Contraction Degeneracy on Cographs

Hans L. Bodlaender and Thomas Wolle

institute of information and computing sciences, utrecht university

technical report UU-CS-2004-031

www.cs.uu.nl

Contraction Degeneracy on Cographs

Hans L. Bodlaender and Thomas Wolle

Institute of Information and Computing Sciences, Utrecht University
P.O.Box 80.089, 3508 TB Utrecht, The Netherlands
hansb@cs.uu.nl thomasw@cs.uu.nl

Abstract. The contraction degeneracy of a graph G is the maximum minimum degree of G' over all minors G' of G. The corresponding decision problem is known to be NP-complete. In this paper, we present a dynamic programming approach for computing the contraction degeneracy of cographs.

1 Introduction

Contracting an edge in a graph is the operation that introduces a new vertex and new edges such that the new vertex is adjacent to all the neighbours of the endpoints of the contracted edge, and then we delete the endpoints of the contracted edge and all their incident edges. Contracting edges has been shown to be of great use for obtaining new lower bound heuristics for treewidth [2]. The new parameter *contraction degeneracy* arose from this research. However, it is an interesting topic in its own right and not only as a treewidth lower bound. The minimum degree of a graph is a lower bound for its treewidth (see [9, 1]), as is the maximum minimum degree (also called *degeneracy*) over all subgraphs [8]. Instead of deleting vertices, we can use edge contraction, which leads to *contraction degeneracy* first defined in [2]. The according decision problem is NP-complete [2]. Thus, it is interesting to look for special graph classes where the contraction degeneracy can be computed in polynomial time. So far, little is known about this. In this paper, we consider the contraction degeneracy problem for the class of cographs. It should be noted that the treewidth of cographs is polynomial time computable [3], and thus our algorithm, while of independent interest, is not necessary to obtain lower bounds for treewidth of given cographs.

Cographs (also called *complement reducible* graphs) are graphs with a special structure. They are perfect and form a proper subset of the permutation graphs. Cographs can be defined recursively using two operations (disjoint union and product, see Section 2.3) on singletons. Another characterisation is that cographs are exactly those graphs that do not contain a P_4 as an induced subgraph. The recursive definition enables a cograph being represented by an expression, and in turn, being represented by a tree, from which we can derive the cotree of the graph [5]. Very often, a dynamic programming approach can be applied to such a cotree, enabling efficient algorithms on cographs for problems that are NP-hard on general graphs.

The paper is organised as follows. In Section 2, we fix some terminology and give more detailed information about edge contractions (Section 2.1), the contraction degeneracy (Section 2.2) and cographs (Section 2.3). Section 3 presents our dynamic programming method. The lemmas showing the correctness of our approach are given in Section 3.1 and 3.2, followed by the main result in Section 3.3. The paper is closed with some remarks in Section 4.

2 Preliminaries

Throughout the paper $G = (V, E)$ denotes a simple undirected graph. Most of our terminology is standard graph theory/algorithm terminology. The open neighbourhood $N_G(v)$ or simply $N(v)$ of a vertex $v \in V$ is the set of vertices adjacent to v in G. As usual, the degree in G of vertex v is $d_G(v)$ or simply $d(v)$, and we have $d(v) = |N(v)|$. $N(S)$ for $S \subseteq V$ denotes the open neighbourhood of S, i.e. $N(S) = \bigcup_{s \in S} N(s) \setminus S$. We denote the minimum degree of a vertex in G:

$$\delta(G) := \min_{v \in V} d(v)$$

After deleting vertices of a graph and their incident edges, we get an *induced subgraph.* A *subgraph* is obtained, if we additionally allow deletion of edges. If we furthermore allow edge-contractions (see Section 2.1), we get a *minor.* We explicitely exclude the null graph (the empty graph on 0 vertices), as a subgraph or minor of a graph. It is known that the treewidth of a minor of G is at most the treewidth of G (see e.g. [1]). A vertex v is *universal* in G, if v is adjacent to each vertex $w \in V$,$w \neq v$.

2.1 Edge Contraction

A more formal approach to edge contractions as well as basic lemmas, which are summarised here, can be found in [10]. Contracting edge $e = \{u, v\}$ in the graph $G = (V, E)$, denoted as G/e, is the operation that introduces a new vertex a_e and new edges such that a_e is adjacent to all the neighbours of u and v and delete vertices u and v and all edges incident to u or v:

$$\begin{aligned} G/e &:= (V', E'), \text{ where} \\ V' &= \{a_e\} \cup V \setminus \{u, v\} \\ E' &= \{\ \{a_e, x\} \mid x \in N(\{u, v\})\} \cup E \setminus \{e' \in E \mid e' \cap e \neq \emptyset\} \end{aligned}$$

Contracting all edges of a cycle in a graph means that the last contraction is unnecessary. Hence, a *contraction-set* is a cycle free set $E' \subseteq E(G)$ of edges. Let be $e, f \in E$. Contracting edges is commutative, i.e. $(G/e)/f$ and $(G/f)/e$ are isomorphic graphs. For a contraction-set $E' = \{e_1, e_2, \dots, e_p\}$, we define $G/E' := G/e_1/e_2/\dots/e_p$. A *contraction* H of G is a graph such that there exists a contraction-set E' with: $H = G/E'$. An edge contraction might decrease the degree of a vertex, but it can never decrease it by more than one.

Note that after each single edge contraction the names of the vertices are updated in the graph. Hence, for two adjacent edges $e = \{u, v\}$ and $f = \{v, w\}$, edge f will be different after contracting edge e, namely in G/e we have $f = \{a_e, w\}$. Using the old names of vertices and edges is formally not correct. However, it is unambiguous which new vertex or edge is meant. Furthermore, using the old names might increase understandability and readability. Thus, we can agree on f representing the same edge in G and in G/e. The same applies also to vertices.

2.2 Contraction Degeneracy

In this section, we define the parameter δC – the contraction degeneracy. A few basic statements on the contraction degeneracy can be found in [10]. Heuristics and experimental evaluations of these for the contraction degeneracy are given in [2].

The degeneracy δD of a graph is the maximum over all its subgraphs of the minimum degree of the subgraph.

$$\delta D(G) := \max_{G'}\{\delta(G') \mid G' \text{ is a subgraph of } G\}$$

The parameter δD is known to be a lower bound for treewidth [8]. Instead of deleting vertices and edges while recording the minimum degree of the occurring subgraphs, contracting edges experimentally proved to be a very good idea for better treewidth lower bounds [2, 7]. Inspired by this lower bound, the parameter δC and the according decision problem were defined.

Definition 1. *The* contraction degeneracy *δC of a graph G is defined as follows:*

$$\delta C(G) := \max_{G'}\{\delta(G') \mid G' \text{ is a minor of } G\}$$

Note that $\delta C(G)$ is defined as the maximum over all minors G' of G of the minimum degree of G'. Using contractions instead of minors in this definition does not lead to an equivalent notion, unless the considered graph G is connected. If G is not connected, it might be necessary to delete one or more connected components, to obtain a minor G' of G with maximum minimum degree. The corresponding decision problem is formulated as usual:

Problem: CONTRACTION DEGENERACY
Instance: Graph $G = (V, E)$ and integer $k \geq 0$.
Question: Is the contraction degeneracy of G at least k?

CONTRACTION DEGENERACY is NP-complete, even for bipartite graphs [2]. Therefore, it is interesting to look at polynomial time computable special cases. Hence, in Section 3 of this paper, we look at contraction degeneracy for cographs.

2.3 Cographs

Cographs can be represented by a tree, enabling dynamic programming approaches on the tree. There are several equivalent definitions of cographs. One is that cographs are exactly those graphs that do not contain a P_4 (a path with four vertices) as an induced subgraph [5]. An alternate, equivalent definition requires two operations.

Definition 2. *Let be given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $V_1 \cap V_2 = \emptyset$.*

- *The* disjoint union *of G_1 and G_2 is*
  $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.
- *The* product *of G_1 and G_2 is*
  $G_1 \times G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{\{v_1, v_2\} : v_1 \in V_1 \wedge v_2 \in V_2\})$.

Note that the operations $\cup$ and $\times$ are commutative and associative. Hence, the result of a sequence of equal operations is well defined, and such a sequence can easily be converted into a tree structure using only the binary versions of these operations.

Definition 3. *A graph G is a cograph if and only if one of the following holds:*

- $|V| = 1$
- $G = G_1 \cup G_2 \cup ... \cup G_p$ *for cographs* $G_1, ..., G_p$.
- $G = G_1 \times G_2 \times ... \times G_p$ *for cographs* $G_1, ..., G_p$.

A consequence from this definition is that a cograph can be represented as an expression using the operations $\cup$ and $\times$ on singletons. Such an expression can be represented by a tree from which we can derive the *cotree* of the cograph.

The cotree T_G of G is a labelled tree. Leaves of the cotree are in a one-to-one correspondence with the vertices of the graph. Internal nodes of the cotree are labelled with either '0' or '1'. To each node of the cotree, we can associate a cograph in the following manner: Each leaf of the cotree represents a graph with a single vertex, hence a cograph. A 0-node represents a cograph that is the disjoint union of the cographs corresponding to the children of the 0-node, and a 1-node represents a cograph that is the product of the cographs corresponding to the children of that 1-node. Note that there is an edge between two vertices v and w if and only if the lowest common ancestor of v and w in the cotree is a 1-node. There are linear time algorithms for recognising cographs and building the cotree [6, 4].

In Definition 3, we can restrict p to be 2. As a consequence, a cotree is a binary tree. This is very helpful for formulating dynamic programming algorithms on trees. The size of such a binary cotree T_G of cograph G is linear in the size of G, because T_G has exactly $n = |V(G)|$ leaves, and after at most $O(n)$ nonempty disjoint union or product operations, we obtain the graph G. Therefore, we have at most $O(n)$ internal nodes in T_G. In the following, we assume that the cotree is a binary cotree, i.e. 'cotree' refers to 'binary cotree'. Clearly, the contraction degeneracy of a disconnected graph is the minimum contraction degeneracy of its connected components. Thus, we may assume that the given cograph is connected, and hence, the root of the cotree is a 1-node.

3 Contraction Degeneracy on Cographs

In this section, we present a dynamic programming method for computing the contraction degeneracy of a cograph G. We assume that we are given G with a binary cotree T (otherwise, a cotree can be build in linear time [6, 4], from which a binary cotree can be easily constructed). The special structure of a binary cotree makes it easier to present a dynamic programming algorithm. As already described earlier, cotrees have two kind of internal nodes: 1-nodes and 0-nodes. Every node i represents a subgraph G_i of the input graph G. Each internal node i of a binary cotree has exactly two children j_1 and j_2. When considering a 0-node i of a cotree, the graph G_i is simply the disjoint union of the two graphs G_{j_1} and G_{j_2}. Consequently, all edges that can be contracted in G_i can either be contracted in G_{j_1} or in G_{j_2}. However, if i is an 1-node, new edges are created between vertices of G_{j_1} and vertices of G_{j_2}. Hence, not all edges that can be contracted in G_i can be contracted in G_{j_1} or in G_{j_2}. Consider some fixed internal node i of T with children j_1 and j_2. We use the following terminology:

- G_i is the graph corresponding to node i, i.e. $G_i = G_{j_1} \cup G_{j_2}$ if i is a 0-node, and $G_i = G_{j_1} \times G_{j_2}$ if i is a 1-node.
- $V_i = V(G_i)$, $V_{j_1} = V(G_{j_1})$, $V_{j_2} = V(G_{j_2})$.
- $E_i = E(G_i)$, $E_{j_1} = E(G_{j_1})$, $E_{j_2} = E(G_{j_2})$.
- $n_i = |V_i|$.
- $e \in E_i$ is called a *cross edge* if i is a 1-node and $e \notin E_{j_1} \cup E_{j_2}$.
- $G \setminus V' = G[V \setminus V']$ for a graph $G = (V, E)$ and $V' \subseteq V$.

The following lemma is easy to see, but it is an important part of the correctness proof of our approach.

Lemma 1. *Contracting a cross edge of G_i creates a universal vertex in G_i.*

Proof. Given a cross edge $e = \{u, v\}$, $u \in V_{j_1}$ and $v \in V_{j_2}$, we know by definition of a 1-node, that u is adjacent to all $w \in V_{j_2}$ in G_i. Also, v is adjacent to all $w \in V_{j_1}$ in G_i. When contracting e, we get a vertex, adjacent to all $w \in V_i \setminus e$ in G_i. □

In our dynamic programming approach, we associate to each node i, i.e. to each subgraph G_i, a function

$$F_i : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$

For nonnegative integers x and y with $x + y \leq n_i$, $x \leq n_i - 1$, the function F_i is defined in the following way:

$$\begin{aligned} F_i(x, y) = \max\{\delta(H) \mid\ & H \text{ can be obtained from } G_i \text{ by} \\ & \text{contracting the edges of a contraction-set of size } x \\ & \text{and then deleting } y \text{ vertices and their incident edges } \} \end{aligned}$$

A table T_i with $O(n_i^2)$ entries can be used to store the values of F_i. The table contains a tuple for every possible x-y-combination. Each tuple consists of three integers (x, y, z), with $z = F_i(x, y)$.

The motivation behind storing the maximum minimum degree after x edge contractions and y vertex deletions is as follows. When considering a graph G_i, we can contract edges in $E(G_i)$ to increase the minimum degree. We also must consider the possibility that in order to obtain the maximum minimum degree of G, i.e. to solve the contraction degeneracy problem on G, it might be necessary to contract some edges not in $E(G_i)$ but in $E(G_h)$ where h is a 1-node in the cotree that is on the path from i to the root r of the cotree. Deleting a vertex can decrease the minimum degree. However, we do not delete vertices from the graph G_i, we only 'deactivate' them, such that they have no influence on the degrees at this moment. In a later stage, deactivated vertices might be used for contracting cross edges, which results in universal vertices (see Lemma 1). The consequences of the introduction of a universal vertex are easily computable, since the degree of each vertex is increased by one. That is the reason why we first 'deactivate' some vertices and later using them again (not explicitely). In the next lemma, we see that we indeed can compute the contraction degeneracy of a cograph by using function F_i.

Lemma 2. *Given the function F_i defined above for node i of a binary cotree, we can solve the contraction degeneracy problem for G_i in $O(n_i)$ time.*

Proof. To solve the problem on G_i means that we can only use contractions of edges in G_i to increase the minimum degree, and we must not delete any vertices. For every number x of contracted edges in G_i, and every number y of deleted vertices, we have the maximum minimum degree. For the contraction degeneracy problem, we only need these values with $y = 0$. Clearly, we have:

$$\delta C(G_i) = \max_{x=0,\ldots,n_i-1} F_i(x,0)$$

□

Hence, if we have the function F_r for the root r of the cotree, i.e. $G_r = G$, we can compute the contraction degeneracy of G in $O(n)$ additional time. Now we have seen that the functions F_i are sufficient to solve the problem, we look at how to compute these for each node i, using the functions for the children of i. As giving such a function for a leaf node i is trivial (there are just two values $F_i(0,0) = 0$ and $F_i(0,1) = \infty$), we describe the methods to compute the values of these functions for 0-nodes and for 1-nodes.

3.1 A Recurrence Relation for 0-nodes

In this section, we give the recurrence relation for a 0-node i with children j_1 and j_2, i.e. we present and prove the correctness of the recursive formula to compute F_i using the already computed values of functions F_{j_1} and F_{j_2}.

Lemma 3. *Let i be a 0-node with children j_1 and j_2, x and y nonnegative integers with $x+y \leq n_i$, $x \leq n_i - 1$. Then, we have:*

$$\begin{aligned} F_i(x,y) = \max \{ \; & \min(F_{j_1}(x_1,y_1), F_{j_2}(x_2,y_2)) \mid \\ & x_1, x_2, y_1, y_2 \in I\!N \wedge x_1 + x_2 = x \wedge y_1 + y_2 = y \wedge \\ & x_1 \leq n_{j_1} - 1 \wedge x_2 \leq n_{j_2} - 1 \; \} \end{aligned}$$

Proof. Let x and y be fixed nonnegative integers with $x + y \leq n_i$, $x \leq n_i - 1$. First, we will prove that $F_i(x,y)$ is at least the stated expression.

Claim. Let x_1, y_1, x_2, and y_2 be fixed integers with $x_1 + x_2 = x$, $y_1 + y_2 = y$, $x_1 \leq n_{j_1} - 1$ and $x_2 \leq n_{j_2} - 1$. Then $F_i(x,y) \geq \min(F_{j_1}(x_1,y_1), F_{j_2}(x_2,y_2))$.

Proof. Let E_1 be the contraction-set with $|E_1| = x_1$, and let V_1 be the vertex set with $|V_1| = y_1$, such that: $F_{j_1}(x_1,y_1) = \delta((G_{j_1}/E_1) \setminus V_1)$. E_2 and V_2 are defined similarly.

Now we can contract in G_i the contraction-set $E_1 \cup E_2$, and then delete all vertices in $V_1 \cup V_2$. Since $F_i(x,y)$ is defined to be a maximum value, and $(G_i/(E_1 \cup E_2) \setminus (V_1 \cup V_2)$ is a possible minor (see the definition of F_i), we have: $F_i(x,y) \geq \delta((G_i/(E_1 \cup E_2)) \setminus (V_1 \cup V_2)) = \min(F_{j_1}(x_1,y_1), F_{j_2}(x_2,y_2))$. ◇

As the claim holds for all nonnegative integers x_1, y_1, x_2, and y_2 with $x_1 + x_2 = x$, $y_1 + y_2 = y$, $x_1 \leq n_{j_1} - 1$ and $x_2 \leq n_{j_2} - 1$, we have:

$$\begin{aligned} F_i(x,y) \geq \max \{ \; & \min(F_{j_1}(x_1,y_1), F_{j_2}(x_2,y_2)) \mid \\ & x_1, x_2, y_1, y_2 \in I\!N \wedge x_1 + x_2 = x \wedge y_1 + y_2 = y \\ & x_1 \leq n_{j_1} - 1 \wedge x_2 \leq n_{j_2} - 1 \; \} \end{aligned}$$

Now we show that $F_i(x,y)$ is at most the stated expression. Consider a contraction-set $E' \subseteq E$ with $|E'| = x$, and a vertex set $V' \subseteq V$ with $|V'| = y$, such that

$$F_i(x,y) = \delta((G_i/E') \setminus V')$$

Note that i is a 0-node, and hence, each edge in E_i belongs to E_{j_1} or E_{j_2}. E' and V' can be partitioned in the following way:

$$\begin{array}{llll} E_1 := E_{j_1} \cap E' & E_2 := E_{j_2} \cap E' & V_1 := V_{j_1} \cap V' & V_2 := V_{j_2} \cap V' \\ x_1 := |E_1| & x_2 := |E_2| & y_1 := |V_1| & y_2 := |V_2| \end{array}$$

Claim. It holds: $x_1 + x_2 = x$, $y_1 + y_2 = y$, $x_1 \leq n_{j_1} - 1$ and $x_2 \leq n_{j_2} - 1$.

Proof. From the partition of E' and V' directly follows $x_1 + x_2 = x$ and $y_1 + y_2 = y$. Each contraction-set is a forest, and hence, each contraction-set has at most $n - 1$ edges, where n is the number of vertices in the considered graph. E_1 is a contraction-set, since $E_1 \subseteq E'$. Therefore, E_1 is a contraction-set for G_{j_1}. Thus, $x_1 \leq n_{j_1} - 1$ and $x_2 \leq n_{j_2} - 1$. ◇

We have $G_i = G_{j_1} \cup G_{j_2}$ and by definition of E_1, E_2, V_1 and V_2, the following is easy to see. $(G_i/E') \setminus V' = (G_{j_1}/E_1) \setminus V_1 \cup (G_{j_2}/E_2) \setminus V_2$, and therefore, we have:

$$F_i(x, y) = \delta((G_i/E') \setminus V') = \min(\delta((G_{j_1}/E_1) \setminus V_1), \delta((G_{j_2}/E_2) \setminus V_2))$$

So, $F_i(x, y) \leq \delta((G_{j_1}/E_1) \setminus V_1) \leq F_{j_1}(x_1, y_1)$. Similarly, $F_i(x, y) \leq F_{j_2}(x_2, y_2)$, and hence, $F_i(x, y) \leq \min(F_{j_1}(x_1, y_1), F_{j_2}(x_2, y_2))$. □

3.2 A Recurrence Relation for 1-Nodes

This section is devoted to the recurrence relation for a 1-node i with children j_1 and j_2. However, before we present and prove the correctness of the recursive formula to compute F_i (using the already computed values of functions F_{j_1} and F_{j_2}), we introduce a modified version F'_i of F_i. The function F'_i is defined especially for 1-nodes. The advantage is that it is easier and faster to compute. Later, we will show that we can compute F_i using function F'_i. Let i be a 1-node with associated graph $G_i = (V_i, E_i)$, and let $E^c \subseteq E_i$ be the set of cross edges of G_i.

$$F'_i : I\!N \times I\!N \longrightarrow I\!N$$

For nonnegative integers x and y with $x + y \leq n_i$ and $x \leq n_i - 1$, the function F'_i is defined in the following way:

$$\begin{aligned} F'_i(x, y) = \max\{\delta(H) \mid\ & H \text{ can be obtained from } G_i \text{ by} \\ & \text{contracting the edges of a contraction-set } E' \text{ of size } x \\ & \text{and then deleting } y \text{ vertices and their incident edges,} \\ & \text{where } \forall e, f \in E^c \cap E' : e \cap f = \emptyset \ \} \end{aligned}$$

The last requirement in the definition of F'_i says that all cross edges in E' are pairwise non-adjacent. Hence, every cross edge in E' creates a unique universal vertex. We first give a number of lemmas which will be used later. For Lemmas 4, 5, and 6, suppose the following is given. The graph $G_i = G_{j_1} \times G_{j_2}$ is the product of G_{j_1} and G_{j_2}. Furthermore, let be given contraction-sets $E' \subseteq E_i$, $|E'| = x$, $E_1 \subseteq E_{j_1}$ and $E_2 \subseteq E_{j_2}$, vertex sets $V' \subseteq V_i$, $|V'| = y$, $V_1 \subseteq V_{j_1}$, $V_2 \subseteq V_{j_2}$, with $E_1 = E' \cap E_{j_1}$, $|E_1| = x_1$, $E_2 = E' \cap E_{j_2}$, $|E_2| = x_2$, $E^* = E' \setminus (E_1 \cup E_2)$, $|E^*| = x^*$, and E^* does not contain any two adjacent edges, i.e. $\forall e, f \in E^* : e \cap f = \emptyset$. Furthermore, let be $V^* = \bigcup_{e \in E^*} e$, $V_1 = (V_{j_1} \cap V') \cup V^*$, $|V_1| = y_1$, $V_2 = (V_{j_2} \cap V') \cup V^*$, $|V_2| = y_2$, and $y_1 \geq x^*$, $y_2 \geq x^*$. Note that E^* is the set of cross edges that is contracted, and V^* is the set of endpoints of edges in E^*.

Lemma 4. *Each of the following holds:*

1. $|V(G_i/E' \setminus V')| = n_i - x - y$.
2. $x + y = n_i \Longrightarrow F_i(x, y) = \infty$.
3. $x + y < n_i \Longrightarrow F_i(x, y) < n_i - x - y$.

Proof. (1.) Note that G_i has n_i vertices. Since E' is a contraction-set, every contraction of an edge in E' results in a decrease of the number of vertices by one. Hence, $|V(G_i/E')| = n_i - x$. Now, we have to delete y vertices in G_i/E'. Clearly, $|V(G_i/E' \setminus V')| = n_i - x - y$.

(2.) From *(1.)*, we can conclude that $G_i/E' \setminus V'$ is the null graph (the empty graph with 0 vertices). Hence, the minimum degree of a vertex over an empty vertex set is the minimum of an empty set of integers, i.e. it is ∞.

(3.) Since $G_i/E' \setminus V'$ has $n_i - x - y$ vertices, the maximum degree of a vertex in $G_i/E' \setminus V'$ is at most $n_i - x - y - 1$. As this holds for every E' and V' with $|E'| = x$, $|V'| = y$, we have $F_i(x, y) \leq n_i - x - y - 1$. □

Note that we explicitely excluded the null graph as a minor and as a subgraph of a graph. However, we will use the minimum degree of the null graph in our recurrence relations, i.e. we need statement *(2.)* of Lemma 4.

Lemma 5. *Let $v_1 \in V(G_{j_1}/E_1 \setminus V_1)$. Then we have:*

$$d_{G_i/E' \setminus V'}(v_1) = d_{G_{j_1}/E_1 \setminus V_1}(v_1) + n_{j_2} - x_2 - y_2 + x^*$$

Proof. Since E_1 and V_1 only change the internal structure and number of vertices in G_{j_1}, we have: $d_{(G_{j_1} \times G_{j_2})/E_1 \setminus V_1}(v_1) = d_{G_{j_1}/E_1 \setminus V_1}(v_1) + n_{j_2}$. With the same argument that E_2 and V_2 only affect G_{j_2}, we have: $d_{(G_{j_1} \times G_{j_2})/(E_1 \cup E_2) \setminus (V_1 \cup V_2)}(v_1) = d_{G_{j_1}/E_1 \setminus V_1}(v_1) + n_{j_2} - x_2 - y_2$ (see *(1.)* in Lemma 4). However, instead of deleting all vertices in $V_1 \cup V_2$, we contract x^* non-adjacent edges with one endpoint in V_1 and the other endpoint in V_2, i.e. we contract all cross edges in E^*. It is easy to see that this results in x^* additional universal vertices (see Lemma 1). Hence, we have:

$$d_{(G_{j_1} \times G_{j_2})/(E_1 \cup E_2 \cup E^*) \setminus [(V_1 \cup V_2) \setminus V^*]}(v_1) = d_{G_{j_1}/E_1 \setminus V_1}(v_1) + n_{j_2} - x_2 - y_2 + x^*$$

□

Lemma 6. *Let $v_1, v_2 \in V(G_{j_1}/E_1 \setminus V_1)$. Then we have:*

$$d_{G_{j_1}/E_1 \setminus V_1}(v_1) \leq d_{G_{j_1}/E_1 \setminus V_1}(v_2) \Longleftrightarrow d_{G_i/E' \setminus V'}(v_1) \leq d_{G_i/E' \setminus V'}(v_2)$$

Proof. This follows directly from Lemma 5. □

Lemma 7. *If $x + y = n_i$, then $F'_i(x, y) = \infty$.*

Proof. This is similar to Lemma 4*(2)*. □

Now in Lemmas 8 and 9, we give the recurrence relation for 1-nodes.

Lemma 8. *Let i be a 1-node with children j_1 and j_2. Let x and y be nonnegative integers with $x + y \leq n_i$, $x \leq n_i - 1$. Then, we have:*

$$\begin{aligned} F'_i(x, y) = \max\{ \min(&F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*, \\ &F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^* \ , \\ &n_i - x - y - 1) \mid \\ &x_1, x_2, y_1, y_2, x^* \in I\!N \wedge \\ &x_1 + x_2 + x^* = x \wedge y_1 + y_2 = y + 2 \cdot x^* \wedge \\ &y_1 \geq x^* \wedge y_2 \geq x^* \wedge \\ &x_1 + y_1 \leq n_{j_1} \wedge x_2 + y_2 \leq n_{j_2} \\ &x_1 \leq n_{j_1} \wedge x_2 \leq n_{j_2} \ \} \end{aligned}$$

Proof. Let x and y be fixed nonnegative integers with $x + y \leq n_i$, $x \leq n_i - 1$. We will first prove that $F'_i(x, y)$ is at least the stated expression. Therefore, let x_1, x_2, y_1, y_2, and x^* be fixed nonnegative integers fulfilling the requirements stated in the lemma. We will first prove:

$$\begin{aligned} F'_i(x, y) \geq \min(&F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*, \\ &F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^*, \\ &n_i - x - y - 1 \ ) \end{aligned}$$

Let E_1 and E_2 be contraction-sets with $|E_1| = x_1$ and $|E_2| = x_2$ and V_1 and V_2 be set of vertices with $|V_1| = y_1$ and $|V_2| = y_2$, such that $F_{j_1}(x_1, y_1) = \delta(G_{j_1}/E_1 \setminus V_1)$ and $F_{j_2}(x_2, y_2) = \delta(G_{j_2}/E_2 \setminus V_2)$. Let E^* be a set of x^* pairwise non-adjacent cross edges, each of them having one endpoint in V_1 and the other in V_2, and let be $V^* = \bigcup_{e \in E^*} e$. As $|V_1| = y_1 \geq x^*$ and $|V_2| = y_2 \geq x^*$, and each vertex in V_1 is adjacent to each vertex in V_2, we can choose such a set E^*. We define: $E' := E_1 \cup E_2 \cup E^*$ and $V' := (V_1 \cup V_2) \setminus V^*$.

Claim. $|E'| = x$ and $|V'| = y$.

Proof. This follows from the definition of the corresponding sets. Note that E_1, E_2 and E^* are pairwise disjoint. Furthermore, observe that V_1 and V_2 are disjoint and $V^* \subseteq (V_1 \cup V_2)$. Hence, $|V'| = |V_1| + |V_2| - |V^*| = y_1 + y_2 - 2 \cdot x^*$. ⋄

Note that $V(G_i/E' \setminus V')$ can be partitioned into three disjoint sets: $W_1 := V(G_{j_1}/E_1 \setminus V_1)$, $W_2 := V(G_{j_2}/E_2 \setminus V_2)$, and the vertices that result from contracting a cross edge $W^* := V(G_i/E' \setminus V') \setminus (W_1 \cup W_2)$.

If W_1 is empty, then $x_1 + y_1 = n_{j_1}$ and $F_{j_1}(x_1, y_1) = \infty$. Otherwise, let v_1 be a vertex in $G_{j_1}/E_1 \setminus V_1$ of minimum degree, i.e. $d_{G_{j_1}/E_1 \setminus V_1}(v_1) = F_{j_1}(x_1, y_1)$. From Lemma 6, we know that all other vertices in $G_{j_1}/E_1 \setminus V_1$ have degree in $G_i/E' \setminus V'$ as least as large as the degree of v_1 in $G_i/E' \setminus V'$. So, by Lemma 5 all vertices in W_1 have degree at least $d_{G_{j_1}/E_1 \setminus V_1}(v_1) + n_{j_2} - x_2 - y_2 + x^* \geq F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*$. Similarly, either W_2 is empty, in which case $F_{j_2}(x_2, y_2) = \infty$, or all vertices in W_2 have degree at least $F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^*$.

Vertices that are the result of a contraction of a cross edge are universal in $G_i/E' \setminus V'$, and hence have degree $n_i - x - y - 1$.

We can conclude that

$$\begin{aligned} F'_i(x, y) &\geq \delta(G_i/E' \setminus V') \\ &\geq \min(\; F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*, \\ &\qquad\quad F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^*, \\ &\qquad\quad n_i - x - y - 1) \end{aligned}$$

Since this holds for all x_1, x_2, y_1, y_2 and x^*, fulfilling the conditions as in the lemma, we can conclude:

$$\begin{aligned} F'_i(x, y) \geq \max\{ \min(\; & F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*, \\ & F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^*, \\ & n_i - x - y - 1) \mid \\ & x_1, x_2, y_1, y_2, x^* \in I\!N \wedge \\ & x_1 + x_2 + x^* = x \wedge y_1 + y_2 = y + 2 \cdot x^* \wedge \\ & y_1 \geq x^* \wedge y_2 \geq x^* \wedge \\ & x_1 + y_1 \leq n_{j_1} \wedge x_2 + y_2 \leq n_{j_2} \wedge \\ & x_1 \leq n_{j_1} \wedge x_2 \leq n_{j_2} \; \} \end{aligned}$$

We now show the equality, i.e. we show that $F'_i(x, y)$ is at most the given expression. Let E' be a contraction-set of size x containing no two adjacent cross edges, and V' be a vertex set of size y such that $F'_i(x, y) = \delta(G_i/E' \setminus V')$. E' and V' can be partitioned in the following way:

$$\begin{array}{lll} E_1 := E_{j_1} \cap E' & E_2 := E_{j_2} \cap E' & E^* := E' \setminus (E_1 \cup E_2) \\ V_1 := V_{j_1} \cap (V' \cup V^*) & V_2 := V_{j_2} \cap (V' \cup V^*) & V^* := \bigcup_{e \in E^*} e \\ x_1 := |E_1| & x_2 := |E_2| & x^* := |E^*| \\ y_1 := |V_1| & y_2 := |V_2| & \end{array}$$

Claim. $x = x_1 + x_2 + x^*$, $y_1 + y_2 = y + 2 \cdot x^*$, $y_1 \geq x^*$, $y_2 \geq x^*$, $x_1 + y_1 \leq n_{j_1}$, $x_2 + y_2 \leq n_{j_2}$, $x_1 \leq n_{j_1}$, $x_2 \leq n_{j_2}$.

Proof. The first equality follows directly from the corresponding definitions.

Note that E^* is a contraction-set containing all cross edges of E' (they are pairwise non-adjacent). Hence, $|V^*| = 2 \cdot x^*$. Furthermore, observe that V_{j_1} and V_{j_2} are disjoint and $V_{j_1} \cup V_{j_2} = V_i \supseteq (V' \cup V^*)$. Therefore, $y_1 + y_2 = |V_1| + |V_2| = |V_{j_1} \cap (V' \cup V^*) \cup V_{j_2} \cap (V' \cup V^*)| = |V_i \cap (V' \cup V^*)| = |V' \cup V^*| = y + 2 \cdot x^*$.

The inequality $y_1 \geq x^*$ follows from $y_1 = |V_1| = |V_{j_1} \cap (V' \cup V^*)| \geq |V_{j_1} \cap V^*| = |E^*| = x^*$. $|V_{j_1} \cap V^*| = |E^*|$ can be seen from the fact that E^* contains exactly x^* non-adjacent cross edges that have one endpoint in V_{j_1} and the other endpoint in V_{j_2}.

Since E_1 is the subset of E', restricted to G_{j_1}, E_1 is a contraction-set of G_{j_1}, and V_1 is a subset of vertices in G_{j_1}/E_1, since V' is a subset of vertices in G_i/E'. Hence, $x_1 + y_1 \leq n_{j_1}$ and $x_1 \leq n_{j_1}$.

In the same way, we can conclude $y_2 \geq x^*$, $x_2 + y_2 \leq n_{j_2}$ and $x_2 \leq n_{j_2}$. ◇

Similar as above, write $W_1 = V(G_{j_1}/E_1 \setminus V_1)$, $W_2 = V(G_{j_2}/E_2 \setminus V_2)$ and $W^* = V(G_i/E' \setminus V') \setminus (W_1 \cup W_2)$ the set of vertices resulting from contracting a cross edge. Note that W_1, W_2 and W^* partition $V(G_i/E' \setminus V')$. We consider four different cases.

Case 1 '$W_1 = \emptyset$ and $W_2 = \emptyset$': In this case, $G_i/E' \setminus V'$ is a clique with x^* vertices, as each vertex in this graph is the result of a contraction of a cross edge. So, $\delta(G_i/E' \setminus V') = x^* - 1 = n_i - x - y - 1$. As $W_1 = \emptyset$, we must have that $n_{j_1} - x - y - 1$, and hence $F_{j_1}(x_1, y_1) = \infty$. Similarly, $F_{j_2}(x_2, y_2) = \infty$. So,

$$\begin{aligned}
F'_i(x,y) &= \delta(G_i/E' \setminus V') \\
&= \min(\; F_{j_1}(x_1,y_1) + n_{j_2} - x_2 - y_2 + x^*, \\
&\qquad\quad F_{j_2}(x_2,y_2) + n_{j_1} - x_1 - y_1 + x^*, \\
&\qquad\quad n_i - x - y - 1 \;)
\end{aligned}$$

Case 2 '$W_1 \neq \emptyset$ and $W_2 = \emptyset$': First note that $W_2 = \emptyset$ implies that $F_{j_2}(x_2, y_2) = \infty$, similar to the previous case. Take a vertex $v_1 \in W_1$ which has minimum degree in $G_i/E' \setminus V'$. Note that vertices in W^* are universal in $G_i/E' \setminus V'$, hence have degree that is at least the degree of v_1 in this graph. So, we have (use Lemmas 5 and 6)

$$\begin{aligned}
F'_i(x,y) &= \delta(G_i/E' \setminus V') \\
&= d_{G_i/E' \setminus V'}(v_1) \\
&= d_{G_{j_1}/E_1 \setminus V_1}(v_1) + n_{j_2} - x_2 - y_2 + x^* \\
&\leq F_{j_1}(x_1,y_1) + n_{j_2} - x_2 - y_2 + x^* \\
&= \min(\; F_{j_1}(x_1,y_1) + n_{j_2} - x_2 - y_2 + x^*, \\
&\qquad\quad F_{j_2}(x_2,y_2) + n_{j_1} - x_1 - y_1 + x^*, \\
&\qquad\quad n_i - x - y - 1)
\end{aligned}$$

The last step follows by using that $F_{j_2}(x_2, y_2) = \infty$, and noting that $F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^* \leq n_{j_1} - x_1 - y_1 - 1 + n_{j_2} - x_2 - y_2 + x^* = n_i - x - y - 1$.

Case 3 '$W_1 = \emptyset$ and $W_2 \neq \emptyset$': Similar to the previous Case 2, with the roles of W_1 and W_2 exchanged.

Case 4 '$W_1 \neq \emptyset$ and $W_2 \neq \emptyset$': As in Case 2, $F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^* \leq n_i - x - y - 1$, and similar $F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^* \leq n_i - x - y - 1$. Take a vertex $v_1 \in W_1$ that has minimum degree in $G_i/E' \setminus V'$ among all vertices in W_1, and similar take a vertex $v_2 \in W_2$ with minimum degree in $G_i/E' \setminus V'$.

Again, vertices in W^* have a degree that is at least the degree of v_1 (or v_2) in $G_i/E' \setminus V'$. So, we have using Lemmas 5 and 6

$$F'_i(x,y) = \delta(G_i/E' \setminus V')$$

$$\begin{aligned}
&= \min(\, d_{G_i/E'\setminus V'}(v_1), \\
&\qquad d_{G_i/E'\setminus V'}(v_2)) \\
&= \min(\, d_{G_{j_1}/E_1\setminus V_1}(v_1) + n_{j_2} - x_2 - y_2 + x^*, \\
&\qquad d_{G_{j_2}/E_2\setminus V_2}(v_2) + n_{j_1} - x_1 - y_1 + x^*) \\
&\le \min(\, F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*, \\
&\qquad F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^*) \\
&= \min(\, F_{j_1}(x_1, y_1) + n_{j_2} - x_2 - y_2 + x^*, \\
&\qquad F_{j_2}(x_2, y_2) + n_{j_1} - x_1 - y_1 + x^*, \\
&\qquad n_i - x - y - 1)
\end{aligned}$$

This ends the last case of the proof, and hence we can conclude Lemma 8. □

Now we have seen that our recurrence relation for F'_i is correct, we will show how we can compute F_i given F'_i.

Lemma 9.

$$F_i(x, y) = \max_{0\le z\le x} F'_i(x - z, y + z)$$

Proof. We will first prove that $F_i(x, y) \ge \max_{0\le z\le x} F'_i(x - z, y + z)$.

Claim. $F_i(x, y) \ge F'_i(x - 1, y + 1)$.

Proof. Let E' be a contraction-set of size $x - 1$ without any adjacent cross edges, and let V' be a vertex set of size $y + 1$ such that: $F'_i(x - 1, y + 1) = \delta(G_i/E' \setminus V')$. Let $v \in V'$ be a vertex that we deleted in $G_i/E' \setminus V'$, i.e. $v \in V'$. Instead of deleting it, we can contract it using a cross edge e adjacent to v. Since v is not contained in $G_i/E' \setminus V'$, contracting it via a cross edge (no matter whether e is adjacent to another cross edge in E'), will not decrease any vertex degree. Note that unless v is the only vertex left, a cross edge incident to v always exists, since G_{j_1} and G_{j_2} are graphs with at least one vertex each. Hence, we have a contraction-set $E' \cup \{e\}$ of x edges and a vertex set $V' \setminus \{v\}$ of y vertices, such that:

$$F_i(x, y) \ge \delta(G_i/(E' \cup \{e\}) \setminus (V' \setminus \{v\})) \ge F'_i(x - 1, y + 1)$$

◇

A similar argument can be used when $2 \le z \le x$. Now we contract z vertices with cross edges incident to them instead of deleting them. Thus, for all z, $0 \le z \le x$, we have $F_i(x, y) \ge F'_i(x - z, y + z)$, and hence

$$F_i(x, y) \ge \max_{0\le z\le x} F'_i(x - z, y + z) \tag{1}$$

We will now show $F_i(x, y) \le \max_{0\le z\le x} F'_i(x - z, y + z)$. Let E' be a contraction-set of size x and let be V' be a vertex set of size y, such that $F_i(x, y) = \delta(G_i/E' \setminus V')$. Let $E^* \subseteq E'$ be the set of all cross edges in E', and let be $V^* := \bigcup_{e\in E^*} e$. Observe that (V^*, E^*) is a forest without isolated vertices. Let c be the number of connected components in (V^*, E^*). We modify E^* in the following way to obtain E^{**}. In each connected component of (V^*, E^*), we delete all but one edge, resulting in the forest (V^*, E^{**}). Let be $V^{**} := \bigcup_{e\in E^{**}} e$. Clearly, E^{**} does not contain any two adjacent cross edges and $|E^{**}| = c$, $|V^{**}| = 2c$. We define:

$$E'' := (E' \setminus E^*) \cup E^{**} \quad \text{and} \quad V'' := V' \cup (V^* \setminus V^{**}) \quad \text{and} \quad z := |E'| - |E''|$$

Claim. $|E''| = |E'| - z = x - z$ and $|V''| = |V'| + z = y + z$.

Proof. $|E''| = |E'| - z$ follows directly from the definition of z. To see the other equality, note that the vertex sets are disjoint or contained in each other when applying basic operations. Furthermore, note that in a forest, the number of edges plus the number of connected components equals the number of vertices. We therefore have:

$$\begin{aligned}|V''| &= |V' \cup (V^* \setminus V^{**})| = |V'| + |V^*| - |V^{**}| = y + |E^*| + c - 2c\\ &= y + |E^*| - c = y + |E^*| - |E^{**}| = y + |E'| - |E'| + |E^*| - |E^{**}|\\ &= y + |E'| - (|E'| - |E^*| + |E^{**}|) = y + |E'| - (|E' \setminus E^*| + |E^{**}|)\\ &= y + |E'| - |(E' \setminus E^*) \cup E^{**}| = y + |E'| - |E''| = y + z\end{aligned}$$

◇

Claim. Let E' contain two adjacent cross edges. $e = \{u_1, u_2\}$ and $f = \{u_2, w\}$, such that there is no other edge in E' than e adjacent to f. Let v be a vertex in $G_i/(E' \setminus \{f\}) \setminus V'$. Then it holds: $d_{G_i/E' \setminus V'}(v) = d_{G_i/(E' \setminus \{f\}) \setminus (V' \cup \{w\})}(v)$.

Proof. If w is not adjacent to v then the claim follows easily. So suppose $\{v, w\}$ is an edge in $G_i/(E' \setminus \{f\}) \setminus V'$. Since e was (w.l.o.g.) contracted before f, the vertex created by contracting e is a universal vertex a_e. Now, we can either delete vertex w, resulting in decreasing the degree of v by one, or we can contract edge f to the universal vertex a_e, which also results in 'losing' one edge incident to v, since $\{a_e, v\}$ is already present in $G_i/(E' \setminus \{f\}) \setminus V'$. Hence, the degree of v is the same in both cases. ◇

Applying the last claim iteratively, we can conclude the following: $\delta(G_i/E' \setminus V') = \delta(G_i/E'' \setminus V'')$. Hence, there is a z, such that $F_i(x, y) = F_i'(x - z, y + z)$. From this fact and Equation 1, the lemma follows. □

3.3 A Dynamic Programming Algorithm for Contraction Degeneracy on Cographs

Now, we are able to formulate our main result. So far, we only presented the recurrence relations necessary for dynamic programming. It is an easy task to transform these relations into an algorithm.

Theorem 1. *There is an $O(n^6)$ time algorithm to compute the contraction degeneracy of a given cograph G with n vertices.*

Proof. We first build the cotree in linear time [6, 4], from which the binary cotree can be easily derived. We then compute in bottom up order for each node i in the cotree the relevant values of F_i. For 1-nodes, we first compute F_i'. These computations are as dictated by Lemmas 3, 7, 8 and 9. After the root values are computed, we use Lemma 2 to compute the contraction degeneracy of G.

For the running time, consider the formula for computing F_i' on 1-nodes, since the computation time of this formula dominates the others. We can implement this formula by five nested loops, for x_1, x_2, y_1, y_2 and x^*, running over the appropriate domains (at most $\{0, ..., n_i\}$). In the innermost loop body, we check the restrictions, compute x, y, and the minimum as given in the formula, and we update $F_i'(x, y)$ if we have found a larger value. This loop takes $O(n^5)$ time, and since we have $O(n)$ internal nodes in the cotree, the theorem follows. □

It is also possible to obtain in $O(n^6)$ time the set of contractions that achieve the contraction degeneracy, using standard techniques for transforming a dynamic programming decision algorithm to one that also constructs solutions.

4 Discussion

The contraction degeneracy of a graph appears to be an interesting graph parameter. So far, little is known on it. Its strong relation to the well understood minimum degree δ and degeneracy δD of a graph, and its elementary nature make it a worthwhile object of study. In this paper we have presented a dynamic programming method for computing the contraction degeneracy of a cograph. The running time of our algorithm seems surprisingly high, since cographs are a very restricted class of graphs with a tree representation, and they usually enable faster algorithms. This might be a consequence of the possible inherent hardness of computing δC of a graph. However, it remains an interesting topic for further research to decrease the running time for cographs, or to develop algorithms for other special graph classes.

References

1. H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comp. Sc.*, 209:1–45, 1998.
2. H. L. Bodlaender, A. M. C. A. Koster, and T. Wolle. Contraction and treewidth lower bounds. In *Proceedings 12th Annual European Symposium on Algorithms ESA'04*, 2004. (to appear).
3. H. L. Bodlaender and R. H. Möhring. The pathwidth and treewidth of cographs. *SIAM J. Disc. Math.*, 6:181–188, 1993.
4. A. Bretscher, D. Corneil, M. Habib, and C. Paul. A simple linear time LexBFS cograph recognition algorithm. In H. L. Bodlaender, editor, *Proceedings 27nd International Workshop on Graph-Theoretic Concepts in Computer Science WG'03*, pages 119–130, 2003.
5. D. G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible graphs. *Annals Discrete Math.*, 1:145–162, 1981.
6. D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM J. Comput.*, 14:926–934, 1985.
7. V. Gogate and R. Dechter. A complete anytime algorithm for treewidth. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004. (to appear).
8. A. M. C. A. Koster, H. L. Bodlaender, and S. P. M. van Hoesel. Treewidth: Computational experiments. In H. Broersma, U. Faigle, J. Hurink, and S. Pickl, editors, *Electronic Notes in Discrete Mathematics*, volume 8. Elsevier Science Publishers, 2001.
9. P. Scheffler. *Die Baumweite von Graphen als ein Maß für die Kompliziertheit algorithmischer Probleme.* PhD thesis, Akademie der Wissenschaften der DDR, Berlin, 1989.
10. T. Wolle and H. L. Bodlaender. A note on edge contraction. Technical Report UU-CS-2004-028, Institute of Information and Computing Sciences, Utrecht University, Utrecht, the Netherland, 2004.