A Mathematical Programming Formulation for the Budding Yeast Cell Cycle

by

Thomas D. Panning[*], Layne T. Watson[†], Clifford A. Shaffer[*], John J. Tyson[‡]

May 2, 2007

[*]Department of Computer Science
[†]Departments of Computer Science and Mathematics
[‡]Department of Biological Sciences
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Abstract. The budding yeast cell cycle can be modeled by a set of ordinary differential equations with 143 rate constant parameters. The quality of the model (and an associated vector of parameter settings) is measured by comparing simulation results to the experimental data derived from observing the cell cycles of over 100 selected mutated forms. Unfortunately, determining whether the simulated phenotype matches experimental data is difficult since the experimental data tend to be qualitative in nature (i.e., whether the mutation is viable, or which development phase it died in). Because of this, previous methods for automatically comparing simulation results to experimental data used a discontinuous penalty function, which limits the range of techniques available for automated estimation of the differential equation parameters. This paper presents a system of smooth inequality constraints that will be satisfied if and only if the model matches the experimental data. Results are presented for evaluating the mutants with the two most frequent phenotypes. This nonlinear inequality formulation is the first step toward solving a large-scale feasibility problem to determine the ordinary differential equation model parameters.

Keywords: systems biology, regulatory networks, eukaryote, nonlinear inequalities, feasibility problem

1. Introduction

Molecular cell biology describes how cells convert genes into behavior. This description includes how a cell creates proteins from genes, how those proteins interact, and how networks of interacting proteins determine physiological characteristics of the cell. The central biological question addressed here is how protein interactions regulate the cell cycle of budding yeast (*Saccharomyces cerevisiae*).

The budding yeast cell cycle consists of four phases (see Figure 1), with cell division occurring in the final phase. A newborn cell starts in G1 phase (unreplicated DNA), during which time it grows to a sufficiently large size to warrant a new round of DNA synthesis (S phase). After DNA synthesis has completed, the cell passes briefly through G2 phase (replicated DNA) and then enters M phase (mitosis, where the two copies of each DNA molecule are separated and the cell divides, creating two new cells that are in G1 phase).

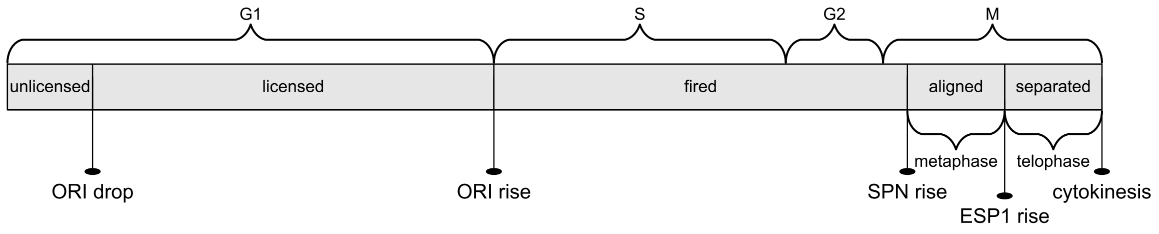

Figure 1. The phases and stages of the cell cycle. The four phases of the cell cycle are shown above the five stages. The events that delineate the cell cycle are at the bottom.

The protein interactions that govern these cell cycle events are modeled using differential equations that describe the rate at which each protein concentration changes. In general, the concentration of protein A, written as [A], changes according to the differential equation

$$\frac{d[\mathrm{A}]}{dt} = \text{synthesis} - \text{degradation} - \text{binding} + \text{dissociation} - \text{inactivation} + \text{activation},$$

where "synthesis" is the rate at which new protein A molecules are synthesized from amino acids (which depends on the concentration of active messenger RNA molecules for a particular protein), "degradation" is the rate at which protein A is broken down into amino acids and polypeptide fragments (which depends on the activity of specific proteolytic enzymes), "binding" is the rate at which protein A combines with other molecules to form distinct molecular complexes, "dissociation" is the rate at which these complexes break apart, "inactivation" is the rate at which certain post-translational modifications (e.g., phosphorylation) of protein A are made, and "activation" is the rate at which these modifications are reversed (e.g., dephosphorylation). Each of these rates is itself a function of the concentrations of the interacting species in the network. For example,

$$\begin{aligned}
\text{synthesis} &= k_1[\text{transcription factor}],\\
\text{degradation} &= k_2[\text{proteolytic enzyme}][\mathrm{A}],\\
\text{binding} &= k_3[\mathrm{A}][\mathrm{B}], \text{where B is a binding partner},\\
\text{dissociation} &= k_4[\mathrm{AB}],\\
\text{inactivation} &= \frac{k_5[\text{kinase}][\mathrm{A}]}{J_5 + [A]},\\
\text{activation} &= \frac{k_6[\text{phosphatase}][\mathrm{A}_p]}{J_6 + [\mathrm{A}_p]}, \text{where } \mathrm{A}_p \text{ is the phosphorylated form of A.}
\end{aligned}$$

In these rate laws, the ks are rate constants and the Js are Michaelis constants. Other differential equations must be used to determine the temporal dynamics of the concentrations of the "transcription factor," "proteolytic enzyme," "kinase," etc.

A simple example illustrating the spirit of the modelling approach in this paper follows. A rudimentary reaction network for the frog egg cell cycle [22] results in the three ordinary differential equations

$$\begin{aligned}
\frac{dM}{dt} &= \big(v_d'(1-D) + v_d''D\big)(C_T - M) - \big(v_w'(1-W) + v_w''W\big)M,\\
\frac{dD}{dt} &= v_d\left(\frac{M(1-D)}{K_{md} + (1-D)} - \frac{\rho_d D}{K_{mdr} + D}\right),\\
\frac{dW}{dt} &= v_w\left(-\frac{MW}{K_{mw} + W} + \frac{\rho_w(1-W)}{K_{mwr} + (1-W)}\right),
\end{aligned}$$

where M, D, and W are normalized protein concentrations, the Ks, ρs, and vs are rate constants, and the constant C_T is total (normalized) cyclin. For C_T above some threshold C_A, the cell enters mitosis and cycles. Finding this threshold and the periodic solution defining the cell cycle could be modelled by the system of constraints

$$\begin{gathered}
0 < \tau < t_1,\\
M(t_1) = M(t_1 + \tau) = M(t_1 + 2\tau),\\
\frac{dM}{dt}(t_1) = \frac{dM}{dt}(t_1 + \tau) = \frac{dM}{dt}(t_1 + 2\tau),\\
C_A < C_T,
\end{gathered}$$

where the variables would be a time t_1, a period τ, and a threshold C_A.

The budding yeast cell cycle model [7] consists of 36 such differential equations for two classes of variables: regulatory proteins and physiological "flags." The regulatory proteins are triggers for specific events of the budding yeast cell cycle: Cln2 triggers budding, Clb5 triggers DNA synthesis, Clb2 drives cells into mitosis, and Esp1 drives cells out of mitosis and back to G1. The physiological "flags" are dummy variables that track the strength of these trigger proteins. For example, "BUD" is an integral of the activity of Cln2; when BUD = 1, a new bud is initiated. "ORI," an integral of [Clb5], represents the state of "origins of replication." When ORI = 1 (this state is called "fired" origins), DNA synthesis is initiated; at cell division, when [Clb2] + [Clb5] drops below a threshold level, ORI is reset to zero (called "licensed" origins). Finally, "SPN" represents the alignment of replicated chromosomes on the mitotic spindle. SPN is driven by Clb2 activity; i.e., SPN is an integral of [Clb2].

In the budding yeast model there are 143 rate constant parameters (ks, Js, etc.). In some cases, these parameters can be calculated directly from laboratory experiments (e.g., apparent protein half-lives), but most parameters are difficult to obtain directly from experimentation. Normally, modelers determine the remaining parameters by making educated guesses, solving the differential equations numerically, comparing the simulation results with laboratory data, and then refining their guesses. (Modelers call this process "parameter twiddling" [2].) For the budding yeast cell cycle, the laboratory data consists of observed phenotypes of more than 100 mutant yeast strains constructed by disabling and/or over-expressing the genes that encode the proteins of the regulatory network.

Although parameter twiddling is extremely tedious, it was used to obtain a parameter vector (s_1, $s_2, \ldots,$ s_{143}) for which the model's predictions are consistent with almost all of the budding yeast mutants being modeled. Obviously, the modelers would prefer a method that allows them to spend more time improving the equations and less time tuning parameters. In addition, a person can only keep track of a few parameters at one time, which makes it easy for him or her to unwittingly miss a portion of the parameter space. For these reasons, modelers would prefer to use a tool that determines "good" parameters automatically, quickly, and accurately.

The current approach to this parameter estimation (nonlinear regression) problem is to assign a penalty to every discrepancy between the ordinary differential equation (ODE) model's predictions and experimental data, using all available mutant data, and then do an unconstrained (or simple bound constrained) minimization of this penalty function over the ODE parameter space ([6], [8], [18], [21]). Due to the qualitative nature of the experimental data (Section 2), some of these penalties are discrete and this penalty function is inherently discontinuous, while the ODE solution is a smooth function of the ODE parameters. Beyond the fact that discontinuous objective functions are difficult to minimize, either locally or globally, this penalty approach has deeper flaws. Biologists do not agree on what the penalty should be for a particular discrepancy, or even on which discrepancies should be penalized.

This paper takes a quite different approach to parameter estimation. The idea is to describe the mutant data as a system of smooth nonlinear inequalities derived from the (smooth) ODE model output and cell biology knowledge. These inequalities should be generally accepted by cell biologists, even though some of the details may be debatable. An ODE parameter vector that satisfies all of the inequalities thus defines an acceptable model, and is a *feasible* solution of the system of inequalities. The proposed approach to parameter estimation is to solve a feasibility problem defined by a system of smooth (continuous, piecewise C^∞) inequalities. One could surmise that what really is desired is the "most interior" point, defined by, e.g., constraint margins. The situation is not so simple, though, since a cell's viability is robust with respect to environmental variations, and therefore it is really the "most robust" feasible point that is sought. Modeling such biological robustness is another research topic.

The paper is organized as follows. After some definitions, Section 2 provides the necessary biology background. As a point of reference, Section 3 describes the penalty function model, and reports parallel computing results obtained on the 2200 processor System X. Section 4 presents the inequality models for all the mutants, the heart of the paper. Some preliminary numerical results for the model are given in Section 5, but an attempt to solve the full model (with approximately 11,500 constraints and 4,000 variables) is a major long term project.

2. Observed and Predicted Phenotypes

Experimental biologists have studied many budding yeast mutants to learn about the cell cycle regulatory system. Of these mutants, 115 were chosen to model (see Appendix A). A model of budding yeast can be considered acceptable only if it is able to duplicate the behavior of most of these mutants. (It would be too much to expect a model to account for *all* the "observations" because of lingering uncertainties about the reaction network and inevitable mistakes in phenotyping mutants.) When the model is used to simulate a mutant, the parameter vector can be changed only in ways that are dictated by the genetic changes in the mutant. Consider the hypothetical proteins A and B presented in the previous section: if a mutant has a modified form of B that does not bind to A, then in the parameter vector for that mutant, k_3 would be set to zero and all the other parameters would be kept at the wild type values.

The *observed* phenotype refers to the phenotype that was recorded in a laboratory experiment. The *predicted* phenotype refers to the phenotype that the mathematical model (with its associated parameters) predicts. The *wild type* is the normal strain of an organism. The *mutant* strains have genetic changes that make them behave differently from the wild type in some way.

When comparing the model to the experimental data, it is important to realize that much of the data from laboratory experiments is qualitative. Such data is of the form "the cell is viable but considerably larger than wild type cells" or "the cell arrests in G1 phase and eventually dies." The quantitative data that is available (e.g., duration of G1 phase, cell mass at division) is generally imprecise. With all these uncertainties, there may be many, clustered parameter vectors that allow the model to reproduce the experimental data sufficiently well. What matters for the model here is the structure of the cell cycle regulatory pathways, not the details of the biochemistry.

2.1 Rules of Viability

To compare solutions of the differential equations with experimental data, it is necessary to predict cell cycle properties from a simulation of regulatory protein dynamics. Viability is determined by four rules:

1. The modeled cell must execute the following events in order, or else the modeled cell is considered inviable:
   (a) DNA licensed for replication (modeled by a drop in [Clb2] + [Clb5] below K_{ez2}, which resets [ORI] to zero);

(b) start of DNA synthesis (due to a subsequent rise in [Clb2] + [Clb5], causing [ORI] to increase above one), signaling the end of G1 phase, before a wild-type cell in the same medium would divide twice;

(c) alignment of DNA copies (due to a rise in [Clb2], causing [SPN] to increase above one) while [Esp1] is less than 0.1;

(d) separation of DNA copies (modeled by [Esp1] increasing above 0.1, due to Pds1 proteolysis at anaphase);

(e) cellular division (modeled by [Clb2] dropping below a threshold K_{ez}), which resets [BUD] and [SPN] to zero.

2. The cell is inviable if division occurs in an "unbudded cell" (i.e., if [BUD] does not reach the value 0.8 before event (e) occurs).
3. The cell cycle should be stable, i.e., the squared relative differences of the masses and G1 phase durations in the last two simulated cycles should both be less than 0.05.
4. Lastly, the modeled cell is considered inviable if cell mass at division is greater than four times or less than one-fourth times the steady-state mass at division of the wild type in the same medium.

If the observed phenotype for a mutant does not complete one of the checkpoints (e.g., the mutant cells do not bud), then the predicted phenotype of that mutant must exhibit the same behavior. It is possible for a cell to complete all of the checkpoints in the first cycle and then become arrested somewhere in the second cycle. If a cell has this type of observed phenotype, then a correct model must predict the same number of cycles before arresting in the same manner. If the observed cell has a viable phenotype, then the model must predict a viable cell with a similar G1-phase length, and a similar mass at division.

2.2 Initial Conditions

In the experimental data set, many of the mutations are conditional, that is, the mutant cells when grown under "normal" conditions (say, glucose medium at room temperature) behave like wild-type cells, but when grown under "restrictive" conditions (say, galactose medium or elevated temperature) the cells express the genetic mutation and the aberrant phenotype. To model this situation at sample points in parameter space, start a "wild-type" simulation from arbitrary (but reasonable) initial conditions and integrate the differential equations for two full cycles, in order to wash out any effects of the initial conditions. Then record the state of the control system just after [Clb2] + [Clb5] falls through K_{ez2} at the beginning of the third cycle. These recorded values are used as initial conditions for simulating a steady state wild-type cell and for simulating each of the mutants.

3. Nonsmooth Penalty Function Formulation

This section describes a typical deviation (model prediction minus observed system response) based formulation of the parameter identification problem as the unconstrained (or at most simple bound constrained) minimization of a nonsmooth objective function. Numerical results using 1024 processors are presented for two different applicable optimization algorithms. Since these runs required many hours on 1024 processors, the need for high performance computing for the smooth inequality formulation in the next section should be clear.

The objective function takes the observed phenotype and predicted phenotype for all of the mutants and computes a nonnegative score. Zero indicates a perfect match and larger numbers indicate increasingly worse matches. The ensuing discussion uses the symbol O for observed phenotype values and P for predicted phenotype values.

A budding yeast phenotype for a single mutant is represented by a six-tuple (v, g, m, a, t, c), where the viability $v \in \{\text{viable, inviable}\}$, the real number $g > 0$ is the steady state length of the G1 phase, the real number $m > 0$ is the steady state mass at division, the stage when arrest occurred is

$$a \in \{\text{unlicensed, licensed, fired, aligned, separated}\},$$

the positive integer t is the arrest type, and the nonnegative integer c is the number of successful cycles completed. The observed and predicted phenotypes are written $O = (O_v, O_g, O_m, O_a, O_t, O_c)$ and $P = (P_v, P_g, P_m, P_a, P_t, P_c)$, respectively. Arrest types cannot be compared unless the stage of arrest is the same for both phenotypes.

In what follows, the ωs and σs are constants defined in Table 1. The rating function, R, compares the observed and predicted phenotypes for a mutant. This rating function is a modified version of the one developed by N. Allen et al. [3]; the only difference is that if O_v or P_v is missing, then $R(O,P) = \omega_v$. The rating function is split into four cases depending on the viability of the observed and predicted phenotypes. If O_v = inviable, P_v = viable, and O_c is missing, then $R(O,P) = \omega_v$, the same as if $O_c = 0$. Otherwise, if a needed classifier is missing, the term is simply dropped and does not contribute to the objective function. In the case that classifiers are missing, this allows the objective function value to be at or near zero when viability is in agreement between the phenotypes, and forces larger objective function values when viability is not in agreement.

The rating function $R(O,P)$ when all classifiers are present is given by

$$\omega_g \times \left(\frac{O_g - P_g}{\sigma_g}\right)^2 + \omega_m \times \left(\frac{\ln \frac{O_m}{P_m}}{\sigma_m}\right)^2,$$

if O_v = *viable* and P_v = *viable*, by

$$\omega_v \times \frac{1}{1+P_c},$$

if O_v = *viable* and P_v = *inviable*, by

$$\delta_{O,P} + \omega_c \times \left(\frac{O_c - P_c}{\sigma_c}\right)^2,$$

if $O_v = \textit{inviable}$ and $P_v = \textit{inviable}$, and by

$$\omega_v \times \frac{1}{1 + O_c},$$

if $O_v = \textit{inviable}$ and $P_v = \textit{viable}$, where δ is a real valued discrete function, used to assess a penalty for the arrest stage and type, given by

$$\delta_{O,P} = \begin{cases} \omega_a, & \text{if } O_a \neq P_a, \\ \omega_t, & \text{if } O_a = P_a \text{ and } O_t \neq P_t, \\ 0, & \text{if } O_a = P_a \text{ and } O_t = P_t. \end{cases}$$

The rating function is tuned by parameters to allow the modeler to adjust the relative importance of classifiers. The parameters given by Table 1 were set so that a rating of around ten indicates a critical error in the model's prediction of a phenotype.

Symbol	Definition	Value
ω_g	G1 length weight	1.0
σ_g	G1 length scale	10.0
ω_m	Mass at division weight	1.0
σ_m	Mass at division scale	$\ln 2$
ω_a	Arrest stage weight	10.0
ω_t	Arrest type weight	5.0
ω_c	Cycle count weight	10.0
σ_c	Cycle count scale	1.0
ω_v	Viability weight	40.0

Table 1. Constants used in objective function.

Denote the real numbers by $\mathcal{R}$, the nonnegative integers $\{0, 1, 2, \ldots\}$ by $\mathcal{Z}_+$, and the integers by $\mathcal{Z}$. Let

$$\begin{aligned} \mathcal{P} &= (v, g, m, a, t, c) \\ &= \{\text{viable, inviable}\} \times (0, \infty)^2 \\ &\quad \times \{\text{unlicensed, licensed, fired, aligned, separated}\} \\ &\quad \times \{1, \ldots, 10\} \times \mathcal{Z}_+ \end{aligned}$$

be the space of all budding yeast phenotypes and let the domain of the objective function be the box

$$\begin{aligned} \Omega = \{x \in \mathcal{R}^{143} : s_i/u_i \leq x_i \leq s_i \times u_i, \\ i = 1, \ldots, 143\}, \end{aligned}$$

where $u \in \mathcal{R}^{143}$ are positive scale factors reflecting modelers' knowledge about the rate constants, and $s \in \mathcal{R}^{143}$ is the modeler's best guess point. Let $T_j : \Omega \to \mathcal{P}$ simulate the jth mutant with the parameters $x_1, \ldots, x_{143}$ and compute the phenotype. Then the objective function $f : \Omega \to [0, \infty)$ is defined by

$$f(x) = \sum_{j=1}^{N_m} \mu_j R(O_j, T_j(x)),$$

where N_m is the number of mutant experiments, and $\mu_i \in \{1, 4\}$ is a weight that indicates whether the ith mutant is of normal or high importance. The objective function value at the biologists' best previously known point [7] is 433.

Two algorithms that show promise for optimizing the discontinuous objective function are briefly described next. Consider the problem of minimizing $f : B \to \mathcal{R}$, where $B = [l, u] \subset \mathcal{R}^n$ is a box.

3.1 DIRECT

The DIRECT (Dividing Rectangles) global minimization algorithm [14] requires the objective function to be Lipschitz continuous to guarantee convergence. Even though the objective function used here is discontinuous, the DIRECT algorithm seems to be an efficient and reasonable deterministic sampling strategy worth trying.

The DIRECT algorithm is one of a class of deterministic direct search algorithms that does not require gradients. It works by iteratively dividing the search domain into boxes that have exactly one function value at the box's center. In each iteration, the algorithm determines which boxes are most likely to contain a better point than the current minimum point—these boxes are called "potentially optimal". It then subdivides the potentially optimal boxes along their longest dimensions. Intuitively, a box is considered potentially optimal if it has the potentially best function value for a given Lipschitz constant.

For an illustration of how the DIRECT algorithm searches the domain on an example problem, see [20]. Both serial [10] and parallel ([11]–[13]) versions of DIRECT have been described in the literature.

3.2 MADS

A MADS (Mesh Adaptive Direct Search) algorithm, as defined by Audet and Dennis [5], minimizes a nonsmooth function $f : \mathcal{R}^n \to \mathcal{R} \cup \{+\infty\}$ under general constraints $x \in \Omega \subseteq \mathcal{R}^n$, $\Omega \neq \emptyset$. If $\Omega \neq \mathcal{R}^n$, the algorithm works with f_Ω, which is equal to f on Ω and $+\infty$ outside Ω. Using f_Ω in lieu of f is called a "barrier" approach to handling arbitrary constraints $x \in \Omega$.

In each iteration, a MADS algorithm evaluates the objective function f_Ω at a finite number of trial points. Central to these algorithms is the concept of a mesh, which is a discrete set of points in $\mathcal{R}^n$. Every previous trial point must lie on the current mesh, and in each iteration the algorithm may only generate new trial points on the current mesh. This is not as restrictive as it might sound because the algorithm changes the mesh after each iteration (with the restriction that at iteration k all previously evaluated points S_k remain in the new mesh).

Each iteration of a MADS algorithm consists of two steps: the SEARCH step and the POLL step. The SEARCH step may evaluate f_Ω at any finite number of mesh points. At which mesh points f_Ω is evaluated depends on the precise MADS algorithm in use. If the SEARCH step fails to find a mesh point at which f_Ω is less than $\min_{x \in S_k} f_\Omega(x)$, then the algorithm performs the POLL step by generating and evaluating f_Ω at new trial points around the current incumbent solution x_k, where $f_\Omega(x_k) = \min_{x \in S_k} f_\Omega(x)$. The *poll size parameter* Δ_k^p limits the distance between x_k and the new trial points. The set of new trial points is called a *frame*, and x_k is called the *frame center*. The algorithm evaluates f_Ω at points in the frame P_k until it encounters an improved point x^* ($f_\Omega(x^*) < f_\Omega(x_k)$) or it has evaluated f_Ω at all of the points in P_k.

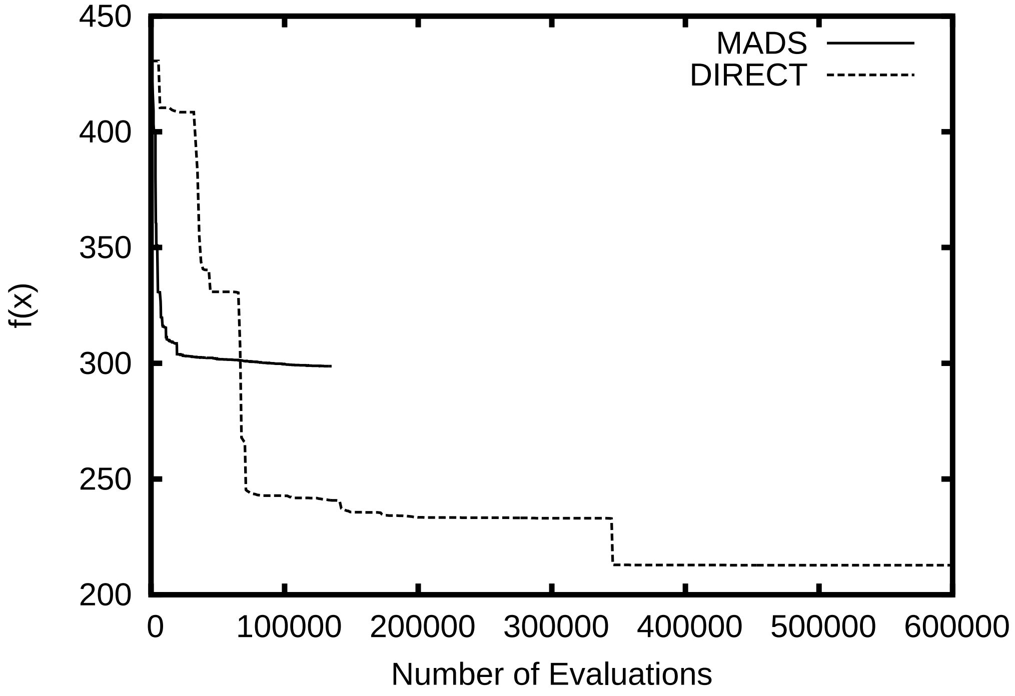

Figure 2. The objective function value at the best point found versus the number of evaluations for MADS and DIRECT.

After the algorithm has executed the SEARCH step and (conditionally) the POLL step, it sets the mesh size and poll size parameters, Δ^m_{k+1} and Δ^p_{k+1}, for the next iteration. Exactly how Δ^m_{k+1} and Δ^p_{k+1} are generated is determined by the individual algorithm in use. More precise descriptions of the MADS class of algorithms with examples can be found in [5] or [18].

3.3 Parallel Optimization Results

All computation took place on System X, a cluster of 1100 dual-processor Mac G5 nodes.

NOMAD is a C++ implementation of the MADS class of algorithms. To take advantage of System X, NOMAD's implementation of the POLL step was parallelized using a master/worker paradigm. The master ran the MADS algorithm as presented and sent requests to the workers whenever objective function values were needed. NOMAD, started from the modeler's best point s, evaluated the objective function 135,000 times over 813 iterations using 128 processors, converging at a point for which the objective function value was 299 (this point correctly models all but ten of the mutants).

pVTDirect [11] is a parallel implementation of DIRECT written in Fortran 95. While the DIRECT algorithm does not have a traditional "starting point", the first sample in each subdomain is always taken at the center of the subdomain bounding box. For this problem, the bounding box was designed so that the modeler's best point would be at the center and therefore would be evaluated before any other points. pVTDirect (with only one subdomain) ran for 473 iterations using 1024 processors and evaluated the objective function 1.5 million times, finding a point at which the objective function value was 212 (this point correctly models all but eight of the mutants).

Figure 2 shows the progress that each program was able to make in minimizing the objective function. While NOMAD was able to quickly find a better point than the modeler's best point,

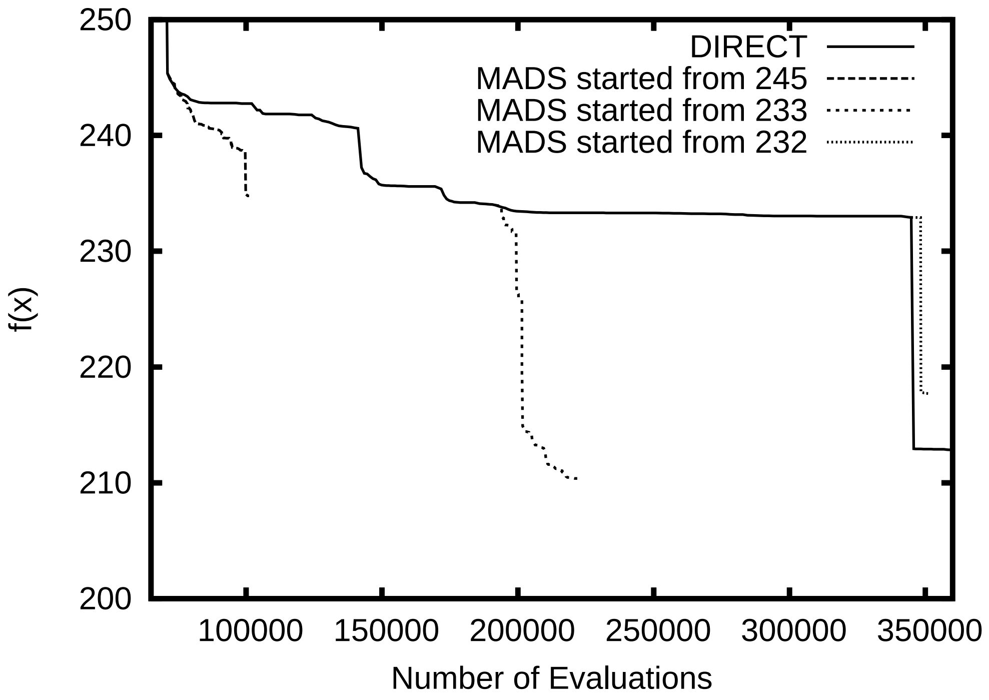

Figure 3. The performance of NOMAD when started from the best point at pVTDirect's 54th, 157th, and 239th iterations. The plots are shown as if the NOMAD runs started as soon as the respective pVTDirect iterations completed.

pVTDirect was eventually able to find an even lower point. This is expected behavior because NOMAD is designed for local optimization and pVTDirect is designed for global optimization, so NOMAD quickly found a nearby local minimum and stopped, but pVTDirect explored the parameter space and eventually found a better minimum. In a later run, NOMAD was started from pVTDirect's lowest point, but NOMAD was unable to make any further progress. After looking at Figure 2, it is tempting to believe that pVTDirect could have been stopped earlier (for instance, after 200,000 evaluations), and NOMAD started at pVTDirect's last best point could have found a point at which the objective function value was 212 or less. To test this, NOMAD was started at the best point at the 54th, 157th, and 239th iterations of pVTDirect. These points correspond to the beginning, middle, and end of the second-lowest plateau in Figure 2. As shown in Figure 3, NOMAD started from the middle point converged to a point at which the objective function value was 210. However, the NOMAD runs started at the beginning and end plateau points converge to worse points than pVTDirect's best point. These four extra NOMAD runs (including the one starting from pVTDirect's best point) show that an algorithm for improving intermediate results from pVTDirect is not so clear.

Finally, recall that $f(x) > 0$ means some mutant experimental data is not being matched, i.e., the parameters found by DIRECT and MADS do not fully explain all the data. These results motivated the smooth inequality formulation given next.

4. Formulation of Conditions as a Nonlinear System of Inequalities

For each phenotype, this section presents a system of constraints that will be satisfied if and only if the simulation predicts the same phenotype. The constraints are written using as much biological notation as possible, so [Esp1](t_6) refers to the concentration of the protein Esp1 at the time t_6. The constraints include time variables t_i where i is a positive integer; although these variables are not part of the model, they must still be found by the parameter estimator. In the constraints, there are several constants which are set as follows: $\epsilon = 0.05$, $\delta_{G_1} = 20$, and $\delta_M = 1.4$.

4.1 Viable Phenotype Constraints

The constraints for a viable cell to go through one cycle are listed below, with annotations denoted by angle brackets.

$$\begin{aligned}
t_1 < t_2 < t_3 < t_4 < t_5 < t_6 < t_8, && \langle 1\rangle \\
t_3 - t_1 < t_w, && \langle 2\rangle \\
t_2 < t_7 < t_8, && \langle 3\rangle \\
G_1 - \delta_{G_1} < t_3 - t_0 < G_1 + \delta_{G_1}, && \langle 4\rangle \\
\min_{t_1 \le t \le t_2} ([\mathrm{Clb2}](t) + [\mathrm{Clb5}](t)) > K_{\mathrm{ez2}} > [\mathrm{Clb2}](t_2+\epsilon) + [\mathrm{Clb5}](t_2+\epsilon), && \langle 5\rangle \\
\max_{t_2+\epsilon \le t \le t_3} [\mathrm{ORI}](t) < 1 < [\mathrm{ORI}](t_3+\epsilon), && \langle 6\rangle \\
\max_{t_1 \le t \le t_4} [\mathrm{SPN}](t) < 1 < [\mathrm{SPN}](t_5), && \langle 7\rangle \\
\max_{t_4 \le t \le t_5} [\mathrm{Esp1}](t) < 0.1 < [\mathrm{Esp1}](t_6), && \langle 8\rangle \\
[\mathrm{BUD}](t_7) > 0.8, && \langle 9\rangle \\
[\mathrm{Clb2}](t_8-\epsilon) > K_{\mathrm{ez}} > [\mathrm{Clb2}](t_8), && \langle 10\rangle \\
\max_{t_1 \le t \le t_8-\epsilon} [\mathrm{mass}](t) < 4m_w, && \langle 11\rangle \\
M - \delta_M < [\mathrm{mass}](t_8-\epsilon) < M + \delta_M. && \langle 12\rangle
\end{aligned}$$

⟨1⟩ These strict inequalities ensure the correct temporal ordering of the events defined by the times t_i. ⟨2⟩ [ORI] must rise above one before a wild type cell would divide twice in the same medium (e.g., glucose or galactose); t_w is set to the amount of time a simulated wild type cell takes to divide twice with the same biological parameters. ⟨3⟩ t_7 (which marks [BUD] rising above one) simply has to occur any time between t_2 and t_8. ⟨4⟩ G_1 is the length of the G1-phase, as observed in experiments. This ensures that the simulated cell is within a predefined distance δ_{G_1} of the observed value. ⟨5⟩ [Clb2] + [Clb5] drops, satisfying viability rule 1(a). ⟨6⟩ [ORI] rises, satisfying rule 1(b). ⟨7⟩ [SPN] rises, satisfying rule 1(c). ⟨8⟩ [Esp1] rises, satisfying rule 1(d). ⟨9⟩ [BUD] rises, satisfying rule 2. ⟨10⟩ [Clb2] drops, satisfying rule 1(e). ⟨11⟩ Mass is always less than four times the mass of the wild type in the same medium; m_w is the mass of a simulated wild type cell with the same biological parameters. ⟨12⟩ M is the observed mass at division.

A cell that meets the above constraints is viable for one cycle. For the first cycle, remember that the starting conditions are just after [Clb2] + [Clb5] dropped through K_{ez2}, so the fifth constraint must be omitted. For the rest of the cycles, repeat the constraints with variables $t_{1+7(n-1)}$, $t_{2+7(n-1)}$, …, $t_{8+7(n-1)}$ for cycle n. Note that the last time of the previous cycle is the first time in the current cycle. To enforce the stability requirements, add the constraints

$$\left(\frac{[\mathrm{mass}](t_{8+7(N-2)}) - [\mathrm{mass}](t_{8+7(N-1)})}{[\mathrm{mass}](t_{8+7(N-2)})}\right)^2 < 0.05$$

and

$$\left(\frac{(t_{3+7(N-2)} - t_{1+7(N-2)}) - (t_{3+7(N-1)} - t_{1+7(N-1)})}{(t_{3+7(N-2)} - t_{1+7(N-2)})}\right)^2 < 0.05,$$

where N is the total number of cycles.

4.2 Phenotypes for pds1Δ Mutants

The *pds1Δ* mutants are incapable of synthesizing Esp1, but some of these mutants still manage to separate the DNA copies. It is suspected that these mutants use a different mechanism to separate the DNA, but that mechanism is not included in this model. So for the purposes of this model, *pds1Δ* mutants are not required to meet viability rule 1(d), and the corresponding constraints should be omitted when evaluating mutants 61, 62, 66, 67, 71, and 113 in Appendix A.

4.3 Inviable Phenotype Constraints

A cell that fulfills the above constraints is considered viable. Some of the mutants have an observed phenotype of inviable, so their constraints will be different. The constraints for an inviable mutant are determined by when and how the cell arrests. There are four major types of arrest stages: G1 arrest, metaphase arrest, G2 arrest, and telophase arrest. The following subsections present the inequality constraints for each of these arrest types.

4.4 G1 Arrest

In terms of the model, a G1 arrest means that none of [ORI], [SPN], or [BUD] should rise to one before the cell cycle is considered arrested. Whether or not [Clb2] + [Clb5] drops below K_{ez} has no effect on whether the cell is G1 arrested, so it is not mentioned in the constraints. A cell arrests in G1 either because its mass has become greater than $4m_w$ (see viability rule 4) or because t_w time has passed (see checkpoint 1(b)). Constraints for a G1 arrested mutant are thus

$$\max\{t_1 - t_w, \max_{0\le t\le t_1} [\text{mass}](t) - 4m_w\} > 0,$$

$$\max_{0\le t\le t_1} [\text{ORI}](t) < 1,$$

$$\max_{0\le t\le t_1} [\text{SPN}](t) < 1,$$

$$\max_{0\le t\le t_1} [\text{BUD}](t) < 1.$$

The first inequality ensures that t_1 is a time after the cell has arrested. Specifically, the quantity $t_1 - t_w$ will be greater than zero if a wild type cell could divide twice before t_1. Similarly, the quantity $\max_{0\le t\le t_1} [\text{mass}](t) - 4m_w$ will be greater than zero if the cell has grown to a mass greater than $4m_w$ before t_1. The maximum of these quantities is used because violating any of the viability rules causes a cell to be arrested.

4.5 G2 Arrest

For a cell to be arrested in G2, it must execute the first two checkpoints of a viable cell, but [SPN] must stay low.

$$t_1 < t_2 < t_3 < t_4,$$

$$t_3 - t_1 < t_w,$$

$$\min_{t_1\le t\le t_2} ([\text{Clb2}](t) + [\text{Clb5}](t)) > K_{\text{ez2}} > [\text{Clb2}](t_2 + \epsilon) + [\text{Clb5}](t_2 + \epsilon),$$

$$\max_{t_2+\epsilon\le t\le t_3} [\text{ORI}](t) < 1 < [\text{ORI}](t_3 + \epsilon),$$

$$\max_{0\le t\le t_4} [\text{SPN}](t) < 1,$$

$$\max_{0\le t\le t_4} [\text{mass}](t) > 4m_w.$$

4.6 Metaphase Arrest

If a cell is arrested in metaphase, its chromosomes are aligned on its spindles (i.e., the [SPN] checkpoint must be reached) but the chromosomes have not separated (i.e., Esp1 has not activated). Metaphase arrested cells may be budded or unbudded. This means that the following constraints must be met.

$$t_1 < t_2 < t_3 < t_4 < t_5 < t_6,$$
$$t_3 - t_1 < t_w,$$
$$\min_{t_1 \le t \le t_2} ([\text{Clb2}](t) + [\text{Clb5}](t)) > K_{\text{ez2}} > [\text{Clb2}](t_2 + \epsilon) + [\text{Clb5}](t_2 + \epsilon),$$
$$\max_{t_2 + \epsilon \le t \le t_3} [\text{ORI}](t) < 1 < [\text{ORI}](t_3 + \epsilon),$$
$$\max_{t_1 \le t \le t_4} [\text{SPN}](t) < 1 < [\text{SPN}](t_5),$$
$$\max_{t_4 \le t \le t_5} [\text{Esp1}](t) < 0.1,$$
$$\max_{0 \le t \le t_6} [\text{mass}](t) > 4m_w.$$

4.7 Telophase Arrest

While the G1 phase is the earliest a cell can be arrested, telophase is the latest a cell can become arrested. A telophase arrested cell must complete all of the checkpoints except that [Clb2] can not drop below K_{ez} before the cell arrests. The first eight constraints for such a cell would be the same as the constraints for a viable cell. The ninth constraint would be removed, and the tenth constraint would be changed to

$$\max_{t_2 + \epsilon \le t \le t_7} [\text{Clb2}](t) < K_{\text{ez}} < \min_{t_7 + \epsilon \le t \le t_8} [\text{Clb2}](t),$$

with the additional constraints

$$t_1 < t_7 < t_8,$$
$$[\text{mass}](t_8) > 4m_w.$$

4.8 Evaluating All of the Mutants

As described earlier, a parameter vector must satisfy all of the constraints for all of the mutants before it can be considered feasible. Each mutant requires a separate simulation and will have its own set of time variables derived from its concentrations' trajectories. The mutants are numbered as in Appendix A, so the time for the ith mutant will be $t^{\{i\}}$. Also, because each mutant modifies the parameter vector slightly, the concentration of a substance at a specific time will vary among the mutants. Rather than explicitly specify which parameter vector is being used, the superscript of the time will indicate the parameter vector being used. For instance, mutant 13 is a G1 arrested mutant, so its constraints would be

$$\max\{t_1^{\{13\}} - t_w, \max_{0 \le t^{\{13\}} \le t_1^{\{13\}}} [\text{mass}](t^{\{13\}}) - 4m_w\} > 0,$$
$$\max_{0 \le t^{\{13\}} \le t_1^{\{13\}}} [\text{ORI}](t^{\{13\}}) < 1,$$
$$\max_{0 \le t^{\{13\}} \le t_1^{\{13\}}} [\text{SPN}](t^{\{13\}}) < 1,$$
$$\max_{0 \le t^{\{13\}} \le t_1^{\{13\}}} [\text{BUD}](t^{\{13\}}) < 1.$$

Table 2. The numbering of the viable constraints for Tables 4, 5, 6, and 7.

Index	**Constraint**
1	$t_1 < t_2$
2	$t_2 < t_3$
3	$t_3 < t_4$
4	$t_4 < t_5$
5	$t_5 < t_6$
6	$t_1 < t_7$
7	$t_7 < t_8$
8	$t_6 < t_8$
9	$t_3 - t_1 < t_w$
10	$\lvert t_3 - t_0 - G_1 \rvert < \delta_{G_1}$
11	G1 stability constraint
12	$\min_{t_1 \le t \le t_2}([\text{Clb2}](t) + [\text{Clb5}](t)) > K_{\text{ez2}}$
13	$K_{\text{ez2}} > [\text{Clb2}](t_2 + \epsilon) + [\text{Clb5}](t_2 + \epsilon)$
14	$\max_{t_2+\epsilon \le t \le t_3} [\text{ORI}](t) < 1$
15	$1 < [\text{ORI}](t_3 + \epsilon)$
16	$\max_{t_1 \le t \le t_4} [\text{SPN}](t) < 1$
17	$1 < [\text{SPN}](t_5)$
18	$\max_{t_4 \le t \le t_5} [\text{Esp1}](t) < 0.1$
19	$0.1 < [\text{Esp1}](t_6)$
20	$[\text{BUD}](t_7) > 1$
21	$[\text{Clb2}](t_8 - \epsilon) > K_{\text{ez}}$
22	$K_{\text{ez}} > [\text{Clb2}](t_8)$
23	$\max_{t_1 \le t \le t_8-\epsilon} [\text{mass}](t) < 4m_w$
24	$\big\lvert [\text{mass}](t_8 - \epsilon)/m_w - M \big\rvert < \delta_m$
25	Mass stability constraint

5. Biological Results

To test this formulation, the constraints for the telophase arrest and viable phenotypes were evaluated at two parameter vectors. The *pds1*Δ mutants were excluded from this test, leaving 61 mutants with an observed phenotype of viable and 15 mutants with an observed phenotype of telophase arrested. The first parameter vector used the best manually obtained biological parameters [7], and the second vector used the best biological parameters found using optimization algorithms on a penalty function formulation of the problem [18].

For both vectors, the time parameters were found using an algorithm that examines the ODE model simulation time series output and picks sensible values. In one pass through the simulation time series output, this algorithm attempts to pick the time variables so that they are in the proper order, and if possible, the constraints are satisfied. During the simulation, the algorithm keeps track of the maximum and minimum values for the model variables in the constraints (e.g., [BUD], [SPN]). When an event (cf. Section 2.1) occurs, the algorithm sets the respective time variables and then checks to see if all of the earlier time variables have been set. If there are earlier time variables that have not been set, the algorithm attempts to set each of them to a time that maintains the ordering of the variables and minimizes the violation of the constraints. A more

Table 3. The numbering of the telophase arrest constraints for Tables 4, 5, 6, and 7.

Index	Constraint
1	$t_1 < t_2$
2	$t_2 < t_3$
3	$t_3 < t_4$
4	$t_4 < t_5$
5	$t_5 < t_6$
6	$t_1 < t_7$
7	$t_7 < t_8$
8	$\max_{t_2+\epsilon \le t \le t_3} [\mathrm{ORI}](t) < 1$
9	$1 < [\mathrm{ORI}](t_3 + \epsilon)$
10	$\max_{t_1 \le t \le t_4} [\mathrm{SPN}](t) < 1$
11	$1 < [\mathrm{SPN}](t_5)$
12	$\max_{t_4 \le t \le t_5} [\mathrm{Esp1}](t) < 0.1$
13	$0.1 < [\mathrm{Esp1}](t_6)$
14	$\max_{t_1+\epsilon \le t \le t_7} [\mathrm{Clb2}](t) < K_{\mathrm{ez}}$
15	$K_{\mathrm{ez}} < \min_{t_7+\epsilon \le t \le t_8} [\mathrm{Clb2}](t)$
16	$[\mathrm{mass}](t_8) > 4m_w$

formal description of this algorithm (for the viable phenotype) is given below (ϵ comes from the beginning of Section 3). There is a similar algorithm for the telophase arrested phenotype.

```
check_*: flags for checking earlier events
o: offset into the time parameters
n_c: number of cycles completed
reset_cycle: flag for resetting best times
s: maximum time that a previous event can be set to
t: current time
t_f: end of the simulation time
t_b: best time for [BUD] rising
t_c: best time for [Clb2] + [Clb5] dropping
t_e: best time for [Esp1] rising
t_r: best time for [ORI] rising
t_s: best time for [SPN] rising
$t := 0$; $n_c := 0$; $o := 1$; $s := 0$; $t_c := 0$; $t_r := 0$; $t_s := 0$; $t_e := 0$; $t_b := 0$
while ($t < t_f$) do
    check_Clb2_Clb5 := false; check_ORI := false; check_ORI := false; check_SPN := false
    check_Esp1 := false; check_BUD := false; reset_cycle := false
    if ($n_c > 1$) then
        $o := 8 + (n_c - 1) \times 7$
    else
        $o := 1$
    end if
    Advance $t$ to the next time in the ODE simulation time series output.
    if ([Clb2] + [Clb5] dropped through $K_{\mathrm{ez2}}$) then
        $t_{o+1} := t - \epsilon/2$; $t_r := -1$; $t_s := -1$; $t_e := -1$
```

```
elseif ([ORI] rose through one) then
    $t_{o+2} := t - \epsilon/2$; $s := t_{o+2} - \epsilon$; $t_s := -1$; $t_e := -1$; $check\_Clb2\_Clb5 :=$ true
elseif ([SPN] rose through one) then
    $t_{o+3} := t - \epsilon$; $t_{o+4} := t + \epsilon$; $s := t_{o+3} - \epsilon$; $t_e := -1$; $check\_ORI :=$ true
elseif ([Esp1] rose through 0.1) then
    $t_{o+5} := t + \epsilon$; $s := t_{o+5} - \epsilon$; $check\_SPN :=$ true
elseif ([BUD] rose through one) then
    $t_{o+6} := t + \epsilon$
elseif ([Clb2] rose through $K_{\mathrm{ez}}$) then
    $t_{o+7} := t - \epsilon/2$; $n_c := n_c + 1$; $s := t_{o+7} - \epsilon$
    $check\_Esp1 :=$ true; $check\_BUD :=$ true; $reset\_cycle :=$ true
end if
if ($check\_BUD =$ true and $t_{o+6}$ has not been set) then
    if ([BUD]($s$) < [BUD]($t_b$)) then
        $t_{o+6} := t_b$
    else
        $t_{o+6} := s$
    end if
end if
if ($check\_Esp1 =$ true and $t_{o+5}$ has not been set) then
    if ([Esp1]($s$) < [Esp1]($t_e$)) then
        $t_{o+5} := t_e$
    else
        $t_{o+5} := s$
    end if
    $check\_SPN :=$ true
end if
$s := t_{o+5} - \epsilon$
if ($check\_SPN =$ true and $t_{o+4}$ has not been set) then
    if ([SPN]($s$) < [SPN]($t_s$)) then
        $t_{o+3} := t_s - \epsilon$; $t_{o+4} := t_s + \epsilon$
    else
        $t_{o+3} := s - \epsilon$; $t_{o+4} := s + \epsilon$
    end if
    $check\_ORI :=$ true
end if
$s := t_{o+3} - \epsilon$
if ($check\_ORI =$ true and $t_{o+2}$ has not been set) then
    if ([ORI]($s$) < [ORI]($t_r$)) then
        $t_{o+2} := t_r$
    else
        $t_{o+2} := s$
    end if
    $check\_Clb2\_Clb5 :=$ true
$s := t_{o+2} - \epsilon$
if ($check\_Clb2\_Clb5 =$ true and $t_{o+1}$ has not been set) then
```

```
        if ([Clb2]($s$) + [Clb5]($s$) > [Clb2]($t_c$) + [Clb5]($t_c$)) then
            $t_{o+1} := t_c$
        else
            $t_{o+1} := s$
        end if
    end if
    if ($reset\_cycle$ = true) then
        $t_c := -1$; $t_r := -1$; $t_s := -1$; $t_e := -1$; $t_b := -1$
    end if
    if ([Clb2]($t$) + [Clb5]($t$) < [Clb2]($t_c$) + [Clb5]($t_c$) or $t_c < 0$) then
        $t_c := t$
    end if
    if ([ORI]($t$) > [ORI]($t_r$) or $t_c < 0$) then
        $t_r := t$
    end if
    if ([SPN]($t$) > [SPN]($t_s$) or $t_c < 0$) then
        $t_s := t$
    end if
    if ([Esp1]($t$) > [Esp1]($t_e$) or $t_c < 0$) then
        $t_e := t$
    end if
    if ([BUD]($t$) > [BUD]($t_b$) or $t_c < 0$) then
        $t_b := t$
    end if
end while
```

For conciseness, the violated constraints for viable mutants are listed as "Cx-Ny", where x indicates the cycle in which the violation occurred, and y is an index into Table 2 that indicates which constraint was violated. The violated constraints for telophase-arrested mutants are listed as "Ny", where y is an index into Table 3. The results of evaluating the viable constraints on all of the mutants with an observed phenotype are shown in Tables 4, 5, 6, and 7. Tables 4 and 5 show the mutants that satisfied and did not satisfy the constraints, respectively, when the manually obtained biological parameters were used. Tables 6 and 7 show the same for the mathematically optimized biological parameters.

Table 4. Mutants that had no violated constraints, manually obtained parameters.

Wild type in glucose
Wild type in galactose
cln1Δ cln2Δ
GAL-CLN2 cln1Δ cln2Δ
cln1Δ cln2Δ sic1Δ
cln1Δ cln2Δ cdh1Δ
cln3Δ
bck2Δ
Multi-copy *BCK2*
cln3Δ bck2Δ GAL-CLN2 cln1Δ cln2Δ

*cln1*Δ *cln2*Δ *cln3*Δ *GAL-CLN2*
*cln1*Δ *cln2*Δ *cln3*Δ *GAL-CLN3*
*cln1*Δ *cln2*Δ *cln3*Δ *sic1*Δ
*cln1*Δ *cln2*Δ *cln3*Δ *cdh1*Δ
*cln1*Δ *cln2*Δ *cln3*Δ multi-copy *CLB5*
*cln1*Δ *cln2*Δ *cln3*Δ *GAL-CLB5*
*cln1*Δ *cln2*Δ *cln3*Δ multi-copy *BCK2*
*sic1*Δ
GAL-SIC1
*GAL-SIC1 GAL-CLN2 cln1*Δ *cln2*Δ
*GAL-SIC1 GAL-CLN2 cln1*Δ *cln2*Δ *cdh1*Δ
*sic1*Δ *cdh1*Δ *GALL-CDC20*
*cdh1*Δ
*cdc6*Δ*2-49*
*sic1*Δ *cdc6*Δ*2-49*
GAL-CLB2
Multicopy *GAL-CLB2*
*CLB2-db*Δ
*CLB2-db*Δ in galactose
*CLB2-db*Δ multicopy *SIC1*
*CLB2-db*Δ *GAL-SIC1*
*CLB2-db*Δ *clb5*Δ
*CLB2-db*Δ *clb5*Δ in galactose
*clb5*Δ *clb6*Δ
GAL-CLB5
*GAL-CLB5 cdh1*Δ
*CLB5-db*Δ
*tem1*Δ
GAL-TEM1
tem1-ts GAL-CDC15
*tem1*Δ *net1-ts*
tem1-ts multicopy *CDC14*
*cdc15*Δ
Multicopy *CDC15*
*cdc15*Δ *net1-ts*
*cdc15-ts*multicopy *CDC14*
net1-ts
GAL-NET1
cdc14-ts
GAL-NET1 GAL-CDC14
*TAB6-1 cdc15*Δ
*mad2*Δ
*bub2*Δ
*mad2*Δ *bub2*Δ
APC-A
*APC-A cdh1*Δ

APC-A cdh1Δ GAL-SIC1
APC-A cdh1Δ GAL-CDC6
APC-A cdh1Δ multicopy *CDC20*
swi5Δ
sic1Δ cdc6Δ2-49 cdh1Δ GALL-CDC20
APC-A sic1Δ
APC-A GAL-CLB2

Table 5. Mutants that had violated constraints, manually obtained parameters.

GAL-CLN2 cln1Δ cln2Δ cdh1Δ	C3-N20, C4-N19, C4-N20, C5-N15, C5-N17, C5-N20, C6-N15, C6-N17, C6-N20, C7-N15, C7-N17, C7-N20, C8-N15, C8-N17, C8-N20
GAL-CLN3	C6-N1, C8-N11, C5-N19, C6-N13, C6-N17, C7-N15, C7-N17, C8-N15, C8-N17
cln1Δ cln2Δ bck2Δ	C8-N24
cdh1Δ cdc6Δ2-49	C8-N10
GAL-CLB2 sic1Δ	N12, N16
GAL-CLB2-dbΔ	N12
GAL-ESP1 cdc20-ts	N5, N12
cdc14-ts GAL-SIC1	C1-N20, C2-N13, C2-N18, C2-N22, C3-N13, C3-N18, C3-N22, C4-N13, C4-N18, C4-N22, C5-N13, C5-N18, C5-N22, C6-N13, C6-N18, C6-N22, C7-N13, C7-N18, C7-N22, C8-N13, C8-N18, C8-N22
TAB6-1	C1-N8
TAB6-1 CLB1 clb2Δ	C1-N17, C2-N15, C2-N17, C3-N15, C3-N17, C4-N15, C4-N17, C5-N15, C5-N17, C6-N15, C6-N17, C7-N15, C7-N17, C8-N15, C8-N17
APC-A cdh1Δ in galactose	C1-N20, C2-N13, C2-N18, C2-N20, C2-N22, C3-N13, C3-N18, C3-N20, C3-N22, C4-N13, C4-N18, C4-N20, C4-N22, C5-N13, C5-N18, C5-N20, C5-N22, C6-N13, C6-N18, C6-N20, C6-N22, C7-N13, C7-N18, C7-N20, C7-N22, C8-N13, C8-N18, C8-N20, C8-N22
APC-A cdh1Δ multicopy *SIC1*	C1-N17, C2-N15, C2-N17
APC-A cdh1Δ multicopy *CDC6*	C8-N11, C7-N22, C8-N13, C8-N18, C8-N22

Table 6. Mutants that had no violated constraints, optimized parameters.

Wild type in glucose
Wild type in galactose
cln1Δ cln2Δ
GAL-CLN2 cln1Δ cln2Δ
cln1Δ cln2Δ sic1Δ

*cln1*Δ *cln2*Δ *cdh1*Δ
*GAL-CLN2 cln1*Δ *cln2*Δ *cdh1*Δ
*cln3*Δ
GAL-CLN3
*bck2*Δ
Multi-copy *BCK2*
*cln1*Δ *cln2*Δ *bck2*Δ
*cln3*Δ *bck2*Δ *GAL-CLN2 cln1*Δ *cln2*Δ
*cln1*Δ *cln2*Δ *cln3*Δ *GAL-CLN2*
*cln1*Δ *cln2*Δ *cln3*Δ *GAL-CLN3*
*cln1*Δ *cln2*Δ *cln3*Δ *sic1*Δ
*cln1*Δ *cln2*Δ *cln3*Δ *cdh1*Δ
*cln1*Δ *cln2*Δ *cln3*Δ multi-copy *CLB5*
*cln1*Δ *cln2*Δ *cln3*Δ *GAL-CLB5*
*cln1*Δ *cln2*Δ *cln3*Δ multi-copy *BCK2*
*sic1*Δ
GAL-SIC1
*GAL-SIC1 GAL-CLN2 cln1*Δ *cln2*Δ
*GAL-SIC1 GAL-CLN2 cln1*Δ *cln2*Δ *cdh1*Δ
*sic1*Δ *cdh1*Δ *GALL-CDC20*
*cdh1*Δ
*cdc6*Δ*2-49*
*sic1*Δ *cdc6*Δ*2-49*
GAL-CLB2
Multicopy *GAL-CLB2*
*CLB2-db*Δ
*CLB2-db*Δ *GAL-SIC1*
*CLB2-db*Δ *clb5*Δ
*CLB2-db*Δ *clb5*Δ in galactose
*clb5*Δ *clb6*Δ
GAL-CLB5
*GAL-CLB5 cdh1*Δ
*CLB5-db*Δ
*tem1*Δ
GAL-TEM1
tem1-ts GAL-CDC15
*tem1*Δ *net1-ts*
tem1-ts multicopy *CDC14*
*cdc15*Δ
Multicopy *CDC15*
*cdc15*Δ *net1-ts*
*cdc15-ts*multicopy *CDC14*
GAL-NET1
cdc14-ts
GAL-NET1 GAL-CDC14
TAB6-1

TAB6-1 cdc15Δ
TAB6-1 CLB1 clb2Δ
mad2Δ
bub2Δ
APC-A
APC-A cdh1Δ
APC-A cdh1Δ in galactose
APC-A cdh1Δ multicopy *SIC1*
APC-A cdh1Δ GAL-SIC1
APC-A cdh1Δ multicopy *CDC6*
APC-A cdh1Δ GAL-CDC6
APC-A cdh1Δ multicopy *CDC20*
swi5Δ
APC-A sic1Δ
APC-A GAL-CLB2

Table 7. The mutants that had violated constraints, optimized parameters.

cdh1Δ cdc6Δ2-49	C8-N10
GAL-CLB2 sic1Δ	N12, N16
CLB2-dbΔ in galactose	N16
CLB2-dbΔ multicopy *SIC1*	C8-N22, C8-N25
GAL-CLB2-dbΔ	N12
GAL-ESP1 cdc20-ts	N5, N12
net1-ts	C1-N8
cdc14-ts GAL-SIC1	C1-N20, C2-N13, C2-N18, C2-N22, C3-N13, C3-N18, C3-N22, C4-N13, C4-N18, C4-N22, C5-N13, C5-N18, C5-N22, C6-N13, C6-N18, C6-N22, C7-N13, C7-N18, C7-N22, C8-N13, C8-N18, C8-N22
mad2Δ bub2Δ	C1-N16
sic1Δ cdc6Δ2-49 cdh1Δ GALL-CDC20	C1-N16, C1-N18

A parallel direct search algorithm [18] applied to the penalty function formulation used over 10,000 CPU hours on 400 processors of a 2200 processor supercomputer (1100 Apple G5 Xserve nodes, Infiniband network). The full inequality formulation for all mutants would have approximately 11,500 constraints (sum of constraints per phenotype accounting for multiple cycles) and 4000 variables (almost all of them times $t_j^{\{i\}}$). One ODE solution and the transforms necessary to match the ODE solution to experimental data take about 17 seconds on a 2.3 GHz G5 processor, so the need for parallel supercomputing is clear. While the nonlinear inequality formulation is definitely large scale, it is well within the range of problems being solved in industry. This work demonstrates the reasonableness of the inequality approach to parameter estimation for cell cycle modeling. Mechanically assembling all 11,500 constraints and then solving the feasibility problem will be a major undertaking requiring several man-years of effort and parallel supercomputing to evaluate the constraints, but is surely doable.

6. Conclusions

A long-range goal of system biology is to develop efficient tools for fitting quantitative models to available types of experimental data. The cell cycle control system in budding yeast is a representative example of this general problem. The model consists of 36 variable protein levels (described by ordinary differential equations) and 143 kinetic parameters that need to be estimated from the data. The data consists of a hodge-podge of qualitative observations and quantitative measurements on wild-type and mutant cells. The challenge is to determine if there exists a feasible set of kinetic parameters for which the ODEs are consistent with the qualitative phenotypes of the collection of mutants.

This problem is formulated as a system of nonlinear inequalities that is satisfied if and only if the model matches all experimental data. The results in Tables 4–7 show that this formulation can accurately compare the simulation results with the experimental data. Using the smooth constraints instead of the discontinuous objective function will make it possible to use mathematical programming algorithms that assume smooth functions.

Note that no 143-dimensional parameter vector is known that will satisfy all the constraints because some experimental data may be wrong, the ODE model may be incomplete, and/or the biologically correct parameter vector may not yet have been found. Regardless of the source of the discrepancy, this inequality formulation provides a qualitatively different approach from the discontinuous penalty function for biologists to use in their quest for a validated cell cycle model.

7. References

[1] Allen, N.A. “Computational Software for Building Biochemical Reaction Network Models with Differential Equations.” Ph.D. thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2005.

[2] Allen, N.A., Calzone, L., Chen, K.C., Ciliberto, A., Ramakrishnan, N., Shaffer, C.A., Sible, J.C., Tyson, J.J., Vass, M.T., Watson, L.T., and Zwolak, J.W. “Modeling regulatory networks at Virginia Tech.” *OMICS*, Vol. 7, pp 285–299, 2003.

[3] Allen, N.A., Chen, K.C., Tyson, J.J., Shaffer, C.A., and Watson, L.T. “Computer evaluation of network dynamics models with application to cell cycle control in budding yeast.” *IEE Systems Biology*, Vol. 153, pp 13–21, 2006.

[4] Allen, N.A., Shaffer, C.A., Ramakrishnan, N., Vass, M.T., and Watson, L.T. “Improving the development process for eukaryotic cell cycle models with a modeling support environment.” *Simulation*, Vol. 79, pp 674–688, 2003.

[5] Audet, C. and Dennis Jr., J.E., “Mesh adaptive direct search algorithms for constrained optimization.” *SIAM J. Optim.*, to appear.

[6] Boggs, P.T., Byrd, R.H., and Schnabel, R.B. “A stable and efficient algorithm for nonlinear orthogonal distance regression.” *SIAM J. Sci. Stat. Comput.*, Vol. 8, pp 1052–1078, 1987.

[7] Chen, K.C., Calzone, L., Csikasz-Nagy, A., Cross, F.R., Novak, B., and Tyson, J.J. “Integrative analysis of cell cycle control in budding yeast.” *Molecular Biology of the Cell*, Vol. 15, pp 3841–3862, 2004.

[8] Conn, A.R., Gould, N.I.M., and Toint, Ph.L. *Trust-Region Methods.* Philadelphia: SIAM, 2000.

[9] Coope, I.D. and Price, C.J. "Frame-based methods for unconstrained optimization." *Journal of Optimization Theory and Applications*, Vol. 107, pp 261–274, 2000.

[10] He, J., Watson, L.T., Ramakrishnan, N., Shaffer, C.A., Verstak, A., Jiang, J., Bae, K., and Tranter, W.H. "Dynamic data structures for a direct search algorithm." *Comput. Optim. Appl.*, Vol. 23, pp 5–25, 2002.

[11] He, J., Sosonkina, M., Shaffer, C.A., Tyson, J.J., Watson, L.T., and Zwolak, J.W. "A hierarchical parallel scheme for a global search algorithm." In *Proceedings of High Performance Computing Symposium 2004*, Soc. for Modeling and Simulation Internat., San Diego, CA, May, 2004.

[12] He, J., Sosonkina, M., Shaffer, C.A., Tyson, J.J., Watson, L.T., and Zwolak, J.W. "A hierarchical parallel scheme for global parameter estimation in systems biology." In *Proceedings of 18th Internat. Parallel & Distributed Processing Symp.*, IEEE Computer Soc., Los Alamitos, CA, 2004.

[13] He, J., Sosonkina, M., Watson, L.T., Verstak, A., and Zwolak, J.W. "Data-distributed parallelism with dynamic task allocation for a global search algorithm." In *Proceedings of High Performance Computing Symposium 2005*, Soc. for Modeling and Simulation Internat., San Diego, CA, 2005.

[14] Jones, D.R., Perttunen, C.D., and Stuckman, B.E. "Lipschitzian optimization without the Lipschitz constant." *J. Optim. Theory Appl.*, Vol. 79, No. 1, pp 157–181, 1993.

[15] Murray, A. and Hunt, T. *The Cell Cycle: an Introduction.* New York: Oxford University Press, 1993.

[16] Nasmyth, K. "At the heart of the budding yeast cell cycle." *Trends in Genetics*, Vol. 12, pp 405–412, 1996.

[17] Nurse, P. "A long twentieth century of the cell cycle and beyond." *Cell*, Vol. 100, pp 71–78, 2000.

[18] Panning, T.D., Watson, L.T., Allen, N.A., Chen, K.C., Shaffer, C.A., and Tyson, J.J. "Deterministic parallel global parameter estimation for a model of the budding yeast cell cycle." *Journal of Global Optimization*, to appear.

[19] Panning, T.D., Watson, L.T., Allen, N.A., Shaffer, C.A., and Tyson, J.J. "Deterministic global parameter estimation for a budding yeast model." In *Proceedings of 2006 Spring Simulation Multiconf., High Performance Computing Symp.*, San Diego, CA, 2006.

[20] Watson, L.T. and Baker, C.A. "A fully-distributed parallel global search algorithm." *Engineering Computations*, Vol. 18, No. 1/2, pp 514–549.

[21] Zwolak, J.W., Boggs, P.T., and Watson, L.T. "Algorithm XXX: ODRPACK95: a weighted orthogonal distance regression code with bound constraints." *ACM Transactions on Mathematical Software*, to appear.

[22] Zwolak, J.W., Tyson, J.J, and Watson, L.T. "Parameter estimation for a mathematical model of the cell cycle in frog eggs." *Journal of Computational Biology*, Vol. 12, pp 48–63, 2005.

[23] Zwolak, J.W., Tyson, J.J, and Watson, L.T. "Globally optimised parameters for a model of mitotic control in frog egg extracts." *IEE Systems Biology*, Vol. 152, pp 81–92, 2005.

8. Appendix A: Mutants

Mutants marked by an asterisk (*) have a phenotype that does not correspond to any of the constraint sets given in Section 3.

Index	Mutant name	Observed Phenotype
1.	Wild type in glucose	Viable, $G_1 = 35.2$
2.	Wild type in galactose	Viable, $G_1 = 109$
3.	*cln1Δ cln2Δ*	Viable, $M = 2w_m$
4.	*GAL-CLN2 cln1Δ cln2Δ*	Viable, $M = 0.5w_m$
5.	*cln1Δ cln2Δ sic1Δ*	Viable
6.	*cln1Δ cln2Δ cdh1Δ*	Viable
7.	*GAL-CLN2 cln1Δ cln2Δ cdh1Δ*	Viable, $M = 1.7w_m$
8.	*cln3Δ*	Viable, $M = 1.7w_m$
9.	*GAL-CLN3*	Viable, $M = 0.44w_m$
10.	*bck2Δ*	Viable, $M = 1.4w_m$
11.	Multi-copy *BCK2*	Viable, $M = 0.8w_m$
12.	*cln1Δ cln2Δ bck2Δ*	Viable, $M = 1.7w_m$
13.	*cln3Δ bck2Δ*	G1 arrest
14.	*cln3Δ bck2Δ GAL-CLN2 cln1Δ cln2Δ*	Viable
15.	*cln3Δ bck2Δ* multi-copy *CLN2*	G1 arrest
16.	*cln3Δ bck2Δ GAL-CLB5*	Inviable
17.	*cln3Δ bck2Δ sic1Δ*	Inviable
18.	*cln1Δ cln2Δ cln3Δ*	G1 arrest
19.	*cln1Δ cln2Δ cln3Δ GAL-CLN2*	Viable
20.	*cln1Δ cln2Δ cln3Δ GAL-CLN3*	Viable
21.	*cln1Δ cln2Δ cln3Δ sic1Δ*	Viable, $G_1 = 10$, $M = 3.5w_m$
22.	*cln1Δ cln2Δ cln3Δ cdh1Δ*	Telophase arrest
23.	*cln1Δ cln2Δ cln3Δ* multi-copy *CLB5*	Viable
24.	*cln1Δ cln2Δ cln3Δ GAL-CLB5*	Viable
25.	*cln1Δ cln2Δ cln3Δ* multi-copy *BCK2*	Viable
26.	*cln1Δ cln2Δ cln3Δ GAL-CLB2*	G1 arrest
27.	*cln1Δ cln2Δ cln3Δ apc-ts*	Metaphase arrest
28.	*sic1Δ*	Viable, $G_1 = 15$, $M = w_m$
29.	*GAL-SIC1*	Viable, $G_1 = 135$, $M = 2w_m$
30.	*GAL-SIC1-dbΔ*	G1 arrest
31.	*GAL-SIC1 cln1Δ cln2Δ*	G1 arrest
32.	*GAL-SIC1 cln1Δ cln2Δ cdh1Δ*	G1 arrest
33.	*GAL-SIC1 GAL-CLN2 cln1Δ cln2Δ*	Viable
34.	*GAL-SIC1 GAL-CLN2 cln1Δ cln2Δ cdh1Δ*	Viable
35.	*sic1Δ cdh1Δ*	Reductive mitosis in second cycle*
36.	*sic1Δ cdh1Δ GALL-CDC20*	Viable
37.	*cdh1Δ*	Viable, $M = 0.6w_m$
38.	Cdh1 constitutively active	G2 arrest
39.	*cdc6Δ2-49*	Viable
40.	*sic1Δ cdc6Δ2-49*	Viable
41.	*cdh1Δ cdc6Δ2-49*	Viable, $G_1 = 20$, $M = 2w_m$

42.	*clb1Δ clb2Δ*	G2 arrest
43.	*GAL-CLB2*	Viable
44.	Multicopy *GAL-CLB2*	Telophase arrest
45.	*GAL-CLB2 sic1Δ*	Telophase arrest
46.	*GAL-CLB2 cdh1Δ*	Inviable
47.	*CLB2-dbΔ*	Telophase arrest
48.	*CLB2-dbΔ* in galactose	Telophase arrest
49.	*CLB2-dbΔ* multicopy *SIC1*	Viable
50.	*CLB2-dbΔ GAL-SIC1*	Viable
51.	*CLB2-dbΔ clb5Δ*	Telophase arrest
52.	*CLB2-dbΔ clb5Δ* in galactose	Viable
53.	*GAL-CLB2-dbΔ*	Telophase arrest
54.	*clb5Δ clb6Δ*	Viable, $G_1 = 65$
55.	*cln1Δ cln2Δ clb5Δ clb6Δ*	G1 arrest
56.	*GAL-CLB5*	Viable
57.	*GAL-CLB5 sic1Δ*	Inviable
58.	*GAL-CLB5 cdh1Δ*	Inviable after many divisions*
59.	*CLB5-dbΔ*	Viable
60.	*CLB5-dbΔ sic1Δ*	Semi-lethal*
61.	*CLB5-dbΔ pds1Δ*	Viable
62.	*CLB5-dbΔ pds1Δ cdc20Δ*	Telophase arrest
63.	*GAL-CLB5-dbΔ*	Inviable
64.	*cdc20-ts*	Metaphase arrest
65.	*cdc20Δ clb5Δ*	Metaphase arrest
66.	*cdc20Δ pds1Δ*	Telophase arrest
67.	*cdc20Δ pds1Δ clb5Δ*	Viable
68.	*GAL-CDC20*	Premature chromosome separation*
69.	*cdc20-ts mad2Δ*	Metaphase arrest
70.	*cdc20-ts bub2Δ*	Metaphase arrest
71.	*pds1Δ*	Viable
72.	*esp1-ts*	Chromosome separation failure*
73.	*PDS1-dbΔ*	Chromosome separation failure*
74.	*GAL-PDS1-dbΔ*	Chromosome separation failure*
75.	*GAL-PDS1-dbΔ esp1-ts*	Chromosome separation failure*
76.	*GAL-ESP1 cdc20-ts*	Telophase arrest
77.	*tem1Δ*	Telophase arrest
78.	*GAL-TEM1*	Viable
79.	*tem1-ts GAL-CDC15*	Viable
80.	*tem1Δ net1-ts*	Viable
81.	*tem1-ts* multicopy *CDC14*	Viable
82.	*cdc15Δ*	Telophase arrest
83.	Multicopy *CDC15*	Viable
84.	*cdc15-ts* multicopy *TEM1*	Inviable
85.	*cdc15Δ net1-ts*	Viable
86.	*cdc15-ts* multicopy *CDC14*	Viable
87.	*net1-ts*	Viable, $G_1 = 50$

88.	*GAL-NET1*	Telophase arrest
89.	*cdc14-ts*	Telophase arrest
90.	*GAL-CDC14*	G1 arrest
91.	*GAL-NET1 GAL-CDC14*	Viable
92.	*net1*Δ *cdc20-ts*	Reductive mitosis*
93.	*cdc14-ts GAL-SIC1*	Weakly viable*
94.	*TAB6-1*	Viable
95.	*TAB6-1 cdc15*Δ	Viable
96.	*TAB6-1 clb5*Δ *clb6*Δ	G1 arrest
97.	*TAB6-1 CLB1 clb2*Δ	Viable
98.	*mad2*Δ	Viable, $G_1 = 35$, $M = w_m$
99.	*bub2*Δ	Viable, $G_1 = 35$, $M = w_m$
100.	*mad2*Δ *bub2*Δ	Viable
101.	*APC-A*	Viable, $G_1 = 20$, $M = 1.5w_m$
102.	*APC-A cdh1*Δ	Telophase arrest
103.	*APC-A cdh1*Δ in galactose	Viable
104.	*APC-A cdh1*Δ multicopy *SIC1*	Viable
105.	*APC-A cdh1*Δ *GAL-SIC1*	Viable
106.	*APC-A cdh1*Δ multicopy *CDC6*	Viable
107.	*APC-A cdh1*Δ *GAL-CDC6*	Viable
108.	*APC-A cdh1*Δ multicopy *CDC20*	Viable
109.	*swi5*Δ	Viable, $G_1 = 20$
110.	*sic1*Δ *cdc6*Δ*2-49 cdh1*Δ	G2 arrest in second cycle
111.	*sic1*Δ *cdc6*Δ*2-49 cdh1*Δ *GALL-CDC20*	Viable
112.	*APC-A cdh1*Δ *clb5*Δ	Inviable
113.	*APC-A cdh1*Δ *pds1*Δ	Inviable
114.	*APC-A sic1*Δ	Viable
115.	*APC-A GAL-CLB2*	Telophase arrest