

Extremal Binary PFAs with Small Number of States ^{*}

Stijn Cambie, Michiel de Bondt, and Henk Don

Department of Mathematics, Radboud University Nijmegen, Postbus 9010, 6500 GL Nijmegen, The Netherlands
 stijncambie@hotmail.com, {m.debondt, h.don}@math.ru.nl

Abstract. The largest known reset thresholds for DFAs are equal to $(n - 1)^2$, where n is the number of states. This is conjectured to be the maximum possible. PFAs (with partial transition function) can have exponentially large reset thresholds. This is still true if we restrict to binary PFAs. However, asymptotics do not give conclusions for fixed n . We prove that the maximal reset threshold for binary PFAs is strictly greater than $(n - 1)^2$ if and only if $n \geq 6$.

These results are mostly based on the analysis of synchronizing word lengths for a certain family of binary PFAs. This family has the following properties: it contains the well-known Černý automata; for $n \leq 10$ it contains a binary PFA with maximal possible reset threshold; for all $n \geq 6$ it contains a PFA with reset threshold larger than the maximum known for DFAs.

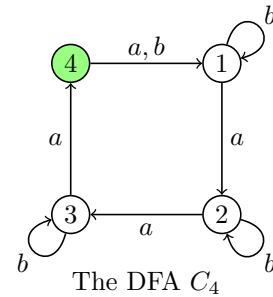
Analysis of this family reveals remarkable patterns involving the Fibonacci numbers and related sequences such as the Padovan sequence. We derive explicit formulas for the reset thresholds in terms of these recurrent sequences.

Asymptotically the Černý family gives reset thresholds of polynomial order. We prove that PFAs in the family are not extremal for $n \geq 41$. For that purpose, we present an improvement of Martyugin's prime number construction of binary PFAs.

Keywords: Finite automata · Synchronization · Černý conjecture.

1 Introduction and Preliminaries

The diagram on the right depicts the *deterministic finite automaton* (DFA) C_4 . Starting in any state q and reading the word ba^3ba^3b leads to state 1. Therefore, w is called a synchronizing word for C_4 . It is also the only synchronizing word for C_4 of length at most 9.



^{*} The first author has been supported by a Vidi Grant of the Netherlands Organization for Scientific Research (NWO), grant number 639.032.614.

Formally, a DFA A is defined as a triple (Q, Σ, δ) . Here Σ is a finite alphabet, Q a finite set of states, which we generally choose to be $[n] = \{1, 2, \dots, n\}$, and $\delta : Q \times \Sigma \rightarrow Q$ the transition function. For $w \in \Sigma^*$ and $q \in Q$, we define qw inductively by $q\varepsilon = q$ and $qwa = \delta(qw, a)$ for $a \in \Sigma$, where ε is the empty word. So qw is the state where one ends, when starting in q and reading the symbols in w consecutively, and qa is a shorthand notation for $\delta(q, a)$. We extend the transition function to sets $S \subseteq Q$ by $Sw := \{qw : q \in S\}$. A word $w \in \Sigma^*$ is called *synchronizing*, if a state $q_s \in Q$ exists such that $qw = q_s$ for all $q \in Q$. The length of a shortest word with this property is the *reset threshold* of A .

A central conjecture in the field (standing since 1964) is due to Černý [4]:

Conjecture 1.1. Every synchronizing DFA on n states admits a synchronizing word of length $\leq (n-1)^2$.

We denote the maximal possible reset threshold for a DFA on n states by $d(n)$, rephrasing the conjecture to $d(n) = (n-1)^2$. The best known upper bounds are still cubic in n . In 1983 Pin [7] established the bound $\frac{1}{6}(n^3 - n)$, using a combinatorial result by Frankl [5]. More than thirty years later, the leading constant was improved to 0.1664 by Szykuła, and subsequently to 0.1654 by Shitov [9]. For a survey on synchronizing automata and the Černý's conjecture, we refer to [10].

If Conjecture 1.1 holds true, the bound is sharp. The DFA C_4 is one in a sequence found by Černý [4]. For $n \geq 2$, the DFA C_n has n states which we denote by $Q = [n]$, a symbol a sending q to $q+1 \pmod n$ and a symbol b sending n to 1 and being the identity in all other states. The shortest synchronizing word for C_n is $b(a^{n-1}b)^{n-2}$ of length $(n-1)^2$, so that $d(n) \geq (n-1)^2$.

The picture changes drastically if we consider *partial finite automata* (PFAs). In a PFA, the transition function is allowed to be partial. This means that qa may be undefined for $q \in Q$ and $a \in \Sigma$. If $q \in S \subseteq Q$ and qw is undefined, then Sw is undefined as well. In this setting a word w is called synchronizing for a PFA if there exists a $q_s \in Q$ such that qw is defined and $qw = q_s$ for all $q \in Q$. Our notion of synchronization for PFAs is equivalent to D1- and D3-direction, and to careful synchronization as in §6.2 of [11], but not to D2-direction and exact synchronization [8]. The last two notions allow qw to be undefined.

For PFAs the maximal reset thresholds grow asymptotically like an exponential function of n , in contrast with the polynomial growth for DFAs. Also the behaviour in terms of alphabet size is different. The upper bound of Conjecture 1.1 is attained by binary DFAs. For PFAs there is evidence that the alphabet size has to grow with n to attain the maximal reset thresholds [2]. Still, also binary PFAs give exponentially growing reset thresholds. We denote the maximal values by $p(n, 2)$. A binary PFA attaining the maximal reset threshold is called *extremal*. For $2 \leq n \leq 10$ the values as found in [2] are given below. For $n \geq 11$, the maximum is unknown.

n	2	3	4	5	6	7	8	9	10
$p(n, 2)$	1	4	9	16	26	39	55	73	94

For all $2 \leq n \leq 10$, these reset thresholds are attained by members of what we will call the *Černý family*. This family of PFAs C_n^c will be introduced in Section 2.

In Section 3 we relate the problem of finding reset thresholds for this family to a minimization problem involving racing pawns. A recursive solution for this problem is presented in Section 4, from which it follows that the maximal reset thresholds in the family grow like $n^2 \log(n)$. In Section 5, we give an exact solution of the minimization problem in terms of recurrent sequences. In addition, we determine the number of different optimal races. In Section 6 we estimate the solution more precisely and find the asymptotic size of a shortest synchronizing word for C_n^c for fixed c . Furthermore, we estimate the optimal choice of c asymptotically in terms of n . In Section 6A, we discuss odd behavior in the optimal choice of c which emerged from computations.

We end with the presentation of another construction of binary PFAs in Section 7, to defeat the Černý family for large n . We show in Section 8 that this construction gives binary PFAs with larger reset thresholds than C_n^c for $n \geq 41$. It is unknown to us if there are constructions that beat the Černý family for some $n < 41$. Our construction is an improvement of Martyugin's prime number construction of binary PFAs [6], which has reset threshold $\exp((1 + o(1))\sqrt{n \ln(n)/2})$. We show in Section 7 that the asymptotic behavior of the reset threshold of our construction is $\exp((1 + o(1))\sqrt{n \ln(n)})$, which is comparable to that of Martyugin's prime number construction of ternary PFAs [6]. We do this by providing sufficiently accurate estimates of the reset threshold for all three prime number constructions. To our knowledge, estimates with this level of accuracy have not been given before.

The current paper extends the earlier work in [3]. The proof of Theorem 3.2 has been formalized and the content has been extended by the results in sections 5, 6, 6A,7 and 8.

2 Extending the Černý Sequence to a Family

The Černý family of binary PFAs, denoted by C_n^c , contains the Černý sequence $C_n = C_n^0$ of binary DFAs. For fixed $c \in \mathbb{N}$ and $n \geq c + 2$, we define the PFA C_n^c with n states and alphabet $\Sigma = \{a, b\}$ by

$$qa = \begin{cases} q+1 & 1 \leq q \leq n-c-1 \\ \perp & n-c \leq q \leq n-1 \\ 1 & q = n \end{cases} \quad qb = \begin{cases} q & 1 \leq q \leq n-c-1 \\ q+1 & n-c \leq q \leq n-1 \\ 1 & q = n \end{cases}$$

The PFA C_n^c is depicted in Figure 1 for $n = 8$ and $c = 2$, next to the DFA C_n^0 of Černý. By analyzing this family, we obtain our main results. In particular, we will conclude that $p(n, 2) > (n-1)^2$ if and only if $n \geq 6$.

Before deriving general formulas for the reset thresholds, we present the values for $2 \leq n \leq 15$ and $0 \leq c \leq 4$ in the following table. Independent of the analysis that will follow, these values were found by an algorithm computing the reset threshold for a given PFA.

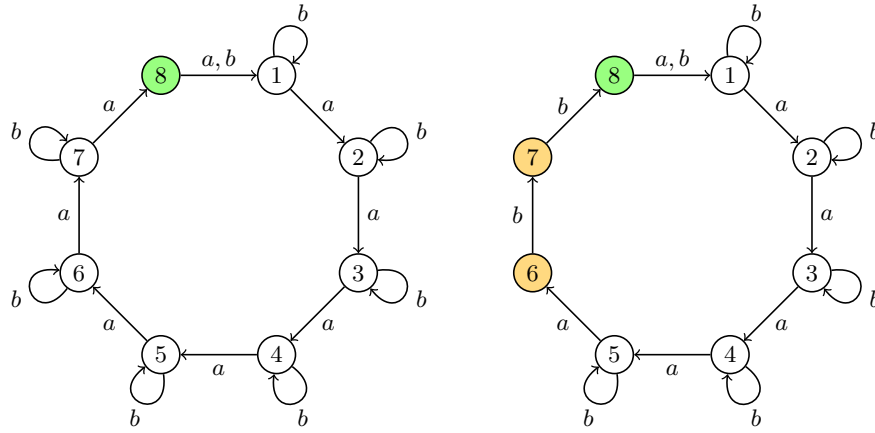


Fig. 1. The DFA C_8^0 and the PFA C_8^2

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$c = 0$	1	4	9	16	25	36	49	64	81	100	121	144	169	196
$c = 1$		2	7	15	26	39	55	73	93	116	141	168	197	228
$c = 2$			3	10	21	35	52	72	94	119	146	176	208	242
$c = 3$				4	13	27	44	65	89	115	144	176	211	248
$c = 4$					5	16	33	53	78	106	136	169	206	246

Values in boldface represent the maximal reset threshold in the family for a given n . For $n = 13$, the maximum is attained twice, see also Figure 2. Later we will see that for large n , the optimal c is close to $n/2$. For $2 \leq n \leq 10$, these maxima exactly match the values of $p(n, 2)$. This means that the Černý family contains a binary PFA on n states with maximal possible reset threshold for all $2 \leq n \leq 10$. In fact, for $6 \leq n \leq 10$, there exists only one binary PFA reaching this maximum [2].

The first line of the table shows the squares $(n - 1)^2$ for the Černý sequence C_n^0 . To give explicit expressions for subsequent lines is much harder. The order of growth is still quadratic for every c , but no formula of the form $a_2 n^2 + a_1 n + a_0$ exists in general, as we will see later in this paper.

We now turn to the analytic derivation of reset thresholds for the Černý family. We use the following interpretation of synchronization: let a pawn be placed in every state of a PFA, let them simultaneously follow the same word w and let two of them merge if they are in the same state after reading some prefix of w . A synchronizing word is then a word that merges all pawns.

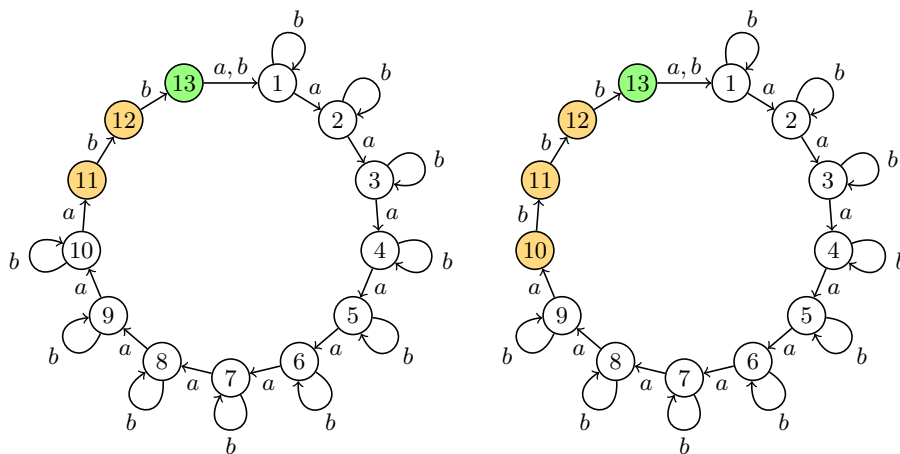


Fig. 2. The PFAs C_{13}^2 and C_{13}^3 both synchronize in 176 steps.

3 Reduction to a Pawn Race Problem

Our first result reduces the question of synchronizing C_n^c to the following problem.

Problem 3.1 (Pawn race problem). We have n pawns on the integers $1, 2, \dots, n$. In every iteration, every pawn has the choice to move from its location k to $k + 1$ or to stay at k . Moving costs $c + 1$, staying costs c . After every iteration, if two pawns are in the same position, they merge. What is the minimum cost for which it is possible to merge all the pawns?

Theorem 3.2. *Let $f_c(n)$ be the solution to Problem 3.1 and denote $n' = n - c - 1$. The reset threshold of C_n^c is equal to*

$$n'(n' - 1) + c + 1 + f_c(n').$$

The rest of this section will be devoted to the proof of Theorem 3.2.

Lemma 3.3. $C_n^c = (Q, \Sigma, \delta_n)$ has a synchronizing word.

Proof. We denote $[k] := \{1, 2, \dots, k\}$ and note that $Qb^{c+1} = [n - c - 1] \subset [n - c]$. Define $\tilde{a} = b^c a$ and $\tilde{b} = b^{c+1}$. Then \tilde{a} acts as a cyclic permutation on $[n - c]$ and \tilde{b} sends $n - c$ to 1 and is the identity otherwise. Here we recognize the Černý automaton C_{n-c}^0 , so that C_n^c is synchronizing. \square

Inspired by the proof of Lemma 3.3, we define the PFA $C_n^* = ([n], \Gamma, \eta_n)$ with state set $[n]$ and alphabet $\Gamma = \{a, \tilde{a}, \tilde{b}\}$. The transition function is defined by

$$\begin{cases} qa = q\tilde{a} = q + 1 \text{ and } q\tilde{b} = q & \text{if } q \neq n, \\ na = \perp, n\tilde{a} = n\tilde{b} = 1. \end{cases} \quad (1)$$

See Figure 3 for an illustration. Observe that restricting the transition function δ_n of C_n^c to $[n-c]$ relates to the PFA C_{n-c}^* in the following way:

$$\delta_n(q, a) = \eta_{n-c}(q, a), \quad \delta_n(q, b^c a) = \eta_{n-c}(q, \tilde{a}), \quad \delta_n(q, b^{c+1}) = \eta_{n-c}(q, \tilde{b}) \quad (2)$$

for all $q \in [n-c]$. By substituting $\tilde{a} = b^c a$ and $\tilde{b} = b^{c+1}$, a word $w \in \Gamma^*$ naturally corresponds to a word $s_c(w) \in \Sigma^*$ with the property $\delta_n(q, s_c(w)) = \eta_{n-c}(q, w)$ for all $q \in [n-c]$. We define the c -weighted length of a word $w \in \Gamma^*$ as the length of $s_c(w)$, which we denote by $|w|_c$.

In Corollary 3.5 below, we prove that a synchronizing word of minimal c -weighted length for C_{n-c}^* corresponds to a shortest synchronizing word for C_n^c . For instance, consider the PFA C_6^* as in Figure 3 and take $c = 0$. Then $\tilde{a} = a$ and $\tilde{b} = b$ have weight 1 and the resulting PFA is equivalent to the Černý automaton C_6^0 . If we instead take $c = 2$, then $\tilde{a} = b^2 a$ and $\tilde{b} = b^3$ both have weight 3 and a word of minimal 2-weighted length for C_6^* corresponds to a shortest synchronizing word for the PFA C_8^2 given in Figure 1.

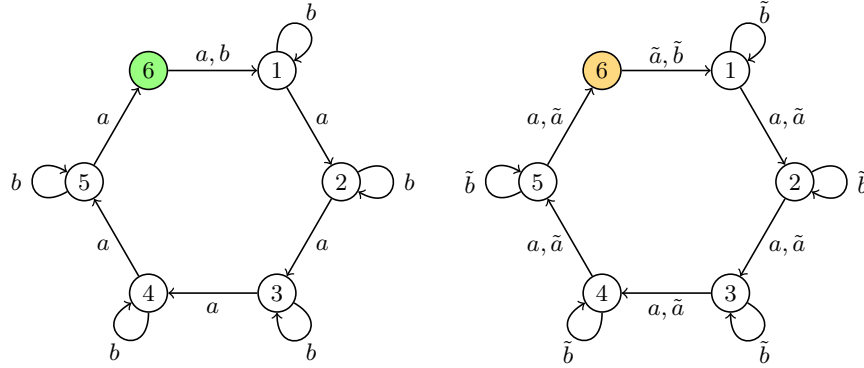


Fig. 3. The DFA C_6^0 and the PFA C_6^* . If \tilde{a} and \tilde{b} have weight 3, and a has weight 1, then a synchronizing word of minimum weighted length for C_6^* corresponds to a shortest synchronizing word for C_8^2 .

Let $S \subseteq Q$ and $w \in \Sigma^*$. We say that w has *minimum length* for $Sw = T$ if $Sw = T$, and if $Sv = T$ implies $|v| \geq |w|$ for all $v \in \Sigma^*$.

Lemma 3.4. *Consider C_n^c . Let $S \subseteq Q = [n]$ and $T \subsetneq [n-c]$. Take $w \in \Sigma^*$ such that $w \neq \varepsilon$, and suppose that w has minimum length for $Sw = T$.*

- (i) *If $S = Q$, then w starts with b^{c+1} .*
- (ii) *If $S \subseteq [n-c]$ and $n-c \in S$, then w starts with $b^c a$ or b^{c+1} .*
- (iii) *If $S \subseteq [n-c]$ and $n-c \notin S$, then w starts with a .*

Proof. Since $qb^{c+2} = qb^{c+1}$ for all states $q \in Q$, the word w cannot contain b^{c+2} . Furthermore, $(n-c)b^m a$ is not defined for $m = 0, 1, \dots, c-1$. Using the fact that $(n-c)b^m \notin [n-c]$ for $m = 1, 2, \dots, c$, statement (ii) follows.

It also follows that w starts with b^c if $S = Q$. Suppose that $S = Q$ and w starts with $b^c a$. Since $Qb^c a = [n - c]$, we infer from (ii) that w starts with $b^c a b^c$. But this contradicts the assumption that w has minimum length, because $Qb^c = [n - c - 1] \cup \{n\} = Qb^c a b^c$. This yields (i).

Statement (iii) follows by observing that $Sb = S$ in this case. □

This lemma gives a corollary to relate words in $C_n = ([n], \{a, b\}, \delta_n)$ to words in $C_{n-c}^* = ([n - c], \{a, \tilde{a}, \tilde{b}\}, \eta_{n-c})$. Essentially, for $S \subseteq [n - c]$ there is a one-to-one correspondence between a word w of minimum length for $\delta_n(S, w) = T$ and a word w' of minimum c -weighted length for $\eta_{n-c}(S, w') = T$.

Corollary 3.5. *Let $c \in \mathbb{N}$, and suppose that $S = [n]$ or $S \subseteq [n - c]$. Let $T \subsetneq [n - c]$. Suppose that $w \in \Sigma^*$ has minimum length for $\delta_n(S, w) = T$. Then there exists $w' \in \Gamma^*$ with $s_c(w') = w$ and minimum c -weighted length for $\eta_{n-c}([n - c] \cap S, w') = T$.*

Proof. If $w = \varepsilon$, then $S = T \subsetneq [n - c]$. Therefore, $[n - c] \cap S = T$ and $w' = \varepsilon$ suffices. So assume that $w \neq \varepsilon$. From Lemma 3.4, it follows that w has a prefix $u \in \{a, b^c a, b^{c+1}\}$. Let $u' \in \Gamma$ satisfy $s_c(u') = u$. By (2), we have $\delta_n(S, u) = \eta_{n-c}([n - c] \cap S, u') \subseteq [n - c]$ if $S \subseteq [n - c]$. The same is true if $S = Q$, since then $u = b^{c+1}$. By induction, we find w' such that $s_c(w') = w$ and $\eta_{n-c}([n - c] \cap S, w') = T$.

If $v' \in \Gamma^*$ were a word with $\eta_{n-c}([n - c] \cap S, v') = T$ and $|v'|_c < |w'|_c$, then we would have $\delta_n(S, s_c(v')) = T$ and $|s_c(v')| < |w|$, contradicting the minimality of $|w|$. So $|w'|_c$ is minimal.

We will now consider C_n^* and investigate words which are applied on a subset of the state set $[n]$. We see a subset of the state set $[n]$ as a collection of pawns on those states. Symbols a and \tilde{a} move these pawns clockwise without merging, but if both n and 1 are occupied by a pawn, then symbol \tilde{b} merges both pawns. This is the only possibility for pawns to merge. We call a pawn a *chaser* if its next merge will be with a pawn in front of it, and a *resigner* otherwise. So a chaser is on state n directly before merging and a resigner on state 1 .

Notice that pawns do not need to be a chaser or a resigner, because they may not take part in a merge. But if the word at hand is a synchronizing word, then all pawns end in the same state, which is state 1 if the word has minimum c -weighted length.

Suppose for now that the word at hand is synchronizing. Since all pawns end in the same state, there is a unique pawn travelling the largest distance. This pawn is always a chaser. We therefore call it the *lanterne rouge*. Similarly, there is a pawn that makes the least number of moves and is always a resigner. This one is called the *yellow jersey*. If the *lanterne rouge* or the *yellow jersey* merges, then we see the pawn which results from the merge as its continuation. Therefore, the *lanterne rouge* and the *yellow jersey* are the chaser and resigner respectively of the last merge.

Lemma 3.6. *Let $c \in \mathbb{N}$ and $S, T \subseteq [n]$, such that $n \in S$. Suppose that $w \in \Gamma^*$ has minimum c -weighted length for $Sw = T$.*

- (i) If the pawn at n is a resigner, then w starts with \tilde{a} .
- (ii) If the pawn at n is a chaser and $c \neq 0$, then w starts with \tilde{b} .
- (iii) If the pawn at n is a chaser and $c = 0$, then w can be chosen to start with \tilde{b} .

The intuition behind Lemma 3.6 is that it is optimal for both chasers and resigners to merge as quickly as possible. Loosely speaking, a chaser in state n can get one step closer to its target by choosing \tilde{b} , while choosing \tilde{a} would mean that the other pawns move as well so that the chaser makes no progress. Choosing \tilde{b} therefore minimizes the time to merge and the c -weighted word length.

On the other hand, one can say that a resigner in state n gets one step farther from its chaser by choosing \tilde{b} , while choosing \tilde{a} would mean that the other pawns move as well so that the resigner does not move relatively to the other pawns. Choosing \tilde{a} therefore minimizes the time to merge and the c -weighted word length.

We start with an informal setup of the proof of Lemma 3.6, which we elaborate in Lemma's 3.7 and 3.8 below. If w starts with \tilde{b} , then by Lemma 3.4(iii), either $w = \tilde{b}$ or w starts with $\tilde{b}a$. The effect of \tilde{a} and $\tilde{b}a$ is similar except for state n : $i\tilde{a} = i + 1 = i\tilde{b}a$ if $i \neq n$, and $n\tilde{a} = 1 \neq 2 = n\tilde{b}a$. So $\tilde{b}a$ places the pawn at n on the successor of the state where it would be placed with \tilde{a} , costing $(c + 2) - (c + 1) = 1$ extra c -weighted word length. On the other hand, \tilde{a} places the pawn at n on the predecessor of the state where it would be placed with $\tilde{b}a$, saving $(c + 2) - (c + 1) = 1$ word length.

The idea is that the relative displacement which is initiated by $\tilde{b}a$ instead of \tilde{a} , or vice versa, can be preserved by adapting the word without adapting its c -weighted length, namely by adapting the order of symbols to match the new positioning of the displaced pawn.

For a chaser, at the last moment when it is on state n before merging, \tilde{b} is applied. With a displacement to its successor, this \tilde{b} can be skipped, so $c + 1$ c -weighted word length can be saved. With the cost of the displacement, this adds up to saving c c -weighted word length. So the chaser must choose \tilde{b} for optimality if $c \neq 0$, and can choose \tilde{b} without harm if $c = 0$.

For a resigner, at the last moment when the pawn is on state n before merging, \tilde{a} is applied. With a displacement to its predecessor, this \tilde{a} can be replaced by a , so c c -weighted word length can be saved. With the saving of the displacement, this adds up to saving $c + 1$ c -weighted word length. So the resigner must choose \tilde{a} for optimality.

Lemma 3.7 below shows that the required c -weighted word length drops by at least c if a resigner is displaced to its predecessor or if it has disappeared by merging. The lemma also shows how to adapt the word in order to preserve displacement.

Lemma 3.7. *Let $S \subseteq [n]$, and assume that $w \in \Gamma^*$ is defined on S . Suppose that $i \in S$ contains a resigner for w . If $i \neq 1$, and*

$$S' = S \setminus \{i\} \quad \text{or} \quad S' = (S \setminus \{i\}) \cup \{i - 1\},$$

then there exists a word $w' \in \Gamma^$ such that $S'w' = Sw$ and $|w'|_c \leq |w|_c - c$.*

Proof. Assume the lemma holds for $|w| < k$. Take $|w| = k$ and let $w_j \in \Gamma$ be the j^{th} symbol of w for all j . Since i contains a resigner for w , implicit assumptions on the length of w in the arguments below are justified. We distinguish 2 cases:

- *Case 1: $i \neq 1$ and $S' = S \setminus \{i\}$.*
 Suppose first that $i \neq n$. Then $iw_1 \neq 1$ and for all $w_1 \in \Gamma$,

$$S'w_1 = Sw_1 \setminus \{iw_1\}.$$

Taking $w'_1 = w_1$, the result follows by induction on $|w|$.
 Suppose next that $i = n$. Then either $w_1 = \tilde{a}$ or $w_1 = \tilde{b}$. If $w_1 = \tilde{a}$ then make w' from w by replacing w_1 by a . If $w_1 = \tilde{b}$, then make w' from w by removing w_1 . In both cases, $|w'|_c \leq |w|_c - c$ and

$$S'w' = S'w \quad \text{and} \quad S'w \cup \{iw\} = Sw.$$

Since the pawn at i is merged by w (it is a resigner), it follows that $S'w' = S'w = Sw$, which gives the result.

- *Case 2: $i \neq 1$, $S' = (S \setminus \{i\}) \cup \{i-1\}$, and $S' \neq S \setminus \{i\}$.*
 Then $i-1 \notin S$. Suppose first that $i \neq n$. Then $iw_1 \neq 1$ and for all $w_1 \in \Gamma$, $(i-1)w_1 = iw_1 - 1$ and

$$S'w_1 = (Sw_1 \setminus \{iw_1\}) \cup \{iw_1 - 1\}.$$

Taking $w'_1 = w_1$, the result follows by induction on $|w|$.
 Suppose next that $i = n$. Then either $w_1 = \tilde{a}$ or $w_1 = \tilde{b}$. From $n-1 = i-1 \notin S$, we infer that $n \notin Sw_1$ and $Sw_1\tilde{b} = Sw_1$. So we may assume that $w_2 \neq \tilde{b}$. If $w_1 = \tilde{a}$, then $iw_1w_2 = 2$ and

$$S'w_2w_1 = (Sw_1w_2 \setminus \{2\}) \cup \{1\}$$

If $w_1 = \tilde{b}$, then $1 \notin S$ because the pawn at i is a resigner, so $2 \notin S'w_2w_1$. Therefore, we obtain the same assertions as in the case $w_1 = \tilde{a}$. Taking $w'_1w'_2 = w_2w_1$, the result follows by induction on $|w|$. \square

If state 1 contains a resigner, then removing it will not decrease the required c -weighted word length if the resigner is about to merge with its chaser, to advance as a chaser. So the condition that $i \neq 1$ in Lemma 3.7 is necessary.

Lemma 3.8 shows that the required c -weighted word length drops by at least $c+1$ if a chaser is displaced to its successor or if it has disappeared by merging. The lemma also shows how to adapt the word in order to preserve displacement.

Lemma 3.8. *Let $S \subseteq [n]$, and assume that $w \in \Gamma^*$ is defined on S . Suppose that $i \in S$ contains a chaser of w . If*

$$S' = S \setminus \{i\} \quad \text{or} \quad i \neq n \text{ and } S' = (S \setminus \{i\}) \cup \{i+1\},$$

then there exists a word $w' \in \Gamma^$ such that $S'w' = Sw$ and $|w'|_c = |w|_c - c - 1$.*

Proof. Assume the lemma holds for $|w| < k$. Take $|w| = k$, and let $w_j \in \Gamma$ be the j^{th} symbol of w for all j . Since i contains a chaser for w , implicit assumptions on the length of w in the arguments below are justified. We distinguish 2 cases:

– *Case 1:* $S' = S \setminus \{i\}$.

Suppose first that $i \neq n$. If $i = 1$ and $w_1 = \tilde{b}$, then $n \notin S$, because the pawn at i is a chaser. In all cases, for all $w_1 \in \Gamma$,

$$S'w_1 = Sw_1 \setminus \{iw_1\}.$$

Taking $w'_1 = w_1$, the result follows by induction on $|w|$.

Suppose next that $i = n$. Then either $w_1 = \tilde{a}$, or $w_1 = \tilde{b}$. If $w_1 = \tilde{a}$, then $iw_1 = 1$ and

$$S'w_1 = Sw_1 \setminus \{1\},$$

and the result follows by induction on $|w|$ by taking $w'_1 = w_1$. So assume that $w_1 = \tilde{b}$. Make w' from w by removing w_1 . Then $|w'|_c = |w|_c - c - 1$ and

$$S'w' = S'w \quad \text{and} \quad S'w \cup \{iw\} = Sw.$$

Since the pawn at i is merged by w (it is a chaser), it follows that $S'w' = S'w = Sw$, which gives the result. Note that $w' = \varepsilon$ is possible if $1 \in S$.

– *Case 2:* $i \neq n$, $S' = (S \setminus \{i\}) \cup \{i+1\}$, and $S' \neq S \setminus \{i\}$.

Then $i+1 \notin S$. Suppose first that $i \neq n-1$. If $i = 1$ and $w_1 = \tilde{b}$, then $n \notin S$, because the pawn at i is a chaser. In all cases, for all $w_1 \in \Gamma$, $(i+1)w_1 = iw_1 + 1$ and

$$S'w_1 = (Sw_1 \setminus \{iw_1\}) \cup \{iw_1 + 1\}.$$

Let $w'_1 = w_1$ and note that $iw_1 \neq n$. The result follows by induction on $|w|$. Suppose next that $i = n-1$. As $n = i+1 \notin S$, $S\tilde{b} = S$. So we may assume that $w_1 \neq \tilde{b}$. Furthermore, either $w_2 = \tilde{a}$ or $w_2 = \tilde{b}$. In all cases, $iw_1w_2 = 1$ and

$$S'w_2w_1 = (Sw_1w_2 \setminus \{1\}) \cup \{2\}.$$

Taking $w'_1w'_2 = w_2w_1$, the result follows by induction on $|w|$. \square

We are now ready to give a formal proof of Lemma 3.6.

Proof (of Lemma 3.6). Let $w_j \in \Gamma$ be the j^{th} symbol of w for all j . Suppose first that the pawn in state n is a resigner, and that $w_1 = \tilde{b}$. Then $1 \notin S$ and $w_1w_2 = \tilde{b}a$. Let $S' = S\tilde{a} = (S\tilde{b}a \setminus \{2\}) \cup \{1\}$. From Lemma 3.7 with $i = 2$, it follows that $S'w' = T$ for a word $w' \in \Gamma^*$ of c -weighted length at most $|w|_c - |w_1w_2|_c - c = |w|_c - 2c - 2$. So $Sv = T$ for the word $v = \tilde{a}w'$ of c -weighted length at most $|w|_c - c - 1$. Contradiction.

Suppose next that the pawn in state n is a chaser, and that $w_1 = \tilde{a}$. Suppose additionally that $w_1 = \tilde{a}$ is inevitable if $c = 0$. Let $S' = S\tilde{b}a = (S\tilde{a} \setminus \{1\}) \cup \{2\}$. From Lemma 3.8 with $i = 1$, it follows that $S'w' = T$ for a word $w' \in \Gamma^*$ of c -weighted length at most $|w|_c - |w_1|_c - c - 1 = |w|_c - 2c - 2$. So $Sv = T$ for the word $v = \tilde{b}a'$ of c -weighted length at most $|w|_c - c$, which starts with \tilde{b} . Contradiction. \square

After this excursion to the auxiliary automaton C_n^* , we return to the automaton C_n^c . Let a pawn start in each of these states. Also in this automaton pawns only move clockwise by steps of size 1. We define chasers, resigners, the lantern rouge and the yellow jersey, analogous to the definitions for C_n^* . A direct implication of Corollary 3.5 and Lemma 3.6 is the following.

Corollary 3.9. *Let $c \in \mathbb{N}$, and suppose that either $n \in S = Q$ or $n \in S \subseteq [n-c]$. Let $T \subsetneq [n-c]$ and $w \in \Sigma^*$, and suppose that w has minimum length for $Sw = T$.*

- (i) *If the pawn at n is a resigner, then w starts with $b^c a$.*
- (ii) *If the pawn at n is a chaser and $c \neq 0$, then w starts with b^{c+1} .*
- (iii) *If the pawn at n is a chaser and $c = 0$, then w can be chosen to start with b^{c+1} .*

On account of Lemma 3.4(i), a shortest synchronizing word is of the form $b^{c+1}w$. Since $Qb^{c+1} = [n-c-1] = [n']$, the start state subset of w is $[n']$.

Suppose the yellow jersey starts in $j \in [n']$. To simplify the further investigation, we will first look to the *shortest full synchronizing word*, which is the shortest word that synchronizes all pawns into state j . Now consider an arbitrary set $S \subseteq [n']$. By Lemma 3.4(iii) and Corollary 3.5, a shortest full synchronizing word for S starts with a and can be partitioned into factors a , $b^c a$ and b^{c+1} . It does not contain b^{c+2} and therefore has a prefix of the form

$$w = aw_{n'}aw_{n'-1}a \dots aw_3aw_2aw_1aw_n, \quad \text{with} \quad w_k \in \{\varepsilon, b^c, b^{c+1}\}. \quad (3)$$

A word of this form will be called an *iteration word*. If w is a prefix of a shortest full synchronizing word for S and wb is not, then we call w an *optimal iteration word*.

Lemma 3.10. *Let w be an optimal iteration word for $S \subseteq [n']$, such that every suffix of w follows the choice in Corollary 3.9(iii) (if $c = 0$). Then*

- (i) *For $k \notin S$, we have $w_k = \varepsilon$. For $k \in S$, we have $w_k = b^c$ or $w_k = b^{c+1}$.*
- (ii) *For all $k \in S$,*

$$kw = \begin{cases} k & \text{if } w_k = b^c, \\ k+1 \pmod{n'} & \text{if } w_k = b^{c+1}. \end{cases}$$

- (iii) *If $w_{n'} = b^{c+1}$, then $w_n = b^{c+1}$. If $w_{n'} \neq b^{c+1}$, then $w_n = \varepsilon$.*

Proof. The word w has the following properties for $1 \leq k \leq n'$:

$$kw = \begin{cases} \perp & \text{if } w_k = \varepsilon \\ k & \text{if } w_k = b^c \\ k+1 & \text{if } w_k = b^{c+1}, k \neq n'. \end{cases} \quad (4)$$

Since w_k can only affect a pawn in state k , we observe that $w_k = \varepsilon$ if and only if $k \notin S$, proving (i). (ii) follows as well, except for the case where $k = n'$ and $w_{n'} = b^{c+1}$.

To complete the proof of (ii), and to prove the first claim of (iii), suppose that $w_{n'} = b^{c+1}$. Then it follows from Corollary 3.9 that the pawn in state n' is a chaser. Write $w = vw_n$. Then $n'v = n - c$ and $kv = kw \neq n - c$ for all $k \neq n'$. Therefore, the pawn under consideration did not merge yet and is still chasing. Using Corollary 3.9 again, combined with the assumption of following the choice in Corollary 3.9(iii), we infer that $w_n = b^{c+1}$. This yields the first part of (iii). Therefore, $n'w = n'vw_n = (n - c)b^{c+1} = 1$. This completes the proof of (ii).

If $w_{n'} = b^c$ or $w_{n'} = \varepsilon$, then $Sv \subseteq [n']$. By Lemma 3.4(iii), a shortest synchronizing word for S then starts with va so that $w_n = \varepsilon$, completing the proof. \square

Proof (of Theorem 3.2). The idea of Lemma 3.10 is that an iteration word can be used to decide for every pawn if it has to move one step (at the cost of $c + 1$ letters b and possibly more if we needed $w_n = b^{c+1}$), or to stay where it is (at the cost of c letters b). The optimal choice depends on the pawn being a chaser or a resigner, where we follow the choice in Corollary 3.9(iii) if $c = 0$. After applying an optimal iteration word, all pawns will be located on a subset of $[n']$. Consequently, every shortest full synchronizing word can be partitioned into iteration words.

As the yellow jersey starts in $j \in [n']$, the lanterne rouge starts in $j + 1 \pmod{n'}$. Observe that after each iteration, the lanterne rouge (being a chaser) will have moved from ℓ to $\ell + 1 \pmod{n'}$, while the yellow jersey is still at j . After $n' - 1$ iterations, both the lanterne rouge and the yellow jersey and hence all initial pawns are in j . For the shortest synchronizing word, it is sufficient to have all pawns in state 1, so we can delete a^{j-1} at the end of the shortest full synchronizing word.

Hence the number of letters a in a shortest synchronizing word equals $(n' - 1) \cdot (n' + 1) - (j - 1)$. We have used $c + 1$ letters b in the beginning and at least $f_c(n')$ letters b in all iteration words. By Lemma 3.10(iii), there is an additional cost of $c + 1$ letters b for each iteration word with $w_{n'} = b^{c+1}$. Now suppose the yellow jersey starts in $j = n'$. This minimizes the number of a 's. Furthermore, since the yellow jersey is always a resigner, $w_{n'} = b^c$ in each iteration. Consequently, there will be no additional costs for w_n , so that the minimal possible length as given in Theorem 3.2 is obtained for $j = n'$. \square

4 Recursive and Asymptotic Results

We will now turn our attention to the analysis of Problem 3.1. The following proposition gives a recursive formula for the solution.

Proposition 4.1. *The function f_c satisfies $f_c(1) = 0$ and*

$$f_c(n) = \min \{f_c(i) + f_c(n - i) + (c + 1)n - i \mid 1 \leq i \leq n - 1\}.$$

Proof. In Problem 3.1, we define chasers and resigners as before. Since we now work on \mathbb{Z} , the pawn at 1 is the lanterne rouge and the pawn at n is the yellow

jersey. In total we will need $n - 1$ iterations (or we can assume so if $c = 0$) by the following simple analog of a special case of Lemma 3.6.

Claim. Let $c \in \mathbb{N}$ and $S = [n] \subseteq \mathbb{N}^*$ be the set of pawn positions. Suppose that the pawns merge to one pawn by an optimal set of iterations. Then the following holds for each pawn in every iteration.

- (i) If the pawn is a resigner, then it will stay.
- (ii) If the pawn is a chaser and $c \neq 0$, then it will move.
- (iii) If the pawn is a chaser and $c = 0$, then we can choose it to move.

This claim can be proved with the same ideas (pawn displacement) Lemma 3.6 has been proved.

Let $\sigma_j(k)$ be the position after j iterations of the pawn that starts in k . After $n - 2$ iterations, all pawns are merged into the lanterne rouge at $n - 1$ and the yellow jersey at n . Let $I = \sigma_{n-2}^{-1}(n - 1) = \{1, 2, \dots, i\}$ (being the peloton) and $J = \sigma_{n-2}^{-1}(n) = \{i + 1, \dots, n\}$ (being the first group). See also Figure 4.

Now note that the pawn at i is a resigner until the full peloton has merged into one pawn in position i . The minimal cost for this is equal to $f_c(i)$. In each of the remaining $n - i$ iterations, this pawn will be a chaser at cost $c + 1$. Similarly, the pawn starting in $i + 1$ is a chaser until the first group has merged into one pawn in position n . This takes $n - i - 1$ iterations and the minimal cost to merge the first group is $f_c(n - i)$. In the remaining i iterations, the pawn at n is a resigner at cost c .

So the minimum cost is indeed $f_c(i) + f_c(n - i) + (c + 1)n - i$, where we have to minimize over all possible $1 \leq i \leq n - 1$. □

In Figure 4 we have presented three ways in which the minimum cost can be attained when $c = 1$ and $n = 7$ (the pawn at place 3 having two choices in the right part). Here resigners are drawn amber (light) and chasers red (dark). In the optimal races in this example, there are either 8 chasers and 13 resigners or 9 chasers and 11 resigners (counted with multiplicity). The total cost $f_1(7)$ therefore is

$$f_1(7) = 8 \cdot 2 + 13 \cdot 1 = 9 \cdot 2 + 11 \cdot 1 = 29.$$

Computing $f_1(1), \dots, f_1(5)$ by the recursion gives 0, 3, 7, 12, 17 which can be used to alternatively express the total cost $f_1(7)$ by

$$f_1(7) = f_1(5) + f_1(2) + 2 \cdot 7 - 5 = f_1(4) + f_1(3) + 2 \cdot 7 - 4 = 29.$$

In Section 5, we will express $f_c(n)$ in terms of recurrent sequences. But this is not necessary to determine the order of growth of the function $f_c(n)$.

Proposition 4.2. *For all $c \geq 0$ and $n \geq 1$, we have*

$$cn \log_2(n) \leq f_c(n) \leq (c + \frac{1}{2})n \lceil \log_2(n) \rceil.$$

Proof. Both bounds can be proved by induction, the base case for $n = 1$ being true. For the lower bound, fix n and assume that $ci \log_2(i) \leq f_c(i)$ for $i < n$. This implies that for every $1 \leq i \leq n - 1$ we have that

$$f_c(i) + f_c(n - i) + (c + 1)n - i \geq c(i \log_2(i) + (n - i) \log_2(n - i) + n).$$

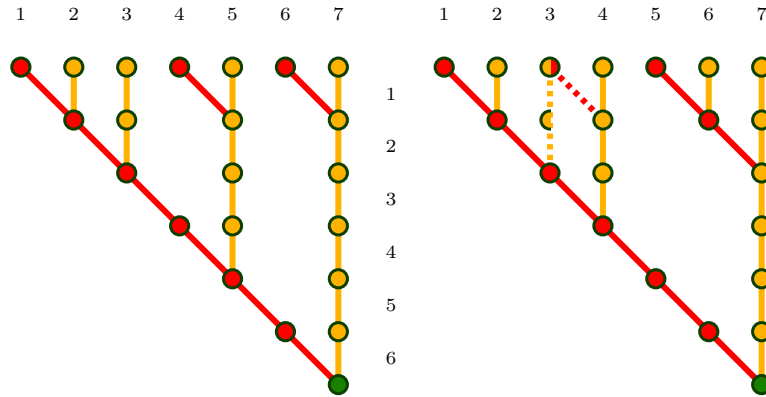


Fig. 4. The three optimal races for $n = 7$ and $c = 1$. The positions are indicated above and the iterations are numbered in the middle. The peloton has size 5 in the left race and size 4 in the other two races.

Note that $i \log_2(i) + (n-i) \log_2(n-i)$ is minimized when $i = \frac{n}{2}$ since its derivative (as a function of i for n fixed) is $\log_2(i) - \log_2(n-i)$. Plugging in $i = \frac{n}{2}$ gives $cn \log_2(n)$ on the right hand side. So, we have $cn \log_2(n) \leq f_c(n)$ and we conclude by mathematical induction.

For the upper bound, assuming it is true for values strictly smaller than n (where $n > 1$), we have

$$\begin{aligned} f_c(n) &\leq f_c\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + f_c\left(\left\lceil \frac{n}{2} \right\rceil\right) + (c+1)n - \left\lfloor \frac{n}{2} \right\rfloor \\ &\leq \left(c + \frac{1}{2}\right)n \left\lceil \log_2\left(\left\lceil \frac{n}{2} \right\rceil\right) \right\rceil + \left(c + \frac{1}{2}\right)n \\ &= \left(c + \frac{1}{2}\right)n \lceil \log_2(n) \rceil. \end{aligned}$$

So again by mathematical induction the bound does hold for every n . \square

As a corollary of Proposition 4.2, we determine the asymptotic growth of maximal reset thresholds in the Černý family.

Theorem 4.3. Denoting the reset threshold of C_n^c by $r(C_n^c)$, we have

$$\max_c r(C_n^c) \sim \frac{1}{4}n^2 \log_2(n).$$

Proof. By Theorem 3.2 and the fact that $n'(n'-1) + c + 1 \leq n^2 = o(n^2 \log_2(n))$ for $n' = n - c - 1$, it is sufficient to prove that $\max_c f_c(n - c - 1) \sim \frac{1}{4}n^2 \log_2(n)$.

By Proposition 4.2, we have for the upper bound that

$$\begin{aligned} f_c(n - c - 1) &\leq \left(c + \frac{1}{2}\right)(n - c - 1) \lceil \log_2(n - c - 1) \rceil \\ &\leq \frac{n^2}{4} \lceil \log_2(n) \rceil = (1 + o(1)) \frac{n^2}{4} \log_2(n). \end{aligned}$$

For the lower bound, we choose $c = \lfloor \frac{n-1}{2} \rfloor$, and from the lower bound in Proposition 4.2 we get that

$$\begin{aligned} f_c(n-c-1) &\geq c(n-c-1) \log_2(n-c-1) \\ &\geq \frac{(n-1)^2 - 1}{4} \log_2\left(\frac{n}{2} - 1\right) = (1 - o(1)) \frac{n^2}{4} \log_2(n). \end{aligned}$$

The two bounds together imply the result. \square

One can show that the optimal choice c' for c satisfies $|\frac{n}{2} - c'| = o(n)$. In Proposition 6.4 in Section 6, we will prove that

$$0 < \frac{n}{2} - c' = \Theta(n/\log(n)).$$

Another corollary of Proposition 4.2 is the following.

Corollary 4.4. $r(C_n^1) > r(C_n^0)$ for all $n \geq 6$. So if the Černý conjecture holds, then $p(n, 2) > d(n)$ for all $n \geq 6$.

Proof. If $6 \leq n \leq 9$, then $r(C_n^1) > r(C_n^0)$ follows from explicit computations. So assume that $n \geq 10$. Then Theorem 3.2 and the lower bound of Proposition 4.2 yields

$$\begin{aligned} r(C_n^1) &= (n-2)(n-3) + 1 + 1 + f_1(n-2) \\ &\geq (n-2)(n-3) + 1 + 1 + (n-2) \log_2(n-2) \\ &\geq (n-2)(n-3) + 2 + (n-2)3 \\ &= (n-1)^2 + 1 \end{aligned}$$

Although it is known that $r(C_n^0) = (n-1)^2$, we provide a proof of that in the next section. So $r(C_n^1) > r(C_n^0)$ holds for all $n \geq 6$. \square

5 Explicit Solution of the Pawn Race Problem

In this section, we determine the solution of Problem 3.1, from which by Theorem 3.2 the exact expressions for the reset thresholds of C_n^c for all n follow as well.

When $c = 0$, we see that the lanterne rouge has to move $n-1$ times. If the other pawns do not move, we get the minimum cost of $n-1$, i.e. $f_0(n) = n-1$. This result can also immediately be derived from Proposition 4.1. Theorem 3.2 now gives that the reset threshold of C_n^0 is indeed equal to $(n-1)(n-2) + 1 + (n-2) = (n-1)^2$, which yields an alternative proof for the well-known reset thresholds of the Černý sequence.

When $c \geq 1$, our approach is based on solving the recursion given in Proposition 4.1. In the conference version of this paper [3], we saw that the Fibonacci numbers play an important role in the solution for $c = 1$, and that the Padovan numbers play a similar role in the solution for $c = 2$. We will see that these

numbers enter the picture when determining the set of values of i for which $f_c(i) + f_c(n - i) + (c + 1)n - i$ is minimal.

Let $p : \mathbb{N}^* \rightarrow \mathbb{N}^*$ be increasing and not bounded. Define $m : \mathbb{N}^* \rightarrow \mathbb{N}^*$ by

$$m(i) = \min\{j \mid i < p(j)\}.$$

We call m the *twinverse* (twisted inverse) of p .

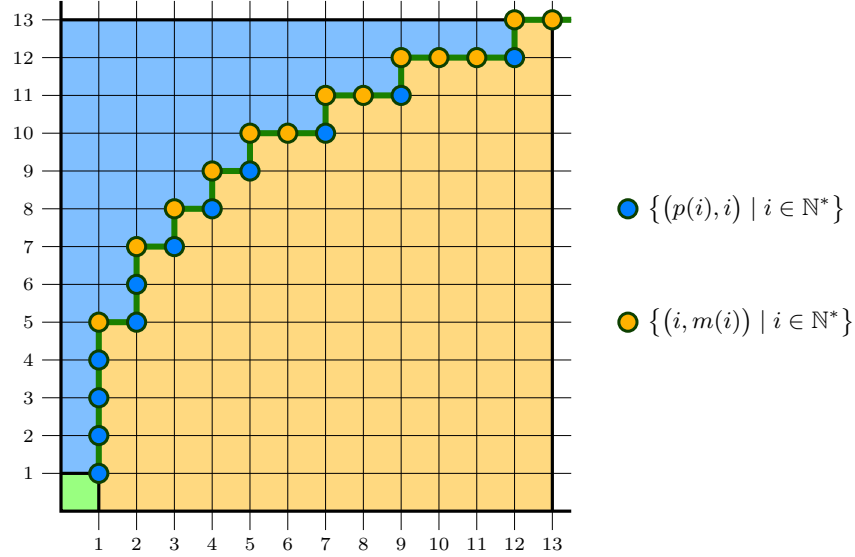


Fig. 5. Illustration of twinverses.

Proposition 5.1. *Let $p : \mathbb{N}^* \rightarrow \mathbb{N}^*$ be increasing and not bounded. Then the twinverse of p is increasing and not bounded as well. Furthermore, the twinverse of the twinverse of p is p .*

Proof. Let m be the twinverse of p , and $1 \leq i \leq i'$. Take $j = m(i)$ and $j' = m(i')$. Then $i \leq i' < p(j')$. Since $i < p(j')$, we have $j \leq j'$ by definition of $m(i)$. So m is increasing. If $i \geq p(j)$, then $m(i) > j$, so m is not bounded.

Let p' be the twinverse of m . Then

$$p'(i) = \min\{j \mid i < m(j)\}.$$

So we must show that $i < m(p(i))$, and that $q < p(i)$ implies $i \geq m(q)$. Indeed

$$m(p(i)) = \min\{j \mid p(i) < p(j)\} > i,$$

and if $q < p(i)$, then

$$m(q) = \min\{j \mid q < p(j)\} \leq i.$$

so $p' = p$. □

Lemma 5.2 below is illustrated in Figure 5, for $m = m_2$, $p = p_2$, and $n = 13$. The definition of m_2 and p_2 follows later. The total area in Figure 5 is $n \cdot m(n) = 13 \cdot 13$ and is the sum of the areas of the three colors.

Lemma 5.2. *Let m be the twinverse of p . Then*

$$\sum_{i=1}^{m(n)-1} p(i) = n \cdot m(n) - \sum_{j=1}^{n-1} m(j) - 1.$$

Proof. We show this equality by induction on n . To prove the base case $n = 1$, we must show that

$$\sum_{i=1}^{m(1)-1} p(i) = m(1) - 1.$$

If $i < m(1)$, then $p(i) = \min\{j \mid i < m(j)\} = 1$. Hence the base case follows.

To prove the induction step, we must show that

$$\sum_{i=m(n)}^{m(n+1)-1} p(i) = (n+1) \cdot (m(n+1) - m(n)).$$

If $m(n) \leq i < m(n+1)$, then $p(i) = \min\{j \mid i < m(j)\} = n+1$. Hence the induction step follows. \square

Lemma 5.3 below is the key to determining the values of i for which the minimum value in the recursive formula of Proposition 4.1 is reached. This is the most crucial part of the solution of the pawn race problem.

Lemma 5.3. *Let m be the twinverse of p . Suppose that*

$$p(k-1) + p(k) \leq n \leq p(k) + p(k+1).$$

Then

$$S(i) := \sum_{j=1}^{n-i-1} m(j) + \sum_{j=1}^{i-1} m(j) - i, \quad i \in \{1, 2, \dots, n-1\}$$

is minimal at i , if and only if

$$p(k-1) \leq n-i \leq p(k) \leq i \leq p(k+1). \quad (5)$$

Proof. Notice that

$$\Delta S(i) := S(i+1) - S(i) = m(i) - m(n-1-i) - 1.$$

We distinguish three cases for i , in order of increasing i .

- $i < p(k)$ or $p(k) < n-i$.
 - * If $i < p(k)$, then $p(k-1) \leq n-p(k) < n-i$, so $m(i) \leq k \leq m(n-i-1)$.
 - * If $p(k) < n-i$, then $i < n-p(k) \leq p(k+1)$, so $m(i) \leq k+1 \leq m(n-i-1)$.

- In both cases, $\Delta S(i) < 0$, so there is no minimum at i .
- $p(k-1) < n-i \leq p(k) \leq i < p(k+1)$.
Then $m(i) = k+1$ and $m(n-i-1) = k$. So $\Delta S(i) = 0$.
 - $n-i \leq p(k-1) \leq p(k) \leq i$ or $n-i \leq p(k) \leq p(k+1) \leq i$.
 - * If $n-i \leq p(k-1) \leq p(k) \leq i$, then $m(i) \geq k+1$ and $m(n-i-1) \leq k-1$.
 - * If $n-i \leq p(k) \leq p(k+1) \leq i$, then $m(i) \geq k+2$ and $m(n-i-1) \leq k$.
- In both cases, $\Delta S(i) > 0$, so there is no minimum at $i+1$.

We conclude that the function $S(i)$ is first decreasing, then possibly constant and then increasing. The function is minimal if and only if (5) is satisfied.

Notice that $p_c(k) \in \{n-i, i\}$ for the smallest i which satisfies (5). So $i = \max\{p_c(k), n-p_c(k)\}$ is the smallest solution of (5).

Let $c \in \mathbb{N}^*$. Our next result will express the solution $f_c(n)$ of Problem 3.1 in terms of the recursive sequence $p_c(k), k \in \mathbb{N}^*$, defined by

$$p_c(k) = \begin{cases} 1 & \text{if } 1 \leq k \leq 2c, \\ p_c(k-c-1) + p_c(k-c) & \text{if } k \geq 2c+1. \end{cases}$$

Let m_c be the twinverse of p_c for all c . The sequence p_1 are the Fibonacci numbers. With the exception of $p_2(1)$ (which does not follow the recurrence formula), the sequence p_2 is a shift of the Padovan numbers. Define the sequence $q_c(k)$ for $k \geq 1$ by

$$q_c(k) = 1 + \sum_{i=1}^{k-1} p_c(i). \quad (6)$$

Then $q_c(k) = k$ for $k \leq 2c$. For $k \geq 2c+1$, the definition of p_c gives the recursion

$$\begin{aligned} q_c(k) &= 1 + 2c + \sum_{i=2c+1}^{k-1} (p_c(i-c-1) + p_c(i-c)) \\ &= 1 + 2c + \sum_{i=c}^{k-c-2} p_c(i) + \sum_{i=c+1}^{k-c-1} p_c(i) = q_c(k-c-1) + q_c(k-c). \end{aligned}$$

From this recursion, we infer that

$$q_c(k) = q_c(k+c+1) - q_c(k+1) = \sum_{i=k+1}^{k+c} p_c(i) \quad (k \geq c).$$

In particular,

$$\begin{aligned} q_1(k) &= p_1(k+1) & (k \geq 1) & \quad (7) \\ q_2(k) &= p_2(k+1) + p_2(k+2) = p_2(k+4) & (k \geq 2). & \quad (8) \end{aligned}$$

We are now ready to formulate the main result of this section.

Theorem 5.4. *Suppose that $c \in \mathbb{N}^*$. Then*

$$f_c(n) = \sum_{j=1}^{n-1} m_c(j) = n \cdot m_c(n) - q_c(m_c(n)). \quad (9)$$

We illustrate Theorem 5.4 for $n = 7$ and $c = 1$. Since $p_1(5) = 5$ and $p_1(6) = 8$, we find $m_1(7) = 6$. Furthermore, $q_1(6) = p_1(7) = 13$ on account of (7), and therefore $f_1(7) = 7 \cdot 6 - 13 = 29$, in agreement with the example in the previous section.

Proof (of Theorem 5.4). The second equality of (9) follows from Lemma 5.2 and (6). So it remains to prove the first equality to obtain (9).

From $p_c(2c) = 1$ and $p_c(2c+1) = 2$, we infer that $m_c(1) = 2c+1$. This yields the cases $n = 1$ and $n = 2$. So assume that $n \geq 3$. We prove (9) by induction on n , so we assume that

$$f_c(i) = \sum_{j=1}^{i-1} m_c(j) \quad (i < n). \quad (10)$$

Choose k such that

$$p_c(k+c) < n \leq p_c(k+c+1). \quad (11)$$

Then $m_c(n-1) = k+c+1$. From $p_c(2c+1) = 2 < n$, we infer that $k+c+1 > 2c+1$, so $k-1 \geq c$. Hence

$$p_c(k+c) = p_c(k-1) + p_c(k), \quad p_c(k+c+1) = p_c(k) + p_c(k+1). \quad (12)$$

By (11) and (12), the condition of Lemma 5.3 is satisfied, so we can choose i such that

$$p_c(k-1) \leq n-i \leq p_c(k) \leq i \leq p_c(k+1).$$

By Proposition 4.1, the induction hypothesis (10) and Lemma 5.3, we find

$$f_c(n) = f_c(n-i) + f_c(i) + (c+1)n - i. \quad (13)$$

Below, we will find similar formulas for $f_c(n-1)$ instead of $f_c(n)$. As $n > p_c(k+c)$, at least one of two cases applies:

– *Case $i > p_c(k)$.* In this case

$$p_c(k-1) \leq n-i \leq p_c(k) \leq i-1 < p_c(k+1).$$

Hence $f_c(n-1) = f_c(n-i) + f_c(i-1) + (c+1)(n-1) - (i-1)$. So

$$f_c(n) - f_c(n-1) = m_c(i-1) + (c+1) - 1 = (k+1) + c = m_c(n-1).$$

– Case $n - i > p_c(k - 1)$. In this case

$$p_c(k - 1) \leq n - i - 1 < p_c(k) \leq i \leq p_c(k + 1).$$

Hence $f_c(n - 1) = f_c(n - i - 1) + f_c(i) + (c + 1)(n - 1) - i$. So

$$f_c(n) - f_c(n - 1) = m_c(n - i - 1) + (c + 1) = k + (1 + c) = m_c(n - 1).$$

This completes the proof of the first equality in (9).

The next result characterizes the number of optimal solutions of the pawn race. A unique optimal solution exists if and only if n is an element of the sequence p_c .

Theorem 5.5. *Let $c \geq 1$ and denote the number of optimal solutions of the pawn race by $o_c(n)$. Let $k = m_c(n) - c - 1$ and define*

$$I_n = \{i \mid p_c(k - 1) \leq n - i \leq p_c(k) \leq i \leq p_c(k + 1)\}.$$

Then $o_c(1) = o_c(2) = 1$ and for $n \geq 3$,

$$o_c(n) = \sum_{i \in I_n} o_c(n - i)o_c(i).$$

Furthermore, $o_c(n) = 1 \iff n = p_c(k')$ for some $k' \iff n = p_c(m_c(n) - 1)$.

We illustrate the result for $n = 7$ and $c = 1$. Since 7 is not a Fibonacci number, there are multiple solutions, see Figure 4. These are precisely the optimal races. Notice that $k = m_1(7) - 1 - 1 = 4$, so $p_1(k - 1) = 2$, $p_1(k) = 3$ and $p_1(k + 1) = 5$. Hence $2 \leq 7 - i \leq 3 \leq i \leq 5$. This is exactly the case if either $i = 4$ or $i = 5$, so

$$o_c(7) = o_c(2)o_c(5) + o_c(3)o_c(4).$$

Since both 2 and 5 are Fibonacci numbers, $o_c(2)o_c(5) = 1 \cdot 1 = 1$. The race which corresponds to the term $o_c(2)o_c(5) = 1$ is illustrated on the left hand side of Figure 4. The size of the first group and the peloton are 2 and 5 respectively, corresponding to the arguments of o_c . For $o_c(3)$ and $o_c(4)$, we have $o_c(3) = o_c(1)o_c(2) = 1$ and

$$o_c(4) = o_c(1)o_c(3) + o_c(2)o_c(2) = 2,$$

so $o_c(3)o_c(4) = 1 \cdot 2 = 2$. The races which correspond to the term $o_c(3)o_c(4) = 2$ are illustrated on the right hand side of Figure 4. The dashes indicate the two distinct optimal solutions for the pawn race of the peloton of size 4.

Proof (of Theorem 5.5). For $n \leq 2$, all three statements in the last line of the theorem are true. For $n \geq 3$, we distinguish two cases.

– Case $n = p_c(k')$ for some k' .

Take k' maximal as such. Then $k' = m_c(n) - 1$, which proves the last equivalence. Furthermore, $k' = k + c$, so $n = p_c(k) + p_c(k - 1)$. Hence Lemma 5.3 applies, and $I_n = \{p_c(k)\}$ follows.

– *Case $n \neq p_c(k')$ for any k' .*

Again, let $k' = m_c(n) - 1$. Then $p_c(k') < n < p_c(k' + 1)$ and $k' = k + c$, so $p_c(k - 1) + p_c(k) < n < p_c(k) + p_c(k + 1)$. In particular, Lemma 5.3 applies and $p_c(k + 1) > p_c(k - 1) + 1$. But by means of the recurrence relation of p_c , one can show by induction that $p_c(k + 1) \leq p_c(k - 1) + 1$ if either $p_c(k - 1) = p_c(k)$ or $p_c(k) = p_c(k + 1)$. Consequently,

$$p_c(k-1) < p_c(k) < n - p_c(k-1) \quad \text{and} \quad n - p_c(k+1) < p_c(k) < p_c(k+1).$$

To prove that there are at least two solutions of (5), suppose that i is a solution and $i + 1$ is not. Then either $n - i = p_c(k - 1)$ or $i = p_c(k + 1)$. In both cases, $n - i < p_c(k) < i$, so $i - 1$ is another solution and $\{i - 1, i\} \subseteq I_n$.

In both cases, the inductive formula of $o_c(n)$ holds. That the solution is unique if and only if $n = p_c(k')$ for some k' follows by induction as well. \square

The function f_c is affinely linear between consecutive values of p_c , i.e.

$$f_c(\lambda p_c(k) + (1 - \lambda)p_c(k + 1)) = \lambda f_c(p_c(k)) + (1 - \lambda)f_c(p_c(k + 1)) \quad (14)$$

for all $\lambda \in [0, 1]$ for which the left hand side makes sense. This is because $m_c(n) = k + 1$ for all n for which $p_c(k) \leq n < p_c(k + 1)$. Since $p_c(k + 1) - p_c(k)$ can be arbitrary large, f_c cannot be represented by a polynomial.

Combining the results in the current section with Theorem 3.2 gives expressions for the reset thresholds of all automata in the Černý family. Asymptotic estimates are given in the next section.

Corollary 5.6. *Suppose that $1 \leq c \leq n - 2$ and denote $n' = n - c - 1$. If w is a shortest synchronizing word for C_n^c , then*

$$|w| = r(C_n^c) = n'(n' - 1) + c + 1 + n'm_c(n') - q_c(m_c(n')).$$

Furthermore, w is unique, if and only if $n' = p_c(k')$ for some k' .

6 Estimates of the Pawn Race Problem

To estimate f_c asymptotically for $c \neq 0$, we need to take a look at the characteristic polynomial $\chi_c(x) = x^{c+1} - x - 1$ of the recurrence relation of p_c and q_c . Since $\chi_c(1) = -1 < 0 < \chi_c(2)$ and χ_c is strictly increasing for $x \geq 1$, we deduce that $\chi_c(x) = 0$ for a unique real number $x > 1$. Let ϕ_c be this number. Note that ϕ_c decreases and approaches 1 as c increases. In Proposition 4.2 we already determined the order of growth of $f_c(n)$. For fixed c , Proposition 6.1 below gives a more precise result and yields an asymptotic estimate for $n \rightarrow \infty$.

Proposition 6.1. *For all $c \geq 1$ and $n \geq 1$, we have*

$$\frac{n \log(n)}{\log(\phi_c)} - 3cn < f_c(n) < \frac{n \log(n)}{\log(\phi_c)} + (c + 1)n.$$

Proof. By mathematical induction we obtain that

$$\phi_c^{k-2c} \leq p_c(k) \leq \phi_c^{k-c} \quad (k \geq c). \quad (15)$$

Since $m_c(n) - 1 \geq 2c$ and $p_c(m_c(n) - 1) \leq n < p_c(m_c(n))$, we infer from (15) that

$$\phi_c^{m_c(n)-2c-1} \leq n < \phi_c^{m_c(n)-c} \quad (n \geq 1), \quad (16)$$

so

$$\frac{\log(n)}{\log(\phi_c)} + c < m_c(n) \leq \frac{\log(n)}{\log(\phi_c)} + 2c + 1 \quad (n \geq 1). \quad (17)$$

If $x > 1$, then $2x > x + 1 > 2\sqrt{x}$, so $2\phi_c > \phi_c^{c+1} > 2\sqrt{\phi_c}$ and $\phi_c^c < 2 < \phi_c^{c+1/2}$. Since $1 < \phi_c^c < 2$, one can prove by mathematical induction that

$$c\phi_c^{k-c} \leq q_c(k) \leq 2c\phi_c^{k-c-1} \quad (k \geq c). \quad (18)$$

If we combine this with (16), then we obtain $cn < q_c(m_c(n)) \leq 2cn\phi_c^c$, so

$$cn < q_c(m_c(n)) < 4cn \quad (n \geq 1). \quad (19)$$

Applying the bounds in (17) and (19) to Theorem 5.4 completes the proof. \square

As a consequence, we now obtain asymptotic estimates for the Černý family.

Corollary 6.2. *For all c_n such that $1 \leq c_n \leq n - 2$,*

$$r(C_n^{c_n}) = n^2 + \frac{(n - c_n)\log(n)}{\log(\phi_{c_n})} - c_n \cdot O(n).$$

Proof. In the proof of Proposition 6.1, we saw that $c < \log(2)/\log(\phi_c) < c + \frac{1}{2}$. Consequently,

$$\frac{(n - c_n - 1)\log(n - c_n - 1)}{\log(\phi_{c_n})} = \frac{(n - c_n)\log(n)}{\log(\phi_{c_n})} - c_n \cdot O(n).$$

Now it is easy to obtain the result from Theorem 3.2 and Proposition 6.1. \square

The reset thresholds $r(C_n^c)$ provide lower bounds for $p(n, 2)$. To get the best lower bounds, one should maximize over c . Just as from the proof of Theorem 4.3, one can infer from the above that $|\frac{n}{2} - c'| = o(n)$ for optimal values c' of c . We end this section with proving that $0 < \frac{n}{2} - c' = \Theta(n/\log(n))$, where we only assume that c' is a local maximum.

First we need a small lemma to compare solutions of the pawn race problem.

Lemma 6.3. *If $n' < c$, then $f_{c-1}(n' + 1) > f_c(n')$.*

Proof. If we replace c by $c + 1$ in Problem 3.1, then the price of moving becomes $1 + 1/(c + 1)$ times larger, and the price of staying becomes $1 + 1/c$ times larger. Consequently,

$$1 + \frac{1}{c + 1} < \frac{f_{c+1}(n')}{f_c(n')} < 1 + \frac{1}{c} \quad (n' \geq 2). \quad (20)$$

On account of Theorem 5.4,

$$\frac{n' \cdot m_c(n')}{f_c(n')} = \frac{f_c(n') + q_c(m_c(n'))}{f_c(n')} = 1 + \frac{q_c(m_c(n'))}{f_c(n')} \quad (n' \geq 2),$$

and

$$\frac{f_c(n' + 1)}{f_c(n')} = 1 + \frac{1}{n'} \left(1 + \frac{q_c(m_c(n'))}{f_c(n')} \right) \quad (n' \geq 2). \quad (21)$$

Since $f_{c-1}(2) > 0 = f_c(1)$, the case $n' = 1$ follows. The case $n' \geq 2$ follows from

$$\frac{f_{c-1}(n' + 1)}{f_{c-1}(n')} > 1 + \frac{1}{n'} \geq 1 + \frac{1}{c - 1} > \frac{f_c(n')}{f_{c-1}(n')}$$

which is an application of (21) and (20). \square

Proposition 6.4. *If $c = c'$ is a local maximum of $r(C_n^c)$, then $c' < \frac{n}{2}$ and $c' = \frac{n}{2} - \Theta(n/\log(n))$.*

Proof. To prove the first part, suppose that $c' \geq \frac{n}{2}$, and take $n' = n - c' - 1$. Then $c' > n'$, so by Lemma 6.3, $f_{c'-1}(n' + 1) > f_{c'}(n')$. Hence $c = c'$ is not a local maximum of $f_c(n - c - 1)$. The sum of the other terms of $r(C_n^c)$ only makes things worse, because

$$c' + (n' + 1)n' = c' - 2n' + n'(n' - 1) > c' + 1 + n'(n' - 1).$$

So $c = c'$ is not a local maximum of $r(C_n^c)$.

So it remains to prove the second part. Suppose that $c = c'$ a local maximum of $r(C_n^c)$, and $n \geq 6$. Then $0 < c' < \frac{n}{2} \leq n - 3$ on account of Corollary 4.4, so $1 \leq c' \leq n - 4$. Since $c = c'$ a local maximum of $r(C_n^c)$, it follows that

$$r(C_n^{c'-1}) \leq r(C_n^{c'}) \geq r(C_n^{c'+1})$$

Let $n' = n - c' - 1$. Using (21) and (20), we infer from $r(C_n^{c'}) \geq r(C_n^{c'+1})$ that

$$\begin{aligned} 0 &\leq \frac{(c' + 1)(n' - 1)}{f_{c'}(n' - 1)} (r(C_n^{c'}) - r(C_n^{c'+1})) \\ &= \frac{(c' + 1)(n' - 1)}{f_{c'}(n' - 1)} (f_{c'}(n') - f_{c'+1}(n' - 1) - 1 + 2(n' - 1)) \\ &\leq (c' + 1) - (n' - 1) + \frac{(c' + 1)q_{c'}(m_{c'}(n' - 1)) + \Theta(c'(n')^2)}{f_{c'}(n' - 1)} \end{aligned}$$

On account of Proposition 4.2 and (19),

$$n' - c' = 2 + \frac{O(c'n'n)}{\Theta(c'n'\log(n'))} = O(n/\log(n))$$

because $n' > \frac{n}{2} - 1$. So we may assume that $c' \geq 2$. Just like $n' - c' = O(n/\log(n))$ was obtained from $r(C_n^{c'}) \geq r(C_n^{c'+1})$, $n' - c' = \Omega(n/\log(n))$ can be obtained from $r(C_n^{c'-1}) \leq r(C_n^{c'})$. So $n' - c' = \Theta(n/\log(n))$ and $c' = \frac{n}{2} - \Theta(n/\log(n))$. \square

6A Drops in the optimal value of c

It is easier to compute $f_c(n)$ with the explicit solution of Theorem 5.4 than with the recursive formula of Proposition 4.1. We did that for all $n, c \leq 10000$, to compute $r(C_n^c)$ for all $n \leq 10000$ and $c \leq n - 2$. It appeared that the optimal choice $c = c'$ does not always increase regularly along with n . Most of the times, c' stays the same or increases 1 if n increases 1. But sometimes, c' drops significantly if n increases 1.

The first drop of c' is between $n = 47$ and $n = 48$: c' drops from 15 to 14. The next drop of c' is at $n = 99$: c' drops from 35 to 33. For $n = 99$, c' has 2 optimal values that are more than 1 apart. This does not occur for other $n \leq 10000$. c' indeed drops from 35 to 33 at $n = 99$, because the values of c' at $n = 98$ and $n = 100$ are 35 and 33 respectively.

So the value $c' = 35$ at $n = 99$ can be seen as the continuation of the value $c' = 35$ at $n = 98$. The value $c' = 33$ at $n = 99$ does not come entirely out of the blue, because $c = 32$ is a local optimal choice for $r(C_n^c)$ at $n = 98$. The value $c' = 33$ at $n = 99$ continues as the value $c' = 33$ at $n = 100$. $c = 36$ is a local optimal choice for $r(C_n^c)$ at $n = 100$, so it can be seen as the continuation of the value $c' = 35$ at $n = 99$.

For larger n , there are intervals $[n_1, n_2]$ in the range of values of n , on which there are 2 tracks of local optimal values of c . For $n = n_1$, the track with the highest value of c gives the optimal value. This track is also the continuation of the optimal values of c for $n < n_1$: the other track appears out of the blue. For $n = n_2$, the track with the lowest value of c gives the optimal value. This track also continues with optimal values of c for $n > n_2$: the other track disappears into thin air. Figure 6 illustrates this phenomenon for $n_1 = 1709$ and $n_2 = 1768$. Somewhere between n_1 and n_2 , the optimal value of c switches from the high-valued track to the low-valued track and drops.

The following table indicates these drops. The table indicates the tracks of local optimal values of c as follows. The first track starts at $n = 2$ and ends at $n = 47$, and is optimal all the time. The $(i + 1)^{\text{th}}$ track starts at $n = n_1$ in the i^{th} row and last but one column, and becomes optimal at the value of n in the i^{th} row and last column. The $(i + 1)^{\text{th}}$ track is optimal for the last time at the value of n in the $(i + 1)^{\text{th}}$ row and first column, and the track ends at $n = n_2$ in the $(i + 1)^{\text{th}}$ row and second column.

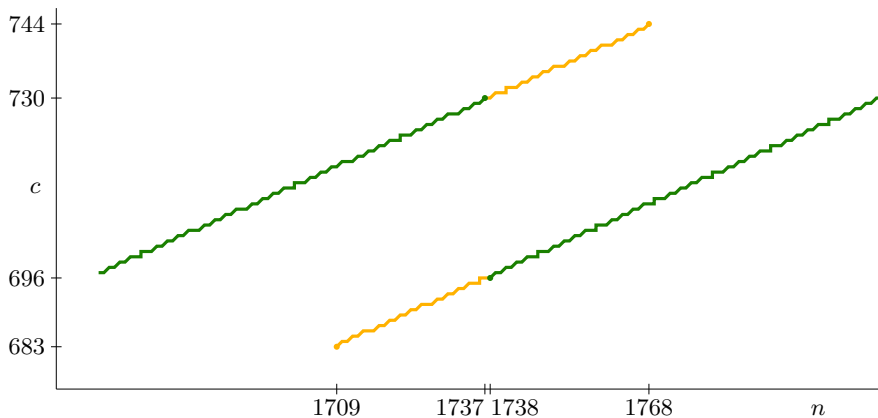


Fig. 6. For $n = 1664, 1665, \dots, 1813$, the optimal values of c are drawn in dark green. Other local optimal values of c are drawn in amber. The graph displays 2 tracks of local optimal values of c . The high-valued track ends at $n = 1768$. The low-valued track begins at $n = 1709$. The optimal value of c for $n = 1738$ is 34 lower than the optimal value of c for $n = 1737$.

n	n_2	$r(C_n^{c'})$	c'	drop	c'	$r(C_n^{c'})$	n_1	n
47	47	3331	15	1	14	3490	48	48
99	100	17323	35	2	33	17323	98	99
204	207	84024	78	5	73	84936	202	205
418	426	396403	166	9	157	398437	412	419
854	869	1836388	350	17	333	1841006	840	855
1737	1768	8347386	730	34	696	8357520	1709	1738
3524	3583	37445730	1508	64	1444	37468248	3468	3525
7132	7246	166023725	3097	120	2977	166072093	7024	7133

Observe that (local) optimal values of c can be double: $c + 1$ can also be a (local) optimal value. This is the case for $n = 13$, where both $c = 2$ and $c = 3$ are optimal (see also Figure 2). We did not find triple (local) optimal values, though. For $n = 3512$, both the optimal value of c and the other local optimal value of c are double:

$$r(C_{3512}^{1438}) = 37170635 = r(C_{3512}^{1439}) \quad r(C_{3512}^{1502}) = 37180596 = r(C_{3512}^{1503}).$$

This is the only value of $n \leq 10000$ for which this occurs.

7 An improvement of Martyugin’s prime number construction of binary PFAs

Although the Černý family contains extremal binary PFAs for all $n \leq 10$, it only gives polynomial reset thresholds for large n . In this section and the next

section, we show that for $n \geq 41$, the Černý family is not extremal anymore. We do this by presenting a construction based on prime numbers.

Our construction is an improvement of the binary prime number construction by Pavel Martyugin in [6], see also §6 of [1]. Martyugin's construction uses $1 + 2 \sum_{i=1}^r p_i$ states and has reset threshold $2 \prod_{i=1}^r p_i$, where p_1, \dots, p_r are the first r primes, which is improved to $2 + 2 \prod_{i=1}^r p_i$ in [1]. Martyugin has a ternary construction as well, with $1 + \sum_{i=1}^r p_i$ states and reset threshold $1 + \prod_{i=1}^r p_i$.

Let the binary PFA with $1 + 2 \sum_{i=1}^r p_i$ states and reset threshold $2 + 2 \prod_{i=1}^r p_i$ be called M^r . For instance, taking $r = 5$ gives $p_1, \dots, p_5 = 2, 3, 5, 7, 11$ and yields a PFA M^5 with $n = 57$ states and reset threshold 4622. This is not yet sufficient to overtake the Černý family, since $r(C_{57}^{18}) = 5152 > 4622$.

Martyugin's construction can be generalized in a very easy manner, because we do not need to restrict ourselves to the use of the first r primes. One can use any list $\mathbf{p} = (p_1, p_2, \dots, p_r)$ of relatively prime numbers, and the construction still works. Denote the corresponding PFA as $M^{\mathbf{p}}$ and let $q := q(\mathbf{p}) = \prod_{i=1}^r p_i$. Compared to M^r , $M^{\mathbf{p}}$ offers more flexibility in the choice of numbers, leading to constructions for more state sets. But larger reset thresholds are possible as well. For instance, if $\mathbf{p} = (2, 3, 5, 7, 11, 13, 17, 19, 23)$, then $M^9 = M^{\mathbf{p}}$ has 201 states, and reset threshold $2 + 2q = 446185742$. But if $\mathbf{p} = (5, 7, 9, 11, 13, 16, 17, 19)$, then $M^{\mathbf{p}}$ only has 195 states, and reset threshold $2 + 2q = 465585122$.

We present a further improvement, based on a list $\mathbf{p} = (p_1, p_2, \dots, p_r)$ of relatively prime numbers as well. The construction of our binary PFA $P^{\mathbf{p}}$ is illustrated in Figure 7 for $r = 4$ and $p_1 = 5, p_2 = 7, p_3 = 8$ and $p_4 = 9$.

The state set Q of the PFA $P^{\mathbf{p}}$ consists of r groups. The i^{th} group contains $p_i + 3$ states, namely

$$p_i 0, p_i 1, p_i 2, \dots, p_i p_i, p_i A, p_i B$$

The transitions are indicated in Figure 7. Groups i and $i + 1$ are connected by a transition of symbol a for each i . Notice that the connection for $i = r - 1$ differs from those for $i < r - 1$. This makes the reset threshold a little larger.

Symbol a is undefined on state $p_i B$ for each i . For that reason, any synchronizing word starts with b^3 . The word b^3 resets group i to state $p_i 0$ for each i , so $Qb^3 = \{p_1 0, p_2 0, \dots, p_r 0\}$. Furthermore, $Qb^3 a^q = \{p_1 p_1, p_2 p_2, \dots, p_r p_r\}$ and $Qb^3 a^q b = \{p_1 A, p_2 A, \dots, p_r A\}$. We infer that $Qb^3 a^q b a$ is defined and that it does not contain states of group 1. By induction on r , it follows that the PFA $P^{\mathbf{p}}$ is synchronizing.

Suppose that w is a shortest synchronizing word which starts with $b^3 a^j b$ for some $j < q$. If $j = 0$, then $Qb^3 a^j b = Qb^3$, which contradicts the minimality of $|w|$. So $j > 0$. Since $0 < j < q$, it follows that there exists an i such that $(p_i 0) a^j \neq (p_i p_i)$. Hence $(p_i 0) a^j b = (p_i B)$. From this, one can infer that either $Qb^3 a^j b^2 = Qb^3$ or $Qb^3 a^j b^3 = Qb^3$, and that $|w|$ is not minimal, which is a contradiction. So w starts with $b^3 a^q$, and $|w| = \Omega(q)$. It is not hard to see that $|w| = O(q)$, so $|w| = \Theta(q)$.

It is a little harder to find the exact reset threshold given in the next proposition.

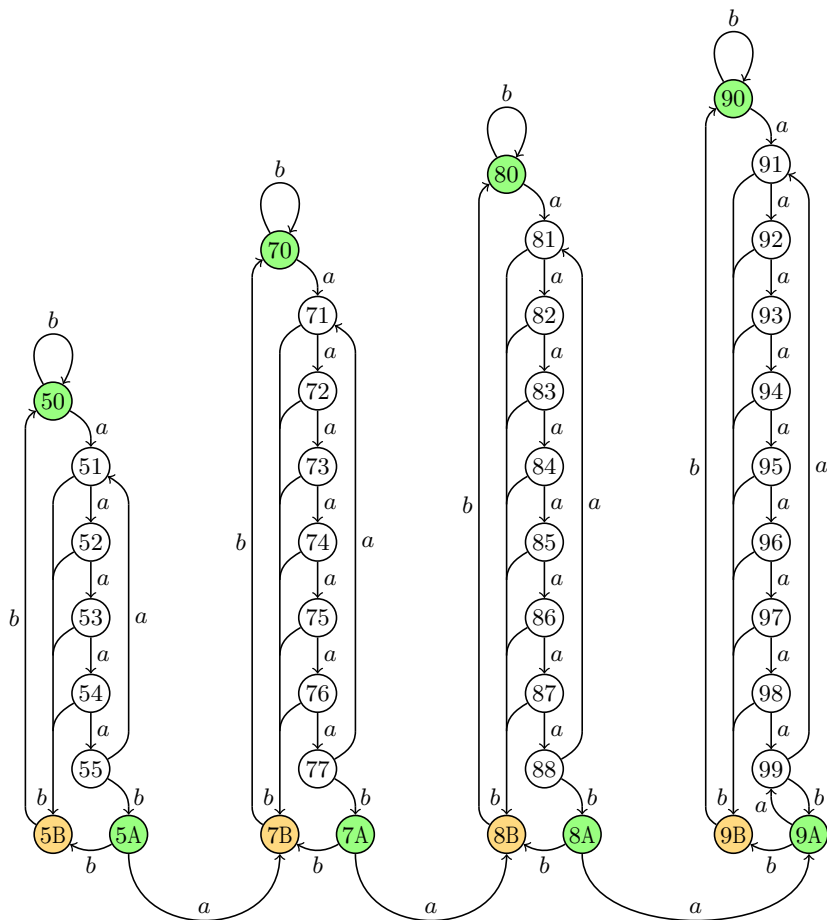


Fig. 7. A binary PFA with 41 states which takes 3114 steps to synchronize, which is more than any PFA of the Černý family with 41 states.

Proposition 7.1. *If the PFA $P^{\mathbf{p}}$ is constructed with relative primes p_1, \dots, p_r , then it has $3r + \sum_{i=1}^r p_i$ states and reset threshold*

$$r(P^{\mathbf{p}}) = 5r - 2 + \sum_{i=1}^{r-1} p_i p_{i+1} \cdots p_r.$$

The proof (by induction) is left as an exercise to the interested reader.
 Let P^r be the PFA $P^{\mathbf{p}}$ with \mathbf{p} the list of the first r primes (in increasing order). The following proposition shows that M^r is defeated by P^{r+1} if $r \geq 6$, and that $M^{\mathbf{p}}$ is defeated if it has at least 62 states.

Proposition 7.2.

- (i) Suppose that $r \geq 6$. Then P^{r+1} has fewer states and larger reset threshold than M^r .
- (ii) Suppose that $M^{\mathbf{P}}$ has at least 62 states. Then there exists a prime power $p \geq 5$ with the following properties: the list \mathbf{p} can be extended with one element by inserting p at any position to obtain \mathbf{p}' , and $P^{\mathbf{p}'}$ has fewer states and larger reset threshold than $M^{\mathbf{P}}$.

Proof. Let $n(A)$ denote the number of states of a PFA A . First observe that $r(P^{r+1}) \geq p_{r+1} \cdot \prod_{i=1}^r p_i > 2 + 2 \cdot \prod_{i=1}^r p_i = r(M^r)$, where p_i is the i^{th} prime for each i . Similarly, $r(P^{\mathbf{p}'}) \geq 5 \cdot \prod_{p \in \mathbf{p}} p > r(M^{\mathbf{P}})$. So it remains to show that $n(P^{r+1}) < n(M^r)$ and $n(P^{\mathbf{p}'}) < n(M^{\mathbf{P}})$.

- (i) We prove $n(P^{r+1}) < n(M^r)$ by induction on r . The PFA M^6 has $2 \cdot (2+3+5+7+11+13)+1 = 83$ states, and P^7 has $2+3+5+7+11+13+17+7 \cdot 3 = 79$ states, so the case $r = 6$ is satisfied. Suppose that $r \geq 7$ and that P^r has fewer states than M^{r-1} . Then M^r has $2p_r$ more states than M^{r-1} , and P^{r+1} has $p_{r+1} + 3$ more states than P^r . From Bertrand's postulate, it follows that $2p_r \geq p_{r+1} + 3$, so $n(P^{r+1}) < n(M^r)$.
- (ii) Let r be the number of elements in \mathbf{p} and let t be the number of distinct prime factors in these elements. We prove $n(P^{\mathbf{p}'}) < n(M^{\mathbf{P}})$ by distinguishing the cases $t \leq 5$ and $t \geq 6$.
Suppose first that $t \geq 6$. Since $\sum_{p \in \mathbf{p}} p$ is at least the sum of the first t prime numbers and $t \geq r$, we infer that

$$n(M^{\mathbf{P}}) - n(P^{\mathbf{P}}) = 1 - 3r + \sum_{p \in \mathbf{p}} p \geq n(M^t) - n(P^t).$$

Since the $(t+1)^{\text{th}}$ prime exceeds both 8 and 9 (the 7th prime is 17), we can construct \mathbf{p}' by choosing a prime power $p \geq 5$ which is at most the $(t+1)^{\text{th}}$ prime. So

$$n(P^{\mathbf{p}'}) - n(P^{\mathbf{P}}) \leq n(P^{t+1}) - n(P^t).$$

Consequently,

$$n(M^{\mathbf{P}}) - n(P^{\mathbf{p}'}) \geq n(M^t) - n(P^{t+1}).$$

So the case $t \geq 6$ follows from the first claim.

Suppose next that $t \leq 5$. Assume that $n(M^{\mathbf{P}}) \geq 62$. Then

$$n(P^{\mathbf{P}}) = \frac{1}{2}(n(M^{\mathbf{P}}) - 1) + 3r < n(M^{\mathbf{P}}) - 31 + 15.$$

Construct \mathbf{p}' by taking for p the smallest power ≥ 5 of the smallest prime number that does not yet occur as a factor in \mathbf{p} . Then

$$n(P^{\mathbf{p}'}) < n(M^{\mathbf{P}}) - 16 + p + 3 \leq n(M^{\mathbf{P}}),$$

because $t+1 \leq 6$ and the 6th prime is 13. □

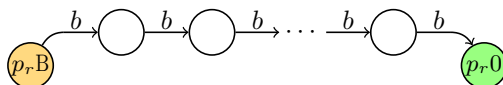
Just like for $M^{\mathbf{P}}$ and M^r , $P^{\mathbf{P}}$ offers more flexibility and sometimes a better reset threshold than P^r . For instance, P^4 has 29 states and $r(P^4) = 368$, and P^5 has 43 states and $r(P^5) = 3950$. So P^r does not exist with 41 states, and adding additional states to P^4 to fix that yields a poor result. But if $\mathbf{p} = (5, 7, 8, 9)$, then $P^{\mathbf{P}}$ has 41 states as well, and $r(P^{\mathbf{P}}) = 3114$, see Figure 7. The following is an example where $P^{\mathbf{P}}$ has a better reset threshold than P^r . If $\mathbf{p} = (2, 3, 5, 7, 11, 13, 17)$, then $P^7 = P^{\mathbf{P}}$ has 79 states, and $r(P^7) = r(P^{\mathbf{P}}) = 870552$. But if $\mathbf{p} = (5, 7, 9, 11, 13, 16)$, then $P^{\mathbf{P}}$ has the same number of states, and $r(P^{\mathbf{P}}) = 887980$. The order of the primes in \mathbf{p} is relevant to obtain a value which exceeds 870552.

The construction $P^{\mathbf{P}}$ (in particular P^r) is not transitive, but we can make it transitive with the following modification. We change $(p_i 0)a$ for all i from $(p_i 0)a = (p_i 1)$ to

$$(p_1 0)a = (p_r 1) \qquad (p_i 0)a = (p_{i-1} 1) \qquad (2 \leq i \leq r)$$

One can verify that this modification gives a transitive PFA $\tilde{P}^{\mathbf{P}}$, which still satisfies $r(\tilde{P}^{\mathbf{P}}) = \Theta(q)$. It is not hard to prove that $r(\tilde{P}^{\mathbf{P}}) \geq q + q / \max\{p_1, p_2, \dots, p_r\}$.

If we want to have extra states as well (to obtain a specific n), then we add them in such a way that transitivity is preserved. We add such states between state $p_r B$ and state $p_r 0$, as follows.



So symbol a is undefined on the extra states. Adding additional states this way makes the reset threshold larger, but only marginally. So $|w| = \Theta(q)$ is not affected.

We can use $|w| = \Theta(q)$ to derive an asymptotic estimate for both Martyugin’s construction and our construction. This has already been done for M^r and its ternary variant in [1], and P^r can be settled in a similar manner. But the estimate in [1] does not distinguish between M^r and P^r . The estimates for $r(M^r)$, its ternary variant, and $r(P^r)$, as given in [1], are

$$\exp(\Theta(1) \cdot \sqrt{n \cdot \ln(n)}).$$

The upper bound is valid for $r(M^{\mathbf{P}})$ and $r(P^{\mathbf{P}})$ as well. For the lower bound, an extra condition is required, since one can make very poor constructions, typically with $r = 1$, where r is the number of elements of \mathbf{p} . A condition which suffices is $n = O(r^2 \log(r))$. Notice that the r^{th} prime is $\Theta(r \log(r))$. So $n = O(r^2 \log(r))$ holds if p_i is at most proportional to the r^{th} prime for each i .

The above estimates were derived to show that Martyugin’s construction was strictly between polynomial and exponential. But to compare Martyugin’s construction and our construction, we need better estimates. To derive such estimates, we will use some classical results of number theory. After that, one can easily conclude that the reset threshold of our construction is better than Martyugin’s construction, by a factor which lies strictly between polynomial and exponential.

Theorem 7.3. *Let $n \in \mathbb{N}$ and suppose that r is maximal such that $n(M^r) \leq n$. Let \mathbf{p} be such that $n(M^{\mathbf{P}}) \leq n$. Then*

$$r(M^r) = \exp\left((1 \pm o(1))\sqrt{n \cdot \ln(n)/2}\right), \quad r(M^{\mathbf{P}}) \leq \exp\left((1 + o(1))\sqrt{n \cdot \ln(n)/2}\right).$$

Let $n \in \mathbb{N}$ and suppose that r is maximal such that $n(P^r) \leq n$. Let \mathbf{p} be such that $n(P^{\mathbf{P}}) \leq n$. Then

$$r(P^r) = \exp\left((1 \pm o(1))\sqrt{n \cdot \ln(n)}\right), \quad r(P^{\mathbf{P}}) \leq \exp\left((1 + o(1))\sqrt{n \cdot \ln(n)}\right).$$

Proof. We only prove the second claim, because the proof of the first claim is similar. Using the asymptotic estimate $\sum_{i=1}^r p_i \sim \frac{1}{2}r^2 \ln(r)$, we infer that

$$n \sim n(P^r) \sim \frac{1}{2}r^2 \ln(r) + O(r) = r^{2+o(1)}.$$

Consequently,

$$n \sim \frac{1}{2}r^2 \ln(r) = \frac{1}{2}r^2 \frac{\ln(n)}{2 + o(1)} \sim \frac{1}{4}r^2 \ln(n),$$

and

$$r \sim \sqrt{4n/\ln(n)} = n^{(1-o(1))/2}.$$

Using the asymptotic estimate $\prod_{i=1}^r p_i \sim r^{(1 \pm o(1))r}$ and $r(P^r) = \Theta\left(\prod_{i=1}^r p_i\right)$, we obtain

$$r(P^r) \sim r^{(1 \pm o(1))r} = n^{(1 \pm o(1))r/2}.$$

This yields the estimate for $r(P^r)$.

We proceed with the estimate for $r(P^{\mathbf{P}})$. Notice that \mathbf{p} contains $r' \leq r$ relatively prime numbers. So $\prod_{p \in \mathbf{p}} p$ is the product of r' numbers whose sum is less than n . This product is at most $(n/r')^{r'}$. Since $n/r' \geq n/r \geq \exp(1)$ for large n , we infer that $(n/r')^{r'} \leq (n/r)^r$ for large n . So by Proposition 7.1,

$$r(P^{\mathbf{P}}) \leq r \cdot \prod_{p \in \mathbf{p}} p \leq r(n/r)^r = r(n^{(1+o(1))/2})^r = n^{(1+o(1))r/2},$$

which yields the estimate for $r(P^{\mathbf{P}})$.

Martyugin's ternary construction satisfies the same estimate as our binary construction.

8 Binary PFAs with larger reset thresholds for $n \geq 41$ states

Now that we have the improved prime number construction, it remains to prove that \mathbf{p} can be chosen such that $P^{\mathbf{P}}$ and $\tilde{P}^{\mathbf{P}}$ defeat the Černý family for all $n \geq 41$. In order to do this, we will construct a series of PFAs of this type, for which $n = O(r^2 \log(r))$ does *not* hold. Instead, we construct PFAs of which the reset threshold is only $\Theta(n^3)$. This is sufficient and more easy.

To show for any construction that it defeats the Černý family for large enough n , we first need an upper bound for the reset thresholds of the automata in the Černý family which is valid for all c , but does not depend on c . Again, let $n' = n - c - 1$. From Theorem 3.2 and Proposition 4.2, it follows that

$$\begin{aligned} r(C_n^c) &\leq n'(n' - 1) + c + 1 + f_c(n') \\ &\leq (n')^2 + (c + 1)^2 + (c + 1)n' \lceil \log_2(n) \rceil \\ &= (n' + c + 1)^2 + (c + 1)n' \lceil \log_2(n) - 2 \rceil \\ &\leq \frac{1}{4}n^2 \lceil \log_2(n) + 2 \rceil. \end{aligned}$$

This is not really a good upper bound, but it will be sufficient for all $n \geq 47$. For $41 \leq n \leq 46$, our general argument will not be precise enough.

We first assume that $n \geq 47$. Notice that $\lceil \log_2(52) + 2 \rceil = 8$, and that

$$\lceil \log_2(m + 52) + 2 \rceil \leq \frac{1}{10}m + 8 \leq \frac{2}{9}(m + 36)$$

for all $m \in \mathbb{N}$. Now take $m := \max\{n - 52, 0\}$. Then it follows from the above that

$$\begin{aligned} 54r(C_n^c) &\leq 3(m + 36)(m + 52)(m + 52) \\ &\leq 3(m + 37)(m + 51)(m + 52) = (m + 51)(m + 52)(3m + 111). \end{aligned} \tag{22}$$

Let $i \in \mathbb{N}$. The PFAs $P^{\mathbf{P}}$ and $\tilde{P}^{\mathbf{P}}$ of the 4 relatively prime numbers $8, 7 + 2i, 9 + 2i, 11 + 2i$ have

$$(3 + 8) + (3 + 7 + 2i) + (3 + 9 + 2i) + (3 + 11 + 2i) = 47 + 6i$$

states. For all $47 + 6i \leq n < 53 + 6i$, we can use the same 4 relatively prime numbers in our constructions of $P^{\mathbf{P}}$ and $\tilde{P}^{\mathbf{P}}$ with $n \geq 47$ states. From $i \geq 0$ and $6i \geq n - 52$, we infer that $2i \geq \frac{m}{3}$. So $q \geq 8(\frac{m}{3} + 7)(\frac{m}{3} + 9)(\frac{m}{3} + 11)$ and

$$\begin{aligned} 54q &\geq 16(m + 21)(m + 27)(m + 33) \\ &\geq 16(m + 21)(m + 26)(m + 34) = (\frac{17}{7}m + 51)(2m + 52)(\frac{56}{17}m + 112). \end{aligned} \tag{23}$$

By way of a factor comparison with (22), we see that $P^{\mathbf{P}}$ and $\tilde{P}^{\mathbf{P}}$ defeat the Černý family if the number of states is at least 47.

With $41 \leq n \leq 46$ states, we can choose specific relatively prime numbers, and compare the reset threshold of the best PFA of the Černý family, say $C_n^{c'}$, with the product q . The columns with r.t. and r.t. t. give the reset thresholds for $P^{\mathbf{P}}$ and the transitive variant $\tilde{P}^{\mathbf{P}}$ respectively.

n	c'	$r(C_n^{c'})$	q	r.t.	r.t. t.	primes	n
41	13	2465	2520	3114	3056	(5,7,8,9)	41
42	13	2601	2520	3117	3062	(5,7,8,9)	42
43	13	2739	3080	3802	3726	(5,7,8,11)	43
44	14	2882	3465	4275	4177	(5,7,9,11)	44
45	14	3028	3960	4869	4683	(5,8,9,11)	45
46	15	3177	3960	4872	4689	(5,8,9,11)	46

It appears that the value of q is insufficient for estimation if $n = 42$. But both constructions $P^{\mathbf{P}}$ and $\tilde{P}^{\mathbf{P}}$ require at least $5 \cdot 7 \cdot 8 \cdot 9 + 5 \cdot 7 \cdot 8 = 2800$ steps to synchronize if $n = 42$, which is sufficient for estimation.

9 Conclusion

The Černý family presented in this paper contains for all $n \leq 10$ a binary PFA with n states and maximal possible reset threshold. The analysis for the different members of the family has been done in general, by determining the maximal reset threshold in terms of recurrent sequences.

We also have shown that for $n \geq 41$ the Černý family does not contain extremal PFAs anymore. This is proved by a new prime number construction which outperforms earlier known constructions. For large n , there are constructions based on rewrite systems as introduced in [2] with exponentially large reset thresholds, but they are insufficient to beat the Černý family for all $n \geq 41$.

We leave it as an open question if the Černý family contains any extremal PFA for $11 \leq n \leq 40$. The largest reset thresholds of the Černý family are given below.

n	11	12	13	14	15	16	17	18	19	20
$r(C'_n)$	119	146	176	211	248	288	332	379	429	483

n	21	22	23	24	25	26	27	28	29	30
$r(C'_n)$	539	599	663	732	804	881	961	1044	1132	1222

n	31	32	33	34	35	36	37	38	39	40
$r(C'_n)$	1317	1416	1517	1624	1733	1846	1963	2082	2207	2334

References

- de Bondt, M.: Subset synchronization of DFAs and PFAs, and some other results (2018), available at <http://arxiv.org/abs/1807.04661>
- de Bondt, M., Don, H., Zantema, H.: Lower bounds for synchronizing word lengths in partial automata. *Internat. J. Found. Comput. Sci.* **30**(1), 29–60 (2019). <https://doi.org/10.1142/S0129054119400021>, <https://doi.org/10.1142/S0129054119400021>
- Cambie, S., de Bondt, M., Don, H.: Extremal binary PFAs in a Černý family. In: Moreira, N., Reis, R. (eds.) *Developments in Language Theory*. pp. 78–89. Springer International Publishing, Cham (2021)
- Černý, J.: Poznámka k homogénnym experimentom s konečnými automatmi. *Matematicko-fyzikálny časopis, Slovensk. Akad. Vied* **14**(3), 208–216 (1964)
- Frankl, P.: An extremal problem for two families of sets. *European Journal of Combinatorics* **3**, 125–127 (1982)

6. Martyugin, P.V.: Lower bounds for the length of the shortest carefully synchronizing words for two- and three-letter partial automata. *Diskretn. Anal. Issled. Oper.* **15**(4), 44–56, 99 (2008)
7. Pin, J.E.: On two combinatorial problems arising from automata theory. *Annals of Discrete Mathematics* **17**, 535–548 (1983)
8. Shabana, H.: Exact synchronization in partial deterministic automata. *Journal of Physics: Conference Series* **1352**, 012047 (oct 2019). <https://doi.org/10.1088/1742-6596/1352/1/012047>, <https://doi.org/10.1088/1742-6596/1352/1/012047>
9. Shitov, Y.: An improvement to a recent upper bound for synchronizing words of finite automata. *J. Autom. Lang. Comb.* **24**(2-4), 367–373 (2019). <https://doi.org/10.15388/na.2019.3.3>, <https://doi.org/10.15388/na.2019.3.3>
10. Volkov, M.: Synchronizing automata and the Černý conjecture. In: *Proceedings of LATA*. Springer LNCS, vol. 5196, pp. 11–27 (2008)
11. Vorel, V.: Subset synchronization and careful synchronization of binary finite automata. *Int. J. Found. Comput. Sci.* **27**(5), 557–578 (2016)