Greedy scheduling of cellular self-replication leads to optimal doubling times with a log-Frechet distribution

Rami Pugatch[1]

Simons Center for Systems Biology, School of Natural Sciences, Institute for Advanced Study, Princeton, NJ 08540

Edited by Ken A. Dill, Stony Brook University, Stony Brook, NY, and approved January 13, 2015 (received for review September 29, 2014)

Bacterial self-replication is a complex process composed of many de novo synthesis steps catalyzed by a myriad of molecular processing units, e.g., the transcription–translation machinery, metabolic enzymes, and the replisome. Successful completion of all production tasks requires a schedule—a temporal assignment of each of the production tasks to its respective processing units that respects ordering and resource constraints. Most intracellular growth processes are well characterized. However, the manner in which they are coordinated under the control of a scheduling policy is not well understood. When fast replication is favored, a schedule that minimizes the completion time is desirable. However, if resources are scarce, it is typically computationally hard to find such a schedule, in the worst case. Here, we show that optimal scheduling naturally emerges in cellular self-replication. Optimal doubling time is obtained by maintaining a sufficiently large inventory of intermediate metabolites and processing units required for self-replication and additionally requiring that these processing units be "greedy," i.e., not idle if they can perform a production task. We calculate the distribution of doubling times of such optimally scheduled self-replicating factories, and find it has a universal form—log-Frechet, not sensitive to many microscopic details. Analyzing two recent datasets of *Escherichia coli* growing in a stationary medium, we find excellent agreement between the observed doubling-time distribution and the predicted universal distribution, suggesting *E. coli* is optimally scheduling its replication. Greedy scheduling appears as a simple generic route to optimal scheduling when speed is the optimization criterion. Other criteria such as efficiency require more elaborate scheduling policies and tighter regulation.

self-replication | scheduling | critical path | random matrix | bacterial growth

An *Escherichia coli* bacterium is a remarkably efficient self-replicating organism. Given the right conditions, this rod-shaped bacterium will consume external metabolites and grow, adding new membrane-bound volume, while concurrently replicating its content composed chiefly from the transcription–translation machinery, metabolism, and DNA. After all of the essential elements have been replicated and spatially segregated, *E. coli* divides by completing the construction of the division plane, approximately midway between the old and the new poles. The two copies can both continue to self-replicate, as long as the permissive conditions persist.

The study of self-replication as an industrial process was pioneered by John von Neumann (1). In his first, less known model, the kinematic self-replicator, he envisioned a room full of elementary parts and a nontrivial self-replicating machine that copies itself by consuming these parts as substrates. The main goal of von Neumann was to understand how a physical system can become more complex over time. Motivated by the introduction of the universal Turing machine to the theory of computation, he introduced the concept of a "universal constructor" (U)—a machine that can read instructions and translate them into assembly of any machine in the factory, including itself, provided all of the substrates are available. A self-replicating factory is called nontrivial if it contains such a U machine as a component. In contrast, trivial self-replication is a simple autocatalytic process, such as template replication or crystal growth.

In von Neumann's model, replication unfolds as follows (2). First, a new chassis is made, and then the universal constructor is triggered to start reading the instructions and assembling all of the internal machinery, including itself. However, the new factory is not functional until a copy of the instructions is made. To keep the design simple, von Neumann suggested that the instructions should not instruct their own replication but rather be template-replicated by a dedicated machine, R (produced by U), that is triggered upon the completion of the instruction translation phase. Remarkably, these observations were made before the discovery of the molecular structure of DNA, and the molecular mechanism of DNA replication, transcription, and translation.

The emphasis of von Neumann's model is on the logical design of such a factory. Here we study the temporal organization of self-replication and, in particular, its scheduling. von Neumann's model assumed serial dynamics but cells do not grow serially but rather parallelize their growth. However, using a nonoptimal parallel scheduling scheme still results in slow overall growth compared with the optimum. This difference can have a significant effect on fitness in a competitive environment.

The "scheduling problem" is the problem of finding such a schedule. It is known that, if the number of processing units available is smaller than the number of production tasks that require them, finding an optimal schedule is typically computationally hard in the worst case, although certain heuristics are known to provide good approximations for the average instances (3–5).

Here, we study the scheduling problem of a self-replicating bacterial cell. Surprisingly, our analysis of recently measured datasets of *E. coli* exponentially growing in a stationary medium (6, 7) reveals that the measured distribution of doubling times fits well to the predicted distribution of doubling times of an

Significance

In production scheduling, a temporal assignment of tasks to their respective processing units is sought, such that precedence and resource constraint are satisfied and an optimization goal is minimized. So how is bacterial self-replication scheduled? Scarcity of resources creates a hard-to-solve scheduling problem in the worst case. Here, we show that proper loading of metabolic and catalytic reservoirs results in optimal scheduling, provided that the catalytic processing units are "greedy," i.e., do not idle if they can perform a task. The distribution of doubling times of optimally scheduled self-replication has a universal form—the log-Frechet distribution. Recent measurements of doubling times of *Escherichia coli* in different media well fit this distribution, suggesting that *E. coli* optimally scheduled its replication in these experiments.

Author contributions: R.P. designed research, performed research, contributed new reagents/analytic tools, analyzed data, and wrote the paper.

The author declares no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

[1]Email: rami.pugatch@gmail.com.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1418738112/-/DCSupplemental.

SYSTEMS BIOLOGY

PHYSICS

optimally scheduled self-replicating factory. This suggests that *E. coli* is optimally scheduling its replication in these media. To explain this result, first, a coarse-grained picture of a bacterial cell is presented, and its reaction graph is briefly introduced. The concept of a project graph, well-known in system engineering, is invoked to represent the temporal precedence constraints that exist among all of the de novo synthesis tasks that define the project. We introduce the project graph of a self-replicating and balanced factory and define the "replicative buffer"—the number of basic self-replicating units within a cell. A replicative buffer greater or equal to 1 allows a large class of random scheduling algorithms known as list algorithms (8) to obtain optimal completion times. We derive the distribution of optimal completion times for a special type of project graph representing balanced production. Finally, we present the aforementioned analysis of a recently published dataset of *E. coli* growing exponentially in a rich medium (6) and in minimal media supplemented with glucose or glucose and amino acids (7), and show that the data fit well to our predicted optimal universal curve, suggesting that *E. coli* in good growth conditions at steady state is optimally scheduling its replication. We then conclude with a short discussion and give an outlook to future extensions of this framework.

The Cell as an Autocatalytic Cycle

In a bacterial cell, prominent examples for processing units include metabolic enzymes, RNA and DNA polymerases (RNAP, DNAP), and ribosomes, which, like processing units in a factory, are required for a production task (biosynthesis), consume input materials and free energy, are not consumed during the process, yet are essential for its successful completion.

There are two unique features characterizing a self-replicating factory. The first is "closure." Processing units convert raw material into products, while consuming free energy. In a self-replicating factory, the products are the processing units. Thus, when all of the processing units complete their production tasks, each processing unit is present in duplicate (or more). To produce a processing unit may require several other processing units, self included. The second feature is "essentiality"—each processing unit is required by at least one other reaction that produces a different type of processing unit.

To illustrate how closure is obtained in a bacterial cell, we present a coarse-grained schematic of it in Fig. 1. Each symbol in the figure represents a family of functionally related macromolecules. For example, the replisome (the *R* machine in von Neumann's model), which is composed of many types of proteins and protein complexes, is represented by a single DNAP.

All proteins are synthesized by ribosomes, composed of ribosomal proteins (represented by the colored hexagons in the figure) and ribosomal RNA (rRNA) represented by a green strand. The rRNA is transcribed from DNA by RNAP, which is a self-assembled protein complex composed, in bacteria, from five different proteins.

To synthesize proteins, ribosomes require a pool of charged transfer RNA (tRNA) and a family of auxiliary proteins such as elongation, initiation, and maturation factors, as well as amino-acyl-synthetase proteins that catalyze the tRNA-amino acids charging. All of these auxiliary proteins are represented by the EF-Tu protein (green circle in Fig. 1). The tRNA is transcribed by RNAP and reaches its mature form with the help of some dedicated proteins (of course, mRNA is also transcribed by RNAP). Finally, membrane and division plane synthesis is facilitated by dedicated proteins. All these processes require numerous metabolites such as ribonucleosides and deoxyribonucleosides, amino acids, lipids and oligosaccharide, ATP, GTP, and NADH. All metabolites are either synthesized or imported from the outside by the metabolic machinery, which is mainly composed of metabolic proteins, represented by orange triangles in Fig. 1.

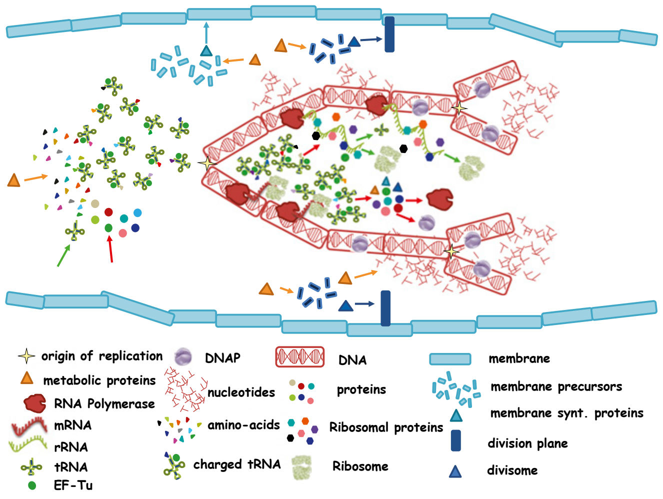

Fig. 1. Coarse-grained schematic to demonstrate the closure property of a bacterial cell. Each element in the figure is a representative member of a family of similar molecules sharing the same function. Arrows represent de novo synthesis. Orange arrows represent metabolic synthesis, with the exception of light and dark blue arrows, which represent membrane and division plane synthesis, respectively (the arrows point to the synthesis product). Synthesis of tRNA and self-assembly of ribosomes is represented by green arrows. Protein synthesis and self-assembly of proteins into protein complexes are represented by red arrows. All proteins are produced by preexisting ribosomes. All ribosomes are self-assembled from rRNA and ribosomal proteins (represented by colored hexagons). All RNA forms are transcribed by RNAP. Although it is possible to further deepen the level of description, any additional functional group not present in this schematic can be produced by one or more groups represented in the schematic. Furthermore, each functional group represented in the figure is essential for the production of members of another group.

The Catalytic Reaction Graph

To model the structure of a self-replicating factory, we follow ref. 9 and introduce a directed graph—the catalytic reaction graph, $G = \{V_G, E_G\}$, with a node set V_G composed of two types of nodes—representing materials (circles in Fig. 2) and reactions (squares in Fig. 2), and an edge set E_G composed of two types of edges, for materials consumed or produced by a reaction (solid arrows going into or out from a square reaction node in Fig. 2) or catalysts, required for the occurrence of certain reactions yet not consumed by them (represented by dashed arrows going into reaction nodes in Fig. 2). Using this formalism, we can represent a coarse-grained model for the cell that is essentially identical to the von Neumann model. This will facilitate a better characterization of different temporal organizations of self-replication, which we discuss subsequently.

Let $F = F_1 \cup F_2 \cup F_3 \cup F_4 \cup F_5$ (food set) be the set of all essential substrates and intermediates that are consumed by at least one de novo biosynthesis reaction in the cell, and f be the external set of materials (assumed to be fixed and abundant). For example, nucleotides and amino acids both belong to the set F. Let I be the DNA (instruction set), and R be the replisome machinery, which catalyzes the template replication of I. Furthermore, let U be the transcription–translation machinery (the universal constructor) that reads instructions from I and produces a copy of itself as well as all other processing units (proteins) that belong to P. The set P is responsible for biosynthesis processes and their control in the cell. It contains all of the proteins that are not members of R or U. For example, P includes metabolic proteins, transporters, and transcription factors. The set V represents the membrane-bound volume, made from all of the membranal layers including, e.g., the embedded protein transporters (which are a subset of P). Note that the membrane-bound

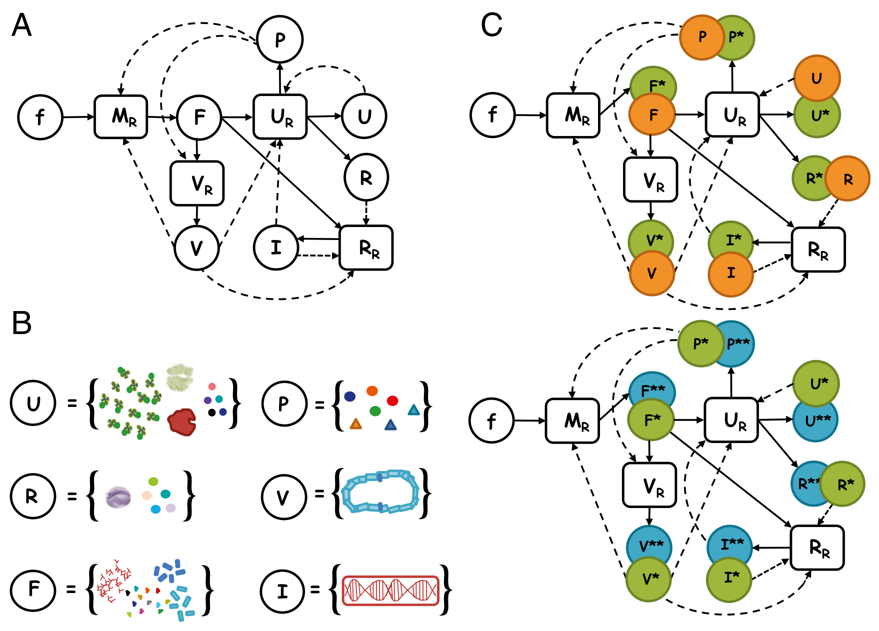

Fig. 2. *A* presents the reaction graph described by Eq. **1**. The material components, described by circles with capital letters in the figure, are shown in *B* using the same graphical representation as Fig. 1. For example, the universal constructor *U* contains ribosomes, RNAPs, tRNAs, and auxiliary proteins including initiation, elongation, and release factors, as well as a set of aminoacyl-transferase proteins, all required for streamlined production. In *C*, we illustrate how charging of the catalytic and metabolic pools allows the production lines for *I*, *U*, *V*, *P*, *R*, and *F* to work concurrently. A material node with one star was produced with the help of unstarred materials, and a two-star material node was produced by using one-star nodes. Thus, by temporal ordering of the material nodes, the reaction graph becomes the project graph, which is directed and acyclic. The level of initial charging determines the amount of overlap between different production cycles. Symmetric division resets the state of the pools to roughly or accurately one-half. The resulting project graph is thus periodic. When the average parallelism *p* is an integer immediately after division, the cell builds *p* basic self-replicating units in parallel.

volume is essential for all reactions, because in its absence reactions come to a halt due to molecular overcrowding.

To complete the definition of the catalytic reaction graph, we define the reaction node set. There are four types of reactions: M_R produces the set F from the external materials in f (metabolism). U_R produces all of the proteins in P, U, and R (as well as RNA for private consumption) from the set F (transcription and translation), and the reactions in V_R assemble the membrane, thus creating essential membrane-bound volume (including the reactions that build the division plane). The reaction R_R produces a copy of I while consuming elements from F using the existing I as template (DNA replication). The catalytic set C is composed of the subsets P, U, V, R, and I.

The following set of reactions summarizes the above description:

$$\begin{aligned}
&f+P_1 \rightarrow F+P_1 (\text{substrates synthesis})\\
&F_1+I_1+U \rightarrow I_1+2U (\text{ribogenesis, RNAP assembly})\\
&F_2+I_2+U \rightarrow I_2+U+P (\text{protein synthesis})\\
&F_3+P_2 \rightarrow V+P_2 (\text{membrane assembly})\\
&F_4+I_3+U \rightarrow I_3+U+R\ (\text{replisome synthesis})\\
&F_5+R+I \rightarrow 2I+R\ (\text{DNA replication}),
\end{aligned} \qquad [1]$$

where F, P, and I are the union of the sets $F_1,\ldots,F_5$, P_1,P_2, and I_1,I_2,I_3, respectively.

In Fig. 2*A*, the directed reaction graph of the system described by Eq. **1** is presented. Fig. 2*B* illustrates the types of molecules composing each material node in the reaction graph using the same graphical notation as in Fig. 1. The exact composition is not assumed to be one molecule per type. For example, U have a different number of ribosomes, RNA polymerases, elongation, initiation maturation and termination factors, aminoacyl transferases, and tRNAs.

From the Reaction Graph to the Project Graph

The project evaluation and review technique (PERT) is extensively used in system engineering as a means for scheduling important events in a project, and for estimating the expected project completion time and its distribution (10). A key concept in PERT is the project graph—a directed acyclic graph whose nodes represent milestone events in the project, marking the initiation or completion of a major project task; the directed edges represent the tasks themselves. In a production project, the initiation or completion of a production task (de novo biosynthesis) are the events, whereas the edge that connects them is the production task. Each edge in the project graph is also equipped with a nonnegative weight—the task duration. Edges also have resource demands—number and type of processing units and input materials required to perform them. We introduce two unique nodes in the project graph, s (start) and t (terminal), that mark the beginning and the end. The s node connects to all nodes that otherwise do not have entry nodes, and all of the otherwise terminating nodes feed into the t node (Fig. 3). We call a path from start to end an *st* path.

The structure of the project graph captures the temporal precedence constraints among all production tasks. The nodes function as "AND" gates, not allowing a new task to start (traverse an outgoing edge) before all predecessor tasks (preceding edges) are completed. Thus, the project graph is directed and acyclic, as it describes the progress in time of all of the tasks in the project.

Upon constructing the project graph and assigning durations to all its edges, the "optimal project completion time" T_c can be derived by finding the duration of the longest *st* path, known as the critical path. Other *st* paths can have "slack"—a duration gap between their completion time and T_c, allowing more flexibility in scheduling activities belonging to them. When fast completion time is a desirable goal, one can alleviate the constraints that cause a given critical path to be dominant, possibly creating a new (faster) critical path. Iteratively repeating this constraint elimination process until convergence, results in a maximally balanced project (11, 12), which is completed faster than the original project.

In Fig. 2*C*, the project graph is constructed from the reaction graph (Fig. 2*A*) by splitting the nodes to account for their temporal order of appearance. For example, when all input material nodes depicted in orange and green in Fig. 2*C* are simultaneously present at $t=0$, two self-replicating processes run in parallel and the doubling time T_μ will be the completion time of the slowest task out of the twelve production tasks, marked by the twelve solid arrows emanating from all of the reaction (square) nodes: $T_\mu = max_{i\in\{1,\ldots,12\}} T_i$.

The Scheduling Problem

In order for a biosynthetic task to be completed, the associated catalysts have to be allocated to the reaction along with the necessary input materials. Assuming the input materials are abundant, the following "scheduling problem" arises—how to temporally assign tasks to processing units such that the completion time of the entire project is minimized (5). In the noisy cellular milieu, we cannot expect scheduling algorithms to be precise. Instead, we ask what is expected from a randomized algorithm. To better characterize the scheduling problem, it is useful to introduce the following parameters. The workload T_w of a specific processor (catalyst) is defined as the sum over all durations of activities that require the processor. The workload represents the total demand for processors in terms of the processing time required. If there is only a single processor of a specific type, the optimal completion time is at least as great as its workload, because this processor has to perform all its tasks serially. We define the average parallelism p_c (13) for a particular processing unit by dividing its associated workload T_w, by the critical path duration T_c, and rounding up: $p_c = T_w/T_c$. The average parallelism is an estimate for the minimum number of processors required to allow for optimal completion time (see *SI Text*, section 7, for an estimate of the number of self-replicating units in *E. coli* doubling every 24 min based on protein and DNA replication workloads).

When there are excess of catalysts, then whenever there is a demand for them, it can readily be met, provided that they never idle if there is an available task (greedy scheduling). Thus, although excess is wasteful in terms of free energy cost, it is beneficial in terms of speed, as it allows for optimal completion time, determined only by the critical path duration. Moreover, this excess also allows for proper balancing in the presence of random delays that affect the actual demand. However, at some point, excess of catalysts can become suboptimal, because without further control, a greedy scheduling policy can deplete either the input resources or the catalytic pool, by allocating catalysts to reactions that have randomly completed earlier, causing a speedup in the progress of certain activities at the expense of others. This problem is mitigated by balancing as discussed below. Note that because the different task durations are stochastic variables, T_w, T_c, and p_c are also stochastic. We can thus generalize p_c to another parameter, $\bar{p}_c$, which is the minimal number of processing units required for optimal completion with a given confidence level c.

Basic Unit of Parallel Self-Replication

Now we apply the average parallelism concept to the self-replicating factory. We first define the initial demand vector $\vec{d}$, such that

$$d_j = d_j^c \oplus d_j^s = \sum_{i=1}^{n_R} C_{ij} + \sum_{i=1}^{n_R} S_{ij}, \qquad [2]$$

where the catalyst input matrix, C, of size $n_R \times n_{\mathrm{tot}}$, has an entry $(C)_{ij}$ equal to the number of catalysts of type j required by reaction i ($n_{\mathrm{tot}} = n_C + n_F$ is the number of input material types, n_C is the number of catalyst types, and n_F is the number of substrate types). The matrix S is defined similarly, i.e., S_{ij} is the number of substrates of type j consumed by reaction i. The vectors $\vec{d^c}$ and $\vec{d^s}$ separately account for the demand for catalysts and substrates for each reaction. The sum $\vec{d_c} \oplus \vec{d_s}$ is the direct sum of the two vectors. Thus, the demand vector counts the total number of processors and substrates required for the completion of a single round of parallel replication, i.e., simultaneously passing through all of the reactions in the reaction graph once.

Let $\vec{n}(t=0)$ be an occupation number vector whose elements account for the number of input materials (processing units and substrates) per type immediately following division ($t=0$), e.g., the initial number of ribosomes, genes, RNAPs, enzymes, and transporters. If any of the elements in $\vec{n}(0)$ is less than the corresponding element in $\vec{d}$, then the process is not fully parallel, because some processors or materials required for performing certain reactions are missing. To achieve full parallelism, the missing processors should be replenished.

When, on the other hand, $\vec{n}(0) = p \times \vec{d}$, with $p \geq 1$, then each reaction can potentially occur simultaneously $\lfloor p \rfloor$ times. In particular, when p is an integer, the doubling time T_μ, defined as $\vec{n}(T_\mu) = 2\vec{n}(0)$, is determined by the duration of the slowest reaction: $T_\mu = \max_{j\in\{1,\ldots,p\}, i\in\{1,\ldots,n_R\}} t_{ij}$, where t_{ij} is the time to complete the ith reaction by a complete set of processing units $j \in \{1,\ldots,p\}$. If, furthermore, the factory is balanced, the resulting growth is exponential, with p parallel self-replicating basic units advancing in unison with some residual statistical noise. In the absence of correlations, the resulting distribution of completion times is the maximum of identically and independently distributed random variables and hence equals to one of the three classical "extreme value statistics" limit distributions. However, due to the requirement for balanced production, the different reaction rates are coupled, and hence correlated. Thus, a different approach for calculating the distribution of doubling times is required.

To dynamically balance their inventory of catalysts and substrates, cells often use end-product feedback inhibition. For example, in the biosynthesis of the amino acid tryptophan, a metabolically costly amino acid, excess tryptophan binds to the enzyme at the top level of the dedicated metabolic pathway that produces it, thus down-regulating its own production (14). A beautiful mechanism for translational control was discovered by Nomura and coworker (15), where excess ribosome proteins that fail to bind to their targets alternatively bind to their respective mRNA, thus stopping their own production. All of these mechanisms use the excess end product, i.e., the part that was not consumed by downstream reactions, as a negative-feedback signal, down-regulating their own production, resulting in a dynamically balanced production line.

We define the basic unit of parallel self-replication as the smallest complete set of processing units and materials that is self-consistent: $\vec{n}(0) = \vec{d}$ (i.e., $p = 1$). This is the minimal set required for parallel execution of all n_R reactions (in Eq. **1**). It is also natural to define the basic "serial" unit of self-replication as the minimal set of all initial states $\vec{n}(t=0)$ with $p < 1$ that can still grow back to $p = 1$ and beyond (the time for this "catching up" is known as the lag phase).

An important class that interpolates between fully serial and fully parallel production is the pipelined self-replication, where the demand vector is charged in a way that allows one basic self-replicating unit to produce another basic self-replicating unit (marked with a star in Fig. 2*C*), with a time lag. Even before the catalysts finish building all of the machinery of the starred unit (next generation), the newly formed elements from the starred unit start replicating the double-star unit (marked with a double star in Fig. 2*C*). Thus, catalysts that finish their role in producing a given unit move to the next job of producing the same unit for a later generation. At steady state, there are multiple basic self-replicating units, with overlapping and lagging production cycles, until finally division (which is also scheduled periodically) resets this process. To conclude, in pipelined self-replication, there is an overlap in the production of several basic self-replicating units and an overall noninteger average parallelism (the integer part counts the number of overlapping cycles). The scheduling of self-replication can naturally lead to pipelined production, if tasks are ordered properly, and full parallelism is also possible when the lag times are set to zero across the factory.

Calculating Optimal Doubling Time

Consider a given project graph, and assume that the duration of each task is a random variable distributed according to a distribution that is exponentially decaying at large times. To calculate the project completion time, we introduce the exponentially weighted adjacency matrix of the project graph, which is defined as follows:

$$M_{ij} = e^{\beta T_{ij}}, \qquad [3]$$

where $\beta \gg 1$ is a large positive number, and $T_{ij} = T(e)$ is the duration of a task $e = (i,j) \in E_G$; otherwise if $(i,j) \notin E_G$, then $T_{ij} = -\infty$. We arrange the indices of the nodes such that the first node ($i = 1$) represents the project start, and the last node $i = n$ represents its end. The $(t,s) = (n,1)$ edge, $M_{n1} = 1$, is the previously

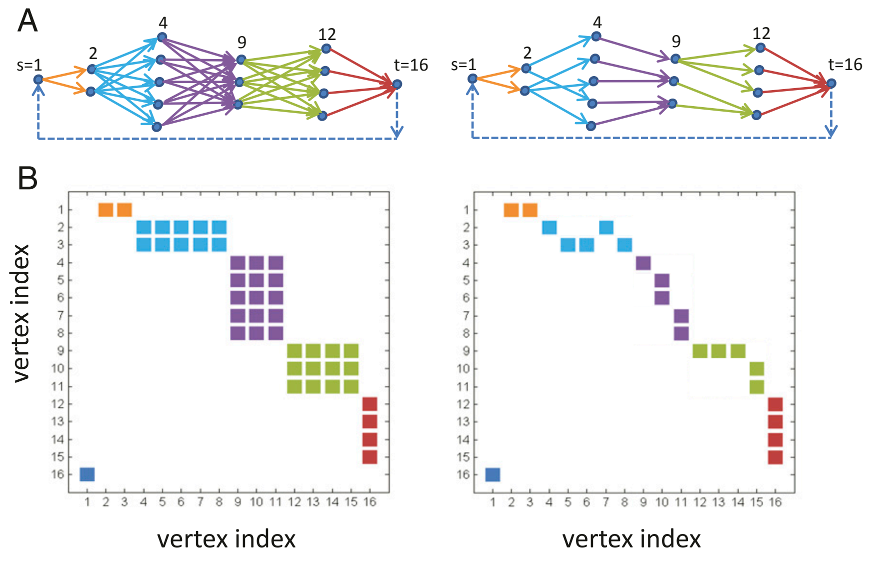

Fig. 3. Production in well-defined stages leads to a periodic project matrix and balanced growth. This figure illustrates this point by presenting two balanced project graphs with five stages (represented by five different edge colors). In the right-hand side, we present a tightly controlled project graph, where the edges that connect each stage form a complete bipartite graph. This is the tightest control possible; no stage can start before the completion of the previous stage. In the left-hand side, we present a less stringent project obtained by pruning the edges of the right-hand side graph. (*A*) The project graph. Note the "division" task (dashed backward arrow) that resets the factory back to the start node ($s=1$) upon completion of the last stage. (*B*) Adjacency matrices A of the two project graphs. Nonzero entries are colored according to the color of the edge they represent. For example, the entry $A_{12,16}$ that represents the edge from node 12 to the terminal node t ($t=16$) is marked by a red square.

defined resetting edge that returns the project back to start, upon the completion of the last task, thus mimicking the division event. We denote this matrix with the exponentially weighted entries and an auxiliary resetting node by M_c, where the dependence on β is suppressed for brevity. The matrix trace $Tr(M_c^k)$ is composed of a sum over all closed paths with k activities, exponentially weighted by their total duration. In the limit $\beta\to\infty$, the leading contribution is the longest path with k steps and its integer multiples. We further divide $Tr(M_c^k)$ by k, because there are k equally contributing nodes on each cycle of length k, depending on where we locate the cycle's start. Summing over all k values, we obtain a discrete path integral, which is a function of β. Dividing by β, we obtain the following:

$$T_c=\frac{1}{\beta}\sum_k\frac{Tr\left(M_c^k\right)u^k}{k}=\frac{1}{\beta}\sum_k\sum_l\frac{\lambda_l^k u^k}{k}=\frac{1}{\beta}\ln\frac{1}{det(I-uM_c)},\qquad [4]$$

where λ_l is the lth eigenvalue of M_c. Eq. **4** becomes exact in the limit $\beta\to\infty$ after regularization at $u=1$. To obtain the distribution of completion times, one has to calculate the probability distribution of $-\frac{1}{\beta}\ln det(I-uM_c)$ with a specified ensemble of random project matrices. The randomness can be for a fixed-graph topology with random weights (durations) or can include the graph topology. In the next sections, we focus on the former and solve for the distribution of completion times for a class of project matrices that describe a balanced production line. For a detailed derivation of Eq. **4** and further explanation, we refer the reader to the *SI Text*, sections 1–3 and Figs. S1 and S2.

Balanced Self-Replication Has a Periodic Project Matrix

Let $\vec{n}(t=0)$ be an occupation number vector with integer elements that counts the number of processing units and input materials present right after division (which we denote by $t=0$). For example, using the coarse-grained picture from Eq. **1** and the notation presented in Fig. 2, $\vec{n}(t=0)=(n_{F_1},\ldots,n_{F_5},n_{P_1},n_{P_2},n_U,n_V,n_R,n_{I_1},\ldots,n_{I_3})$. Each reaction produces other catalysts or substrates that allow other reactions to proceed. When the initial state is $(1,1,\ldots,1)$, doubling occurs at $(2,2,\ldots,2)$; if, however, the initial state is $(1,1,\ldots,1)\pm\delta\vec{n}$ with $0.5<\max_i(\delta n_i)<1$, then there are two overlapping self-replication cycles.

The balancing mechanism is important, as it prevents overproduction of certain materials on the expense of others. It locks the factory to the critical path. Recall that the node set V_G of a periodic digraph has a unique property that it can be partitioned into q disjoint sets, $V_G=V_1\cup V_2\cup\ldots\cup V_q$, such that every edge that originates from V_i ends up in V_{i+1} with $V_{q+1}=V_1$ (16). In our case, the node sets are milestone biosynthetic events, i.e., additions of a given fraction of F, P, U, V, R, or I, to $\vec{n}$. In a balanced self-replicating factory, the progress in the production of each processor type is regulated such that the production front—the plane perpendicular to the occupation number $\vec{n}(t)$, progresses uniformly (i.e., parallel to itself), up to some statistical tolerance $\delta\vec{n}$. Thus, $\vec{n}(t)$ progresses from one plane to another uniformly until reaching (within the allowed error margin) to the doubling plane, where it splits back, resulting in a global periodic matrix that describe the progress of the project. Hence, balanced growth in the biological sense (keeping the proportion fixed) amounts to locking all production rates to a global parameter, the critical path, and a balanced factory in the system engineering sense defined above, which is self-replicating, is also balanced in the biological sense. We note that the relevant scheduling scheme that supports such growth at steady state is cyclic scheduling (17).

The project completion time of an irreducible periodic matrix with a large spectral gap (*SI Text*, section 3) can be calculated analytically, due to the discrete rotational symmetry of the adjacency matrix spectrum. The eigenvalues satisfy $\lambda_l=|R_l|\mathbf{e}^{2\pi i l/q}$. The maximal eigenvalue is real and has a value, which we denote by $\lambda_{max}=R_{max}$. Plugging in Eq. **4** and noticing $R_{max}\gg 1$ because it scales exponentially with β (see *SI Text*, section 3, for more details), we obtain the following:

$$T_\mu=-\lim_{\beta\to\infty}\frac{1}{\beta}\sum_l\log(1-u|R_l|)=\lim_{\beta\to\infty}\frac{1}{\beta}\log R_{max}.\qquad [5]$$

The periodic matrix is asymmetric and contains elements that are distributed with a power law tail above the diagonal (because an exponent of a random variable with an exponential tail has a power law distribution) (*SI Text*, section 2). The distribution of the maximal eigenvalue of a random Wishart matrix with heavy tails was recently shown to belong to the Frechet class when the entries are power law distributed (18). Thus, for $\beta\gg 1$, the maximal eigenvalue of $M_c^tM_c$, a heavy-tailed Wishart matrix, is also distributed according to the Frechet distribution. The largest eigenvalue of $M_c^tM_c$ is real (Perron–Forbenius theorem) and equals the square of the maximal eigenvalue of M_c; hence the desired distribution of T_c is proportional to $\log R_{max}$, which is the log-Frechet distribution (*SI Text*, sections 1–6). This relation to the Wishart ensemble can breakdown if (i) β is small, where the Tracy–Widom distribution is expected (18). (ii) The spectrum is not gapped, e.g., when the project is not balanced (as in a totally serial project, where there is only one st path).

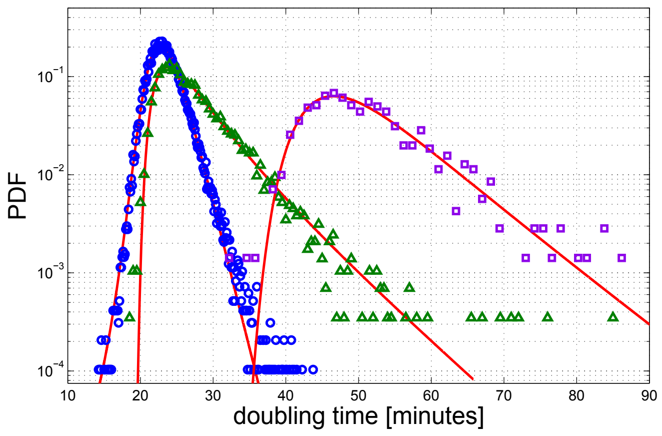

Fig. 4. Log-Frechet fits to the measured doubling times in three environments. Data from ref. 6 for *E. coli* cells growing in LB medium is shown in blue circles. Data from ref. 7 of *E. coli* cells growing in M9 medium supplemented with glucose and amino acids is marked with green triangles, and for growth in M9 supplemented with glucose, in purple squares. Log-Frechet curves with location, scale, and shape parameters estimated from a maximum-likelihood estimator (and with β fixed to 1) are shown in red lines. Roughly speaking, the shape parameter α of the log-Frechet distribution is a measure of the amount overlap between different *st* paths. When $\alpha = 1$, all *st* paths are independent of each other (not correlated). For the data shown, $\alpha_{ML} = 0.66$ (circles), 0.85 (triangles), and 0.74 (squares).

Our hypothesis regarding the existence of a replicative buffer can be tested experimentally by measuring the number of origins of replication right after division n_{OriC} and verifying that $\log_2(n_{OriC})$ is the replicative buffer, which is the maximal number of divisions of this cell upon switching to a depleted environment that only support growth of a single self-replicating unit (e.g., right before reaching the growth plateau in batch culture experiments, or like the conditions in a slow-growth chemostat).

Comparison with Measured Distributions of Doubling Times

To test our prediction of optimal doubling time, we analyzed two recent datasets of *E. coli* growing in different media. The first dataset from ref. 6 is of strain K12, substrain MG1655 growing in Luria–Bertani (LB) medium. The second dataset is taken from ref. 7 and is of *E. coli* strains SX701 and JE116, based on *E. coli* strain BW25993 (7) growing in M9 medium supplemented with glucose, or M9 medium supplemented with glucose and amino acids. The experiment of ref. 6 contains an impressive number of single-cell growth and division measurements, out of which we filter roughly 200,000 clean divisions of either mother, daughter, or granddaughter *E. coli* cells exponentially growing in LB broth at 37 °C within a narrow microfluidic canal, at an average doubling time of 21 min (6). The experiment of ref. 7 contains 6,000 growth and division measurements of *E. coli* growing in *M9* medium supplemented with glucose and amino acids, and 1,500 divisions in M9 media with glucose (7). The doubling time, T_μ, was calculated by the following formula:

$$T_\mu = \frac{T_{div}}{\log_2 R(T_{div})}, \qquad [6]$$

where T_{div} is the interdivision time, and $R(T_{div})$ is the length ratio (ratio of the length of the cell at division divided by its initial length). We compared two methods to calculate T_μ. The first is described by Eq. **6**, and the second is by fitting an exponent to the measured growth curve. We obtained practically identical doubling times in the two methods. After filtering noisy data (fits with $R^2 < 0.98$), we use the maximum-likelihood method to estimate the location, scale, and shape parameters of the Frechet distribution of the transformed data $e^{\beta(T_\mu/\overline{T}_\mu)}$, where $\overline{T}_\mu$ is the average doubling time and β is an arbitrary positive number. These parameters were then used for the log-Frechet distribution. As evident in Fig. 4, all doubling times from the three environments fit very well with the log-Frechet distribution. We tested two competing distributions, the gamma and the log-normal distributions, and found that they do not fit well with the data in contrast to the log-Frechet distribution (*SI Text*, Figs. S3 and S4).

Conclusion and Outlook

We studied the distribution of doubling times of an optimally scheduled self-replicating bacteria. Invoking PERT and the notion of balanced growth (19), a periodic structure for the project graph of a self-replicating production process was suggested. The distribution of optimal doubling times for this graph structure was calculated and found to have a universal shape—the log-Frechet distribution. Contrary to hard combinatorial optimization problems, finding the optimal scheduling in a self-replicating factory is possible using a greedy scheduling scheme, provided that the inventory of processing units and material inputs are a larger-than-one multiple of the demand vector. To maintain such a stoichiometric balance, some level of control over the rate of production of different types of processing units is required, which leads to a periodic project graph (i.e., a project graph with a periodic adjacency matrix). To corroborate our model, we analyzed *E. coli* in three different media using the dataset of ref. 6 for LB media and of *E. coli* in M9 media with glucose and either with or without amino acids (data from ref. 7) and showed that their doubling-time distributions agrees well with the predicted optimal completion time distribution—the log-Frechet distribution, suggesting that *E. coli* is optimally scheduling its self-replication in these experiments. This natural tendency toward optimality when minimizing the doubling time is the optimization criterion, suggests that avoiding optimal growth in a permissive environment is in fact a challenge for the cell, which requires additional regulation.

ACKNOWLEDGMENTS. I gratefully acknowledge the support of the Fulbright Foundation, the Eric and Wendy Schmidt Foundation, and the Jensen Fellowship.

1. von Neumann J, Burks AW (1966) *Theory of Self-Reproducing Automata* (Univ of Illinois Press, Urbana, IL).
2. Cairns-Smith AG (1971) *The Life Puzzle: On Crystals and Organisms and on the Possibility of a Crystal as an Ancestor* (Univ of Toronto Press, Toronto).
3. Garey MR, Johnson DS, Sethi R (1976) The complexity of flowshop and jobshop scheduling. *Math Oper Res* 1:117–129.
4. Das P, Moll M, Stamati H, Kavraki L, Clementi C (1979) Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann Discrete Math* 5:287–326.
5. Sinnen O (2007) *Task Scheduling for Parallel Systems* (Wiley-Interscience, Hoboken, NJ).
6. Wang P, et al. (2010) Robust growth of *Escherichia coli*. *Curr Biol* 20(12):1099–1103.
7. Ullman G, et al. (2013) High-throughput gene expression analysis at the level of single proteins using a microfluidic turbidostat and automated cell tracking. *Philos Trans R Soc Lond B Biol Sci* 368(1611):20120025.
8. Graham R, Lawler E, Lenstra J, Kan AR (1979) Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann Discrete Math* 5(2):287–326.
9. Hordijk W, Kauffman SA, Steel M (2011) Required levels of catalysis for emergence of autocatalytic sets in models of chemical reaction systems. *Int J Mol Sci* 12(5):3085–3101.
10. Kerzner H (2003) *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (Wiley, New York).
11. Trietsch D (2005) From management by constraints to management by criticalities. *Hum Syst Manag* 24:105–115.
12. Goldratt EM, Cox J (1992) *The Goal: A Process of Ongoing Improvement* (North River Press, Great Barrington, MA).
13. Eager DL, Zahorjan J, Lazowska ED (1989) Speedup versus efficiency in parallel systems. *IEEE Trans Comput* 38(3):408–423.
14. Stryer L (1995) *Biochemistry* (W. H. Freeman, New York).
15. Cole JR, Nomura M (1986) Translational regulation is responsible for growth-rate-dependent and stringent control of the synthesis of ribosomal proteins L11 and L1 in *Escherichia coli*. *Proc Natl Acad Sci USA* 83(12):4129–4133.
16. Jensen JB, Gutin GZ (2002) *Digraphs* (Springer, London).
17. Chretienne P (2000) On Graham's bound for cyclic scheduling. *Parallel Comput* 26:1163–1174.
18. Auffinger A, Arous GB, Peche S (2008) Poisson convergence for the largest eigenvalue of heavy tailed random matrices. *Ann Inst H Poincare* 45(3):589–610.
19. Cooper S (1991) *Bacterial Growth and Division: Biochemistry and Regulation of Prokaryotic and Eukaryotic Division Cycles* (Academic, San Diego).