Communication Complexity of EC P Systems with Energy and Sevilla Carpet

Richelle Ann B. Juayong
Algorithms and Complexity Laboratory
University of the Philippines Diliman
Quezon City, Philippines
rbjuayong@up.edu.ph

Henry N. Adorna
Algorithms and Complexity Laboratory
University of the Philippines Diliman
Quezon City, Philippines
ha@dcs.upd.edu.ph

ABSTRACT
Sevilla Carpets are useful tools for analyzing time and space complexity of computations over P systems. In this paper, we explore the role of Sevilla Carpets, this time, in analyzing communication resources. In particular, we relate dynamical parameters for communication complexity, defined in a previous literature, in terms of Sevilla Carpets. The use of these carpets are evaluated by investigating computations over EC P systems with Energy solving problems.

1. INTRODUCTION
In theoretical computer science, Membrane Computing [7] is an unconventional computing paradigm which aims to provide a framework for designing compartmentalized computations inspired by the architecture of living cells. The models used for this paradigm are called P systems, introduced by Gheorghe Păun in 1998 [3]. The main ingredient of P systems is a collection of membranes that function as delimiters and passageways of objects. Over the years, different types of P systems have been proposed and evaluation of the time [5] and space [6] complexity of computations over these models are being standardized.

However, despite P systems being massively parallel and distributed computing devices, the development of a framework for analyzing communication complexity in this domain has not yet been attained. In 2003, this issue has been one of the open problems mentioned in [4]. As a preliminary work on this direction, [1] has provided definition of dynamical parameters that can be used to measure communication complexity. These measures were investigated in a new variant called EC P systems with Energy (ECPe) and is also expected to be extended to other variants of P systems. The parameters enumerated involves counting the number of communication steps, the number of communication rules and the total amount of energy utilized per communication.

In order to study the communication complexity of computations solving a problem, [1] also define an ECPe that solves the equality problem. The equality problem involves comparing a set of values, and checking if the values are all equal. Authors of the paper conjecture that for a successful computation, the maximum number of communication steps is directly proportional to the number of values being compared. Such claim is asked to be further evaluated by considering the number of communication rules. Sevilla Carpets have been mentioned as an aid in validating these claim.

Sevilla Carpets, as introduced in [2], are means to measure descriptional complexity of computations over P systems. They are inspired by the Szilard Language employed for evaluating time and space resources used for types of grammars mentioned in the Chomsky Hierarchy. Sevilla Carpet also serves as a tool for visualizing (as in [12]) computations in P systems. By calculating information as in weight, surface, height and average weight, quantitative information can be deduced giving insights on the resources used for the computation.

There are several types of Sevilla Carpets depending on the objective of its user. In this paper, we would like to address the issue mentioned in [1] which involves establishing connection between dynamical parameters for communication and Sevilla Carpets in order to evaluate goodness of communication over ECPe. We shall explore the communication effort for the example ECPe of the paper, used to solve the equality problem. Furthermore, we look at two ECPe solutions solving the same problem and, using Sevilla Carpets, compare the quality of communication in both solutions.

The outline of this paper proceeds as follows: in Section 2, we establish basic notations and define EC P systems with Energy, dynamical measures for communication complexity in ECPe and the solution to the equality problem. The main work is presented in Section 3. Conclusions are drawn out in Section 4.

2. DEFINITION

Before we define ECPe, we first give a brief overview of how we compute using P systems. We refer to [8] for a more detailed description of the basics of membrane computing.

As mentioned in Section 1, P systems have a set of membranes. The regions delimited by these membranes serve as placeholders of a multiset of objects. Aside from being transported (through transport rules), these objects may *evolve* through multiset-rewriting rules called evolution rules. When an object is involved in an evolution rule, we say that this object is 'consumed'. Rules in P systems are usually applied in a nondeterministic and maximally parallel manner. The nondeterminism is manifested when several rules can be applied to a single object. Since there can only be one rule consuming this object, the system picks out one among all applicable rules and evolves the object through this chosen one. Maximal parallelism, on the other hand, is a property of rule application since, at a single unit of time, everything that can evolve should evolve.

2.1 EC P systems with Energy (ECPe)

EC P systems, as introduced in [9], are combinations of two variants of P systems called Transition P systems [7] and P systems with Symport and Antiport [10]. A new variant of these model have been introduced in [1] to evaluate communication that are dependent on some energy produced from evolution rules. To do this, a special object e is introduced to the system to represent a quantum of energy. However, in the construct definition shown below, object e is not part of the *alphabet of objects*, because it has a different role in the system.

We use the definition for EC P system with Energy (ECPe) from [1], as follows,

Definition 1. *An EC P system with Energy is a construct of the form*

$$\Pi = (O, e, \mu, w_1, \ldots, w_m, R_1, R'_1, \ldots, R_m, R'_m, i_{out})$$

where:

(i) m pertains to the total number of membranes;

(ii) O is the alphabet of objects;

(iii) μ is the membrane structure which can be denoted by a set of paired square brackets with labels;

(iv) $w_1, ..., w_m$ are strings from O^* denoting the multiset of objects present in the regions bounded by membranes;

(v) $R_1, ..., R_m$ are a set of evolution rules associated with each region of membranes in μ;

- An evolution rule is of the form $a \rightarrow v$ where $a \in O$, $v \in (O \cup \{e\})^*$. In the event that this type of rule is applied, the object a transforms into a multiset of objects v, in the next time step. Through evolution rules, object e can be produced, but e should never be in the initial configuration and object e is not allowed to evolve.

(vi) $R'_1, ..., R'_m$ are sets of communication rules associated with each membrane in μ; A communication rule can either be a symport or an antiport rule:

- A symport rule can be of the form (ae^i, in), (ae^i, out), where $a \in O, i \geq 1$. By using this rule, i copy of e objects are consumed to transport object a inside (denoted by in) or outside (denoted by out) the membrane where the rule is defined. To consume copies of object e means that upon the completion of the transportation of object involved in the rule, the occurrences of e are lost, they do not pass from a region to another one. We say that i is the *energy* of this rule.
- An antiport rule is of the form $(ae^i, out; be^j, in)$ where $a, b \in O$ and $i, j \geq 1$; By using this rule, we know that there exist an object a in the region immediately outside the membrane where the rule is declared, and an object b inside the region bounded by the membrane. In the application of this rule, we swap object a and object b using i and j copies of object e in the different regions, respectively. We, then, say that the number $i + j$ is the *energy* of this rule.

Note that no communication can be applied without the utilization of object e.

(vii) $i_{out} \in \{0, 1, ..., m\}$ is the output membrane. If $i_{out} = 0$, this means the environment shall be the placeholder of the output.

A computation is successful when there exists a set of valid transitions from the initial configuration leading to a halting state; this occurs when the system reaches a configuration wherein none of the rules can be applied. If there is no halting configuration—that is, if the system does not halt—computation fails, because the system did not produce any output. Output can either be in the form of objects sent outside the skin or objects sent into an output membrane.

2.1.1 Dynamical Communication Complexity Measures for ECPe

A computation is a set of transitions, denoted: $\delta : C_0 \rightarrow C_1 \rightarrow ... \rightarrow C_n$ where C_i denotes the configuration at the i^{th} step, i.e. the membrane structure and objects of each membrane in the i^{th} step. The notation $C_i \rightarrow C_{i+1}$ represents a transition where a set of rules is used in order to transform configuration of a P system at step i to another configuration at step $i+1$. Based on [1], the dynamical communication complexity parameters associated with a given computation for ECPe are:

$$ComN(C_i \Longrightarrow C_{i+1}) = \begin{cases} 1 & \text{if a communication rule is used in this transition,} \\ 0 & \text{otherwise} \end{cases}$$

$$ComR(C_i \Longrightarrow C_{i+1}) = \text{the number of communication rules used in this transition,}$$

$$ComW(C_i \Longrightarrow C_{i+1}) = \text{the total energy of the communication rules used in this transition.}$$

These parameters are related in that $ComN \leq ComR \leq ComW$. They can be extended in the natural way to results of computations, systems, and sets of numbers. We let $N(\Pi)$ be the set of numbers computed by the system. For $ComX \in \{ComN, ComR, ComW\}$, the following is defined:

$$\begin{aligned} ComX(\delta) &= \sum_{i=0}^{h-1} ComX(C_i \Longrightarrow C_{i+1}), \\ &\quad \text{for } \delta : C_0 \Longrightarrow C_1 \Longrightarrow \ldots \Longrightarrow C_h \\ &\quad \text{is a halting computation,} \\ ComX(n, \Pi) &= \min\{ComX(\delta) \mid \\ &\quad \delta : C_0 \Longrightarrow C_1 \Longrightarrow \ldots \Longrightarrow C_h \\ &\quad \text{in } \Pi \text{ with the result } n\}, \\ ComX(\Pi) &= \max\{ComX(n, \Pi) \mid n \in N(\Pi)\}, \\ ComX(Q) &= \min\{ComX(\Pi) \mid Q = N(\Pi)\}. \end{aligned}$$

2.1.2 A Solution to the Equality Problem in ECPe

The Equality Problem

The equality problem, denoted Qeq_k, is a decision problem that can be represented by the pair (I_{Qeq_k}, Θ_{Qeq_k}) where $I_{Qeq_k} \subseteq \mathbf{N}^k$ corresponding to an instance of Qeq_k and Θ_{Qeq_k} is a total boolean function over I_{Qeq_k}. Given $I_{Qeq_k} = (n_1, n_2, ..., n_k)$,

$$\Theta_{Qeq_k}(I_{Qeq_k}) = true \text{ iff } n_1 = n_2 = ... = n_k$$

The equality problem is solved in [1] using ECPe with input membrane, denoted $\Pi(k)$. The input membrane refers to the membrane to where the coded input instance is placed. We define first the ECPe that solves the problem. Afterwards, we give the representation (or encoding), denoted *cod*(I_{Qeq_k}), of each instance of the problem to be placed in an input membrane.

$$\Pi(k) = (\{a_1, a_2, ..., a_k, d, \#\}, e, [_1 [_2]_2]_1, d, \lambda, R_1, \emptyset, R_2, R_2')$$

where

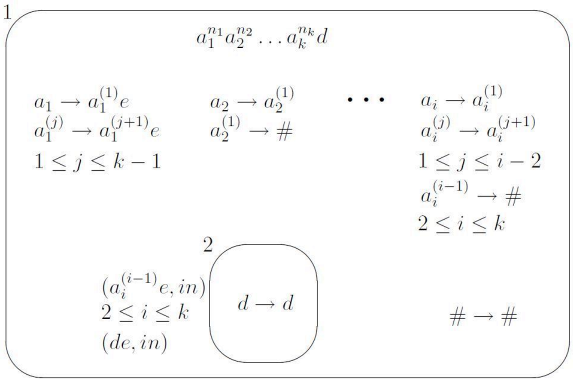

Figure 1: Graphical representation of $\Pi(k)$ solving Qeq_k. Adapted from [1].

$$\begin{aligned} R_1 &= \{a_1 \rightarrow a_1^{(1)} e, a_1^{(j)} \rightarrow a_1^{(j+1)} e\} \text{ for } 1 \leq j \leq k-1 \\ &\bigcup \{a_i \rightarrow a_i^{(1)} e, a_i^{(j)} \rightarrow a_i^{(j+1)}\} \text{ for } 1 \leq j \leq i-2 \\ &\qquad\qquad 2 \leq i \leq k \\ &\bigcup \{a_i^{(i-1)} \rightarrow \#, \# \rightarrow \#\} \text{ for } 2 \leq i \leq k \\ R_2 &= \{d \rightarrow d\} \\ R_2' &= \{(a_i^{(i-1)} e, in), (de, in)\} \text{ for } 2 \leq i \leq k \end{aligned}$$

Given instance I_{Qeq_k}, *cod*(I_{Qeq_k}) $= \{a_1^{n_1} a_2^{n_2} \ldots a_k^{n_k}\}$. Shown in Figure 1 is the graphical illustration for $\Pi(k)$. A successful computation proceeds as follows: In the initial step, all object a_i evolves through rules $a_i \rightarrow a_i^{(1)}$, $(2 \leq i \leq k)$. Simultaneously, object a_1 are evolved through rules $a_1 \rightarrow a_1^{(1)} e$. The objects e produced in the evolution will be used in the next step in order to transport objects $a_2^{(1)}$ in region 2. While copies of object $a_2^{(1)}$ are being transported, objects left in region 1 evolves through rules $a_1^{(j)} \rightarrow a_1^{(j+1)} e$ and $a_i^{(j)} \rightarrow a_i^{(j+1)}$. The new copies of e will be used to transport $a_3^{(2)}$. At the same time, new e objects are produced in region 1. The computation goes on until all $a_i^{(i+1)}$ objects are transported to region 2, $2 \leq i \leq k$. Notice that we are actually comparing n_1 to all other n_is. Thus, in every communication step, all object e must be consumed using rules $(a_i^{i-1} e, in)$. Otherwise, object d will be communicated to region 2 leading the system to a nonhalting computation. Moreover, in case any object $a_i^{(i-1)}$ was not transferred, this object will produce a trap symbol $\#$ through rules $a_i^{(i-1)} \rightarrow \#$ that will lead the system to a nonhalting state.

A computation will only halt if $\Theta_{Qeq_k}(I_{Qeq_k}) = true$. Otherwise, the system computes forever. It can be observed that given k compared values, the system needs $k-1$ communication steps in order to attain a successful computation. Formally, if we denote the family of languages computed using k communication steps as $FComN(k)$, the conjecture states that $Qeq_k \in FComN(k-1)$. Paper [1] conjectures that for ECPe, the minimum number of steps needed to compute for k is $k-1$, i.e. there doesn't exist an

ECPe solution that uses $k-2$ communication steps in solving an instance of Qeq_k.

2.2 Sevilla Carpets

Sevilla Carpets are tables that keep track of information for a halting computation over a P system. Each of its column correspond to a time step, while the rows represent either membrane or rules depending on the type of carpet used. In [2], several types of Sevilla Carpets have been specified:

(i) Tables where cell values indicate, in each time unit and for each membrane, whether at least one rule was used in its region or not.

(ii) Tables where cell values indicate, in each time unit and for each rule, whether it was used or not.

(iii) Tables where cell values indicate the number of applications of each rule in each time unit; this is 0 when the rule is not used and can be arbitrarily large when the rules are dealing with arbitrarily large multisets.

(iv) Tables where cell values are categorized to three cases: a rule cannot be used, a rule can be used but it is not because of the nondeterministic choice and a rule is actually used.

(v) Tables for the case where there is assignment of cost for each and cell values correspond to the number of times a rule is used multiplied to its cost.

As can be observed, the types of Sevilla Carpets are itemized in increasing amount of given information. Below are some helpful information from [2] and [11] that are used to describe the amount of time and space resources for each computation.

- **Weight** - the total number of applications of rules along the computation, represented by the sum of all the elements in the carpet. For Sevilla Carpet (type v), weight measures the total cost of the computation.
- **Surface** - multiplication of the number of steps by the total number of rules used. It can be considered as the *potential size* of the computation.
- **Height** - maximum number of applications of any rule in a step along the computation. Graphically, it represents the highest point reached by the Sevilla Carpet.
- **Average Weight** - value computed by dividing the weight to the surface of the Sevilla Carpet. This concept gives a quantitative value for evaluating how massive parallelism has been exploited on a computation.

Membrane	Rules	Step1	Step2	Step3
1	$a_1 \to a_1'e$	n	-	-
	$a_1' \to a_1''e$	-	n	-
	$a_2 \to a_2'e$	n	-	-
	$a_2' \to a_2''e$	-	0	-
	$a_3 \to a_3'e$	n	-	-
	$a_3' \to a_3''e$	-	n	-
	$a_3'' \to a_3'''e$	-	-	0
2	(de, in)	-	0	0
	(a_2'e, in)	-	n	-
	(a_3''e, in)	-	-	n
	d -> d	-	-	-

Figure 2: Sevilla Carpet type (iv) for a halting computation of $\Pi(3)$

Shown in Figure 2 is the carpet for a halting computation of $\Pi(k)$ when $k = 3$. Note that, rules that lead to a nonhalting configuration will never be applied since Sevilla Carpets are only defined for halting computation. Moreover, carpets only give details about behaviors of computations. It can not be used to compare the efficiency of two P system models in general.

3. USING SEVILLA CARPETS FOR COMMUNICATION COMPLEXITY ANALYSIS

Given the description of Sevilla Carpets, we can construct such carpet (of all enumerated types) for ECPe. We let its corresponding carpet be denoted S. Given an ECPe's definition:

$$\Pi = (O, e, \mu, w_1, \ldots, w_m, R_1, R_1', \ldots, R_m, R_m', j_{out}),$$

the rows of S will be the $\bigcup_{\forall j} R_j \cup R_j'$. Given halting computation: $\delta : C_0 \to C_1 \to \ldots \to C_n$ where C_i denotes the configuration at the i^{th} step, carpet S will have n columns, each representing these steps. The elements of S will be denoted $S[< rule >, < step >]$.

In order to analyze communication efforts in ECPe, we need to define first a mapping in carpet S that corresponds to the dynamical communication complexity parameters given in Section 2.1.1. Let the $r_{jl}' \in R_j'$ be the l^{th} rule ($1 \leq l \leq |R_j'|$) in R_j', we can formally define parameters $ComN$ and $ComR$ as given below:

$$ComN(C_i \to C_{i+1}) = \begin{cases} 1, & \text{if } \exists\, r_{jl}' \text{ such that } S[r_{jl}', i+1] > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$ComR(C_i \to C_{i+1}) = |R'| \text{ where } R' = \{r_{jl}' \mid S[r_{jl}', i+1] > 0\}.$$

For $ComN$, we know that the presence of a communication rule r_{jl}' with $S[r_{jl}', i+1] > 0$ means that rule r_{jl}' is applied to C_i leading to C_{i+1}. Thus, the existence of r_{jl}' assures that at least one communication rule is applied in the transition $C_i \rightarrow C_{i+1}$. For $ComR$, since all rules in R' are applied in C_i leading to C_{i+1}, the cardinality of the set R' represent the number of communication rules used in transition $C_i \rightarrow C_{i+1}$.

If carpet S is of type (v) where the cost pertains to the energy consumed in a communication rule, and the cost of an evolution rule is zero:

$$ComW(C_i \rightarrow C_{i+1}) = \Sigma_{\forall r'_{jl}} S[r'_{jl}, i+1]$$

This is true since only the rules r'_{jl} with $S[r'_{jl}, i+1] > 0$ are applied in transition $C_i \rightarrow C_{i+1}$. Given that $S[r'_{jl}, i+1]$ denotes the total quanta of energy upon the application of rule r'_{jl}, the summation of all $S[r'_{jl}, i+1]$ gives us the total quanta of energy upon the application of all rules.

Inspired by the auxiliary information achieved from carpets to measure descriptive complexity, we can further derive values that may aid in analyzing the quality of communication over ECPe computations. Taking into account the number of communication steps and rule applications, we can also measure

- **Communication Surface** - the total number of communication steps multiplied to the total number of communication rules. This static parameter can be regarded as the *potential size* of communication, where we can think of the total number of communication rules as the size of the communication protocol defined over a specific ECPe.
- **Communication Height** - pertains to the cost of the most expensive communication step.
- **Average Communication Weight** - gives a ratio between the total number of rule application (Communication Weight) and the potential size of the communication (Communication Surface). This value refers to the average cost of a computation taking into account the potential size of the communication.

In the next subsection, we show how dynamical communication parameters and the additional information enumerated above can be derived using Sevilla Carpets. Our objective is to evaluate communication on the ECPe solving the equality problem.

3.1 Sevilla Carpet for the Equality Problem solved in ECPe

Shown in Figure 3 are the Sevilla Carpets type (iv) for computations solving equality problem having 2 and 3 compared values, respectively. The figures highlight the parts considered to measure communication. We assume here that all compared values are equal, so

Membrane	Rules	Step1	Step2
1	$a_1 \text{ -> } a_1'e$	n	-
	$a_2 \text{ -> } a_2'e$	n	-
	$a_2' \text{ -> } a_2''e$	-	0
2	(de, in)	-	n
	($a_2'e$, in)	-	-
	d -> d	-	-

Membrane	Rules	Step1	Step2	Step3
1	$a_1 \text{ -> } a_1'e$	n	-	-
	$a_1' \text{ -> } a_1''e$	-	n	-
	$a_2 \text{ -> } a_2'e$	n	-	-
	$a_2' \text{ -> } a_2''e$	-	0	-
	$a_3 \text{ -> } a_3'e$	n	-	-
	$a_3' \text{ -> } a_3''e$	-	n	-
	$a_3'' \text{ -> } a_3'''e$	-	-	0
2	(de, in)	-	0	0
	($a_2'e$, in)	-	n	-
	($a_3''e$, in)	-	-	n
	d -> d	-	-	-

Figure 3: Sevilla Carpet type (iv) for a halting computation of $\Pi(2)$ (above) and $\Pi(3)$ (below). The highlighted rules and steps are the rules and steps considered to find $ComX$ values and other information

	Π(2)	Π(3)	...	Π(k)
ComN	1	2	...	k-1
ComR	1	2	...	k-1
ComW	n	2n	...	(k-1)n
C. Surface	2	6		(k-1)k
C. Height	n	n	...	n
Ave.ComW	n/2	n/3		n/k

Figure 4: Communication complexity parameters and other communication information for ECPe solving the equality problem.

that the system will always halt, and we let their value be n.

It was mentioned in Section 1 that carpets are also useful tools to achieve a visualization that represents the behavior of a P system computation. Thus, using carpets as in Figure 3 to visualize computation for different input values, we can observe the behavior of communication of the previously defined ECPe for the equality problem. From the tables alone, it can already be shown that apart from the first step, the succeeding steps all involves application of communication rules. Also, at each step, only one communication rule is active.

Presented in Figure 4 are the evaluation parameters mentioned in Section 3 for the ECPe solving the equality problem. From the figure, it can be shown that while $Qeq_k \in FComN(k-1)$, also, $Qeq_k \in FComR(k-1)$ where $FComR$ denotes the family of languages solved in ECPe having $ComR$ equal to $k-1$. This direct proportionality with respect to the number of compared values are also reflected on communication weight and surface, with a decreasing average communication weight as k increases. Thus, the conjecture over the number of communication of steps seems true as well for other computed values. If the conjecture on the number of communication steps is true, then this means the value of $ComR$ given in the table is already optimal since $ComN \leq ComR$.

3.2 Comparing Communication over different ECPe

We present in here two solutions in ECPe solving the problem: given values n and k, $1 \leq n, k$ and $n \geq k$, is n divisible by k? We denote this problem as Q_{div_nk}. For both solutions, we define the coding $cod(I_{Q_{div\_nk}}) = \{a^n, c^k\}$ and we let the input membrane be the skin. Our objective is to use carpets to gain insight on the performance of communication over the two solutions.

The first solution is defined as:

$$\Pi_1(n,k) = (\{a, c, c^{(1)}, ..., c^{(\frac{n}{k})}, \#\}, e, [_1[_2]_2]_1, \lambda, \lambda, R_1, \lambda, R_2, R_2')$$

where

$$R_1 = \{c \rightarrow c^{(1)}e, c^{(i)} \rightarrow c^{(i+1)}e \ 1 < i \leq \tfrac{n}{k}\}$$
$$R_2 = \{c^{(\frac{n}{k})} \rightarrow \#, \# \rightarrow \#\}$$
$$R_2' = \{(ae, in), (c^{(\frac{n}{k})}e, in)\}$$

The first solution solves by repeatedly evolving copies of object c through rules $c \rightarrow ce$ and $c^{(i)} \rightarrow c^{(i+t)}e$ $(1 < i \leq \frac{n}{k})$ and using the object e produced to continuously transfer k copies of a in membrane 2 via rule (ae, in). If n is not divisible by k, the last transfer step will not be able to consume all object e. These object e will be used to communicate some object c to membrane 2. This will eventually lead to a production of the trap symbol $\#$ resulting to a nonhalting state due to the activation of rule $\# \rightarrow \#$.

	Solution1	Solution2
ComN	n/k	1
ComR	n/k	1
ComW	n	n
C.Surface	2n/k	2
C.Height	k	n/k
Ave.ComW	k/2	n/2

Figure 5: Communication Complexity parameters and other communication information for ECPe solving the divisibility problem. Solution 1 and Solution 2 corresponds to $\Pi_1(n,k)$ and $\Pi_2(n,k)$ respectively.

The second solution is defined as:

$$\Pi_2(n,k) = (\{a, c, d, \#\}, e, [_1[_2]_2]_1, \{d\}, \lambda, R_1, \lambda, R_2, R_2')$$

where

$$R_1 = \{a \rightarrow e\}$$
$$R_2 = \{d \rightarrow \#, \# \rightarrow \#\}$$
$$R_2' = \{(c^{(1)}e^{\frac{n}{k}}, in), (de, in)\}$$

The second solution works by producing n copies of object e via rule $a \rightarrow e$. Each object c will then be communicated to membrane 2 using $\frac{n}{k}$ copies of object e. If n is not divisible by k, there will be extra copies of object e which will be used to transfer object d to membrane 2. In the event that object d is communicated and it evolves to the trap symbol $\#$, the rule $\# \rightarrow \#$ will always be applied, thus the system will never halt.

Shown in Figure 5 is the computed communication information derived from the Sevilla Carpets of $\Pi_1(n,k)$ and $\Pi_2(n,k)$. From the table, we can see that basing on the dynamical communication parameters, it seems like Solution 2 is better than the first solution. This is likely the case since it only takes one step and one communication rule for the second solution to solve the problem. On the other hand, the value of $ComN$ and $ComR$ for the first solution are dependent on the input values. Moreoever, if we are going to check the average communication weight of both solutions, the table relays that Solution 2 better capitalizes the parallelism inherent in P systems.

4. CONCLUSION

In this paper, we have shown that Sevilla Carpets may be useful in evaluating the communication effort in EC P systems with Energy. After defining the dynamical complexity parameters in terms of Sevilla Carpets, we have shown a sample evaluation of communication for EC P systems with Energy solving the equality problem. While we have not yet shown that the hierarchy conjectured in [1] can also be true for communication rules, we have shown that Sevilla Carpets give insight on communication which may be of help to accomplishing such work.

The use of Sevilla Carpet for analyzing communication provides a means to represent behavior of communication visually. Moreover, given the dynamical communication parameters and the additional computed information describing the behavior of communication, carpets are promising tools to compare quality of communication of computations over different ECPe solving the same problem.

The use of Sevilla Carpet for communication analysis may also be extended to other types of P systems. We recommend the exploration of such work to employ Sevilla Carpets for comparing communication over different P system models.

5. REFERENCES

[1] H. Adorna, Gh. Păun, M.J. Pérez-Jiménez : On Communication Complexity in Evolution-Communication P systems, *Romanian Journal of Information Science and Technology, 13,2,113-130. (2010)*

[2] G. Ciobanu, Gh. Păun, Gh. Ştefănescu: Sevilla Carpets associated with P systems. In *Proc. of the first Brainstorming Week on Membrane Computing* (M. Cavaliere et al., eds.), Report GRLMC 26/03, Rovira i Virgili University, 135-140. (2003)

[3] Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1, 108–143, (2000), (and Turku Center for Computer Science-TUCS Report 208, November 1998, www.tucs.fi).

[4] Gh. Păun: Further open problems in membrane computing, *Proc. the Second Brainstorming Week on Membrane Computing, Sevilla, February 2004* (Gh. Păun et al., eds.), Technical Report 01/04 of Research Group on Natural Computing, Sevilla University, Spain, 354–365. (2004)

[5] M.J. Pérez-Jiménez: A Computational Complexity Theory In Membrane Computing. In *Proceedings of WMC10, Curtea de Argęs, Romania, August 2009 (Gh. Păun et al., eds.)*, LNCS 5957, Springer, 2010, 125148. (2009)

[6] A.E. Porreca, A. Leporati, G. Mauri, C. Zandron: Introducing a Space Complexity Measure for P systems. *Intern. J. Computers, Communications and Control*, 4, 3, 301-310. (2009)

[7] Gh. Păun: Introduction to Membrane Computing. In: G. Ciobanu, M.J. Pérez-Jiménez and Gh. Păun, eds: Applications of Membrane Computing, Natural Computing Series. Springer, pp.142. (2006)

[8] Gh. Păun: Membrane Computing. An Introduction. *Springer-Verlag, Berlin.* (2002)

[9] M. Cavaliere: Evolution-communication P systems. *Membrane Computing. Proc. WMC 2002, Curtea de Argeş* (Gh. Păun et al., eds.), LNCS 2597, Springer, Berlin, 134–145. (2003)

[10] A. Păun, Gh. Păun: The power of communication: P systems with symport/antiport. *New Generation Computing*, 20, 3 (2002), 295-306.

[11] M.A. Gutiérrez-Naranjo, M.J Pérez-Jiménez, A. Riscos-Nuñez: On Descriptive Complexity of P systems. In *G. Mauri, Gh. Paun, M. Pérez-Jiménez, G. Rozenberg, A. Salomaa, eds., Membrane Computing.* LNCS 3365, Springer, 2005, 320.330.

[12] D. Díaz-Pernil, P. Gallego-Ortiz, M. Gutiérrez-Naranjo, M.J Pérez-Jiménez, A. Riscos-Nuñez: Cell-like versus tissue-like P systems by means of Sevilla Carpets. 7th Brainstorming Week on Membrane Computing, 02/02/2009, Volume I, Sevilla, Spain, p.109-121. (2009)

[13] The P systems Website: www.ppage.psystems.eu.