International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com

Solving NP Complete Problems Using KP System

Ajitha P S* and Dersanambika K.S †

*Department of Mathematics,

College Of Engineering,Kottarakara, Kerala 691531, India

† Department of Mathematics,

Fatima Mata National College, Kollam 691 001, India

Keyword:

1. Psystem
2. KP System
3. Independent set
4. Clique.

Abstract

In this paper we have produced a polynomial time solution for independent set problem and clique problem.

1 Introduction

The research area of Membrane Computing originated as an attempt to fomulate a model of computation motivated by the structure and functioning of a living cell. The initial models were based on a cell like arrangement of membrane delimiting compartments were multiset of chemicals evolved according

*email- psajithamurali@gmail.com
†email-dersanapdf@yahoo.com

to given evolution rules. Many variants of P System have been studied so far

Kernel P System (KP System) represents a new class of membrane systems, use well known features of P System and also include some new elements, in order to provide a coherent and comprehensive description of various classes of P System and allow the possiblity of integrating different types of P Systems into same fomalism.Kernel P System includes rules of two types
1) Object Processing rules, which transform and move objects across compartments, they include rewriting and communication rules guarded by promoters and inhibitors and also symport/ antiport like rules.
2) System sructure rules which change the topology of the system and include membrane division and disolution and also creation of membranes.
The system was introduced by Marian Gheorghe etal. Here a broad range of strategies to use the rule against the multiset of objects available in each compartments is provided. Now we will see the definition of compartments and KP System

Definition 1.1. *Given a finite set A called alphabet of elements, called objects and a finite set L , of elements called labels , a compartment is a tuple* $C = (l, \omega_0, R^\sigma)$ *where* $l \in L$ *is the label of the comparments,* ω_0 *is the intial multiset over A and* R^σ *denotes the DNA code of C, which comprises the set of rules, denoted by R , applied in this comparments and a regular expression* $\sigma, overLab(R)$ *the labels of the rule of R*

Definition 1.2. *A kernel P System of degree n is a tuple*
$K\Pi = (A, L, \mu, C_1, C_2, ..., C_n, i_0)$
where A and L are as in definition 1.1, the alphabet and the set of labels respectively; μ *defines the membrane structure which is a graph* (V, E) *, where V are vertices,* $V \subseteq L$ *(the nodes are labels of these comparments) , and E edges,* $C_1, C_2, ..., C_n$ *are n compartments of the system - the inner part of each compartment is called region , which is delimited by a membrane , the labels of the comparments are from L and initial multiset are over A .* i_0 *is the output*

compartment where the result is obtained .

2 NP Complete Problems

A decision problem C is NP Complete if
1) C is in NP and
2) Every problem in NP is reducible to C in polynomial time.
3) C can be shown to be NP by demonstrating that a candidate solution to C can be verified in polynomial time

The abbreviation NP refers to non-deterministic polynomial time. Any given solution to an NP Complete problem can be verified in polynomial time. Here we will consider two particular NP complete problems
- the independent set problem and
- the clique problem.

2.1 The independent set problem

An independent set or stable set is a set of vertices in a graph, no two of which are adjacent. That is it is a set S of vertices in S such that for every two vertices in S, there is no edge connecting the two. Equivalently each edge in the graph has atmost one end point in S. The size of the independent set is the number of vertices it contains.

A maximal independent set is an independent set such that adding any other vertex for the set forces the set to contain an edge

2.2 Solution by KP System

Consider a KP System $K\Pi = (A, L, I_0, \mu, C_1, C_2, 0)$ where A is the alphabet $L = \{1, 2\}$ is the label of the compartments. I_0 consists of yes or no at the end. After atmost $n + 4$ steps one of the two possible answers will be send out
$C_1 = (1, \omega_{1,0}, R_1)$
$C_2 = (2, \omega_{2,0}, R_2)$
where $\omega_{1,0} = S$
$\omega_{2,0} = A_1 S$
V_i is a collection of vertices of the graph. An edge $\{i, j\}$ is codified A_{ij} , $1 \leq i \leq j \leq n$, μ is a membrane structure given by the graph with edge $1, 2$ the rules $R_1$and R_2 are given below
R_1 contains
$r_{11} : S \rightarrow (yes, 0)\{\geq T\}$
$r_{12} : S \rightarrow (no, 0)\{\geq F \geq \overline{T}\}$
R_2 contains membrane division rules
$r_{2,i} : [A_i]_2 \rightarrow [V_i A_{i+1}]_2\ [A_{i+1}]_2$, $1 \leq i \leq n - 1$
$r_{2,n} : [A_n]_2 \rightarrow [V_n X_1]_2\ [X_1]_2$
These rules generate in n steps all the subsets of V (2^n subsets) each of them being a potential subset with atmost n vertices
Rewriting rules
$r_{2,n+1} : X_1 \rightarrow X_2$
$r_{2,n+2} : S \rightarrow \lambda\{V_i \cap V_j = A_{ij}\}$
$\mathrm{r}_{2,n+3} : X_2 \rightarrow X_3$
$r_{2,n+4} : SX_3 \rightarrow (T, 1)$
$\mathrm{r}_{2,n+5} : X_3 \rightarrow (F, 1)\{= \overline{S}\}$
The computaion leads to an answer yes or no in n+4 steps. Indeed in n steps there are generated all the subset of V.
In the $(n+1)^{th}$ step X_1 is transformed to X_2. In the $(n+2)^{th}$ step for any two vertices V_i, V_j there is an edge A_{ij} connecting these two vertices are found,then S is transformed to λ

By the above steps we can ensure that no edge has been formed.Then X_2 is transformed to X_3.In $(n+3)^{rd}$ step all vertices present along with SX_3 in compartment 2 will represent independent sets and the largest set represent the maximal independent set. Then we have a solution in $(n+4)^{th}$step, otherwise an F is send instead.In the final step the answer is sent to the environment from the compartment C_1 by using one of the rules $r_{11}, r_{1,2}$

2.3 Clique Problem

In computer science, the clique problem is the computational problem of finding Cliques (subsets of vertices,all adjacent to each other, also called complete subgraph) in a graph. Common formulation of clique problem include finding a maximum clique (a clique with the largest possible number of vertices)
The clique problem arises in the following real world setting. Consider a social network, where the graphs vertices represent people ,and the edge represents mutual aquaintance. Then a clique represents a subset of people who all know each other
The clique problem also has many applications in bio informatics and Computational Chemistry.

2.4 A solution using KP System

Consider a KP System
$KP = (A, L, I_0, \mu, C_1, C_2, 0)$
Where A is the alphabet
$L = \{1, 2\}$
I_0 consists of yes or no at the end. After atmost $n + 4$ steps one of the two possible answers will be sent out.
$C_1 = (1, \omega_{1.0}, R_1)$
$C_2 = (2, \omega_{2,0}, R_2)$ where

$\omega_{1,0} = S$
$\omega_{2.0} = A_1 S$
V_i is a collection of vertices of the graph. An edge (i, j) is codified as $A_{ij}, 1 \leq i \leq j \leq n$
μ is the membrane structure $[_0[_1[_2]_2]_1]_0$
The rules R_1 and R_2 are given below
R_1 contains
$r_{11} : S \rightarrow (yes, 0)\{\geq T\}$
$r_{12} : S \rightarrow (n_0, 0)\{\geq F \geq \overline{T}]$
R_2 contains membrane division rules
$r_{2,i} : [Ai]_2 \rightarrow [V_i A_{i+1}]_2 [A_{i+1}]_2, 1 \leq i \leq n-1$
$r_{2,n} : [An]_2 \rightarrow [V_n X_1]_2 [X_1]_2$
these rules generate in n steps all the subsets of V (2^n subsets) each of them being a potential subset with atmost n vertices.

Rewriting rules

$$
\begin{aligned}
r_{2,n+1} : X_1 \rightarrow X_2 \; \{ & V_i \cap V_j = A_{ij} \cup \\
& V_i \cap V_j \cap V_k = A_{ij} = A_{ik} = A_{jk} \;\cup \\
& \qquad \qquad ... \cup \\
& V_1 \cap V_2 \cap V_3 \cap ... \cap V_n = A_{12} = A_{13} ... = A_{1n} = A_{23} = A_{24} \\
& = ... = A_{2n} = A_{34} = A_{35} = ... = A_{3n} = ... = A_{n-1\;n}\}, \\
& \qquad \qquad \qquad \qquad 1 \leq i \leq j \leq k \leq n
\end{aligned}
$$

$$
\begin{aligned}
r_{2,n+2} : S \rightarrow \lambda \; \{ & V_i \cap V_j \neq A_{ij} \cup \\
& V_i \cap V_j \cap V_k \neq A_{ij} \neq A_{ik} \neq A_{jk} \cup \\
& \qquad \qquad ... \cup \\
& V_1 \cap V_2 \cap V_3 \cap ... \cap V_n = A_{12} \neq A_{13} \neq ... \neq A_{1n} \\
& \neq A_{23} \neq A_{24} \neq ... \neq A_{2n} \neq A_{34} \neq A_{35} \neq ... \neq A_{3n} \neq ... \\
& \neq A_{n-1\;n}\}, 1 \leq i \leq j \leq k \leq n
\end{aligned}
$$

$r_{2,n+3} : X_2 \to X_3$

$r_{2,n+4} : SX_3 \to (T, 1)$

$r_{2,n+5} : X_3 \to (F, 1)\{\overline{S}\}$

In $(n+3)^{rd}$ step the vertices in the power set present along with SX_3 in the second compartment will represent all cliques.The maximum number of vertices present in the power set represent the maximum clique. In $(n+4)^{th}$ step a T will be send to the first comparment.
If all the vertices are isolated vertices ,then there is no clique.In that case in $(n+5)^{th}$ step an F is transferred to compartment 1.
In the final step the answer is sent to the enviornment from the comparment C_1 using the rules r_{11}, r_{12}.

3 Illustration

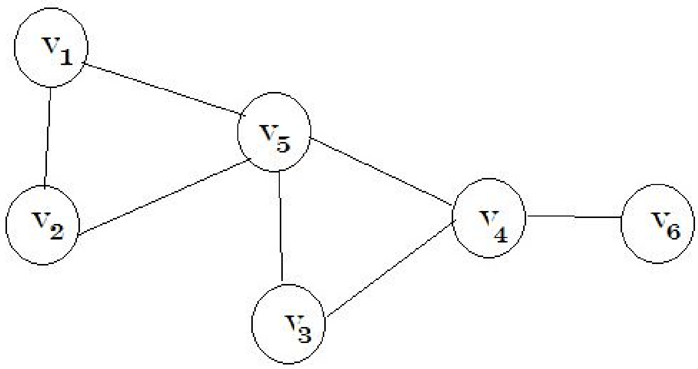

Figure 1:

Consider the above graph and taking the rules as same as in section 2.2 and 2.4
According to the given graph R_2, the membrane division rule in both cases

can be written as
$r_{2,1}$: $[SV_1A_2], [SA_2]$

$r_{2,2}$: $[SV_1V_2A_3]$,$[SV_1A_3]$,$[SV_2A_3]$, $[SA_3]$

$r_{2,3}$: $[SV_1V_2V_3A_4]$,$[SV_1V_2A_4]$,$[SV_1V_3A_4]$,$[SV_1A_4]$,$[SV_2V_3A_4]$,$[SV_2A_4]$,$[SV_3A_4]$,$[SA_4]$

$r_{2,4}$: $[SV_1V_2V_3V_4A_5]$,$[SV_1V_2V_3A_5]$,$[SV_1V_2V_4A_5]$, $[SV_1V_2A_5]$,$[SV_1V_3V_4A_5]$,$[SV_1V_3A_5]$, $[SV_1V_4A_5]$,$[SV_1A_5]$,$[SV_2V_3V_4A_5]$,$[SV_2V_3A_5]$,$[SV_2V_4A_5]$,$[SV_2A_5]$, $[SV_3V_4A_5]$,$[SV_3A_5]$, $[SV_4A_5]$,$[SA_5]$

$r_{2,5}$: $[SV_1V_2V_3V_4V_5A_6], [SV_1V_2V_3V_4A_6], [SV_1V_2V_3V_5A_6], [SV_1V_2V_3A_6], [SV_1V_2V_4V_5A_6], [SV_1V_2V_4A_6], [SV_1V_2V_5A_6], [SV_1V_2A_6], [SV_1V_3V_4V_5A_6], [SV_1V_3V_4A_6], [SV_1V_3V_5A_6], [SV_1V_3A_6], [SV_1V_4V_5A_6], [SV_1V_4A_6], [SV_1V_5A_6], [SV_1A_6], [SV_2V_3V_4V_5A_6], [SV_2V_3V_4A_6], [SV_2V_3V_5A_6], [SV_2V_3A_6], [SV_2V_4V_5A_6], [SV_2V_4A_6], [SV_2V_5A_6], [SV_2A_6], [SV_3V_4V_5A_6], [SV_3V_4A_6], [SV_3V_5A_6], [SV_3A_6], [SV_4V_5A_6], [SV_4A_6], [SV_5A_6], [SA_6]$

$r_{2,6}$: $[SV_1V_2V_3V_4V_5V_6X_1], [SV_1V_2V_3V_4V_5X_1], [SV_1V_2V_3V_4V_6X_1], [SV_1V_2V_3V_4X_1], [SV_1V_2V_3V_5V_6X_1], [SV_1V_2V_3V_5X_1], [SV_1V_2V_3V_6X_1], [SV_1V_2V_3X_1], [SV_1V_2V_4V_5V_6X_1], [SV_1V_2V_4V_5X_1], [SV_1V_2V_4V_6X_1], [SV_1V_2V_4X_1], [SV_1V_2V_5V_6X_1], [SV_1V_2V_5X_1], [SV_1V_2V_6X_1], [SV_1V_2X_1], [SV_1V_3V_4V_5V_6X_1], [SV_1V_3V_5X_1], [SV_1V_3V_4V_6X_1], [SV_1V_3V_4X_1], [SV_1V_3V_5V_6X_1], [SV_1V_3V_5X_1], [SV_1V_3V_6X_1], [SV_1V_3X_1], [SV_1V_4V_5V_6X_1], [SV_1V_4V_5X_1], [SV_1V_4V_6X_1], [SV_1V_4X_1], [SV_1V_5V_6X_1], [SV_1V_5X_1], [SV_1V_6X_1], [SV_1X_1], [SV_2V_3V_4V_5V_6X_1], [SV_2V_3V_4V_5X_1], [SV_2V_3V_4V_6X_1], [SV_2V_3V_4X_1], [SV_2V_3V_5V_6X_1], [SV_2V_3V_5X_1], [SV_2V_3V_6X_1], [SV_2V_3X_1], [SV_2V_4V_5V_6X_1], [SV_2V_4V_5X_1], [SV_2V_4V_6X_1], [SV_2V_4X_1], [SV_2V_5V_6X_1], [SV_2V_5X_1], [SV_2V_6X_1], [SV_2X_1], [SV_3V_4V_5V_6X_1], [SV_3V_4V_5X_1], [SV_3V_4V_6X_1], [SV_3V_4X_1], [SV_3V_5V_6X_1], [SV_3V_5X_1], [SV_3V_6X_1], [SV_3X_1], [SV_4V_5V_6X_1], [SV_4V_5X_1], [SV_4V_6X_1], [SV_4X_1], [SV_5V_6X_1], [SV_5X_1], [SV_6X_1], [SX_1]$

Hence we get 2^6 subsets
The rewriting rules for both problems can be considered seperately.

CliqueProblem − Rewriting rules

$r_{2,7}$: $[SV_1V_2X_2], [SV_1V_5X_2], [SV_2V_5X_2], [SV_3V_5X_2], [SV_3V_4X_2], [SV_4V_5X_2], [SV_4V_6X_2], [SV_1V_2V_5X_2], [SV_3V_4V_5X_2]$

In the above step X_1 is transformed to X_2, where the edges that forms the complete subgraphs.

$r_{2,8}$: $S \rightarrow \lambda$ in all other elements in $r_{2,6}$ that are not in $r_{2,7}$

$r_{2,9}$: $[SV_1V_2X_3], [SV_1V_5X_3], [SV_2V_5X_3], [SV_3V_4X_3], [SV_4V_5X_3], [SV_3V_5X_3], [SV_4V_6X_3], [SV_1V_2V_5X_3], [SV_3V_4V_5X_3]$

In the above step X_2 is transformed into X_3. This is applicable for elements that contains X . These represents cliques

Here $n = 6$, in $(n + 3)^{rd}$ step , that is in the 9^th step, we get all cliques. Here the maximum cliques are
$[V_1V_2V_5]$ and $[V_3V_4V_5]$

That is they are the sets $\{V_1, V_2, V_5\}$ and
$\{V_3, V_4, V_5\}$

IndependentSetProblem − Rewriting rules

In R_2 the rules upto $r_{2,6}$ are as same as in click problem.

$r_{2,7}$: All X_1 are transformed to X_2
$r_{2,8}$: Every vertices that are end points of the same edges vanishes when we apply the rule $r_{2,8}$.
$r_{2,9}$: All vertices which do not forms an edge remains, which is an independent set

From our example we get the independent set of vertices as
$\{V_1\}$, $\{V_2\}$,$\{V_3\}$,$\{V_4\}$,$\{V_5\}$,$\{V_6\}$,$\{V_1, V_3\}$ $\{V_1, V_4\}\{V_1, V_6\}\{V_2, V_3\}\{V_2, V_4\}\{V_2, V_6\}\{V_3, V_6\}\{V_5, V_6\}$ $\{V_1, V_3, V_6\}$ $\{V_2, V_3, V_6\}$
The maximal independent sets are $\{V_1, V_3, V_6\}$ $\{V_2, V_3, V_6\}$

In the 9^{th} steps we get the answer

4 Conclusion

In this paper we have generated a polynomial time solution for both independent set problem and the clique problem. Both solutions are linear

References

[1] Ajitha P S and Desanamabika K S,Regulated Contextual KP Systems,International Jounrnal Of Mathematics Trends and Techonologies,Vol 34,Part 11 , June 2016

[2] Ajitha P S and Desanamabika K S,Contextual KP Systems with minimal/maximal use of selectors,International Jounrnal Of Computer Trends and Techonologies,Vol 28,issue 1, October 2015

[3] Ajitha P S and Desanamabika K S,Contextual KP Systems,International Jounrnal Of Mathematics Trends and Techonologies,Vol 17,issue 1 January 2015

[4] Gh. Paun, Marcus Contextual Grammars ,Kluwer Academic publishers, 1997

[5] Gheorghe Păun, *Membrane Computing An Introduction*, Springer–Verlag, Berlin, 2002.

[6] Madhu Mutyam, Kamala Krithivasan, Contextual P Systems, Fundamenta Informaticae, 2002

[7] Marian Gheorghe, Florentin Ipate, Ciprian Dragomir, Laurentiu Mierla, LuisValencia-Cabrera, Manuel Garcia-Quismondo,and Mario J, Kernel P System - Version 1, International Confrerence On Membrane Computing, 2013

[8] Marian Gheorghe Florentin Ipate , Ciprian Dragomir, Kernel P Systems - version 1 proceedings of the eleventh brain storming week on membrane computing,p97-124 , Sevilla,E T S de ingeniera informatica,428 de Febrero, 2013.

[9] Georghe Paun, Crzecorz Rozenberg,Arto Salomaa,The Oxford Hand book of Membrane computing, Oxford University press, Oxford,Newyork,2010

[10] M. Gheorghe, Ciprian Dragomir , L Valenua Cabera etal, (Simple)Kernel PSystems - examples 2012.