Concurrency in graphs with maximum degree 4: complexity and an approximation algorithm[1]

Gladstone M. Arantes Jr[2,4] Luerbio Faria [3] Felipe M. G. França [4]

Abstract. The decision problem CONCURRENCY (CON) is stated as follows: Given a graph $G = (V, E)$, and a pair of positive integers m and p, decide whether there exists an acyclic orientation ω over E associated to an amount of *concurrency* $\gamma(\omega) \geq \frac{m}{p}$, where m is the number of times every vertex of V becomes a sink inside a *period* of lenght p repeated acyclic orientations resulting from the graph dynamics known as *scheduling by edge reversal* (SER). Barbosa and Gafni have shown that CON is NP-complete for graphs in general based on an instance with an universal vertex. It is established here the NP-completeness of CON for graphs with maximum degree 4. In addition, it is shown, via an L-reduction, that the well-known negative result of the non-approximability of COLOURING within $n^{\frac{1}{7}-\epsilon}$, unless P=NP, is also valid for an optimisation equivalent version of CON. Moreover, a $\frac{1}{2}$-approximation algorithm for finding acyclic orientations in graphs with maximum degree 4 is given and, more generally, given a graph G with maximum degree Δ, it is shown how to design a polynomial time $\left(\frac{2}{\Delta}\right)$-approximation.

Keywords: complexity classes; computational difficulty of problems; NP-complete; L-reduction; concurrency; distributed algorithms; approximation algorithm with a fixed ratio; bounded degree graphs.

1 Introduction

A natural question when studying the complexity of a graph-theoretical decision problem is to determine for which special graph classes and upper bounds on the vertex degrees the problem remains NP-complete and, ideally, provide a polynomial approximation algorithm for it. The decision problem considered in this work is CONCURRENCY (CON), which is stated as follows: Given a graph $G = (V, E)$ and a pair of positive integers m and p, decide whether there exists an acyclic orientation ω over E, from which a *scheduling by edge reversal* (SER) graph dynamics, associated to an amount of *concurrency* $\gamma(\omega) \geq \frac{m}{p}$, starts from.

$G = (V, E)$, or simply G, represents a distributed system where processing nodes, represented by V, operate upon scarce atomic shared resources in a mutual exclusion style. In order to complete the representation of this neighborhood-constrained system, E is defined by the set of edges resulting from representing each atomic shared resource by a clique in G. SER works in the following way: all (and only) sink vertices in ω reverse the orientation of their edges, each one becoming a source. This ensures that neighboring nodes in the target distributed

[1] Supported by CNPq, CAPES, FAPERJ, FINEP, Brazilian research agencies.
[2] BNDES. glads@bndes.gov.br
[3] Departamento de Matemática, FFP-UERJ. luerbio@cos.ufrj.br
[4] COPPE-Sistemas e Computação UFRJ - Rio de Janeiro-Brazil. felipe@cos.ufrj.br

system cannot operate simultaneously upon shared resources. Another acyclic orientation ω' is formed and it is easy to see that a new set of sinks will be enabled to operate in the next step. SER consists of such consecutive acyclic orientations being defined over G through time. Considering G finite and, consequently, a finite number of acyclic orientations over G, eventually a repetition, i.e., a *period* of lenght p, will occur. Another interesting property of SER lies in the fact that, inside any given period, each vertex operates, i.e., becomes a sink, the same number m of times [3]. Hence, another important aspect of the SER dynamics is on the definition of a distributed scheduling scheme ensuring fairness in the operation among all processing elements of the target system.

Depending on the initial acyclic orientation ω over a given graph G, different periods can be reached [2] through the SER dynamics. So, different combinations of m and p can exist for a given graph G so that Barbosa and Gafni defined the measure $\frac{m}{p}$ as the *concurrency* $\gamma(\omega)$ associated to the orientation ω. Barbosa and Gafni showed that CON is a NP-complete problem for graphs in general reducing CON from COLOURING [3]. That transformation used an instance of CON which had a graph $G = (V, E)$ with maximum degree $\Delta = n - 1$, where $n = |V|$.

In the present work, it is proven that CON remains NP-complete restricted to graphs with maximum degree 4. This NP-completeness result is obtained by first showing, in Section 2, the NP-completeness of a version of the decision problem NOT ALL EQUAL 3SAT (NAEQ3SAT), where each variable occurs exactly three times, each literal occurs, and each sized two clause has at least one literal which occurs once. An instance $I = (U, C)$ with set of variables U and collection of clauses C of this version of (NAEQ3SAT) is used to obtain, in polynomial time, an instance of CON $G = (V, E)$ with maximum degree 4 and the pair of positive integers $m = 1$ and $p = 3$, such that I is satisfiable if and only if G allows an acyclic orientation of E associated to an amount of concurrency $\gamma = \frac{m}{p}$.

Bellare *et al.* [4] showed that MINIMUM GRAPH COLOURING for general graphs is not approximable within $n^{\frac{1}{7}-\varepsilon}$, for any $\varepsilon > 0$. On the other hand, a result of Halldórsson [7] establishes that MINIMUM GRAPH COLOURING for general graphs is approximable within $O(n\frac{(\log\log n)^2}{(\log n)^3})$. In Section 3, Bellare's negative result for MINIMUM GRAPH COLOURING is extended to CON via an L-reduction.

Up to the present date, no approximation algorithm for finding maximum concurrency in general graphs is known. By restricting the universe to connected graphs with maximum degree 4, Section 4 introduces a $\frac{1}{2}$-approximation algorithm for finding maximum concurrency in this class of graphs. Actually, this algorithm has as input a connected graph $G = (V, E)$ with maximum degree Δ and outputs an acyclic orientation leading to concurrency $\frac{2}{\Delta}$.

1.1 Defining problems

We describe next all decision and optimisation problems adressed in this work.

1. CON - CONCURRENCY (decision)
   INSTANCE: Graph $G = (V, E)$ and positive integers m and p.
   QUESTION: Is there an acyclic orientation over E with concurrency $\gamma(\omega) \geq \frac{m}{p}$?

2. MAXCON - MAXIMUM CONCURRENCY (optimisation)
INSTANCE: Graph $G = (V, E)$.
GOAL: Find an acyclic orientation ω over E which maximizes the concurrency $\gamma(\omega) = \frac{m}{p}$.
3. MIN^{-1}CON - MINIMUM OF THE INVERSE OF THE CONCURRENCY (optimisation)
INSTANCE: Graph $G = (V, E)$.
GOAL: Find an acyclic orientation ω over E which minimizes the value $\frac{p}{m}$.
4. NAEQ3SAT - NOT ALL EQUAL 3 SAT (decision)
INSTANCE: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$ literals.
QUESTION: Is there a satisfying truth assignment for U such that each clause in C has at least one true literal and one false literal?
5. NAEQ3SAT$_{\underline{\bar{k}}}$ - NOT ALL EQUAL 3 SAT (decision)
INSTANCE: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $2 \leq |c| \leq 3$ literals, each variable occurs exactly k times in C, each literal occurs and each clause of size 2 has a literal which occurs once, i.e., each clause of size 2 contains a literal, such that the other two occurrences of the corresponding variable in C appear as the negation of that literal.
QUESTION: Is there a satisfying truth assignment for U such that each clause in C has at least one true literal and one false literal?

Given an optimisation problem Π and I an instance of Π, the optimal value of Π for I is denoted here by $Opt_{\Pi}(I)$. It is observed that are defined here two optimisation problems: MAXCON and MIN^{-1}CON, both associated with the decision problem CON. In this paper it is established a negative result for MIN^{-1}CON and an approximation algorithm for MAXCON. Although MIN^{-1}CON and MAXCON are related, $Opt_{\text{MAXCON}} = \frac{1}{Opt_{\text{MIN}^{-1}\text{CON}}}$, when dealing with approximation theory not necessarily a result for one of the problems holds for other.

2 CONCURRENCY is NP-complete for graphs with maximum degree 4.

In Lemma 1 is proven that NAEQ3SAT$_{\underline{\bar{3}}}$ is an NP-complete problem. NAEQ3SAT$_{\underline{\bar{3}}}$ is used later to prove that CON is NP-complete for maximum degree 4 graphs.

Lemma 1. NAEQ3SAT$_{\underline{\bar{3}}}$ *is NP-complete.*

Proof. Let $I = (U, C)$ be an instance of the NP-complete problem NAEQ3SAT. One yields in polynomial time, in the size of I, the instance $I' = (U', C')$ of NAEQ3SAT$_{\underline{\bar{3}}}$ such that I is satisfiable if and only if I' is satisfiable. One starts by setting $U' := U$ and $C' := C$ and updating U' and C' according to the number of occurrences of each variable u of U. Let u be a variable of U which occurs $k = k_1 + k_2$ times in C, where k_1 is the number of times u appears positively in C, and k_2 the number of times u appears negated in C. Three cases are to be considered: cases (1) and (2) which are, when $k \leq 2$; and case (3), when $3 \leq k$. It is observed that in cases (1) and (2), I' is adjusted, such that u appears 3 times in C. On the other hand, it is adjusted the over 3 occurrence of u in case (3), where the construction serves the purpose of forcing all literals of a same type to take the same value (items (a) and (b)), and opposite type literals to receive opposite values(item (c)).

1. ($k = 1$) in this case $(u \vee \overline{u})$ is added to C'.
2. ($k = 2$) this case leads to 2 subcases:
   (a) ($k_1 = 2$) in this case u_1 is added to U' and $(u_1 \vee \overline{u}), (u_1 \vee \overline{u_1})$ added to C'.
   (b) ($k_2 = 2$) in this case u_1 is added to U' and $(\overline{u_1} \vee u), (u_1 \vee \overline{u_1})$ added to C'.
   (c) ($k_1 = k_2 = 1$) in this case u_1, u_2 are added to U' and $(u_1 \vee \overline{u} \vee u_2), (\overline{u_1} \vee u_2)$, $(u_1 \vee \overline{u_2})$ added to C'.
3. ($k_1 \geq 3$ or $k_2 \geq 3$) in this case one replaces the k_1 occurrences of u by the new literals $u_1, u_2, u_3, \ldots, u_{k_1}$, and one replaces the k_2 occurrences of $\bar{u}$ by the new literals $v_1, v_2, v_3, \ldots, v_{k_2}$. Moreover, we add the 2 new sets of variables $T = \{t_1, t_2, t_3, \ldots t_{k_1}\}$ and $W = \{w_1, w_2, w_3, \ldots w_{k_2}\}$, which are added to C', respectively, according to $k_1 \neq 0$, or $k_2 \neq 0$.
   (a) If u occurs positively one adds to C' the collection (I) of clauses below:

$$(I) = \left\{ \begin{array}{cc} \underline{\text{order of occurrence of } u} & \underline{\text{clauses of } (I)} \\ \downarrow & \downarrow \\ 1 & (u_1 \vee \overline{t_1} \vee t_2), (\overline{u_1} \vee t_2) \\ 2 & (u_2 \vee \overline{t_2} \vee t_3), (\overline{u_2} \vee t_3) \\ 3 & (u_3 \vee \overline{t_3} \vee t_4), (\overline{u_3} \vee t_4) \\ \vdots & \vdots \\ k_1 - 1 & (u_{k_1-1} \vee \overline{t_{k_1-1}} \vee t_{k_1}), (\overline{u_{k_1-1}} \vee t_{k_1}) \end{array} \right.$$

   (b) If u occurs negatively one adds to C' the collection (II) of clauses below:

$$(II) = \left\{ \begin{array}{cc} \underline{\text{order of occurrence of } u} & \underline{\text{clauses of } (II)} \\ \downarrow & \downarrow \\ 1 & (\overline{v_1} \vee w_1 \vee \overline{w_2}, (v_1 \vee \overline{w_2}) \\ 2 & (\overline{v_2} \vee w_2 \vee \overline{w_3}, (v_2 \vee \overline{w_3}) \\ 3 & (\overline{v_3} \vee w_3 \vee \overline{w_4}, (v_3 \vee \overline{w_4}) \\ \vdots & \vdots \\ k_2 - 1 & (\overline{v_{k_2-1}} \vee w_{k_2-1} \vee \overline{w_{k_2}}, (v_{k_2-1} \vee \overline{w_{k_2}}) \end{array} \right.$$

   (c) Further, the following clauses are added according to $k_1 = 0$ or $k_2 = 0$, or $k_1 \neq 0$ and $k_2 \neq 0$,
      i. If($k_2 = 0$), then one adds $(u_{k_1} \vee \overline{t_{k_1}} \vee t_1), (\overline{u_{k_1}} \vee t_1)$ to C'.
      ii. If($k_1 = 0$), then one adds $(\overline{v_{k_2}} \vee w_{k_2} \vee \overline{w_1}), (v_{k_2} \vee \overline{w_1})$ to C'.
      iii. If(($k_1 \neq 0$) and ($k_2 \neq 0$)), then one adds $(u_{k_1} \vee \overline{t_{k_1}} \vee t_1), (\overline{u_{k_1}} \vee w_1), (\overline{v_{k_2}} \vee w_{k_2} \vee \overline{w_1}), (v_{k_1} \vee \overline{t_1})$ to C'.

This concludes the construction of the instance $I' = (U', C')$ of NAEQ3SAT$_{\underline{3}}$. Suppose that $I = (U, C)$ is a satisfiable instance of NAEQ3SAT. One proceeds by extending a satisfiable truth assignment of U to the corresponding variables of U'. Obviously, every corresponding clause of C in C' is satisfied and it is not difficult to see that once the truth value assigned to a variable $u \in U$ is fixed, then the same value is assigned to the new variables added to U' (due to u), satisfying all the new clauses added to C' (due to u).

Suppose $I' = (U', C')$ is a satisfiable instance of NAEQ3SAT$_{\underline{3}}$. Consider η a satisfiable truth assignment of U'. If $u \in U$ occurs more than three times in C, then, because η is satisfiable, all the variables $u_1, u_2, u_3, \ldots, u_{k_1}$ and $v_1, v_2, v_3, \ldots, v_{k_1}$ as well as the variables of T and W must have the same truth value in η. Hence this truth value shared by all these new variables can be assigned to the variable u in order to be a satisfiable truth assignment of U. □

Note that, in respect to finding $Opt_{\mathrm{CON}}(G)$, the following types of graphs are known [3] to have polynomial algorithms: bipartite graphs and trees ($Opt_{\mathrm{CON}}(G) = \frac{1}{2}$); complete graphs (($Opt_{\mathrm{CON}}(G) = \frac{1}{n}$); cycles ($Opt_{\mathrm{CON}}(G) = \frac{\lfloor \frac{n}{2} \rfloor}{n}$). Moreover, if G has a clique K_s of size s as a subgraph, any acyclic orientation ω over E is associated to concurrency $\frac{m}{p} \leq \frac{1}{s}$ [3].

From a general instance I for NAEQ3SAT$_{\underline{3}}$, a special instance G for MINCON can be constructed in polynomial time in the size of $I = (U, C)$. The special instance $G = (V, E)$ for MINCON constructed from a general instance $I = (U, C)$ for NAEQ3SAT$_{\underline{3}}$ satisfies: I is satisfiable if and only if E admits an acyclic orientation ω such that the concurrency $\gamma(\omega)$ satisfies $\gamma(\omega) = \frac{1}{3}$.

2.1 Building the special instance G

Graph $G = (V, E)$ contains 2 types of subgraphs: the *Truth Setting* (T_i) and the *Satisfaction Testing* (S_j) subgraphs defined in Figures 1(a) and 1(b).

Let $U = \{u_1, u_2, u_3, \ldots, u_n\}$ and $C = \{c_1, c_2, c_3, \ldots, c_m\}$ be the corresponding set of variables and collection of clauses of I. For each variable $u_i, i \in \{1, 2, 3, \ldots, n\}$ of U there is one corresponding subgraph T_i of G as defined in Figure 1(a) in vertices $u_i, \overline{u}_i, a_i$, b_i and d_i. For each clause $c_j, j \in \{1, 2, 3, \ldots, m\}$ of C there is one corresponding subgraph S_j of G as defined in Figure 1(b), a complete graph on 3 vertices. There is an additional set of vertices $W = \{w_1, w_2, w_3, \ldots, w_{n-1}\}$, and an additional set of edges $E_W = \{w_i a_i, w_i b_i : i \in 1, 2, 3, \ldots, n-1\} \cup \{w_{n-1} a_n, w_{n-1} b_n\}$ of edges emanating from the vertices of W.

The only part in the construction of G that depends on which literals occur in which clauses are the following sets of edges. For each clause $c_j \in C$, we select a different vertex x of S_j corresponding to a literal y of c_j and add the edge xy to $E(G)$. If c_j has only two literals and x is the vertex of S_j not yet connected to a vertex of a T_i, then the edge xy is added to $E(G)$, where y is the literal of c_j which occurs once. This concludes the construction of the special instance of CON. Figure 1(c) shows an example of a special instance G obtained from the satisfiable NAEQ3SAT$_{\underline{3}}$ instance $I = (U, C) = (\{u_1, u_2, u_3, u_4\}, \{(u_1 \vee \overline{u}_4), (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_3), (u_2 \vee \overline{u}_3), (u_3 \vee u_4), (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_4)\})$.

Lemma 2. *Let $G = (V, E)$ be a graph, such that there is an acyclic orientation ω with concurrency $\gamma(\omega) = \frac{m}{p} = \frac{1}{3}$. If for all $x \in V$ there is a triangle T on vertices x, y, z, such that T is a subgraph of G then, for each acyclic orientation of the associated SER dynamics, there is one vertex operating in T. Moreover $m = 1$ and $p = 3$.*

Proof. Let $\omega_1, \omega_2, \omega_3, \ldots \omega_p$ be the consecutive acyclic orientations of the SER dynamic, where ω_i is obtained from the reversion of the set of sinks of ω_{i-1},

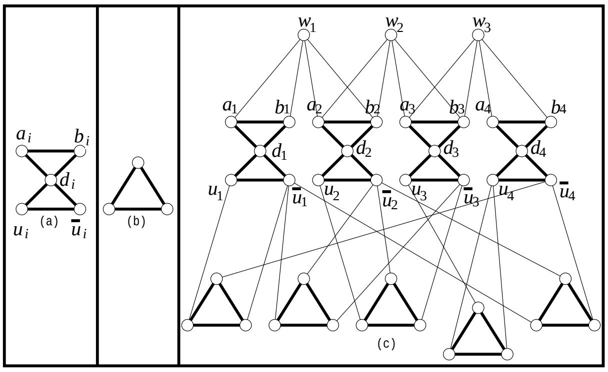

Fig. 1. Truth Setting (a) and Satisfaction Testing (b) subgraphs. And (c), instance $G = (V, E)$ of CONCURRENCY obtained from the NAEQ3SAT$_{\underline{3}}$ instance $I = (U, C) = (\{u_1, u_2, u_3, u_4\}, \{(u_1 \vee \overline{u}_4), (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_3), (u_2 \vee \overline{u}_3), (u_3 \vee u_4), (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_4)\})$.

$i \in \{2, 3, 4, \ldots, p\}$ and ω_1 is obtained from the reversion of the set of sinks of ω_p. Given i a positive integer, we call t_x^i the index of the orientation $\omega_{(t_x^i)}$ where x operates by the $i^{\underline{th}}$ time, $i \in \{1, 2, 3, \ldots, m\}$. Because x, y and z belong to T, then $t_x^{i+1} \geq t_x^i + 3$, $t_y^{i+1} \geq t_y^i + 3$ and $t_z^{i+1} \geq t_z^i + 3$. It is assumed that $t_x^i < t_y^i < t_z^i$.

Suppose $m = \alpha$ and $p = 3\alpha$. Since any pair in x, y, z cannot operate at the same time and each vertex operates α times, by the Dirichlet's Box Principle one has that, for each acyclic orientation of the associated SER dynamics, there is only one vertex operating in T. Since each vertex of G belongs to at least one triangle, after 3 operations one has the same set of sinks which defines $\alpha = 1$. □

Theorem 3. CONCURRENCY *is NP-complete for graphs with maximum degree 4.*

Proof. Given a general graph $G = (V, E)$, where $n = |V|$ and an acyclic orientation ω over E, Barbosa and Gafni [3] showed a n^6 polynomial algorithm to determine the associated $\gamma(\omega)$, hence the problem belongs to NP. It is shown here that, given an instance $I = U, C$ of NAEQ3SAT$_{\underline{3}}$ and the special instance G of CON, then I is satisfiable if and only if G has an acyclic orientation ω associated to a concurrency $\frac{1}{3}$. First, it is supposed that I is satisfiable. Let η be a satisfiable truth assignment for U. From η, the construction of an acyclic orientation ω over E associated to an amount of concurrency $\frac{1}{3}$ is defined by the following three steps:

1. If the literal x in η has value true, then vertex x in G is a sink in ω and vertex $\overline{x}$ is a source in ω.
2. For each $j \in \{1, 2, 3, \ldots, m\}$ select the vertex x of S_j corresponding to a non satisfied literal to be a sink and select the vertex y of S_j corresponding to a satisfied literal to be a source.
3. For all $i \in \{1, 2, 3, \ldots, n\}$ vertex a_i is a source and vertex b_i is a sink in ω.

This concludes the construction of ω. Next, it is proved that ω has concurrency exactly $\frac{1}{3}$ by defining the three orientations of the period.

Observe that by item 1, all vertices corresponding to true literals operate in ω. As η is a satisfiable truth assignment of an instance $I = (U, C)$ of NAEQ3SAT$_{\bar{3}}$, by item 2 there is a vertex operating in each S_j. By item 3 all vertices b_i operate in ω.

When the sinks of ω are reversed, an orientation ω_1 in which, by item 3, all vertices w_i operate, is obtained. Due to itens 1 and 3 all vertices w_i operate in ω_1. Since all vertices corresponding to the literals with value true were sinks in ω, one has that one additional vertex of each S_j is sink in ω_1. Observe that in ω_1 all vertices of all S_j's corresponding to the false literals of η have already operated. When the sinks of ω_1 are reversed an orientation ω_2 is obtained in which in each S_j, an additional vertex corresponding to a true literal is operating, and because all vertices d_i have operated in ω_1, the vertices of each T_i corresponding to the false literal in ω_2 are operating. Note also that the vertices a_i operate in ω_2.

Hence, all vertices of G operate once in these three orientations and the set of sinks and the set of sources of the orientation obtained from the reversion of the set of sinks of ω_2 is the same as ω. Thus, one has defined that ω belongs to a SER dynamics having concurrency $\frac{m}{p} = \frac{1}{3}$.

For the convenience of the reader, it is offered, in Figure 2, an example showing how to obtain an orientation ω Figure 2(a), and the corresponding orientations ω_1 Figure 2(b) and ω_2 Figure 2(c) defining a SER period characterised by $m = 1$ and $p = 3$. Next, it is proven that if there is an acyclic orientation ω over E with concurrency $\frac{1}{3}$, then I is satisfiable. Suppose that there is an acyclic orientation ω over E with concurrency $\frac{1}{3}$. Next, ω is used in order to yield a satisfiable truth assignment η for U.

Let $\omega_1, \omega_2, \omega_3$ be the consecutive acyclic orientations of the SER dynamics. Assume that a vertex $w_i \in W, i \in \{1, 2, 3 \ldots, n-1\}$ operates in ω_1. It is claimed that all vertices of W operate in ω_1. Assume for a moment that w_{i+1} is not a sink in ω_1. Then, neither a_{i+1} nor b_{i+1} is a sink in ω_1, so the triangle on vertices $a_{i+1}, b_{i+1}, w_{i+1}$ contradicts Lemma 2.

Since all vertices of W operate in a same orientation, consequently all vertices of C operate in ω_1. Hence, there is one literal vertex operating in ω_1 for each variable of U. This defines a truth assignment η for U by setting a literal as true if and only if the corresponding vertex of G is operating in ω_1.

Next, it is proven that η is a satisfiable truth assignment. Suppose by contradiction that there is a non satisfied clause c_j in $C, j \in \{1, 2, 3, \ldots, m\}$. Then, c_j either has all literals true or all literals false. Now suppose that the literals of c_j are all true. Let S_j be a clique on vertices x, y, z. Assume that x, y and z are sinks, respectively, in ω_1, ω_2 and ω_3. Hence, the vertex correponding to the literal of z does not operate, contradicting Lemma 2. Suppose that all literals of c_j are false. Then there is one vertex of S_j which does not operate, contradicting again Lemma 2. □

3 CONCURRENCY: a negative result for general graphs

Given a graph $G = (V, E)$, Bellare et al. proved in [4] that, unless P=NP, COLOURING cannot be approximable in a ratio less than $n^{\frac{1}{7}-\epsilon}$, for every $\epsilon > 0$.

Theorem 4. *Given a graph $G = (V, E)$ with n vertices, then, unless P=NP,* MIN^{-1}CON *cannot be approximable in a ratio less than $n^{\frac{1}{7}-\epsilon}$, for every $\epsilon > 0$.*

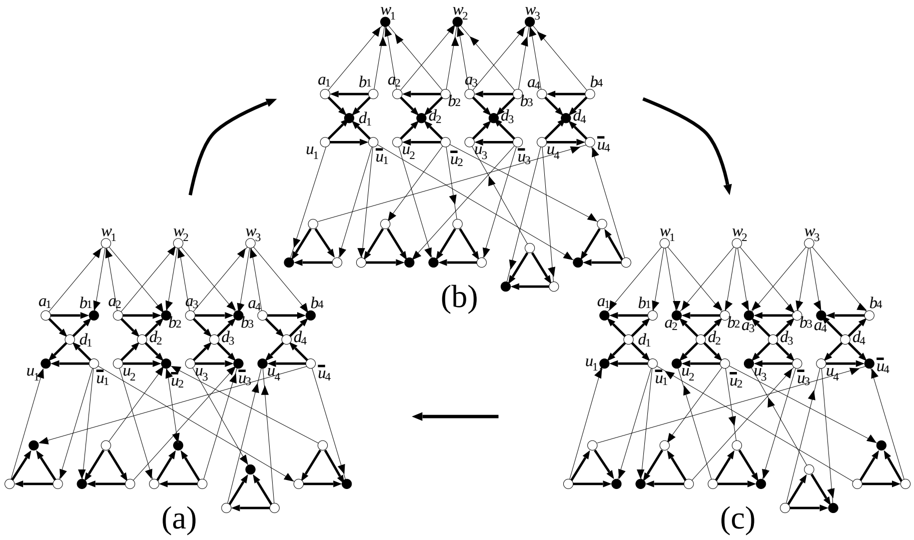

Fig. 2. Acyclic orientation ω with concurrency $\frac{1}{3}$ **(a)** defined for the instance $G = (V, E)$ of CONCURRENCY obtained from the satisfiable NAEQ3SAT$_{\overline{3}}$ instance $I = (U, C) = (\{u_1, u_2, u_3, u_4\},\ \{(u_1 \vee \overline{u}_4),\ (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_3),\ (u_2 \vee \overline{u}_3),\ (u_3 \vee u_4),\ (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_4)\})$, with the truth assignment $u_1 = \overline{u}_2 = \overline{u}_3 = u_4 = T$; and corresponding SER dynamics $(\omega, \omega_1, \omega_2)$ **(a)**, **(b)** and **(c)**. Black vertices depict operating vertices (sinks) and white vertices depict idle vertices.

Proof. It is enough to exhibit an L-reduction [9] from COLOURING to MIN^{-1}CON, because if a problem P_1 L-reduces to a problem P_2 and the problem P_2 has a polynomial r-approximation algorithm one has that, up to the constants, problem P_1 has a polynomial r-approximation algorithm.

According to the fundamental paper of Papadimitriou and Yannakakis [9], for L-reducing COLOURING to MIN^{-1}CON, one must yield f and g a pair of polynomial time algorithms in the size of G and a pair of positive reals α and β, such that, given an instance $G = (V, E)$ of COLOURING, algorithm f produces $f(G) = H = (V_H, E_H)$ an instance of MIN^{-1}CON, satisfying:

1. $Opt_{\mathrm{MIN}^{-1}\mathrm{CON}}(H) \leq \alpha Opt_{colouring}(G) = \alpha\chi(G)$ and;
2. that given a feasible solution η_H for MIN^{-1}CON in H, algorithm g obtains a feasible solution ξ_G for COLOURING in G such that
$$|\chi(G) - \xi_G| \leq \beta\, |Opt_{\mathrm{MIN}^{-1}\mathrm{CON}}(H) - \eta_H|.$$

Algorithm f, which yields the instance $H = f(G)$ of MIN^{-1}CON, is defined by adding an *universal vertex* v to G, $v \notin V(G)$ and for all $u \in V(G), vu \in E(H)$, i.e., $V(H) = V(G) \cup \{v\}$, and $E(H) = E(G) \cup \{vu : u \in v(G)\}$. This turns possible to prove the first and easiest part of the L-reduction.

Given a colouring of G with colours $1, 2, 3, \ldots, \chi(G)$, one can extend this colouring to H by assigning an extra colour $\chi + 1$ to v. Next, consider an orientation for the edges of G by setting the orientation of uw **from** u **to** w if u has colour c_u, w has colour c_w and $c_w < c_u$. Observe that the universal vertex v will be a single sink in a particular acyclic orientation in the period and the orientations of the edges from all vertices of G to v force that each set of sinks, respectively, with the colours $1, 2, 3, \ldots, \chi$ operate in sequence following the v operation and restart again with a new operation of v. This defines a period of length $p = \chi+1$. Hence, $Opt_{\text{MIN}^{-1}\text{CON}}(H) \leq p = \chi+1 \leq 2\chi = 2Opt_{colouring}(G)$. Hence, $\alpha = 2$ suffices and this conclude the first part of the L-reduction.

If η_H is a solution for MIN^{-1}CON in H with cost $\frac{p}{m}$, then necessarily $m = 1$. Because at the moment the universal vertex is a sink none vertex can operate too, one has that the vertices of $V(H) \setminus \{v\}$ become sinks in a sequence until v becomes sink again and none vertex operates twice before v. Hence, the sinks of $H \setminus \{v\}$ define a partition into $p-1$ independent sets for $V(G)$ and this is the definition of algorithm g. Hence, $|\chi(G) - (p-1)| \leq |\chi(G) + 1 - p| = |Opt_{\text{MIN}^{-1}\text{CON}}(H) - p|$, which shows that $\beta = 1$ suffices. □

4 An approximation algorithm for CONCURRENCY in graphs with maximum degree Δ

In 1941, Brooks [5] proved that if a graph $G = (V, E)$ has maximum degree Δ, chromatic number χ, is connected and is neither an odd cycle nor a complete graph, then $\chi \leq \Delta$. Later in 1975, Lovász [8] exhibited an polynomial algorithm which obtains a Δ-colouring of G. The algorithm introduced here is strongly based in the algorithm of Lovász and is defined as follows.

Algorithm A

Input: Graph $G = (V, E)$ with maximum degree Δ, where G is neither a complete graph nor an odd cycle on n vertices.
Output: Acyclic orientation $A(G)$ over E associated to concurrency $\frac{2}{\Delta}$.

1. Run the Lovász's Algorithm obtaining the partition into independent sets $(V_1, V_2, V_3, \ldots, V_\Delta)$ to V.
2. For each edge $e = uv \in E$, where $u \in V_i$ and $v \in V_j$ with $i < j$ orient e from v to u.

Lemma 5. *Let $G = (V, E)$ be a connected graph with maximum degree Δ, where G is neither an odd cycle nor a complete graph, let ω be an acyclic orientation over E and $\gamma(\omega) = \frac{m}{p}$ its associated concurrency, then $\frac{1}{\Delta} \leq \frac{m}{p} \leq \frac{1}{2}$.*
Proof. The first inequality shall be proven first. Note that, the transformation of sink vertices into source vertices of an acyclic orientation does not increase the maximum oriented path of the graph. Note also that a oriented path in $A(G)$ has size at most Δ. Then, the maximum number of consecutive steps of SER containing $A(G)$ in which a vertex is not a sink, is Δ. Hence, after $m\Delta$ steps all vertices of G will have operated at least m times, and so the p orientations of the period must have been concluded, i.e., $m\Delta \geq p$. The second inequality is proved by the observation that a vertex cannot operate in 2 consecutives orientations of a period with length p. Hence, $2m \leq p$ and this concludes the second inequality. □

Theorem 6. *The performance ratio of the algorithm A is at most $\frac{2}{\Delta}$.*

Proof. By Lemma 5 one has that $\frac{1}{\Delta} \leq \frac{m}{p} \leq \frac{1}{2}$. Hence, the performance ratio of A is bounded by, $R_A = \frac{|A(G)|}{|Opt_{\text{MIN}^{-1}\text{CON}}(G)|} \geq \frac{\frac{1}{\Delta}}{\frac{1}{2}} = \frac{2}{\Delta}$. □

5 Conclusion

Although one could consider that knowing in advance the degree of a target network to be scheduled by SER would facilitate the finding of its associated maximum concurrency, the first contribution of this work was to show that such a problem would still belong to NP-complete from maximum degree 4 input. Moreover, this is the first NP-completeness reduction of CON which had not been made from COLOURING.

Also, a negative result for CON in general graphs was introduced here: there is no approximation algorithms within $n^{\frac{1}{7}-\epsilon}$, unless P=NP. This result was obtained from the same well-known result for COLOURING, reinforcing the correlation between these two problems. From Brook's Theorem [5], it follows that COLOURING is a polynomial problem for instances with maximum degree 3 and it is well-known that COLOURING is NP-complete even for maximum degree 4 graphs [6]. It is left as an open problem whether concurrency is a polinomial or a NP-complete problem for instances with maximum degree 3.

Finally, notice that no approximation algorithm with a fixed ratio for finding maximum concurrency in general graphs is still not known. This paper has also introduced a $\frac{2}{\Delta}$-approximation algorithm for finding maximum concurrency in general connected graphs and, in particular, a $\frac{1}{2}$-approximation algorithm for finding maximum concurrency in connected graphs with maximum degree 4.

References

1. S. Arora, C. Lund, R. Montwani and M. Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the 33rd IEEE Symp. on Foundations of Computer Science*, 14-23, 1992.
2. V. C. Barbosa. An Atlas of Edge-Reversal Dynamics. Chapman&Hall/CRC, London, 2000.
3. V. C. Barbosa and E. Gafni. Concurrency in heavily loaded neighborhood-constrained systems. In *ACM Transactions on Programming Languages and Systems*, **11**, 562-584, 1989.
4. M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPS and non-approximability - towards tight results. *SIAM J. Comp.*, **27**, 804-915, 1998.
5. R. L. Brooks. On coloring nodes of a network. In *Proceedings of Cambridge Philos. Soc.*, **37**, 194-197, 1941.
6. M. R. Garey, D. S. Johnson and L. Stockmayer. Some simplified NP-complete problems. *Theoretical Computer Science*, **1**, 237-267, 1976.
7. M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, **45**, 19-23, 1993.
8. L. Lovász. Three short proofs in graph theory. *Journal of Combinatorial Theory B*, **19**, 111-113, 1975.
9. C. Papadimitriou and M. Yannakakis. Optimisation, approximation, and complexity classes. *J. of Computer and System Sciences*, **43**, 425-440, 1991.

This article was processed using the LaTeX macro package with LLNCS style