



Tero Harju | Chang Li | Ion Petre

Examples on the parallel complexity of signed graphs

TURKU CENTRE *for* COMPUTER SCIENCE

TUCS Technical Report
No 811, January 2007



Examples on the parallel complexity of signed graphs

Tero Harju

Department of Mathematics, University of Turku
FIN-20014 Turku, Finland
harju@utu.fi

Chang Li

Turku Center for Computer Science
Department of Information Technologies, Åbo Akademi University
FIN-20520 Turku, Finland
lchang@abo.fi

Ion Petre

Academy of Finland and
Turku Center for computer Science
Department of Information Technologies, Åbo Akademi University
FIN-20520 Turku, Finland
ipetre@abo.fi

TUCS Technical Report

No 811, January 2007

Abstract

We consider a graph-based model for the study of parallelism in ciliate gene assembly, where a signed graph is associated to each micronuclear gene and the gene assembly is modeled as a graph rewriting process. We show that the complexity measure counting the number of steps needed to fully reduce a graph in parallel varies greatly. The general problem of whether there exists a finite upper bound for the graph parallel complexity remains open.

TUCS Laboratory
Computational Biomodelling Laboratory

1 Introduction

Ciliates emerged as an evolutionary group more than 10^9 years ago [5], and through evolution they diverged into a rich eukaryotic group containing many thousand of species [14]. A unique feature among these species is that they contain two types of functionally different nuclei in the same cell - a micronucleus and a macronucleus. Depending on species multiple copies of each type of nuclei may be present in the same cell, e.g. *U.grandis* (a stichotrich) has 5 to 20 micronuclei and hundreds of macronuclei [14].

During sexual reproduction, ciliates destroy all macronuclei and transform one haploid micronucleus into a macronucleus. The process is especially interesting in a species of ciliates called *stichotrichs*, where the difference between the micronuclear and macronuclear genomes is striking. The macronuclear genes form the functional genome of the ciliate, and the same genes appear on the micronuclear chromosomes as sequences of DNA, called MDS (*macronuclear destined sequences*), separated by non-coding blocks called IESs (*internally eliminated sequences*). The process is being driven by the special structure of the MDSs: each MDS \mathcal{M} ends with a specific sequence of nucleotides that is repeated in the beginning of the MDS that should follow \mathcal{M} in the assembled macronuclear gene. These sequences are called *pointers* using the terminology of computer science. It is currently believed that ciliates splice on the common pointers to assemble the MDSs together. There are two main models for ciliates gene assembly, see [11, 12] and [4, 16], that both agree on this generic mechanism. For more details on how macronuclear genes are assembled, we refer to [10, 14, 18].

In this paper we consider the intramolecular model for gene assembly introduced in [4, 16]. In this model, three molecular operations were conjectured : loop recombination (ld), hairpin recombination (hi) and double-loop recombination (dlad), shown in Figure 1. In each of them, the micronuclear chromosome is folded in such a way that specific folds are formed and recombination takes place on aligned positions. For a detailed discussion on each of the operations we refer to the book [2].

The intramolecular model may be formalized on three levels of abstraction [2]: *signed permutations* (denoting the sequence and the orientation of the MDSs), *signed double occurrence strings* (denoting the sequence and the orientation of the pointers), and *signed graphs* (denoting the overlap structure of the pointers). Correspondingly, the gene assembly is modeled as the processes of sorting permutations, of reducing strings, and of reducing graphs.

A systematic study of parallelism in gene assembly has been initiated in [7]. Regarding the parallel complexity of graphs, we conjectured there that there is no graph with parallel complexity greater than four, see [7, 8]. Based on the recent research [6] and a newly available gene assembly simulator [19], we present in this paper new examples of graphs with higher complexity. The techniques used in several of our examples extend readily to larger sets of graphs.

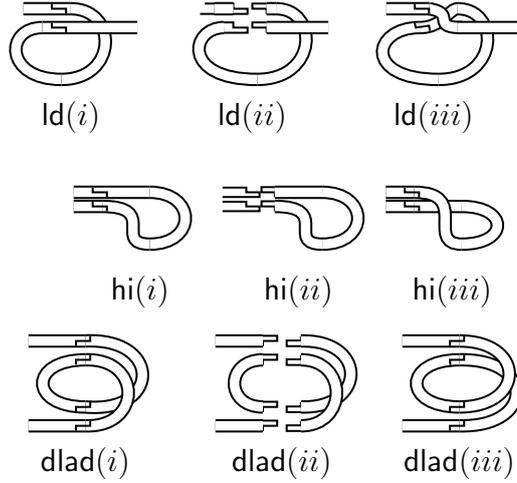


Figure 1: Illustration of the ld, hi, dlad molecular operation showing in each case: (i) the folding, (ii) the recombination, and (iii) the result.

2 Definitions

In this section we give some basic notions concerning signed graphs, the reduction operations for signed graphs, and the parallel application of those operations.

A *signed graph* G is a structure $G = (V, E, \sigma)$, where V a finite set of *vertices*, (V, E) is a undirected graph and $\sigma : V \rightarrow \{+, -\}$ is a vertex-labeling function. We say that a vertex $v \in V$ is *positive* (*negative*, resp.) if $\sigma(v) = +$ ($\sigma(v) = -$, resp.). In the pictures, we denote v^+ or v^- , if $\sigma(v) = +$ or $\sigma(v) = -$, resp. We say that a signed graph is *(all-)negative* (*(all-)positive*, resp.) if all its vertices are negative (positive, resp.). The *neighborhood* of a vertex $v \in V$ is denoted by $N_G(v)$. We call G *discrete* if all its vertices are isolated, i.e., there are no edges in G .

For all $p \in V$ let $G - p$ be the graph induced by the set of vertices $V \setminus \{p\}$. We also denote by $\text{loc}_p(G)$ the *local complement* of G at p : $\text{loc}_p(G) = (V, E', \sigma')$, where $\{x, y\} \in E'$ if and only if either $\{x, y\} \notin E$ for $x, y \in N_G(p)$ or $\{x, y\} \in E$ for $x \notin N_G(p)$ or $y \notin N_G(p)$. Also, $\sigma'(x) = +$ if and only if $\sigma(x) = -$, for all $x \in N_G(p)$ and $\sigma'(x) = \sigma(x)$, otherwise. For an example of local complementation, see Figure 2.

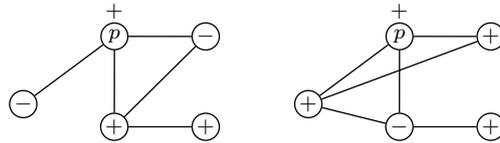


Figure 2: The local complement $\text{loc}_p(G)$ of the graph on the left is given on the right. The signs of the vertices, other than p , are given inside the circles.

The *square* C_4 is the negative signed graph $C_4 = (V_4, E_4, \sigma_4)$, where $V_4 = \{1, 2, 3, 4\}$, $E_4 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 4\}\}$, and $\sigma_4(i) = -$, for all $i \in V_4$. The *diamond* D_4 is the negative signed graph $D_4 = (V_4, E'_4, \sigma_4)$, where $E'_4 = E_4 \cup \{\{1, 3\}\}$.

The molecular operations ld, hi, and dlad may be formalized for signed graphs as follows. Let $G = (V, E, \sigma)$ be a signed graph.

1. For all $p \in V$, ld_p is applicable to G if and only if p is an isolated negative vertex in G . In this case, $\text{ld}_p(G) = G - p$. The domain of ld_p is $\text{dom}(\text{ld}_p) = \{p\}$. The set of all ld-operations is denoted by Ld.
2. For all $p \in V$, hi_p is applicable to G if and only if p is an positive vertex in G . In this case, $\text{hi}_p(G) = \text{loc}_p(G) - p$. The domain of hi_p is $\text{dom}(\text{hi}_p) = \{p\}$. The set of all hi-operations is denoted by Hi.
3. For all $p, q \in V$, $\text{dlad}_{p,q}$ is applicable to G if and only if p and q are adjacent negative vertices in G . Then $\text{dlad}_{p,q}(G) = (V \setminus \{p, q\}, E')$, where E' is obtained from E by complementing the edges that join vertices in $N_G(p)$ with vertices in $N_G(q)$. This means that $(x, y) \in (E' \setminus E) \cup (E \setminus E')$ if and only if

$$\begin{aligned} & x \in N_G(p) \setminus N_G(q) \text{ and } y \in N_G(q), \text{ or} \\ & x \in N_G(q) \cup N_G(p) \text{ and } y \in (N_G(p) \setminus N_G(q)) \cup (N_G(q) \setminus N_G(p)), \text{ or} \\ & x \in N_G(q) \setminus N_G(p) \text{ and } y \in N_G(p). \end{aligned}$$

The domain of $\text{dlad}_{p,q}$ is $\text{dom}(\text{dlad}_{p,q}) = \{p, q\}$. The set of all dlad-operations is denoted by Dlad.

Let

$$\text{LHD} = \text{Ld} \cup \text{Hi} \cup \text{Dlad}$$

be the set of all graph operations corresponding to the three types of molecular operations.

Let $\varphi = \varphi_k \varphi_{k-1} \dots \varphi_1$ be a composition of operations $\varphi_i \in \text{LHD}$. Then $\text{dom}(\varphi) = \bigcup_{i=1}^k \text{dom}(\varphi_i)$, and we say that φ is applicable to a signed graph G , if φ_1 is applicable to G and φ_i is applicable to $\varphi_{i-1} \dots \varphi_1(G)$ for each $i = 2, \dots, k$. We say φ is a *reduction strategy* for graph G if it is applicable to G and $\varphi(G) = \emptyset$.

Intuitively, a set of operations can be applied in parallel to a gene pattern if and only if each operation's applicability is independent of the other's. In other words, a number of operations can be applied in parallel to a gene pattern if they can be (sequentially) applied in any order to that gene pattern. Note that this is consistent with how parallelism and concurrency are defined in Computer Science. The following gives the definition of parallel application of the three operations on a signed graph.

Definition 1 ([7]). Let $S \subseteq \text{LHD}$ be a set of k operations and let $G = (V, E, \sigma)$ be a signed graph. We say that the operations in S can be applied in parallel to G if for any ordering $\varphi_1, \varphi_2, \dots, \varphi_k$ of S , the composition $\varphi_k \dots \varphi_1$ is applicable to G .

The following result provides a simple criterium for two operations to be applicable in parallel. For the proof we refer to [7].

Theorem 1 ([7]). Let $G = (V, E, \sigma)$ be a signed graph and let $\varphi, \psi \in \text{LHD}$ be two operations applicable to G with $\text{dom}(\varphi) \cap \text{dom}(\psi) = \emptyset$.

- (i) If $\varphi \in \text{Ld}$, then φ and ψ can be applied in parallel to G .
- (ii) If $\varphi \in \text{Hi}$, say $\varphi = \text{hi}_p$ with $p \in V$, then φ and ψ can be applied in parallel to G if and only if $N_G(p) \cap \text{dom}(\psi) = \emptyset$.
- (iii) If $\varphi, \psi \in \text{Dlad}$, then φ and ψ can be applied in parallel to G if and only if the subgraph of G induced by $\text{dom}(\varphi) \cup \text{dom}(\psi)$ is not isomorphic to C_4 or D_4 .

In addition to Definition 1, it was shown in [7] that the result of applying different compositions of operations is the same.

Theorem 2 ([7]). Let G be a signed graph and let $S \subseteq \text{LHD}$ be a set of operations applicable in parallel to G . Then for any two compositions φ, ψ of the operations of S , $\varphi(G) = \psi(G)$.

Based on Theorem 2, we can write $S(G) = \varphi(G)$ for any set S of operations applicable in parallel to G and any composition φ of these operation. Parallel complexity is defined as follows.

Definition 2 ([9]). Let G be a signed graph, and let $S_1, \dots, S_k \subseteq \text{LHD}$ be a set of k operations applicable in parallel to G . If $(S_k \circ \dots \circ S_1)(G) = \emptyset$, then we say that $S = S_k \circ \dots \circ S_1$ is a *parallel reduction strategy* for G . In this case the *parallel complexity* of S is $\mathcal{C}(S) = k$. The parallel complexity of the graph G is defined as follows:

$$\mathcal{C}(G) = \min\{\mathcal{C}(S) \mid S \text{ is a parallel reduction strategy for } G\}.$$

3 Parallel complexity of trees

In this section we investigate the parallel complexity of signed trees. While the general problem of whether the complexity is bounded remains open even in this special case, we give here several examples of trees with complexity up to five.

3.1 Uniformly signed trees have bounded complexity

In the following, we consider the cases of negative and positive trees. It was proved in [6] that these types of signed trees have low parallel complexity, indeed, the parallel complexity is at most three. We give several examples of such trees.

Theorem 3 ([6]). *Negative trees have parallel complexity at most two.*

Example 1. Let G be the negative tree shown in Figure 3. Here the set $\text{dlad}_{1,2}$ and $\text{dlad}_{3,4}$ can be applied parallel to G , and therefore a reduction of G in two steps is $\{\text{ld}_5\} \circ \{\text{dlad}_{1,2}, \text{dlad}_{3,4}\}$.

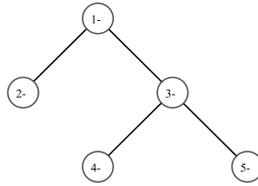


Figure 3: A negative tree.

Theorem 4 ([6]). *Positive trees have parallel complexity at most three.*

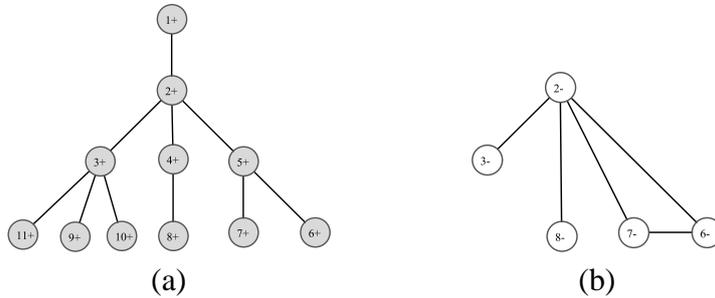


Figure 4: (a) A positive tree; (b) a negative clique connecting to a negative tree.

Example 2. The positive G be the positive tree shown in Figure 4(a). Then a 3-step parallel reduction for G is $\{\text{ld}_8\} \circ \{\text{dlad}_{2,3}, \text{dlad}_{6,7}\} \circ \{\text{hi}_1, \text{hi}_4, \text{hi}_5, \text{hi}_9, \text{hi}_{10}, \text{hi}_{11}\}$. The result of reducing G after the first parallel step is shown in Figure 4(b).

We may have trees of complexity up to three even by considering only uniformly signed *paths*:

Lemma 5. *Any negative path with $2n$ vertices, $n \geq 1$, has complexity one. Any negative path with $2n + 1$ vertices, $n \geq 1$, has complexity two.*

Proof. Let G be the path of negative $2n$ vertices with the edges $\{p_i, p_{i+1}\}$ for $i = 1, 2, \dots, 2n - 1$. Then $S = \{\text{dlad}_{p_{2i-1}, p_{2i}} \mid i = 1, \dots, n\}$ can be applied in parallel to G resulting in the empty graph $S(G)$. On the other hand, if the length of the path is odd, the above choices leave one isolated negative vertex p_{2n+1} , which is then destroyed in the second step by an application of $\text{ld}_{p_{2n+1}}$. Finally, it is obvious that these bounds are optimal. \square

The same idea can be used to prove the following lemma.

Lemma 6. *Let $n \geq 1$. Any positive path with $3n$ or $3n + 1$ vertices has parallel complexity two. Any positive path with $3n + 2$ vertices has parallel complexity three.*

Proof. Let G have the edges $\{p_i, p_{i+1}\}$ for $i = 1, 2, \dots, k$. First of all, by Lemma 1(ii), we have $\mathcal{C}(G) \geq 2$. If $k = 3n$, then $S = \{\text{hi}_{3i+2} \mid i = 0, 1, \dots, n - 1\}$ can be applied in parallel to G , and $S(G)$ is a negative path of $2n$ vertices. This path has complexity one by Lemma 5, and hence $\mathcal{C}(G) = 2$.

If $k = 3n + 1$, then $S = \{\text{hi}_{3i+1} \mid i = 0, 1, \dots, n - 1\}$ can be applied in parallel to G , and $S(G)$ is again a negative path of $2n$ vertices. Hence Lemma 5 guarantees that $\mathcal{C}(G) = 2$.

Finally, if $k = 3n + 2$, then $S = \{\text{hi}_{3i+1} \mid i = 0, 1, \dots, n - 1\}$ can be applied in parallel to G , and $S(G)$ is a negative path of $2n + 1$ vertices. By Lemma 5, $\mathcal{C}(G) \leq 3$. It can be seen that, in this case, there is no strategy reducing G in two steps: indeed, any positive vertex remains in G after the first step should be isolated. Hence $\mathcal{C}(G) = 3$. \square

3.2 Arbitrary signed trees

In the following we give examples of signed trees with complexity up to five. Some of these examples have been found through an automated search based on the gene assembly simulator [19].

Example 3. The following are examples of signed trees with parallel complexity two to three.

- (a) Let G_1 be the graph in Figure 5(a). Clearly, $\mathcal{C}(G_1) = 2$.
- (b) Let G_2 be the graph in Figure 5(b). Obviously, only hi_1 is applicable in the first step, which leads to a tree of complexity two. Thus, $\mathcal{C}(G_2) = 3$.
- (c) Let G_3 be the graph in Figure 5(c). We see either $\{\text{dlad}_{1,2}, \text{hi}_4\}$ or $\{\text{hi}_3\}$ is applicable to G_3 . Applying $\{\text{dlad}_{1,2}, \text{hi}_4\}$ on G_3 reduces the graph to an isolated negative vertex, in this case, G_3 is reduced in two steps. Otherwise, hi_3 reduces G_3 to a signed tree as in Figure 5(b), which requires three more reduction steps. Thus, $\mathcal{C}(G_3) = 2$.

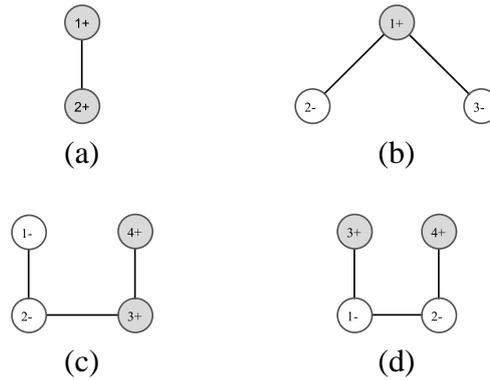


Figure 5: The graphs of Example 3: (a) the smallest signed tree with parallel complexity two; (b) the smallest signed tree with parallel complexity three; (c) a signed path with parallel complexity two; (d) a signed path with parallel complexity three.

(d) Let G_4 be the graph in Figure 5(d). Either $\{\text{dlad}_{1,2}\}$ or $\{\text{hi}_3, \text{hi}_4\}$ is applicable to G_4 . Applying $\text{dlad}_{1,2}$ creates an edge $\{3, 4\}$ between positive vertices, in this case, G_4 is reduced in three steps. Otherwise, applying $\{\text{hi}_3, \text{hi}_4\}$ changes vertices $\{1, 2\}$ to positive, then it requires two more steps. Applying either hi_3 or hi_4 to G_4 , also leads to strategies in three steps. Thus, $\mathcal{C}(G_4) = 3$.

Example 4. The smallest example of a signed tree with $\mathcal{C}(G) = 4$ is shown in Figure 6(a). Let G be that graph. Then it is possible to apply hi_4 , $\text{dlad}_{1,3}$ or $\text{dlad}_{2,3}$ to G , but not in parallel. If $\text{dlad}_{1,3}$ (the case $\text{dlad}_{2,3}$ is symmetric) is applied, then we obtain an isolated vertex and a tree isomorphic to the one in Figure 5(b), which requires three more reduction steps. Otherwise, $G' = \text{hi}_4(G)$ is shown in Figure 6(b). It has been proved that $\mathcal{C}(G') = 3$, see [6]. Thus, $\mathcal{C}(G) = 4$.

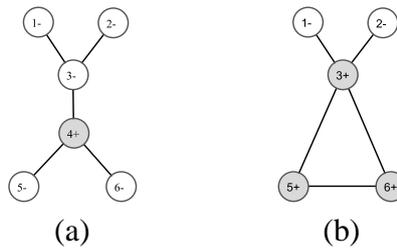


Figure 6: (a) A signed tree G has parallel complexity four; (b) the reduced graph $G' = \text{hi}_4(G)$.

Example 5. The smallest example of a signed tree G with parallel complexity five is shown in Figure 7(a).

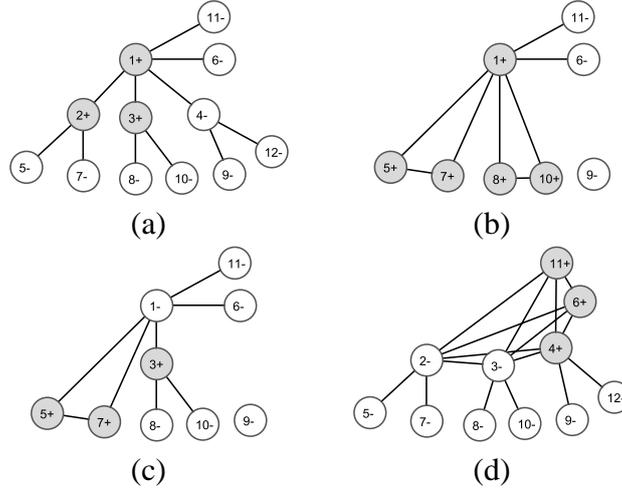


Figure 7: (a) A signed tree G with parallel complexity five; (b) a reduced graph $G' = \{hi_2, hi_3, dlad_{4,9}\}(G)$; (c) a reduction $G'' = \{hi_2, dlad_{4,9}\}(G)$; (d) a reduction $G''' = hi_1(G)$.

Note that hi_1 , hi_2 , hi_3 , $dlad_{4,9}$ (or $dlad_{4,12}$) are applicable to G , and hi_1 is applicable in parallel with none of other operations.

For $S = \{hi_2, hi_3, dlad_{4,9}\}$, the graph $G' = S(G)$ is shown in Figure 7(b). Apart from symmetric cases, the only operations applicable to G' are hi_1 , hi_5 and $\{hi_5, hi_8\}$. According to our previous examples, it is straightforward to see that all the resulting graphs have complexity at least three.

Let then $S = \{hi_2, dlad_{4,9}\}$. The graph $G'' = S(G)$ is shown in Figure 7(c) (replacing hi_2 with hi_3 and/or $dlad_{4,9}$ with $dlad_{4,12}$ leads to isomorphic graphs). Clearly, $\{ld_9, dlad_{1,6}\}$ and $\{ld_9, hi_3, hi_5\}$, and their subsets, are applicable to G'' , which leads to graphs having complexity at least three.

Finally, let $G''' = hi_1(G)$ be the graph shown in Figure 7(d). Apart from symmetric cases, only hi_4 , hi_6 and $\{dlad_{2,5}, dlad_{3,8}\}$, and its subsets, are applicable to G''' , leading in all cases to graphs of complexity at least three.

4 Parallel complexity of arbitrary graphs

In this section we present some examples of general signed graphs with parallel complexity up to six.

4.1 Graphs with parallel complexity three

In the following, we present a smallest example of a negative graph having parallel complexity three. Note that it has been conjecture in [8] that negative graphs have

parallel complexity bounded by three.

Example 6. An example of a negative graph having parallel complexity three is shown in Figure 8. It is easy to see that modulo symmetry, the only two possible edge selections for the first step are $\{\text{dlad}_{1,2}, \text{dlad}_{3,4}\}$, $\{\text{dlad}_{1,2}, \text{dlad}_{4,5}\}$, $\{\text{dlad}_{1,2}, \text{dlad}_{4,7}\}$, or their subsets. All these choices lead to strategies that require at least three steps. A three-step reduction strategy is the following: $\{\text{ld}_5\} \circ \{\text{dlad}_{6,7}\} \circ \{\text{dlad}_{1,2}, \text{dlad}_{3,4}\}$.

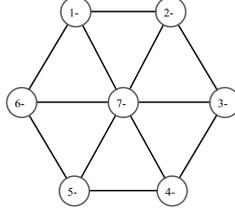


Figure 8: A negative graph with parallel complexity three.

Note that the graph in Figure 8 is in fact a tripartite graph, with partitions $A = \{7\}$, $B = \{1, 3, 5\}$, $C = \{2, 4, 6\}$. The upper bound of the parallel complexity of negative tripartite graphs is discussed in [6]. The following result is about the simpler case of *complete* bipartite and tripartite positive graphs.

Theorem 7. *The complete bipartite and tripartite positive graphs have parallel complexity at most three.*

Proof. Let G be a complete bipartite or tripartite positive graph. Then hi is applicable in parallel to all vertices in any partition of G . Let P be an arbitrary partition of G . If G is bipartite, then $\text{hi}_p(G)$ is either a discrete graph, or a negative clique, i.e., $\mathcal{C}(G) \leq 2$. Assume that G is tripartite, then $\text{hi}_p(G)$ is either discrete or a complete positive bipartite graph, i.e., $\mathcal{C}(G) \leq 3$. \square

4.2 Graphs with parallel complexity at least four

The following examples are positive graphs of parallel complexity four: one tripartite positive graph and one arbitrary positive graph. Moreover, as shown by an automated search based on [19], these are the smallest such graphs.

Example 7. The smallest example of a tripartite positive graph G with parallel complexity four is shown in Figure 9. In G , three partitions are: $A = \{1, 4\}$, $B = \{2, 3\}$, $C = \{5, 6\}$. Note that $\text{hi}_A(G)$, $\text{hi}_B(G)$, $\text{hi}_C(G)$ are all isomorphic to the graph G' shown in Figure 9(b). It is easy to check that $\mathcal{C}(G') = 3$. Consequently, $\mathcal{C}(G) = 4$.

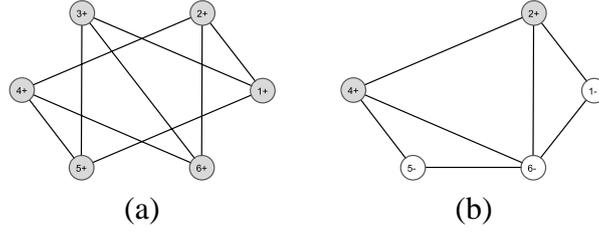


Figure 9: (a) A positive tripartite graph with parallel complexity four; (b) a non-maximum reduction: $hi_3(G)$.

Example 8. The graphs in the following have similar structure, yet they have different complexities.

The graph G_1 in Figure 10(a) has complexity four, indeed $G'_1 = hi_1(G_1)$ is shown in Figure 10(a'), then $\{dlad_{2,8}, dlad_{3,6}\}(G'_1)$, reducing to a graph isomorphic to that in Figure 5(a), which requires no more than 2 reduction steps. Thus, whole reduction is in 4 steps.

The graph G_2 in Figure 10(b) has complexity five, in this case, it is possible to apply hi on two positive vertices in diagonal, e.g. $\{hi_1, hi_3\}$, then the neighbors of reduced vertices (by hi) forms new edges (eventually cliques), see in Figure 10(b'), to reduce these cliques by applying i.e., $\{hi_5, hi_9\}$, remaining the graph containing some 3-step signed trees and some discrete vertices.

The graph G_3 in Figure 10(c) has complexity five, one reduction is applying hi_1 in the first step, $G'_3 = hi_1(G_3)$ is shown in Figure 10(c'). Then $\{hi_3, hi_7\}$ reduces G'_3 to 3-step signed tree, negative trees, etc, thus the whole reduction is in 5 steps.

A different 5-step strategy for reducing G_3 is obtained by applying $\{hi_1, hi_4\}$, after which the remaining positive vertex 5 forms two positive cliques with $\{6, 7\}$ and $\{12, 13\}$. Reducing both cliques by $\{hi_6, hi_{12}\}$, there remains some 3-step signed trees, thus the whole reduction is in 5 steps.

We can generalize Example 8 as follows.

Theorem 8. Let $G = (V, E, \sigma)$ be a signed graph such that $V = V_1 \cup V_2$, with $V_1 = \{x_1, \dots, x_n\}$, $V_2 = \{y_1, z_1, y_2, z_2, \dots, y_n, z_n\}$, and $E = \{\{x_i, y_i\}, \{x_i, z_i\}, \{x_j, x_{j+1}\}, \{x_1, x_n\} \mid 1 \leq i \leq n, 1 \leq j \leq n-1\}$, $\sigma(x_i) = +$, $\sigma(y_i) = -$, $\sigma(z_i) = -$, for all $1 \leq i \leq n$. If $n = 3k$, for some $k \geq 1$, then $\mathcal{C}(G) \leq 4$. Otherwise, $\mathcal{C}(G) \leq 5$.

Proof. Consider first the case when $n = 3k$, for some $k \geq 1$. Denote $S_1 = \{x_1, x_4, \dots, x_{3k-2}\}$ and note that hi_{S_1} is applicable to G . Let $G' = hi_{S_1}(G)$:

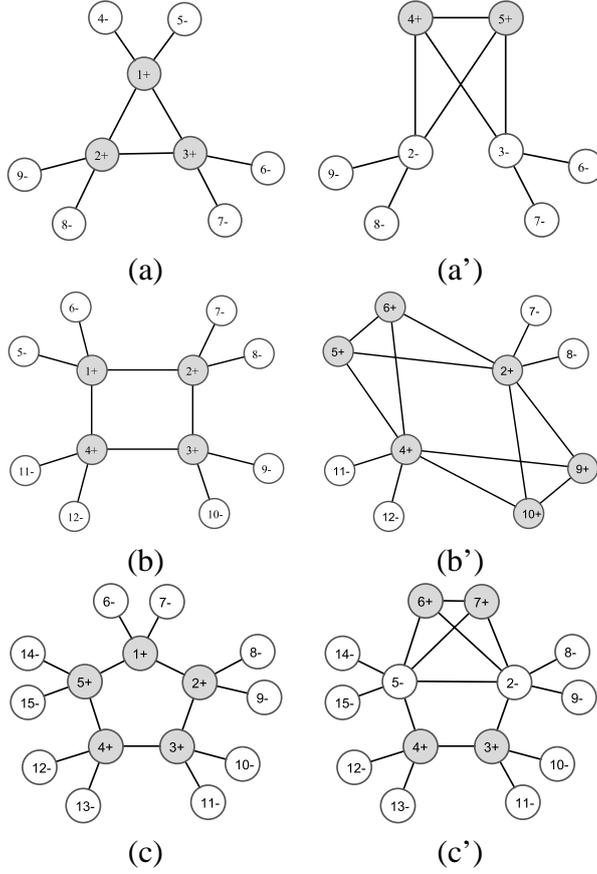


Figure 10: (a) A graph G_1 with a positive circle of $3n$ vertices; (a') $hi_1(G_1)$; (b) a graph G_2 with a positive circle of $3n+1$ vertices; (b') $hi_1(G_2)$; (c) a graph G_3 with a positive circle of $3n+2$ vertices; (c') $hi_1(G_3)$.

$G' = (V', E', \sigma')$, where $V' = V - S_1$,

$$E' = \left\{ \{x_{3i-1}, y_{3i-1}\}, \{x_{3i-1}, z_{3i-1}\}, \{x_{3i}, y_{3i}\}, \{x_{3i}, z_{3i}\}, \{x_{3i-1}, y_{3i-2}\}, \right. \\ \left. \{x_{3i-1}, z_{3i-2}\}, \{x_{3i-3}, y_{3i-2}\}, \{x_{3i-3}, z_{3i-2}\}, \{y_{3i-2}, z_{3i-2}\}, \right. \\ \left. \{x_{3i-3}, x_{3i-1}\}, \{x_{3i-1}, x_{3i}\}, \{x_2, x_n\} \mid 1 \leq i \leq k \right\}$$

and $\sigma'(z) = +$ if and only if $z \in \{y_{3i-2}, z_{3i-2} \mid 1 \leq i \leq k\}$.

Denote $S_2 = \left\{ \{x_{3i-1}, y_{3i-1}\}, \{x_{3i}, y_{3i}\} \right\}$ and note that $dlad_{S_2}$ is applicable to G' . Then $G'' = dlad_{S_2}(G')$ consists of k isolated negative vertices and k graphs isomorphic to the graph in Figure 5(a), having complexity two. Consequently $\mathcal{C}(G) \leq 4$.

The case $n = 3k + 1$ is similar, with one exception: the graph corresponding to G'' as above also has a subgraph isomorphic to the graph in Figure 5(b). Thus, $\mathcal{C}(G) \leq 5$.

In the case $n = 3k + 2$, consider in the first step $hi_{S'_1}$, where $S'_1 = \{x_1, x_4, \dots, x_{3k-2}, x_{3k+1}\}$. In the second step consider

$$S'_2 = S_2 - \{\{x_2, y_2\}, \{x_{3k}, y_{3k}\}, \{x_{3k+2}, y_{3k+2}\}\} \text{ and } T_2 = \{y_1, y_{3k+1}\},$$

and apply in parallel $dlad_{S'_2}$ and hi_{T_2} . The resulting graph H consists of a number of isolated negative vertices, a number of graphs isomorphic to the graph in Figure 5(a), and three graphs isomorphic to the graph in Figure 5(b). Thus, $\mathcal{C}(G) \leq 5$. \square

Example 9. A more involved graph with parallel complexity six is shown in Figure 11(a). Modulo isomorphism, only $\{hi_1, hi_3\}$, $\{hi_1, hi_7\}$, $\{hi_1\}$ and $\{hi_7\}$ are applicable to G . Applying $\{hi_1, hi_3\}$ in the first step, and followed by $\{dlad_{5,17}, dlad_{6,19}, dlad_{7,21}, dlad_{8,23}\}$, G is reduced to a graph isomorphic to that in Figure 10(b'), which takes 4 reduction steps. Thus, G has a reduction in six steps. Otherwise, if $\{hi_1, hi_7\}$ is applied in the first step, and $\{dlad_{5,17}, dlad_{4,16}, dlad_{8,24}, dlad_{3,13}\}$ in the second step, then G is reduced to a graph as in Figure 11(b) having parallel complexity four, since two 3-step trees have an edge in between, refer the proof in [6]. An automated search using the gene assembly simulator [19] suggests that no reduction in fewer than six steps exists. The computer search takes several tens of hours for on a “standard” computer.

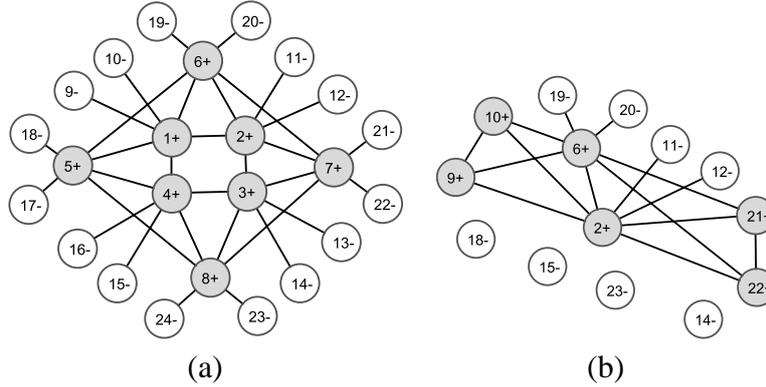


Figure 11: (a)A graph with complexity six, (b) a reduced graph with complexity four.

5 Discussion

The intriguing problem, whether the parallel complexity of signed graphs is bounded by a constant, remains open, even for “simple” graphs such as the trees. We state here the problem in several setups.

Problem 1. *Are the parallel complexity of a) trees; b) negative graphs; c) arbitrary signed graphs finitely bounded?*

Acknowledgment The authors gratefully acknowledge support by Academy of Finland (TH – project 39802, CL – project 203667, IP – project 108421).

References

- [1] Chang, W.J., Bryson, P.D., Liang, H., Shin, M.K., Landweber, L., The evolutionary origin of a complex scrambled gene. *Proceedings of the National Academy of Sciences of the US* **102**(42) (2005) 15149–15154
- [2] Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D. M., and Rozenberg, G. (2004) *Computation in Living Cells: Gene Assembly in Ciliates*, Springer
- [3] Ehrenfeucht, A., Petre, I., Prescott, D. M., and Rozenberg, G., Universal and simple operations for gene assembly in ciliates. In: V. Mitrana and C. Martin-Vide (eds.) *Words, Sequences, Languages: Where Computer Science, Biology and Linguistics Meet*, Kluwer Academic, Dordrecht, (2001) pp. 329–342
- [4] Ehrenfeucht, A., Prescott, D. M., and Rozenberg, G., Computational aspects of gene (un)scrambling in ciliates. In: L. F. Landweber, E. Winfree (eds.) *Evolution as Computation*, Springer, Berlin, Heidelberg, New York (2001) pp. 216–256
- [5] Fleury, A., Delgado, F., Adoutte, A. (1992) Molecular phylogeny of ciliates: what does it tell us about the evolution of the cytoskeleton and of developmental strategies? *Dev. Genet.* **13** pp 247–254
- [6] Harju, T., Li, C., and Petre, I., Results on parallel reductions of signed overlap graphs, manuscript (2006)
- [7] Harju, T., Li, C., Petre, I., and Rozenberg, G., Parallelism in gene assembly, In: Proceedings of the 10th International Meeting on DNA-based computers DNA 10, Milan, Italy, *Lecture Notes in Computer Science* **3384** (2005) 140–150
- [8] Harju, T., Li, C., Petre, I. and Rozenberg, G., Parallelism in gene assembly, *Natural Computing*, **5** (2006) 203–223.
- [9] Harju, T., Li, C., Petre, I., Rozenberg, G., Complexity measures for gene assembly. *Lecture Notes in Bioinformatics (LNBI)* Volume number 4366, Springer, 2007
- [10] Jahn, C. L., and Klobutcher, L. A., Genome remodeling in ciliated protozoa. *Ann. Rev. Microbiol.* **56** (2000), 489–520.
- [11] Landweber, L. F., and Kari, L., The evolution of cellular computing: Nature’s solution to a computational problem. In: *Proceedings of the 4th DIMACS Meeting on DNA-Based Computers*, Philadelphia, PA (1998) pp. 3–15
- [12] Landweber, L. F., and Kari, L., Universal molecular computation in ciliates. In: L. F. Landweber and E. Winfree (eds.) *Evolution as Computation*, Springer, Berlin Heidelberg New York (2002)
- [13] Mayo, K.A., Orlas, E. (1981) Further evidence for lack of gene expression in the *Tetrahymena thermophila*. *Nucleic Acids Res.* **16** pp 2189–2201
- [14] Prescott, D. M., The DNA of ciliated protozoa. *Microbiol. Rev.* **58**(2) (1994) 233–267
- [15] Prescott, D. M., DNA manipulations in ciliates. In: W. Brauer, H. Ehrig, J. Jarhumäki, A. Salomaa (eds.) *Formal and Natural Computing: essays dedicated to Grzegorz Rozenberg*, LNCS 2300, Springer (2002) 394–417

- [16] Prescott, D. M., Ehrenfeucht, A., and Rozenberg, G., Molecular operations for DNA processing in hypotrichous ciliates. *Europ. J. Protistology* **37** (2001) 241–260
- [17] West, D. B. (1996) *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, NJ
- [18] Yao, M.C., Fuller, P., Xi, X., Programmed DNA Deletion As an RNA-Guided System of Genome Defense, *Science* 300 (2003) 1581–1584
- [19] Gene assembly simulator (2006). <http://combio.abo.fi/simulator/simulator.php>

TURKU
CENTRE *for*
COMPUTER
SCIENCE

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www.tucs.fi



University of Turku

- Department of Information Technology
- Department of Mathematics



Åbo Akademi University

- Department of Computer Science
- Institute for Advanced Management Systems Research



Turku School of Economics and Business Administration

- Institute of Information Systems Sciences

ISBN 978-952-12-1889-7
ISSN 1239-1891