

An improved Cellular Automata model of enzyme kinetics based on Genetic Algorithm

Saurajyoti Kar¹, Kaustuv Nag^{2†}, Abhishek Dutta³, Denis Constaes⁴, Tandra Pal^{2*}

¹Department of Biotechnology, National Institute of Technology, Durgapur – 713209, West Bengal, India

²Department of Computer Science & Engineering, National Institute of Technology, Durgapur – 713209, West Bengal, India

³Department of Chemical Engineering, Groep T - Leuven Engineering College, (Associatie KU Leuven), A. Vesaliusstraat 13, B-3000 Leuven, Belgium

⁴Department of Mathematical Analysis, Ghent University, Galglaan 2, B-9000 Ghent, Belgium

[†]presently associated with Instrumentation and Electronics Engineering, Jadavpur University, Kolkata – 700 032, West Bengal, India

* corresponding author

E-mail address: tandra.pal@gmail.com (Tandra Pal)

Tel: +91-9434537021, fax: +91-343-2547375.

Introduction

Natural processes from population dynamics (Soarse-Filho et al., 2002; Lauf et al., 2012) to fluid dynamics (Somers, 1993; Chopard and Masselot, 1999), from biofilm growth (Pizarro et al., 2005; Fagerlind et al., 2012) to tumor growth (Kansal et al., 2002; Alemani et al., 2012) all occur in discrete space and time so that a probabilistic nature of representation is needed to understand precisely the dynamics of such processes. Stochasticity is an intrinsic property of any natural system, more specifically at a microscopic scale (Deutsch and Dormann, 2003; Frange et al., 2010). Several studies have observed occurrence of stochastic fluctuations in living systems (Abkowitz et al., 1996; Meng et al., 2004; Wu and Higgs 2012). Most biological events are triggered as network of chemical reactions which pose larger challenges due to computational representation and requirements (Kier et al., 2005). The noise of one event can alter the outcomes of the succeeding event and simultaneously affect the final outcome (Blake et al., 2003). When concentrations of species in an enzymatic reaction are high, the reaction can be mathematically modeled using deterministic approaches, like Michaelis-Menten kinetics. But, at lower concentrations, it is advisable to use modeling tools which have some stochastic property within the rule structure. Such models shall evolve to show certain global properties and are also unpredictable until the event actually occurs to completion. Thus an accurate mathematical model can help clarify the roles of individual components within a process and generate specific, testable hypotheses and predictions (Apte et al., 2008). Cellular Automata (CA) is one such stochastic approach in which a natural event can be successfully simulated. Using probabilistic rule application, CA can capture the properties that evolve from the stochastic nature present in microscopic scale in natural systems assuming both time and space to be discrete. In our earlier studies (Kar et al., 2010; Dutta et al., 2011), a generalized two-dimensional (2D) interaction

model was created to represent two-agent kinetic interactions of an enzymatic reaction. From the sensitivity analysis study of Dutta et al. (2013) based on several input parameters, it was realized that there cannot be any specific mathematical relationship between the input parameters and the model output. It is also not feasible to define a strict mathematical relation between the chemical kinetic constants and the CA model parameters as the model is an abstract representation of a realistic reaction. This makes repeated trial-and-error interactions of random parameter values the only way to capture specifically the resulting state of a model system. When there are multiple objective functions of a model to be satisfied, useful combinations of parameter values become rarer in the infinite pool of solutions (Konak et al., 2006). If the problem is NP-hard, finding the optimal solutions by exhaustive search may be too costly or practically impossible. Genetic Algorithm (GA) is a popular meta-heuristic tool frequently used for solving optimization problems (Holland et al., 1975).

Acid phosphatases (APs) are a family of enzymes that are widespread in nature, that non-specifically catalyze the hydrolysis of orthophosphate monoesters to produce inorganic phosphate and can be found in many animal and plant species (Bull et al., 2002). These enzymes are competitively inhibited by inorganic phosphate (Odds and Hierholzer, 1972). Alvarez (1962) was the first to characterize the kinetics and mechanism of the hydrolysis of phosphoric acid esters by potato acid phosphatase. He found that the enzymatic activity depended on three ionizable groups. Based on these findings, a reaction mechanism was proposed (see Fig 1). After the nucleophilic attack of one of the ionizable groups on the phosphorous atom of p-nitrophenyl phosphate, the enzyme-substrate complex is formed. After releasing p-nitrophenol, the enzyme-phosphate complex gets hydrolyzed, resulting in the formation of inorganic phosphate and recycled acid phosphatase.

In this study, an NSGA-II based Multi-Objective Genetic Algorithm (MOGA) approach is implemented for a stepwise improvement of the CA model. Finally, this approach is validated using the enzymatic hydrolysis of p-nitrophenyl phosphate by acid phosphatase with inhibition reaction.

Cellular Automata model

In accordance with our previous studies, the CA model is framed on a two-dimensional grid. Extended von-Neumann neighborhoods are considered for application of local rules. To represent the boundary as virtually infinite, periodic boundary condition are implemented. This facilitates the boundary molecules to interact with the molecules present in the boundary of opposite direction, as on the surface of toroid. Also, such a neighborhood implementation facilitates the application of probability rules in a combinatorial fashion without any exception due to edge restrictions (Dutta et al., 2011). Each of the reaction compound molecules is considered as a single agent in the model. Each cross section of the grid represents a single lattice node. An agent is allowed to occupy only one lattice node. The types of agents with their percentage occupancy are predefined, and the probability rules are declared as input parameters along with the grid size and iterations. The model can support a maximum of two agent interaction (first order and second order reactions) with equal numbers of molecules on either side of any reaction. This means, if a reaction complex forms at any time of the reaction, the model is valid for complexes so formed with two individual agents. The algorithm for rule application remains similar to Dutta et al. (2013). As earlier, the model is structured on several stochastic rules. There are four basic types of stochastic rules: the probability of Join (P_J) to form complexes, the probability of break (P_B) to break a formed complex, the probability of transition

(P_T) to define the reaction transition and the free movement probability (P_M) which captures the Brownian mixing phenomenon in the model. For a particular agent, each of its neighboring directions is independently examined for feasible events, each of whose probability is obtained in a multiplicative manner. A re-normalization of the directional event probability value is performed. A random direction is then marked for occurrence of the designated event. This makes the model a pure instantaneous model, rather than a memory based model. Each agent is updated once per iteration. Rules are applied in a continuous manner based on the fundamental hypothesis no two events occur simultaneously.

Genetic algorithm for multi-objective optimization

Genetic Algorithm is inspired by Darwin's law of survival of fittest (Goldberg et al., 1989). It mimics the process of natural evolution. It uses operators inspired by biological processes of Inheritance, Mutation, Selection and Crossover. The solutions are represented as chromosomes. Each chromosome is composed of values that can either be binary, real or floating point in representation. Using a set of Objective Functions (OFs), the chromosomes are mapped from variable (genotypic) space to objective (phenotypic) space. A set of solutions defines the population of a generation. The population evolves as more fit solutions (chromosomes) replaces less fit solutions. This way GA gradually nears to optimal solutions. A multi-objective optimization problem can be stated as a minimization problem as in Equation 1.

$$\begin{aligned} &\text{minimize } (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \in \mathfrak{R}^M \\ &\text{where } \mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathfrak{R}^n, \text{ and } x_i^L \leq x_i \leq x_i^U \end{aligned} \quad (1)$$

Here n is the number of variables. Let $\mathbf{u} = (u_1, u_2, \dots, u_n)^T \in \mathfrak{R}^n$ and $\mathbf{v} = (v_1, v_2, \dots, v_n)^T \in \mathfrak{R}^n$ be two solutions. Then \mathbf{u} dominates \mathbf{v} (or $\mathbf{u} \succ \mathbf{v}$), if $f_i(\mathbf{u}) \leq f_i(\mathbf{v}), \forall i$ and $\exists j$ such that $f_j(\mathbf{u}) < f_j(\mathbf{v})$.

Again \mathbf{u} and \mathbf{v} are said to be nondominated to each other (or $\mathbf{u} \parallel \mathbf{v}$) if neither $\mathbf{u} \succ \mathbf{v}$ nor $\mathbf{v} \succ \mathbf{u}$.

A solution that is not dominated by any other possible solution is called Pareto optimal solution. The set of all Pareto optimal solutions is called the Pareto Set (PS). The set of all corresponding objective vectors to PS is called the Pareto front (PF).

Given a set of solutions A , a subset of solutions $A_1 \subset A$ is obtained such that A_1 is a nondominated set of solutions and $(A - A_1)$ is a dominated set of solutions. Then A_1 becomes the first front and nondomination rank of all solutions of A_1 is 1. If $(A - A_1) \neq \phi$, another set of solutions $A_2 \subset (A - A_1)$ is obtained such that A_2 is a nondominated set of solutions. Then A_2 becomes the second front and all the solutions of A_2 have nondomination rank 2. Similarly an i^{th} front can be defined and therefore, all the solution of i^{th} will have nondomination rank i . Any MOGA tries to find a set of solutions which is close to PF, well spread and covers the whole spectrum of PF. Further details on multi objective optimization using genetic algorithms can be found in (Deb et al., 2001; Coello et al., 2007). Several multi-objective genetic algorithms can be found in literature (Zitzler et al., 2001; Deb et al., 2002; Nag et al., 2012). For further reviews on MOGA, the interested reader is referred to Coello et al. (1999; 2006) and Konak et al. (2006).

NSGA-II (Deb et al., 2002) is an elitist, population based genetic algorithm. The diversity of the population for a particular generation is measured by its *crowding distance*. Crowding distance can be assigned to each of the solutions of any nondominated set of solutions. The extreme (boundary) solutions are assigned infinity as crowding distance and for other solutions half the perimeter of its bounding cuboid is set as their crowding distance. It can be conceptualized that in each corner of the cuboid there exists the nearest neighbor according to the corresponding objective axis. A crowding distance assignment procedure is shown below:

```
assignCrowdingDistance (SolutionSet P) {
    for each i, set P[i].crowdingDistance = 0;
```

```

for each objective m,
    P = sortAscending(P, m);
    P[1].crowdingDistance = P[|P|].crowdingDistance = INFINITY;
    for i = 2 to (|P|-1),
        P[i].crowdingDistance += (P[i+1].m - P[i-1].m) / (P[|P|].m -
P[1].m);
    }

```

NSGA-II algorithm

Step-I (Population Initialization): Randomly initializes N number of solutions, where N is the population size.

Step-II (Generate N number of offspring): A mating pool is created using a binary tournament selection where a two-tier fitness mechanism is used. The solution with lower rank is preferred. If the solutions have the same nondomination rank, then the solution with higher *crowding distance* is the winner. Then, Simulated Binary Crossover (SBX) (Deb et al., 1995) is performed on the mating pool to create N offspring. Polynomial mutation (Deb et al., 2006) is followed by crossover to mutate the offspring. Next, all the mutated offspring are evaluated. Thus a set of N offspring is generated.

Step-III (Modify the population): The offspring and the parent population are added together to create a union of size $2N$. After that, from these $2N$ solutions, the best N solutions are selected to form the next generation. In this selection procedure, NSGA-II uses a two-tier fitness mechanism. Lower nondomination rank is given the higher priority and kept in the first tier. Crowding distance is used in the second tier to differentiate the quality of the solutions having same rank. To frontify, NSGA-II uses fast-non-dominated-sort (Deb et al., 2002) procedure.

Step-IV (Terminate the algorithm): If the termination condition (may be a maximum number of function evaluations or generations) is satisfied, perform a fast-non-dominated-sort and return the nondominated solutions of the archive, else go to Step-II.

Model formulation

The CA model in this study acts as a black box function for NSGA II. The input parameters are selected by the GA, which is then used to simulate the CA model. The simulation output is then used to calculate the fitness functions. Finally, new sets of input parameter values are obtained after implementing the genetic operators of the GA. The model iterates until the final targeted generation count is reached. The following parameters are considered as target optimization parameters: the joining probabilities between enzyme and substrate (Join (ES)), between enzyme and inhibitor (Join (EI)), between enzyme and product (Join (EP)), between water and substrate (Join (WS)) and between two water agents (Join (WW)), the breaking probabilities between enzyme and substrate (Break (ES)), between enzyme and inhibitor (Break (EI)), between enzyme and product (Break (EP)), between substrate and water (Break (WS)) and between two water agents (Break (WW)), the transition probabilities from enzyme-substrate to enzyme-product (Trans (ESEP)) and from enzyme-product to enzyme-substrate (Trans (EPES)). Due to the probabilistic nature of these parameters, the values are varied within the range of $1.0\text{E-}10$ to $9.99\text{E-}10$ (correct to ten decimal places). Each of these parameters can have any real value within the range.

The optimization approach used in this study is based on mainly three objective functions (OFs). The first OF (Eq. 2a) is the summation of all the probabilistic parameter values. The second OF (Eq. 2b) is the absolute difference of the product concentration between the target and the simulation output, thereby to obtain the model output product concentration as close as possible to the target product concentration. A third OF (Eq. 2c) has been tested during model validation work and is used to replace the first OF. It is defined to measure the initial speed of reaction for

product formation. This initial speed of the reaction is defined as first few n iterations of the model. The iteration number can be defined as the initial simulation settings. The first two functions are targeted to be minimized using the NSGA-II algorithm. For the third OF, whose target is to maximize in the positive axis, the effect is achieved by considering the negative of the value. The minimization of the summation of the parameter values (first OF) facilitates the optimizer to optimize the function while keeping the individual parameter values to minimum. Since the initial state of the CA model considers product concentration to be zero, these probabilistic parameter values have to increase from zero towards unity to reach the target product concentration. As such the summation value will always have an increasing tendency so that the model output product concentration can reach nearer to the target product concentration. This makes the first OF to have a mutually inverse relationship with the expected property. Furthermore, the second OF facilitates increase of final product formation and the third OF targets to increase the initial speed of the reaction so that more products are formed with fewer iterations. The flowchart in Figure 2 shows the logical flow structure of the GA-CA coupled approach.

$$\text{objective } f_1 = \min \left(\sum_{i=1}^{12} P_i \right) \quad (2a)$$

$$\text{objective } f_2 = \min \left| P_{\text{target}} - P_{\text{model}} \right| \quad (2b)$$

$$\text{objective } f_3 = \max \left(P_{\text{target}} - P_n \right) \quad (2c)$$

The CA program is developed in the C language. The Java coded open source jMETAL Optimization package (Durillo and Nebro, 2011) is used for the NSGA II algorithm. The communication between the GA and the CA occurs through file based I/O operation. This helps

the simulation design to be robust as the simulations can be restarted from the final terminated GA generation by using the status storage files.

Results and discussions

Initially, some pseudo real reaction sets are considered in order to study the capability of the model. A competitive enzyme inhibition reaction scheme is represented for all the simulation runs as shown in Figure 3. Such a reaction structure is modeled so that one can study the ability of the GA optimization. The representation considers substrate-water affinity and affinity of two water molecules among each other. This also illustrates the possibility of parallel reactions. The reaction scheme, in total, has twelve probability rule parameters.

The optimization study is performed in two grid sizes namely, 110×110 and 310×310 so as to compare the results. The initial parameter settings are set for four different target Substrate Conversion to Product (ScP): 10%, 18%, 23% and 90%. The initial concentration of water, substrate, enzyme, inhibitor and product are fixed to 60%, 4%, 1%, 1% and 0% respectively. 34% of void is maintained as per the molecular theory of intermediate space. The total number of iteration steps for each CA is fixed to 3000. Each optimization simulation is run for 250 generations, with 100 as the population size of each generation. Each of the single threaded optimization simulations for a 310×310 grid took approximately 22 days, while each of the optimization simulations for a 110×110 grid took approximately 4 days for completion, with the HPC Flemish computing facility. Figures 4(a-d) shows the characteristic Pareto front for the 110×110 grid for the corresponding ScP. The Pareto fronts for the 310×310 grid are shown in Figures 5(a-d).

There can be many combinations of probability parameter values that can be considered for simulation to reach a single ScP within considerable SD. Hence each optimization simulation results in multiple sets of near-optimal parameter values. But all these result sets are not realistic as some may have highly optimized first OF which may lead to poor second OF. To filter out such anomalies, the Two sample Z-Test method (see Eq. 3) is implemented on the simulation results sets, after calculating mean and standard deviation over 100 runs using the following relation:

$$Z = \frac{|M_E - M_i|}{\sqrt{s_E^2 + s_i^2}}, \quad (3)$$

where M_E is expected mean P count, M_i is obtained mean P count, s_E and s_i are the standard deviations of M_E and M_i respectively. A parameter set is considered acceptable only if Z value is less than 2. The acceptable detailed result sets and the corresponding probability parameter values are listed in Appendix (see Tables A(1-8) in Appendix).

Model Validation

To represent any kinetic reaction through a CA model, some informations are needed a priori such as the intermediate steps of the reaction, the initial concentration of the reacting agents and the final concentration of the target product. Since the CA model considers one node to one agent representation, the concentration of the reaction agents can be converted to % occupancy based on the calculation of Avogadro number. The void consideration of 34% on the CA grid is kept constant in all representations. Specifically for enzymatic reaction data, it is observed that the concentration scales of the reacting substrate and that of the participating enzyme are widely different. Such difference arises from the reusability and high activity properties of the enzyme. To be able to represent such a broad scale of concentrations, the size of the CA grid requires to

be increased proportionally. But a larger CA grid will require significantly more computational time. Hence, to keep the grid within manageable limit, the minimal concentration representation format is followed. The lowest-concentration agent will have the least possible count in the CA grid, possibly just one. But such a representation can cause reaction completion time reach infinity. After an initial exponential phase of the reaction the number of forward reacting agents reduce. This causes the overall probability of feasible neighborhood interaction events among the agents become unpredictably high. The randomness of free movement increases thereby delaying the expected forward reaction. This can be observed from the fact that the simulation takes only 83611 iterations for the first 98.64% of substrate conversion, whereas it takes as much as 131657 iterations for the next 1% of substrate conversion. Hence during optimization, the number of iterations required for automata evolution can be set. To choose the number of iterations for a particular reaction structure, sample runs of the CA model are performed with sufficiently high number of iterations until the reaction equilibrium is reached, while the values of the probability rules are changed in these simulations. Further while analyzing the concentration curves, additional iterations are considered so that it is well beyond the completion or attainment of reaction equilibrium. In this way, the resulting optimized parameter value sets can be used to simulate the CA model that can reach the target ScP provided. However, it must be noted that a large number of CA iterations may be required to reach the final targeted ScP due to the reduction in reaction speed once reaction completion/equilibrium is attained.

As mentioned earlier, the enzymatic hydrolysis of p-nitrophenyl phosphate by acid phosphatase with inhibition reaction is chosen for model validation. In the experiment, acid phosphatase (AP) from potato (P3752, Sigma Aldrich) is used. The enzymatic function in the production, transport and recycling of phosphate is critical for the metabolic and energy transduction processes of a

living cell. In the first reaction, the enzyme-substrate complex is formed by a nucleophilic attack on the phosphorous atom of the substrate. Subsequently, p-nitrophenol is released. Next, the enzyme-Pi complex is hydrolyzed to yield phosphate ions and recycled enzyme. Sodium phosphate monohydrate ($\text{NaH}_2\text{PO}_4 \cdot \text{H}_2\text{O}$) (Figure 6a) is used as inhibitor in the reaction. The intermediate steps of the reaction as represented by the CA-GA model are shown in Figure 6b. Four sets of experimental data with varying reactant and enzyme concentrations for different reaction times are collected for the validation. The concentration of the enzyme is proportionally less by a large factor of 10 for which the minimal agent representation format is followed, as described earlier. The CA iteration is fixed to 5000. Each of the GA optimization runs for 250 generations with population size 100. Equations 2b and 2c are the two optimization functions used for the validation scheme. The concentration gradient for the initial speed of reaction (for Eq. 2c) is measured for the first 10000 CA iterations. The results of the GA optimization are shown in Tables 1(a-d). Each element of the optimization set has a varying number of resultant optimized parameter sets. Due to the stochastic nature, the number of obtained optimized parameter sets may vary for different runs and for different targets of optimization. The obtained optimal parameters are then used to run independent simulations, each for a considerable number of CA iterations steps. The concentration graphs are plotted along the number of iterations for corresponding reaction set, as shown in Figures 7(a-d). The general characteristics of these graphs match well with the concentration profiles. Although most of the curves are overlapping, at times the characteristics can be drastically affected due to stochastic noise, as seen in Figure 7c (red curve, line1). Such a noise can be minimized only if the observations are made over a mean of several runs. However due to extremely high computational cost, this approach was avoided. The product yield (%P) obtained from the simulation tables and the simulation curves

indicates the validity of the CA-GA optimization tool. The CPU time for each simulation is approximately 6 hours. The simulations are run using a high performance compute cloud available on credit basis from Amazon Web Service (AWS). Each simulation is enabled to run with 36 parallel threads in dedicated processing cores available through AWS hardware level virtualization.

Conclusion

Cellular automata can represent a discrete system with a simple yet abstract representation of a natural process governed by probability rules. However, an abstract representation makes it highly inadequate to have a mathematical relationship between the process parameters, so that parameter estimation by trial and error is the only choice left to synchronize the results. In the current study, it is successfully shown that genetic algorithms can be used to optimize the parameter sets of a kinetic model. The validation study using enzymatic hydrolysis of disodium phosphate by acid phosphatase with inhibition reaction aptly describes the usefulness of the CA model when used with a GA optimizer to represent (bio)chemical reaction in silico. A practical implication of this study is that the stochastic CA model can represent each and every intermediate step of the kinetic reaction, which possibly cannot be observed in a deterministic model due to its limitations. Further it can be used to represent complex biological reactions and the simulation results can be analyzed without actually performing in vitro. It is also shown that an extensive need for computational resources for discrete modeling can be met using high performance cloud clusters and farm computational clusters, which are available at a reasonable price.

Acknowledgement

SK and AD would like to acknowledge Advait Apte, Arizona State University (USA) for his valuable inputs and discussions on the cellular automata model. AD would like to thank Paul Vandecruys, Laboratory of Molecular Cell Biology, Institute of Botany and Microbiology, KU Leuven (Belgium) for his help with the experimental work. For the pseudo-artificial reaction simulations, computational resources (Stevin Supercomputer Infrastructure) and services were used in this work and provided by Ghent University, the Hercules Foundation and the Flemish Government – Department EWI while for the model validation, computational resources of Amazon Web Service (AWS) were used to run parallel threaded simulations. The resources were available on credit basis with its elastic cloud infrastructure (EC2).

References

- Abkowitz, J., Catlin, S., Gutter, P., 1996. Evidence that hematopoiesis may be a stochastic process in vivo. *Nat. Med.* 2, 190-197.
- Aleman, D., Pappalardo, F., Pennisi, A., Brusci, V., 2012. Combining cellular automata and lattice Boltzmann method to model multiscale avascular tumor growth coupled with nutrient diffusion and immune competition. *Journal of Immunological Methods* 376, 1-2, 55-68.
- Alvarez, E.F., 1961. The kinetic and mechanism of the hydrolysis of phosphoric acid esters by potato acid phosphatase. *Biochimica et Biophysica Acta* 59, 672-680.
- Apte, A., Cain, J.W., Bonchev, D.G., Fong, S.S., 2008. Cellular automata simulation of topological effects on the dynamics of feed-forward motifs. *Journal of Biological Engineering* 2, 2, 1-12.
- Blake, W.J., Kaern, M., Cantor, C.R., Collins, J.J., 2003. Noise in eukaryotic gene expression. *Nature* 422, 633-637.
- Bull, H., Murray, P.G., Thomas, D., Fraser, A.M., Nelson, P.N., 2002. Acid phosphatases. *Molecular Pathology* 55, 2, 65-72.

Chopard B., Masselot, A., 1999. Cellular automata and lattice boltzmann methods: a new approach to computational fluid dynamics and particle transport. *Future Generation Computer Systems* 16, 2-3, 249-257.

Coello, C.A.C., 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge Information Systems* 1, 3, 269–308.

Coello, C.A.C., 2006. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine* 1, 1, 28–36.

Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation, second ed. Springer.

Deb, K., Agrawal, R.B., 1995. Simulated binary crossover for continuous search space, *Complex Systems* 9, 2, 115–148.

Deb, K., 2001. *Multi objective optimization using evolutionary algorithms*, New York: Wiley.

Deb, K., Pratap, A., Agarwal, A., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6, 182–197.

Deb, K., Tiwari, S., 2006. Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research* 185, 3, 1062-1087.

Deutsch, A., Dormann, S. *Cellular automaton modelling of biological pattern formation*. Birkhauser, Boston, 2005.

Durillo, J.J., Nebro, A.J., 2011. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* 42, 10, 760-771.

Dutta, A., Kar, S., Constales, D., Nopens, I., 2011. Modeling of first-order enzyme kinetic reactions: a Cellular Automata approach. In: 7th Intl. Workshop on Mathematics in Chemical Kinetics and Engineering (MaCKiE), Heidelberg, Germany.

Dutta, A., Kar, S., Constales, D., Nopens, I. Modeling of first-order enzyme kinetic reactions: a Cellular Automata approach. *Sadhana* (submitted).

Fagerlind, M.G., Webb, J.S., Barraud, N., McDougald, D., Jansson, A., Nilsson, P., Harlén, M., Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MIT: MIT Press.

Kansal, A.R., Torquato, S., Harsh IV, G.R., Chiocco, E.A., Deisboeck, T.S., 2002. Simulated Brain Tumor Growth Dynamics Using a Three-Dimensional Cellular Automaton. *Journal of Theoretical Biology* 203, 4, 367-382.

Kar, S., Ganai, S., Dutta, A., Dutta, D., Chaudhuri, S., 2010. A sensitivity analysis study of enzyme inhibition kinetics through cellular automata, AIP conference Proceedings (ICMOC) 98, 1, 301-306.

Kier, L.B., Bonchev, D., Buck, G.A., 2005. Modeling Biochemical Networks: A Cellular-Automata Approach. *Chemistry and Biodiversity* 2, 2, 233-243.

Kjelleberg, S., Rice, S.A., 2012. Dynamic modelling of cell death during biofilm development. *Journal of Theoretical Biology* 295, 23-36.

Konak, A., Cott, D.W., Smith, A.E., 2006. Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering Systems Safety*. Elsevier 91, 992–1007.

Lauf, S., Haase, S., Hostert, P., Lakes, T., Kleinschmit, B., 2012. Uncovering land-use dynamics driven by human decision-making – A combined model approach using cellular automata and system dynamics. *Environmental Modeling and Software*, 27-28, 71-82.

Meng, T. C., Somani, S., Dhar, P., 2004. Modeling and simulation of biological systems with stochasticity. *In Silico Biology* 4, 293-309.

Nag, K., Pal, T., 2012. A new Archive-based Steady-state Genetic Algorithm, *World Congress on Computational Intelligence, WCCI 2012 IEEE*, 853-859, Brisbane, Australia.

Pizarro, G., Teixeira, J., Sepúlveda, M., Noguera, D., 2005. Bitwise implementation of a two-dimensional cellular automata biofilm model. *Journal of Computing in Civil Engineering* 19, 3, 258-268.

Somers, J.A. 1993. Direct simulation of fluid flow with cellular automata and the lattice-Boltzmann equation. *Applied Scientific Research*, 51, 127-133.

Soarse-Filho, B.S., Cerqueira, G.C., Pennachin, C.L., 2002. Dinamica - A stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. *Ecological modeling*, 154, 3, 217-235.

Wu, M., Higgs, P.G., 2012. The origin of life is a spatially localized stochastic transition. *Biology Direct* 24;7-42.

Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength pareto evolutionary algorithm. *Computational Engineering Network Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Technical Report 103*.

P_Join_ES	P_Join_AB	P_Join_AW	P_Join_EI	P_Break_ES	P_Break_AB	P_Break_EP	P_Break_EI	P_trans_ESAB	P_trans_ABES	P_Trans_AWEP	Iterations	% P
0.702837	0.121707	0.781171	0.029324	0.133784	0.590446	0.622463	0.079896	0.978648	0.312356	0.990690	163552	1.30
0.653945	0.177088	0.817983	0.887521	0.393030	0.868474	0.839000	0.922149	0.636805	0.131928	0.990050	183957	1.30
0.613252	0.370823	0.790154	0.797484	0.349440	0.600341	0.358055	0.953598	0.646044	0.269840	0.991711	215268	1.30

Table 1a: Optimized results of the coupled CA-GA simulation using initial concentrations (Substrate 1 (S) = 43.12%, Substrate 2 (W) = 1.31%, Inhibitor (I) = 21.56%, Enzyme (E) = 0.005% and Void () = 34%) for a 150×150 CA grid size, 5000 CA iterations, 100 GA population size and 250 GA generations of simulation. Final target Product (P) = 1.30%. The chemical reaction time is 30 mins. Simulation time with 36 parallel threads took approximately 6 hours.

P_Join_ES	P_Join_AB	P_Join_AW	P_Join_EI	P_Break_ES	P_Break_AB	P_Break_EP	P_Break_EI	P_trans_ESAB	P_trans_ABES	P_Trans_AWEP	Iterations	% P
0.031686	0.044148	0.890304	0.117166	0.524927	0.991674	0.711479	0.710109	0.531081	0.001208	0.994324	272196	8.41
0.117018	0.037198	0.984235	0.160812	0.520963	0.868183	0.311518	0.866367	0.959528	0.028873	0.995971	400000	8.39
0.985828	0.681940	0.979454	0.492411	0.649391	0.719708	0.619185	0.740343	0.950976	0.012735	0.979529	289698	8.41
0.034687	0.442827	0.979739	0.163103	0.557778	0.865898	0.311518	0.866367	0.948876	0.013018	0.995933	321388	8.40
0.956729	0.005193	0.986940	0.457661	0.651398	0.945841	0.619185	0.740930	0.968777	0.024097	0.979792	400000	8.39

Table 1b: Optimized results of the coupled CA-GA simulation using initial concentrations (Substrate 1 (S) = 38.39%, Substrate 2 (W) = 8.41%, Inhibitor (I) = 19.19%, Enzyme (E) = 0.002823% and Void () = 34%) for a 150×150 CA grid size, 5000 CA iterations, 100 GA population size and 250 GA generations of simulation. Final target Product (P) = 8.41%. The chemical reaction time is 2.5 mins. Simulation time with 36 parallel threads took approximately 6 hours.

P_Join_ES	P_Join_AB	P_Join_AW	P_Join_EI	P_Break_ES	P_Break_AB	P_Break_EP	P_Break_EI	P_trans_ESAB	P_trans_ABES	P_Trans_AWEP	Iterations	% P
0.477159	0.034581	0.773444	0.029921	0.086241	0.884724	0.936365	0.943321	0.922237	0.008024	0.989975	400000	11.08
0.465103	0.045059	0.805223	0.268620	0.078904	0.905332	0.890624	0.888123	0.857370	0.020236	0.988668	400000	13.00
0.774559	0.436236	0.805223	0.268620	0.078919	0.989605	0.958038	0.888123	0.857370	0.020236	0.988668	400000	13.01
0.765923	0.350941	0.834075	0.633454	0.531113	0.989776	0.991090	0.810838	0.947486	0.017457	0.960751	400000	12.98
0.767359	0.033105	0.688106	0.274920	0.153293	0.979561	0.981325	0.329886	0.945152	0.017488	0.986868	400000	13.03

Table 1c: Optimized results of the coupled CA-GA simulation using initial concentrations (Substrate 1 (S) = 35.31%, Substrate 2 (W) = 13.04%, Inhibitor (I) = 17.64%, Enzyme (E) = 0.002596% and Void () = 34%) for a 150×150 CA grid size, 5000 CA iterations, 100 GA population size and 250 GA generations of simulation. Final target Product (P) = 13.04%. The chemical reaction time is 5 mins. Simulation time with 36 parallel threads took approximately 6 hours.

P_Join_ES	P_Join_AB	P_Join_AW	P_Join_EI	P_Break_ES	P_Break_AB	P_Break_EP	P_Break_EI	P_trans_ESAB	P_trans_ABES	P_Trans_AWEP	Iterations	% P
0.816042	0.031226	0.811154	0.258073	0.006672	0.961697	0.921515	0.998066	0.999853	0.000574	0.994197	400000	24.26
0.993704	0.103822	0.811589	0.846079	0.007089	0.999228	0.948470	0.915764	0.999829	0.002215	0.995784	400000	24.25

Table 1d: Optimized results of the coupled CA-GA simulation using initial concentrations (Substrate 1 (S) = 27.81%, Substrate 2 (W) = 24.27%, Inhibitor (I) = 13.90%, Enzyme (E) = 0.0204% and Void () = 34%) for a 150×150 CA grid size, 5000 CA iterations, 100 GA population size and 250 GA generations of simulation. Final target Product (P) = 24.27%. The chemical reaction time is 2.5 mins. Simulation time with 36 parallel threads took approximately 6 hours.

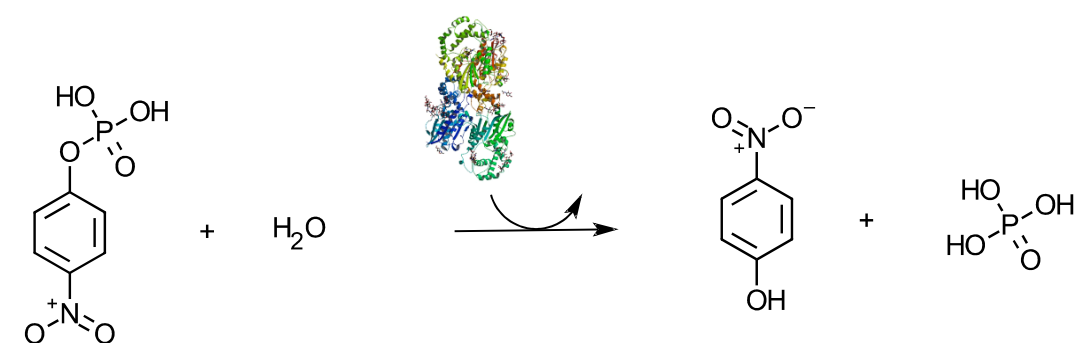


Figure 1: Molecular reaction structure of P-nitrophenylphosphate and water reacting in presence of acid phosphatase enzyme to form P-nitrophenols and Phosphoric acid.

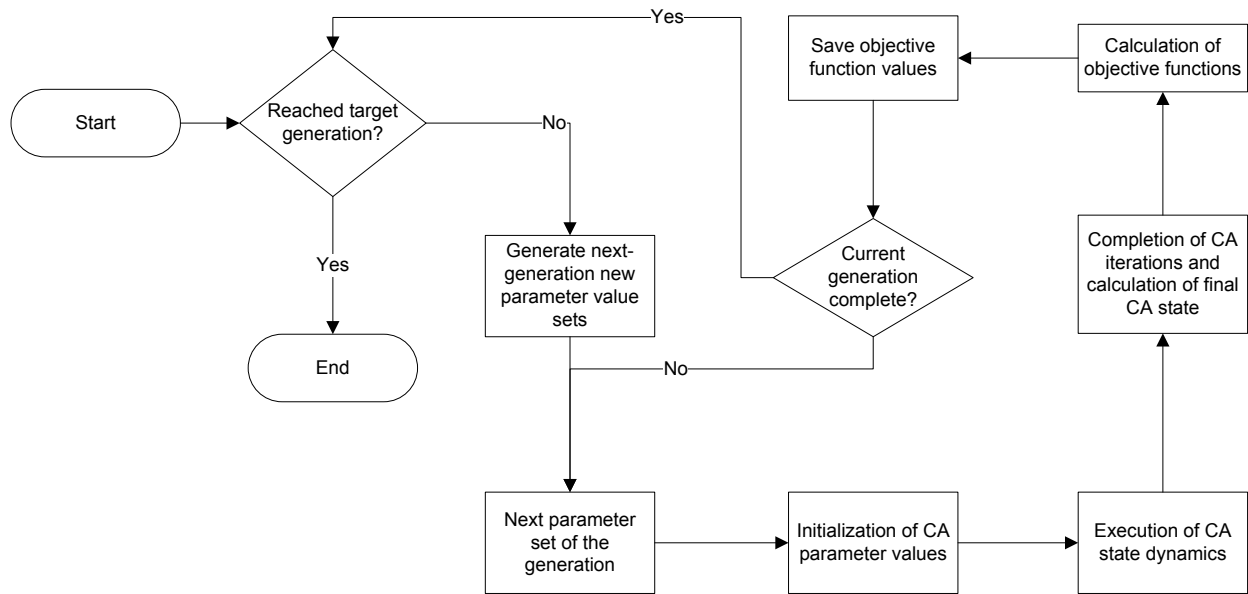


Figure 2: A simplified flowchart for the Genetic Algorithm (GA) coupled with Cellular Automata (CA) model system implemented in this study.

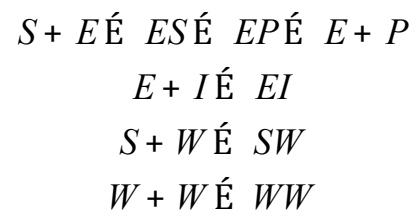


Figure 3: Reaction steps for the pseudo-artificial competitive enzyme inhibition reaction representation in the CA model. The symbols indicate Substrate (S), Enzyme (E), Product (P), Inhibitor (I) and Water (W).

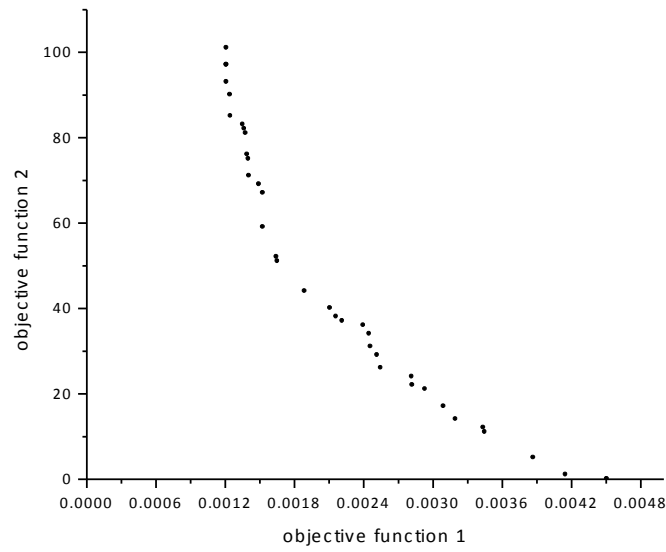


Figure 4a: Pareto front at 250th generation using 110×110 grid size obtained from the two objective functions (Eqs. 2a and 2b), for each member of the population. (Target S conversion = 23%).

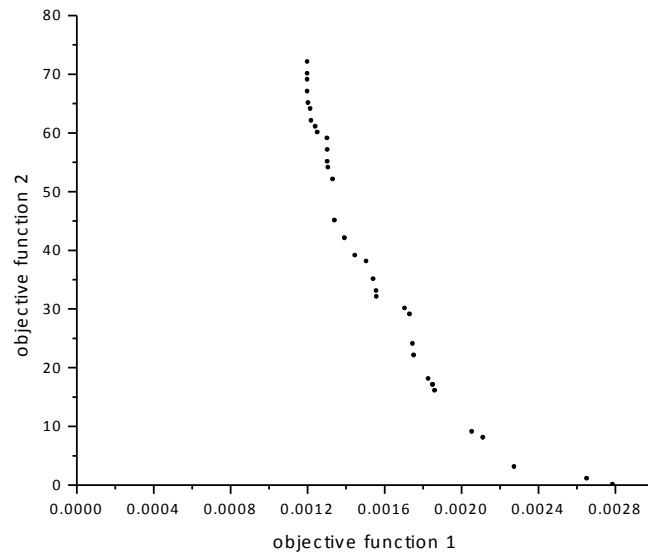


Figure 4b: Pareto front at 250th generation using 110×110 grid size obtained from the two objective functions (Eqs. 2a and 2b), for each member of the population. (Target S conversion = 18%).

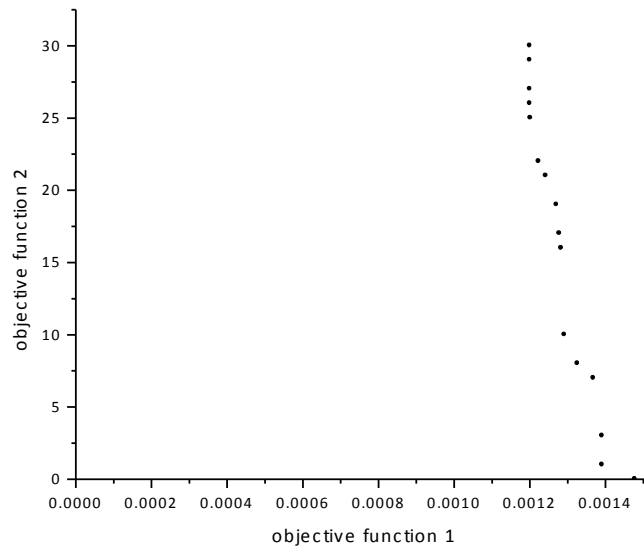


Figure 4c: Pareto front at 250th generation using 110×110 grid size obtained from the two objective functions (Eqs. 2a and 2b). for each member of the population. (Target S conversion = 10%).

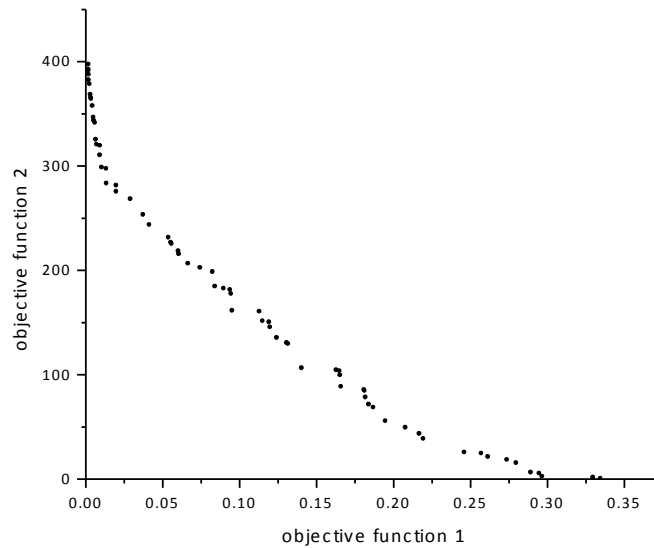


Figure 4d: Pareto front at 250th generation using 110×110 grid size obtained from the two objective functions (Eqs. 2a and 2b). for each member of the population. (Target S conversion = 90%).

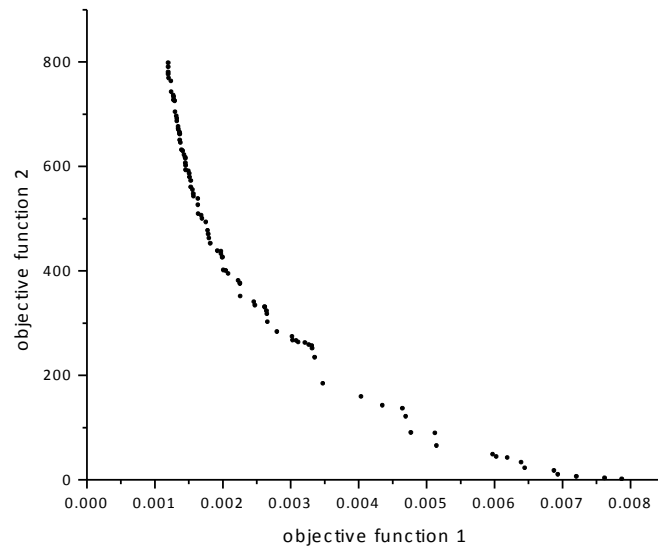


Figure 5a: Pareto front at 250th generation using 310×310 grid size obtained from the two objective functions (Eqs. 2a and 2b). for each member of the population. (Target S conversion = 23%).

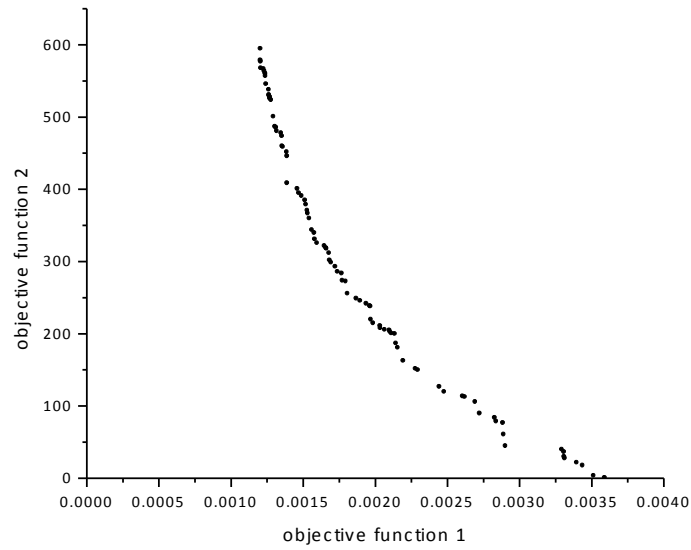


Figure 5b: Pareto front at 250th generation using 310×310 grid size obtained from the two objective functions (Eqs. 2a and 2b). for each member of the population. (Target S conversion =18%).

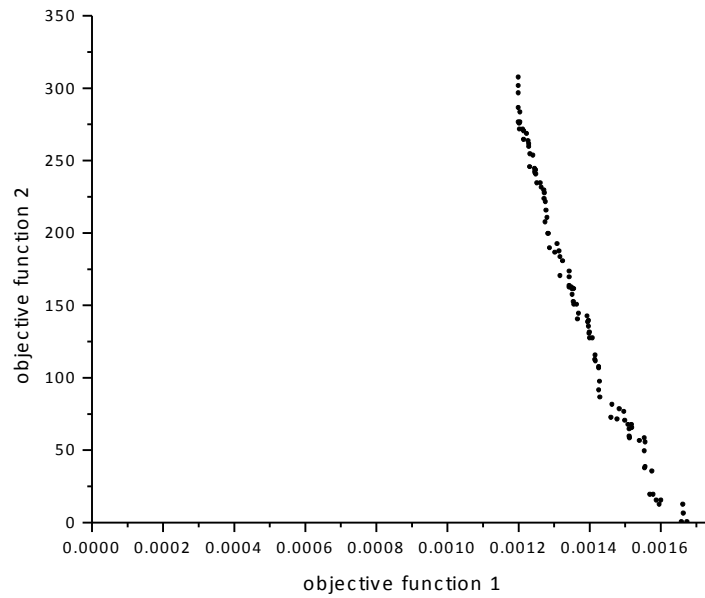


Figure 5c: Pareto front at 250th generation using 310×310 grid size obtained from the two objective functions (Eqs. 2a and 2b). for each member of the population. (Target S conversion =10%).

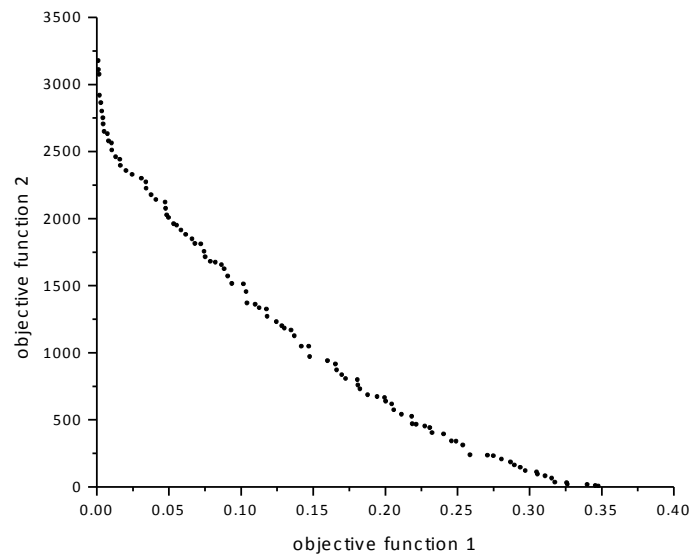


Figure 5d: Pareto front at 250th generation using 310×310 grid size obtained from the two objective functions (Eqs. 2a and 2b). for each member of the population. (Target S conversion = 90%).

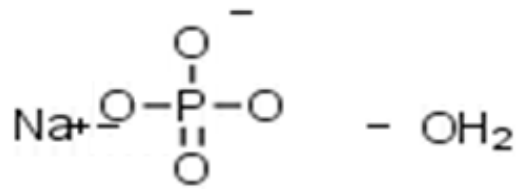


Figure 6a: Molecular structure of inhibitor sodium phosphate monohydrate.

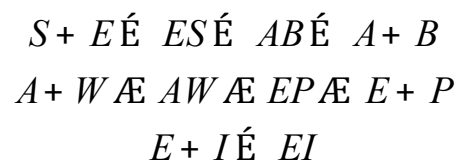


Figure 6b: Intermediate steps of the enzymatic reaction (S: P-nitrophenylphosphate, E: acid phosphatase enzyme, A: P_i-enzyme complex, B: P-nitrophenol, W: water, P: Phosphate ion and I: inhibitor) used for model validation in the study.

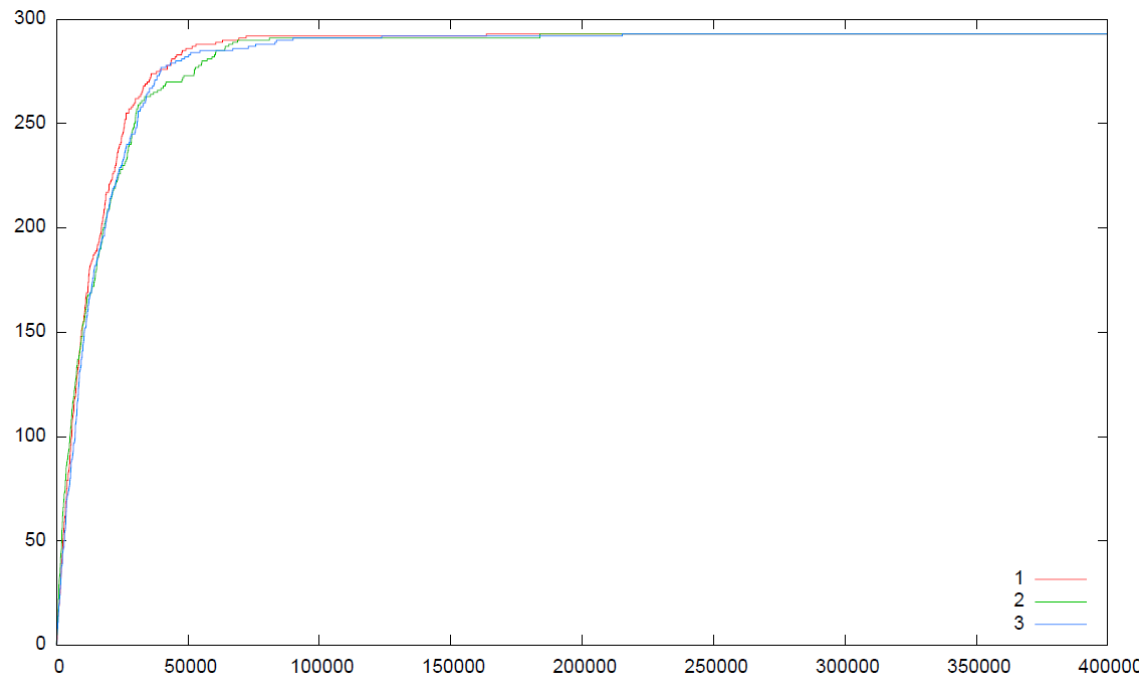


Figure 7a: The substrate concentration graph obtained from simulation run using the optimized parameter sets (see Table A1). The x-axis represents the iteration number and y-axis represents the substrate count at its corresponding iteration step.

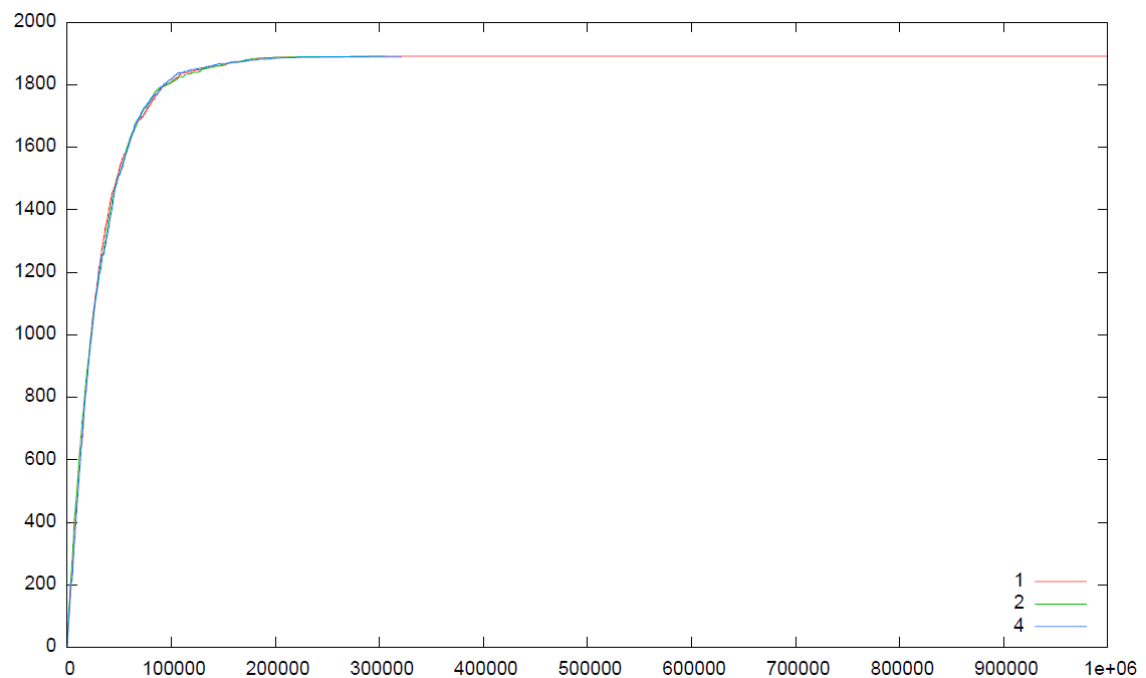


Figure 7b: The substrate concentration graph obtained from simulation run using the optimized parameter sets (refer Table A2). The x-axis represents iteration number and y-axis represents the substrate count at its corresponding iteration step.

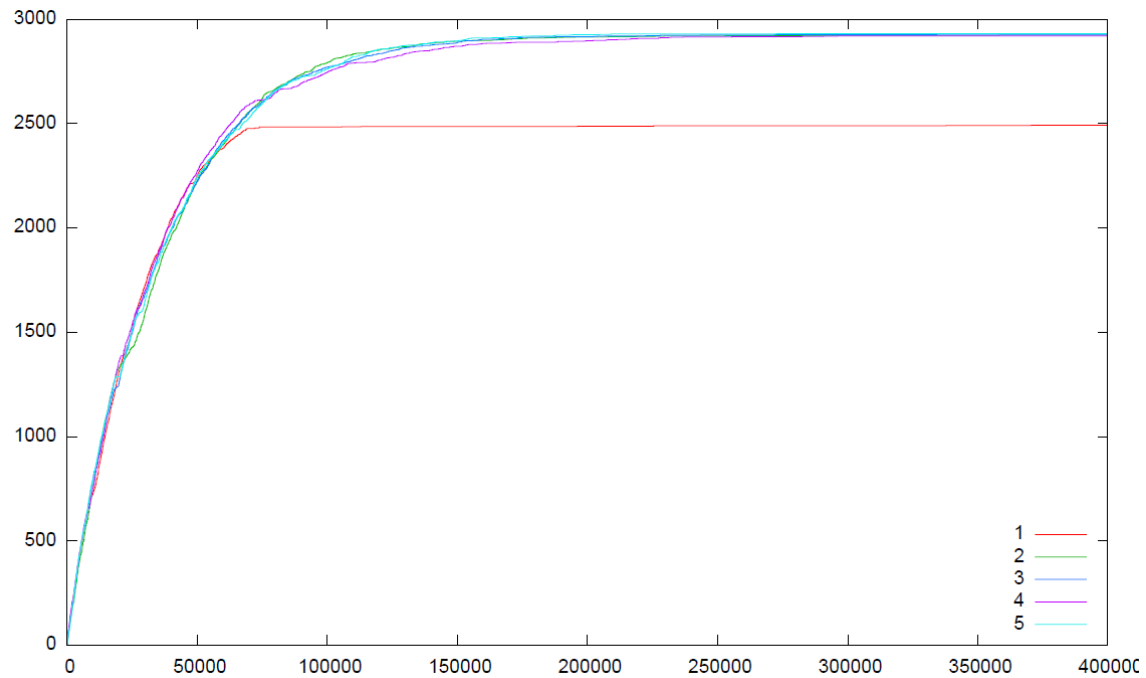


Figure 7c: The substrate concentration graph obtained from simulation run using the optimized parameter sets (refer Table A3). The x-axis represents iteration number and y-axis represents the substrate count at its corresponding iteration step.

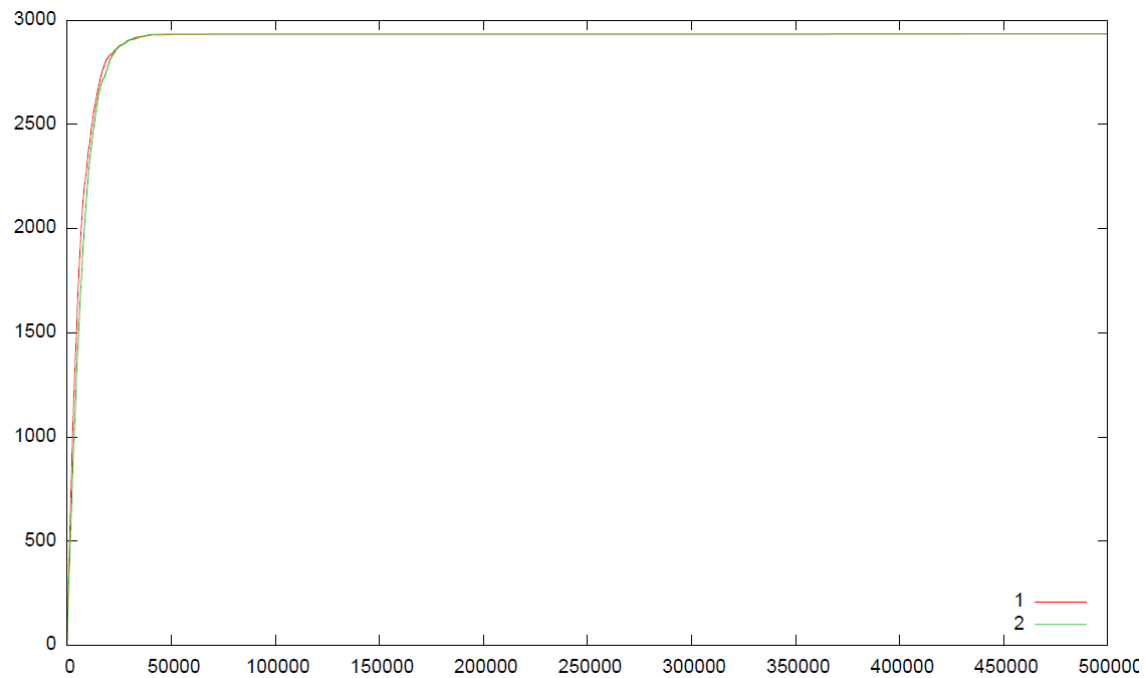


Figure 7d: The substrate concentration graph obtained from simulation run using the optimized parameter sets (refer Table A4). The x-axis represents iteration number and y-axis represents the substrate count at its corresponding iteration step.