Towards a General Framework for Membrane Algorithms

Jos Antonio Andreu-Guzmn[1] and Luis Valencia-Cabrera[1]

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, Spain
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
josandguz@gmail.com,lvalencia@us.es

Summary. Optimization problems have always been, and nowadays even more than ever, a subject of major concern, given their influence in the solution of very relevant daily matters (*i.e.*, worldwide packaging and transport, involving problems like `KNAPSACK` or `TSP`). Many classical meta-heuristics have been proposed to solve these kinds of problems, from simulated annealing to evolutionary algorithms or collective intelligence approaches like ant or bee colonies. Membrane computing brought an alternative set of hybrid algorithms taking the best of P systems and *classical* meta-heuristics in an attempt to improve the previous approaches. However, the heterogeneity of those approaches also brought some mess to the discipline, and a thesis project was proposed to try to formalise a general conceptual framework for membrane algorithms, and also provide a first set of tools inside P-Lingua for the simulation of those approaches formalised within the framework.

1 Introduction

Humanity has always faced challenges throughout its history. Along with the need of solving whatever problems emerging during the evolution, comes an inherent desire of moving forward and set major achievements as individuals and as a society. In this sense, the development of science and technology over the last centuries has stimulated the improvements in all areas of life. Thus, theoretical findings enable qualitative jumps in the resolution of daily-life problems faced by individuals, companies, institutions and countries. For instance, if we pay attention to the extent the packages transport and delivery is increasing worldwide, we will notice how important optimizing these processes is becoming, from the point of view of space utilization, or time/fuel consumption, among others. As already mentioned, in the essence of these real problems we can find very well known optimization problems as `KNAPSACK` [?] or `TSP` [?].

There exist many possible algorithms dealing with this kind of problems, ranging from brute-force algorithms to artificial intelligence techniques based on classical search variants, meta-heuristics, collective intelligence approaches or deep learning methods. No solution is perfect in any situation, so we must take into account the advantages and drawbacks involved. Thus, while the first approaches based on brute-force algorithms generally offered exact solutions whenever it was possible, there are many scenarios where those solutions would be impractical,non-tractable from a computational point of view, so that big instances relevant for real applications could not be processed in the reasonable time required. On the other hand, more sophisticated techniques can provide solutions being *good enough* depending on the purposes, not guaranteeing finding the perfect solution but getting as close as possible given a set of computational restrictions in terms of space and time.

Within this context, probably the most widely used approaches to address these optimization problems during the last few decades have been based on classical meta-heuristics iteratively enhancing a solution or set of solutions until reaching the halting condition set. A bunch of these meta-heuristics have been proposed along the years, from simulated annealing [**?**] to many variants of evolutionary algorithms [**?**]. Additionally, different forms of collective intelligence approaches have also appeared, from particle swarm optimization to ant and bee colonies, generally inspired by nature, and in other cases by physics (see quantum-inspired computing).

Since the proposal by Nishida in [**?**], the previous efforts were combined with the new ideas brought by Membrane Computing [**?**], showing relevant contributions in both areas. As we will detail in the next sections, this thesis project aims to progress in this direction, from both a theoretical and practical point of view, from the formalization and organized classification to the development of sotware tools for their practical application.

2 General idea

Taking into account the context outlined in the previous section, we thought it would be interesting to organize the main approaches detected in membrane algorithms and provide a general formalization, a global conceptual framework and a first set of general-purpose software tools to handle these systems through the infrastructure provided by `P-Lingua` [**?**] and `MeCoSim` [**?**]. These are the main goals of the thesis project where its author, Jos Antonio Andreu Guzmn, is working. More specifically, the focus of the thesis is to provide a review of all contributions to membrane algorithms, classify and extract conclusions about the usefulness, the opportunities emerged from the synergy between Membrane Computing [**?**] and approximate algorithms to solve **NP**-complete problems (these including *classical* meta-heuristics, different branches in Evolutionary Computing[**?**], forms of collective intelligence and so on).

The aim of this previous review and organized classification is to set a first step towards a conceptual framework unifying the plethora of techniques based on the refinement of a set of tentative approximate solutions to a problem through a collective effort of different *individuals* cooperating. The idea would be producing a global formal definition encompassing the different membrane algorithms (MA)[**?**, **?**]. This could provide something similar to the general formalized model existing for the multienvironment P systems, as shown in **??**.

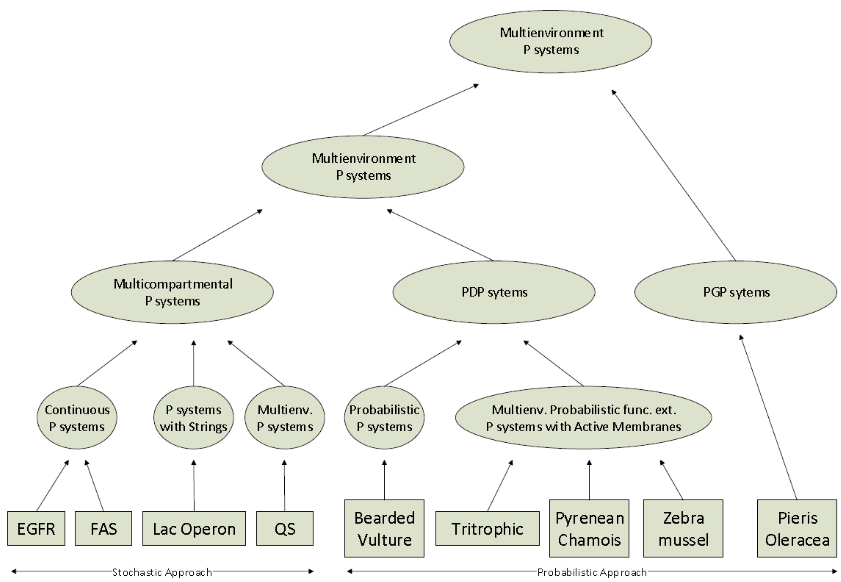

Fig. 1. A general framework for multi-environment P systems

As we can observe and recall, that view collected within the same definition different types of membrane devices characterized by the presence of many P systems distributed among a set of environment or compartments. Initially, each variant (multicompartmental, PDP, PGP systems and so on) had been formalized in a different way, with its own ingredients making them potentially useful for different scenarios; however, a general definition was provided as a unifying framework to comprehend them all and have a clearer overview. Thus, creating this general structure for membrane algorithms would be advisable for the development and visibility of this set of algorithms. This would make it easier for new collaborators or students to understand the main principles, and to decide which MA is better for each situation. Additionally, it could be useful as a helper tool depicting the global scene making easier to decide at first sight which variants or branches might be potentially worth being further explored, as it is another goal of our work: designing alternative types and variants of membrane algorithms, analysing the benefits and drawbacks of considering certain structural and/or functional ingredients.

Besides the inherent advantage of that clarifying view just mentioned, the general formal definition would ease the extension of P-Lingua to incorporate membrane algorithms, initially including some of these hybrid methods with elements from Ant colony Optimization (ACO) [?], Quantum-Inspired Evolutionary Algorithm (QIEA) [?], Genetic Algorithm (GA)[?, ?], Differential Evolution (DE) [?] or Particle Swarm Optimization [?] (PSO), among others. Therefore, another major goal of this thesis project is to provide new extensions in the language, parsers, simulators and accompanying tools within P-Lingua framework for some variants of membrane algorithms, along with possible improvements in MeCoSim to facilitate the work with MA, thus triggering the evolution of this kind of algorithms and the development of the discipline. The provision of visualization tools to follow the simulations of the new variants of MA proposed within the project would aid in the design of solutions and the analysis of the results obtained when testing with certain benchmark problems.

To sum up, amongst the goals of the thesis we intend to offer a perspective of the advantages and limitations of the different types of MA in opposition to classic meta-heuristics, providing a vision of the state-of-the-art of the discipline. Additionally, we will propose variants to improve the performance of certain types of MA, presenting the profit in execute time of the application of MA in parallel computers provided by this improved solutions.

3 Current status

From the beginning of our work we intended not to have a prefixed idea regarding the general formal model to define nor the structure of our classification. Bearing in mind that focus, we started with a first review of the bibliography related with membrane algorithms and their corresponding *classical* predecessors. Thus, we considered the valuable collections provided in [?] and started analysing the different types of algorithms detailed there.

Thus, while deepening into the analysis of the different techniques and progressively building the general formal model, we decided to start paying attention to those types of MA where the density of previous papers was lower, hence offering a bigger space for further studies and potential improvements. Thus, it is possible to divide or classify the contributions according to the type of meta-heuristic used, or depending on the type of membrane structure used. We have started with Ant Colony Optimization P Systems (ACOPS), as explained above, given the fact that, as far as we are concerned, few works addressed these algorithms, and so far only one type of membrane structure (in this case OLMS) was proposed, so we can explore several new variants to overcome some possible limitations and compare their performance. This decision was based on the low amount of papers in [?] studying ACOPS, in comparison with other MAs like GA or QIEA.

According to the promising works we can find in the state of art of ACOPS, previous to our contributions, it was proved it is efficient the use of a membrane structure, taking advantage of the inherent parallelism and the movements of the different ants crossing the membranes to combine good results along the different iterations. This implied an improvement with respect to the classic ACO algorithm, and we considered it would be worth exploring them more in depth. Consequently, we are studying different variants of ACOPS [**?**], looking for desirable properties with respect to time or space complexity, along with other features related with the ease of use and design. Thus, on the one hand, we made changes in the form that ants communicate across the membranes, modifying the kind of colonies and kind of ants used. On the other hand, we started exploring changes in the hierarchical membrane structure used. To check the efficiency of this changes our processes generally are NP-complete problems like Travelling Salesman Problem (TSP) or Knapsack.

In summary, we chose a classic meta-heuristic with a clear intention on going from particular to general. We started with ACO, but not forgetting to gather, during this specific study, the common elements we might extract from this and other meta-heuristics and their hybrid variants in membrane computing. With a deeper exploration, we will be able to include new branches of Membrane Algorithms and progress with the formal model generalizing the different types of MA. At the same time, we have included ACOPS in P-Lingua framework, and prepared P-Lingua and MeCoSim to check the new variants developed against a solution of TSP. Once we have recently finished with the first simulator for ACOPS within P-Lingua, we have started conducting new studies about the influence of the different parameters in the results obtained with ACOPS, and then moving some steps forward with new structural and functional ingredients to improve the results currently published.

References

1. Eiben, A.E., Smith, J. *Introduction to Evolutionary Computing*, Springer-Verlag, 2003.
2. Dorigo, M., Sttzle, T. *Ant Colony Optimization*, Bradford Company, Scituate, MA, USA, 2004.
3. Garca-Quismondo, M., Gutierrez-Escudero, R., Prez-Hurtado, I., Prez-Jimnez, M.J., Riscos-Nez, A. An overview of P-Lingua 2.0. *Lecture Notes in Computer Science*, 5957 (2010), 264-288.
4. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
5. Holland, J.H. Adaptation in Natural and Artificial Systems, *University of Michigan Press*, Ann Arbor, (1975).
6. Kennedy, J., Eberhart, R.C. *Swarm Intelligence*. Morgan Kaufman Publishers Inc., San Francisco, CA, USA, 2001.
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. Optimization by simulated annealing.*Science*, **220** (1983), 671680.

8. Mathews, G.B. On the partition of numbers. *Proceedings of the London Mathematical Society*, **28** (1897), 486490.
9. Menger, K. Das botenproblem. In Menger, K. (Ed.) *Ergebnisse eines Mathematischen Kolloquiums 2*, Teubner, Leipzig, (1932), 11-12.
   Note: Previous works treated this problem since the 1800s (Hamilton, W.R., Kirkman, T.P., among others), previous to its general form by Menger.
10. Nishida, T.Y. Membrane Algorithms: Approximate Algorithms for NP-Complete Optimization Problems. In: Ciobanu G., Păun G., Prez-Jimnez M.J. (eds) *Applications of Membrane Computing.* Natural Computing Series. Springer, Berlin, Heidelberg, 2006, 303-314.
11. Olsson, A. *Particle Swarm Optimization: Theory, Techniques and Applications, Engineering Tools, Techniques and Tables.* Nova Science Publishers, Incorporated, 2011.
12. Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) *The Oxford Handbook of Membrane Computing.* Oxford University Press, New York, 2010.
13. Păun, Gh. Computing with membranes. *Turku Center for Computer Science*, TUCS Technical report 208 (1998), 1–42.
14. I. Prez-Hurtado, L. Valencia-Cabrera, M.J. Prez-Jimnez, M.A. Colomer, A. Riscos-Nez, Mecosim: A general purpose software tool for simulating biological phenomena by means of P systems, in: Proceedings of International Conference on Bio-Inspired Computing: Theories and Applications, 2010, pp. 637643.
15. K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
16. G. Zhang, Quantum-inspired evolutionary algorithms: a survey and empirical study, . *Journal of Heuristics*, **17**, 3 (2011), 303351.
17. Zhang, G., Cheng, J., Gheorghe, M. A membrane-inspired approximate algorithm for traveling salesman problems. *Romanian Journal of Information Science and Technology*, **14** (2011), 319.
18. Zhang, G., Gheorghe, M., Pan L., Prez-Jimnez M.J. Evolutionary membrane computing: A comprehensive survey and new results. *Information Sciences*, **279** (2014), 528-551.
19. Zhang, G., Liu, C.X., Rong, H.N. Analyzing radar emitter signals with membrane algorithms. *Mathematical and Computer Modelling*, **52** (2010), 19972010.