

Article

Solving the Graph Problem on the Maximal Clique Problem on the P-Systems with Mitochondria Enzymes Layer

Ford Lumban Gaol^{1,*} and Tokuro Matsuo²

¹ Computer Science Department, BINUS Graduate Program-Doctor of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

² Graduate School of Industrial Technology, Advanced Institute of Industrial Technology, Tokyo 140-0011, Japan

Abstract: P systems with mitochondria enzymes layer computing is a recent version of P systems; it integrates a new feature inspired from the enzymes gate of a natural cell to the cell-like P systems. The model of a computational layer is well known as a problem of Non-Deterministic (NP-complete) in polynomial time. In this paper, we propose a P system with enzymes to solve one of the most NP-hard problems, which is the determination of the maximal clique in a given undirected graph. In this context, the evolution strategy proposed is based on using objects under the control of enzymes placed on layers.

Keywords: maximal clique; computational layer; P systems; enzymes layers; enzymes control

1. Introduction

The concept of P systems with mitochondria enzymes on layers as introduced in [1] is a model of a cell-like P system consisting of the combination of P Systems [2,3] with the neuron brain [4], where the reactions processed in the regions of a P system are controlled by a set of enzymes placed on layers. It was initiated with the aim of: (1) extending the space of reactions, (2) improving the model of the basis P systems by integrating the features and/or considering the mechanisms of the biological model. To show how effective systems are and their applicability, some works that exploited the invoked systems well are cited. For instance, in P systems enzymes-based, Păun et al. [5] propose a current approach to solve the NP-complete SAT problem which solvable in $O(nm)$ time, in which n is the number of variables and m is the number of that written in the form of normal model. Thus, the authors have provided an in-depth investigation into the computational complexity of the complex system and show that can be solvable even in the case of systems that do not even use the layer division and have only one layer. Another interesting work which has taken into account the concept of enzymes applied on layers is presented by Păun et al. [6,7] which showed that P-systems with enzymes on layers can be solved in polynomial time just like the class of problems Polynomial Space (PSPACE-Complete). Therefore, we can conclude that these P-systems are computationally equivalent to the Turing machine. Thus, among problems that require most execution time and considered to develop a finite solution in linear or polynomial time, maximal clique problem is a good example. This problem is well known as the algorithm computationally with other graph theory problems, for instance, the stable maximal independent set problem and the minimum nodes cover problem. As it is known that these are NP-hard problems, we understand that no solution to the polynomial time algorithms is expected to be found. Moreover, these problems are well known and have several important practical applications;

indeed, it will become very challenging to explore and enhance efficient, exact algorithms for many possible problems.

Thus, given a group of nodes which have edges between them, the problem on maximal clique has an objective to find the largest subset of vertices in which each point is in conjunction with every other node. Indeed, provided that the maximal clique problem has many applications in bioinformatics, computational chemistry, social networks, etc. several directions of research have been explored to find an “optimal” solution. For instance, Pardalos et al. [8] presented an efficient and precise parallel algorithm for the maximal clique on the graphs. The experiments on computation with sparse graphs with combination with the test graphs from applications are shown. The algorithm is parallelized using the message passing interface (MPI) standard. In [9], a practical algorithm for finding the maximal clique is proposed. It is based on colour classes and the backtracking technique. It already shown that the algorithm was tested on The Center for Discrete Mathematics and Theoretical Computer Science (DIMACS)—DIMACS graphs and provide the comparison with other well-known algorithms. Base on the experiments has shown the good performance and can achieve more than 10,000 times faster than other better known algorithms on some graph types.

Konc et al. in [10], provide in depth discussion of an algorithm in an undirected graph with the solution of maximal clique. There are some improvements on the approximate coloring algorithm that used to provide bounds to the size of the maximal clique. This approximate coloring algorithm was then expanded to include with the constraint on the dynamically varying bounds. The experiment showed that the resulting algorithm is significantly better in the term of computation time than the comparable algorithm. The implication of the processing speed of the maximal clique algorithm of varying the tightness of upper bounds dynamically during the search are investigated experimentally. In another interesting work presented by Pullan et al. [11], the authors introduced with the probabilistic stochastic on the adverse search algorithm that intend for the maximal clique problem. The adverse search algorithm alternates between phases of iterative improvement, during which suitable vertices are added to the current clique, and plateau search, during which the vertices of the current clique are changing with vertices not contained in the current clique. Thus, the authors showed empirically that the proposed algorithm achieves substantial performance improvements. In [12], Shinano et al. have already shown that the success in combining the latest results for the maximal clique problem with DIMACS. Therefore, several improvements of DIMACS gave a significantly increased performance, even if the some increment procedure of the problem is much complex than the processing time base on the link used. Consequently, more than 20 previously open problems that taken from from DIMACS benchmarks were solved using 71 networked work-stations.

Finally, whether we consider the biological context of layer computing or not, a novel contribution consisting in adding some biological concepts like DNA molecules is introduced by Ouyang et al. [13] to solve by means of the molecular biology technique for the maximal clique problem. On the current research, a collection of DNA molecules that corresponds with the total ensemble of 20-node cliques was built, that accompanied with series of selection processes. The algorithm was design on the parallel and shown with very good satisfactory result.

This paper aims to provide the alternative solution for the maximal clique problem based in P systems with mitochondria enzymes on layers which involves the layer objects rules, division and dissolution rules under the control of multisets of Mitochondria enzymes. The rest of the paper is organized as follows. Section 2 shows the inherent concepts of P systems based on mitochondria enzymes. Hence, the main rules used in developing the proposed algorithm were reported. Section 3 describes in further detail of the functioning of the proposed approach in order to finally reach the maximal clique of a given graph. Section 4 will provide with the numerical example and sensitivity analysis and Section 5 will provide the concluding of the work.

2. P Systems with Enzymes on Layers

Layer systems or P systems are already known as a class of parallel modified algorithm and classified as non0deterministic algorithm that are inspired by the structure and the functions of living cells. The fundamental structure of a P system [2,3,14] is a levelling arrangement of layers that delimits structure called also areas; each area contains collection of objects. The components of the system will be developed gradually by a set of rules associated with the areas, which are classified in three main types [2]: the rewriting of multiset that transformed the objects into another one or introduced a new one; the relationship rules which bring the objects from regions; and the correlation for handling layers. The P system with mitochondria enzymes on layers is a P system with two main types of components; mitochondria enzymes and common components, which stated that the previous are placed on the layers, and the next are located in the areas that separated by layers. In some senses, the areas of layers can contain collection of objects and of mitochondria enzymes respectively. Two types of such enzymes in the layer mentioned in literature; integral enzymes, and peripheral enzymes see [2]. The types of rules that can be found in this current model are: proceed components across layers under the control of layer enzymes with changing or not the objects and/or the enzymes during these processes. More than that, it is possible to interchange two objects through the enzymes permitting the P system with enzymes to be an expanded set of antiport rules [2,15]. We could define the P system with mitochondria enzymes to be represented by the following construction:

$$\Pi = (O, P, \mu, w_1/P_1, \dots, w_m/P_m, R_1, \dots, R_m, i_0)$$

where:

- O is an alphabet; its elements are called components;
- P is an alphabet; its sets are called enzymes, $P_1, \dots, P_m$ are the collection of enzymes on the m layers of μ ;
- μ is a layer structure consists of m layers, m constitute also the level of Π ;
- $w_i, 1 \leq i \leq m$, are collection of strings which part of the initial multisets over V associated with the regions $1, 2, \dots, m$ of μ ;
- $R_i, 1 \leq i \leq m$, are sets of rules in a maximal parallel manner that associated to the regions $1, 2, \dots, m$ of μ ;
- i_0 , is a indexing between 1 and m which shown on the output layer of Π .

Based on the types of rules that introduced in [1,5]. In all of these rules, $a; b; c; d$ are components where p is a enzyme, and i is a label of layer ('reg' stands for 'regulated'), in what follow the fact that a enzyme p is on a layer labeled i is written in the form $[\mid p]_i$:

Type	Rule	Effect
1reg	$[a \mid p]_i \rightarrow [b \mid p]_i$	
	$a[\mid p]_i \rightarrow b[\mid p]_i$	Change an object, but not moving
2reg	$[a \mid p]_i \rightarrow a[\mid p]_i$	
	$a[\mid p]_i \rightarrow [a \mid p]_i$	Moving an object, but not design
3reg	$[a \mid p]_i \rightarrow b[\mid p]_i$	
	$a[\mid p]_i \rightarrow [b \mid p]_i$	Design and move a component
4reg	$a[b \mid p]_i \rightarrow b[a \mid p]_i$	Correlate two components
5reg	$a[b \mid p]_i \rightarrow c[d \mid p]_i$	Correlate and design two components

Based on all possible situations above, the enzyme is not altered; it plays the role of a catalyst, assisting only the evolution of objects. A more generalized is to permit rules of the model below (now, "alt" means "altered enzyme"):

Type Rule Effect

- 1alt $[a|p]_i \rightarrow [b|p]_i$
 $a[|p]_i \rightarrow b[|p]_i$ Change an object, but not moving
 2alt $[a|p]_i \rightarrow a[|p]_i$
 $a[|p]_i \rightarrow [a|p]_i$ Moving an object, but not design
 3alt $[a|p]_i \rightarrow b[|p]_i$
 $a[|p]_i \rightarrow [b|p]_i$ Design and move a component
 4alt $a[b|p]_i \rightarrow b[a|p]_i$ Correlate two components
 5alt $a[b|p]_i \rightarrow c[d|p]_i$ Correlate and design two components
-

To divide a layer, the following type of control is defined, where p, p^l, p^u are enzymes (possibly the same): $[|p]_i \rightarrow [|p^l]_i [|p^u]_i$ means that the layer i is assumed not to have any join.

3. Solving the Graph Maximal Clique Problem

The maximal clique problem is a one of the famous graph constraint satisfaction problems with some applied-world implementations. The maximal clique problem is classified as NP-complete and computationally intractable even to approximate with certain absolute performance bounds. In this part, we provide the proposed alternatives approach for solving the max clique problem for the graph G using P systems with enzymes on layers, where $G = (V, E)$ be an undirected graph, where V is the set of n nodes and E is the set of m edges. A clique is a complete sub graph of G , a set of vertices from G that are pair wise adjacent. The maximal clique problem has goal to find the maximal complete sub graph of G .

We will discuss of Figure 1 that outlines an instance of a graph G with nodes (n) = 5 and edges (m) = 10. Each illustration is drawn base on the edge x_i that represents the with i^{th} edge of the graph. All the edges are enumerated in chronological order such that from one node j the connecting edges to this node are labelled from $k > i$. As illustrated on Figure 1, x_1 represents the edge $e(1, 2)$, i.e., the edge between the node 1 and the node 2; x_2 represents the edge $e(1, 3)$, etc. For the need of the used approach, the graph is virtually decomposed in levels of nodes where each node j is connected to the nodes of the next higher level. Also, in addition to being an edge of the graph, x_i is also considered as a variable or an object in the following manipulated rules of the P system. This means that each variable is equivalent to a clique of two nodes in the graph. Effectively, the set of edges is not only the variables of the system. Other pertinent elements are useful to model because they are being used to represent the construction of clique and to lead to the determination of the maximal length of a given clique.

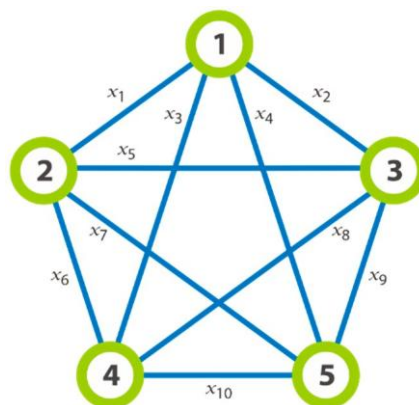


Figure 1. Instance of a graph G .

As well as the graph G of Figure 1 could be decomposed in three complete graphs such that: $G_1 = \{x_1, x_2, x_3, x_4\}$, $G_2 = \{x_5, x_6, x_7\}$, and $G_3 = \{x_8, x_9\}$. Let us taken $G_1 = \{x_1, x_2, x_3, x_4\}$ as an illustration on the example for the required transformation made after we have chosen an instance

of graph. Thus, Figure 2a outlines a graph with 4 nodes which will be transformed in new graph that provided the detail on the right part of the same image. As we can see, the nodes of this new graph are labelled by x_i and its edges are generated by $q_{i,j}$, that formed by the edges x_i, x_j which having the common extremity. It is significant to notice that an edge q_{ij} that symbolizes a clique of three nodes. For instance, the edge $q_{1,3}$ is formed by the edges (x_1, x_3) and subsists if and only if the edge 6 (Figure 2a) is created. In general, we consider that the set of all x_i is divided into $(n - 1)$ subsets where each subset includes the edges of the same level. According to Figure 2 on the left, we can say that the set of $\{x_1, x_2, x_3, x_6, x_8\}$ is divided into the first subset $\{x_1, x_2, x_3\}$ which represents the edges of level 1, and all these nodes have the common extremity the node N_1 ; the second subset $\{x_6\}$ which represents the edge of level 2; etc.

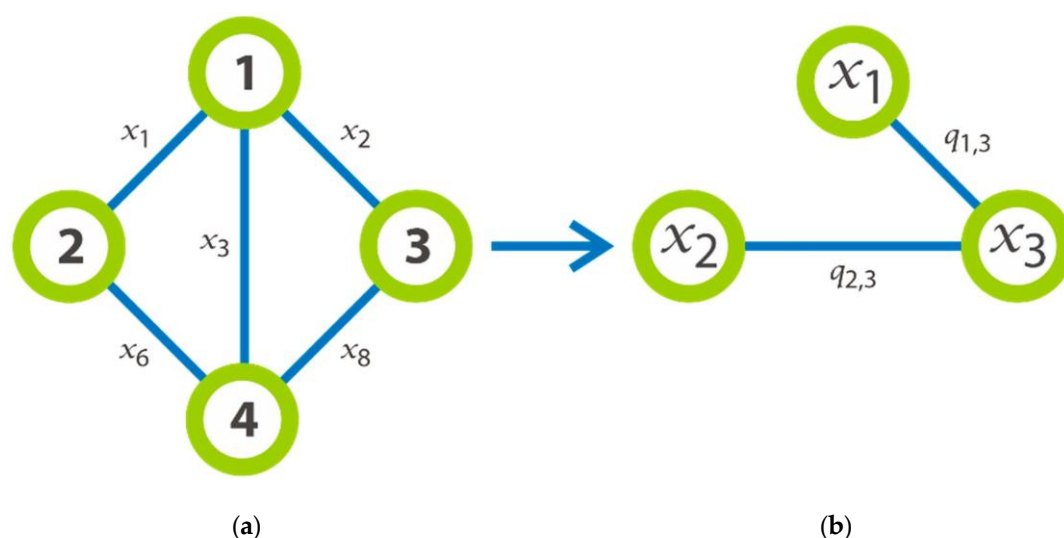


Figure 2. The instance of a graph based on Figure 1 with 4 nodes that transformed into a second reduced graph. (a) The existing graph, (b) the transformation graph.

We could say that a clique of size $Q \geq 4$ possibly could be decomposed in $D = 4(Q - 4)$ cliques of four nodes. This also means that $4(Q - 4)$ cliques of four nodes, each two of them having two common extremities can form a clique of size Q .

In what follows, let $G = (V, D)$ be a graph with n nodes and m edges. We assume that the least clique in this graph is a clique of five nodes (i.e., $n \geq 5$). The preliminary configuration of the proposed model associated to a graph G is illustrated in Figure 3. Where w_i are the initial sets of objects $p_E, z_E, z_{E+1}, \dots, p_R, h_{ij}$, the sets of enzymes associated to the layer i , and $\dot{p}, h_i$ is a enzyme associated to layers c_i and to layer n such that:

$$1 \leq i \leq n - 1;$$

$$E = (i - 1)n - [(i - 1) + (i - 2) + \dots + (i - i)] + 1; \quad (1)$$

$$R = E + n - (i + 1); \quad (2)$$

$$j = n - i. \quad (3)$$

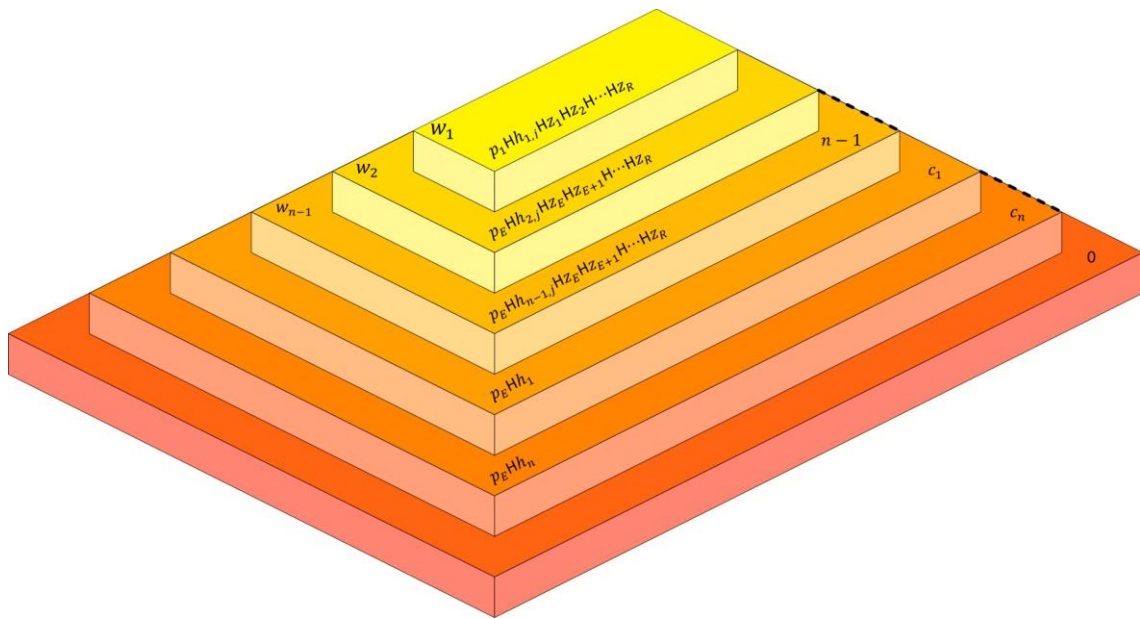


Figure 3. Preliminary configuration of the proposed model.

The system is composed by $(2n - 1)$ outer layers embedded in each layer. The elementary layer, labeled 1, is associated to the node N_1 and is included in the non-elementary layer, labeled 2 and which is included in a non-elementary layer and associated to the node N_2 and so on, until we reach a layer labeled with $(n - 1)$ which included on its turn in a set of embedded layers labeled with c_i . The initial multisets consist in:

$$w \quad \{a_u, k_0\} \cup \{x_1, x_2, \dots, x_R\} \cup \{q_{1,2}, q_{1,3}, \dots, q_{1,R}\} \cup \{q_{2,3}, q_{2,4}, \dots, q_{2,R}\} \cup \dots \cup \{q_{R-1,R}\}; \quad (4)$$

$$w \quad = \{k_0\} \cup \{x_E, x_{E+1}, \dots, x_R\} \cup \{q_{E,E+1}, q_{E,E+2}, \dots, q_{E,R}\} \cup \{q_{E+1,E+2}, q_{E+2,E+3}, \dots, q_{E+2,R}\} \cup \dots \cup \{q_{R-1,R}\}; \quad (5)$$

Initially, for a given graph G , we suppose that a graph is completely connected, i.e., we assign to each node N_i ($i < n$), $n - 1$ edges enumerated by $\{x_k\}$. Thus, each variable x_k is considered a Boolean variable with value $\{0, 1\}$. This manner of operating will help us later to generate all the instances of the graph G . We design by a_u ($1 \leq u \leq 2n$) the real existing edges of an instance of the graph defining the input edges. The object k_0 is an object that will take the size of the maximal clique size in each level of each instance, and q_{gh} represents the possibly cliques of three nodes in each level that will be transformed in $\bar{q}_{gh}$ if its edges do not appear in the given base on the problem.

The set $\{p, p^E, p^{ZE}, p^{ZE+1}, \dots, p^R, h_{ij}, h_g\}$ is collection of the initial enzymes associated to different layers in the system, which controlled the evolution of objects and/or layers. The enzymes can be also changed during their control base on the following set of rules that applied in a maximal parallel approach:

Ri:

$$\begin{aligned} [p]_j &\xrightarrow{h} [p^h]_j \xrightarrow{i} [p^i]_j, E \leq i \leq R; \\ [p]_i &\xrightarrow{h} [p^h]_i \xrightarrow{i} [p^i]_i, 1 \leq i \leq n-1, 2 \leq j \leq n-1, E \leq h \leq R. \end{aligned} \quad (6)$$

$$1 \leq i \leq n-1, 2 \leq j \leq n-1, E \leq h \leq R.$$

The first rule is based on the weak division rule for elementary layer, while the second rule is assigned to a strong division for a non-elementary layer. These rules will be used to generate all of the given graph G under the control of the enzyme p_i ; where E and R are the limits min and max of the interval of Boolean variables x_i associated with the layer j , where:

$$E = (j-1)n - [(j-1) + (j-2) + \dots + (j-j)] + 1; \quad (8)$$

$$R = E + n - (j+1). \quad (9)$$

$$\mathbf{R_2:} \quad \begin{matrix} \mathbf{h} & \mathbf{i} \\ x_i | p_i^l & \end{matrix} \rightarrow [t_i | p_{i+1}]_j : \quad (10)$$

$$E \leq i \leq R-1 \text{ and } 1 \leq j \leq n-1;$$

$$\text{for } i = R :$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} \\ x_i | p_i^k & \end{matrix} \rightarrow [t_R | p_e]_j \quad (11)$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} & \mathbf{h} & \mathbf{i} \\ x_i | p_i^u & \rightarrow & f_i p_{i+1} & \end{matrix} \quad (12)$$

$$E \leq i \leq R \text{ and } 1 \leq j \leq n-1;$$

$$\text{for } i = R :$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} \\ x_i | p_i^u & \end{matrix} \rightarrow \begin{matrix} \mathbf{h} & \mathbf{i} \\ f_R p_e & \end{matrix} \quad (13)$$

$$[s | p_e]_j \rightarrow [| p_s]_j \quad (14)$$

After the execution of the rule R_1 , the rules (11) and (12) can provide the state of the variable x_i by their transformation into the object t_i (true) or into f_i (false) meaning that the edge represented by this variable exists or not in this instance according to the enzyme presented in the layer. In all the instances where s exists, s indicates that this instance is not the instance that we aim to found its maximal clique. So in this moment, p_e is changed to p_s by applying the rule (14) that means the evolution of this subsystem is suspended.

$$\mathbf{R_3:} \quad \begin{matrix} \mathbf{h} & \mathbf{i} & \mathbf{h} & \mathbf{i} \\ f_i h_{jg} & \rightarrow & f_j h_{j(g-1)} & \end{matrix} \quad (15)$$

$$E \leq i \leq R, 1 \leq j \leq n-1 \text{ and } n-j \leq g \leq 1;$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} & \mathbf{h} & \mathbf{i} \\ f_j z_i & \rightarrow & | z_i^u & \end{matrix} \quad (16)$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} & \mathbf{h} & \mathbf{i} \\ t_i z_i & \rightarrow & | z_i^l & \end{matrix} :$$

$$E \leq i \leq R, \text{ and } 1 \leq j \leq n-1$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} & \mathbf{h} & \mathbf{i} \\ a_i | z_i^l & \rightarrow & | z_i^l & \end{matrix} \quad (17)$$

$$E \leq i \leq R-1;$$

$$\begin{matrix} \mathbf{h} & \mathbf{i} & \mathbf{h} & \mathbf{i} \\ a_i | z_i^u & \rightarrow & s | z_i^u & \end{matrix} \quad (18)$$

The role of the rule (15) tends to be decreased the size of the possibility maximal clique in each level represented by the index g and having the enzyme h_{jg} . When the marker f_i appeared in layer j , it signifies that two nodes at this level are not linked. Rule (16) will provide the specified objects and changes the enzymes z_i according to the value of x_i : thus, the enzyme z_i^l will indicate that the value of x_i is true and respectively z_i^u will indicate that the value of x_i is false. The rule (17) and (18) are used to check whether the current instance is the correct instance in the graph instance where the maximal clique is aimed.

$$R_4: \quad \begin{array}{c} h \quad i \\ q_{gh} z_{h-j}^w \rightarrow q_{gh} z_{h-j}^i \end{array} \quad (19)$$

$$E \leq g, h \leq R \text{ and } 1 \leq j \leq n-1;$$

$$\begin{array}{c} h \quad i \\ q_{gh} p_e \rightarrow q_{gh}^w [p_e]_j \end{array} \quad (20)$$

$$E \leq w \leq R \text{ and } 2 \leq j \leq n-1;$$

$$w = i \times n - [(i + (i-1) + (i-2) + \dots + (i-i)) - (h-g)];$$

$$\begin{array}{c} \begin{array}{c} i \\ q_{gh}^w z_{h-j}^w \rightarrow q_{gh}^w z_{h-j}^i \end{array} \\ \begin{array}{c} v = g - E \\ i = v + j \end{array} \end{array} \quad (21)$$

$$\begin{array}{c} i \\ q_{gh}^w z_{h-j}^w \rightarrow q_{gh}^w z_{h-j}^i \end{array} \quad (22)$$

$$\begin{array}{c} q_{gh}^w p_e \rightarrow q_{gh}^w [p_e]_j \end{array} \quad (23)$$

$$E \leq R \leq w \text{ and } 2 \leq j \leq n-1$$

This collection of rules are introduced to check the existence of q_{gh} (which indicate the relation between the two edges x_g and x_h). Hence, according to our definition, to check that q_{gh} exists in an instance, we must firstly verify that x_g and x_h exist (i.e., they have true values), and whether this situation is true, passing on to verify their linkage. Rule (19) is devoted to eliminate all the possibly cliques of three nodes represented by q_{gh} having at least one of their extremities x_g or x_h assigned to the false value (the test is based on the presence of enzymes z_{h-j}^w or z_{h-j}^i). In this case, the object q_{gh} will be destroyed, however, in the opposite case, the rule (20) may be applied, which consists of changing and moving q_{gh} to the outer layer. This is done in order to apply a second check concerning the existence of x_w . Clearly, we say that the object q_{gh} maybe introduced by rule (22) whether x_w is absent, otherwise it has a false assignment. The rule (23) sends down q_{gh}^w until it reaches the layer on the enzyme z_{h-j}^w or z_{h-j}^i .

R5:

$$\begin{array}{c} [k_0 | p_e]_i \rightarrow [k | p_e]_i \end{array} \quad (24)$$

$$\begin{array}{c} k | h_{ij} \rightarrow k_{ij} [h_{i-1}]_i \end{array} \quad (25)$$

$$\begin{array}{c} q_{gh} h_j \rightarrow q_{gh} h^j \\ k_{jl} h_{j-i} \rightarrow k_{j(l-1)} [h_{j-1}]_i \end{array} \quad (26)$$

$$\begin{array}{c} k_{jl} h_{j-i} \rightarrow k_{j(l-1)} [h_{j-1}]_i \end{array}$$

$$\begin{array}{c} h \quad i \quad h \quad i \\ k_{jg} h_{j-i} \rightarrow k_{jg} [h_{j-1}]_i, 2 \leq i \leq n-1 \end{array} \quad (27)$$

This set of rules transmit and update the size of the probable maximal clique at each level according to the objects q_{gh} , until when they reached the layer c_1 , where:

$$E \leq g, h \leq R; n \leq l \leq 1; 2 \leq i \leq n-1; \text{ And } n-1 \leq j \leq 1$$

$$E = (j-1)n - [(j-1) + (j-2) + \dots + (j-j)] + 1$$

$$R = E + n - (j+1)$$

R6:

$$[h_0]_j \rightarrow \lambda; \quad (28)$$

This rule is part of layer dissolution rule for $1 \leq j \leq n - 1$ whether it is controlled by the enzyme h_0 , otherwise it can say that it be considered just as a cleaning rule; λ design empty string, where h_0 is appeared when all the possibly tests of the objects q_{gh} in this level are completed.

$$\begin{array}{c} \text{R7:} \\ h \quad i \quad h \quad i \\ k_{ij} h_i \rightarrow k_{ij} h_{ij} \quad c_i, \quad c_1 \leq c_i \leq c_{n-1}; \end{array} \quad (29)$$

$$\begin{array}{c} h \quad i \\ k_{gh} h_{ij} \rightarrow k_{gh}, \quad \text{if } j < h; \\ 1 \leq g \leq n - 1 \end{array} \quad (30)$$

$$\begin{array}{c} h \quad i \quad h \quad i \\ k_{gh} h_{ij} \rightarrow k_{gh}^j | h_{ij} \quad c_i, \quad \text{if } h < j; \end{array} \quad (31)$$

Rules (30) and (31) will be executed only if the specified condition is accomplished. In rule (29) the enzyme h_i in c_i takes the maximal clique at this level represented by the index j . The layer c_i can be dissolved by the execution of rule (30) in the case where the maximal clique is carried in the next level (below). Rule (31) transforms and moves the specified object that will be used later for eliminating some levels.

$$k^j h_i \quad ij \quad c_i \rightarrow k_{ij}^j \quad (32)$$

This set of rules consists to research the level where the maximal clique of the given graph is found. If there are several maximal cliques in different levels, the application of these rules enables at least one to be found from them.

R8:

$$\bar{q}_{gh} p_{c_i} \rightarrow | p_{g \quad c_i}^j | p_{h \quad c_i}^i; \quad (33)$$

$$[[|]_{c_i} [|]_{c_i} p]_n \rightarrow [[|]_{c_i} p]_n [[|]_{c_i} p]_n; \quad (34)$$

$$\begin{array}{c} h \quad i \\ q_{ij} p_{c_i}^j \rightarrow [| p]_{c_i}; \end{array} \quad (35)$$

In this part of our model, we will generate all the sets of cliques of three nodes which are able to be put together. This is done in order to construct a clique at a given level based on cliques of previous parts of our system.

R9:

$$\begin{array}{c} h \quad i \quad h \quad i \\ k_{ij} p_{c_i} \rightarrow k_{ij} | p_{c_i}^u; \end{array} \quad (36)$$

$$u = 3(j - 3);$$

$$\begin{array}{c} h \quad i \quad h \quad i \\ q_{gh} p_{c_i}^u \rightarrow N_i, N_j, N_l | p_{c_i}^{u-1}; \end{array} \quad (37)$$

$$\begin{cases} j = (g - E) + (i + 1) \\ l = (h - E) + (i + 1) \end{cases};$$

$$\begin{array}{c} E = (i - 1)n - [(i - 1) + (i - 2) + \dots + (i - i)] + 1 \\ h \quad i \\ k_{ij} | p_{c_i}^0 \rightarrow k_{ij}; \end{array} \quad (38)$$

$$\begin{array}{c} h \quad i \quad h \quad i \\ k_{ij} | p_n \rightarrow | p_n^i; \end{array} \quad (39)$$

These rules hold all sets of nodes where their numbers are equal to the size of the maximal clique, and eliminate all the others. When the set of nodes of the resulting maximal clique found in the layer are labeled with n , the role of the dissolution layer rule (38) comes after its execution, and the provided result reach the outer layer labeled with n . Finally, rule (39) evolves the enzyme p to take the maximal clique.

4. Numerical Example and Sensitivity Analysis

First of all, the utilized parameters for experimental design and their surfaces are described; subsequently the statistical results of modeling with upper and lower limit and the alpha cut method are determined; finally, each parameter's influence on objective function is illustrated by sensitivity analysis.

Table 1 indicates the utilized parameters for experiments and their numerical surfaces considering the research of Konc et al. in [8],

$$\Pi = (O, P, \mu, w_1/P_1, \dots, w_m/P_m, R_1, \dots, R_m, i_0) \quad (40)$$

where:

- (a) O is an alphabet; its elements are called components;
- (b) P is an alphabet; its sets are called enzymes, $P_1, \dots, P_m$ are the collection of enzymes on the m layers of μ ;
- (c) μ is a layer structure consists of m layers, m constitute also the level of Π ;
- (d) $w_i, 1 \leq i \leq m$, are collection of strings which part of the initial multisets over V associated with the regions $1, 2, \dots, m$ of μ ;
- (e) $R_i, 1 \leq i \leq m$, are sets of rules in a maximal parallel manner that associated to the regions $1, 2, \dots, m$ of μ ;
- (f) i_0 , is a indexing between 1 and m which shown on the output layer of Π .

Table 1. Parameter levels.

Parameter	Parameter Surfaces
O	2, 8, 14, 20
p	0.01, 0.04, 0.07, 0.1
μ	4 M, 14 M, 22 M, 30 M
w_i	3 K, 12 K, 21 K, 30 K
R_i	6 M, 14 M, 22 M, 30 M
i_0	2, 4, 6, 8
i	1, 2, 3, 4

Employing Design Expert software and response surface methodology, this has shed light on the parameters' impact on the proposed model. Based on Konc et al. in [8] regarding the demand to be fuzzy, considering an alpha $\alpha = 0.1$ tolerance Figure 4 presents objective function as a function of parameters, O and P , respectively.

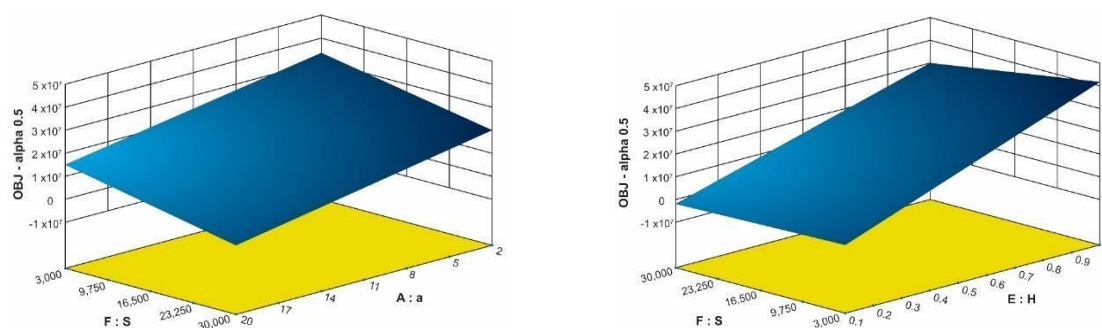


Figure 4. Objective function diagram as a function of (O and P) parameters with lower demand limit and $\alpha = 0.1$.

Furthermore, Figure 5 pictures the objective function based on parameter μ and w_i with lower demand limit and $\alpha = 0.1$.

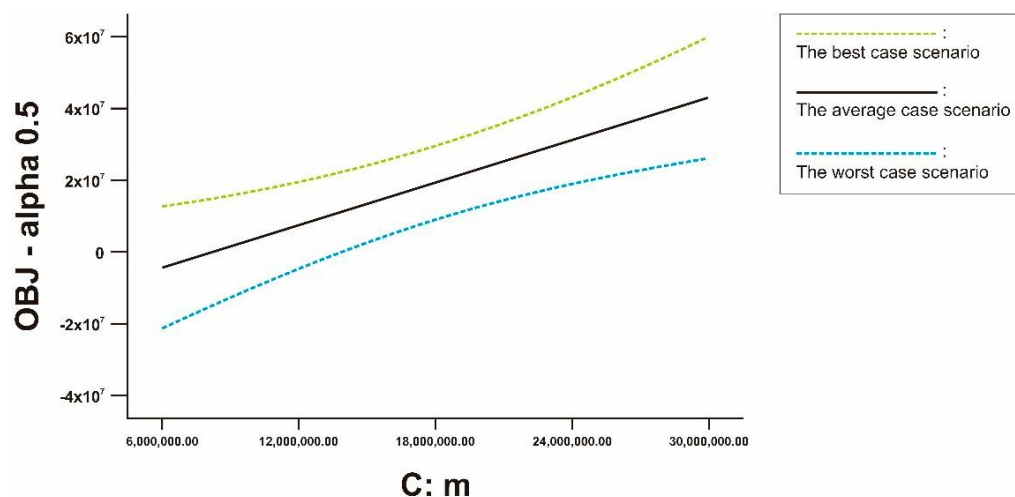


Figure 5. Objective function diagram as a function of parameter μ and w_i with lower demand limit and $\alpha = 0.1$.

As depicted, escalation in enzymes size has a positive correlation with the objective function. We also noted that a collection of P systems with active layers and enzymes used in this experiment already show the polynomial time $O(n(n - 1))$ (n is a number of nodes in the graph).

Moreover, in each problem parameter's coefficient function, response surface prediction of the mentioned issue is explicit which is indicated in Table 2.

Table 2. The coefficients of the parameters in the surface estimation function with a lower demand level and $\alpha = 0.1$.

Parameter	Parameter Coefficient in Surface Estimation Function
O	-9.52
P	-466.62
μ	-0.32
w_i	-454.55
R_i	+1.97
io	-30.8
i	+3.70

5. Conclusions

In this paper we elaborated a P system using the enzymes on layers to determinate the maximal clique in a given graph, where the objects and the layers evolved under the control of enzymes. The experimental result also showed that the main type of enzyme considered is an enzyme that controlled the evolution of objects and layers, and was also evolving itself. The constructed system showed that a collection of P systems with active layers and enzymes are used to determine polynomial time $O(n(n - 1))$ (n is a number of nodes in the graph). In this experiment, in order to search the maximal clique in a given graph, the enumeration will be started by generating all the possible instances of a graph G , then applying a validation step that takes into account only the result of the input instance of the graph, by determining exactly the embodied level of the maximal clique.

There will be some future works deriving from this research such as parallelization of the algorithm and additionally some constraints on the problems.

Author Contributions: Conceptualization, F.L.G.; Methodology, F.L.G.; Validation, T.M.; Formal analysis, F.L.G.; Investigation, F.L.G. and T.M.; Resources, F.L.G.; Data curation, T.M.; Writing—original draft preparation, F.L.G.;

writing—review and editing, F.L.G. and T.M.; Visualization, F.L.G.; supervision, F.L.G.; project administration, T.M.; funding acquisition, T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We would like to thanks for the anonymous reviewers that provide valuable feedback on our paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Păun, A.; Popa, B. P systems with proteins on membranes. *Fundamenta Informaticae* **2006**, *72*, 46–483.
2. Păun, G. *Introduction to Membrane Computing*; Springer: Berlin/Heidelberg, Germany, 2002.
3. Păun, G. Membrane Systems: A Quick Introduction. In Proceedings of the 1st International Workshop Unconventional Programming Paradigms—UPP 2004, Le Mont Saint Michel, France, 15–17 September 2004; pp. 188–195.
4. Cardelli, L. Brane calculi-interactions of biological membranes. In *Computational Methods in Systems Biology*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 257–280.
5. Păun, A.; Popa, B. *P Systems with Proteins on Membranes and Membrane Division*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 292–303.
6. Sosík, P.; Păun, A.; Rodríguez-Patón, A. P systems with proteins on membranes characterize PSPACE. *Theor. Comput. Sci.* **2013**, *488*, 78–95. [[CrossRef](#)]
7. Sosík, P.; Păun, A.; Rodríguez-Patón, A.; Pérez, D. On the Power of Computing with Proteins on Membranes. *Membr. Comput. Lect. Notes Comput. Sci.* **2010**, *5957*, 448–460.
8. Pardalos, P.M.; Rappe, J.; Resende, M.G.C. An Exact Parallel Algorithm for the Maximum Clique Problem. In *High Performance Algorithms and Software in Nonlinear Optimization*; de Leone, R., Murli, A., Pardalos, P.M., Toraldo, G., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1998; pp. 279–300.
9. Kumlander, D. A simple and efficient algorithm for the maximum clique finding reusing a heuristic vertex colouring. *IADIS Int. J. Comput. Sci. Inf. Syst.* **2006**, *1*, 32–49.
10. Konc, J.; Janežič, D. An improved branch and bound algorithm for the maximum clique problem. *MATCH Commun. Math. Comput. Chem.* **2007**, *58*, 569–590.
11. Pullan, W.; Hoos, H.H. Dynamic Local Search for the Maximum Clique Problem. *J. Artif. Intell. Res.* **2006**, *25*, 159–185. [[CrossRef](#)]
12. Shinano, Y.; Fujie, T.; Ikebe, Y.; Hirabayashi, R. Solving the Maximum Clique Problem Using PUBB. In Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing, Orlando, FL, USA, 30 March–3 April 1998; pp. 326–332.
13. Ouyang, Q.; Kaplan, P.D.; Liu, S.; Libchaber, A. DNA Solution of the Maximal Clique Problem. *Science* **1997**, *278*, 446–449. [[CrossRef](#)] [[PubMed](#)]
14. Păun, G. Bio-inspired Computing Paradigms. In Proceedings of the 2004 International Conference on Unconventional Programming Paradigms, Le Mont Saint Michel, France, 15–17 September 2004; pp. 155–160.
15. Păun, A.; Păun, G. The power of communication: P systems with symport/antiport. *New Gener. Comput.* **2002**, *20*, 295–306. [[CrossRef](#)]