Membrane Computing Applications in Computational Economics

Eduardo Sánchez Karhunen, Luis Valencia-Cabrera

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: esanchek@gmail.com, lvalencia@us.es

Summary. Major efforts have been made along the last decade on the modelling and simulation of phenomena within areas such as Biochemistry, Ecology or Robotics, providing solutions for relevant problems (signalling pathways, population dynamics or logic gene networks, or robot control and planning, among others). However, other areas initially explored have not received the same amount of attention. This is the case of computational economics, where an initial model of the so-called producer-retailer problem was proposed by Gh. and R. Păun making use of membrane computing modelling and simulation tools. In the present paper, we start designing a solution for that problem based on PDP systems, obtaining results comparable with the foundational paper. Then, an enhanced and enriched model is proposed, including several economic issues not considered in the initial model as: depreciation of production capacity, capacity increase decision mechanism, dividends payment and costs associated to production factors. Additionally, both models have been simulated making use of the framework provided by P-Lingua and MeCoSim, and delivering a custom application based on them to reproduce the virtual experiments. Finally, several scenarios have been analysed focusing on different elements included in the model.

Key words: Membrane Computing, Economy, producer-retailer problem, Computational Modelling, PDP Systems

1 Introduction

The main goal of this paper is to extend the success obtained by membrane computing as a modelling tool in different fields to a less explored one, as computational economics. In the context of the so-called producer-retailer problem, multiset rewriting rules for modelling some economic processes were proposed [10], mainly for production of goods from raw material, reception of orders from consumers and purchase transactions. Also, basic numerical evolution of this system was suggested.

The paper mentioned implied a great starting point to show the capabilities of the paradigm in certain fields, but it was not focused on the reproducibility with specific conceptual and software tools. Thus, there were no indications for the reader about the specific framework within membrane computing used to obtain the results presented, neither hints about the membrane structure underlying the system nor the rest of the implementation details.

In order to reinforce the interest in Computational Economics as a promising research path within the applications of Membrane computing, the present paper details the implementation of a PDP system that replicates the results obtained by Gh. and R. Păun. We call this model "Initial producer-retailer model", explaining in depth its design in Section 3, right after introducing the context of this work in Section 2. Once obtained this first result, we propose an "Enhanced producer-retailer model" in Section 4, including several economic issues not considered in the initial model. In both cases, implementation details are provided, along with the analyses of the results obtained under different scenarios. Finally, we outline the main conclusions of this work in Section 5.

2 Preliminaries

This section starts introducing the topic of computational modelling, discussing some widely spread approaches and the choice made with membrane computing. More specifically, it will present the framework of PDP systems, used to model the economic processes presented at the end of the section.

2.1 Modelling approaches

Traditionally, biological systems have been mainly modelled using ordinary differential equations. This approach has several drawbacks: model complexity usually requires a numerical approach; model extension or improvements requires a reconstruction of the model from scratch and difficulties arise handling cases when objects appear in a reduced number of copies or processes have a strong discrete nature.

On the contrary, membrane computing [9] has many advantages for modelling systems. It presents a high degree of generality as a modelling framework (objects, multisets and evolution rewriting rules can be used to model many different situations). It is easy to add any number of membranes and/or evolution rules without essentially changing the type of P system. This modularity allows to introduce extensions or improvements to the model. Additionally, parallelism is introduced in a natural way in the model, and there are no limits to the number of variables interacting simultaneously.

Due to these previous properties, membrane computing has been applied with great success for modelling biological systems, both at a micro level, for cellular reactions [3], and at macro level, for population dynamics [2]. Although there is a wide variety of ecosystems, they share many basic common features: there are several species interacting and a huge number of members of each one; the cyclic repetition of basic processes as feeding, growing, reproduction and death and environment influences on the system evolution.

The following section will present th framework of PDP systems, used to model the phenomena studied along this work.

2.2 Population Dynamic P systems

PDP Systems (Population Dynamic P system) were developed to consider the computational impact of the previous issues [4]. Formally, a PDP system of degree (q, m) and $T \geq 1$ units of time is a tuple $\Pi = (G, \Gamma, \Sigma, T, \{\Pi_k : 1 \leq k \leq m\}, \{E_j : 1 \leq j \leq m\}, R_E)$, where:

- $G = (V, S)$ is a directed graph with $m \geq 1$. $V = \{e_1, , e_m\}$.
- Γ and Σ are alphabets such that $\Sigma \subsetneq \Gamma$.
- $T \geq 1, n \geq 1$ are natural numbers.
- $\forall k, 1 \leq k \leq m, \Pi_k = (\Gamma, \mu, M_1, \cdots, M_q, \mathcal{R}, i_{in})$, where:
  - μ is a rooted tree with $q \geq 1$ nodes labelled with elements of $\{1, \cdots, q\} \times \{0, +, -\}$.
  - $\forall i, 1 \leq i \leq q, M_i \in M_f(\Gamma)$.
  - $\mathcal{R}$ is a finite set of rules of the type: $u[v]_i^{\alpha} \xrightarrow{p} u'[v']_i^{\alpha'}$, where $u, v, u', v' \in M_f(\Gamma), 1 \leq i \leq q, \alpha, \alpha' \in \{0, +, -\}$, and p is a probability function with domain $\{0, \cdots, T\}$. Also, the sum of probabilities of rules whose left hand side (LHS) is $u[v]_i^{\alpha}$ is 1 at each instant $t (0 \leq t \leq T)$.
  - i_{in} is a node of μ.
- $\forall j, 1 \leq j \leq m, E_j \in M_f(\Sigma)$.
- R_E is a finite set of environment rules of the type: $(x)_{e_j} \xrightarrow{p_1} (y_1)_{e_{j,1}}, \cdots, (y_h)_{e_{j,h}}$, where $x, y_1, \cdots, y_h \in \Sigma, \{(e_j, e_{j,i}) \in S, 1 \leq j \leq m, 1 \leq i \leq h\}$, and p_1 is a probability function with domain $\{0, \cdots, T\}$. Also, at each instant t, with $0 \leq t \leq T$, the sum of all probability function values associated to rules whose LHS is $(x)_{e_j}$ must be 1.
- There are no rules of the type $u[v]_i^{\alpha} \xrightarrow{p} u'[v']_i^{\alpha'}$ in the skin membrane of P Systems and environment rules of the type: $(x)_{e_j} \xrightarrow{p_1} (y_1)_{e_{j,1}}, \cdots, (y_h)_{e_{j,h}}$ such that $x \in u$.
- Each environment e_j contains exactly one system Π_k.

Therefore, A PDP system Π of degree (q, m) presents m environments $e_1, \cdots, e_m$ interconnected by edges of a directed graph G. Each of these environments e_j can only contain symbols of alphabet Σ, and a unique ordinary P System $\Pi_k = (\Gamma, \mu, M_1, \cdots, M_q, \mathcal{R}, i_{in})$ with this same skeleton inside each environment, but such that the initial multisets of Π_k depend on e_j and the probability functions associated with rules of Π_k depend on e_j. Finally, the semantics of PDP systems depends on the algorithm of simulation.

2.3 Economic process modeling

Traditionally, economics processes have been modelled using differential equation systems. Although these modelling techniques are predominant, many efforts have been made to investigate other techniques such as multi-agent techniques [5]. Due to the good performance of PDP systems modeling dynamics of biological systems, many researchers have proposed the idea of using membrane computing in modeling economic processes [7, 8, 1, 10, 11].

A parallelism between biological and economic processes can be identified, as Păun analyses in [10]. Many elements of membrane computing can be interpreted in economic terms. An object can represent any unit of a generic item involved in different economic processes. Also, they can represent elements of diverse nature: material good, monetary units, depreciation representation, authorization for transactions, caps of production, etc. A membrane can be any entity as producer, consumers, markets, whole economy or any other element interacting with another one. As usual in P systems, any membrane has objects associated with it. Objects can have multiplicities greater than one, so a natural way or handling them is to use multisets. Finally, multiset rewriting rules can model different kind of interactions between objects of different or the same membrane and they can represent a huge variety of processes as purchase transactions, production of goods or depreciation phenomena.

3 Initial Retailer Producer model

This section will explore in certain detail some usual economic phenomena, through the reference problem well known as "Retailer-producer" problem. After its general description, the formal model designed is presented, along with the interpretation of the elements included, the parameters involved and the deep analysis of the different modules of rules involved in the evolution of the system. Finally, the simulation results are shown and analysed.

3.1 General description

Informally, the retailer-producer problem can be described as a one good market with several players interacting with each other. A set of producers P_i that transform raw material produced by a generic source S into units of good d and a set of retailers R_j, that receive orders $\bar{d}$ from a generic consumer C. Both try to match units of d with $\bar{d}$ by means of transactions. These players are represented in 1 as circles. Each one is characterized by a parameter: P_i has a production capacity, R_j has a storage capacity, S produces raw material a at a constant rate and C generates a demand $\bar{d}$ at a constant rate. Transactions between players are represented as double arrow lines. Each of these transactions imply the exchange of monetary units characterized by its owner. u_S in possession of S, obtained from

P_i who have paid a price for each unit of a; u_i in possession of P_i, obtained from R_j who have paid a price for each unit of d; v_j in possession of C who have paid a price for each unit of $\bar{d}$ and u_C in possession of C. These are injected externally into the system allowing the consumer to throw orders.

Prices must be added to the different interactions: wholesale distributors price, that is, price at which S sells a unit of a to P_i; the price of a unit of d when sold by R_j to C and the price of a unit of d when sold by P_i to R_j. Simultaneously to the existence of prices, there are budget restrictions associated to each player. No more units of goods can be bought than the equivalent ones to the total number of monetary units owned by each player. The existence of prices enriches the evolution of the system, introducing the possibility of lack of money and making impossible to apply certain rules.

Finally, the system evolves cyclically with five steps: 1) generation of the initial conditions, 2) production of goods and reception of orders for those goods, 3) generation of purchase authorizations, 4) purchase transactions and 5) technical and cleaning rules.

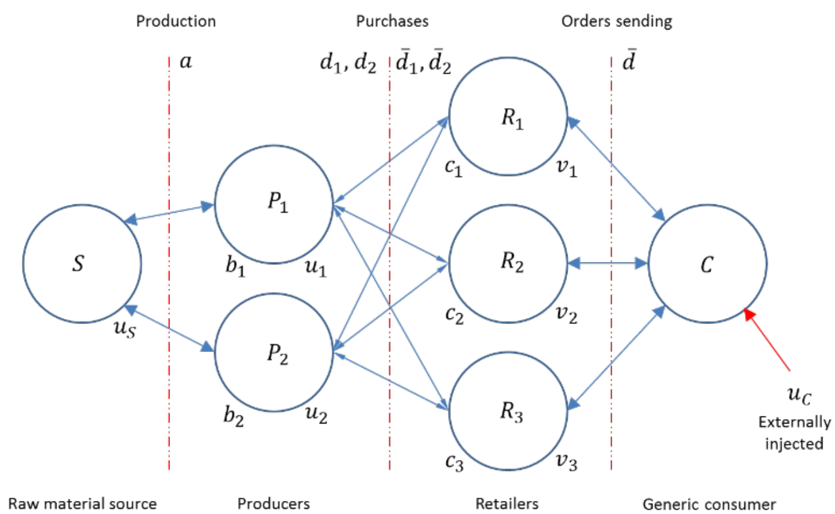


Fig. 1. Schematic representation of retailer producer problem

Production side

In economic theory, P_i has a production function (number of goods or services produced) with the following general form: $Y_i = f_i$ (factors of production). These factors are the different physical inputs used to produce goods. Typically, they are classified into three main categories: raw material, labor of workers and capital stock. While f_i specifies how factors are transformed into goods. For simplicity, we make some assumptions. All P_i have access to the same technology, thus they all have the same production function $\forall i(Y_i = Y)$. Also, each P_i takes as factors: raw material provided by S; production capacity (also known as capital stock) and, for simplicity, labor is not considered. Thus, we can simplify the production function obtaining: $Y_i = Y = f(rawmaterial, capital) = f(a, b_i)$.

The multiplicity of a represents the total amount of raw material available for production and the multiplicity b_i represents the total production capacity of P_i. Additionally, we consider the simplest form for f, where only one unit of a and b_i are consumed to produce one unit of d. This exchange rate can be easily changed to consider more complex situations.

Demand side

In real markets, there is a bunch of individual consumers requiring units of good d. In the context of the so-called, economic rational behavior model, the behavior of each individual consumer is captured by a utility function of the form: $U_i = U_i(consumedinputs) = U_i(consumption, leisure)$. U quantifies in monetary units the happiness of individuals, making explicit their preferences about the simultaneous consumption of multiple disposable goods. Units of d obtained by consumer is called consumption and leisure can be considered as the time not dedicated to work (with a clear cost of opportunity). Classical economical models consider that rational individuals try to maximize his utility function. For simplicity, we make some assumptions. The only factor for utility is the consumption of d and labor (as complementary to leisure) is not considered. All consumers have the same utility function, same preferences and, thus, the same behavior $\forall i(U_i = U)$. This gives rise to the concept of representative consumer (more generally, representative agents). We can consider the sum of the utility functions of the population of consumers, generating a, so called, aggregate demand of d (each unit is denoted by $\bar{d}$). This can be represented as a generic consumer C. Thus, we can simplify the utility function obtaining: $U_i = U = g(\bar{d})$.

3.2 Model formalization

A simple membrane structure for the PDP system is selected with a unique environment containing one P system with two membranes. Membrane 1 is used for the R_j and P_i operations of good and order generation. Membrane 2 is used for performing the purchase transactions. The previous system will be modelled by a PDP system of degree $(2, 1)$ and $T \geq 1$ units of time $\Pi = (G, \Gamma, \Sigma, T, \mathcal{R}_E, \mu, \mathcal{R}_\Pi, \{f_r \in \mathcal{R}_\Pi\}, M_1, M_2)$, where $G = (V, E)$, with $V = \{e_1\}$ and $E = (e_1, e_1)$ and working alphabet: $\Gamma = \{b_i, d_i, u_i, c_j, \bar{d}_j, v_j, \bar{e}_j, f_(i, j) : 1 \leq i \leq k_1, 1 \leq j \leq k_2\} \cup \{R_1, R_2\} \cup \{C, S, \bar{d}, a, u_C, u_S\}$, where:

- C: aggregate generic consumer.
- S: raw material supplier.

- $\bar{d}$: unit of aggregate demand from C.
- a: unit of supplied raw material provided by S.
- u_C: monetary unit owned by C.
- u_S: monetary unit owned by S.
- b_i: unit of production capacity of $P_i, 1 \leq i \leq k_1$.
- d_i: unit of good supplied by $P_i, 1 \leq i \leq k_1$.
- u_i: monetary unit owned by $P_i, 1 \leq i \leq k_1$.
- c_j: unit of capacity of $R_j, 1 \leq j \leq k_2$.
- $\bar{d}_j$: unit of good demanded by $R_j, 1 \leq j \leq k_2$.
- v_j: monetary unit owned by $R_j, 1 \leq j \leq k_2$.
- $\bar{e}_j$: unit of good demanded by R_j and authorized for transaction unit of $\bar{d}_j, 1 \leq j \leq k_2$.
- $f_(i,j)$: authorization for $\bar{d}_j$ to be exchanged with d_i, for $1 \leq i \leq k_1$, $1 \leq j \leq k_2$.
- R_1, R_2: for technical reasons.
- $\Sigma = \emptyset$.
- $R_E = \emptyset$.
- $\Pi = \{\Gamma, \mu, M_1, M_2, \mathcal{R}_\Pi\}$, where $\mu = [[\,]_2]_1$ and $M_1 = \{C, S, R_1, R_2\} \cup \{b_i^{k_{i,1}}, u_i^{k_{i,2}} : 1 \leq i \leq k_1\} \cup \{c_j^{k_{j,3}} : 1 \leq j \leq k_2\}$

Model parameters

- k_1: total number of producers.
- k_2: total number of retailers.
- k_3: units of a inserted into the system by S.
- k_4: allowed deviation from k_3.
- k_5: units of $\bar{d}$ inserted into the system by C.
- k_6: allowed deviation from k_5.
- k_7: price fixed by S for each unit of a.
- k_8: price fixed by C as an estimation of each order of good.
- $k_{i,1}$: initial production capacity of P_i, $1 \leq i \leq k_1$.
- $k_{i,2}$: initial monetary units of P_i, $1 \leq i \leq k_1$.
- $k_{j,3}$: initial capacity of R_j, $1 \leq j \leq k_2$.
- $k_{m,4}$: discrete prob distribution of units of a inserted into the system by S, $1 \leq m \leq 3$.
- $k_{m,5}$: discrete prob distribution of units of $\bar{d}$ inserted into the system by C, $1 \leq m \leq 3$.
- $k_{i,6}$: price fixed by P_i for each unit of d_i, $1 \leq i \leq k_1$.
- $k_{j,7}$: price fixed by R_j for each order of good, $1 \leq j \leq k_2$.

3.3 Modules of rules

Module 1: Initialization

The initial conditions for the cycle are generated, including the disposability of $\bar{d}$ and a. We assume that S can supply a nearly fixed amount of a at the beginning of each cycle. To introduce some variability (not associated to any concrete real economic behavior), we will decompose the basic rule in a bunch of rules differing slightly around k_3 in the number of generated units of a. This variability is controlled by parameter k_4 and the associated probability of each rule $k_{m,4}$.

$$r_1 \equiv R_1\, s[\,]_2 \xrightarrow{p=k_{1,4}} a^{k_3+k_4} s[R_1]_2^+ \quad r_2 \equiv R_1\, s[\,]_2 \xrightarrow{p=k_{2,4}} a^{k_3} s[R_1]_2^+$$
$$r_3 \equiv R_1\, s[\,]_2 \xrightarrow{p=k_{3,4}} a^{k_3-k_4} s[R_1]_2^+ \quad r_4 \equiv R_1\, s[\,]_2 \xrightarrow{p=1-k_{1,4}-k_{2,4}-k_{3,4}} a^{k_3-2*k_4} s[R_1]_2^+$$

We also assume that C generates a nearly fixed amount of $\bar{d}$ at the beginning of each cycle. Again, we decompose the basic rule in a bunch of rules differing slightly around k_5 in the number of generated units of $\bar{d}$. This variability is controlled by parameter k_6 and the associated probability of each rule $k_(m,5)$. C also"generates" the amount of money estimated to throw orders to R_j to be able to satisfy completely the demand $\bar{d}$, controlled by k_8.

$$r_5 \equiv R_2\, c[\,]_2 \xrightarrow{p=k_{1,5}} \bar{d}^{k_5+k_6} u_C^{(k_5+k_6)k_8} c[R_2]_2^+$$
$$r_6 \equiv R_2\, c[\,]_2 \xrightarrow{p=k_{2,5}} \bar{d}^{k_5} u_C^{k_5 k_8} c[R_2]_2^+$$
$$r_7 \equiv R_2\, c[\,]_2 \xrightarrow{p=k_{3,5}} \bar{d}^{k_5-k_6} u_C^{(k_5-k_6)k_8} c[R_2]_2^+$$
$$r_8 \equiv R_2\, c[\,]_2 \xrightarrow{p=(1-k_{1,5}-k_{2,5}-k_{3,5})} \bar{d}^{k_5-2k_6} u_C^{(k_5-2k_6)k_8} c[R_2]_2^+$$

Despite being considered in theoretical models, this idea of generating money from "nothing" at the beginning of each cycle is completely counterintuitive and do not reflects the real behavior of actual systems. This is one of the ideas that leads to an enhancement and reformulation of this initial model in following chapters.

Module 2: Producer & Retailer operation

Objects P_i have at their disposal the amount of a generated in Step 1. They compete to obtain units of a, so that they can generate units of d according to their production function. For each unit of a used by P_i it must pay a price k_7, reducing the number of u_i owned by P_i and increasing the ones u_S owned by S. Finally, each unit of d produced by P_i is denoted by d_i, with $1 \leq i \leq k$.

$$r_9 \equiv ab_i u_i^{k_7}\, c[\,]_2^+ \rightarrow u_S^{k_7} [d_i]_2^0, 1 \leq i \leq k_1$$

R_j must provide service to $\bar{d}$ generated in Step 1. They compete to get units of $\bar{d}$ to serve the demand of C. It may also be interpreted as R_j receives orders from C. For each unit ordered by C to R_j it must pay a price $k_{j,7}$, reducing the number of u_C owned by C and increasing the ones v_j owned by R_j. We will allow different prices for order to each R_j (parameter $k_{j,7}$). Each unit of good necessity $\bar{d}$ served by R_j is denoted by $\bar{d}_j$, with $1 \leq j \leq k_2$.

$$r_{10} \equiv \bar{d} c_j u_C^{k_{j,7}}\, c[\,]_2^+ \rightarrow [\bar{d}_j {v_j}_S^{k_{j,7}}]_2^0, 1 \leq i \leq k_2$$

Module 3: Performing transactions

Once orders $\bar{d}_j$ have been received by R_j and units d_i are generated by P_i, the commercial transactions can take place. One item of d is purchased by R_j from P_i to satisfy the order $\bar{d}$ carried by R_j. For each unit bought by R_j it must pay a price, reducing the number of v_j owned by R_j and increasing the ones u_i owned by P_i. Capacities c_j and b_i consumed in the production of d_i and $\bar{d}_j$ are set free.

$$[d_i \bar{d}_j v_j^{price}]_2 \xrightarrow{probability} [b_i c_j u_i^{price}]_2$$

Additionally, we can associate a probability to each possible transaction comprising many effects: the confidence of R_j on P_i; the price of the product offered by P_i; the willing of R_j to buy a good or the necessity of P_i to sell a good. Depending on the effects considered and their variability during the process, these probabilities must be computed once at the beginning of the process or recalculated after each cycle. An intuitive way of thinking about these probabilities is that the unit probability is distributed among a bunch of rules of this type:

$$\begin{array}{ll} [d_1 \bar{d}_1 v_1^{price}]_2 \xrightarrow{p_{11}} [b_1 c_1 u_1^{price}]_2 & \\ & [d_2 \bar{d}_1 v_1^{price}]_2 \xrightarrow{p_{12}} [b_2 c_1 u_2^{price}]_2 \\ [d_3 \bar{d}_1 v_1^{price}]_2 \xrightarrow{p_{13}} [b_3 c_1 u_3^{price}]_2 & \end{array}$$

so that $p_{11} + p_{12} + p_{13} = 1$. On the other hand, PDP systems do not allow a direct translation of rules of this type because the evolution of any possible LHS is determined by a set of rules summing probability one. This drawback is solved creating for each exchange transaction between P_i and R_j, a symbol $f_{j,i}$ that acts as an authorization card for the transaction. This $f_{j,i}$ follows the originally desired probability distribution and can be used to simulate geographical barriers between players or preference for one of the producers. Now, probabilities associated to rules with the same LHS sum up to one and the purchase transactions can take place but now with probability one.

$$\begin{array}{ll} r_{14} \equiv [\bar{d}_1]_2 \xrightarrow{p=1} [\bar{d}_1 f_{1,1}]_2 & r_{15} \equiv [\bar{d}_1]_2 \xrightarrow{p=0} [\bar{d}_1 f_{1,2}]_2 \\ r_{16} \equiv [\bar{d}_2]_2 \xrightarrow{p=0.5} [\bar{d}_2 f_{2,1}]_2 & r_{17} \equiv [\bar{d}_2]_2 \xrightarrow{p=0.5} [\bar{d}_2 f_{2,2}]_2 \\ r_{18} \equiv [\bar{d}_3]_2 \xrightarrow{p=0.15} [\bar{d}_3 f_{3,1}]_2 & r_{20} \equiv [\bar{d}_3]_2 \xrightarrow{p=0.85} [\bar{d}_3 f_{3,2}]_2 \end{array}$$

$$r_{20} \equiv [d_i \bar{d}_j f_{j,i} v_j^{k_{2,6}}]_2^0 \rightarrow [b_i c_j u_i^{k_{2,6}}]_2^-, 1 \leq i \leq k_1, 1 \leq j \leq k_2$$

Further developments of the model could explore the possibility of considering dynamic probabilities for these transactions.

Technical & cleaning rules

Finally, some rules are necessary for technical reasons. $f_{i,j}$ not exhausted in purchase transactions have no utility and symbols r_1 and r_2 are restored to their original location.

$$r_{26} \equiv [f_{i,j}]_2^- \rightarrow [\,]_2^0, 1 \leq i \leq k_1, 1 \leq j \leq k_2$$
$$r_{30} \equiv [r_1 r_2]_2^- \rightarrow r_1 r_2 [\,]_2^0$$

d_i, $\bar{d}_j$ and v_j not exchanged represent real things and cannot be cleaned.

$$r_{27} \equiv [\bar{d}_j]_2^- \rightarrow \bar{d}_j\,[\,]_2^0, 1 \leq j \leq k_2 \qquad r_{28} \equiv [v_j]_2^- \rightarrow v_j [\,]_2^0, 1 \leq j \leq k_2$$
$$r_{29} \equiv [d_i]_2^- \rightarrow d_i\,[\,]_2^0, 1 \leq i \leq k_1$$

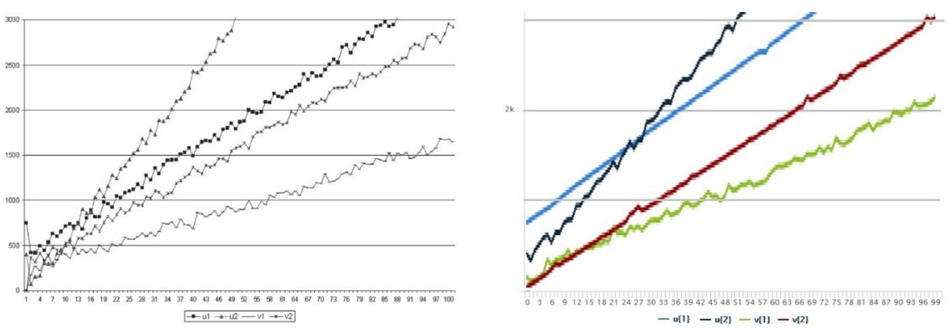
Similarly, we have the symmetric operations at the beginning of the cycle, pushing these elements into the operational membrane 2.

$$r_{12} \equiv [\bar{d}_j]_2^+ \rightarrow \bar{d}_j\,[\,]_2^0, 1 \leq j \leq k_2 \qquad r_{13} \equiv [v_j]_2^+ \rightarrow v_j [\,]_2^0, 1 \leq j \leq k_2$$
$$r_{11} \equiv [d_i]_2^+ \rightarrow d_i\,[\,]_2^0, 1 \leq i \leq k_1$$

3.4 Simulation results

To compare the results of our implementation to the ones described by Paun, we will try to use the same set of values and number of cycles. In our model, 200 cycles with 5 steps in each cycle using DNDP-4 algorithm as inference engine.

For this purpose we consider the chart provided in Pauns article and compare it to the plot of the same output variables in our model (u_1, u_2, v_1, v_2). Both charts are represented with the same axis scale and number of T cycles.



(a) Pauns Model (b) Initial Model

Fig. 2. Evolution of monetary units owned by retailers - Paun Model vs Initial Model

4 Enhanced model

After the initial model presented, it started a process to go deeper into economic phenomena, leading to an enhanced model, enriched with a number of features of

Parameter	Value/s	Description
k_1	2	Total number of producers
k_2	3	Total number of retailers
k_3	60	Units of a inserted into the system by S
k_4	1	Deviation from k_3
k_5	60	Units of $\bar{d}$ inserted into the system by C
k_6	1	Deviation from k_5
k_7	11	Price fixed by S for each unit of a
k_8	14	Price fixed by C as an estimation of each order of good
$k_{i,1}$	$(65, 35)$	Initial production capacity of P_i, $1 \leq i \leq k_1$
$k_{i,2}$	$(750, 400)$	Initial monetary units of P_i, $1 \leq i \leq k_1$
$k_{j,3}$	$(50, 30, 20)$	Initial capacity of R_j, $1 \leq j \leq k_2$
$k_{m,4}$	$(0.01, 0.95, 0.03)$	Prob distrib. of a inserted into the system by S
$k_{m,5}$	$(0.03, 0.90, 0.04)$	Prob distrib. of units of $\bar{d}$ inserted into the system by C
$k_{i,6}$	$(12, 13)$	Price fixed by P_i for each unit of d_i
$k_{j,7}$	$(13, 14, 15)$	Price fixed by R_j for each order of good j, $1 \leq j \leq k_2$

Table 1. MeCoSim simulation parameter values

interest. This section focuses in the description of the new ingredients involved, the new processes studied and the details of the new model designed, following the section a structure similar to the previous one, from the foundations about the processing to the exploration of the model and analysis of the simulations.

4.1 General description

The behavior of the previous initial model can be condensed as follows: a steady increase of monetary units owned by P_i, R_j and C; nearly stable P_is and R_j's capacities and monetary units obtained by S get out of circulation in the system. These facts can be explained by the absence of variations in the rest of parameter of the model. First, prices associated to goods, raw material and aggregate demand are initially settled and remain unchanged during the system evolution (absence of a natural process trying to find a price of equilibrium). Secondly, P_i's and R_j's capacities are fixed and no changes are allowed (lack of a capital market). Additionally, there is an artificial exogenous injection of monetary units at the beginning of each cycle. To get our initial model nearer to real situations, new issues must be modelled. In the next chapters, some of the latter restrictions will be relaxed or modified. First, variations of P_i's and R_j's capacities will be allowed (associated to capital stock depreciation and investment decisions). Secondly, the external injection of monetary units will be substituted by new cyclic monetary flows in the systems. Finally, randomness arise in a much more elegant way, characteristic of PDP models. This enhanced model is represented in Fig. 3.

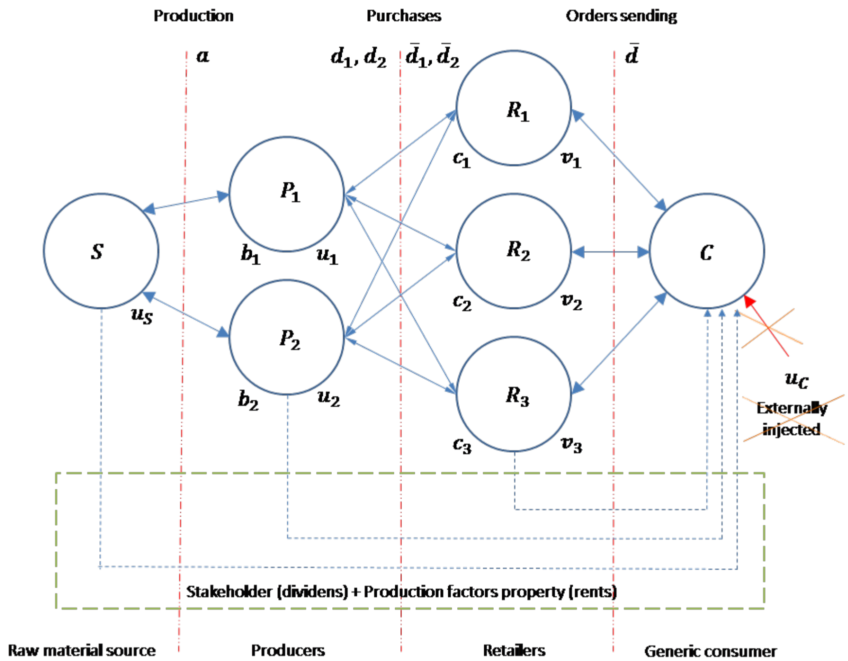


Fig. 3. Schematic representation of retailer producer problem - Enhanced Model

Randomness in a PDP-way

In our first model, the randomness of a and $\bar{d}$ was introduced in a very naive way. Making use of a bunch of rules to generate that random behavior where each rule represented a small variations around a central value. However, a more elegant "PDP-way" of generating randomness in rewriting rules has been proposed. Consider a set of generic rules with the following structure:

$[s] \rightarrow [sa^{N-L}w^{2L}]$ $\quad [w] \xrightarrow{0.5} [\#]$ $\quad [w] \xrightarrow{0.5} [a]$

Its willing is to generate approximately N units of a (a^N). First, we generate $a^{N-L}w^{2L}$, where, $N-L$ represents the lower limit of a hypothetic range for multiplicity of a and $N+L$, its upper limit. Secondly, two possible rules are applied to this new symbol w, each one with probability 0.5. Transforming w in one unit of

a or clearing it. The range of possible values for multiplicity of a is $[a^{N-L}, a^{N+L}]$. In our enhanced model, this strategy will be used at the beginning of each cycle in the amount of a generated by S, of $\bar{d}$ produced by C and in the investment decision mechanism. Further developments of the model could consider more sources of variability: in prices, in the probabilities of performing transactions between P_i and R_j.

Ownership of production factors and stakeholders

In real situations, there are no external injections of monetary units into the system to maintain it evolving. On the contrary, real economy dynamically adjust its parameters internally to maintain its activity cycle after cycle. Hence, we must consider in our model some alternatives to replace this artificially injected money. In a typical macroeconomic model, factors are property of the aggregate consumer C. Thus, P_i (and R_j as intermediate producers) must hire these factors out from its owners paying an amount of money for them (producers costs). In our model, there are only costs associated to production capacities. Secondly, in a typical economy, C is a stakeholder of P_i and R_j. Thus, the initial number of monetary units in possession of P_i and R_j can be interpreted as the initial investment of C. At the end of each cycle, stakeholders expect receiving a certain amount of dividend depending on the benefits obtained by the company. Benefits not distributed remain in the company allowing to pay costs of factors. These two mechanisms generate a flow of monetary units from u_i and v_j to u_C. Finally, to make our system closed with no external factors or agents acting on it, C must also be stakeholder of S. For simplicity, in our model, S does not need any production capacity to generate units of a (no production costs). Therefore, there will be a flow of monetary units from u_S to u_C. Provided these three sources of monetary units for u_C, there is no more need of an external injection of monetary units. In this enhanced model, the total number of monetary units flowing in the system is constant, and transactions between P_i, R_j, S and C creates monetary unit flows preventing them from accumulation.

Investment decision capacity increase

Once purchase transactions have been performed, P_i have probably obtained a surplus. Although in our initial model they simply accumulated these monetary units, in real situations they must decide what to do with their earnings. This is known as the investment saving decision. There are two choices: to dedicate part of it to accumulate more production capacity. In other words, take decisions about capital stock increase. Or instead of it, remain capacity unchanged, leaving earnings accumulated as savings. This decision, should be based on concrete facts.

For this purpose, we will extend the utility of the “authorization” system $(f_{j,i})$ created in the initial model. Basically, not exhausted authorizations will be interpreted as the existence of demand from R_j not satisfied. In an ideal situation, these $f_{j,i}$ are consumed while purchase transactions are performed. However, in some cases transactions cannot be performed but symbol $f_{j,i}$ is present (aborted transaction). If it is due to the lack of capacity, it should be increased. Otherwise, it remains unchanged.

Capital stock depreciation

Production capacity suffers a phenomenon called depreciation. There exist several economic interpretations for this depreciation: obsolescence; a reduction in the remaining value of future goods this capital stock can produce or a reduction of the market price of capital. In macroeconomic theory, the behavior of capital stock is: $K_t = K_{t-1} - D_{t-1} + I_t$, where: K_t is the capital stock value (production capacity) at time t; K_{t-1} is the capacity at time $t-1$; D_{t-1} is the depreciation of K_{t-1} and I_t is the inversion at time t. Thus, a mechanism for increasing capacity is needed to ensure recovery from depreciation if needed to satisfy orders not exhausted. For simplicity, we assume a constant depreciation rate: $D_{t-1} = \delta K_{t-1}$. Thus, the previous equation can be written as: $K_t = (1-\delta)K_{t-1} + I_t$. This depreciation can be modelled as a fixed reduction of the multiplicity of b_i.

4.2 Model formalization

The initial model must be modified to consider the new phenomena considered. New symbols are added to the working alphabet, mainly associated to the new way of random generation (p, q, m_i) and the capacity increase mechanism (g_i, y_i, z_i); meanwhile, other ones are eliminated due to the creation of monetary flows in the system (u_S, R_2): $\Gamma_{extended} = \Gamma / \{u_S, R_2\} \cup \{g_i, y_i, , m_i, z_i, h_i : 1 \leq i \leq k_1\} \cup \{p, q\}$.

Additionally, the set of rules suffer changes: a) the generation of a and $\bar{d}$ is adapted to PDP mechanism of randomness, b) P_i and R_j operation are slightly modified to consider Cs property of raw material source, c) new rules to consider payments for capacity and clearing of non-paid capacity units and, also, for capacity increase mechanism and dividend distribution and d) purchase transaction rules are adapted for the following steps of capacity depreciation. Also, new necessary parameters are considered in the model meanwhile other ones are given a new interpretation and other are no longer needed.

It is no necessary to modify the original membrane structure in our new model. Finally, this modified system will be modelled by a PDP system of degree $(2, 1)$ and $T \geq 1$ units of time $\Pi = (G, \Gamma, \Sigma, T, R_E, \mu, \mathcal{R}_\Pi, \{f_r \in \mathcal{R}_\Pi\}, M_1, M_2)$, where $G = (V, E)$, with $V = \{e_1\}$ and $E = \{(e_1, e_1)\}$, and working alphabet: $\Gamma = \{b_i, d_i, u_i, c_j, \bar{d}_j, v_j, \bar{e}_j, f_{j,i}, g_i, y_i, z_i, m_i, h_i : 1 \leq i \leq k_1, 1 \leq j \leq k_2\} \cup \{R_1\} \cup \{C, S, \bar{d}, a, u_C, p, q\}$, where:

- C: aggregate generic consumer.
- S: raw material supplier.
- a: unit of supplied raw material provided by S.

- p: randomness generator for a provision by S.
- $\bar{d}$: unit of aggregate demand from C.
- q: randomness generator for $\bar{d}$ generation by C.
- u_C: monetary unit owned by C.
- b_i: unit of production capacity of P_i, $1 \leq i \leq k_1$.
- h_i: unit of production capacity of P_i before depreciation, $1 \leq i \leq k_1$.
- d_i: unit of good supplied by P_i, $1 \leq i \leq k_1$.
- u_i: monetary unit owned by P_i, $1 \leq i \leq k_1$.
- c_j: unit of capacity of R_j, $1 \leq j \leq k_2$.
- $\bar{d}_j$: unit of good demanded by R_j, $1 \leq j \leq k_2$.
- v_j: monetary unit owned by R_j, $1 \leq j \leq k_2$.
- $\bar{e}_j$: unit of good demanded by R_j and authorized for transaction, $1 \leq j \leq k_2$.
- $f_(i,j)$: authorization for $\bar{d}_j$ to be exchanged with d_i, $1 \leq i \leq k_1$, $1 \leq j \leq k_2$.
- y_i: unit (in idle state) of aborted purchase transactions considered for capacity increase, $1 \leq i \leq k_1$.
- m_i: randomness generator for y_i, $1 \leq i \leq k_1$.
- z_i: activated unit of aborted purchase transactions considered for capacity increase, $1 \leq i \leq k_1$.
- R_1: for technical reasons.
- g_i: for technical reasons, $1 \leq i \leq k_1$.
- $\Sigma = \emptyset$.
- $R_E = \emptyset$.
- $\Pi = (\Gamma, \mu, M_1, M_2, \mathcal{R}_\Pi, \{f_r \in \mathcal{R}_\Pi\})$, where $\mu = [[\,]_2]_1$, $M_1 = \{C, S, R_1\} \cup \{g_i, u_i^{7k_{i,1}k_{10}} : 1 \leq i \leq k_1\} \cup \{v_j^{7k_{j,3}k_{10}} : 1 \leq j \leq k_2\}$ and $M_2 = \{c_j^{k_{j,3}} : 1 \leq j \leq k_2\} \cup \{b_i^{k_{i,1}} : 1 \leq i \leq k_1\}$

Model parameters

In comparison with the previous initial model, we have introduced modifications in the set of parameters. Some of them have been modified in terms of their meaning, while others have been simply relabeled. Finally, the set of parameters for our model is:

- k_1: total number of producers.
- k_2: total number of retailers.
- k_3: raw material inserted into the system by S min value of range.
- k_4: raw material inserted into the system by S max value of range.
- k_5: aggregate demand inserted into the system by C min value of range.
- k_6: aggregate demand inserted into the system by C max value of range.
- k_7: price fixed by S for each unit of a.
- k_8: number of failed purchases considered for increasing capital stock min value.
- k_9: number of failed purchases considered for increasing capital stock max value.

- k_{10}: cost of capital stock per cycle.
- k_{11}: depreciation rate of capital stock.
- k_{12}: step of capacity increase.
- k_{13}: dividend percentage.
- $k_{i,1}$: initial production capacity of P_i, $1 \leq i \leq k_1$.
- $k_{i,2}$: price fixed by P_i for each unit of d_i, $1 \leq i \leq k_1$.
- $k_{j,3}$: initial capacity of R_j, $1 \leq j \leq k_2$.
- $k_{j,6}$: price fixed by R_j for each order of good j, $1 \leq j \leq k_2$.

4.3 Modules of rules

Based on the cyclic evolution of the initial model we have expanded it including the new operations (Fig. 4): 1) generation of aggregate demand and raw material disposability and rents payment for P_i's and R_j's capacity, 2) production of goods and reception of orders, 3) generation of the authorizations for purchase transactions, 4) purchase transactions and 5) capacity increase decision, capacity depreciation and dividend payment.

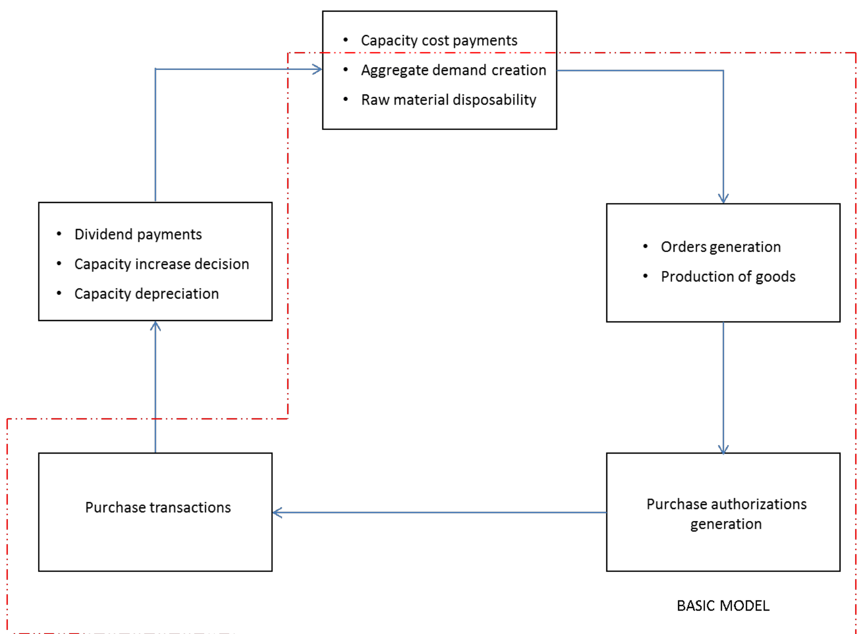


Fig. 4. Modules of rules Enhanced Model

Module 1: Production factor and demand generation

Our first step consists in generating the initial conditions of the cycle: restoring aggregate demand and ensuring raw material disposability. Unlike the initial model, we will use the PDP-way of generating randomness in the amount of generated a and $\bar{d}$. Using new symbols p (respectively, q) to control the range $[k_3, k_4]$ (respectively, $[k_5, k_6]$) of possible values of a (respectively, $\bar{d}$). This operation can be performed with a simple set of rules:

$$r_1 \equiv R_1 sc\,[\,]_2^- \rightarrow a^{k_3} p^{k_4-k_3} \bar{d}^{k_5} q^{k_6-k_5} sc\,[\,]_2^+$$

$$r_2 \equiv p\,[\,]_2^- \xrightarrow{p=0.5} a\,[\,]_2^+ \qquad r_4 \equiv q\,[\,]_2^- \xrightarrow{p=0.5} [\,]_2^+$$
$$r_3 \equiv p\,[\,]_2^- \xrightarrow{p=0.5} [\,]_2^+ \qquad r_5 \equiv q\,[\,]_2^- \xrightarrow{p=0.5} \bar{d}\,[\,]_2^+$$

Additionally, The generation of a and $\bar{d}$ is unified in a single rule but rules on p and q remain separated allowing their independent random behaviour. Also, any reference to the spontaneous appearance of monetary units is removed from the rules taking place at this step.

Module 2: Producers costs

The idea of Cs property of factors means that R_j and P_i must pay, at the beginning of each cycle, for their capacities c_j and b_i. For simplicity, a unique common parameter k_{10} has been selected for both capacities cost. For each unit of b_i used by P_i it must pay a price, reducing the number of u_i owned by P_i and increasing the u_C owned by C. Similarly, for each unit of c_i used by R_j it must pay a price, reducing the number of v_j owned by R_j and increasing the ones u_C owned by C. If R_j and P_i do not have enough monetary units to pay for their capacities, they must give up using them and restore the value of each of these capacity units to their proprietaries via u_C units.

$$r_9 \equiv u_i^{k_{10}}\,[b_i]_2 \rightarrow b_i u_C^{k_{10}}\,[\,]_2^+, 1 \le i \le k_1 \qquad r_9 \equiv v_j^{k_{10}}\,[c_j]_2 \rightarrow c_j u_C^{k_{10}}\,[\,]_2^+, 1 \le j \le k_2$$

$$r_{11} \equiv [b_i]_2^+ \rightarrow u_C^{k_{10}}[\,]_2, 1 \le i \le k_1 \qquad r_{11} \equiv [c_j]_2^+ \rightarrow u_C^{k_{10}}[\,]_2, 1 \le j \le k_2$$

Module 3: Producers & retailers operations

P_is rules are slightly modified to include another big conceptual change of our new model: C is the owner of S. This idea means a revolution in the system: in the initial model, S simply accumulated the monetary units u_S. Now, these units travel from P_i to S and, again, return to C. For each unit of a used by P_i it must pay a price (k_7 monetary units), reducing the number of u_i owned by P_i and increasing the ones u_C owned by C.

$$r_14 \equiv a b_i u_i^{k_7} [\,]_2^+ \rightarrow u_C^{k_7} [d_i]_2^0, 1 \leq i \leq k_1$$

R_j must face to an amount of aggregate demand generated previously competing to catch units of $\bar{d}$ to serve necessities of C. It also can be interpreted as R_j receives orders from C. We will allow each retailer to fix a different price. For each unit of $\bar{d}$ ordered by C to R_j it must pay a price $k_{j,6}$, reducing the number of u_C owned by C and increasing the ones v_j owned by R_j. Finally, capacity units not consumed are transferred out of membrane 1, waiting for later depreciation operations.

$$\begin{aligned}
r_{15} &\equiv \bar{d} c_j u_C^{k_{j,6}} [\,]_2^+ \rightarrow [\bar{d}_j v_j^{k_{j,6}}]_2^0, 1 \leq j \leq k_2\\
r_{16} &\equiv b_i [\,]_2 \rightarrow [b_i]_2, 1 \leq i \leq k_1\\
r_{17} &\equiv c_j [\,]_2 \rightarrow [c_j]_2, 1 \leq j \leq k_2
\end{aligned}$$

Module 4: Purchase transactions

This module remains almost unchanged with respect to the initial model. A first step of generation of transaction authorizations. Once generated, we can perform the purchase transactions but now with probability one. Again, the discrete probability distributions are embedded in the authorizations generation rules. One item of d_i is purchased by R_j from P_i to satisfy the order $\bar{e}_j$ carried by R_j. For each unit of d_i, R_j must pay a price, reducing the number of v_j and increasing the ones u_i owned by P_i. Capacities c_j and b_i consumed producing d_i and $\bar{e}$ are freed. Finally, free b_i are transformed into new symbols h_i waiting for depreciation operations.

$$\begin{aligned}
r_{18} &\equiv [\bar{d}_1]_2 \xrightarrow{p_{1,1}=1} [\bar{e}_1 f_{1,1}]_2 & r_{19} &\equiv [\bar{d}_1]_2 \xrightarrow{p_{1,2}=1} [\bar{e}_1 f_{1,2}]_2\\
r_{20} &\equiv [\bar{d}_2]_2 \xrightarrow{p_{2,1}=0.5} [\bar{e}_2 f_{2,1}]_2 & r_{21} &\equiv [\bar{d}_2]_2 \xrightarrow{p_{2,2}=0.5} [\bar{e}_2 f_{2,2}]_2\\
r_{22} &\equiv [\bar{d}_3]_2 \xrightarrow{p_{3,1}=0.15} [\bar{e}_3 f_{3,1}]_2 & r_{23} &\equiv [\bar{d}_3]_2 \xrightarrow{p_{3,2}=0.85} [\bar{e}_3 f_{3,2}]_2
\end{aligned}$$

$$r_{24} \equiv [d_i \bar{e}_j f_{j,i} v_j^{k_{i,2}}]_2 \rightarrow u_i^{k_{i,2}} [h_i c_j]_2^-, 1 \leq i \leq k_1, 1 \leq j \leq k_2$$

Further developments of the model could explore how to consider dynamic probabilities for these transactions.

Module 5: Dividends distribution

Once purchase transactions have been performed, the remaining monetary units owned by R_j and P_i can be interpreted as their benefits. On the other hand, C can be interpreted as stakeholder of R_j and P_i, and their initial monetary units can be considered the amount of money already invested by them. In this context, a dividend payment can be considered. This dividend percentage is controlled by parameter k_{13}. For simplicity, this will be considered only in P_i:

$$r_{25} \equiv [v_j]_2^- \rightarrow v_j\,[\,]_2^0, 1 \leq j \leq k_2$$
$$r_{26} \equiv [u_i]_2^- \xrightarrow{p=k_{13}} u_C\,[\,]_2^0, 1 \leq i \leq k_1$$
$$r_{27} \equiv [u_i]_2^- \xrightarrow{p=1-k_{13}} u_i\,[\,]_2^0, 1 \leq i \leq k_1$$

Module 6: Capacity depreciation

As explained in previous chapter, depreciation will be considered as a reduction of production capacity. For simplicity, it will only be considered a P_i's capacity depreciation. This can be easily modelled as a reduction of b_i's multiplicity with a probability controlled by parameter k_{11}, representing capacity disappearance rate.

$$r_{31} \equiv [h_i]_2^- \xrightarrow{p=1-k_{11}} [b_i]_2^0, 1 \leq i \leq k_1 \qquad r_{32} \equiv [h_i]_2^- \xrightarrow{p=k_{11}} [\,]_2^0, 1 \leq i \leq k_1$$

The global evolution suffered by b_i can be outlined in the following flow:

$$[b_i]_2^0 \xrightarrow{payrents} b_i[\,]_2^+ \xrightarrow{productionofgoods} [h_i]_2^- \xrightarrow{depreciationrules} [b_i]_2^0$$

Further developments of the model could extend the depreciation rules to all actors' capacities of the system. Indeed, the application of this depreciation rules to any capacity is paired to the necessity of a production capacity increase decision mechanism. If only depreciation acts, it will be reached a point of capacity exhaustion that stops system evolution.

Module 7: Capacity increase decision

The number of aborted transactions considered to increase capacity is controlled by the multiplicity of a new symbol m_i. This will be arbitrarily low to generate a reasonably rate of capacity increasing. Additionally, randomness will be included in the generation of symbol m_i.

In the previous sections, we analysed the circumstances accompanying a non-performed authorized purchase transaction and determined that this could be a good signal to trigger a capacity increase mechanism. If it is not due to a lack of producer capacity, it is not necessary to increase it.

$$r_{28} \equiv [f_{j,i}d_i]_2^- \rightarrow \quad [d_i]_2^0,\ 1 \leq i \leq k_1, 1 \leq j \leq k_2$$
$$r_{29} \equiv [f_{j,i}h_i]_2^- \xrightarrow{1-k_{11}} [b_i]_2^0,\ 1 \leq i \leq k_1, 1 \leq j \leq k_2$$
$$r_{30} \equiv [f_{j,i}h_i]_2^- \xrightarrow{k_{11}} \quad [\,]_2^0,\quad 1 \leq i \leq k_1, 1 \leq j \leq k_2$$

Otherwise, it is necessary to increase it (controlled by parameter k_{12}). To ensure a gradual adaptation of capacity it will be introduced a limit to the number of considered aborted transactions (represented by multiplicity of symbol y_i). Additionally, to incorporate randomness into the process, rules in the PDP-way will be included for the generation of symbol y_i, using symbol m_i. The range of values for

y_i varies in range $[k_8, k_9]$. Symbol z_i is simply an evolved form of y_i to determine the exact moment of activating this operation. Finally, non-exhausted units of $f_{j,i}$ and z_i are removed.

$$r_6 \equiv g_i[\,]_2^0 \rightarrow [g_i y_i^{k_8} m_i^{k_9-k_8}]_2^+, 1 \leq i \leq k_1$$
$$r_7 \equiv [m_i]_2^+ \xrightarrow{0.5} [\,]_2^0, 1 \leq i \leq k_1 \qquad r_8 \equiv [m_i]_2^+ \xrightarrow{0.5} [y_i]_2^0, 1 \leq i \leq k_1$$
$$r_{33} \equiv [y_i]_2^- \rightarrow [z_i]_2^0, 1 \leq i \leq k_1$$
$$r_{34} \equiv [f_{j,i} z_i]_2^0 \rightarrow b_i^{k_{12}}[\,]_2^+, 1 \leq i \leq k_1, 1 \leq j \leq k_2$$
$$r_{35} \equiv [f_{j,i}]_2^+ \rightarrow [\,]_2^0, 1 \leq i \leq k_1, 1 \leq j \leq k_2 \qquad r_{36} \equiv [z_i]_2^+ \rightarrow [\,]_2^0, 1 \leq i \leq k_1$$

Technical & cleaning rules

Finally, some rules are necessary for technical reasons. $\bar{e}_j$ and v_j not exchanged represent real received orders and monetary units so cannot be eliminated.

$$r_{13} \equiv v_j[\,]_2^+ \rightarrow [v_j]_2^0, 1 \leq j \leq k_2 \qquad r_{37} \equiv [\bar{e}_j]_2^+ \rightarrow [\bar{d}_j]_2^0, 1 \leq j \leq k_2$$

Symbols r_1 and g_i are restored to their initial location. r_1 controls the generation of a and $\bar{d}$, while g_i controls the generation of symbols y_i.

$$r_{38} \equiv [r_1]_2^- \rightarrow r_1[\,]_2^0 \qquad r_{39} \equiv [g_1]_2^- \rightarrow g_1[\,]_2^0, 1 \leq j \leq k_2$$

4.4 Simulation results

To make these results comparable with the ones obtained from the previous model, we will use a similar set of values. The complete relation of parameters is:

Simulations will be performed (again 200 cycles with 5 steps in each cycle using DNDP-4 algorithm as inference engine) for different situations to show the effect of each phenomenon included in the model, along with its contribution to the global behavior and stability of the system. In the following section, several situations are discussed. Case A: (capacity depreciation standalone) we will show producers capacity evolution when depreciation rate = 0.1 and capacity increase mechanism is deactivated. Initial capacity of producers has been set to $k_{1,1} = 65$ and $k_{2,1} = 35$. As expected, along the cycles capacities are reduced until they are completely exhausted. The slope of these curves is controlled by k_{11}. As seen in the previous chapters, a mechanism of capacity increase is necessary to maintain the evolution of the system.

Case B (capacity depreciation + capacity increase mechanism): we will show how the previous evolution changes when capacity increase mechanism and dividend payment mechanisms are activated. These mechanisms parameters are $k_{12} = 1$ (step of capacity increase), $k_8 = 3$ and $k_9 = 5$ (range of aborted purchase transaction considered), and $k_{13} = 0.01$ (dividend percentage). As expected, in a cycle each producers producer capacities suffer depreciation. This diminishing of production generates the abortion of multiple purchase transactions that activate

Parameter	Value/s	Description
k_1	2	Total number of producers
k_2	3	Total number of retailers
k_3	59	Units of a inserted into the system by S min value of range
k_4	62	Units of a inserted into the system by S max value of range
k_5	59	Units of $\bar{d}$ inserted into the system by C min value of range
k_6	62	Units of $\bar{d}$ inserted into the system by C max value of range
k_7	11	Price fixed by S for each unit of a
k_8	3	# failed purchases considered for increasing capital min value
k_9	5	# failed purchases considered for increasing capital max value
k_{10}	2	cost of capital stock per cycle
k_{11}	0.1	depreciation rate of capital stock
k_{12}	1	step of capacity increase
k_{13}	0.01	Dividend percentage
$k_{i,1}$	$(65, 35)$	Initial production capacity of P_i, $1 \leq i \leq k_1$
$k_{i,2}$	$(13, 13)$	Price fixed by P_i for each unit of d_i
$k_{j,3}$	$(50, 30, 20)$	Initial capacity of R_j, $1 \leq j \leq k_2$
$k_{j,6}$	$(15, 15, 15)$	Price fixed by R_j for each order of good j, $1 \leq j \leq k_2$

Table 2. Parameters utilized for simulation of enhanced model

the capacity increase mechanism. As we have seen in the previous chapters, during the evolution of the system, depreciation mechanism pushes capacity down and capacity increase mechanism competes with the previous one to maintain system evolution alive.

Case C: (capacity depreciation + capacity increase mechanism + dividend payment deactivated). In this case, we will show how the previous evolution changes when dividend payment mechanism is deactivated. Depreciation mechanism pushes capacity down, capacity increase mechanism competes with the previous one to maintain system evolution alive but all these processes are not possible is not possible if there are no enough movement of monetary units between all the actors in the system.

Once situation is restored to case B, the following stable behavior of u_C is obtained:

Clearly these three mechanisms cooperation (capacity depreciation, capacity increase decision and monetary unit flow mechanisms) is crucial to the stable

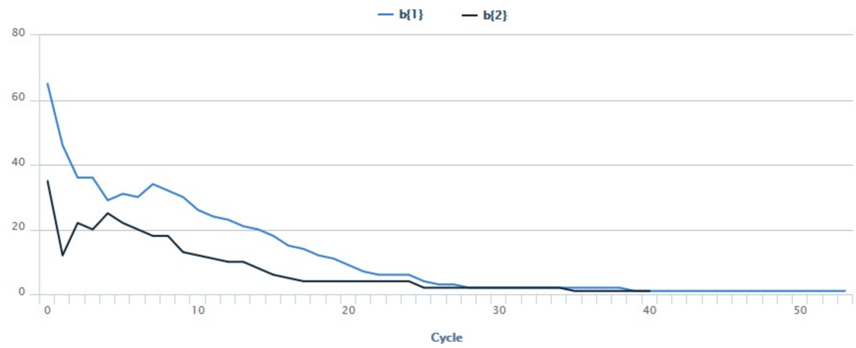


Fig. 5. Evolution of b_i in presence of depreciation standalone

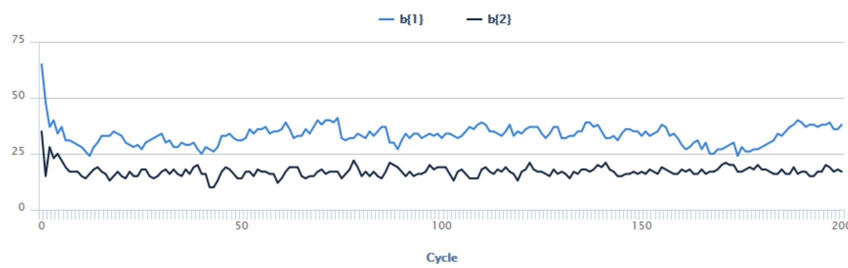


Fig. 6. Evolution of b_i in presence of depreciation and capacity increase mechanism

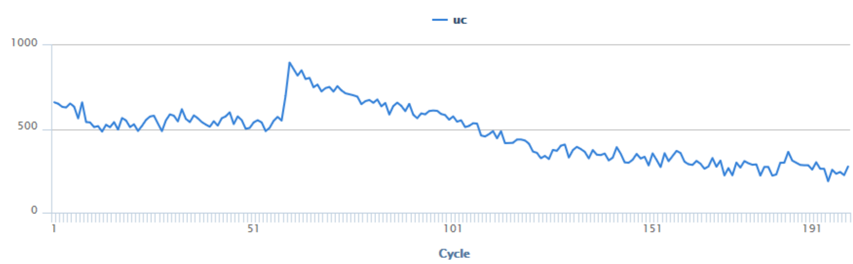


Fig. 7. Evolution of C monetary units with dividend payment mechanism deactivated

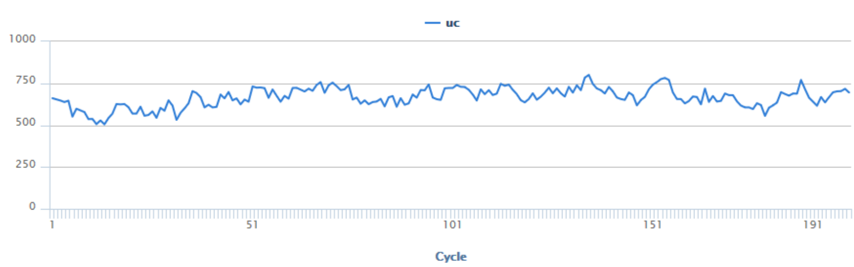


Fig. 8. Evolution of C monetary units with dividend payment mechanism activated

evolution of the system. Next figures show the evolution of the rest of variables of the system.

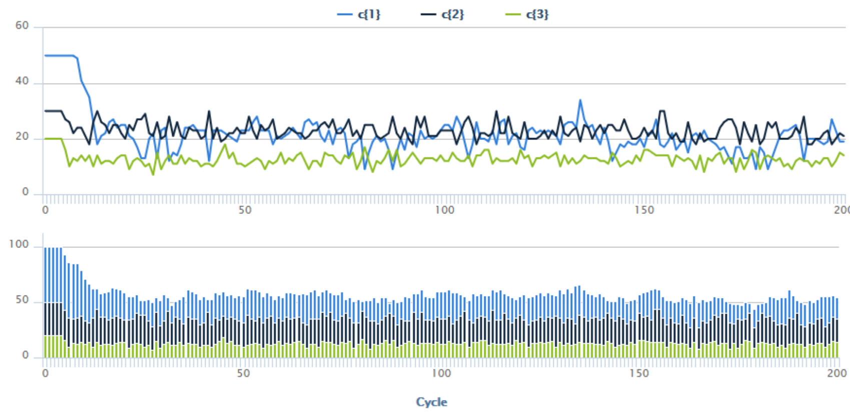


Fig. 9. Evolution of retailers capacities in line chart and accumulated columns

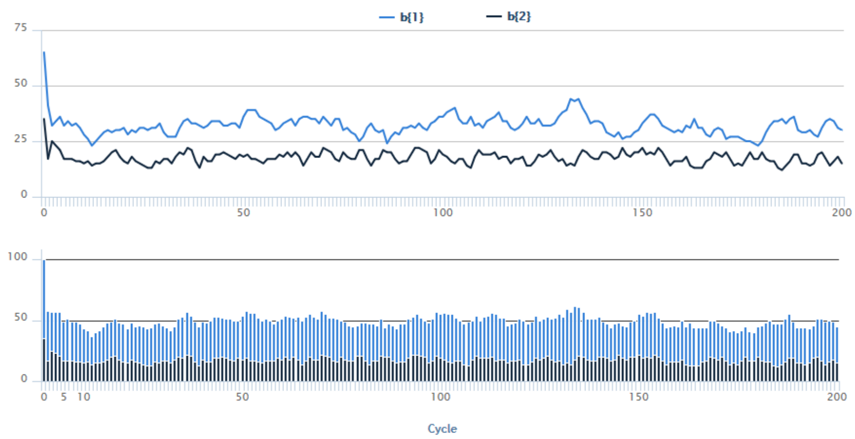


Fig. 10. Evolution of producers capacities in line chart and accumulated columns

5 Conclusions

In our work, we have implemented the ideas sketched by Gh. and R. Păun [10] into a specific P system framework (more specifically, PDP systems). The results

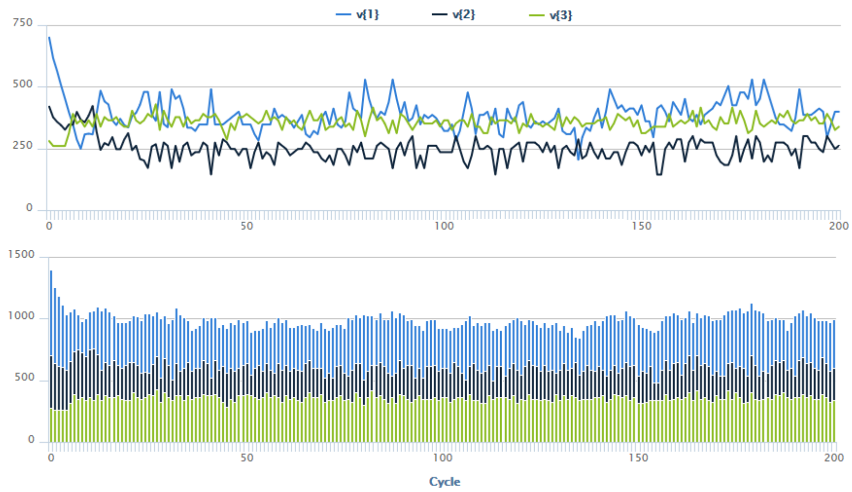


Fig. 11. Evolution of retailers monetary units in line chart and accumulated columns

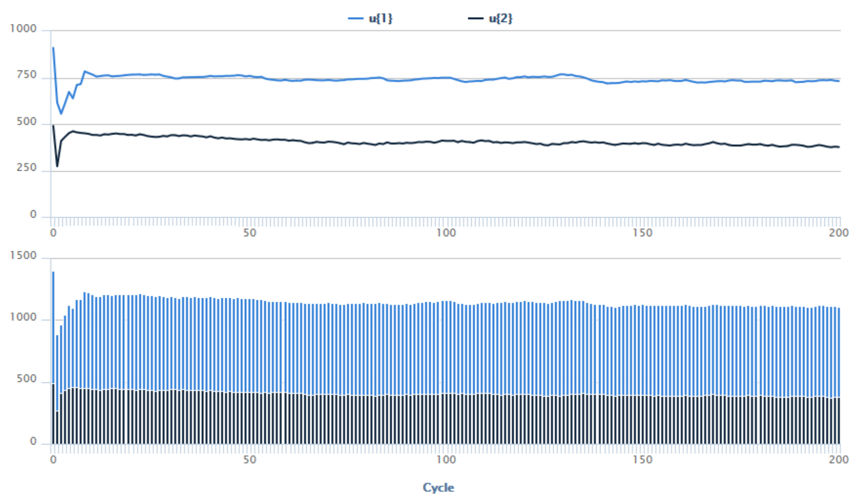


Fig. 12. Evolution of producers monetary units in line chart and accumulated columns

obtained look very promising. Firstly, we have been able to replicate Păuns' numerical results, designing a model for the producer-retailer problem and simulating it using `MeCoSim` [12]. In this initial model, some basic interactions between producers and retailers are considered, such as, production of goods from raw material, reception of orders from consumers and purchase transaction to match this goods and orders. It includes several simplifications and an exogenous artificial injection of monetary units that prevents system from reaching a halting configuration.

Aimed by these results we have proposed an enhanced model for the producer-retailer problem. The initial model has been enriched considering a plethora of phenomena to move it closer to the complexities of real world. We have taken advantage from **modularity**, one of the main membrane computing advantages for systems modelling. Hence, it has not been necessary to build again the model from scratch, we have added complexity to the model over the initial layer. Many real economic world interactions have been included in the model as a new layer: capital stock depreciation, capacity increase decision mechanism, costs of capital (rents for its owners), dividend payments, and a general idea of making monetary units flow across the system. Additionally, randomness has been introduced, in several steps of the model, by means of mechanisms frequently used in PDP world.

This enhanced model has also been simulated using `MeCoSim` and system evolution was analysed in depth. Some remarkable facts are that system can evolve autonomously without any exogenous influence. Although initial values of variables are settled, they change their values reaching an equilibrium point. Once reached this stability point, system varies slightly around it. From the previous results, we can derive that multiple economic issues can be modelled using membrane computing. Therefore, more efforts must be done in this direction.

Acknowledgements

This work was partially supported by Grant numbers 61472328 and 61320106005 of the National Natural Science Foundation of China.

References

1. J. Bartosik: Paun's systems in modeling of human resource management. *Second Conference on Tools and Methods of Data Transformation*, WSU Kielce, 2004.
2. M. Cardona, M.A. Colomer, M.J. Prez-Jimnez, Sanuy, D., Margalida, A., 2009. Modeling ecosystems using P Systems: The Bearded vulture, a case study. *Lecture Notes in Computer Science*, **5391** (2009), 137156.
3. S. Cheruku, A. Păun, F.J. Romero, M.J. Pérez-Jiménez, O.H. Ibarra. Simulating FAS-induced apoptosis by using P systems. *Progress in Natural Science*, **17**, 4 (2007), 424-431.
4. M.A. Colomer, S. Lavin, I. Marco, A. Margalida, I. Prez-Hurtado, M.J. Prez-Jimnez, D. Sanuy, E. Serrano, and L. Valencia-Cabrera. Modeling population growth of pyrenean chamois (rupicapra p. pyrenaica) by using p-systems. *Lecture Notes in Computer Science*, **6501** (2011), 144-159.
5. J.M. Epstein, R. Axtell. *Growing Artificial SocietiesSocial Science from the Bottom Up*. Brookings Institution Press and The MIT Press, 1996.
6. García-Quismondo, M., Gutiérrez-Escudero, R., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A. An overview of P-Lingua 2.0. *Lecture Notes in Computer Science*, 5957 (2010), 264-288.

7. W. Korczynski. On a model of economic systems. *Second Conference on Tools and Methods of Data Transformation*, WSU Kielce, 2004.
8. W. Korczynski. Paun's systems and accounting. In *Pre-Proceedings of Sixth Workshop on Membrane Computing*, WMC6, Vienna, July 2005, 461-464.
9. Gh. Păun. *Membrane Computing: An introduction*. Springer, 2002.
10. Gh. Păun, R. Păun. Membrane computing as a framework for modeling economic processes. *Proceedings of SYNASC 05*, Timisoara, Romania, IEEE Press, 2005, 1118.
11. Gh. Păun, R. Păun. Membrane Computing and Economics. In Gh. Păun, G. Rozenberg, A. Salomaa, (eds.) *The Oxford Handbook of Membrane Computing*. Oxford University Press, New York, 2010, 632-644.
12. I. Pérez-Hurtado, L. Valencia-Cabrera, M.J. Pérez-Jiménez, M.A. Colomer, A. Riscos-Núñez. Mecosim: A general purpose software tool for simulating biological phenomena by means of p systems. *IEEE Fifth International Conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010)*, I (2010), 637–643.