A membrane computing inspired packing solution and its application to service center workload distribution

Daniel Moldovan, Georgiana Copil, Ioan Salomie, Ionut Anghel, Tudor Cioara
Technical University of Cluj-Napoca, Cluj Napoca,Romania
{daniel.moldovan,georgiana.copil,ioan.salomie,ionut.anghel,tudor.cioara}@cs.utcluj.ro

***Abstract*—This paper presents a bio-inspired system based on membrane computing for solving packing problems in complex dynamic systems where the characteristics of the packed items change continuously. For representing such systems, a bio-inspired model is defined, having as core entities cells and molecules, the packing problem translating into the problem of matching molecules to cells. A symbiotic relationship involving a mutual exchange of chemicals and energy between cells and molecules is defined and used to control the matching process. The system is evaluated in the context of workload distribution in service centers, having as goal the reduction of service center energy consumption by minimizing the number of used servers without affecting the workload resource requirements.**

***Keywords*-bio-inspired, membrane computing, cloud computing, workload distribution, energy efficiency**

I. Introduction

Membrane computing is a branch of bio-inspired computing which extracts computing models from the architecture and the functioning of living cells [1] in order to design so called P-systems. The expressiveness of membrane inspired models has been studied in [2], which shows that even systems which lack features like polarization, label change or division of non-elementary membranes describe universal Turing machines. The computational power of different variants of P-systems has been studied in [3] which targets P-systems with active membranes and two polarizations per membrane and in [4], which focuses on P-systems with mobile membranes, showing that P-systems are computationally efficient and equivalent with Turing machines, being able to solve NP-problems in polynomial time under certain conditions.

Over the last years the energy efficiency management of service centers has emerged as a critical environmental challenge. A U.S. Environmental Protection Agency report to Congress [5] describes an alarming trend in the rise of electricity consumed by service centers and their additional infrastructure. One of the major sources of the energy consumption problem is the inefficient utilization of computing resources. According to [6],in a service center about 30% of servers having an average utilization ratio between 5 and 10 percent. This under-utilization provides a huge opportunity for organizations to reduce the service center energy consumption by employing energy-aware workload distribution techniques to reduce the workload dispersion and turning off unused servers.

In this paper we present a generic membrane-computing inspired computational model which can be applied for solving packing problems. We consider as packing problem any problem in which items need to be grouped with respect to some constraints, from distributing virtual machines in cloud computing infrastructures to packaging different items into boxes for shipment from warehouses to stores. The defined membrane-computing inspired model extracts rules from the biological cell behavior and symbiotic relationships found in nature and applies them in building a rule-based packing solution. For validating the described approach, the presented computational model is applied to a service center workload distribution scenario and evaluated in terms of decision time and solution quality against a best fit first approach.

The rest of the paper is structured as follows: Section II presents the state of the art and real-world applications for membrane-inspired systems, Section III introduces the membrane-inspired context representation model, Section IV describes the symbiotic process between cells and molecules, Section V details the process through which the system evolves, Section VI presents two evaluation scenarios for the membrane-inspired system, and Section VII concludes the paper.

II. Related work

The presented state of the art contains practices and models used in successfully applying bio-inspired concepts from self-regulation biological systems to autonomic computing.

Membrane computing systems have been successfully applied for solving NP problems in various domains. In [7], a P-system is used to implement a depth-first search for finding the solution to the N-Queens problem. The described P-system is shown to solve the N-queens problem for a 20 X 20 board in approximately 15 seconds. Another approach to search problems using P-systems, Reference [8] presents a local search solution using P-systems, successfully applied to the same N-queens problem, showing that while local search algorithms do not guarantee that a solution can be found, such algorithms use less memory and are well suited for problems with large search space. A different search problem solved using P-systems is presented in [9], where the authors construct a system combining membrane computing and ant colony optimization.

978-1-4673-2952-1/12/$31.00 ©2012 IEEE

A real world application of P-systems is presented in [10], the authors applying membrane computing concepts to gasoline blending scheduling. For solving the gasoline blending scheduling problem, the authors define a bio-inspired algorithm which can solve both unconstrained and constrained optimization problems with large number of variables. The presented approach is shown to find optimal or close to optimal solutions in an efficient manner.

A bio-inspired concept which has increasingly found its way in computer science is Symbiosis. As defined in [11], symbiosis is a relationship between dissimilar species and is of three main types: mutualism, commensalism and parasitism. In mutualism all involved parties receive mutual benefits from the symbiosis, in commensalism one party benefits while the other neither loses nor benefits, while in parasitism one party benefits while the other party losses from the symbiotic relationship. Symbiosis is ubiquitous in nature, where it defines associations between different types of entities. Such symbiotic relationships can be mapped to computer science situations under various forms, i.e. a web server associated to a particular server, or a resource supplier associated to a business process.

Symbiosis is used in [11] for enhancing a particle swarm optimization (PSO) algorithm, which, by introducing multiple species cooperation in order to maximize the life of the particles, performs better than the classic PSO in terms of accuracy, robustness and convergence speed. Another symbiotic approach for solving computer science problems is presented in [12], the authors presenting a symbiotic system called SymbioticSphere. The SymbioticSphere defines three types of entities (hosts, platforms and agents) which despite different goals, cooperate by exchanging energy and evolve together to achieve their respective goals. The described symbiotic system possesses a series of characteristics such as decentralization, autonomy and natural selection, characteristics desirable for building self-*(configuring, healing, optimizing and protecting) computer systems.

III. Membrane Inspired Context Model

The founder of membrane computing, G.Paun, defines in [13] three types of P-systems: (i) cell-like P-systems, (ii) tissue-like P-systems, and (iii) neural-like P-systems. Cell-like P-systems consist of a single cell with multiple membranes, tissue-like P-systems consist of a collection of cells, each cell having a single membrane, while neural-like P-systems try to mimic the brain neural structure and are further split into tissue-like and spiking systems. The computational model presented in this work is an enhanced version of tissue-like P-systems, the enhancement over the simple tissue-like system being the addition of one outer membrane and multiple internal membranes to each cell. The concepts defined in the proposed bio-inspired model are described in the rest of this section.

The *tissue* is the central entity in the system, as it hosts the molecules, the cells, and controls the cell population. Two approaches to cell population control are defined which use:

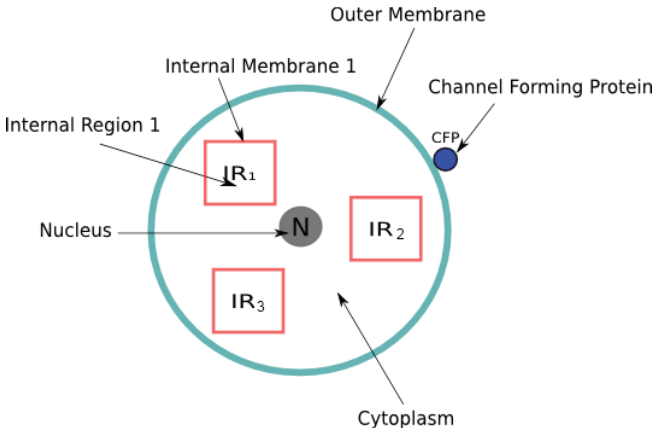


Fig. 1. Cell structure

(i) division and dissolution for increasing/decreasing the cells number and (ii) a fixed immutable pool of cells.

The *cell* is the second most important entity consisting of: (i) a nucleus, (ii) cytoplasm, (iii) an outer membrane and (iv) several internal membranes (Fig. 1). The *nucleus* stores the cell energy and controls the cell energy usage. The *cytoplasm* is the interior region of the cell, separated from the surrounding tissue by the *outer membrane*. The *cytoplasm* holds charged ions used by the cell to produce energy. The *outer membrane* controls what molecules can enter or leave the cell, its permeability being controlled by the presence or absence of a *channel forming protein* (CFP), which creates a channel that allows molecules to pass trough the outer membrane. The channel width is determined by the charged ion concentration in the cell cytoplasm, different for each ion type. A molecule cannot enter a cell unless the molecule produces maximum the amount of ions specified by the CFP, in order to avoid over-flooding the cell with ions. Several *internal membranes* separate the area inside the cell cytoplasm, dividing the cell interior as previously seen in Fig. 1, separating the cytoplasm into: cytoplasm free area and cytoplasm areas surrounded by internal membranes. Each internal membrane is selectively permeable to electrically charged ions and can accept a different flow rate for each ion type.

The *molecule* is the only entity which produces electrically charged ions and each molecule can produce different quantities and different types of electrically charged ions. The ion production is used in the symbiotic relationship between a cell and a molecule in which the molecule generates ions and the cell uses those ions to generate energy. Simple molecules can be grouped in complex molecules if more complex entities need to be represented in the system.

IV. Symbiosis Process

Both cells and molecules require energy to function. Molecules can produce charged ions, but not energy. Cells can produce energy using charged ions, but can't produce ions. In order to survive, molecules and cells need to co-

exists, the molecule providing the ions and the cell using them to generate energy which is used in sustaining both the cell and the molecule. This co-existence defines the symbiosis mechanism between the cells and the molecules.

A. Matching molecules to cells

To find a suitable host cell, a host-less molecule inspects all the cells in the tissue for their Channel Forming Protein (CFP) located on the cell outer membrane. If present, the CFP exposes information about the type and maximum amount of ions its cell can accept. For each cell, a *distance* to the molecule is computed as defined in (1). A difference is computed between the ion production value requested by the cell CFP and the ion production value exposed by the molecule for each ion type. The difference are normalized for each cell and an Euclidean distance is applied over the normalized values, resulting in a number indicating the distance of a particular cell to the molecule. The difference normalization is done to ensure an equal contribution to the distance by each ion type. The molecule sorts all cells in the tissue according to the computed distance and tries to enter the cell with the smallest distance. This behavior is aimed at increasing the usage of each cell, each molecule wanting to enter the most used cell it can fit in. Each molecule executes this process independent of other molecules, the whole symbiotic process possessing a great degree of parallelism.

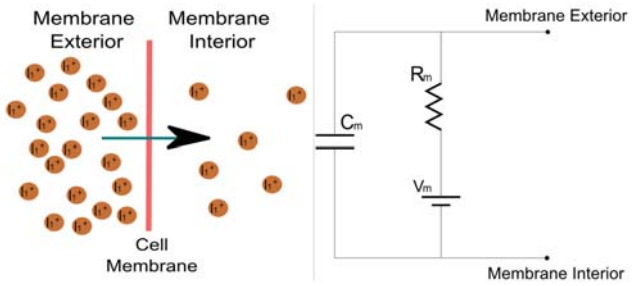
$$d(cell, molecule) = \sqrt{\sum_{CFP\ ions} (accepted_{CFP} - produced_{molecule})^2} \quad (1)$$

After a molecule has found a host cell, both the cell and the molecule work together to produce energy. The energy production and usage mechanisms are explained in the rest of this section.

B. Energy production

The membrane potential, sometimes referred to as membrane voltage, has been studied as early as in the 1950s in works such as [14] or [15]. Membrane electrical potential appears due to a difference in the concentrations of electrically charged ions on the two sides of the cell membrane, as seen in Figure 2a. Due to this phenomenon, the membrane can be represented as an equivalent electrical circuit having a fixed capacitance, a variable resistance and a voltage source (Fig. 2b). The membrane property of electric potential is used for energy production in the symbiotic relationship between the cells and molecules.

The membrane electric potential appears in our system when the concentration of a particular type of electrically charged ion increases in the cytoplasm free area to a level higher than in the areas surrounded by internal membranes. If the internal membranes become permeable and charged ions flow from the cytoplasm free area through the internal membranes, an electric flow is created (Fig. 3). For simplicity, energy (electricity) is considered to be generated during this electric flow.



(a) Ion concentration (b) Equivalent electric circuit

Fig. 2. Membrane potential

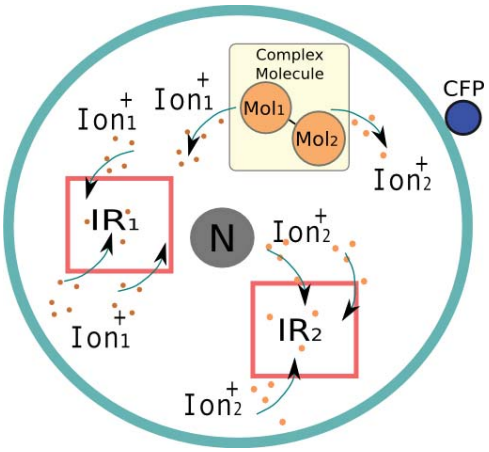


Fig. 3. Charged ions flow

C. Energy consumption

Both cells and molecules need energy to survive. The energy generated from the symbiotic relationship between them is split between the cell and the molecules which reside in its cytoplasm. If a molecule or cell receives less energy than it uses, it will go into an energy starvation state. If a molecule or cell uses less energy than it receives, the surplus energy is stored in the cell nucleus or in the molecule, to be used at a latter time. A limit on the maximum stored energy can be defined, in which case the molecule or the cell enters a state called ”intoxication” whenever the energy limit is exceeded. When a cell remains in an energy starvation state until its energy level drops below 0, the cell will die and expel all the molecules it hosts into the surrounding tissue. When a molecule energy level drops below 0, the molecule will leave its host and migrate into the surrounding tissue.

V. System Evolution

The system evolution is defined as the tissue evolution process and consists of 3 stages: (A) Matching molecules to cells, (B) Symbiosis process execution and (C) System health management. By taking advantage of the independent nature of biological entities, the operations in the mentioned evolution steps are executed in a maximally parallel manner, increasing the response time of the system.

A. Matching molecules to cells

Algorithm 1 Matching molecules to cells

Input: tissue
Output: nothing

```
for all molecule ∈ tissue do
    cells := SortByDistanceToMolecule(tissueCells)
    for all cell ∈ cells do
        cfp := GetChannelFormingProtein(cell)
        if cfp.channelSize ≥ molecule.size then
            AddMoleculeToCell(cell, molecule)
        end if
    end for
end for
```

In this step, each host-less molecule executes the host-search process described in Section IV, with the goal of finding a suitable host (Algorithm 1).

B. Symbiosis process execution

Algorithm 2 Symbiosis process execution

Input: tissue
Output: nothing

```
{ion production phase}
for all cell ∈ tissue do
    for all molecule ∈ GetMolecules(cell) do
        for all ionType ∈ ProductionInfo(molecule) do
            ions := ProduceIons(molecule, ionType)
            AddIonsToCellCytoplasm(cell, ions)
        end for
    end for
    UpdateChanelFormingProtein(cell)
end for

{energy production phase}
for all cell ∈ tissue do
    for all internalMembrane ∈ cell do
        for all acceptedIon ∈ cytoplasmIons do
            energy = Pass(internalMembrane, acceptedIon)
            StoreEnergy(GetCellNucleus(cell), energy)
        end for
    end for
end for

{energy processing phase}
remainingEnergy = UseEnergy(cell, storedEnergy)
for all molecule ∈ cell do
    if remainingEnergy > 0 then
        UseEnergy(molecule, remainingEnergy)
    end if
end for
```

The symbiosis process has three phases (Algorithm 2): ion production, energy production and energy processing. In the first symbiosis process phase, all molecules hosted in cells release charged ions into their host's cytoplasm. Based on the new cytoplasm ions concentrations, the values for the channel forming protein are updated for each cell. After the ion generation is complete, all cell internal membranes become permeable to certain amounts and types of charged ions, allowing ions from the cytoplasm free area to flow through them, generating energy which is stored in the nucleus. In the last phase, each cell, arbitrary or based on certain constraints, divides the energy stored in its nucleus between itself and the molecules it hosts.

C. System health management

Algorithm 3 System health management

Input: tissue
Output: nothing

```
for all cell ∈ tissue do
    if GetEnergy(cell) ≤ 0 then
        for all molecule ∈ cell do
            ExpellFromCell(cell, molecule, tissue)
        end for
    else
        for all molecule ∈ cell do
            if GetEnergy(molecule) ≤ 0 then
                ExpellFromCell(cell, molecule, tissue)
            end if
        end for
    end if
end for
```

This stage handles the situations in which molecules or cells are in energy starvation or intoxication states, controlling the outward migration of molecules from their host cells into the surrounding tissue (Algorithm 3).

The possibility of the molecules to migrate between cells redistributes some of the molecules in the system, avoiding situations in which packing solutions get blocked in local optimum situations, allowing the system to continue optimizing itself, similar in behavior with simulated annealing.

VI. Case Study: Service Center Workload Distribution

The membrane-inspired model presented in Section III is applied to a service center workload distribution scenario and evaluated in the context of two different service center workloads. The properties of the membrane-inspired approach are highlighted and compared to a standard best fit first (BFF) packing algorithm. For mapping the membrane-inspired computational model onto a service center scenario, the target service center is considered to be a platform of homogeneous servers, which run heterogeneous virtual machines, with different computing resource requirements. The detailed mapping of the concepts from the membrane-inspired model to the service

center domain is presented in TABLE I. From here onward we will use the concepts presented in TABLE I interchangeably.

For assessing the membrane-inspired computational model, a workload generator has been built which can generate computing resource requirements for service center virtual machines. The workload generator is configurable and can generate virtual machine requirements both in a random manner or following some particular sequence. Each newly generated virtual machine requirement is entered in the system as a new complex molecule composed of two simple molecules: CPU molecule and Memory molecule, representing the virtual machine's cpu and memory requirements. The generated virtual machine requirements are extracted randomly from a pool of predefined virtual machine configurations, presented in TABLE II. The generated workload is persisted and reused in order to guarantee that the same workload is used in multiple test cases.

Two different service center workloads have been generated with different characteristics in order to properly evaluate the membrane-inspired model from all points of view. The first workload has a monotonic pattern, emphasizing the cell provisioning mechanism, the active cells number following the number of molecules in the system. The second workload is a random workload which focuses on testing the system's response to more real-life conditions in which the workload pattern is not monotonic. To provide a benchmark to compare the membrane-inspired approach with, a best fit first (BFF) packing algorithm was also implemented for solving the same workload distribution problem. In the BFF approach, the cells are sorted in ascending order of their capacity to absorb molecules (free space). For each molecule searching for a host, the sorted list is iterated in increasing order and the first cell which can accommodate the molecule is chosen as host. For both membrane-inspired and BFF approaches, the system was tested using two cell population management mechanisms: (i) a mechanism in which a fixed immutable pool of cells is present in the tissue from the system initialization and (ii) another mechanism in which cells are divided/dissolved depending on the current workload.

The membrane-inspired model and the BFF approaches where evaluated by analyzing their solution quality (number of active cells needed for a particular molecular distribution) and decision time (the time in which all molecules in the tissue find a suitable host cell).

A. Monotonic workload evaluation scenario

The first evaluation scenario consists of 10.000 simulation steps: in the first 5000 steps the virtual machines count (complex molecules no) increases monotonically by 1 at each simulation step, while for the last 5000 steps the virtual machines number decreases monotonically by 1 at each simulation step. The number of molecules in the system at each step in the evaluation scenario simulation is depicted in Fig. 4. In order to test both cell population management approaches, in the tests without cell division, the cell population was set to 1500 cells, number determined from the results of the tests using cell division.

The distribution of active cells trough the evaluation scenario is presented in Fig. 5 and mimics the molecules distribution pattern (Fig. 4). It can be seen that over the entire evaluation, both BFF approaches, with and without cell division, behaved similar in terms of number of cells used. In the first 5000 steps, in which the molecule count continued to increase, both membrane-inspired versions used similar numbers of active cells as the BFF approaches, but on the last 5000 steps, the membrane-inspired approach generated solutions which used less active cells by redistributing the molecules

TABLE I
CONCEPTS MAPPING

Bio-Inspired Concept	Service Center Concept
Tissue	Service center
Cell	Service center server
Cell internal membrane	Server resource (MEM, CPU)
Channel forming protein values	Server optimum workload indicators
Molecule	Virtual machine resource (MEM,CPU)
Complex molecule	Virtual machine
Charged ion	Server computing resource usage unit
Molecule ion production	VM computing resource usage

TABLE II
VIRTUAL MACHINE REQUIREMENTS POOL

Configuration type	CPU cores X [min, max] MHz	Memory [min, max] MB
Type 1	8 X [1500, 2000]	[100, 600]
Type 2	5 X [100, 700]	[100, 600]
Type 3	6 X [100, 700]	[500, 1000]
Type 4	4 X [700, 1500]	[200, 500]
Type 5	4 X [1000, 1700]	[400, 800]

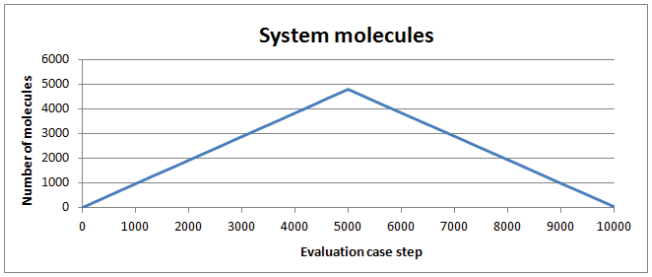


Fig. 4. Molecules distribution

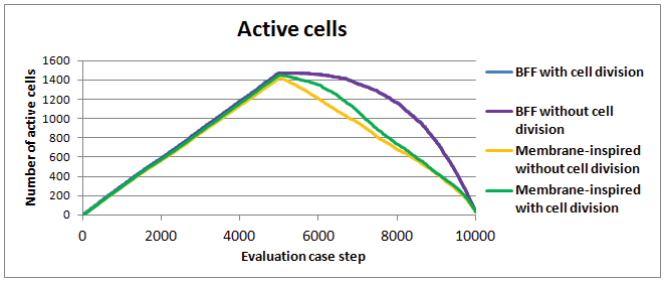


Fig. 5. Active cells distribution

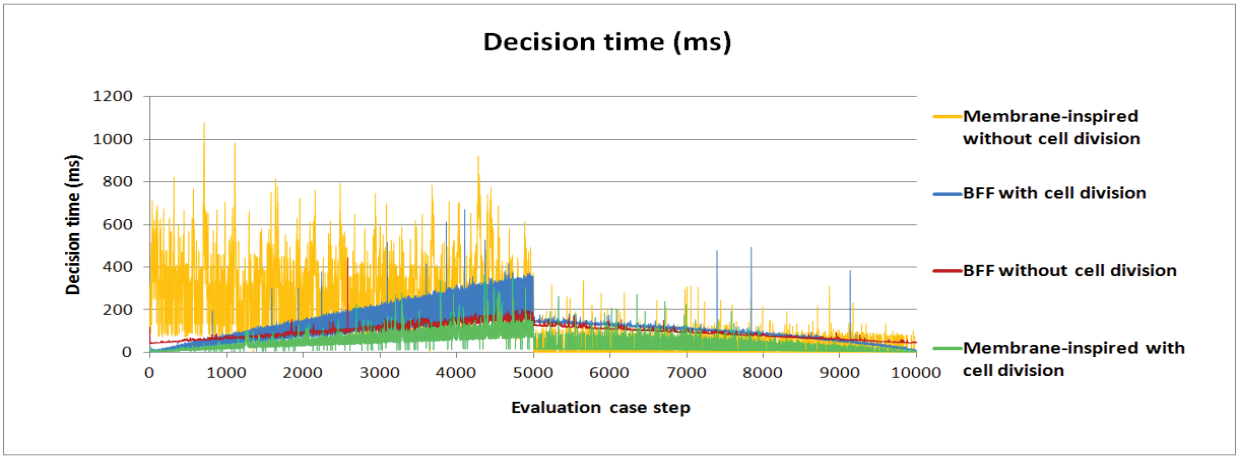


Fig. 6. Decision time distribution

TABLE III
FIRST EVALUATION SCENARIO RESULTS

Approach	Average active cells count			Total decision time (s)			Max decision time (ms)
	Steps (1-5000)	Steps (5001-10000)	Complete test (1-10000)	Steps (1-5000)	Steps (5001-10000)	Complete test (1-10000)	
BFF with cell division	738.18	1115.17	926.68	609.53	458.24	1067.42	671
BFF without cell division	738.19	1116.2	927.56	548.96	440.67	989.47	446
Membrane with cell division	722.85	875.68	799.26	275.87	91.40	367.16	364
Membrane without cell division	708.55	805.75	757.15	1435.10	89.60	1524.33	1081

between cells and reducing the cells free-space fragmentation. As a result, in the first half of the test, the division-enabled membrane-inspired approach used 2% less cells than the BSF with division approach, while the simple membrane-inspired approach used 5% less cells. In the last half of the test, the division-enabled membrane-inspired approach used with 22% less cells, while the simple membrane-inspired approach used 28% less cells compared to the BSF with division approach. Over the entire evaluation scenario, the division-enabled membrane-inspired approach used with 14% less cells, and the simple membrane-inspired approach used with 19% less cells than the BSF with division approach.

The decision time chart (Fig. 6) shows the decision time needed to match all free molecules in the tissue. From the point of view of decision time, compared to the division-enabled BFF, the division-enabled membrane-inspired model obtained a 65% improvement in decision time, while the simple membrane-inspired version introduced a performance penalty of 50%. The performance degradation encountered in the membrane-inspired version without cell division is due to the fact that each molecule sorts the entire cell population each time it searches for a host, the division-less approach having no means of reducing the cell population, the search space for each molecule remains high.

The results of the first evaluation case are summarized in TABLE III. These results highlight the self-optimizing nature of the membrane-inspired model, which redistributes the molecules between cells, decreasing the cells free-space fragmentation and thus reducing the total number of used cells.

B. Random workload evaluation scenario

In this second evaluation scenario, 1700 complex molecules (virtual machines) are randomly generated in the first step of the evaluation and matched to cells (service center servers). This first step is aimed at constructing a real-life scenario in which there already are a number of active cells and molecules in the system. After all 1700 molecules have been successfully matched, the workload generator randomly creates/destroys random sets of molecules for the next 500 steps of the evaluation scenario.

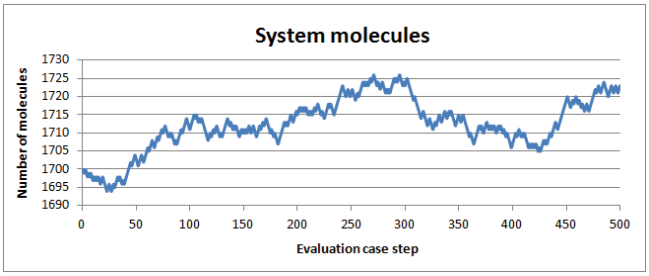


Fig. 7. Molecule distribution

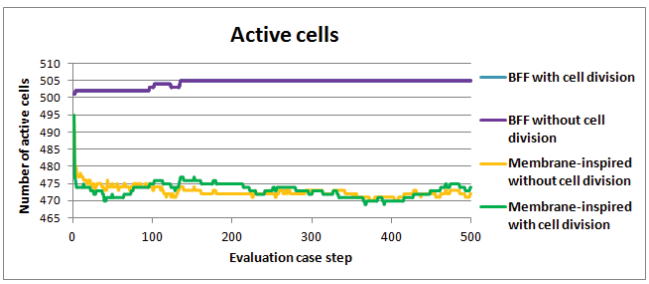


Fig. 8. Active cells distribution

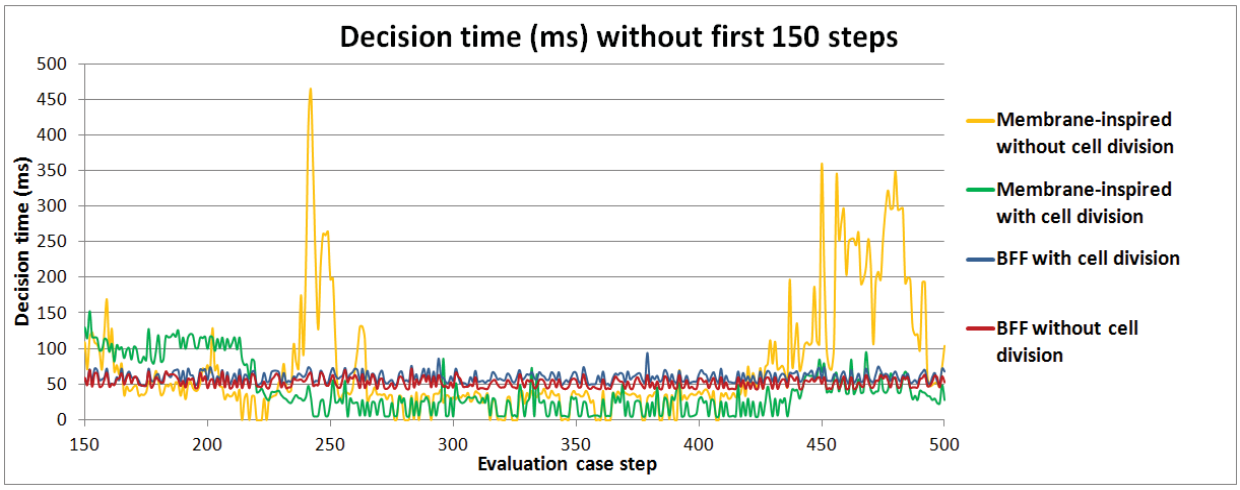


Fig. 9. Decision time distribution without initial 150 steps

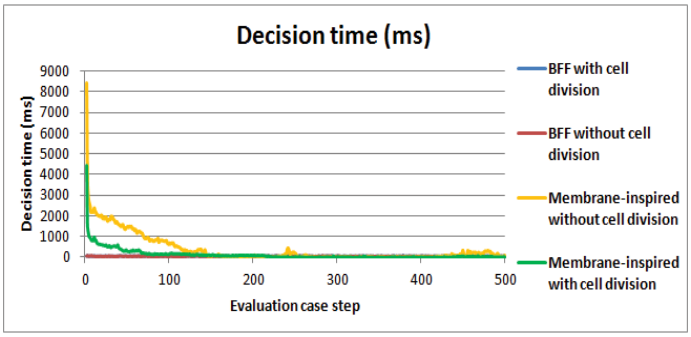


Fig. 10. Decision time distribution

TABLE IV
SECOND EVALUATION SCENARIO RESULTS

Measurement	BFF with cell division	BFF without cell division	Membrane-inspired with cell division	Membrane-inspired without cell division
Matching time for 1700 molecules(s)	15.01	15.13	2495.30	104.48
Total decision time except initial matching(s)	30.38	26.42	68.04	197.23
Decision time without first 150 steps (s)	21.17	18.52	14.68	26.13
Max decision time (s)	0.14	0.76	4.42	8.42
Average active cells No.	500	500	469	468

The molecule distribution through the evaluation scenario is depicted in Fig. 7 and the distribution of active cells through the second evaluation scenario is illustrated in Fig. 8. The decision time without considering the first 150 steps in which the system continued to stabilize is shown in Fig. 9 and the decision time over the entire evaluation scenario without considering the initial step of matching 1700 molecules is illustrated in Fig. 10.

The active cells and decision time charts illustrate that during approximately the first 100 to 150 steps of the evaluation scenario, the membrane-inspired approaches continued to optimize the molecule distribution in cells, both reducing the number of active cells and introducing decision time overhead. From Fig. 10 it can be observed that after the system stabilizes, the decision time for both membrane inspired approaches falls within the bounds of the decision time for the BFF approaches.

The evaluation case results are summarized in TABLE IV. Without considering the first step, over the rest of the evaluation, compared with the simple BFF approach, the division-enabled membrane-inspired approach introduced a 157% overhead in decision time, while the division-disabled one introduced a 189% overhead. Inspecting in detail the cause of this performance degradation, it is visible that the overall decision time was strongly affected by the first 150 steps in which the membrane-inspired systems continues to optimize the molecules distribution. If the decision time is analyzed without considering the first 150 steps (Fig. 9), a different result is obtained, in which the division-enabled membrane-inspired approach achieved a 21% improvement in decision time compared to the basic BFF, while the division-less version still had a 78% degradation in decision time.

Ignoring the first matching step, the maximum decision time was of just 4.42 seconds for the division-enabled membrane-inspired system and of 8.42 seconds for the simple division-less version, results which are acceptable, considering that, at its workload peak, the system handled around 1725 complex molecules (virtual machines) and 475 cells (service center servers).

Regarding the number of active cells, both membrane-inspired approaches brought an overall reduction in the number of active cells of around 7% compared to the simple BFF approach (without cell division). Given that the average number of active cells used by the BFF approach was 501, a 7% reduction translates into turning off 35 cells, number which in a service center can be of great importance from several points of view, from the impact on the cooling infrastructure to the service center energy usage.

VII. Conclusion

This paper presents a bio-inspired membrane-based model for solving packing problems. The model extracts rules from the biological cell behavior and applies them in building a rule-based system. The presented system does not use rule probabilities, as seen in genetic-inspired systems, resulting in a system which is easy to monitor and predict. The predictability of the system is an advantage if it is to be applied in a real-case scenario, where repeatable simulations need to be performed for a different range of reasons, from the molecule distribution estimation in tracking inefficient solutions, to predicting the number of cells needed for a particular molecule set.

The presented bio-inspired system was evaluated in the context of a service center workload distribution scenario, highlighting the possibility of applying it to real-world situations. Two evaluation cases where presented, comparing the membrane-inspired system to a best fit first packing approach. Each evaluation scenario illustrated the behavior of our system under different workload conditions, highlighting the strengths and weaknesses of the proposed approach. From the evaluations results, it can be seen that the membrane-inspired approach can bring a reduction of 7% to 20% in the number of servers used in a service center, depending on the servers and service center workload characteristics. The unused servers can be turned off to reduce the energy consumption of the service center or maintain certain temperature levels.

Another goal of the evaluation was to compare two methods of managing the system's cell population: (i) using division and dissolution to keep the cell population as large as needed and (ii) a method which keeps a fixed-sized pool of cells in the system. Both cell population management mechanism have advantages and disadvantages. Using cell division and dissolution, the cell population is kept as small as possible, thus reducing the search space and decision time overhead. Maintaining a fixed pool of cells introduces performance penalty, but it allows the system to better optimize itself, producing packing strategies which use less cells than the version using division and dissolution.

The presented system was shown to exhibit autonomy, self-optimization, and a large degree of parallelism, making it suitable for implementing efficient autonomous systems.

Acknowledgment

This work has been supported by the European Commission within the GAMES project [16] funded under EU FP7

References

[1] G. Paun, "Computing with membranes," *J. Comput. Syst. Sci.*, vol. 61, no. 1, pp. 108–143, 2000.

[2] S. N. Krishna, "The power of mobility: Four membranes suffice," in *CiE*, ser. Lecture Notes in Computer Science, S. B. Cooper, B. Löwe, and L. Torenvliet, Eds., vol. 3526. Springer, 2005, pp. 242–251.

[3] A. Alhazov, R. Freund, and G. Paun, "Computational completeness of p systems with active membranes and two polarizations," in *MCU*, ser. Lecture Notes in Computer Science, M. Margenstern, Ed., vol. 3354. Springer, 2004, pp. 82–92.

[4] G. Ciobanu and S. N. Krishna, "Enhanced mobile membranes: Computability results," *Theory Comput. Syst.*, vol. 48, no. 3, pp. 715–729, 2011.

[5] "Energy star program, report to congress on server and data center energy efficiency," Public Law 109-431, U.S. Environmental Protection Agency, 2007.

[6] M. Uddin and A. A. Rahman, "Server consolidation: An approach to make data centers energy efficient and green," *CoRR*, vol. abs/1010.5037, 2010.

[7] M. A. Gutiérrez-Naranjo and M. J. Pérez-Jiménez, "Depth-first search with p systems," in *Proceedings of the 11th international conference on Membrane computing*, ser. CMC'10. Springer-Verlag, 2010, pp. 257–264.

[8] M. A. Gutiérrez-Naranjo and M. J. Pérez-Jiménez, "Local search with p systems: A case study," *IJNCR*, vol. 2, no. 2, pp. 47–55, 2011.

[9] M. G. Ge-xiang Zhang, Ji-xiang Cheng, "An approximate algorithm combining p systems and ant colony optimization for traveling salesman problems," 2010.

[10] J. Zhao and N. Wang, "A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling," *Computers & Chemical Engineering*, vol. 35, no. 2, pp. 272–283, 2011.

[11] H. Chen and Y. Zhu, "Optimization based on symbiotic multi-species coevolution," *Applied Mathematics and Computation*, vol. 205, no. 1, pp. 47–60, 2008.

[12] C. Paskorn and S. Junichi, "Exploring adaptive data centers through cooperative symbiotic networking," in *Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems*, ser. FTDCS '07, 2007, pp. 107–118.

[13] G. Paun, "A quick introduction to membrane computing," *J. Log. Algebr. Program.*, vol. 79, no. 6, pp. 291–294, 2010.

[14] B. A. Mobley and E. Page, "The influence of potassium and chloride ions on the membrane potential of single muscle fibres," *The Journal of physiology*, pp. 49–70, 1959.

[15] A. Huxley and A. L. Hodgkin, "The dual effect of membrane potential on sodium conductance in the giant axon of loligo," *The Journal of physiology*, pp. 497–506, 1952.

[16] Games FP7 research project. http://www.green-datacenters.eu/.