On Stabilization in Herman's Algorithm*

Stefan Kiefer[1], Andrzej S. Murawski[2], Joël Ouaknine[1],
James Worrell[1], and Lijun Zhang[3]

[1] Department of Computer Science, University of Oxford, UK
[2] Department of Computer Science, University of Leicester, UK
[3] DTU Informatics, Technical University of Denmark, Denmark

Abstract. Herman's algorithm is a synchronous randomized protocol for achieving self-stabilization in a token ring consisting of N processes. The interaction of tokens makes the dynamics of the protocol very difficult to analyze. In this paper we study the expected time to stabilization in terms of the initial configuration.
It is straightforward that the algorithm achieves stabilization almost surely from any initial configuration, and it is known that the worst-case expected time to stabilization (with respect to the initial configuration) is $\Theta(N^2)$. Our first contribution is to give an upper bound of $0.64N^2$ on the expected stabilization time, improving on previous upper bounds and reducing the gap with the best existing lower bound. We also introduce an asynchronous version of the protocol, showing a similar $O(N^2)$ convergence bound in this case.
Assuming that errors arise from the corruption of some number k of bits, where k is fixed independently of the size of the ring, we show that the expected time to stabilization is $O(N)$. This reveals a hitherto unknown and highly desirable property of Herman's algorithm: it recovers quickly from bounded errors. We also show that if the initial configuration arises by resetting each bit independently and uniformly at random, then stabilization is significantly faster than in the worst case.

1 Introduction

Self-stabilization is a concept of fault-tolerance in distributed computing. A system is self-stabilizing if, starting in an arbitrary state, it reaches a correct or legitimate state and remains in a legitimate state thereafter. Thus a self-stabilizing system is able to recover from *transient errors* such as state-corrupting faults. The study of self-stabilizing algorithms originated in an influential paper of Dijkstra [4]. By now there is a considerable body of work in the area, see [18, 5].

In this paper we consider self-stabilization in a classical context that was also treated in Dijkstra's original paper—a *token ring*, i.e., a ring of N identical processes, exactly one of which is meant to hold a token at any given time. If, through some error, the ring enters a configuration with multiple tokens, self-stabilization requires that the system be guaranteed to reach a configuration with

* Research supported by EPSRC (EP/G069158/1). Stefan Kiefer is supported by a postdoctoral fellowship of the German Academic Exchange Service (DAAD).

only one token. In particular, we are interested in analyzing a self-stabilization algorithm proposed by Herman [11].

Herman's algorithm is a randomized procedure by which a ring of processes connected uni-directionally can achieve self-stabilization almost surely. The algorithm works by having each process synchronously execute the following action at each time step: if the process possesses a token then it passes the token to its clockwise neighbor with probability 1/2 and keeps the token with probability 1/2. If such a process decides to keep its token and if it receives a token from its neighbor then the two tokens are annihilated. Due to the way the algorithm is implemented we can assume that an error state always has an odd number of tokens, thus this process of pairwise annihilation eventually leads to a configuration with a single token.

While the almost-sure termination of Herman's algorithm is straightforward, computing the time to termination is a challenging problem. This is characteristic of systems of interacting particles under random motion, which are ubiquitous in the physical and medical sciences, including statistical mechanics, neural networks and epidemiology [15]. The analysis of such systems typically requires delicate combinatorial arguments [6]. Our case is no exception, and we heavily exploit work of Balding [1], which was motivated by a scenario from physical chemistry.

Given some initial configuration, let $\mathbf{T}$ be the time until the token ring stabilizes under Herman's algorithm. We analyze the expectation of $\mathbf{T}$ in three natural cases: the worst case (over all initial configurations); the case in which the initial configuration is chosen uniformly at random; the case in which the initial configuration arises from a legitimate configuration by a bounded number of bit errors. In addition we introduce and analyze an asynchronous variant of Herman's algorithm. The latter dispenses with the successive time steps required in the synchronous algorithm, and instead has each process pass its token after an exponentially distributed time delay.

Herman's original paper [11] showed that $\mathbb{E}\mathbf{T} \leq (N^2 \log N)/2$ in the worst case (i.e., over all initial configurations with N processes). It also mentions an improved upper bound of $O(N^2)$ due to Dolev, Israeli, and Moran, without giving a proof or a further reference. In 2005, three papers [9, 16, 17] were published, largely independently, all of them giving improved $O(N^2)$ bounds. The paper [16] also gives a lower bound of $4N^2/27$, which is the expected stabilization time starting from a configuration with three equally spaced tokens. It was conjectured in [16] that this is the worst case among all starting configurations, including those with more than three tokens. This intriguing conjecture is supported by experimental evidence [2].

Our first result, Theorem 2, gives an upper bound of $0.64N^2$ for the expected stabilization time in the synchronous version of Herman's protocol (improving the constant in the hitherto best bound by a third). We also give an upper bound in the asynchronous case. To the best of our knowledge this is the first analysis of an asynchronous version of Herman's algorithm.

To understand the other main results of the paper requires some detail of the implementation of Herman's algorithm. We assume that each process has a bit that it can read and write, and that each process can read the bit of its counterclockwise neighbor. A process's bit does not directly indicate the presence of a token, rather a process has a token if it has the same bit as its counterclockwise neighbor. Token passing is then implemented by having processes flip their bits.

In Theorem 7 we provide an upper bound on the expected time to stabilize starting from the random initial configuration, that is, the configuration in which each process's bit is reset independently and uniformly at random. Herman's algorithm is such that the random configuration is obtained in one step from the *full configuration*, i.e., the configuration in which every process has a token. The upper bound for the random configuration is far better than the worst-case bound in Theorem 2; in particular, there are three-token configurations for which $\mathbb{E}\mathbf{T}$ is provably larger than the upper bound for the random configuration.

In Theorem 8 we show that for configurations that are obtained from a legitimate configuration by flipping a constant number of process bits, we have $\mathbb{E}\mathbf{T} = O(N)$; i.e., the expected *restabilization* time is linear in N. This contrasts with the fact that there are configurations, even with only three tokens, that need $\Omega(N^2)$ expected time for self-stabilization. Intuitively, our result points at a highly desirable—and, to the best of our knowledge, previously unknown—feature of Herman's protocol: it recovers quickly from bounded errors. This is related to the notion of a *time adaptive protocol* from [14], which refers to a protocol whose recovery time depends on the number of state-corrupted nodes rather than the total number of nodes.

Full proofs are given in [13].

Related Work. One parameter in the design of self-stabilizing algorithms is the number of states per machine. In [8], three different self-stabilizing algorithms with two states per machine are investigated. Only one of those algorithms works in a unidirectional ring, the other algorithms need more connections. The ring algorithm is probabilistic, but it is not symmetric: it requires an "exceptional machine" which executes different code. Herman's algorithm is mentioned in [8] as another two-state algorithm, but it is criticized by saying "it requires that all machines make moves synchronously which is not easily done". In this paper, we suggest and analyze an asynchronous variant of Herman's algorithm, which is symmetric and has only two states per machine.

The protocol of [12], also described in [2], is similar to Herman's protocol in that tokens are passed on a ring of processors. A scheduler selects a processor among those with a token; the selected processor passes the token to left or right neighbor, with probability 0.5, respectively. Two colliding tokens are *merged* to a single token. Our analysis of the asynchronous version of Herman's protocol could possibly be adapted to this protocol, by assuming that a processor passes its token after an exponentially distributed holding time. Of course, the fact that meeting tokens are merged and not annihilated would have to be taken into account.

2 Preliminaries

We assume N processors, with N odd, organized in a ring topology. Each processor may or may not have a token. Herman's protocol in the traditional *synchronous variant* [11] works as follows: in each time step, each processor that has a token passes its token to its clockwise neighbor with probability r (where $0 < r < 1$ is a fixed parameter), and keeps it with probability $1-r$; if a processor keeps its token and receives another token from its counterclockwise neighbor, then both of those tokens are annihilated. Notice that the number of tokens never increases, and can decrease only by even numbers.

Herman's protocol can be implemented as follows. Each processor possesses a bit, which the processor can read and write. Each processor can also read the bit of its counterclockwise neighbor. In this representation having the same bit as one's counterclockwise neighbor means having a token. In each time step, each processor compares its bit with the bit of its counterclockwise neighbor; if the bits are different, the processor keeps its bit; if the bits are equal, the processor flips its bit with probability r and keeps it with probability $1 - r$. It is straightforward to verify that this procedure implements Herman's protocol: in particular a processor flipping its bit corresponds to passing its token to its clockwise neighbor.[4]

We denote the number of initial tokens by M, where $1 \leq M \leq N$. The token representation described above enforces that M be odd. A configuration with only one token is called *legitimate*. The protocol can be viewed as a Markov chain with a single bottom SCC in which all states are legitimate configurations. So a legitimate configuration is reached with probability 1, regardless of the initial configuration, that is, the system *self-stabilizes* with probability 1.

In this paper we also propose and analyze an *asynchronous variant* of Herman's protocol which works similarly to the synchronous version. The asynchronous variant gives rise to a continuous-time Markov chain. Each processor with a token passes the token to its clockwise neighbor with rate λ, i.e., a processor keeps its token for a time that is distributed exponentially with parameter λ, before passing the token to its clockwise neighbor (i.e., flipping its bit). The advantage of this variant is that it does not require processor synchronization. Note that a processor can approximate an exponential distribution by a geometric distribution, that is, it can execute a loop which it leaves with a small fixed probability at each iteration. A more precise approximation can be obtained using a random number generator and precise clocks. For our performance analyses we assume an exact exponential distribution.

Let $\mathbf{T}$ denote the time until only one token is left, i.e., until self-stabilization has occurred. In this paper we analyze the random variable $\mathbf{T}$, focusing mainly

[4] Notice that flipping all bits in a given configuration keeps all tokens in place. In fact, in the original formulation [11], in each iteration each bit is effectively flipped once more, so that flipping the bit means keeping the token, and keeping the bit means passing the token. The two formulations are equivalent in the synchronous version, but our formulation allows for an asynchronous version.

on its expectation $\mathbb{E}\mathbf{T}$. Many of our results hold for both the synchronous and the asynchronous protocol version.

To aid our analysis we think of the processors as numbered from 1 to N, clockwise, according to their position in the ring. We write $m := (M-1)/2$. Let $z : \{1, \ldots, M\} \to \{1, \ldots, N\}$ be such that $z(1) < \cdots < z(M)$ and for all $i \in \{1, \ldots, M\}$, the processor $z(i)$ initially has a token; in other words, $z(i)$ is the position of the i-th token. We often write z_{uv} for $z(v) - z(u)$.

3 Bounds on $\mathbb{E}\mathbf{T}$ for Arbitrary Configurations

The following proposition gives a precise formula for $\mathbb{E}\mathbf{T}$ in both the synchronous and asynchronous protocols in case the number of tokens is $M = 3$.

Proposition 1 (cf. [16]). *Let N denote the number of processors and let a, b, c denote the distances between neighboring tokens, so that $a + b + c = N$. For the synchronous protocol with parameter r let $D = r(1-r)$, and for the asynchronous protocol with parameter λ let $D = \lambda$. Then the expected time to stabilization is*

$$\mathbb{E}\mathbf{T} = \frac{abc}{DN}\,.$$

Proposition 1 is shown in [16] for the synchronous case with $r = \frac{1}{2}$. Essentially the same proof works for $0 < r < 1$, and also in the asynchronous case.

We call a configuration with $M = 3$ equally spaced tokens an *equilateral configuration*. If N is an odd multiple of 3 then $a = b = c = N/3$ for the equilateral configuration. If N is not a multiple of 3 then we ask that a, b, c equal either $\lfloor N/3 \rfloor$ or $\lceil N/3 \rceil$. By Proposition 1 the expected stabilization time for a equilateral configuration is $\mathbb{E}\mathbf{T} = \frac{N^2}{27D}$. It follows that for configurations with $M = 3$ the worst case is $\mathbb{E}\mathbf{T} = \Omega(N^2)$ and this case arises for the equilateral configuration. In fact it has been conjectured in [16] that, for all N, the equilateral configuration is the worst case, not only among the configurations with $M = 3$, but among all configurations. This conjecture is supported by experiments carried out using the probabilistic model checker PRISM—see [2].

Finding upper bounds on $\mathbb{E}\mathbf{T}$ in the synchronous case goes back to Herman's original work [11]. He does not analyze $\mathbb{E}\mathbf{T}$ in the journal version, but in his technical report [11], where he proves $\mathbb{E}\mathbf{T} \leq N^2\lceil \log N \rceil/2$. He also mentions an improvement to $O(N^2)$ due to Dolev, Israeli, and Moran, without giving a proof or a further reference. In 2005, three papers [9, 16, 17] were published, largely independently, all of them giving improved $O(N^2)$ bounds. In [9] path-coupling methods are applied to self-stabilizing protocols, which lead in the case of Herman's protocol to the bound $\mathbb{E}\mathbf{T} \leq 2N^2$ for the case $r = \frac{1}{2}$. Independently, the authors of [16] claimed $O(N^2)$. Their proof is elementary and also shows $\mathbb{E}\mathbf{T} \leq 2N^2$ for the case $r = \frac{1}{2}$. Finally, the author of [17] (being aware of the conference version of [9]) applied the theory of coalescing random walks to Herman's protocol to obtain $\mathbb{E}\mathbf{T} \leq \left(\frac{\pi^2}{8} - 1\right) \cdot \frac{N^2}{r(1-r)}$, which is about $0.93N^2$ for

the case $r = \frac{1}{2}$. By combining results from [17] and [16], we further improve the constant in this bound (by 8/27, which is about 32%), and at the same time generalize it to the asynchronous protocol.

Theorem 2. *For the synchronous protocol with parameter r let $D = r(1-r)$, and for the asynchronous protocol with parameter λ let $D = \lambda$. Then, for all N and for all initial configurations, we have*

$$\mathbb{E}\mathbf{T} \leq \left(\frac{\pi^2}{8} - \frac{29}{27}\right) \cdot \frac{N^2}{D}\,.$$

Hence, $\mathbb{E}\mathbf{T} \leq 0.64N^2$ in the synchronous case with $r = \frac{1}{2}$.

4 Expressions for $\mathbb{E}\mathbf{T}$

Our analysis of Herman's protocol exploits the work of Balding [1] on annihilating particle systems. Such systems are a special case of *interacting particle systems*, which model finitely or infinitely many particles, which, in the absence of interaction, would be modeled as independent Markov chains. Due to particle interaction, the evolution of a single particle is no longer Markovian. Interacting particle systems have applications in many fields, including statistical mechanics, neural networks, tumor growth and spread of infections, see [15]. Balding's paper [1] is motivated by a scenario from physical chemistry, where particles can be viewed as vanishing on contact, because once two particles have met, they react and are no longer available for reactions afterwards. We refer the reader to [10] and the references therein for more information on such chemical reaction systems.

We transfer results from [1] to Herman's protocol. The setup is slightly different because, unlike chemical particles, the tokens in Herman's protocol move only in one direction. This difference is inconsequential, as the state of a system can be captured using only relative token (or particle) distances. Care must be taken though, because Balding does not consider "synchronous" particle movement (this would make no sense in chemistry), but particles moving "asynchronously" or continuously in a Brownian motion.

Given two tokens u and v with $1 \leq u < v \leq M$, we define a random variable $\mathbf{T}_{uv}$ and events $A_{(uv)\downarrow}$ and $A_{(uv)\uparrow}$ in terms of a system in which collisions between tokens u and v cause u and v to be annihilated, but the movement of the other tokens and their possible collisions are ignored. In that system, $\mathbf{T}_{uv}$ denotes the time until u and v have collided. Further, let $A_{(uv)\downarrow}$ and $A_{(uv)\uparrow}$ denote the events that tokens u and v eventually collide *down* and *up*, respectively. By colliding down (resp. up) we mean that, upon colliding, the token u (resp. v) has caught up with v (resp. u) in clockwise direction; more formally, if $d_u, d_v \geq 0$ denote the distances travelled in clockwise direction by the tokens until collision, then the collision is said to be down (resp. up) if $z(u) + d_u = z(v) + d(v)$ (resp. $z(u) + d_u + N = z(v) + d_v$). The behavior of two such tokens is equivalent to

that of a one-dimensional random walk on $\{0,\ldots,N\}$, started at z_{uv}, with absorbing barriers at 0 and N: the position in the random walk corresponds to the distance between the tokens, and colliding down (resp. up) corresponds to being absorbed at 0 (resp. N). By this equivalence we have $\mathcal{P}\left(A_{(uv)\downarrow}\right) = 1 - z_{uv}/N$ and $\mathcal{P}\left(A_{(uv)\uparrow}\right) = z_{uv}/N$ (see, e.g., [7]).

Proposition 3 below allows to express the distribution of $\mathbf{T}$ in terms of the distribution of $\mathbf{T}_{uv}$, conditioned under $A_{(uv)\downarrow}$ and $A_{(uv)\uparrow}$, respectively. Those distributions are well-known [7, 3]. For the statement we need to define the set W_M of all *pairings*. A pairing is a set $w = \{(u_1, v_1), \ldots, (u_m, v_m)\}$ with $1 \leq u_i < v_i \leq M$ for all i, such that there is $w_0 \in \{1,\ldots,M\}$ with $\{u_1, v_1, \ldots, u_m, v_m, w_0\} = \{1,\ldots,M\}$. Define $s(w) = 1$ if the permutation $(u_1 v_1 \cdots u_m v_m w_0)$ is even, and $s(w) = -1$ otherwise. (This is well-defined: it is easy to see that $s(w)$ does not depend on the order of the (u_i, v_i).) We have the following proposition:

Proposition 3 (cf. [1, Theorem 2.1]). *Let $M \geq 3$. For all $t \geq 0$:*

$$\mathcal{P}\left(\mathbf{T} \leq t\right) = \sum_{w \in W_M} s(w) \prod_{(u,v)\in w} \left(\mathcal{P}\left(\mathbf{T}_{uv} \leq t \ \cap \ A_{(uv)\downarrow}\right) - \mathcal{P}\left(\mathbf{T}_{uv} \leq t \ \cap \ A_{(uv)\uparrow}\right)\right) .$$

Balding's Theorem 2.1 in [1] is more general in that it gives a generating function for the number of remaining tokens at time t. Strictly speaking, Balding's theorem is not applicable to the synchronous version of Herman's protocol, because he only considers tokens that move according to the asynchronous version (in our terms), and tokens in a Brownian motion. In addition, his proof omits many details, so we give a self-contained proof for Proposition 3 in [13].

Theorem 4 below yields an expression for $\mathbb{E}\mathbf{T}$. We define the set $\overrightarrow{W_M}$ of all *directed pairings* as the set of all sets $\overrightarrow{w} = \{(u_1, v_1, d_1), \ldots, (u_m, v_m, d_m)\}$ such that $\{(u_1, v_1), \ldots, (u_m, v_m)\} \in W_M$ and $d_i \in \{\downarrow, \uparrow\}$ for all $i \in \{1,\ldots,m\}$. For a directed pairing $\overrightarrow{w} = \{(u_1, v_1, d_1), \ldots, (u_m, v_m, d_m)\}$ we define

$$\overrightarrow{s}(\overrightarrow{w}) := s(\{(u_1, v_1), \ldots, (u_m, v_m)\}) \cdot (-1)^{|\{i | 1 \leq i \leq m, \ d_i = \uparrow\}|}$$

and the event $A_{\overrightarrow{w}} := \bigcap_{i=1}^{m} A_{(u_i v_i) d_i}$. Notice that $\mathcal{P}\left(A_{\overrightarrow{w}}\right) = \prod_{i=1}^{m} \mathcal{P}\left(A_{(u_i v_i) d_i}\right)$. Further, we set $\mathbf{T}_{\overrightarrow{w}} := \max\{\mathbf{T}_{u_i v_i} \mid 1 \leq i \leq m\}$. We have the following theorem:

Theorem 4. *For $M \geq 3$:*

$$\mathbb{E}\mathbf{T} = \sum_{\overrightarrow{w} \in \overrightarrow{W_M}} \overrightarrow{s}(\overrightarrow{w}) \cdot \mathbb{E}\left[\mathbf{T}_{\overrightarrow{w}} \mid A_{\overrightarrow{w}}\right] \cdot \mathcal{P}\left(A_{\overrightarrow{w}}\right) .$$

A Finite Expression for $\mathbb{E}\mathbf{T}$. In the rest of the section we focus on the synchronous protocol. We obtain a closed formula for $\mathbb{E}\mathbf{T}$ in Proposition 5 below.

For $1 \leq u < v < M$, we define $z_{uv\downarrow} := z_{uv}$ and $z_{uv\uparrow} := N - z_{uv}$. For sets $\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w} \in \overrightarrow{W_M}$ with $\overrightarrow{x} = \{(u_1, v_1, d_1), \ldots, (u_k, v_k, d_k)\}$ and $\overrightarrow{w} =$

$\{(u_1, v_1, d_1), \ldots, (u_m, v_m, d_m)\}$ we write

$$y_F(\overrightarrow{x}, \overrightarrow{w}) := \left(\frac{z_{u_1 v_1 d_1}}{N}, \ldots, \frac{z_{u_k v_k d_k}}{N}\right) \quad \text{and}$$
$$y_G(\overrightarrow{x}, \overrightarrow{w}) := \left(\frac{z_{u_{k+1} v_{k+1} d_{k+1}}}{N}, \ldots, \frac{z_{u_m v_m d_m}}{N}\right) .$$

Let

$$g(j, y; u) := \frac{\sin(j\pi y) \cdot \sin(j\pi u)}{1 - \cos(j\pi u)} \qquad \text{and} \qquad h(j; u) := 1 - 2r(1-r)\left(1 - \cos(j\pi u)\right) ,$$

and define, for $k \in \mathbb{N}_+$ and $\ell \in \mathbb{N}_+$,

$$F_k^{(N)}(y_1, \ldots, y_k) := -\left(\frac{-1}{N}\right)^k \cdot \sum_{j \in \{1, \ldots, N-1\}^k} \frac{\prod_{i=1}^k g(j(i), y_i; 1/N)}{1 - \prod_{i=1}^k h(j(i); 1/N)} \qquad \text{and}$$

$$G_\ell(y_1, \ldots, y_\ell) := \prod_{i=1}^{\ell} (1 - y_i) .$$

We drop the subscripts of $F_k^{(N)}$ and G_ℓ, if they are understood. Observe that $F^{(N)}$ and G are continuous and do not depend on the order of their arguments. The following proposition gives, for the synchronous protocol, a concrete expression for $\mathbb{E}\mathbf{T}$.

Proposition 5. *Consider the synchronous protocol. For $M \geq 3$:*

$$\mathbb{E}\mathbf{T} = \sum_{\overrightarrow{w} \in \overrightarrow{W_M}} \overrightarrow{s}(\overrightarrow{w}) \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} F^{(N)}(y_F(\overrightarrow{x}, \overrightarrow{w})) \cdot G(y_G(\overrightarrow{x}, \overrightarrow{w})) .$$

An Approximation for $\mathbb{E}\mathbf{T}$. The function $F^{(N)}$ in Proposition 5 depends on N, and also on r. This prohibits a deeper analysis as needed in Section 6. Proposition 6 gives an approximation of $\mathbb{E}\mathbf{T}$ without those dependencies. To state it, we define, for $k \in \mathbb{N}_+$, a function $\widetilde{F}_k : [0,1]^k \to \mathbb{R}$ with

$$\widetilde{F}_k(y_1, \ldots, y_k) = \frac{-1}{\pi^2}\left(\frac{-2}{\pi}\right)^k \sum_{j \in \mathbb{N}_+^k} \frac{\prod_{i=1}^k \sin(y_i j(i)\pi)}{\left(\prod_{i=1}^k j(i)\right)\left(\sum_{i=1}^k j(i)^2\right)} .$$

We drop the subscript of $\widetilde{F}_k$, if it is understood. It is shown in [13] that the series in $\widetilde{F}_k$ converges. We have the following proposition.

Proposition 6. *Consider the synchronous protocol. Let*

$$\widetilde{E} := \frac{N^2}{r(1-r)} \sum_{\overrightarrow{w} \in \overrightarrow{W_M}} \overrightarrow{s}(\overrightarrow{w}) \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} \widetilde{F}(y_F(\overrightarrow{x}, \overrightarrow{w})) \cdot G(y_G(\overrightarrow{x}, \overrightarrow{w})) .$$

Then, for each fixed $M \geq 3$ and $r \in \left(\frac{1}{2} - \frac{\sqrt[4]{27}}{6}, \frac{1}{2} + \frac{\sqrt[4]{27}}{6}\right) \approx (0.12, 0.88)$ and $\varepsilon > 0$,

$$\mathbb{E}\mathbf{T} = \widetilde{E} + O(N^\varepsilon) .$$

The proof of Proposition 6 is elementary but involved.

5 The Full Configuration

In this section we consider the initial configuration in which every processor has a token, i.e., $N = M$. We call this configuration *full*. Notice that in the full configuration, with all bits set to 0, in the successor configuration each bit is independently set to 1 with probability r. Thus we study the full configuration in lieu of the random configuration. We have the following theorem:

Theorem 7. *For the synchronous protocol with parameter r let $D = r(1-r)$. For the asynchronous protocol with parameter $\lambda > 0$ let $D = \lambda$. For almost all odd $N \in \mathbb{N}_+$, we have for the full configuration:*

$$\mathbb{E}\mathbf{T} \leq 0.0285N^2/D \quad \textit{and} \quad \mathcal{P}\left(\mathbf{T} \geq 0.02N^2/D\right) < 0.5\,.$$

Recall from Proposition 1 that, for N an odd multiple of 3, we have $\mathbb{E}\mathbf{T} = \frac{1}{27}\frac{N^2}{D} \approx 0.0370\frac{N^2}{D}$ if we start from the equilateral configuration. It follows that, for large N, the full configuration (with $M = N$) stabilizes faster than the equilateral configuration (with $M = 3$). This is consistent with the aforementioned conjecture of McIver and Morgan that the equilateral configuration with $M = 3$ is the worst case among all configurations for a fixed N.

6 Restabilization

In this section we restrict attention to the synchronous version of Herman's algorithm and consider the standard bit-array implementation. Theorem 2 shows that the worst-case expected time to termination, considering all initial configurations, is $\mathbb{E}\mathbf{T} = O(N^2)$. We imagine that an initial configuration represents the state of the system immediately after an error, that is, the ring of tokens has become illegitimate because some of positions in the bit array were corrupted. In this light a natural restriction on initial configurations is to consider those that arise from a one-token configuration by corrupting some fixed number m of bits. We call these *flip-m* configurations. Notice that, by the token representation in Herman's protocol, a single bit error can lead to the creation of two neighboring tokens. So, m bit errors could lead to the creation of m new pairs of neighboring tokens. It could also happen that two bit errors affect neighboring bits, leading to a new pair of tokens at distance 2. To account for this, we characterize flip-m configuration as those with at most $2m + 1$ tokens such that the tokens can be arranged into pairs, each pair at distance at most m, with one token left over.

Fixing the number of bit errors we show that the expected time to restabilization improves to $O(N)$. Formally we show:

Theorem 8. *Consider the synchronous protocol. Fix any $m \in \mathbb{N}_+$ and $r \in \left(\frac{1}{2} - \frac{\sqrt[4]{27}}{6}, \frac{1}{2} + \frac{\sqrt[4]{27}}{6}\right) \approx (0.12, 0.88)$. Then for any flip-m configuration we have $\mathbb{E}\mathbf{T} = O(N)$.*

Proof. It suffices to consider flip-m configurations with $M = 2m + 1$ tokens. Without loss of generality, we assume that, when removing token $2m + 1$, the token pairs $(1, 2), (3, 4), \ldots, (2m - 1, 2m)$ have distances at most m; i.e., we assume $z(u + 1) - z(u) \leq m$ for all odd u between 1 and $2m - 1$.

For each directed pairing $\overrightarrow{w} \in \overrightarrow{W_M}$, we define its *class* $Cl(\overrightarrow{w})$ and its *companion pairing* $\overrightarrow{w}' \in \overrightarrow{W_M}$. For the following definition, we define $\widetilde{u} := u + 1$, if u is odd, and $\widetilde{u} := u - 1$, if u is even.

- If $(u, M, d) \in \overrightarrow{w}$ for some u, then $Cl(\overrightarrow{w}) = 0$. Its companion pairing is obtained, roughly speaking, by u and $\widetilde{u}$ switching partners. More precisely:
  - If $(\widetilde{u}, v, d')$ (resp. $(v, \widetilde{u}, d')$) for some (v, d'), then the companion pairing of w is obtained by replacing (u, M, d) and $(\widetilde{u}, v, d')$ with $(\widetilde{u}, M, d)$ and (u, v, d') (resp. (v, u, d')).
  - Otherwise (i.e., $\widetilde{u}$ does not have a partner), the companion pairing of w is obtained by replacing (u, M, d) with $(\widetilde{u}, M, d)$.
- If $\overrightarrow{w} = \{(1, 2, d_1), (3, 4, d_2), \ldots, (M - 2, M - 1, d_m)\}$ for some $d_1, \ldots, d_m$, then $Cl(\overrightarrow{w}) = m$. In this case, $\overrightarrow{w}$ does not have a companion pairing.
- Otherwise, $Cl(\overrightarrow{w})$ is the greatest number i such that for all $1 \leq j \leq i - 1$, the tokens $2j - 1$ and $2j$ are partners (i.e., $(2j - 1, 2j, d)$ for some d). Notice that $0 < Cl(\overrightarrow{w}) < m$. The companion pairing of $\overrightarrow{w}$ is obtained by $2i - 1$ and $2i$ switching partners.

It is easy to see that, for any $\overrightarrow{w} \in \overrightarrow{W_M}$ with $Cl(\overrightarrow{w}) < m$, we have $Cl(\overrightarrow{w}) = Cl(\overrightarrow{w}')$, and the companion pairing of $\overrightarrow{w}'$ is $\overrightarrow{w}$, and $\overrightarrow{s}(\overrightarrow{w}) = -\overrightarrow{s}(\overrightarrow{w}')$. Partition $\overrightarrow{W_M}$ into the following sets:

$$\overrightarrow{W_M}^{(+)} := \{\overrightarrow{w} \in \overrightarrow{W_M} \mid Cl(\overrightarrow{w}) < m \text{ and } \overrightarrow{s}(\overrightarrow{w}) = +1\} \qquad \text{and}$$
$$\overrightarrow{W_M}^{(-)} := \{\overrightarrow{w} \in \overrightarrow{W_M} \mid Cl(\overrightarrow{w}) < m \text{ and } \overrightarrow{s}(\overrightarrow{w}) = -1\} \qquad \text{and}$$
$$\overrightarrow{W_M}^{(m)} := \{\overrightarrow{w} \in \overrightarrow{W_M} \mid Cl(\overrightarrow{w}) = m\}.$$

The idea of this proof is that, in the sum of Proposition 6, the terms from $\overrightarrow{W_M}^{(+)} \cup \overrightarrow{W_M}^{(-)}$ cancel each other "almost" out, and the terms from $\overrightarrow{W_M}^{(m)}$ are small. To simplify the notation in the rest of the proof, let $y(\overrightarrow{x}, \overrightarrow{w}) := (y_F(\overrightarrow{x}, \overrightarrow{w}), y_G(\overrightarrow{x}, \overrightarrow{w}))$ and $H(y(\overrightarrow{x}, \overrightarrow{w})) := \widetilde{F}(y_F(\overrightarrow{x}, \overrightarrow{w})) \cdot G(y_G(\overrightarrow{x}, \overrightarrow{w}))$. Since $\widetilde{F}$ and G are continuous and bounded, so is H.

- Let $(\overrightarrow{x}, \overrightarrow{w})$ with $\overrightarrow{x} \subseteq \overrightarrow{w} \in \overrightarrow{W_M}^{(+)} \cup \overrightarrow{W_M}^{(-)}$. To any such $(\overrightarrow{x}, \overrightarrow{w})$ we associate a companion $(\overrightarrow{x}', \overrightarrow{w}')$ such that $\overrightarrow{w}'$ is the companion pairing of $\overrightarrow{w}$, and $\overrightarrow{x}' \subseteq \overrightarrow{w}'$ is obtained from $\overrightarrow{x}$ in the following way: if $\overrightarrow{w}'$ is obtained from $\overrightarrow{w}$ by replacing one or two triples (u, v, d), then $\overrightarrow{x}'$ is obtained by performing the same replacements on $\overrightarrow{x}$ (of course, only if $(u, v, d) \in \overrightarrow{x}$). Note that $y(\overrightarrow{x}, \overrightarrow{w})$ and $y(\overrightarrow{x}', \overrightarrow{w}')$ are equal in all components, except for one or two components, where they differ by at most $\frac{m}{N}$. Hence we have (for constant m) that

$$y(\overrightarrow{x}', \overrightarrow{w}') = y(\overrightarrow{x}, \overrightarrow{w}) + O(1/N) \cdot (1, \ldots, 1).$$

Since H is continuous, it follows

$$H(y(\overrightarrow{x}', \overrightarrow{w}')) = H(y(\overrightarrow{x}, \overrightarrow{w})) + O(1/N)\,.$$

– Let $(\overrightarrow{x}, \overrightarrow{w})$ with $\overrightarrow{x} \subseteq \overrightarrow{w} \in \overrightarrow{W_M}^{(m)}$. Note that all components of $y_F(\overrightarrow{x}, \overrightarrow{w})$ are at most $\frac{m}{N}$ or at least $1 - \frac{m}{N}$. Also note that for any vector $e \in \{0,1\}^{|\overrightarrow{x}|}$ it holds $H(e, y_G(\overrightarrow{x}, \overrightarrow{w})) = 0$. Since H is continuous, it follows

$$H(y(\overrightarrow{x}, \overrightarrow{w})) = O(1/N)\,.$$

Take $0 < \varepsilon < 1$. By Proposition 6 and the above considerations, we have:

$$\begin{aligned}
\mathbb{E}\mathbf{T} &= O(N^\varepsilon) + \frac{N^2}{r(1-r)} \sum_{\overrightarrow{w} \in \overrightarrow{W_M}} \overrightarrow{s}(\overrightarrow{w}) \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} H(y(\overrightarrow{x}, \overrightarrow{w})) \\
&= O(N^\varepsilon) + \frac{N^2}{r(1-r)} \cdot \left(\sum_{\overrightarrow{w} \in \overrightarrow{W_M}^{(+)}} \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} H(y(\overrightarrow{x}, \overrightarrow{w})) \right. \\
&\qquad\qquad - \sum_{\emptyset \neq \overrightarrow{x}' \subseteq \overrightarrow{w}'} H(y(\overrightarrow{x}', \overrightarrow{w}')) \\
&\qquad\qquad \left. + \sum_{\overrightarrow{w} \in \overrightarrow{W_M}^{(m)}} \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} H(y(\overrightarrow{x}, \overrightarrow{w})) \right) \\
&= O(N^\varepsilon) + \frac{N^2}{r(1-r)} \cdot \left(\sum_{\overrightarrow{w} \in \overrightarrow{W_M}^{(+)}} \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} O(1/N) \right. \\
&\qquad\qquad \left. + \sum_{\overrightarrow{w} \in \overrightarrow{W_M}^{(m)}} \sum_{\emptyset \neq \overrightarrow{x} \subseteq \overrightarrow{w}} O(1/N) \right) \\
&= O(N^\varepsilon) + O(N) = O(N)\,.
\end{aligned}$$

□

7 Conclusions and Future Work

We have obtained several results on the expected self-stabilization time $\mathbb{E}\mathbf{T}$ in Herman's algorithm. We have improved the best-known upper bound for arbitrary configurations, and we have given new and significantly better bounds for special classes of configurations: the full configuration, the random configuration, and, in particular, for configurations that arise from a fixed number of bit errors. For the latter class, $\mathbb{E}\mathbf{T}$ reduces to $O(N)$, pointing to a previously unknown feature that Herman's algorithm recovers quickly from bounded errors. We have also shown that an asynchronous version of Herman's algorithm not requiring synchronization behaves similarly. For our analysis, we have transferred techniques that were designed for the analysis of chemical reactions.

The conjecture of [16], saying that the equilateral configuration with three tokens constitutes the worst-case, remains open. We hope to exploit our closed-form expression for $\mathbb{E}\mathbf{T}$ to resolve this intriguing problem. While we have already shown that many relevant initial configurations provably converge faster, solving this conjecture would close the gap between the lower and upper bounds for stabilization time for arbitrary configurations. We would also like to investigate the performance of the algorithm in case the number of bit errors is not fixed, but is small (e.g., logarithmic) in the number of processes.

References

1. D. Balding. Diffusion-reaction in one dimension. *J. Appl. Prob.*, 25:733–743, 1988.
2. PRISM case studies. Randomised self-stabilising algorithms. `http://www.prismmodelchecker.org/casestudies/self-stabilisation.php`.
3. D. Cox and H. Miller. *The theory of stochastic processes*. Chapman & Hall/CRC, 2001.
4. E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, 1974.
5. S. Dolev. *Self-Stabilization*. MIT Press, 2000.
6. R. Durrett and H. Kesten (eds). *Random Walks, Brownian Motion and Interacting Particle Systems*. Birkhauser Verlag AG, 1991.
7. W. Feller. *An introduction to probability theory and its applications*, volume 1. John Wiley & Sons, 1968.
8. M. Flatebo and A.K. Datta. Two-state self-stabilizing algorithms for token rings. *IEEE Trans. Softw. Eng.*, 20(6):500–504, 1994.
9. L. Fribourg, S. Messika, and C. Picaronny. Coupling and self-stabilization. *Distributed Computing*, 18:221–232, 2005.
10. S. Habib, K. Lindenberg, G. Lythe, and C. Molina-Paris. Diffusion-limited reaction in one dimension: Paired and unpaired nucleation. *Journal of Chemical Physics*, 115:73–89, 2001.
11. T. Herman. Probabilistic self-stabilization. *Information Processing Letters*, 35(2):63–67, 1990. Technical Report at `ftp://ftp.math.uiowa.edu/pub/selfstab/H90.html`.
12. A. Israeli and M. Jalfon. Token management schemes and random walks yield self-stabilizing mutual exclusion. In *Proceedings of PODC'90*, pages 119–131. ACM, 1990.
13. S. Kiefer, A. Murawski, J. Ouaknine, J. Worrell, and L. Zhang. On stabilization in Herman's algorithm. Technical report, arxiv.org, 2011. Available at `http://arxiv.org/abs/1104.3100`.
14. S. Kutten and B. Patt-Shamir. Stabilizing time-adaptive protocols. *Theor. Comput. Sci.*, 220(1):93–111, 1999.
15. T.M. Liggett. *Interacting particle systems*. Springer, 2005.
16. A. McIver and C. Morgan. An elementary proof that Herman's ring is $\theta(n^2)$. *Inf. Process. Lett.*, 94(2):79–84, 2005.
17. T. Nakata. On the expected time for Herman's probabilistic self-stabilizing algorithm. *Theoretical Computer Science*, 349(3):475–483, 2005.
18. M. Schneider. Self-stabilization. *ACM Comput. Surv.*, 25(1):45–67, 1993.