

Evolving Petri Nets to Represent Metabolic Pathways

Jeremiah Nummela
Department of Computer Science
St. Cloud State University
St. Cloud, MN 56301 USA
nuje0301@stcloudstate.edu

Bryant A. Julstrom
Department of Computer Science
St. Cloud State University
St. Cloud, MN 56301 USA
julstrom@csci.stcloudstate.edu

ABSTRACT

Given concentrations of metabolites over a sequence of time steps, the metabolic pathway prediction problem seeks a set of reactions and rate constants for them that could yield the concentration-time data. Such metabolic pathways can be modeled with Petri nets: bipartite graphs whose nodes are called places and transitions and in which tokens move from place to place through the transitions. Thus the pathway prediction problem can be addressed by searching a space of Petri nets, and such a search can be undertaken evolutionarily.

Here, a genetic algorithm performs such a search. The GA seeks only the net's structure; a hill-climbing step applied as part of evaluation approximates parameters associated with the net's transitions. On one contrived problem instance, the GA sometimes identifies the pathway used to generate the given data, but on a second contrived instance, apparently no harder, it fails. On an instance drawn from real biology—the pathway for phospholipid synthesis—the genetic algorithm identifies a Petri net whose pathway is very similar, but not identical to, the real one. In all three cases, the GA develops Petri nets that represent pathways that closely reproduce the target concentration-time data.

Categories and Subject Descriptors

J.3 [Computer Applications]: Life and Medical Sciences; I.2.6 [Computing Methodologies]: Artificial Intelligence—Learning; G.2.1 [Mathematics of Computing]: Discrete Mathematics—Combinatorics

General Terms

Algorithms

Keywords

Petri nets, metabolic pathways, evolutionary search, genetic algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

1. INTRODUCTION

In an inverse problem, the output of a system is known, and the task is to identify a system of the appropriate kind—its structure and parameters—that could have produced the output. Many inverse problems arise in biology; one of them is the problem of predicting metabolic pathways: Given concentrations of metabolites over a sequence of time steps, identify the metabolic pathway—the reactions in which the metabolites participate and the rate constants of the reactions—that produced the concentration-time data.

This is a problem in chemical kinetics, which describes the progress of chemical reactions. Chemical kinetics in general, and metabolic pathways in particular, are traditionally modeled with systems of differential equations that describe the rates of change of metabolites as functions of their concentrations [11], as Section 2 summarizes.

Petri nets offer another model of metabolic pathways. A Petri net is a bipartite graph whose nodes are called places and transitions. Places can represent metabolites, transitions can represent reactions, and values associated with transitions can represent rate constants, which describe the rates at which reactions proceed. Section 3 describes Petri nets and their relation to metabolic pathways.

One approach to the problem of predicting a metabolic pathway from concentration-time data, then, is to search a space of Petri nets for one that represents a pathway that conforms to the data. Such a search can be conducted evolutionarily. We present, and Section 4 describes in detail, a genetic algorithm that conducts such a search. Its genotypes encode the transitions of Petri nets that represent candidate metabolic pathways; the rate constants for the represented reactions are approximated in a hill-climbing step that is part of evaluation. Thus the GA searches a space of Petri net structures. It performs adequately on one contrived problem, not so well on a second, and comes close to exactly identifying the metabolic pathway for phospholipid synthesis, as Section 5 describes. On all three problems, the GA develops Petri nets that represent pathways that closely replicate the target concentration-time data.

Other researchers have used evolutionary algorithms to search spaces of Petri nets. Tohme *et al.* [19] applied evolutionary search to identify Petri nets that represented good solutions to the periodic job-shop scheduling problem, which seeks the minimum cycle time in a job-shop system. Saitou, Malpathak, and Qvam [18] represented manufacturing systems as Petri nets and used a genetic algorithm to search for robust flexible systems.

Yim and Bourdeaud’huy [22] described a genetic algorithm that sought Petri net models of controllers for discrete events systems. Moore and Hahn [12] evolved grammars in Backus-Naur form that described Petri nets and noted their applications to problems in genetics and biochemistry. Mauch [10] considered evolving Petri nets as a form of genetic programming and searched for nets to solve problems of Boolean function learning and classification.

Yen *et al.* [21] augmented a genetic algorithm with a stochastic extension of the simplex method and applied the resulting hybrid algorithm to search for good rate constants for a known metabolic pathway, a problem addressed with traditional techniques by Mendes and Kell [11]. In tests on this problem and two problems of function optimization, they reported better results with their technique than with five alternative techniques.

Kitagawa and Iba [7] used a genetic algorithm to search for Petri nets that represented metabolic pathways for given concentration-time data. Their GA’s chromosomes encoded not only Petri nets but also rate constants for the reactions the nets represented and thus evolved rate constants as well as Petri nets’ structures. The present genetic algorithm searches a space of Petri nets for one that represents well a metabolic pathway, but it develops each represented pathway’s rate constants in a hill-climbing step that is part of evaluation.

Petri nets are formal computing machines, and search in spaces of such objects is generally regarded as difficult because changing one feature of such a machine generally changes the role and meaning of other features of it. The search in a space of Petri nets described here is thus related to evolutionary search in spaces of other formal machines such as deterministic finite automata [4] [6] [9] [13], push-down automata [8], and Turing machines [5] [20].

2. CHEMICAL KINETICS

Chemical kinetics describes the changes in concentrations of reactants and products as reactions proceed. The rates of change of these concentrations over time describe a reaction, and they are traditionally modeled using differential equations. For a single mechanistic step—that is, a reaction involving the traversal of a single potential energy barrier—the reaction rate is proportional to the concentrations of the reactants:



where the R_i and P_j are reactants and products, respectively, and the a_i and b_j are their stoichiometric constants. The rates of change can be expressed as

$$\frac{d[R_i]}{dt} = -a_i k [R_1][R_2] \cdots [R_m]$$

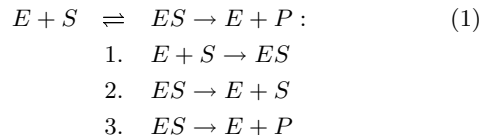
for reactants and as

$$\frac{d[P_i]}{dt} = b_i k [R_1][R_2] \cdots [R_m]$$

for products, where k is a constant characteristic of the reaction and $[R_i]$ is the concentration of R_i .

An enzyme-catalyzed reaction consists of several steps that typically include the reversible formation of an enzyme-substrate complex followed by the conversion of the sub-

strate and the release of the product:



where E is the enzyme, S the substrate, and ES the complex. Metabolic pathways are networks of enzyme-catalyzed reactions.

A simplified expression for the rate of the reactions (1) can be obtained by applying the steady-state approximation to the concentration of ES or, equivalently, by assuming that the reversible formation of the enzyme-substrate complex is always in an equilibrium state. Either assumption results in the Michaelis-Menton expression for the overall reaction rate:

$$\frac{d[P]}{dt} = \frac{k_2[E]_t[S]}{K_m + [S]},$$

where $[E]$ is the total concentration of the enzyme in all forms (free and bound in the enzyme-substrate complex), and K_m is the equilibrium constant for the reversible formation of the complex.

When K_m is much larger than $[S]$, this expression can be further simplified by neglecting $[S]$ in the denominator, so that

$$\frac{d[P]}{dt} = \frac{k_2[E]_t[S]}{K_m}.$$

This is called the pseudo-first-order conditions equation, because the rate equation is now first-order with respect to the substrate concentration $[S]$.

In the reaction model that we investigate here, k_2 , k_m , and $[E]_t$ are all constant over the course of the reaction and can be combined into a single constant k that is characteristic of the reaction:

$$\frac{d[P]}{dt} = \frac{-d[S]}{dt} = k[S]. \quad (2)$$

An equation like (2) describes each reaction in the metabolic pathway. Products of a reaction do not affect the rate of the reaction, but may be reactants of other reactions in the pathway.

3. PETRI NETS

Petri nets [14] [15] are formal automata developed to model concurrent processes. A Petri net consists of a finite set P of *places* and a finite set T of *transitions*. Associated with each transition in T is a set of *input places* and a set of *output places*. Each place holds zero or more *tokens*. A transition is *enabled* if each of its input places holds at least one token, and an enabled transition may *fire*. Firing a transition removes one token from each of its input places and adds one token to each of its output places. Petri nets are nondeterministic; any enabled transition may fire.

Directed bipartite graphs can represent Petri nets. One set of nodes represents the places P , the other the transitions T , and directed edges connect input places to transitions and transitions to output places. When these graphs are drawn, circles represent places and bars represent transitions. Dots within places represent the tokens they hold. Figure 1 illustrates a Petri net with five places and three

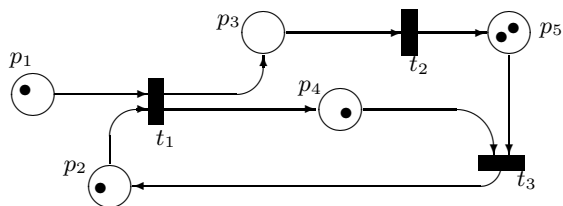


Figure 1: A Petri net with five places and three transitions. Transitions t_1 and t_3 are enabled; firing transition t_3 will remove a token from p_4 and one from p_5 and add a token to place p_2 .

transitions. Transitions t_1 and t_3 are enabled; firing, for example, t_3 removes a token from place p_4 and one from place p_5 and adds a token to place p_2 .

Petri nets are elegantly suited to represent networks of chemical reactions. Places represent reactants and products, and transitions represent reactions. A reaction’s reactants are represented by its transition’s input places, and its products are represented by its output places. A place’s tokens represent the available amount of the corresponding substance. Firing a transition represents the occurrence of its reaction, which consumes reactants (removes tokens from its input places) and yields products (adds tokens to its output places). Weights assigned to a Petri net’s edges can indicate numbers of tokens consumed and produced when a transition fires and thus represent amounts of reactants consumed and products produced by the corresponding reaction.

Two extensions of the Petri net formalism have also been used to represent chemical kinetics. In *continuous and hybrid Petri nets* [2], places may hold non-integer numbers of tokens; these nets can represent arbitrary amounts of reactants and products. In *functional Petri nets* [3], the weights assigned to edges may be functions of the numbers of tokens in the nets’ places; these nets can represent reactions’ rate equations. Because the present GA evolves only Petri net structures and develops rate constants for the pathways they represent through hill-climbing, these extensions are not necessary here.

4. EVOLVING PETRI NETS

Given concentrations of reactants and products over time in a metabolic pathway, a genetic algorithm searches a space of Petri nets for one that represents a pathway that, with appropriate rate constants, accurately reproduces the concentration-time data. This section describes the GA.

4.1 Representation

Distinct integers represent the metabolites listed in the given concentration-time data; that is, in the target metabolic pathway. In the genotype that represents a candidate pathway, each individual reaction is represented by a list of its reactants and products; the genotype is then a list of all the reactions in the pathway, corresponding to all the transitions in the genotype’s Petri net. This representation is similar to that used by Kitagawa and Iba [7], but it does not include reactions’ rate constants; that is, the values associated with each transition that indicate how quickly tokens from its input places are consumed and tokens in its output places are generated. Those values are assigned as part of evaluation, as the following section describes.

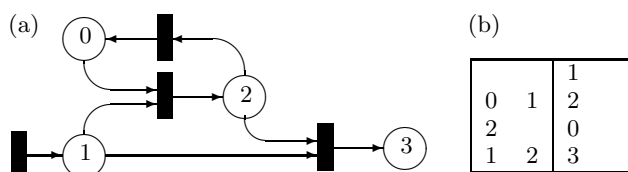


Figure 2: A Petri net (a) that represents a metabolic pathway and a genotype (b) that in turn represents the Petri net. The numbers labeling the Petri net’s places indicate reactants and products.

We assume that no reaction has more than two reactants or products; that is, that no transition has more than two edges in or two edges out. This is a biologically reasonable restriction. Figure 2 shows a Petri net that represent the Petri net. Numbers in the places indicate metabolites.

To generate a random genotype, the size of each set of reactants and products is set to zero with probability 0.1, to one with probability 0.4, and to two with probability 0.5. Metabolites are chosen at random to fill the sets.

The number of reactions and the numbers of reactants and products of each reaction are not fixed, so genotypes may be of different sizes.

4.2 Fitness Evaluation

To evaluate a candidate genotype, the metabolic pathway it represents is simulated, starting from the given metabolite concentrations at time $t = 0$. Reactions are assumed to conform to the Michaelis-Menton kinetic expression under pseudo-first-order conditions (2). The set of differential equations that the pathway implies under this assumption is integrated numerically using a fourth-order Runge-Kutta method [16]. The genotype’s fitness F is its deviation from the given concentration-time data, computed according to this formula:

$$F = \sum \frac{|c_{mi} - c_{mi0}|}{n_m n_p} + 0.1 \cdot n_r,$$

where c_{mi} is the computed concentration of the m^{th} metabolite at time step i ; c_{mi0} is the corresponding target concentration; n_m is the number of metabolites; n_p is the number of time steps; and n_r is the number of reactions. The sum is taken over all the metabolites at every time step. It measures the simulation’s similarity to the given data. The term involving n_r penalizes Petri nets that have more transitions; that is, that represent more-complex pathways. Of course, we seek to minimize fitness values.

As noted above, genotypes do not include rate constants for the reactions of the pathways the genotypes represent. Suitable rate constants are approximated by a stochastic hill-climbing step that precedes the final evaluation of a genotype.

In particular, the rate constants are initialized with random integers on the interval [1, 19]. These values are multiplied by 10^{-1} for bimolecular reactions, 10^1 for unimolecular reactions, and 10^2 if a transition represents the appearance of a metabolite due to diffusion. The resulting pathway is evaluated, then the rate constants are randomly modified and the simulation is repeated. If the new values yield better fitness than the old, the new values replace them. This

process continues for a fixed number of steps. The fitness of the simulation with the best rate constants this process discovers is returned as the fitness of the genotype.

This hill-climbing yields good results in tests. It is more computationally economical than standard methods of optimizing rate constants [11] [21], and it is more appropriate for discrete rate constant values than is steepest descent. Finally, there is no guarantee that the rate constants used to evaluate a candidate genotype are optimal for the pathway the genotype represents. Thus evaluation is noisy.

4.3 Variation Operators

Novel genotypes are generated by recombining two existing genotypes or mutating one. Recombination copies into its offspring all reactions that are common to both parents, then adds each remaining parental reaction with probability 0.5. This is similar to recombination operators for subset encodings such as Radcliffe’s Random Respectful Recombination (RRR) [17]. Mutation reinitializes each set of reactants and products with probability 0.5. Note that this is a very aggressive mutation operator.

4.4 A Genetic Algorithm

The representation and operators described in the previous sections were implemented in a generational genetic algorithm for the problem of identifying a likely metabolic pathway from concentration-time data. The algorithm initializes its population with random genotypes as described in Section 4.1 and chooses genotypes to be parents in tournaments with replacement. It applies recombination and mutation independently; each offspring genotype is generated by one or the other, never both. The GA is 1-elitist, preserving the best genotype of the current generation unchanged into the next, and it runs through a fixed number of generations.

In the tests that the next section describes, the GA’s population contained 100 genotypes. Evaluation executed 500 steps of hill-climbing to identify rate constants. The size of selection tournaments was two, and half the offspring in each new generation were produced by recombination, the other half by mutation. The GA ran through 100 generations.

5. TESTS

The genetic algorithm was tested on three instances of the metabolic pathway problem, two contrived and one drawn from real biology. Each instance involved six metabolites and five reactions.

Figure 3(a) shows the concentration-time data for the first contrived instance, in which only one metabolite is present at time $t = 0.0$ and the remaining metabolites in turn appear and then diminish. The exception is the sixth metabolite, whose concentration grows throughout the process. The GA was run 50 independent times on these data. These runs often yielded Petri nets whose pathways closely reproduced the concentration-time data, and two successfully identified the Petri net that represents the pathway from which the data were derived. Figure 3(b) shows this Petri net; in its pathway, the reactions form a chain in which the first metabolite yields the second, the second yields the third, and so on, and all the metabolites except the last are eventually exhausted.

Figure 4(a) shows the concentration-time data for the second contrived instance, in which three metabolites are ini-

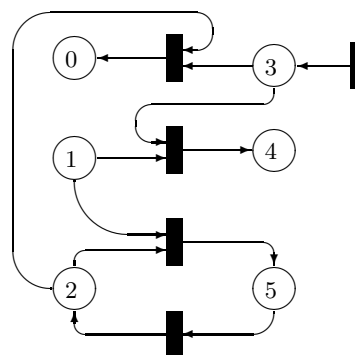


Figure 6: The most-fit Petri net identified by the GA on the phospholipid synthesis concentration-time data. It includes one extraneous transition, representing a reverse reaction that converts glycerol and fatty acid to diacyl-glycerol.

tially present and are consumed while the remaining three are initially absent, then appear in increasing concentrations. Again, the GA was run 50 independent times on the data and again the GA often found Petri nets whose pathways closely reproduced the concentration-time data, but in this case none of the runs successfully identified the target pathway, shown in Figure 4(b). Three metabolites are reactants only, the inputs to reactions that produce the remaining three metabolites, which are products only. Figure 4(c) shows the Petri net with best fitness identified by the GA. It includes extraneous reactions that do not appear in the target net.

The third instance is drawn from nature; it is the metabolic pathway for phospholipid synthesis, and it is shown in Figure 5(a). Kitagawa and Iba [7] also addressed this instance; the details of this pathway and its rate constants are well known. Concentration-time data were generated by simulating the pathway; Figure 5(b) shows the results of this simulation; that is, the input to the GA.

The GA was run 50 independent times on these data. The target pathway has five reactions, so the best possible fitness that a genotype could achieve was $0.1 \cdot 5 = 0.5$. The GA never reached this value, though it performed better on this instance than on the second contrived instance. The average fitness of the best genotypes in the GA’s initial populations was 7.46, and the average of its best final fitnesses was 1.21.

The best Petri net identified in all the runs of the GA had fitness 0.71. Concentration-time data generated by simulating the pathway this Petri net represents closely matched the input data and the results reported by Kitagawa and Iba [7].

Figure 6 shows this Petri net, which represents the reactions of the actual pathway, and in addition indicates a reverse reaction that converts glycerol and fatty acid to diacyl-glycerol. This extra reaction damages the fit between the represented pathway’s concentration-time values and the given data only slightly, but its existence adds 0.1 to its genotype’s fitness value. The GA never succeeded in reproducing the known pathway exactly, though the concentration-time data for the best pathway found is quantitatively very similar to that of the target pathway.

Why is the GA not more effective on the genuine biolog-

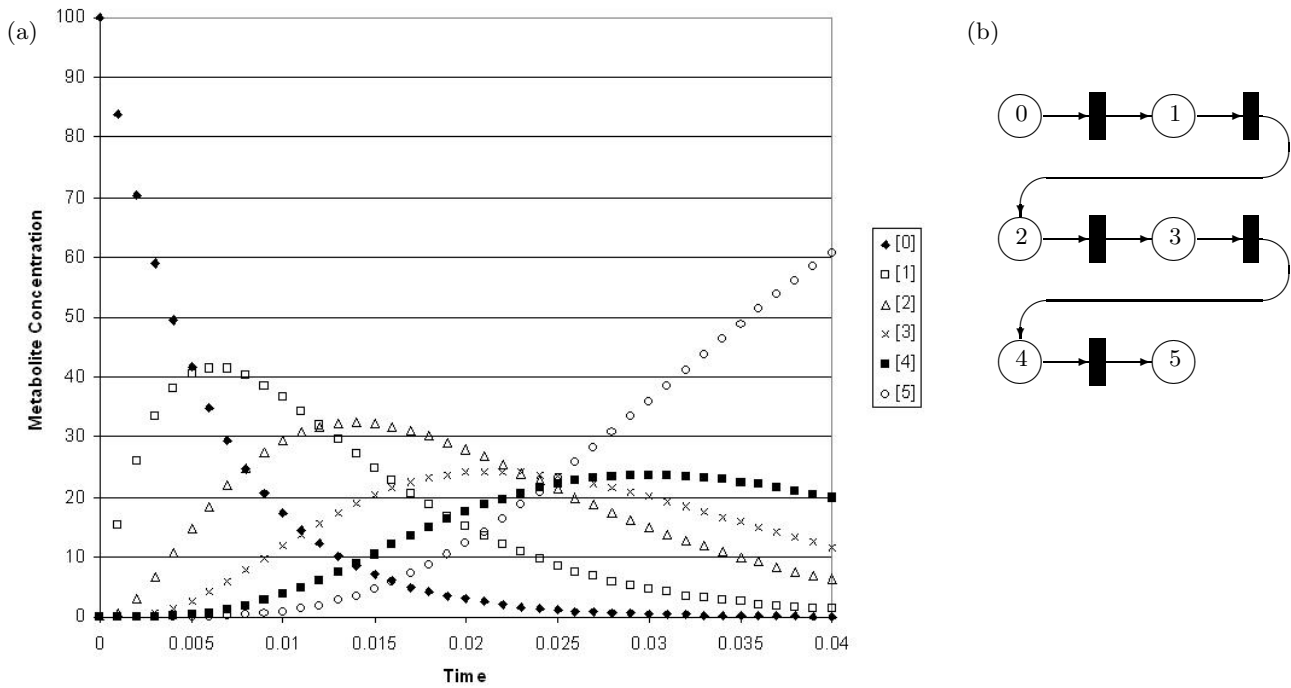


Figure 3: The concentration-time data (a) for the first contrived instance, whose solution (b) is a chain of places and transitions.

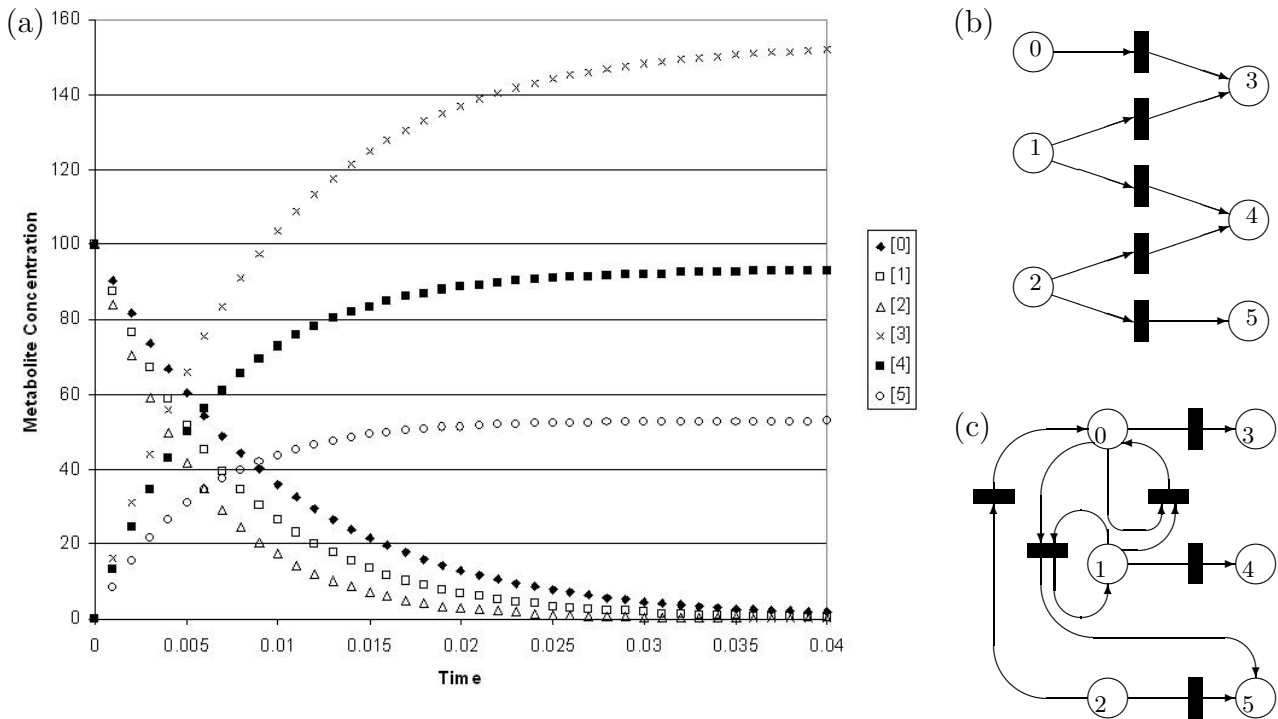


Figure 4: The concentration-time data (a) for the second contrived problem instance, the Petri net (b) that represents the pathway from which the data were developed, and the best Petri net (c) that the GA identified for these data.

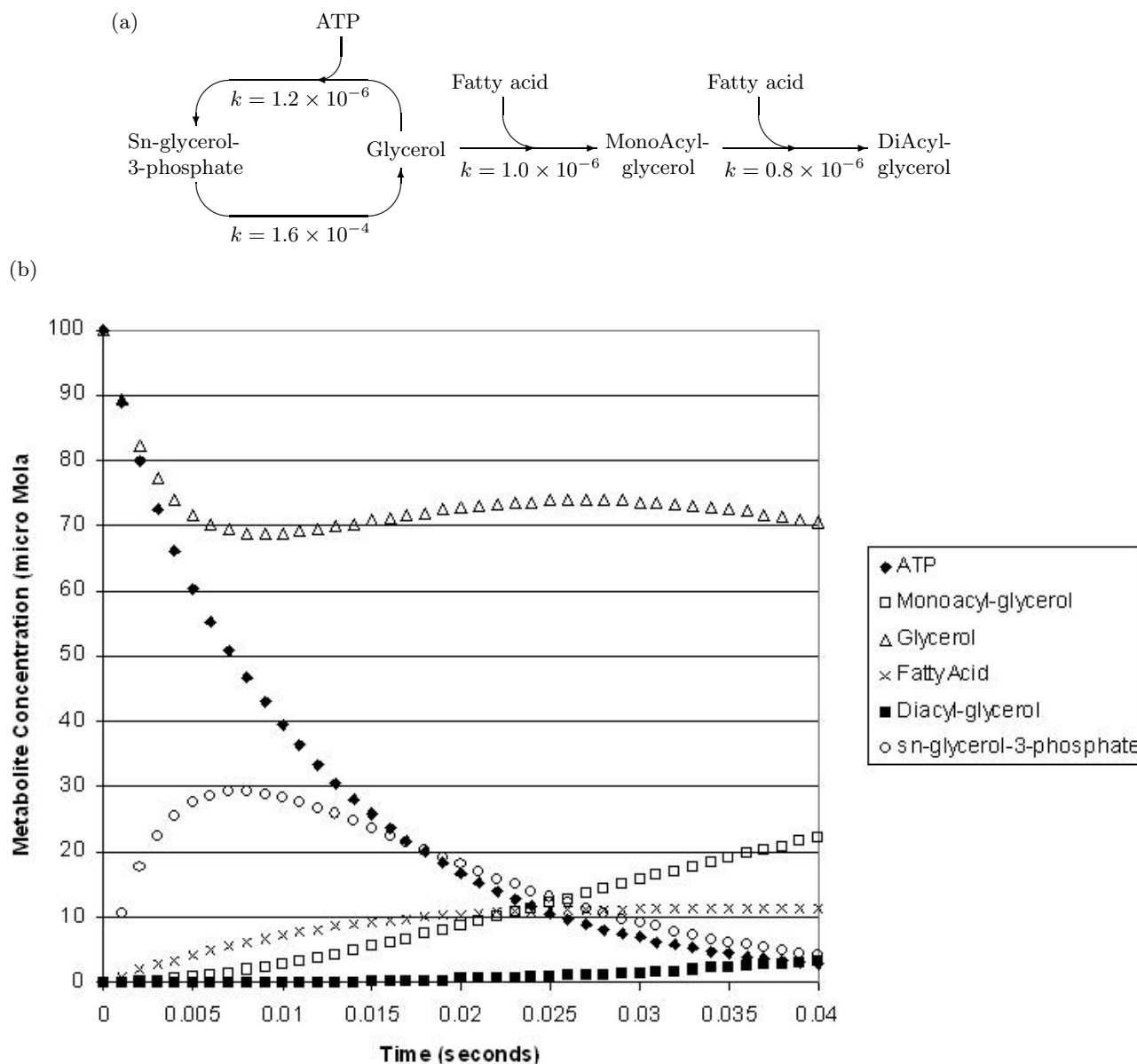


Figure 5: The phospholipid synthesis pathway (a) and concentration-time data (b) for it. These data form the third test instance for the genetic algorithm.

ical problem instance? It appears to have difficulty identifying reactions whose reactants have zero initial concentrations unless reactions to generate those reactants are already present. The reactions most often reproduced in the GA's best pathways convert glycerol to diacyl-glycerol or sn-glycerol-3-phosphate; the reactants of these reactions have positive initial concentrations.

More generally, as with representations of other automata such as finite state machines, push-down automata, and Turing machines, encodings of Petri nets are highly epistatic. The value or occurrence of one symbol in a genotype often strongly affects the meaning of other symbols. In particular, the presence or absence of a metabolite as a product of one

reaction enables or prevents any reaction that requires that metabolite. More sophisticated, possibly heuristic, recombination and mutation operators might partly overcome this difficulty and provide more effective evolutionary search. In particular, mutation could favor the introduction of reactions whose products are the reactants of other reactions already represented in the parental genotype.

6. CONCLUSION

Given concentration-time data for a metabolic pathway, the inverse problem seeks a pathway and its parameters that generate the given values. Petri nets can represent metabolic

pathways; places represent metabolites and transitions represent reactions. Thus one approach to the inverse problem seeks a Petri net that represents a pathway that yields a close approximation to the given concentration-time data.

A genetic algorithm for this problem encodes the structure of Petri nets, thus of metabolic pathways, as lists of inputs and outputs of transitions, thus of reactants and products of reactions. A stochastic hill-climber that precedes evaluation develops rate constants for the represented reactions. The GA applies both recombination and mutation to generate novel genotypes; its mutation operator is particularly aggressive.

The GA achieves good results on one contrived problem instance with a simple structure and straightforward solution, but it is stymied on a second contrived instance that appears no more difficult. On an instance drawn from real biology, repeated trials of the GA fail to identify the pathway that is known to implement phospholipid synthesis, though some trials come close. On all three test instances, however, the GA identified pathways that closely reproduce the target concentration-time data, indicating the general utility of the approach. Possible avenues for improving the GA's performance include more effective search for good rate constants and modified genetic operators.

More generally, this method might be adapted to investigating the feasibility of particular pathways. Given target concentration-time data, might there exist a plausible sequence of reactions to generate it?

7. REFERENCES

- [1] Erick Cantú-Paz, editor. *Genetic and Evolutionary Computation – GECCO 2003*, volume 2724 of *LNCS*, Berlin, 2003. Springer-Verlag. Part II.
- [2] René David and Hassane Alla. On hybrid Petri nets. *Discrete Event Dynamic Systems: Theory and Applications*, 11(1–2):9–40, 2001.
- [3] Atsushi Doi, Sachie Fujita, Hiroshi Matsuno, Masao Nagasaki, and Satoru Miyano. Constructing biological pathway models with hybrid functional Petri nets. *In Silico Biology*, 4, 2004.
- [4] Philip Hingston. A genetic algorithm for regular inference. In Lee Spector and et al., editors, *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, pages 1299–1306, San Francisco, CA, 2001. Morgan Kaufman.
- [5] Terry Jones and Gregory J. E. Rawlins. Reverse hillclimbing, genetic algorithms and the Busy Beaver problem. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 70–75, San Mateo, CA, 1993. Morgan Kaufmann Publishers.
- [6] Bryant A. Julstrom. Evolutionary discovery of DFA size and structure. In K. M. George, Janice H. Carroll, Dave Oppenheim, and Jim Hightower, editors, *Applied Computing 1996: Official Program of the 1996 ACM Symposium on Applied Computing*, pages 263–268, New York, 1996. ACM Press.
- [7] Junji Kitagawa and Hitoshi Iba. Identifying metabolic pathways and gene regulation networks with evolutionary algorithms. In Gary B. Fogel and David W. Corne, editors, *Evolutionary Computation in Bioinformatics*, pages 255–278. Morgan Kaufmann, San Francisco, CA, 2003.
- [8] Marc M. Lankhorst. A genetic algorithm for the induction of nondeterministic pushdown automata. Technical Report CS-R 9502, University of Groningen, 2002.
- [9] Simon M. Lucas and T. Jeff Reynolds. Learning DFA: Evolution versus evidence driven state merging. In *Congress on Evolutionary Computation 2003*, volume 1, pages 351–358, 2003.
- [10] Holger Mauch. Evolving Petri nets with a genetic algorithm. In Cantú-Paz [1], pages 1810–1811. Part II.
- [11] P. Mendes and D. B. Kell. Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14:869–883, 1999.
- [12] Jason H. Moore and Lance W. Hahn. Grammatical evolution for the discovery of Petri net models of complex genetic systems. In Cantú-Paz [1], pages 2412–2413. Part II.
- [13] N. Niparnan and P. Chongstitvatana. An improved genetic algorithm for the inference of finite state machines. In *2002 International Conference on Systems, Man, and Cybernetics*, volume 7, 2002.
- [14] James L. Peterson. Petri nets. *Computing Surveys*, 9(3):223–252, 1977.
- [15] James L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [16] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, second edition, 1992.
- [17] Nicholas J. Radcliffe. Genetic set recombination. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 203–219. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [18] Kazuhiro Saitou, Samir Malpathak, and Helge Qvam. Robust design of flexible manufacturing systems using colored Petri net and genetic algorithm. *Journal of Intelligent Manufacturing*, 13:339–351, 2002.
- [19] H. Tohme, M. Nakamura, K. Hachiman, and K. Onaga. Evolutionary Petri net approach to periodic job-shop scheduling. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, volume 4, pages 441–446, Tokyo, 1999. IEEE Computer Society Press.
- [20] Edgar E. Vallejo and Fernando Ramos. Evolving Turing machines for biosequence recognition and analysis. In *EuroGP '01: Proceedings of the 4th European Conference on Genetic Programming*, pages 192–203, London, 2001. Springer-Verlag.
- [21] John Yen, James C. Liao, Bogju Lee, and David Randolph. A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 28(2):171–191, 1998.
- [22] Pascal Yim and Thomas Bourdeaud'huy. Petri net controller synthesis using genetic search. In *Proceedings of the Second IEEE International Conference on Systems, Man and Cybernetics (SMC'02)*, volume 1, pages 528–533, Hammamet, Tunisia, 2002. IEEE Computer Society Press.