Applied Mathematical Sciences, Vol. 11, 2017, no. 44, 2151 - 2173
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ams.2017.77232

On the Synchronization of Small Sets of States

J. Andres Montoya

Universidad Nacional de Colombia
Departamento de Matemáticas
Bogotá, Colombia

Christian Nolasco

Universidad Nacional de Colombia
Departamento de Matemáticas
Bogotá, Colombia

Abstract

We study some problems related to the synchronization of finite state automata and The Černy's conjecture. We focus on the synchronization of small sets of states, and more specifically on the synchronization of triples. We argue that it is the most simple synchronization scenario that exhibits the same intricacies of the original Černy's scenario (all states synchronization). Thus, we argue that it is complex enough to be interesting, and tractable enough to be studied via algorithmic tools and experiments. We use those tools to establish a long list of facts related to those issues.

1 Introduction

We investigate the synchronization of deterministic finite state automata (DFAs, for short), focusing on *Černy's Conjecture*. Recall that a DFA is a triple $\mathcal{M} = (Q_{\mathcal{M}}, \Sigma_{\mathcal{M}}, \delta_{\mathcal{M}})$ such that:

- $Q_{\mathcal{M}}$ is a finite set, the set of internal states of automaton $\mathcal{M}$.
- $\Sigma_{\mathcal{M}}$ is a finite alphabet, the input alphabet of $\mathcal{M}$.

- $\delta_{\mathcal{M}}$ is the transition function of $\mathcal{M}$, which is a function from $\Sigma_{\mathcal{M}} \times Q_{\mathcal{M}}$ to $Q_{\mathcal{M}}$.

Let $\mathcal{M} = (Q_{\mathcal{M}}, \Sigma_{\mathcal{M}}, \delta_{\mathcal{M}})$ be a DFA. We use the symbol $\Sigma_{\mathcal{M}}^*$ to denote the set of finite strings over the *alphabet* $\Sigma_{\mathcal{M}}$. The function $\widehat{\delta_{\mathcal{M}}} : \Sigma_{\mathcal{M}}^* \times Q_{\mathcal{M}} \to Q_{\mathcal{M}}$, defined by the equation:

$$\widehat{\delta_{\mathcal{M}}}(w_1...w_n, q) = \delta_{\mathcal{M}}\left(w_n, \widehat{\delta_{\mathcal{M}}}(w_1...w_{n-1}, q)\right),$$

determines the state that is reached when automaton $\mathcal{M}$ scans the string $w_1...w_n$, beginning in the state q.

We say that an automaton $\mathcal{M}$ is *synchronizing,* if and only if, there exists a *synchronizing string* $w \in \Sigma_{\mathcal{M}}^*$, such that for all $p, q \in Q_{\mathcal{M}}$, the equality

$$\widehat{\delta_{\mathcal{M}}}(w, p) = \widehat{\delta_{\mathcal{M}}}(w, q)$$

holds. There are many works related to this notion (see [22] for an account). Most of those works are focused on the study of minimal synchronizing strings. It is easy to prove that given a n-state synchronizing automaton $\mathcal{M}$ and a minimal synchronizing string w, the length of w is $O(n^3)$. Černy conjectured that $|w| \leq (n-1)^2$ (see [5]), it is the famous Černy's Conjecture.

Given an automaton $\mathcal{M}$, and given $q_1, ..., q_k \in Q_{\mathcal{M}}$, a *synchronizing string for those k states*, is a string w, such that for all $i, j \leq k$, the equality

$$\widehat{\delta_{\mathcal{M}}}(w, q_i) = \widehat{\delta_{\mathcal{M}}}(w, q_j)$$

holds. We would like to shed some light on Černy's conjecture by studying the synchronization of small sets of states.

Synchronization is an ubiquitous topic in computer science [11]. We say that a system is synchronizing if all their components can be lead to work at the same pace and after the same goal. Synchronizing systems are resilient: If an error occurs, and the system is taken out of control, then it can be driven to a certain specific state regardless of the uncertainty concerning its actual state. Synchronizing automata constitute an elementary model of synchronizing system: If one of those automata is taken out of control, and one cannot determine its actual state, then he can drive the automaton to a specific state using a short synchronizing string (also called a *reset word*) to this end. Thus, it is not a surprise if synchronizing automata have found its way into the control theory of discrete systems [17].

If one has to reset a given automaton, it is better to use a short reset word. Therefore, the study of minimal synchronizing strings is one of the most studied topics related to synchronizing automata. It is known that the problem of computing minimal synchronizing strings is NP-hard [7], and that it is also hard to approximate [9]. It is also known that the length of a minimal synchronizing strings is upperbounded by $\frac{n(n-1)^2}{2}$. This upperbound can achieved by applying a naive polynomial time synchronizing strategy:

Given $q_1, ..., q_n$, the n states of the automaton, compute a string w_1 that synchronizes the pair q_1, q_2. Then, compute a string w_2 that synchronizes a pair from the set $\left\{\widehat{\delta}(q_1, w_1), ..., \widehat{\delta}(q_n, w_1)\right\}$. Continue in this way until a reset word $w_1 \cdot w_2 \cdot \cdots \cdot w_{n-1}$ is computed.

We use the term *reduction to pairs* to denote the above synchronization strategy. It is natural to ask: Does there exist an efficient synchronizing strategy outperforming the above naive strategy? It can be argued that it is the meaning of Černy's conjecture [5], which states that the minimal reset length of a n-state synchronizing automaton is upperbounded by $(n-1)^2$. Černy's conjecture suggests that there must exist synchronizing strategies achieving short reset lengths, (close to Černy's bound) and which must outperform the naive reduction to pairs.

Thus, in some sense, the investigations related to Černy's conjectured refer the following question: Is it possible to beat brute force (reduction to pairs) synchronization? Notice that it is a kind of question that is ubiquitous in theoretical computer science.

Suppose that one wants to synchronize three states p, q, r. Then, he can use the naive strategy consisting in synchronizing the pair p, q, using a string w_1, and then the pair $\widehat{\delta}(p, w_1), \widehat{\delta}(r, w_1)$ using a second string w_2. If w_1 and w_2 are minimal synchronizing strings for their respective pairs, then the inequality $|w_1 w_2| \leq n(n-1)$ must hold. Can one do better? Notice that we are asking the same kind of question as above: Can one beat the naive synchronization strategy called reduction to pairs?

We are firmly convinced that the study of optimal synchronization strategies for triples of states can shed some light on the intricacies of synchronization, and hence we believe that those investigations could be a key ingredient in the solution of the long-standing Černy's problem.

Structure of the work, contributions and relations to previous work.

We studied the synchronizing times of k-tuples of states, focussing on the synchronizing times of the hardest k-tuples. Moreover, we focus on the case $k = 3$. There are some previous works related to subset synchronization. The first one is the work of Kirnasov [14], which contains some upperbounds on the synchronization times of k-tuples and sharp estimates for the expected time of synchronizing r states, provided that $r \leq \log(\log(n))$. A second work is the paper of Gonze and Jungers [10], who studied the synchronizing times of k-tuples, focussing on the synchronizing times of the easiest tuples. Vorel [23], studied the synchronizing times of the hardest k-tuples, but it is important to remark that he chose to work with general automata, including nonsynchronizing ones. Vorel obtained sharp upper bounds for the synchronizing times of k-tuples, but those upperbounds are meaningless in our framework, given that those bounds are exponential and can only be achieved by nonsynchronizing automata (given the, previously discussed, cubic upperbound). Finally, one has to mention the work of Pereira [20], who studied exactly the same problem we study in this paper, which is the problem of upperbounding the synchronizing times of the hardest k-tuples. Pereira formulate a general conjecture which entails our $\frac{2}{3}$-*Černy's Conjecture* (see below). We think that focussing on the case $k = 3$, and relaxing a little bit the conjectured inequality (as we have done), allow one to focus on the core problems of subset synchronization.

We have chosen to try an experimental approach:

We defined an scenario (triples synchronization), which is the most simple scenario where the core problems of automata synchronization can be investigated. We identify some algorithmic tools that can be used in our investigation, and we use them in a consistent way. Moreover, we study the algorithmic obstacles that appear when one considers more general scenarios.

It is worth to remark that there are some previous works related to synchronizing automata and Černy's conjecture that have tried some different experimental approaches, see for instance the papers [15], [16], and [21]. It seems that experimental approaches can be useful in the study of synchronizing automata, take into account that the questions that are being investigated refer a special type of discrete dynamics which can be easily and efficiently simulated by computer means.

2 Synchronizing Small Sets of States: Synchronizing Times

Let $\mathcal{M}$ be a synchronizing automaton and let $q_1, q_2, ..., q_k \in Q_{\mathcal{M}}$, we use the symbol $s(\mathcal{M}, q_1, q_2, ..., q_k)$ to denote the length of a minimal synchronizing string for those k states. By an abuse of language we say that $s(\mathcal{M}, q_1, q_2, ..., q_k)$ is the *synchronizing time* of the tuple $\{q_1, q_2, ..., q_k\}$. We use the symbol $s_k(\mathcal{M})$ to denote the quantity

$$\max(\{s(\mathcal{M}, q_1, q_2, ..., q_k)\} : q_1, q_2, ..., q_k \in Q_{\mathcal{M}}),$$

which is equal to the synchronizing time required by the *hardest to synchronize* k-tuple of states of automaton $\mathcal{M}$. We use the symbol sw_k to denote the function defined by

$$sw_k(n) = \max(\{s_k(\mathcal{M}) : \mathcal{M} \text{ is a } n\text{-state synchronizing automaton}\}).$$

We want to study the sequence $\{sw_k\}_{k \geq 2}$.

2.1 On the Synchronization of Pairs

We begin considering the case $k = 2$.

Let $\mathcal{M}$ be a n-state synchronizing automaton, and let q_1, q_2 be two states of $\mathcal{M}$. *The square automaton of* $\mathcal{M}$, denoted by $\mathcal{M}^2$, is the automaton defined by:

- $Q_{\mathcal{M}^2} = \{\{p, q\} : p, q \in Q_{\mathcal{M}}\}$ (p and q could be equal).
- $\Sigma_{\mathcal{M}} = \Sigma_{\mathcal{M}^2}$, and given $c \in \Sigma_{\mathcal{M}}$ the equality

  $$\delta_{\mathcal{M}^2}(c, \{p, q\}) = \{\delta_{\mathcal{M}}(c, p), \delta_{\mathcal{M}}(c, q)\}$$

  holds.

A minimal synchronizing string for p and q corresponds to a minimal path in the transition digraph of $\mathcal{M}^2$, which connects the node $\{p, q\}$ and the set

$$\Delta_{\mathcal{M}^2} = \{\{p, q\} \in Q_{\mathcal{M}^2} : p = q\}.$$

The length of such a path is upperbounded by the size of $Q_{\mathcal{M}^2} \backslash \Delta_{\mathcal{M}^2}$, which is equal to $\frac{n(n-1)}{2}$. Then, we have that for all $p, q \in Q_{\mathcal{M}}$, the inequality $s(\mathcal{M}, q_1, q_2) \leq \frac{n(n-1)}{2}$ holds.

We can prove that the above upperbound is tight, we can prove that for infinitely many n's there exists a n-state automaton, say $\mathcal{C}_n$, such that $s_2(\mathcal{C}_n) = \frac{n(n-1)}{2}$. This task, (which at first sight could seem easy), is by no means trivial. It is known that the expected value of $s_{|Q_{\mathcal{M}}|}(\mathcal{M})$ is $O(|Q_{\mathcal{M}}|)$ [13]. It means that most synchronizing automata can be synchronized using short strings of linear size. And then, it happens that for most synchronizing automata all their pairs of states can be synchronized with short strings of linear size.

Notice that a n-state automaton achieving the bound $\frac{n(n-1)}{2}$ must exhibit the following singular feature: There exists a pair of states, which visits all the others pairs of states before it gets synchronized. The canonical construction of a sequence exhibiting such a singular feature is the following one:

Let n be a fixed positive integer.

- Use the n states labelled $0, 1, ..., n-1$ to built a directed cycle of length n, whose arrows are colored with the letter a, and which are directed according to the cyclic order $0 \leq 1 \leq \cdots \leq n-1 \leq 0$.
- Add a loop colored with the letter b to any node in the set $\{1, ..., n-1\}$. Add an arrow $(0, 1)$, colored with the same letter b.

We use the symbol $\mathcal{C}_n$ to denote the synchronizing automaton that is obtained from the above construction. Automaton $\mathcal{C}_n$ is best described by the graphic in figure 1.

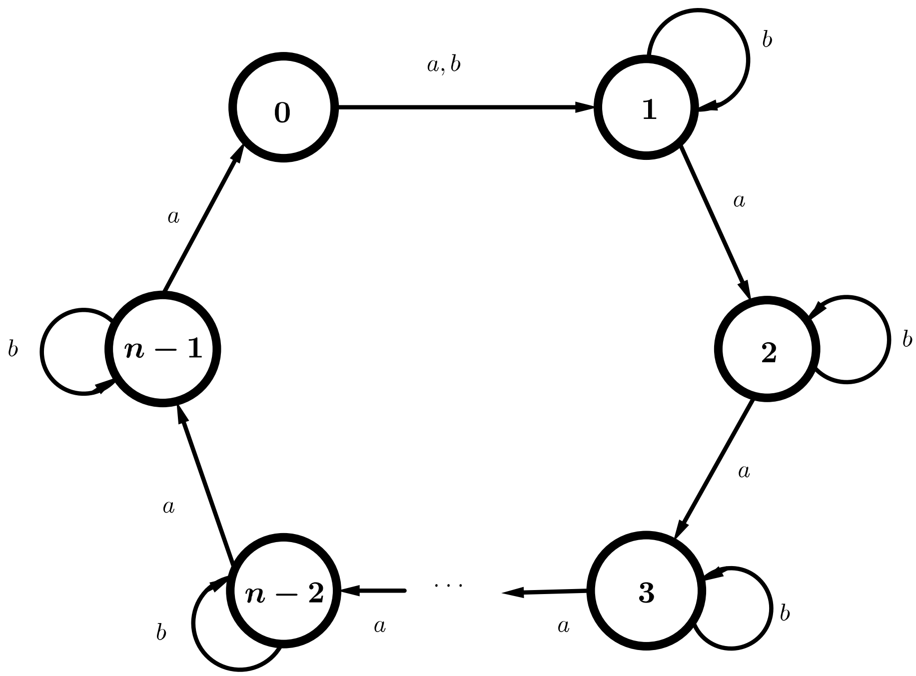

Figure 1: Automaton $\mathcal{C}_n$

Theorem 1 *Given $n \geq 2$, if n is even the equality $sw_2(n) = \frac{n(n-1)}{2}$ holds.*

Proof. Consider the pair $\left(1, \frac{n}{2}+1\right)$. It is easy to check that this pair will visit any possible pair of states before it gets synchronized. Then, $s_2(\mathcal{C}_n) = \frac{n(n-1)}{2} \leq sw_2(n)$, and the theorem is proved. ■

Thus, $k = 2$ is a solved case.

2.2 Synchronization of triples

In this section we study the function $sw_3(n)$. This work is closely related to the work of Gonze and Jungers [10], who studied a complementary problem: They studied the synchronizing time required by the *easiest to synchronize* triples of states.

How large could be the synchronizing time of a given triple? A first upper bound is $n(n-1)$. It follows from the following fact: In order to synchronize three states (say p, q and r), one can choose to synchronize the first two states, (which costs at most $\frac{n \cdot (n-1)}{2}$ characters), and then to synchronize the (at most) two remaining states. Thus, we have.

Theorem 1 *Given $n \geq 2$, the inequality $sw_3(n) \leq n(n-1)$ holds.*

In order to get a lower bound, one could try the experimental method:

He picks a sequence of synchronizing automata, say $\{\mathcal{M}_n\}_{n \geq 2}$, and then he estimates the value of $s_3(\mathcal{M}_n)$.

Notice that the inequality $s_3(\mathcal{M}_n) \leq sw_3(|Q_{\mathcal{M}_n}|)$ holds by definition. Now, if one wants to get a nontrivial lower bound using such a naive idea, he must choose a sequence of *slowly synchronizing automata* [1]. We chose to work with the sequence $\{\mathcal{C}_n\}_{n \geq 2}$, which is the paradigmatic example of a hard to synchronize sequence of automata (see below).

Theorem 2 *Given n, we have that $s_3(\mathcal{C}_n) \leq \frac{2}{3}n^2$, and for all $n = 3k$ the equality $s_3(\mathcal{C}_n) = \frac{2}{3}n(n-1)$ holds.*

Proof. Let $G(\mathcal{C}_n)$ be the underlying digraph of $\mathcal{C}_n$. An interval is a subset of $Q_{\mathcal{C}_n}$ which is connected in $G(\mathcal{C}_n)$.

Let I be an interval of size m, it can be clockwise rotated until it gets equal to $\{0, ..., m-1\}$. To this end, one can apply a string a^{k_I}, where k_I is a suitable positive integer that is smaller than n. Then, if he applies letter b, he gets the interval $\{1, ..., m-1\}$, that is: There exists $k_I \leq n-1$ such that $\widehat{\delta}\left(I, a^{k_I}b\right)$ is equal to $\{1, ..., m-1\}$. Notice that this contraction procedure can be iterated a large enough number of times until the interval gets equal to a singleton, which corresponds to the interval being synchronized. Observe that given $A \subset Q_{\mathcal{C}_n}$, synchronizing A is the same as synchronizing the minimal length interval containing A. Given a triple p, q, r, there exists an interval I such that $p, q, r \in I$ and $|I| \leq \frac{2}{3}n + 1$. The synchronization of I corresponds to $|I| - 1$ contractions, and the cost (in characters) of each one of those contractions is upperbounded by n. Then, we have that

$$s(\mathcal{C}_n, p, q, r) \leq n(|I| - 1) \leq \frac{2}{3}n^2.$$

Let $n = 3k$, and let $I = \{1, k+1, 2k+1\}$. It is easy to check that the synchronization time of the triple I is equal to $\frac{2}{3}n(n-1)$ (see [20]), it implies that $s(\mathcal{C}_n, 1, k+1, 2k+1) = \frac{2}{3}n(n-1)$. ■

Theorem 1 $sw_3(n) \leq \frac{2}{3}n(n-1)$.

2.3 The synchronizing time of k-tuples and Černy's conjecture

Notice that for all k, the inequality $sw_k(n) \leq (k-1)\frac{n(n-1)}{2}$ holds. On the other hand, it is not hard to prove the following proposition (see [20])

Theorem 3 $s_k(\mathcal{C}_n) \leq \left(\frac{k-1}{k}\right) n(n-1)$

Based on this result, we propose the following generalization of Černy's conjecture.

Theorem 2 ***(Parameterized Černy's Conjecture)***
Let $\mathcal{M}$ be a n-state synchronizing automaton and let k be an integer, the inequality $s_k(\mathcal{M}) \leq \left(\frac{k-1}{k}\right)(n(n-1))$ holds.

Next proposition shows that the above conjecture is stronger than Černy's conjecture

Theorem 4 *If the parameterized Černy's conjecture holds, then Černy's conjecture must also hold.*

Proof. Suppose that for all n, for all n-state synchronizing automaton $\mathcal{M}$, and for all $k \leq n$ the inequality

$$s_k(\mathcal{M}) \leq \left(\frac{k-1}{k}\right)(n(n-1))$$

holds. Then, given $\mathcal{M}$ it happens that

$$s_n(\mathcal{M}) \leq \left(\frac{n-1}{n}\right) n(n-1) = (n-1)^2,$$

and then, the inequality $s_n(\mathcal{M}) \leq (n-1)^2$ must hold. ■

The above proposition indicates that studying the synchronization of small sets of states could be a fruitful way of studying the synchronization of automata and the related Černy's conjecture

3 Focussing on triples

We showed, in last section, that studying the synchronizing times of tuples can pave the way to solve Černy's problem. To this end, it is necessary to study the synchronizing times of k-tuples for varying k. Nevertheless, we decided to focus on the case $k = 3$. Why? We consider that the synchronization of triples is at the very hearth of Černy's problem. Notice that the problem of synchronizing triples is the simplest synchronization scenario where it makes sense to ask our motivating question: Does exist a synchronization strategy outperforming the naive reduction to pairs? Moreover, as we will see, we count with an algorithmic tool that can be used to make some experiments. Let H-TRIPLES be the problem defined by:

- *Input:* $\mathcal{M}$, *where* $\mathcal{M}$ *is a synchronizing automaton.*
- *Problem: Compute the hardest triple of* $\mathcal{M}$ together with a minimal synchronizing string for this triple.

Theorem 2 *H-TRIPLES can be solved in time* $O(n^9)$.

Proof. Let $\mathcal{M}$ be a n-state synchronizing automaton, and let $\mathcal{M}^3$ be the *triples automaton* defined by:

- $Q_{\mathcal{M}^3} = \{\{p,q,r\} : p,q,r \in Q_{\mathcal{M}}\}$ (p,q and r are not necessarily pairwise different).
- $\Sigma_{\mathcal{M}} = \Sigma_{\mathcal{M}^3}$, and given $c \in \Sigma_{\mathcal{M}}$ the equality

$$\delta_{\mathcal{M}^3}(\{p,q,r\}, c) = \{\delta_{\mathcal{M}}(p,c), \delta_{\mathcal{M}}(q,c), \delta_{\mathcal{M}}(r,c)\}$$

  holds.

Computing a minimal synchronizing string for states p,q,r is the same as computing a minimal path in the transition digraph of $\mathcal{M}^3$ and which connects the state $\{p,q,r\}$ with the set $\Delta_{\mathcal{M}^3} = \{A \in Q_{\mathcal{M}^3} : |A| = 1\}$. Notice that the size of $\mathcal{M}^3$ is $O(n^3)$, hence the later problem can be solved in time $O(n^6)$ using Dijkstra's algorithm [6]. The computation of the hardest triple of $\mathcal{M}$ can be carried out in time $O(n^9)$. ■

From now on, we use the symbol TRIPLES to denote the above algorithm. Algorithm TRIPLES can be easily implemented. We have implemented it and we have sued the implementation to perform some experiments.

It is important to remark that one can use a similar construction to prove that, given $k \geq 3$, computing the hardest k-tuple together with a minimal synchronizing string for this k-tuple can be done in time $O\left(n^{3k}\right)$. It seems efficient. However, we have to observe that parameter k occurs in the exponent, which is, in some sense, the worst possible behavior. We have to ask: Is this occurrence of k unavoidable? Notice that it if were the case, we could not claim that the problem of computing the hardest k-tuple is feasible (it becomes unfeasible for large values of k).

3.1 Algorithmic Barriers: Parameterized complexity

We show in this section that if k is not bounded above by a constant, then the problem of computing the hardest k-tuple is not feasible. To this end, we use the tools of parameterized complexity theory. We refer the reader to [8] for the basics of parameterized complexity.

First, we define a parameterized decision problem related to the above problem. Let *p-Synch* be the parameterized problem defined by:

- *Input:* $(\mathcal{M}, \{q_1, ..., q_k\}, r, k)$, *where* k, r *are positive integers and* $q_1, ..., q_k$ *are* k *states of* $\mathcal{M}$.
- *Parameter:* k.
- *Problem: Decide if there exists a synchronizing string of length* r *for the states* $q_1, ..., q_k$.

Is *p-Synch fix parameter tractable*? Recall that a parameterized problem is fix parameter tractable, if and only if, it can be solved in time $O(f(k) \cdot n^c)$, for some function f and some constant c (see reference [8]). We prove that the problem *p-Synch* is NWL hard. The class NWL is supposed to be the parameterized analogue of PSPACE [12] and [8]. This class is located above of the W-hierarchy, and hence our result implies that *p-Synch* is W[t] hard for all $t \geq 1$ (see [8]).

We prove that *p-Synch* is NWL hard by exhibiting a fpt reduction of *The Parameterized Longest Common Subsequence Problem* (*p-LCS*, for short) in the problem *p-Synch*. Guillemot [12] proved that *p-LCS* is hard for NWL. The parameterized longest common subsequence problem is the parameterized problem defined by:

- *Input:* $(\{w_1, ..., w_k\}, \Sigma, m)$; *where* Σ *is a finite alphabet,* $w_1, ..., w_k \in \Sigma^*$ *and* m *is a positive integer.*
- *Parameter:* k
- *Problem: Decide if there exists a string* $w \in \Sigma^*$, *such that for all* $i \leq k$ the *string* w *is a substring of* w_i, *and such that* $|w| = m$.

Theorem 3 *p-Synch is* NWL *hard.*

Proof. Let $X = (\{w_1, ..., w_k\}, \Sigma, m)$ be an instance of $p\text{-}LCS$. Given $i \leq k$, we use Baeza-Yates construction (see [2]) to compute a DFA, say $\mathcal{M}_i$, such that automaton $\mathcal{M}_i$ accepts the language constituted by all the subsequences of w_i. It is important to remark, at this point, that the size of $\mathcal{M}_i$ is bounded above by $|w_i| + 1$. Notice that for all $i \leq k$, we are using the automaton $\mathcal{M}_i$ as a language acceptor, it implies that for all $i \leq k$, there exists a marked state (the initial state of $\mathcal{M}_i$) which we denote with the symbol q_0^i. Moreover, for all $i \leq k$, there exists a nonempty subset of Q_i, denoted with the symbol A_i, and which is equal to the set of accepting states of automaton $\mathcal{M}_i$.

We use the set $\{\mathcal{M}_i : i \leq k\}$ to define an automaton $\mathcal{M} = (\Omega, Q, \delta)$ in the following way:

1. $\Omega = \Sigma \cup \{d\}$, where $d \notin \Sigma$.
2. $Q = \left(\bigsqcup_{i \leq k} Q_i \right) \sqcup \{q, p_1, ..., p_{m+1}\}$, where $\sqcup$ denotes disjoint union, and given $i \leq k$, the symbol Q_i denotes the set of states of the automaton $\mathcal{M}_i$. Moreover, we have that $q_1 \notin \bigsqcup_{i \leq k} Q_i$.
3. The transition function of $\mathcal{M}$, which we denote with the symbol δ, is defined as follows

$$\delta(p, a) = \begin{cases} \delta_i(p, a), \text{ if } p \in Q_i \text{ and } a \neq d \\ q, \text{ if } p \in \bigsqcup_{i \leq k} A_i \text{ and } a = d \\ p_1, \text{ if } p \in (Q_i \backslash A_i) \text{ and } a = d \\ q, \text{ if } p = q \\ p_{j+1}, \text{ if } p = p_j,\ j < m+1, \text{ and } a \in \Sigma \\ p_1, \text{ if } p = p_j,\ j < m+1, \text{ and } a = d \\ q, \text{ if } p = p_{m+1} \text{ and } a = d \\ p_1, \text{ if } p = p_{m+1} \text{ and } a \neq d \end{cases}$$

Let $Y(X)$ be equal to $\left(\mathcal{M}, \left\{q_0^1, ..., q_0^k, p_1\right\}, k+1, m+1\right)$, we have that $Y(X)$ it is the output of the reduction. It is easy to check that $X \in p\text{-}LCS$, if and only if, the states $q_0^1, ..., q_0^k, p_1$ can be synchronized in time $m+1$, that is: It can be easily checked that $X \in p\text{-}LCS$, if and only if, $Y(X) \in p\text{-}Synch$. Then, we have that $p\text{-}Synch$ is NWL-complete. ■

Let $p\text{-}Synch[bin]$ be the restriction of $p\text{-}Synch$ to the set of instances

$$\{(\mathcal{M}, \{q_1, ..., q_k\}, l, k) : \mathcal{M} \text{ is a binary automaton}\}.$$

The construction used in [3] allows one to reduce the problem $p\text{-}Synch$ to its restriction $p\text{-}Synch[bin]$. Thus, we can conclude that the synchronization of small sets of states is NWL hard, even for fixed alphabets of size larger than 1. We can also conclude that the occurrence of parameter k in the exponent is unavoidable (unless extraordinary things happen). Then, if we want to use efficient algorithmic tools to perform massive experiments regarding the synchronization of k-tuples, we will have to focus on small values of k.

4 The $\frac{2}{3}$-Černy's Conjecture

Let us relax a little bit Conjecture 1.

Theorem 3 ***(The $\frac{2}{3}$-Černy's Conjecture)***
For all n, the inequality $sw_3(n) \leq \frac{2}{3}n^2$ holds.

Thus, we conjecture that the synchronizing time of triples is two thirds of the upperbound that is achieved using the reduction to pairs algorithm, or equivalently: Optimal triple synchronization strategies could beat pair synchronization by a ratio that is equal to $\frac{2}{3}$.

4.1 Providing evidence: Strong refutations and menacing sequences

A *weak refutation* is an automaton $\mathcal{M}$ such that

$$s_3(\mathcal{M}) > \frac{2}{3}|Q_{\mathcal{M}}|^2 .$$

A *strong refutation* is a sequence $\left\{\mathcal{M}_{k(i)}\right\}_{i\geq 1}$ such that for all $i \geq 1$ we have that:

1. $k(i+1) > k(i)$.
2. $\left|Q_{\mathcal{M}_{k(i)}}\right| = k(i)$.
3. $s_3\left(\mathcal{M}_{k(i)}\right) > \frac{2}{3}\left|Q_{\mathcal{M}_{k(i)}}\right|^2$

We are interested in proving that there do not exist strong refutations.

Suppose that The $\frac{2}{3}$-Černy's Conjecture is wrong. Strong refutations cannot be found by chance. Recall that most synchronizing automata can be synchronized with short strings of linear length [13]. We say that a sequence of synchronizing automata is *superlinear,* if and only if, the synchronizing time required by this sequence cannot be bounded above by a linear function. The sequence $\{\mathcal{C}_n\}_{n\geq 2}$ is an important example of a superlinear sequence.

Theorem 1 *A sequence* $\{\mathcal{M}_{k(i)}\}_{i\geq 1}$ *is an extremal sequence, if and only if, the inequality* $s_{k(i)}\left(\mathcal{M}_{k(i)}\right) \geq \frac{2}{3}k\left(i\right)^2$ *holds.*

Notice that any strong refutation is an extremal sequence. Notice also that any extremal sequence is superlinear.

Extremal sequences are very scarce [19]. A first example of an extremal sequence is Černy's sequence, but we know that it cannot be a strong refutation of the $\frac{2}{3}$-Černy's Conjecture. We also know of the 8 extremal sequences included in the paper [1], the two sequences included in the reference [16], and one further sequence included in the reference [4]. We can show that all those 11 sequences are not strong refutations.

To begin with we notice that five out of the eight sequences included in [1] are sequences of circular automata. It can be proved that The $\frac{2}{3}$-Černy's Conjecture holds for circular automata [20]. Then, we know that those five sequences cannot give place to strong refutations. It follows easily from the work of Pereira [20] that the remaining three sequences, which are denoted with the symbols $\mathcal{D}"_n, \mathcal{H}_n$ and $\mathcal{E}_n$ (in the references [1] and [20]) are not strong refutations.

We check that the sequences included in the references [16] and [?] are not strong refutations.

Kisilewicz and Szykula [16] introduced two extremal sequences of automata, which they denote with the symbols $\mathcal{M}$ and $\mathcal{M}$', and which are defined in the following way:

Given $n \geq 3$, we use the symbol $\mathcal{M}_n$ ($\mathcal{M}_n$') to denote the n^{th} automaton in the sequence $\mathcal{M}$ ($\mathcal{M}$'). First, we define the automaton $\mathcal{M}_n = (Q_n, \Sigma, \delta_n)$. The set Q_n is equal to $\{1, ..., n\}$, the alphabet Σ is equal to $\{a, b, c\}$, and the transition function δ_n is defined by

$$\delta_n\left(i,a\right) = \begin{cases} i+1 \text{ if } 1 \leq i \leq n-1 \\ 2 \text{ if } i = n \end{cases}$$

$$\delta_n\left(i,b\right) = \begin{cases} 2 \text{ if } i = 1 \\ i \text{ if } 2 \leq i \leq n \end{cases}$$

$$\delta_n\left(i,c\right) = \begin{cases} n \text{ if } i = 1 \\ i \text{ if } 2 \leq i \leq n-1 \\ 1 \text{ if } i = n \end{cases}$$

The definition of $\mathcal{M}_n$'$= (Q_n, \Sigma, \delta_n$') is almost the same, except that the transition function δ_n' behaves different when the second argument (the character being read) is equal to c, in this case we have

$$\delta_n\text{'}\left(i,c\right) = \begin{cases} i \text{ if } 1 \leq i \leq n-1 \\ 1 \text{ if } i = n \end{cases}$$

Those two sequences of automata are best described by the following two graphics

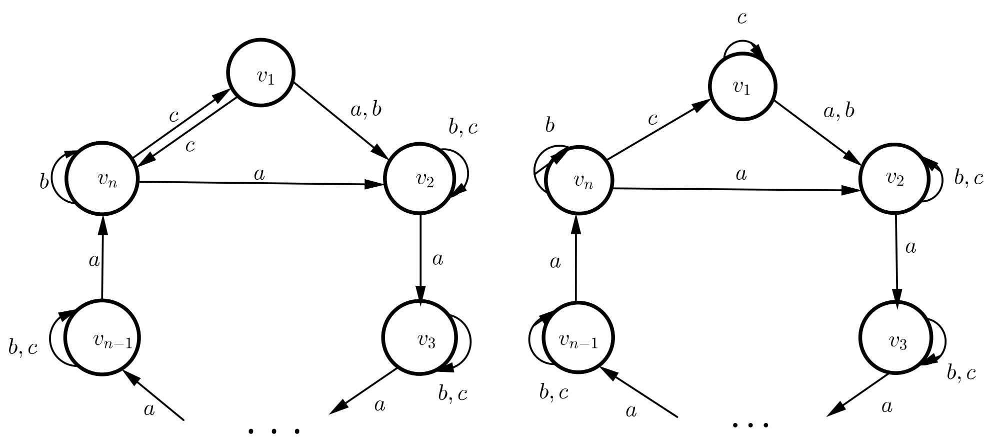

Figure 2: The first and second sequence of Kisilewicz-Szykula

One interesting fact about those two sequences is that they are the first (and unique) quadratic sequences of irreducible non-binary automata registered in the literature [16]. We prove that those both sequences are not strong refutations.

Theorem 5 *Sequences $\mathcal{M}$ and $\mathcal{M}$' are not strong refutations.*

Proof. We make the proof for the sequence $\mathcal{M}$, the proof for $\mathcal{M}$' is very similar.

Let $n \geq 1$ and let $\mathcal{M}_n$ be the n-th automaton in the sequence $\mathcal{M}$. We suppose that $n = 3k$, the proof for the remaining two cases is very similar. Let p, q and r be the three states to be synchronized. There exists $m \leq n-1$ such that $\widehat{\delta}(\{p, q, r\}, a^m)$ is a triple l, g, f fulfilling the following conditions:

1. $l = n$
2. $g \leq f$
3. $f \leq 2k$

In order to synchronize the triple l, g, f one can use the string $cb\left(a^{n-2}cb\right)^{f-1}$. Then, given p, q and r, there exists $m \leq n-2$ such that the string $a^m cb\left(a^{n-2}cb\right)^{f-1}$ synchronizes the triple p, q and r. The length of this string is bounded above by $n(2k-1)+n$. Notice that

$$\begin{aligned} n(2k-1)+n &= 2(n \cdot k) - n + n \\ &= \frac{2}{3}n^2. \end{aligned}$$

The proposition is proved. ■

It is worth to remark that the bound $\frac{2}{3}n^2$ is never achieved, we used algorithm TRIPLES to analyze both sequences. The experimental results are summarized in the following table.

Case	hardest triple	synchronizing string	length
$n = 3(k+1)$	$\{2, 3+k, 3+2k\}$	$a^{2k}cb\left(a^{n-2}cb\right)^{2k}$	$\frac{2}{3}n^2 - \frac{4}{3}n$
$n = 3k+4$	$\{2, 3+k, 4+2k\}$	$a^{k}cb\left(a^{n-2}cb\right)^{2k+1}$	$\frac{2}{3}n^2 - \frac{4}{3}n + \frac{2}{3}$
$n = 3k+5$	$\{2, 3+k, 4+2k\}$	$\left(a^{n-2}cb\right)^{2(k+1)}$	$\frac{2}{3}n^2 - \frac{4}{3}n$

Theorem 2 *Let $\lambda > 0$, we say that an automaton $\mathcal{M}$ is λ-extendable, if and only if, for all $A \subset Q$ there exists $w \in \Sigma^*$ such that $\left|\delta^{-1}(A, w)\right| > |A|$ and $|w| \leq \lambda \cdot |Q|$.*

Kisilewicz and Szykula introduced a further sequence of automata whose synchronizing time is $\Omega\left(n^2\right)$ [16]. This third sequence is used to prove that for all $\lambda \geq 1$ there exist automata that are not λ-extendable. It is important to remark that this sequence is not extremal, given that the synchronizing time of the n-th automaton in the sequence, which has size n, is equal to $\frac{n^2+3}{2}$ [16]. The later sequence is related to a sequence that was previously introduced by Berlinkov (see reference [4]), and which was used to prove that there exist automata which are not 2-extendable. Berlinkov sequence is a doubly parameterized sequence of automata that we denote with the symbol $\{\mathcal{A}(n,k)\}_{n,k \geq 1}$. Although the non-extensibility results of Kisilewicz and Szykula are stronger, Berlinkov sequence is harder to synchronize, the synchronizing time of $\mathcal{A}(n,k)$

is equal to $(n-1)(n-1)+3$, and its size is equal to $n+k+1$. Notice that if we fix k, the subsequence $\{\mathcal{A}(n,k)\}_{n\geq 1}$ becomes an extremal sequence.

We prove that the sequence of Berlinkov automata cannot give place to a strong refutation. First we have to define the automaton $\mathcal{A}(n,k) = (Q_{nk}, \Sigma, \delta_{nk})$. The set of states of $\mathcal{A}(n,k)$ is the set $Q_{nk} = \{q_0, ..., q_n, s_1, ..., s_k\}$. The alphabet is equal to $\{a, b\}$, and the transition function δ_{nk} is defined by

$$\delta_{nk}(q,a) = \begin{cases} q_{i+1} \text{ if } q = q_i,\ 0 \leq i \leq n-1 \\ q_0 \text{ if } q = q_n \\ q_2 \text{ if } q = s_j,\ 1 \leq j \leq k \end{cases}$$

$$\delta_{nk}(q,b) = \begin{cases} s_{j+1} \text{ if } q = s_j,\ 1 \leq j \leq k-1 \\ q_0 \text{ if } q = s_k \\ s_1 \text{ if } q = q_0 \\ q_i \text{ if } q = q_i,\ 1 \leq i \leq n \end{cases}$$

The sequence of Berlinkov automata is best described by the following graphic which represents the automaton $\mathcal{A}(m,1)$

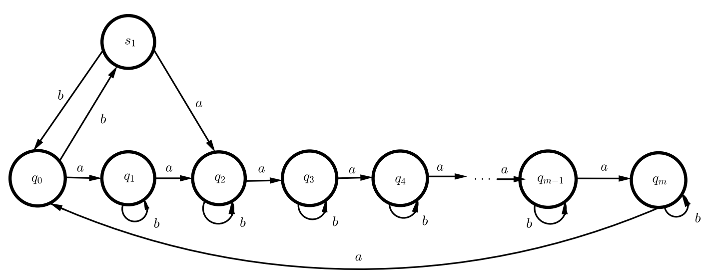

Figure 3: Berlinkov automata

Theorem 6 *The sequence of Berlinkov automata cannot give place to a strong refutation.*

Proof. Let $n, k \geq 1$. The automaton $\mathcal{A}(n,k)$ has size $n+k+1$, we prove that the synchronization time of any triple is bounded above by $\frac{2}{3}(n+k+1)^2$. We suppose that $n = 3t$, the other two cases are proved in a similar way. Suppose that $\{p,q,r\} \subseteq \{q_0, ..., q_n\}$. There exists $m \leq n-1$ such that $\widehat{\delta}(\{p,q,r\}, a^m)$ is a triple q_i, q_j and q_k fulfilling the following conditions:

1. $i = 0$.

2. $j \leq k$

3. $r \leq 2t$

In order to synchronize the triple q_i, q_j, q_r one can use the string $ba(a^{n-1}ba)^{r-1}$. Then, there exists $m \leq n$ such that $a^m ba(a^{n-1}ba)^{r-1}$ synchronizes the triple p, q, r. The length of this synchronizing string is bounded above by $(n+1)\,2t$, and we have that for all $k \geq 1$ the inequality

$$(n+1)\,2t \leq \frac{2}{3}\,(n+k+1)^2$$

holds. The proposition is proved. ■

It is worth to remark that the bound $\frac{2}{3}n^2$ is never achieved. The synchronizing behavior of Berlinkov automata is described in full detail by the following table.

Case	hardest triple	synchronizing string	length
$n = 3\,(t+1)$	$\{s_1, q_{t+1}, q_{2t+2}\}$	$ba^{2t-2}\,(ba^{n-2})^{2t+1}$	$\frac{2}{3}n^2 - n$
$n = 3t+4$	$\{s_1, q_{t+1}, q_{2k+2}\}$	$b\,(ba^{n-2})^{2t+1}$	$\frac{2}{3}n^2 - \frac{4}{3}n + \frac{5}{3}$
$n = 3t+5$	$\{s_1, q_{t+2}, q_{2t+3}\}$	$ba^{2t+2}\,(ba^{n-2})^{2t+2}$	$\frac{2}{3}n^2 - \frac{4}{3}n + 1$

It is important to remark that all the other slowly synchronizing sequences registered in the literature, like for the sequence of sink automata discovered by Martugin [18], are not extremal, and hence they cannot give place to strong refutations.

Given an extremal sequence, there are many naive ways of constructing new extremal sequences from the given one. All those naive constructions exhibit the following feature: If one receives as output (of such a construction) a strong refutation, then the input was already a strong refutation. Thus, at this point, we do not know of a further extremal sequence to be checked.

Gonze and Jungers introduced a further sequence of synchronizing automata (see [10]). We are not sure if such a sequence is an extremal one, but it seems to be a threat to our conjecture (see below). We will prove that Gonze-Jungers sequence cannot give place to a strong refutation.

A long remark: The triple rendezvous time of Gonze and Jungers

Gonze and Jungers [10] studied the synchronization times of the easiest to synchronize tuples of states.

Let $\mathcal{M}$ be a synchronizing automaton, let

$$t_k(\mathcal{M}) = \min\left(\{s(\mathcal{M}, q_1, ..., q_k) : q_1, ..., q_k \text{ are } k \text{ different states of } \mathcal{M}\}\right),$$

and let

$$T_k(n) = \max(\{t_k(\mathcal{M}) : \mathcal{M} \text{ is a } n\text{-state synchronizing automaton}\}).$$

Gonze and Jungers studied the sequence $\{T_k\}_{k\geq 2}$. Notice that for all n, the equality $T_n(n) = sw_n(n)$ holds. It means, among other things, that the study of one of those sequences could pave the way to solve Černy's problem.

Observe that $T_2(n) = 1$. Thus, we have that once again the case $k = 2$ is a solved case. The first interesting case, the case $k = 3$, is highly nontrivial. Gonze and Jungers focus on this case. They use the term *triple rendezvous time* to refer the function $T_3(n)$.

Notice that $T_3(\mathcal{C}_n) = n + 1$, where $n + 1$ is the synchronizing time of the triple $\{0, 1, 2\}$. One can conjecture, based on this fact, that $T_3(n) \leq n + 1$. It makes sense, given that Černy automata are supposed to be the hardest to synchronize finite state automata. Does this later conjecture holds true?

It was believed that all the synchronizing automata are 1-extendable, and it took a long period of time until Berlinkov found a sequence of automata that refutes the later conjecture [4]. Let $\mathcal{M}$ be a n-state synchronizing automaton and let $A \subset Q_{\mathcal{M}}$,.we define $E(\mathcal{M}, A)$ as

$$E(\mathcal{M}, A) = \min\left\{|w| : \exists q \left(\delta^{-1}(q, w) \supseteq A\right)\right\},$$

and given $k \leq n$, we set

$$E(\mathcal{M}, k) = \min\{A \subset Q;\ \ |A| = k : E(\mathcal{M}, A)\}.$$

We say that $\mathcal{M}$ is weakly extendable, if and only if, for all $k \leq n$ it happens that $E(\mathcal{M}, k) \leq n(k-2) + 1$.

Theorem 7 *If $\mathcal{M}$ is weakly extendable, then $\mathcal{M}$ can be synchronized by a short string whose length is upperbounded by $(n-1)^2$.*

Notice that the equality $E(\mathcal{M}, k) = t_k(\mathcal{M})$ holds. Thus, the *weakly extensibility conjecture* can be expressed as follows.

Theorem 4 *For all $k \leq n$, the inequality $T_k(n) \leq n(k-2) + 1$ holds.*

Notice that the above conjecture entails Černy's Conjecture, notice also that it holds true for $k = 2$. The later conjecture also implies that for $k = 3$, the inequality $T_k(n) \leq n + 1$ must hold. Gonze and Jungers [10] refuted this special case of the conjecture (and hence the conjecture). To this end, they constructed a sequence $\{\mathcal{GJ}_{2n+1}\}_{n\geq 4}$ of synchronizing automata, such that for all $n \geq 4$ the equality

$$T_3(\mathcal{GJ}_{2n+1}) = \left|Q_{\mathcal{GJ}_{2n+1}}\right| + 3 = 2n + 4$$

holds. It shows, among other things, that Černy automata are not completely representative of the hardness of synchronization. It also shows that, once again, the intricacies of synchronization begin to occur at $k = 3$.

The sequence of Gonze and Jungers, which we describe below, seems to be a hard to synchronize sequence of automata: Notice that all their triples are hard to synchronize. Then, it could provide us with the counterexamples we are looking for. We show that it is not the case, we show that it is not a strong refutation.

We will use the Symbol GJ to denote the sequence $\{\mathcal{GJ}_{2n+1}\}_{n\geq 4}$. Given $i \geq 4$, automaton $\mathcal{GJ}_{2i+1}$ is a $2i+1$-state synchronizing automaton over the alphabet $\{a, b\}$. It is constituted by two linear graphs, connected together by an hexagon, that we call the *core*, and whose nodes are labelled with the symbols $1, 2, 3, 4, 5$, and 6. The graphic in figure 2 corresponds to the automaton $\mathcal{GJ}_9$, which is the smallest automaton in the sequence.

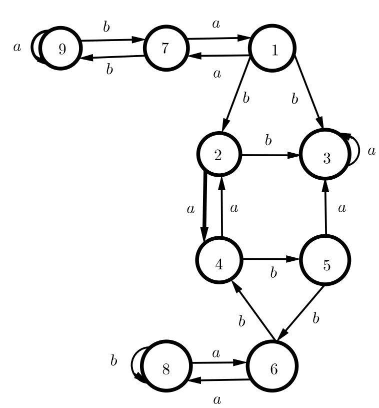

Figure 4: Automaton $\mathcal{GJ}_9$

Automaton $\mathcal{GJ}_{2n+3}$ is constructed from $\mathcal{GJ}_{2n+1}$ in the following way:

1. We add two additional nodes labelled $2n + 2$ and $2n + 3$. Node $2n + 2$ is added to the linear graph on the bottom, while node $2n + 3$ is added to the linear graph on the top.

2. We remove the loop attached to node $2n$, and we add a bidirectional edge connecting nodes $2n$ and $2n + 2$, this new edge has the same label as the removed loop. A loop with the complementary label is attached

to node $2n + 2$. We make the same work on the linear graph placed at the top.

It was not hard to identify the hardest triples of the automata occurring in the sequences of Kisielewicz-Szykula as well as in the sequence of Berlinkov, it was also easy to compute the synchronizing times of those triples. It was not the case with the sequence GJ. After some though we could not determine which are the hardest triples of those automata. Then, we decided to run algorithm TRIPLES on some of the smallest automata included in the sequence, but we could not identify an useful pattern. Therefore, we decided to try a more theoretical strategy: We proved that for all $n \geq 4$, the synchronizing time of any pair of states of the automaton $\mathcal{GJ}_{2n+1}$ is bounded above by $\frac{2n+1}{2} + 2 \cdot (2n+1) + 10$. It implies that $s_3(\mathcal{GJ}_{2n+1}) \leq 5(2n+1) + 20$. Therefore, we get as a corollary that the GJ sequence is not a strong refutation.

Theorem 1 *Let $n \geq 9$ be an odd integer, we have that $s_2(\mathcal{GJ}_n) \leq \frac{n}{2} + 2 \cdot n + 10$.*

Proof. Let $n \geq 9$ be an odd integer, and let p, q be two states of $\mathcal{GJ}_n$. We show that those states can be synchronized using a string of length $\frac{n}{2} + 2 \cdot n + 10$. The synchronization process is divided in three stages. In the first stage one of the states is sent to node 3. In the second stage the remaining state is sent to node 1, while taking care of maintaining the position of the first state (at node 3). In the third stage the states 1 and 3 are synchronized.

1. The first stage of the process can be accomplished with a string of length $\frac{n}{2}$.
2. In the second stage we make the following:

   (a) Let x be the position of the second state. We check if there exists an outgoing edge labelled with the letter b and pointing to the right. If it is the case we say that node x fulfills condition B. If it is not the case, we apply letter a. Notice that, after the application of letter a, the second state will be located at a node fulfilling condition B, while the first state will be located at node 3.

   (b) Now, the location of the second state fulfills condition B. We move this state to the core. To this end, we use a string $(bbba)^m$, where m is some integer lesser than $\frac{n}{4}$. Notice that, for all $m \geq 1$, it happens that after the application of the string $(bbba)^m$ the first state will be still located at node 3.

   (c) If the second state enters the core at node 1, we have finished. Otherwise, it must enter the core at node 6. In this second case we use the string $bbaab$ to move this state to node 1. Notice that the first state will maintain its position at node 3. The total cost of this second stage is bounded above by $n + 6$.

3. The synchronization of the pair $\{1,3\}$ can be accomplished using the string $babbbbb\,(ab)^{\frac{n-7}{2}}\,abba$. The cost of this stage is bounded above by $n+4$.

■

Theorem 1 *Let n be an odd integer, the inequality* $s_3\left(\mathcal{GJ}_n\right) \leq 5\cdot n+20$ *holds. Moreover, for all* $n \geq 1$ *the inequality* $s_3\left(\mathcal{GJ}_n\right) \leq \frac{2}{3}\cdot n^2$ *holds.*

5 Concluding remarks and open problems

We would like to finish this paper formulating some few questions for future work. The Status of Černy's conjecture and the related $\frac{2}{3}$-Černy's conjecture are the major open questions, however we would like to discuss a minor one.

The *weak Černy's conjecture* states that the synchronizing time of a n-state synchronizing automaton is $O\left(n^2\right)$. Proving this weaker conjecture would be a major step towards the solution of Černy's problem, as well as a major step towards a complete understanding of synchronizing automata. We studied, in this work, the quadratic sequences that have been registered in the literature, and it is interesting to observe that all those sequences are constituted by planar automata (automata whose transition digraph is planar). It suggests that the largest reset tresholds are achieved by planar automata. Take into account that planarity is a constraint on the connectivity of (di)graphs. We conjecture that the existence of superquadratic sequences of automata implies the existence of superquadratic sequences of planar automata. It means that if The weak Černy's conjecture holds true for planar automata, then it must hold true for all the synchronizing automata. Our conjecture estates that planar automata are universal with respect to The weak Černy's conjecture, in the same sense that strongly connected automata are universal with respect to Černy's conjecture.

References

[1] D. Ananichev, V. Gusev, M. Volkov, Slowly Synchronizing Automata and Digraphs, *Mathematical Foundations of Computer Science 2010*, Springer-Verlag Berlin Heidelberg, 2010, 55-65. https://doi.org/10.1007/978-3-642-15155-2_7

[2] R. Baeza-Yates, Searching Subsequences, *Theo. Comput. Sci.*, **78** (1991), no. 2, 363-376. https://doi.org/10.1016/0304-3975(91)90358-9

[3] M.V. Berlinkov, On Two Algorithmic Problems about Synchronizing Automata, *Developments in Language Theory*, Springer International Publishing Switzerland, 2014, 61-67.
https://doi.org/10.1007/978-3-319-09698-8_6

[4] M.V. Berlinkov, On a Conjecture by Carpi and D'Alessandro, *Int. J. Found. Comput. Sci.*, **22** (2011), no. 7, 1565-1576.
https://doi.org/10.1142/s012905411100888x

[5] J. Černý, Poznámka k. homogénnym experimentom s konecnymi automatmi, *Mat.-Fyz. Casopis*, **14** (1964), 208–216.

[6] E.W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik*, **1** (1959), 269–271.
https://doi.org/10.1007/bf01386390

[7] D. Eppstein, Reset sequences for monotonic automata, *SIAM J. Comput.*, **19** (1990), 500–510. https://doi.org/10.1137/0219033

[8] J. Flum, M. Grohe, *Parameterized Complexity Theory*, Springer Verlag, Berlin, 2006. https://doi.org/10.1007/3-540-29953-x

[9] P. Gawrychowski, D. Straszak, Strong Inapproximability of the Shortest Reset Word, *Mathematical Foundations of Computer Science 2015*, Springer-Verlag Berlin Heidelberg, 2015, 243-255.
https://doi.org/10.1007/978-3-662-48057-1_19

[10] F. Gonze, R.M. Jungers, On the Synchronizing Probability Function and the Triple Rendezvous Time for Synchronzing Automata, *SIAM Journal on Discrete Mathematics*, **30** (2016), 995-1014.
https://doi.org/10.1137/15m1024603

[11] H. Jurgensen, Synchronization, *Inf. Comput.*, **206** (2008), no. 9-10, 1033-1044. https://doi.org/10.1016/j.ic.2008.03.005

[12] S. Guillemot, Parameterized complexity and approximability of the Longest Compatible Sequence problem, *Discrete Optimization*, **8** (2011), no. 1, 50-60. https://doi.org/10.1016/j.disopt.2010.08.003

[13] P.M. Higgins, The range order of a product of i transformations from a finite full transformation semigroup, *Semigroup Forum*, **37** (1988), 31-36.
https://doi.org/10.1007/bf02573120

[14] A.E. Kirnasov, On glueing states of an automaton, *Diskr. Mat.*, **15** (2003), no. 4, 66–83. (In Russian). https://doi.org/10.4213/dm216

[15] A. Kisielewicz, M. Szykula, Generating Small Automata and the Černý Conjecture, *Implementation and Application of Automata*, Springer-Verlag Berlin Heidelberg, 2013, 340-348.
https://doi.org/10.1007/978-3-642-39274-0_30

[16] A. Kisielewicz, M. Szykula, Synchronizing Automata with Extremal Properties, *Mathematical Foundations of Computer Science 2015*, Springer-Verlag Berlin Heidelberg, 2015, 331-343.
https://doi.org/10.1007/978-3-662-48057-1_26

[17] D. Lee, M. Yannakakis, Testing Finite State Machines, *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing-STOC '91*, (1991), 476-485. https://doi.org/10.1145/103418.103468

[18] P.V. Martugin, A series of slowly synchronizing automata with a zero state over a small alphabet, *Inf. Comput.*, **206** (2008), no. 9-10, 1197-1203. https://doi.org/10.1016/j.ic.2008.03.020

[19] B. Natarajan, An algorithmic approach to the automated design of parts orienters, *Proceedings IEEE Annual Symp. Foundations of Comp. Sci.*, (1986), 133-142. https://doi.org/10.1109/sfcs.1986.5

[20] A. Pereira, *The Černy Conjecture and other Synchronization Problems*, PhD. Thesis, Universidade do Porto, 2014.

[21] A. Trakhtman, An eficient algorithm finds noticeable trends and examples concerning the Cerny conjecture, *Mathematical Foundations of Computer Science 2006*, Springer-Verlag Berlin Heidelberg, 2006, 789-800.
https://doi.org/10.1007/11821069_68

[22] M. Volkov, Synchronizing Automata and the Černý Conjecture, *Language and Automata Theory and Applications 2008*, Lecture Notes in Computer Science, Vol. 5196, Springer-Verlag Berlin Heidelberg, 2008, 11–27.
https://doi.org/10.1007/978-3-540-88282-4_4

[23] V. Vorel, Subset Synchronization of Transitive Automata, AFL, (2014), 370-381.

Received: August 2, 2017; Published: August 29, 2017