H=(V_H,E_H)$ as follows:

The vertices of G are called $C_1^*,\dots,C_m^*$ and $x_1^*,\dots,x_n^*$. There is an edge in G between C_i^* and x_j^* iff the literal x_j or its negation occurs in C_i and there are no other edges.

In H we have vertices $C_1^1,\dots,C_1^{k_1},C_2^1,\dots,C_2^{k_2},\dots,C_m^1,\dots,C_m^{k_m}$ and vertices $x_1^{\mathrm{T}},x_1^{\mathrm{F}},x_2^{\mathrm{T}},x_2^{\mathrm{F}},\dots,x_n^{\mathrm{T}},x_n^{\mathrm{F}}$. Again edges only exist between C's and x's. If x_j or its negation occurs in C_i, but not in the lth place, put an edge between C_i^l and x_j^{T} and an edge between C_i^l and x_j^{F}. If x_j occurs in the lth place of C_i, put an edge between C_i^l and x_j^{T}. If $\overline{x_j}$ occurs in the lth place of C_i, put an edge between C_i^l and x_j^{F}. These are all the edges.

It is now easy to verify (cf. [4, Theorem 4.1]), that the instance of SAT has a satisfying assignment iff there is an edge-preserving map f from G to H such that $f(C_i^*)\in\{C_i^1,\dots,C_i^{k_i}\}$ and $f(x_j^*)\in\{x_j^{\mathrm{T}},x_j^{\mathrm{F}}\}$: Indeed, if there is such a map, then the assignment of 'true' to x_j iff $f(x_j^*)=x_j^{\mathrm{T}}$ gives a satisfying assignment. For the converse, fix a satisfying assignment. Then for each C_i there is a literal or a negation thereof in C_i that must evaluate to being true, say one is in the l_ith place. Map C_i^* to $C_i^{l_i}$ and map x_j^* to x_j^{T} iff x_j is assigned the value 'true'. This map is as desired.

The Xia configuration to be used would have the conditions

$$\sigma(p,q) := [(p=C_i^*)\Rightarrow(q\in\{C_i^1,\dots,C_i^{k_i}\})]$$

$$\wedge[(p=x_j^*)\Rightarrow(q\in\{x_j^{\mathrm{T}},x_j^{\mathrm{F}}\})],$$

$$\tau(p,q,p',q') := [(\{p,p'\}\in E_G)\Rightarrow(\{q,q'\}\in E_H)].$$

The associated Xia graph has $2n+\sum_{l=1}^m k_l$ vertices. These configurations are another source of $(m+n)$-colored Xia graphs without an $(m+n)$-clique for which Xia's algorithm does not return a graph with no edges. For example, the Xia configuration associated with the non-satisfiable 3-variable, 8-clause instance of 3SAT has no 11-clique, but Xia's algorithm does not return a graph with no edges.

This, in turn should (via the NP-completeness proofs) make it possible to generate instances of NP-complete problems that can be tackled with Xia configurations and Xia's algorithm such that no clique as desired exists, but Xia's algorithm does not return a graph with no edges. For the fixed point property for ordered sets, the above-mentioned instance of 3SAT paired with the construction in [4] does exactly that. It is worth pointing out however, that the reduction that Xia's algorithm does for these sets still is tremendous and that so far all examples of ordered sets with the fixed point property for which Xia's algorithm returns a graph with edges are constucted in this fashion.

3. Performance guarantees

A run-time and memory estimate for Xia's algorithm is quite easy in terms of the number k of vertices of the Xia graph (note that $k \leqslant |P| \cdot |Q|$ for all Xia graphs). If our Xia graph has k vertices, it has at most $(k/2)(k-1)$ edges. The edges, of course, are the bulk of the memory requirement for the algorithm, so that the memory requirement is on the order of

$$\frac{k}{2}(k-1) + \mathrm{O}(k).$$

(The $\mathrm{O}(k)$ is needed to store the 'encoding and decoding' for the specific Xia-configuration.)

To check if an edge can be removed we need at most $(k-2)$ 'triangle-checks' (checks if the edge forms a triangle with a given vertex). Thus it takes at most $(k/2)(k-1)(k-2)$ 'triangle-checks' until we can remove an edge (resp. until it is confirmed that no edges can be removed). Since we have $\leqslant (k/2)(k-1)$ edges, it takes at most

$$\frac{k^2}{4}(k-1)^2(k-2)$$

'triangle-checks' until Xia's algorithm terminates. Initial testing with the fixed point property and with rigidity for ordered sets suggests, that the algorithm actually takes on the order of $(k/2)(k-1)(k-2)$ (possibly multiplied with a slowly growing function) 'triangle-checks' if implemented such that all edges are checked sequentially for removability, removing each removable edge until a pass occurs without a single removal.

As Example 2.6 shows, for each NP-complete problem which can be tackled using Xia graphs (with, say, r colors) there should be instances in which no r-clique exists and yet Xia's algorithm terminates with a graph that still has edges. In such instances a depth-first-search for an r-clique becomes necessary. This is usually done with a backtracking algorithm. (For a multi-purpose backtracking algorithm, cf. [11].) From an implementation point of view, depth-first-search using a backtracking algorithm is most efficient if the conditions to be checked can be checked quickly. The Xia-encoding, even without the reduction, provides an improvement in the efficiency of depth-first-search for many problems as it reduces all the needed checks of the condition τ to checking the one condition whether there is an edge in the Xia graph. This improvement, however, pales in comparison to the improvement for many problems achieved through the reduction.

The time a depth-first-search for an r-clique would take depends on how 'dense' r-cliques are in the reduced Xia graph, respectively, how many edges are left in a reduced Xia graph with no r-cliques. Thus, it becomes interesting to look for results that guarantee removal of certain classes of edges (performance guarantees of the first kind) and for results that prove that in certain instances Xia's algorithm must terminate with a graph with no edges (performance guarantees of the second kind).

In the following, we present some such performance guarantees. These can only be a start of a deeper investigation of how useful Xia's algorithm is for the many problems to which it can be applied. We limit ourselves to investigations related to relation preserving maps with primary focus on the fixed point property for ordered sets and rigidity and isomorphism for ordered sets and graphs. For these problems experiments so far suggest Xia's algorithm to be an excellent tool. In fact, for the fixed point property for ordered sets, Xia's algorithm, followed by depth-first-search if necessary, is the best algorithm available to date.

3.1. Performance guarantees of the first kind: vertices and edges that are guaranteed to be removed

The performance guarantees of the first kind presented here are applicable to relation preserving maps in general. The motivation however comes from considering the fixed point property and graph/order isomorphism/rigidity. Results formulated for rigidity naturally have obvious analogues for the isomorphism problem.

The following definition is motivated by the notion of distance in a graph, resp. notions of distances in ordered sets.

Definition 3.1. Let P be a set, let ρ be a binary relation on P and let $x, y \in P$. Then the sequence $(x = a_0, a_1, \ldots, a_n = y)$ with $n \geqslant 1$ of $n+1$ elements of P with $a_{i-1} \neq a_i$ is called a *ρ-trail* from x to y iff for all $i \in \{1, \ldots, n\}$ we have $a_{i-1} \rho a_i$ or $a_i \rho a_{i-1}$. If $(a_0, \ldots, a_n)$ is a ρ-trail, then the sequence $(s_1, \ldots, s_n)$ of $s_i \in \{0, 1, 2\}$ is called the *signature* of $(a_0, \ldots, a_n)$ iff for all $i \in \{1, \ldots, n\}$ we have that

1. $s_i = 2$ iff $a_{i-1} \rho a_i$ and $a_i \rho a_{i-1}$,
2. $s_i = 1$ iff $a_{i-1} \rho a_i$ and $a_i \not\rho a_{i-1}$,
3. $s_i = 0$ iff $a_i \rho a_{i-1}$ and $a_{i-1} \not\rho a_i$.

A sequence $(s_1, \ldots, s_n)$ is called a *subsignature* of the signature $(t_1, \ldots, t_m)$ iff there is a strictly increasing function $f : \{1, \ldots, n\} \to \{1, \ldots, m\}$ such that $s_i = t_{f(i)}$ or $t_{f(i)} = 2$.

Lemma 3.2. *Let P, Q be sets, let ρ_P be a binary relation on P, let ρ_Q be a binary relation on Q and let $f : P \to Q$ be a (ρ_P, ρ_Q)-homomorphism. If there is a ρ_P-trail $(x = a_0, \ldots, a_n = y)$ from $x \in P$ to $y \in P$ with signature $(s_1, \ldots, s_n)$, then there is a ρ_Q-trail from* $f(x)$ to $f(y)$, *whose signature is a subsignature of* $(s_1, \ldots, s_n)$.

Proof. The sequence $(f(a_0), \ldots, f(a_n))$ can be transformed into a ρ_Q-trail by successively removing the subsequences $(f(a_k), \ldots, f(a_j))$ for all instances in which $k \leqslant j$ and $f(a_{k-1}) = f(a_j)$. This ρ_Q-trail has a signature that is a subsignature of $(s_1, \ldots, s_n)$. $\square$

Theorem 3.3. *Let (P, Q, σ, τ) be a Xia configuration, let ρ_P be a binary relation on P, let ρ_Q be a binary relation on Q and suppose that $\tau(p, q, p', q') = \tilde{\tau}(p, q, p', q') \wedge [(p\rho_P p') \Rightarrow (q\rho_Q q')]$. If $\{(x, u), (y, v)\}$ is an edge of the Xia graph such that there is a ρ_P-trail from x to y with signature $(s_1, \ldots, s_n)$ and such that no ρ_Q-trail from u to v has a signature that is a subsignature of $(s_1, \ldots, s_n)$, then Xia's algorithm erases the edge $\{(x, u), (y, v)\}$.*

Proof. This is an induction on n with $n = 1$ being vacuously true: If $x\rho_P y(y\rho_P x)$ is true and $u\rho_Q v$ $(v\rho_Q u)$ is false, then $\{(x, u), (y, v)\}$ is not an edge of the Xia graph.

For the induction step $\{1, \ldots, n-1\} \to n$ assume the result is true for ρ_P-trails of length less than n and let G_{ret} be the graph that is returned by Xia's algorithm applied to the Xia graph. Let $(x = a_0, \ldots, a_n = y)$ be a ρ_P-trail from x to y with signature $(s_1, \ldots, s_n)$ such that no ρ_Q-trail from u to v has a signature that is a subsignature of $(s_1, \ldots, s_n)$. If $a_{n-1} = y$, we are done by induction hypothesis. Moreover, note that $a_{n-1} \neq x$, since otherwise the ρ_Q-trail (u, v) has a signature that is a subsignature of (s_n) and hence of $(s_1, \ldots, s_n)$. Thus we can assume $a_{n-1} \notin \{x, y\}$ Now for each vertex (a_{n-1}, z) of the Xia graph for which $\{(a_{n-1}, z), (y, v)\}$ is an edge in G_{ret} the edge $\{(a_{n-1}, z), (x, u)\}$ cannot be an edge of G_{ret}. Indeed, if it were, there would be a ρ_Q-trail from u to z that has a signature which is a subsignature of $(s_1, \ldots, s_{n-1})$ and extending this trail by appending v would give a ρ_Q-trail from u to v that has a signature which is a subsignature of $(s_1, \ldots, s_n)$, which is a contradiction. Thus, the edge $\{(x, u), (y, v)\}$ would have no triangle with an a_{n-1} corner in G_{ret} and can thus not be an edge of G_{ret}. $\square$

The above says for order-preserving maps and graph homomorphisms that in the associated Xia graph every edge $\{(x, u), (y, v)\}$ with the distances between the images u, v exceeding the distances between the preimages x, y will be erased. It can also be used as a tool to derive further performance guarantees. Note that trails allow for repetition of vertices.

When looking for relational isomorphisms one can easily modify the above proof to show every edge $\{(x, u), (y, v)\}$ such that there is a trail from x to y such that no trail from u to v has the *same* signature will be erased.

Definition 3.4. Let P be a finite ordered set. The *rank* of $p \in P$ is the length of the longest chain from p to a minimal element of P. It is denoted by rank(P).

Theorem 3.5. *Let P be a finite ordered set with order $\leqslant$. Let G_{ret} be the graph obtained through application of Xia's algorithm to the Xia graph $X(P, P, [p = p], \tau_{\text{inj}} \wedge \tau_{(\leqslant, \leqslant)\text{-iso}})$. Then each vertex (x, u) with* rank$(x) >$ rank(u) *is isolated in G_{ret}.*

Proof. Assume the contrary. Let k be the smallest number such that there is an element x of rank k and an element u of rank $< k$ such that (x, u) is not isolated in G_{ret}. Let $y < x$ with rank$(y) =$ rank(u) (since rank$(x) >$ rank$(u) \geqslant 0$ such a y exists) and let

$v \in P$ be such that $\{(x,u),(y,v)\}$ is an edge in G_{ret} (by Lemma 1.8, part 1 such a v must exist). Then $v \leqslant u$ and $\text{rank}(v) \geqslant \text{rank}(y) = \text{rank}(u)$, which implies $u = v$, a contradiction to the injectivity condition in our Xia configuration. Thus all vertices (x,u) with $\text{rank}(x) > \text{rank}(u)$ are isolated in G_{ret}. $\square$

Remark 3.6. Unfortunately, in the above situation Xia's algorithm does not necessarily isolate all vertices (x,u) with $\text{rank}(x) \neq \text{rank}(u)$. As an example, consider the disjoint sum of a singleton ordered set $\{x\}$ with five 2-chains $C_i = \{c_1^i < c_2^i\}$. For every edge $\{(x,c_a^k),(c_b^i,c_b^j)\}$ and each $c_u^v \neq c_b^i$, there is a 2-chain C_z that does not contain any of the c's so far and such that $\{(x,c_a^k),(c_b^i,c_b^j),(c_u^v,c_u^z)\}$ is a triangle. Since vertices (c_x^i,c_x^j) are part of automorphisms, they will not be isolated and thus edges $\{(x,c_a^k),(c_b^i,c_b^j)\}$ will not be erased independent of the rank of c_a^k.

This example also shows that vertices (x,u) with x and u having different degrees need not necessarily be isolated. It is worth noting that the example presented here is not rigid. There are no known examples of rigid ordered sets that behave as above.

Moreover, for the isomorphism/rigidity problem one could run a variant of Xia's algorithm which first uses the vertex coloring given by projection on the first component and then uses the vertex coloring given by projection on the second component. Such an algorithm would isolate all vertices (x,u) with $\text{rank}(x) \neq \text{rank}(u)$.

Performance guarantees of the first kind as presented here show that Xia's algorithm automatizes some reductions that one would want to implement before searching for a map/isomorphism. Observation shows that actual performance is much more impressive.

The above performance guarantees can be used to prove results about performance of Xia's algorithm in special classes of ordered sets/graphs. In practice, working with graph or order rigidity, one could naturally implement conditions on rank or degrees as pre-processors. One can also make them part of the Xia configuration as $\sigma_{\text{rank}}(p,q) := [\text{rank}(p) = \text{rank}(q)]$ resp. as $\sigma_{\text{deg}}(p,q) := [\deg(p) = \deg(q)]$. In doing so, results such as Theorem 3.5 recede into the theory. However, when tackling graph isomorphism/rigidity using σ_{deg} one obtains the following performance guarantee: Every edge $\{(x,u),(y,v)\}$ such that there is a path (repetitions allowed) from x to y with degree sequence $\deg(x) = d_0, d_1, \ldots, d_n = \deg(y)$ and such that there is no path (repetitions allowed) from u to v with the same degree sequence will (by the remark after Theorem 3.3) be erased by Xia's algorithm. Implementing such a condition directly appears impractical.

3.2. Performance guarantees of the second kind: special cases in which Xia's algorithm returns a graph with no edges

This section presents a new performance guarantee for Xia's algorithm that says that for certain input Xia's algorithm applied to the fixed point problem terminates with a graph with no edges. As described in the open Problem 3.7, performance guarantees of the second kind can lead to new results stating polynomial solvability of special cases of NP-complete problems. This partly explains the small number of such performance guarantees available so far.

The first performance guarantee of the second kind was proved by Xia [17, Section 5]. In our language, it is shown in [17, Section 5] that if, for a finite ordered set P, Xia's algorithm applied to $X(P,P,[q \not\sim p],\tau_{(\leqslant,\leqslant)\text{-homo}})$ returns a graph that has edges, then P has a fixed-point-free isotone relation (for isotone relations, cf. [16]). Since dismantable ordered sets have no isotone relations (cf. [16, Theorem 5.7]), we conclude that Xia's algorithm applied to $X(P,P,[q \not\sim p],\tau_{(\leqslant,\leqslant)\text{-homo}})$ for finite dismantable ordered sets returns a graph that has no edges. Related is the following.

Open Question 3.7. Connectedly collapsible ordered sets are a generalization of dismantable ordered sets introduced in [14, Definition 4.26; 15]. As proved in [7], Theorem 6 and mentioned in [14], Corollary 5.7 ordered sets of dimension 2 have the fixed point property iff they are connectedly collapsible. A proof that shows Xia's algorithm applied to $X(P,P,[q \not\sim p],\tau_{(\leqslant,\leqslant)\text{-homo}})$ for finite connectedly collapsible ordered sets returns a graph that has no edges would thus show that the fixed point property for ordered sets of dimension 2 is decidable in polynomial time. Such a proof currently eludes the authors. More detailed investigation of the properties of isotone relations could shed a light on this problem and also on the performance of Xia's algorithm for the fixed point property in general.

Our new performance guarantee of the second kind is the 'Xia-version' of a result by Baclawski and Björner (cf. [2, Corollary 2.6]). We use a simple modification of the proof of this result as shown by Edelman [5]. Recent advances in finding more combinatorial proofs for Baclawski and Björner's results (cf. [1,10] or the overview in [14]) might lead to further insights when using Xia's algorithm on structures that so far were only accessible through algebraic topology.

Definition 3.8. A subset C of an ordered set P is called a *cutset* iff every maximal chain of P intersects C. A cutset is called *coherent* iff each of its nonempty bounded subsets has a supremum or an infimum (but not necessarily both, even if upper and lower bounds exist).

Definition 3.9. Let P be a finite ordered set with a coherent cutset C. Define

$$S_C := \left\{x \in P\colon x = \bigvee A \text{ for some } A \subseteq C \text{ with } |A| \geqslant 2\right\},$$

$$I_C := \left\{x \in P\colon x = \bigwedge A \text{ for some } A \subseteq C \text{ with } |A| \geqslant 2\right\}.$$

We will call $x, y \in S_C \cup I_c$ *in regular position* iff $x > y$ and

1. $x \in S_C$ and $y \in I_C$, or
2. $x, y \in S_C$, or
3. $x, y \in I_C$.

For $x, y \in S_C \cup I_C$ in regular position we let

$$M(x,y) := \begin{cases} \{c \in C\colon x \geqslant c \geqslant y\} & \text{if } x \in S_C,\ y \in I_C, \\ \{c \in C\colon x \geqslant y \geqslant c\} & \text{if } x, y \in S_C, \\ \{c \in C\colon c \geqslant x \geqslant y\} & \text{if } x, y \in I_C. \end{cases}$$

Lemma 3.10. *Let P be a finite ordered set with a coherent cutset C. For edges $\{(x,u),(y,v)\}$ with $x, y \in S_C \cup I_C$ in regular position, let*

$$H(x,u,y,v) := \begin{cases} \{p \in P\colon u \geqslant p \geqslant v\} & \text{if } x \in S_C,\ y \in I_C, \\ \{p \in P\colon u \geqslant v \geqslant p\} & \text{if } x, y \in S_C, \\ \{p \in P\colon p \geqslant u \geqslant v\} & \text{if } x, y \in I_C. \end{cases}$$

Then Xia's algorithm applied to the Xia graph $X(P, P, [p \nsim q], \tau_{(\leqslant,\leqslant)\text{-homo}})$ will erase all edges $\{(x,u),(y,v)\}$ with $x, y \in S_C \cup I_C$ in regular position such that $C \cap H(x,u,y,v) \subseteq M(x,y)$ and $C \cap H(x,u,y,v)$ is a cutset of $H(x,u,y,v)$.

Proof. Let $G_{\text{ret}} = (V_{\text{fpfree}}, E_{\text{ret}})$ be the graph obtained by applying Xia's algorithm to the Xia graph $X(P, P, [p \nsim q], \tau_{(\leqslant,\leqslant)\text{-homo}})$. The proof that G_{ret} does not contain the edges in question is an induction on $n := |C \cap H(x,u,y,v)|$.

For $n=1$ assume $\{(x,u),(y,v)\} \in E_{\text{ret}}$ is an edge such that $x > y$ and $x \in S_C$, $y \in I_C$ (resp. $x, y \in S_C$, resp. $x, y \in I_C$), $C \cap H(x,u,y,v) \subseteq M(x,y)$ and $C \cap H(x,u,y,v)$ is a cutset of $H(x,u,y,v)$. Let $\{c\} = C \cap H(x,u,y,v)$. Since c is comparable to u and v we have $c \notin \{x,y\}$. Moreover, $c \in M(x,y)$ and thus $x > c > y$ (resp. $x > y > c$, resp. $c > x > y$). Suppose $d \in P$ is such that $\{(x,u),(y,v),(c,d)\}$ is a triangle in G_{ret}. Then, by definition of our Xia graph, $u \geqslant d \geqslant v$ (resp. $u \geqslant v \geqslant d$, resp. $d \geqslant u \geqslant v$). By definition of a cutset there is an element $c' \in C \cap H(x,u,y,v)$ such that $\{u,d,v,c'\}$ is a chain. However then since $C \cap H(x,u,y,v)$ has only one element we have $c' = c$ and thus $c \sim d$, a contradiction.

For the induction step $\{1,\ldots,n-1\} \to n$ the induction hypothesis is that no edge $\{(\tilde{x},\tilde{u}),(\tilde{y},\tilde{v})\}$ with $\tilde{x}, \tilde{y} \in S_C \cup I_C$ in regular position, $C \cap H(\tilde{x},\tilde{u},\tilde{y},\tilde{v}) \subseteq M(\tilde{x},\tilde{y})$ and $C \cap H(\tilde{x},\tilde{u},\tilde{y},\tilde{v})$ being a cutset of $H(\tilde{x},\tilde{u},\tilde{y},\tilde{v})$ with $< n$ elements is in E_{ret}. Assume $\{(x,u),(y,v)\}$ is an edge of G_{ret} with $x, y \in S_C \cup I_C$ in regular position, $C \cap H(x,u,y,v) \subseteq M(x,y)$, and $C \cap H(x,u,y,v)$ being a cutset of $H(x,u,y,v)$ with n elements.

First consider the case in which $C \cap H(x,u,y,v)$ has a supremum s. Then $s \notin \{x,y\}$, since s is comparable to u and to v. If $x \in S_C$, $y \in I_C$ (resp. $x, y \in S_C$, resp. $x, y \in I_C$) we have $x > s > y$ (resp. $x > y > s$, resp. $s > x > y$). Let $t \in P$ be such that $\{(x,u),(y,v),(s,t)\}$ is a triangle in G_{ret}. Then no element of $C \cap H(x,u,y,v)$ is above t, since otherwise $s \sim t$, which is not possible. Moreover by definition of our Xia graph $u \geqslant t \geqslant v$ (resp. $u \geqslant v \geqslant t$, resp. $t \geqslant u \geqslant v$), which means $H(s,t,y,v) \subseteq H(x,u,y,v)$ (resp. $H(y,v,s,t) \subseteq H(x,u,y,v)$, resp. $H(s,t,x,u) \subseteq H(x,u,y,v)$).

Now $C\cap H(s,t,y,v)$ (resp. $C\cap H(y,v,s,t)$, resp. $C\cap H(s,t,x,u)$) is a cutset of $H(s,t,x,v)$ (resp. $H(y,v,s,t)$, resp. $H(s,t,x,u)$) with fewer elements than $C\cap H(x,u,y,v)$: Indeed, let $K\subseteq H(s,t,y,v)$ (resp. $K\subseteq H(y,v,s,t)$, resp. $K\subseteq H(s,t,x,u)$) be a chain. Then $K\cup\{t\}$ is a chain in $H(x,u,y,v)$. Thus there is a $c\in C\cap H(x,u,y,v)$ such that $K\cup\{t,c\}$ is a chain. Since $c\not\geqslant t$ we have $c\in C\cap H(s,t,y,v)$ (resp. $c\in C\cap H(y,v,s,t)$, resp. $c\in C\cap H(s,t,x,u)$) also. Thus $C\cap H(s,t,y,v)$ (resp. $C\cap H(y,v,s,t)$, resp. $C\cap H(s,t,x,u)$) is a cutset of $H(s,t,y,v)$ (resp. $H(y,v,s,t)$, resp. $H(s,t,x,u)$). It is contained in $C\cap H(x,u,y,v)$ and must be properly contained and hence have less than n elements since otherwise $s\sim t$. Moreover, then $C\cap H(s,t,y,v)\subseteq M(s,y)$ (resp. $C\cap H(y,v,s,t)\subseteq M(y,s)$, resp. $C\cap H(s,t,x,u)\subseteq M(s,x)$), since all elements of $C\cap H(x,u,y,v)$ are below s and above y (resp. below y, resp. above x). But then by induction hypothesis $\{(s,t),(y,v)\}$ (resp. $\{(y,v),(s,t)\}$, resp. $\{(s,t),(x,u)\}$) is not an edge of G_{ret}, contradiction.

The case in which $H(x,u,y,v)$ has an infimum is treated dually. □

Theorem 3.11 (Cf. Edelman [5]). *Let P be a finite ordered set with a coherent cutset C that has a supremum or an infimum (or, equivalently, C is a cutset such that each nonempty subset has a join or a meet, but not necessarily both, even if upper and lower bounds exist). Then Xia's algorithm applied to the Xia graph $X(P,P,[p\not\sim q],\tau_{(\leqslant,\leqslant)\text{-homo}})$ returns a graph with no edges.*

Proof. Suppose, without loss of generality, that C has a supremum s. Let (s,t) be a vertex of $G_{\mathrm{fpfree}}(P)$. Then $t\not\sim s$. Thus for all $c\in C$ with $c\sim t$ we have $c\leqslant t$. Let $C':=\{c\in C\colon c\leqslant t\}$. Since $s\not\sim t$ we have $C'\neq C$. If C' has a supremum s', then $s\geqslant s'$. Moreover $s\neq s'$, since otherwise $s\sim t$. Thus s,s' are in regular position. For all edges $\{(s,t),(s',t')\}$ of $G_{\mathrm{fpfree}}(P)$ we have $C\cap H(s,t,s',t')\subseteq C\cap(\downarrow t)=M(s,s')$. Moreover $C\cap H(s,t,s',t')$ is a cutset of $H(s,t,s',t')$: Every chain $K\subseteq H(s,t,s',t')$ is such that $K\cup\{t\}$ is a chain. Thus there is a $c\in C\cap(\downarrow t)=C'$ such that $K\cup\{t,c\}$ is a chain. Now $c\geqslant t'$ would mean that $s'\geqslant c\geqslant t'$. Thus $c\leqslant t'$, i.e., $c\in H(s,t,s',t')$. Thus by Lemma 3.10 all edges $\{(s,t),(s',t')\}$ are being erased by Xia's algorithm. By Lemma 1.8, part 1 this means that all edges that contain (s,t) are being erased by Xia's algorithm.

If C' has an infimum i, then $s\geqslant i$ and s,i are in regular position ($i=s$ would mean that C has one element, in which case P is dismantable and there is nothing to prove). For all edges $\{(s,t),(i,j)\}$ of $G_{\mathrm{fpfree}}(P)$ we have $C\cap H(s,t,i,j)\subseteq C\cap(\downarrow t)=M(s,i)$. Moreover $C\cap H(s,t,i,j)$ is a cutset of $H(s,t,i,j)$: Every chain $K\subseteq H(s,t,i,j)$ is such that $K\cup\{t\}$ is a chain. Thus there is a $c\in C\cap(\downarrow t)=C'$ such that $K\cup\{t,c\}$ is a chain. Now $c\leqslant j$ would mean that $i\leqslant c\leqslant j$. Thus $c\geqslant j$, i.e., $c\in H(s,t,i,j)$. Thus by Lemma 3.10 all edges $\{(s,t),(i,j)\}$ are being erased by Xia's algorithm. By Lemma 1.8, part 1 this means that all edges that contain (s,t) are being erased by Xia's algorithm.

In summary all edges that contain a vertex in $\gamma^{-1}(s)$ are being erased, which means Xia's algorithm must return a graph with no edges. □

Remark 3.12. The above modification of Edelman's proof illustrates a new possibility of proving that the fixed point property can be verified in polynomial fashion in certain special cases. Let the j-simplex modification of Xia's algorithm be the algorithm that eliminates edges $\{x, y\}$ such that there are $(j-2)$ colors $c_1, \ldots, c_{j-2}$ in $C \backslash \{\gamma(x), \gamma(y)\}$ such that $\{x, y\}$ is not part of a K_j with corner colors $c_1, \ldots, c_{j-2}, \gamma(x), \gamma(y)$. Suppose one has a combinatorial proof of a characterization of the fixed point property in a special class, which only involves a fixed finite number j of points at any given stage (just as Edelman's proof involves only up to three points at every stage). Then one should be able to modify the proof into a proof that says that the j-simplex modification of Xia's algorithm terminates with a graph with no edges for all sets in the given class with the fixed point property (just as Edelman's proof shows this for sets with the cutset property). Note that this is independent of the complexity of the combinatorial condition given. Verifying the cutset property takes on the order of

$$\binom{|P|}{\text{width}(P)}$$

steps. If the goal is to determine if the ordered set has the fixed point property, this is much less efficient than Xia's algorithm for widths $\geqslant 10$.

3.3. Tests run

The performance guarantees shown here, show that in many cases Xia's algorithm is guaranteed to remove many edges of a Xia graph. Test performance is even more impressive. Our tests were performed with an implementation in BORLAND C++ as an easywin application. Let the truncated chain lattice of an ordered set be the semi-lattice of all nonempty chains. For example, we tested all ordered sets given in [12,13] and their truncated chain lattices (except the truncated chain lattice of P^{3323} in [13], for which our implementation ran out of memory for the Xia graph) for fixed-point-free maps with Xia's algorithm. We also enumerated the fixed-point-free order-preserving maps of crowns with up to 36 elements using Xia's algorithm. In all cases Xia's algorithm only left the edges which were part of r-cliques, i.e., it erased all edges that it could have possibly erased. On a Pentium-75 PC for ordered sets with less than 50 elements the algorithm never took longer than 5 min. In no test for larger ordered sets (between 50 and 90 elements) did the run-time exceed two hours.

Several hundred random tests with the graph rigidity condition were performed for graphs with between 20 and 50 vertices. Either we were able to determine that Xia's algorithm erased all edges except for the edges that were part of r-cliques or a brute-force depth-first-search showed the graph had $\geqslant 300$ automorphisms within less than 10 s of the termination of the reduction. The complete code can be found in [3] and is also available on disk from the authors.

These results are encouraging, since if Xia's algorithm indeed removed all edges that are not part of an r-clique in a large number of instances for a given problem (as seems to be the case for fixed-point-free maps or graph rigidity), then it will for such instances of large enough size be faster than any backtracking method, since it is polynomial.

Run-time comparisons with other algorithms will have to take the specific implementations into account and will need to compare how the 'overhead' generated by a Xia reduction fares versus, say, a very refined backtracking algorithm or an algorithm that is adapted to the intrinsic properties of a problem. After the writing of this paper and of [3] a new way to perform Xia's algorithm was found, which should substantially speed up the algorithm and also fit the Xia graphs into less memory than in the current implementation as a character array. Comparisons between such a more efficient implementation of Xia's algorithm (which is currently under way) and proven algorithms for specific problems will be the subject of further research.

Further refinements of the algorithm inspired by looking at a particular problem will have effects on the performance for all related problems: For example, using the coloring of the vertices induced by projecting on the second component (inspired by looking at isomorphism/rigidity in Example 3.6) can thus also be applied to all other problems in which one looks at injective maps, most notably the problem of finding Hamiltonian cycles.

Another refinement that could be implemented is eliminating edges $\{x, y\}$ that are not part of a K_n with vertex colors $\{\gamma(x), \gamma(y), c_3, c_4, \dots, c_n\}$. However, this would increase the time requirement significantly.

4. Conclusion

Xia configurations link many interesting decision problems to the NP-complete problem of finding an r-clique in a vertex r-colored graph. Xia's algorithm is a polynomial pre-processor for the search for an r-clique, which is highly efficient for investigating the fixed point property and shows great promise for the isomorphism problem. Known criteria that help restrict the search time in depth-first-search (such as only searching for maps that map minimal/maximal elements to minimal/maximal elements for the fixed point problem, or only searching for maps that preserve rank, dual rank and the number of upper/lower bounds and covers for the order-isomorphism problem) can easily be incorporated in the logic requirements. Thus, Xia's algorithm should provide a speed-up even when many conditions to accelerate depth-first-search are known.

While the efficiency of the algorithm depends on the problem to which it is applied, its versatility makes it already very useful as a day-to-day first exploration tool. For example, for the 8-queen problem the algorithm does almost no reduction on the graph. However adaptation of the problem to the 'Xia-system' the authors are developing and implementation into it takes less than 10 min using the condition given in the introduction of Section 2. The modifications are only a few lines in the 'logic-part'. Since the problem is small, the depth-first-search after the reduction found the number of solutions immediately (92). Thus Xia's algorithm is in its versatility and ease-of-use similar to the algorithm discussed in [11].

While finding further performance guarantees of the first kind could be guided by existing conditions for a given problem, these results would in last consequence only state that 'the algorithm does not make the following obvious mistakes …'. Performance guarantees of the second kind seem more interesting as they might lead to discovery of new polynomially accessible subclasses of NP-complete problems.

Finally, one could involve logical conditions with k variables, by going from graphs to hypergraphs. Every set of vertices that satisfies a condition with k variables would be a k-simplex. The reduction procedure would eliminate k-simplexes S for which there is a color $c \notin \gamma[S]$ such that for all vertices v of color c, our Xia hypergraph does not contain the boundary of $S \cup \{v\}$. Unfortunately this refinement in the logic increases the memory and time requirements.

Acknowledgements

We would like to thank M. Roddy for invaluable help with Example 2.6, H. Priestley for 'asking the right question at the right time' and the referees of this paper for many helpful remarks which improved the presentation of the paper and for pointing out the reference [11].

References

[1] K. Baclawski, A fixed point algorithm for ordered sets, Proceedings of the Rotafest, 1996, submitted for publication.
[2] K. Baclawski, A. Björner, Fixed points in partially ordered sets, Adv. in Math. 31 (1979) 263–287.
[3] M. Donalies, Implementation and analysis of Xia's algorithm, MS Thesis, Hampton University, Summer 1997.
[4] D. Duffus, T. Goddard, The complexity of the fixed point property, Order 13 (1996) 209–218.
[5] P. Edelman, On a fixed point theorem for partially ordered sets, Discrete Math. 15 (1979) 117–119.
[6] J.D. Farley, The uniqueness of the core, Order 10 (1993) 129–131.
[7] T. Fofanova, I. Rival, A. Rutkowski, Sets of dimension 2 and the fixed point property, Order 13 (1996) 245–253.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[9] Yu I. Manin, A course in mathematical logic, Graduate Texts in Mathematics, vol. 53, Springer, New York, 1977.
[10] T. McKee, E. Prisner, An approach to graph-theoretic homology, Proceedings of the Eighth Quadrennial International Conference in Graph Theory, Combinatorics, Algorithms and Applications, Kalamazoo, MI, 1996, submitted for publication.
[11] H.A. Priestley, M.P. Ward, A multipurpose backtracking algorithm, J. Symbolic Comp. 18 (1994) 1–40.
[12] A. Rutkowski, The fixed point property for small sets, Order 6 (1989) 1–14.
[13] B. Schröder, Fixed point property for 11-element sets, Order 10 (1993) 329–347.
[14] B. Schröder, Algorithms for the fixed point property, Theoretical Computer Science, Proceedings of the 1996 ORDAL Conference, (1996, to appear) currently available at `http://www.csi.uottawa.ca/ordal/papers/schroder/FINSURVE.html`.
[15] B. Schröder, Fixed cliques and generalizations of dismantlability, Proceedings of the Eighth Quadrennial Conference on Graph Theory, Algorithms and Applications, 1997, to appear.
[16] J.W. Walker, Isotone relations and the fixed point property for posets, Discrete Math. 48 (1984) 275–288.

[17] W. Xia, Fixed point property and formal concept analysis, Order 9 (1992) 255–264.
[18] N. Zaguia, Isotone maps: enumeration and structure, in: Sauer, Woodrow, Sands (Eds.), Finite and Infinite Combinatorics in Sets and Logic, NATO ASI Series C 411, Kluwer, Dordrecht (1993) pp. 421–430.

Note added in proof by the second author (BS). In early 1999 the second author realized the connection between the subject of this paper and constraint networks in computer science. What is presented here as Xia graphs is also known there as the expanded constraint network, though it seems they have not seen extensive use. A resource that gives an overview of the area of constraint satisfaction is E. Tsang's book 'Foundations of Constraint Satisfaction' (Academic Press Series on Computation in Cognitive Science, San Diego, 1993). As Tsang states in his foreword, works on constraint satisfaction are scattered and there is a great diversity in terminology. As all results in this paper are proved in graph-theoretical language I decided to leave the paper in its refereed and accepted form and to add this note. Many of the basics on Xia graphs are rediscoveries of results in constraint networks. The type of performance guarantees presented here and their applications to the fixed point property and order isomorphism are new. I hope that the results presented here and this note will direct the attention of other researchers in Discrete Mathematics to this fascinating area. The potential for cross-fertilization is great. Application of results in constraint satisfaction to order-theoretical problems should allow order theorists to solve problems more efficiently, while the use of specific insights in order-theoretical problems can lead to new results about the performance of algorithms. A first example is that in later tests Xia's algorithm (called path-consistency in constraint satisfaction) compared favorably with [11].