How to Color a French Flag*

Biologically Inspired Algorithms for Scale-Invariant Patterning

Bertie Ancona[1], Ayesha Bajwa[1], Nancy Lynch[1], and Frederik Mallmann-Trenn[2]

[1] Massachusetts Institute of Technology
{bancona,abajwa}@alum.mit.edu, lynch@csail.mit.edu
[2] King's College London frederik.mallmann-trenn@kcl.ac.uk

Abstract. In the *French flag problem*, initially uncolored cells on a grid must differentiate to become blue, white or red. The goal is for the cells to color the grid as a French flag, i.e., a three-colored triband, in a distributed manner. To solve a generalized version of the problem in a distributed computational setting, we consider two models: a biologically-inspired version that relies on morphogens (diffusing proteins acting as chemical signals) and a more abstract version based on reliable message passing between cellular agents.

Much of developmental biology research focuses on concentration-based approaches, since morphogen gradients are an underlying mechanism in tissue patterning. We show that both model types easily achieve a *French ribbon* - a French flag in the 1D case. However, extending the ribbon to the 2D flag in the concentration model is somewhat difficult unless each agent has additional positional information. Assuming that cells are identical, it is impossible to achieve a French flag or even a close approximation. In contrast, using a message-based approach in the 2D case only requires assuming that agents can be represented as logarithmic or constant size state machines.

We hope that our insights may lay some groundwork for what kind of message passing abstractions or guarantees, if any, may be useful in analogy to cells communicating at long and short distances to solve patterning problems. We also hope our models and findings may be of interest in the design of nano-robots.

Keywords: Distributed Computing · French Flag · Biologically Inspired Algorithms.

1 Introduction

In the *French flag problem*, initially uncolored cells on a grid must differentiate to become blue, white or red, ultimately coloring the grid as a three-colored triband without centralized decision-making. Lewis Wolpert's original French flag problem formulation [19, 20] has been applied and extended to understand how organisms determine cell fate, or final differentiated cell type, a question central to developmental biology. Wolpert's formulation of positional information models is both complementary

* The authors were supported in part by NSF Award Numbers CCF-1461559 and CCF-0939370.

to and contrasted with Turing's earlier formulation of reaction-diffusion instability [18], which relies on random asymmetries that arise from activator-inhibitor dynamics in a developmental system. Our methods make use of both positional information and initial asymmetry. However, we distinguish between absolute and relative positional information to probe whether full knowledge of the coordinates is needed to solve the problem, or if strictly less information suffices.

Broadly speaking, our work is inspired by the biological mechanisms leading to cell fate decisions in the original French flag problem. These long and short-distance mechanisms inform the design of algorithms and analyses of the problem in two distributed computing contexts. More precisely, we relate a reliable message passing model (Section 2.2) to local cell-cell communication, and a concentration-based model (Section 2.1) to morphogen gradients over long distances.

We analyze a generalized French flag problem for k colors in these two computational models. We aim to understand the resources and minimum set of assumptions required to solve the problem exactly or approximately. In particular, we study whether cells must know their exact positions and the grid dimensions in order to solve the k-flag problem. We hope that characterizing the resources and information required might have some translation back to the mechanisms enabling scale-invariant patterning.

We begin by studying the *French ribbon problem*, the 1D scenario in our models. Both exact and approximate solutions are possible, with a general tradeoff between precision and space complexity. While both models easily achieve a French ribbon, extending 1D decision-making to the 2D setting is provably difficult in the concentration model. We show that in a 2D grid with point sources at the corners, each agent knowing its absolute distance to every source is insufficient positional information to color the grid even approximately correctly. On the other hand, extending to the 2D setting is easy in the message passing model. We analyze numerous algorithms to demonstrate tradeoffs between time complexity, message size, memory size and precision of the obtained French flag.

We do not claim more accurate or thorough models than those proposed by the biology community. However, we hope this work may illuminate computational abstractions or guarantees that may be useful in analogy to cells communicating at long and short distances to solve patterning problems.

1.1 Biology Background and Related Work

A key principle of our models is that initial asymmetry and local communication eventually leads to long-distance transmission of the relative positional information of cellular agents, allowing for distributed decision-making. Morphogens, or molecules acting as chemical signals, underlie cell-cell communication over long distances. Two well-studied morphogens are *Bicoid (Bcd)* for anterior-posterior patterning in fruit flies [5,14], and *Sonic hedgehog (Shh)*, a morphogen for neural patterning in vertebrates, including humans [4, 15]. Exactly how these morphogens produce scale-invariant patterns in organisms and tissues of varying size is an interesting biological question [8].

Mechanisms for local cell-cell communication include cell surface receptors and ligands, such as the Notch-Delta system previously studied in a distributed computing context [1]. There are also physical channels for signalling molecules, such as gap junctions in animal cells and plasmodesmata in plant cells [2]. We liken local signalling to message passing between neighboring agents.

Building on earlier work on gradients [12, 16], Wolpert focused the French flag problem and model [19, 20] on the concept of positional information and its generalization to other patterning mechanisms. Subsequent papers validated the importance of positional information through empirical studies in model species [5, 14, 17]. Turing had previously studied reaction-diffusion instability as a driver of morphogenesis [18], theorizing that periodic patterns could spontaneously arise from activator-inhibitor dynamics. Turing's paradigm is often contrasted with Wolpert's notion of positional information. The idea that cells may learn positional information via concentration has fundamentally altered the field of developmental biology [7, 10]. The French flag problem has been studied using various models, including growth and repair simulation models [11] and reaction-diffusion experimental models [21].

1.2 Results

Here we summarize results in the two computational models. We first present our results for the concentration model, where we assume that each node on a line has access to just morphogens concentrations c_1 and c_2, each emitted from an endpoint of the line, and no other information. We define the model formally in Section 2.1.

On the positive side, it is possible to solve the French ribbon problem exactly.

Theorem 1. *Algorithm Exact Concentration Ribbon solves the concentration model k-ribbon for an n-agent line graph of arbitrary finite length a with constant time and communication complexity, given that agents have knowledge of morphogen concentrations c_1 and c_2, which have reached steady states, as well as the gradient function.*

On the negative side, we show that extending to the French flag (2D-case) with just four point-sources at the corners is infeasible. Here, symmetry prevents us from obtaining a ε-approximate algorithm in this model.

Theorem 2. *Consider the concentration model. Fix any $\varepsilon \in (0,1/6)$. No algorithm can produce an ε-approximate French flag.*

The concentration model contrasts the message passing model, in which even exact solutions are possible. Results for the message passing model are summarized in Table 1 below, and the exact statements can be found in section 4. Finally, we show in Section 4.1 how these algorithms can be extended to the 2D case.

Algorithm	Rounds	Agent Memory	Msgs	Msg Bits	Exact	Reference
Exact Count	$(2-1/k)n$	$3\log n+O(1)$	$O(n)$	$O(\log n)$	✓	*Thm. 3*
Exact Silent Count	$3n$	$2\log n+O(1)$	$O(n)$	$O(1)$	✓	*Thm. 4*
Bubble Sort	$3n$	$O(\log k)$	$O(n^2)$	$O(\log k)$	✓	*Thm. 7*
Approx Count	$2n$	$2\log\log n+O(1)$	$O(n)$	$O(\log\log n)$	×	*Thm. 6*

Table 1. Comparison of k-ribbon algorithms in the message passing model. For brevity we ignore additive $O(k)$ terms in the round complexity. The time complexity of *Exact Count* is tight up to an additive $2k$ term, regardless of k and the starting agent. The memory and message complexity of *Bubble Sort* are independent of n and in fact constant assuming $k=O(1)$.

2 Models and Notation

2.1 Concentration Model

For concentration-based solutions to the French flag problem, we assume that each agent receives concentration inputs from up to four source agents s_1, s_2, s_3, and s_4. The *measured concentration* a cell at 2D coordinate $C=(x,y)$ receives from source s_i, $i\in[4]$ is given by the following *gradient function*, which is assumed to be invertible and monotonically decreasing in $\text{dist}(C,s_i)$, the distance between cell C and the source s_i. For concreteness, consider the following power-law function

$$\lambda_i(C)=\frac{1}{\text{dist}(C,s_i)^\alpha} \tag{1}$$

where α is the power-law constant. This family of functions is also handy for the 1D case with coordinate $C=x$ and source s_i, $i\in[2]$ in section 3, where we argue that coloring correctly can be reduced to comparing $\lambda_1(C)/\lambda_2(C)$ to 2^α and $2^{-\alpha}$.

Though we choose a power-law for convenience, our upper bounds and lower bounds hold for more general gradient functions satisfying the above constraints. Deriving precise thresholds for $\lambda_1(C)$ and $\lambda_2(C)$ is more difficult when the thresholds fall close together or when the gradient function is complicated. The more difficult these conditions, the less biologically practical it may be.

We do not assume any noise, so agents have arbitrarily good precision in measuring concentration. Additionally, we assume that the cells do not receive any other input apart from measured concentration. In particular, they do not have any other positional information such as knowledge of their coordinate or the total ribbon or flag size. We assume all agents behave identically, performing the same algorithms. No messages are passed between agents, so we consider only local computation for time complexity, assuming morphogen concentrations have reached steady state.

For the French ribbon, we assume that the two sources s_1 and s_2 are positioned at the ends of the line. We have two sources rather than one because a single source only gives an agent information about the distance of that agent to the source, without giving information about the agent's distance to the other side of the line.

For the French flag we assume the $s_i \in [4]$ are positioned at the four corners. We make this assumption in order to understand if the concentration model is 'strong' enough to solve the French flag problem without any additional communication. Assuming that additional sources are placed at convenient positions such as $(a/3,0)$ for example, defies the idea of of scale invariant systems. The corner points are already distinguished in that they only have two neighbors, and if one were to place a constant number of sources, these positions are somewhat natural.

2.2 Message Passing Model

We first consider a 1D version of the French flag problem which we call the *French ribbon problem*. We assume a line graph consisting of n nodes which we refer to as agents. We later consider the 2D version, the standard *French flag problem*, where the graph is a $a \times b$ grid on $n = a \cdot b$ agents.

Our message-passing model is similar to the standard LOCAL distributed model, with a few exceptions. Though agents have no knowledge of their global position, they do have a common sense of direction $dir \in \{up, down, left, right\}$. Additionally, agents know which of their neighbors exist, meaning they know whether they are endpoints of rows or columns (or both, if they are corners). Initially, all but one arbitrary agent called the *starting agent* s, representing the source of the communication signal, are *asleep* and thus perform no computation. Sleeping agents wake upon receiving a message.

The goal is to design algorithms that solve the French ribbon problem. Eventually, each agent must output a color so that the line is segmented into three colors: blue, white, and red from left to right. Formally, if b, w, and r denote the number of agents of each respective color, $\max\{|b-w|,|b-r|,|w-r|\} \leq 1$. In addition, each color should be in a single, contiguous sub-line of the graph—blue, white, red from left to right. We also define the more general 1D k-Ribbon problem in the same model, in which there are k distinct colors $\{1, ..., k\}$ which must form bands of approximately equal size, in increasing numerical order, along a line graph of n agents.

The 2D model is similar to the static, oriented 1D line graph model, but the system consists of an r by c grid of agents, oriented with *up* and *down* as well as $left$ and *right*. A solution to the French flag problem requires that every agent outputs a single color, such that the grid is divided into three vertical blocks. Every row must abide by the requirements of the French ribbon problem, such that the left side is blue and the right side is red. Furthermore, an agent should be the same color as the agent above and below it in its column. The 2D k-Flag problem generalizes in the same manner as above.

2.3 Approximation Definition

Intuitively speaking, the definition of approximation ensures two properties. First, agents that are clearly within one stripe should have the corresponding color. Second, agents that are close to a color border (c_1,c_2) should have either color c_1 or c_2.

We say a k-colored flag of dimensions $a \times b$ is an ε-*approximate* (French) flag if for every color $z \in \{1,...,k\}$ the following hold. For each agent u with coordinates (x,y):

1. if $x \in \left[(\frac{z-1}{k}+\varepsilon)\cdot a,(\frac{z}{k}-\varepsilon)\cdot a\right]$, then the agent has color z.
2. if u has color z, then $x \in \left[(\frac{z-1}{k}-\varepsilon)\cdot a,(\frac{z}{k}+\varepsilon)\cdot a\right]$.

3 Concentration Model Results

3.1 1D Exact Concentration Ribbon

Algorithm *Exact Concentration Ribbon* We consider an n-agent line of arbitrary finite length a in the concentration model. Assume morphogens m_1 and m_2 (with concentrations c_1 and c_2) are each secreted by one of the endpoint agents. We assume the underlying gradient function for concentration given position x is the inverse power law in α, which is assumed to be noiseless.

Assume that m_1 is secreted at $x=0$ and m_2 is secreted at $x=a$, we have $c_1 = 1/x^\alpha$ and $c_2 = 1/(a-x)^\alpha$. The ratio of c_2 to c_1 is then $(a-x)^\alpha/x^\alpha$. Each agent computes this ratio independently from the measured values of c_1 and c_2. Let *ratio* $= c_2/c_1$. After calculating its measured ratio, each agent computes the smallest color z such that *ratio* $\geq ((z-1)/(k-z))^\alpha$, decides color z, and halts.

The algorithm is size-invariant and works for a line graph of arbitrary finite length.

3.2 2D Concentration Lower Bound

In this section we sketch a proof of Theorem 2, showing that the concentration model, without absolute positional information, cannot produce a correct French flag (or even a good approximation) regardless of the gradient function.

Given an arbitrary flag G of dimensions $a \times b$, we show that we can construct a flag G' with dimensions $a' \times b'$ such that there are two agents in both flags that 1) have exactly the same distances from the respective sources and 2) must choose different colors. Since the two agents have the same respective distance to every source, they receive the same concentration input and cannot distinguish between settings, making it impossible to always color correctly. See Figure 1 for an illustration. To show that such a flag G' exists, we frame the constraints as a system of equations and we show that there exists a valid solution.

4 Message Passing Model

Before we present our algorithms, note there is a trivial algorithm that works as follows for $k=3$. The starting agent sends a wakeup message to the leftmost and rightmost agents. Then start a counter from each of these agents. When an agent receives the counters n_ℓ and n_r, it can determine in which stripe it is by testing whether $n_\ell/n_r \geq 2$ or $n_\ell/n_r \leq 1/2$. This idea generalizes to arbitrary k.

The algorithms we present improve on the trivial algorithm in various ways. Table 1 summarizes the tradeoffs of our approaches in the message passing model.

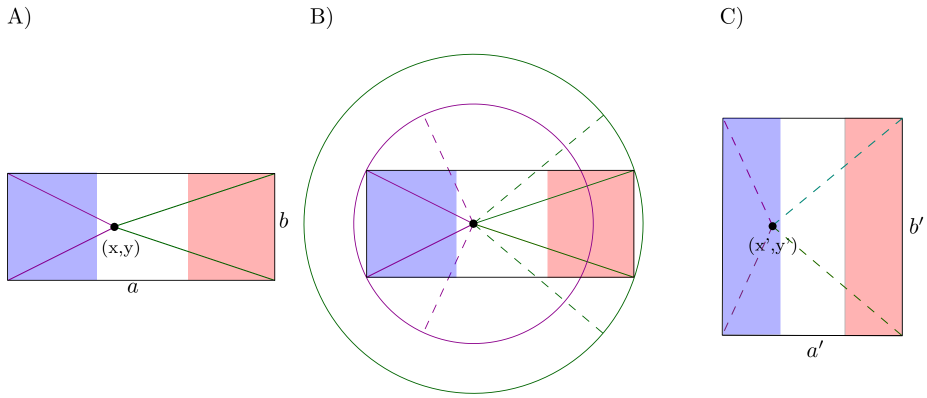

Fig. 1. A) depicts an arbitrary original flag. In the proof of Theorem 2 we argue how to construct a new flag as in C) such that there are two agents in both flags that have exactly the same distances from the respective sources and must also choose different colors. Since the two agents have the same respective distance to every source, they receive the same concentration input and cannot distinguish between the settings, making it impossible to always color correctly. We construct the new flag by changing the aspect ratio in a way that maintains the distances. B) depicts this transformation.

As a starting point, we observe that each agent can learn the number of agents to its left and right, from which information it can determine its own color [20]. This principle is central to some of our algorithms.

Note 1. An agent in the k-ribbon problem may determine its correct color knowing the number of agents on each side of it in line, and knowing which side should be color 1.

Algorithm *Exact Count* The starting agent stores the value $n_{mid} \leftarrow 0$ and sends $n_{mid}+1$ in both directions. Intuitively, the value measures the distance to the starting agent. All other agents upon waking store the received value as n_{mid} and forward the value $n_{mid}+1$ to the next agent in the same direction. Each agent also stores $t \leftarrow n_{mid}$ and increments t every round after.

When the left endpoint receives a value for n_{mid}, it decides on color 1 and sends $n_\ell = 1$ to its right neighbor. When the right endpoint receives a value for n_{mid}, it decides on color k and sends $n_r = 1$ to its left neighbor. Each agent stores n_d for either direction $d \in \{\ell, r\}$ which is the number of agents to the left (right, respectively). Upon receiving n_d, the agents forwards n_d+1 in the same direction.

After an agent receives both n_ℓ and n_r, it decides its color using Note 1. In order to get an improved time complexity, an agent may also decide early: if an agent has a value n_d and $t \geq 2((k-1)\cdot n_d) - n_{mid}$, it should decide color 1 if d is ℓ or color k otherwise.

Theorem 3. *Algorithm Exact Count solves the k-ribbon problem and requires at most $(2-\frac{1}{k})\cdot n+k$ rounds, $(4-\frac{2}{k})\cdot n \log n$ message bits, and $3\log n + \log k + O(1)$ bits of memory per agent.*

In reliable and synchronous models, it is well-known that silence conveys information. We improve the message bit complexity in Theorem 3 using the absence of a message as information, at a small cost to round complexity.

Algorithm *Exact Silent Count* The starting agent sends the message 0 to the left and 1 to the right. If it is an endpoint, the starting agent sends a 0 and a 1 in the same, 2-bit message to its neighbor. Agents will forward any received messages in the same direction, except endpoints which will send the messages back.

The agents do additional processing. The endpoint on the d side sets $n_d \leftarrow 0$ upon waking and never modifies it. Otherwise, the first time an agent receives a message from direction d, it sets $n_{\bar{d}} \leftarrow 0$, and each round thereafter the agent increments $n_{\bar{d}}$, until it receives a message from the $\bar{d}$ direction, at which point it stops incrementing $n_{\bar{d}}$ and sets $n_{\bar{d}} \leftarrow n_{\bar{d}}/2$. When an agent has final values for n_ℓ and n_r, and has sent 0 to the left and 1 to the right, it decides its color based on its stored values of n_ℓ and n_r using Note 1 and halts. [1]

Theorem 4. *Algorithm Exact Silent Count solves the k-Ribbon problem and requires $3n$ rounds, $6n$ message bits, and $2\log n + \log k + O(1)$ bits of memory per agent.*

Proof. We show correctness for the case when the starting agent is not an endpoint; we leave that end-case for the reader. W.l.o.g. consider an agent that first receives a 0 from the right. After $2n_\ell$ rounds, the 0 bit will return to the agent after having been forwarded to the left endpoint and back, so the stored value of n_ℓ at the end of the round will be correct. After $2n_r$ more rounds, the 0 bit is received again from the right and n_r is correctly set. Thus, as long as the agent receives the 0 bit 3 times, it will color itself correctly. The 0 bit must then travel from the starting agent to the left, back to the right endpoint, then back to the left endpoint; at that point, all agents to the left of the starting agent will correctly color themselves. As long as the agents to the right of the starting agent return the 0 bit leftward, this will occur. We thus have correctness, because all agents only halt after forwarding the opposite bit back to the other side. The same argument applies to the 1 bit in the other direction.

A bit travels at most 3 times down the line, so all agents terminate after $3n$ rounds. Each round has 2 bits sent, so the message bit complexity is $6n$. Each agent stores k and two values in $\Theta(n)$, requiring only $2\log n + \log k + O(1)$ bits of memory each. □

Next, we use the approximation approach of Morris [13] and Flajolet [6] to reduce space complexity in exchange for a slight increase in error for the final k-ribbon. The randomized modification is made to our deterministic exact counting algorithm.

The following theorem gives the guarantees of each counter.

Theorem 5 ([6]). *Let $\beta = 2^{2^{-\delta}}$. Consider the counter procedure of [6], in which we maintain a counter c over n increments, and increase the counter by one only with probability $(\frac{1}{\beta})^c$ at each increment. Using $\log\log n + \delta$ bits for the counter, the expected value of the counter is $\log_\beta((\beta-1)\cdot n + \beta)$, and the value of n we could recover from the counter has standard deviation at most $n/2^{-\delta}$.*

[1] We note that a similar algorithm may use a single token rather than binary messages, at an additional constant-factor increase in round complexity.

Algorithm *Approximate Count* The starting agent sends a bit in either direction to wake all agents. When the endpoint in the d direction wakes up, it sets a counter c_d to 0, increments it as in [6], and sends the resulting value to its neighbor. Each agent upon receiving a message from direction d, stores the message as c_d, increments it in the same way and forwards the result to the next agent.

When an agent has received two values of c_d, it does the following: For each i in the sequence $1,...,k$, if $c_\ell - c_r \leq \log_\beta \frac{i}{k-i}$, then the agent decides on color i. If the agent has not decided on a color yet after all i, the agent decides on color k. After deciding on a color, the agent halts.

Theorem 6. *Fix any k. For n large enough, Algorithm Approximate Count solves the ϵ-approximate k-Ribbon problem for constant $\epsilon < \frac{1}{2(k-1)}$ with probability $1-\frac{1}{32k}$ and requires $2n$ rounds, $O(n\log\log n)$ total message bits, and $2\log\log n + O(1)$ bits of memory per agent.*

We restrict $\epsilon < \frac{1}{2(k-1)}$ because otherwise the color thresholds would bleed into each other and we would have regions with more than two valid colors. The core idea of using an approximate counter as proposed in [6] is that when subtracting the counter from the left and from the right, we get for some β, ignoring small standard deviations,

$$\log_\beta((\beta-1)n_\ell+\beta) - \log_\beta((\beta-1)n_r+\beta) \approx \log(n_\ell/n_r).$$

Using thresholds for each color then gives the right color. Using monotonicity of the counters, we only need to consider $O(k)$ different counters which allows us to take a union bound over $O(k)$ of them, showing that all n counters are 'correct'. The proof can be found in the full version [3].

We next demonstrate how to use bubble sort to color the flag exactly. Assume blue, white and red are 1, 2, and 3 respectively.

Algorithm *Bubble Sort* The algorithm is an application of the parallel sorting algorithm of [9]. The idea is to naively color agents, in alternating fashion with the colors of the flag, to ensure correct total counts of each color regardless of the ribbon length. The algorithm then performs swaps in parallel to ensure that blue elements ripple to the left, white elements to the middle, and red elements to the right. In an even round, any agent at an even position swaps the value (color) with its right neighbor if the right neighbor has a larger value. Odd rounds are analogous.

In order to avoid cases in which a agent would like to swap its color with both neighbors at same time, we also ensure through message passing that each agent knows whether it is at an odd or even position and whether the current round is odd or even.

Theorem 7. *Algorithm Bubble Sort solves the 1-D k-Ribbon problem and requires at most $2n$ rounds, $n^2\log k$ message bits, and $O(\log k)$ bits of memory per agent.*

Proof. The algorithm of [9] requires at most n time-steps to sort an array using neighbor swaps in parallel. However, to assign each node a starting color, a message must be propagated from the leader to all nodes, requiring up to an additional n rounds.

Each round, up to half of the nodes send messages of size $\log k$ to broadcast their current values to one of their neighbors, for a total of at most $n^2 \log k$ message bits. Each node must store its own value using $\log k$ bits. □

4.1 Extending from Ribbon to Flag

We may solve the k-flag problem by extending any k-ribbon algorithm, with little loss in most parameters.

Algorithm *Up & Down* The starting agent begins the k-ribbon algorithm for its row, and all agents in the row follow the algorithm to completion once awakened. However, after deciding on a color but before halting, each agent in the row tells its color to above and below neighbors. When an agent is awoken with a color, it decides that color and forwards the color either above or below before halting.

Theorem 8. *Given an algorithm for the k-ribbon problem which takes $T(n,k)$ rounds, $M(n,k)$ total message bits, and $S(n,k)$ bits of memory per agent, Algorithm Up & Down solves the k-flag problem on a $a \times b$ grid with at most $a + T(b,k)$ rounds, $ab\log k + M(b,k)$ total message bits, and $S(b,k)$ bits of memory per agent.*

Other reductions to the k-ribbon problem that optimize for round complexity rather than space and message bit complexity are left to the reader.

4.2 Message-Passing Lower Bounds

There are straightforward lower bounds for the 1D and 2D cases.

Theorem 9. *No algorithm exists that can solve the k-Ribbon problem on an oriented line graph if all agents are identical, even if endpoints know that they are endpoints, in less than $(2-\frac{1}{k})\cdot n-3$ rounds.*

Theorem 10. *No algorithm exists to solve the k-flag problem on an $a \times b$ grid in less than $\max\{(2-\frac{1}{k})\cdot b-k, a+b-2\}$ rounds.*

Conclusion

The 1D French ribbon problem can be solved exactly and approximately in both the concentration and the message passing models. However, the 2D French flag problem requires additional positional information in order to satisfy size invariance.

One direct extension of this work is a randomized version of the *Silent Count* algorithm (Theorem 4). An exciting new research direction is how other pattering problems can be solved in more general settings and under the influence of noise. Future work could develop models that better capture important biological constraints. For example, one could study models in which part of an organism (e.g., a finger or the beak of a bird) grows over time.

Acknowledgements

We thank Ama Koranteng, Adam Sealfon, and Vipul Vachharajani for valuable discussions and contributions.

References

1. Afek, Y., Alon, N., Barad, O., Hornstein, E., Barkai, N., Bar-Joseph, Z.: A biological solution to a fundamental distributed computing problem. Science **331**(6014), 183–185 (2011)
2. Alberts, B.: Molecular Biology of the Cell. CRC Press (2017)
3. Ancona, A., Bajwa, A., Lynch, N., Mallmann-Trenn, F.: How to color a french flag–biologically inspired algorithms for scale-invariant patterning (2019)
4. Dessaud, E., McMahon, A.P., Briscoe, J.: Pattern formation in the vertebrate neural tube: a sonic hedgehog morphogen-regulated transcriptional network. Development **135**(15), 2489–2503 (2008)
5. Driever, W., Nüsslein-Volhard, C.: A gradient of bicoid protein in drosophila embryos. Cell **54**(1), 83–93 (1988)
6. Flajolet, P.: Approximate counting: A detailed analysis. BIT **25**(1), 113–134 (1985)
7. Green, J.B.A., Sharpe, J.: Positional information and reaction-diffusion: two big ideas in developmental biology combine. Development **142**(7), 1203–1211 (2015)
8. Gregor, T., Bialek, W., van Steveninck, R.R.d.R., Tank, D.W., Wieschaus, E.F.: Diffusion and scaling during early embryonic pattern formation. Proceedings of the National Academy of Sciences **102**(51), 18403–18407 (2005)
9. Habermann, N.: Parallel neighbor-sort (or the glory of the induction principle). Paper 2087 (1972)
10. Jaeger, J., Martinez-Arias, A.: Getting the measure of positional information. PLoS Biology **7**(3), e1000081 (2009)
11. Miller, J.F.: Evolving a self-repairing, self-regulating, french flag organism. In: Genetic and Evolutionary Computation – GECCO 2004. pp. 129–139. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
12. Morgan, T.H.: "polarity" considered as a phenomenon of gradation of materials. Journal of Experimental Zoology p. 2:495–506 (1905)
13. Morris, R.: Counting large numbers of events in small registers. Commun. ACM **21**(10), 840–842 (1978)
14. Nüsslein-Volhard, C., Wieschaus, E.: Mutations affecting segment number and polarity in drosophila. Nature **287**(5785), 795–801 (1980)
15. Patten, I., Placzek, M.: The role of sonic hedgehog in neural tube patterning. Cellular and Molecular Life Sciences CMLS **57**(12), 1695–1708 (2000)
16. Stumpf, H.F.: Mechanism by which cells estimate their location within the body. Nature **212**(5060), 430–431 (1966)
17. Summerbell, D., Lewis, J.H., Wolpert, L.: Positional information in chick limb morphogenesis. Nature **244**(5417), 492–496 (1973)
18. Turing, A.M.: The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society B: Biological Sciences **237**(641), 37–72 (1952)
19. Wolpert, L.: The french flag problem: a contribution to the discussion on pattern development and regulation. Towards a Theoretical Biology pp. 125–133 (1968)
20. Wolpert, L.: Positional information and the spatial pattern of cellular differentiation. Journal of Theoretical Biology **25**(1), 1 – 47 (1969)
21. Zadorin, A.S., Rondelez, Y., Gines, G., Dilhas, V., Urtel, G., Zambrano, A., Galas, J.C., Estevez-Torres, A.: Synthesis and materialization of a reaction–diffusion french flag pattern. Nature Chemistry **9**(10), 990–996 (2017)