Evolutionary Design of a Simple Membrane System

X.L. Huang, G.X. Zhang and F. Ipate

Aug. 2011, France

Outline

- Motivation
- Evolutionary Design
- Experimental Results
- Conclusions

Motivation (1)

- Work on Membrane Computing
  - Much attention paid to membrane computing in the area of natural computing
  - Various membrane systems well defined
  - Little work on automatic design of a membrane system

Motivation (2)

- Difficulties in Designing a Membrane System
  - A problem about the programmability of membrane systems
  - An ongoing and challenging issue
  - Still difficult to design a membrane system, even a simple one

Motivation (3)

Main idea:

Presenting an approach for automatically designing a membrane system by employing a quantum-inspired evolutionary algorithm (QIEA)

- Main points in the Evolutionary Design:
  - A pre-defined membrane structure, initial objects, a common set of evolution rules
  - Encoding membrane systems
  - Evolutionary operators
  - Evaluating a membrane system

Evolutionary Design (1)

- Problem Statement
  - Considering only cell-like P systems
  - Given an initial configuration:
    - A membrane structure μ
    - An alphabet of objects O
    - Initial multisets $W = \{w_1, w_2, \ldots, w_m\}$
  - Pre-defining a set of (redundant) rules $\Re$ that can be used for solving a class of problems

Evolutionary Design (2)

The problem can be summarized as follows:

Using a QIEA to find a proper set of rules, by appropriately evolving a P system for a specific task.

- A Family of P Systems

$$\prod = \{\Pi_1, \Pi_2, \ldots, \Pi_i\}_{i \in I}$$

  - with an identically initial configuration
  - With different sets of rules R_i ($R_i \subseteq \Re$ is a subset of the pre-defined $\Re$)

Evolutionary Design (3)

- The Pseudocode Algorithm of a QIEA

```
Begin
    t ← 1
1)  Generate an initial population Q(t) represented by
    Q-bit individuals;
      While (not termination-condition) do
2)      Make a binary population X(t) by observing the
        states of Q(t);
3)      Evaluate all the individuals in X(t);
4)      Update Q(t) using Q-gates and produce offspring;
        t ← t + 1
      End
End
```

Evolutionary Design (4)

1) Population initialization

$$Q(t) = \left\{ q_1^t, q_2^t, \ldots, q_n^t \right\}$$

Where $$q_j^t = \begin{bmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \cdots & \alpha_{jk}^t \\ \beta_{j1}^t & \beta_{j2}^t & \cdots & \beta_{jk}^t \end{bmatrix} \quad (j=1,2,\ldots,n)$$

$$\left|\alpha_{ji}^t\right|^2 + \left|\beta_{ji}^t\right|^2 = 1 \qquad (i = 1, 2, \ldots, k)$$

Initial values: $\alpha_{ji}^t = \pm \frac{1}{\sqrt{2}}, \ \beta_{ji}^t = \pm \frac{1}{\sqrt{2}} \quad (t = 1)$

Evolutionary Design (5)

2) Observation

$$[\alpha\ \beta]^T \rightarrow x \text{ (binary)}$$

Begin

If *random* $[0,\ 1) < |\alpha|^2$

Then $x \leftarrow 0$

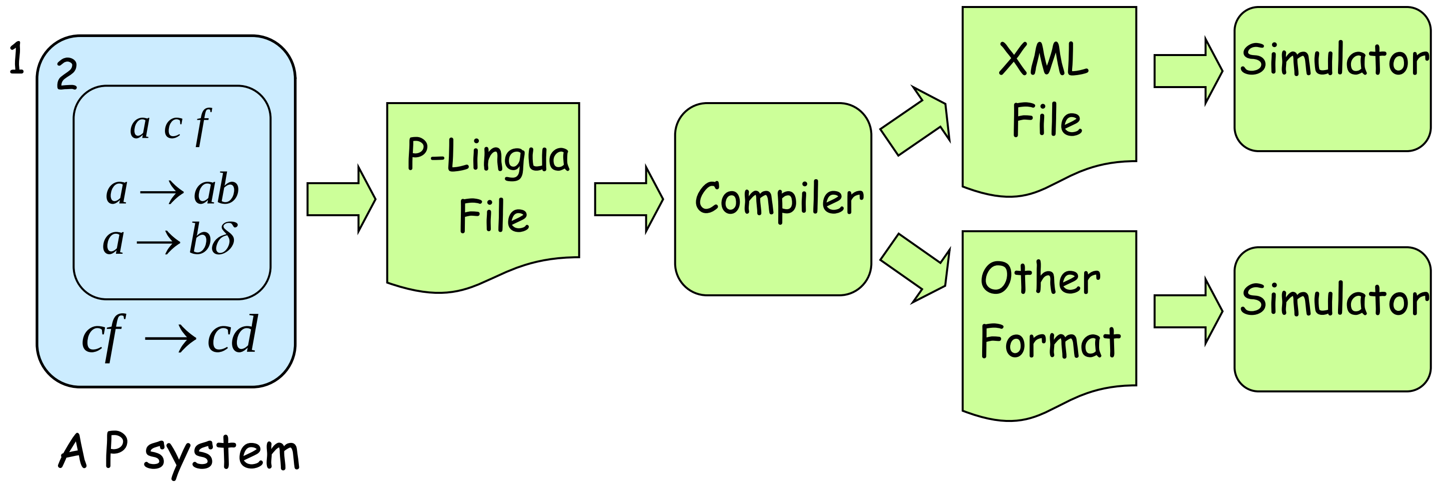
Else $x \leftarrow 1$

End

Evolutionary Design (6)

3) Evaluation

$$fitness = |\text{simulation result} - \text{expected result}|$$



The simulation of a P system based on the P-Lingua simulator

Evolutionary Design (7)

4) Update

$$\begin{bmatrix} \alpha_{ji}^{t+1} \\ \beta_{ji}^{t+1} \end{bmatrix} = G_{ji}^{t}(\theta) \begin{bmatrix} \alpha_{ji}^{t} \\ \beta_{ji}^{t} \end{bmatrix}$$

Q-gate: $G_{ji}^{t}(\theta) = \begin{bmatrix} \cos\theta_{ji}^{t} & -\sin\theta_{ji}^{t} \\ \sin\theta_{ji}^{t} & \cos\theta_{ji}^{t} \end{bmatrix}$

θ_{ji}^{t} --an adjustable rotation angle

Experimental Results (1)

- Two examples:

  (1) A P system for computing n^2

  (2) A P system for computing 4^2

  (bringing a comparison with the method in [11] (Escuela and Gutiérrez-Naranjo,2010))

- Platform for the experiments:

  Eclipse 3.5.1, JDK 1.6.0 and P-Lingua 2.1

Experimental Results (2)

1) An example for computing n^2

- Initial configuration: $\Pi = (O, H, \mu, w_1, w_2, R)$

$$O = \{a, b, c, d, f\} \quad \mu = [[\ ]_2]_1 \quad w_1 = \varnothing \quad w_2 = a^2bd$$

- A predefined common set of rules $\mathfrak{R}$:

$$\mathfrak{R} = \left\{ \begin{array}{lll} r_1 \equiv [a \to ab]_2 & r_7 \equiv [c]_2 \to a & r_{13} \equiv [b \to \lambda]_1 \\ r_2 \equiv [b \to bc]_2 & r_8 \equiv [b \to c]_2 & r_{14} \equiv [d \to \lambda]_1 \\ r_3 \equiv [c]_2 \to b & r_9 \equiv [a^2 \to c]_2 & r_{15} \equiv [f \to \lambda]_1 \\ r_4 \equiv [a \to bc]_2 & r_{10} \equiv [c \to \lambda]_2 & r_{16} \equiv [f^n]_2 \to \lambda \\ r_5 \equiv [a \to c]_2 & r_{11} \equiv [d \to d\ f]_2 & r_{17} \equiv [f^{(n+1)}]_2 \to \lambda \\ r_6 \equiv [b^2 \to c]_2 & r_{12} \equiv [a \to \lambda]_1 & r_{18} \equiv [f^{(n-1)}]_2 \to \lambda \end{array} \right\}$$

Experimental Results (3)

- Independent 30 runs: 21 successes, 9 failures to obtain any desired P systems
- The 21 successful runs: 12 P systems are distinct and their sets of rules are as follows:

$$R_1 = \left\{ \begin{array}{ll} [a \to ab]_2 & [d \to \lambda]_1 \\ [b \to bc]_2 & [f \to \lambda]_1 \\ [d \to d\ f]_2 & \left[f^{(n-1)} \to \lambda \right]_1 \\ [b \to \lambda]_1 & \end{array} \right\} \qquad R_2 = \left\{ \begin{array}{ll} [a \to ab]_2 & [b \to \lambda]_1 \\ [b \to bc]_2 & [d \to \lambda]_1 \\ [d \to d\ f]_2 & \left[f^{(n-1)} \to \lambda \right]_1 \\ [a \to \lambda]_1 & \end{array} \right\}$$

$$R_3 = \left\{ \begin{array}{ll} [a \to ab]_2 & [a \to \lambda]_1 \\ [b \to bc]_2 & [d \to \lambda]_1 \\ [d \to df]_2 & \left[f^{(n-1)} \right]_2 \to \lambda \end{array} \right\} \qquad R_4 = \left\{ \begin{array}{ll} [a \to ab]_2 & [b \to \lambda]_1 \\ [b \to bc]_2 & [f \to \lambda]_1 \\ [d \to df]_2 & \left[f^{(n-1)} \right]_2 \to \lambda \end{array} \right\}$$

Experimental Results (4)

$$R_5 = \left\{ \begin{array}{ll} [a \to ab]_2 & [d \to \lambda]_1 \\ [b \to bc]_2 & [f \to \lambda]_1 \\ [d \to df]_2 & \left[f^{(n-1)}\right]_2 \to \lambda \end{array} \right\}$$

$$R_6 = \left\{ \begin{array}{ll} [a \to ab]_2 & [a \to \lambda]_1 \\ [b \to bc]_2 & [f \to \lambda]_1 \\ [d \to df]_2 & \left[f^{(n-1)}\right]_2 \to \lambda \end{array} \right\}$$

$$R_7 = \left\{ \begin{array}{ll} [a \to ab]_2 & [d \to \lambda]_1 \\ [b \to bc]_2 & [f \to \lambda]_1 \\ [d \to d\ f]_2 & \left[f^{(n-1)} \to \lambda\right]_1 \\ [a \to \lambda]_1 & \end{array} \right\}$$

$$R_8 = \left\{ \begin{array}{ll} [a \to ab]_2 & [b \to \lambda]_1 \\ [b \to bc]_2 & [f \to \lambda]_1 \\ [d \to d\ f]_2 & \left[f^{(n-1)} \to \lambda\right]_1 \\ [a \to \lambda]_1 & \end{array} \right\}$$

$$R_9 = \left\{ \begin{array}{ll} [a \to ab]_2 & [d \to \lambda]_1 \\ [b \to bc]_2 & \left[f^{(n-1)}\right]_2 \to \lambda \\ [d \to df]_2 & \end{array} \right\}$$

$$R_{10} = \left\{ \begin{array}{ll} [a \to ab]_2 & [a \to \lambda]_1 \\ [b \to bc]_2 & \left[f^{(n-1)}\right]_2 \to \lambda \\ [d \to df]_2 & \end{array} \right\}$$

Experimental Results (5)

$$R_{11} = \begin{Bmatrix} [a \to ab]_2 & [f \to \lambda]_1 \\ [b \to bc]_2 & \left[f^{(n-1)}\right]_2 \to \lambda \\ [d \to df]_2 & \end{Bmatrix} \qquad R_{12} = \begin{Bmatrix} [a \to ab]_2 & [d \to df]_2 \\ [b \to bc]_2 & [f^{(n-1)}]_2 \to \lambda \end{Bmatrix}$$

Table 1

Experiments		Successful runs/total runs	Average evolutionary generations	Time per run (s)
QIEA	n^2	21/30	18.47	2.97

Experimental Results (6)

2) Comparison with Escuela and Gutierrez-Naranjo [11]

- Initial configuration: $\Pi = (O, H, \mu, w_1, w_2, R)$

$$O = \{a, b, c, z_1, z_2, z_3, z_4\} \quad \mu = [[\ ]_2]_1 \quad w_1 = \varnothing \quad w_2 = a^2 b z_1$$

- The predefined common set of rules $\mathfrak{R}$:

$$\mathfrak{R} = \begin{Bmatrix} r_1 \equiv [a \to ab]_2 & r_7 \equiv [z_2 \to z_1]_2 & r_{13} \equiv [a \to \lambda]_1 \\ r_2 \equiv [b \to bc]_2 & r_8 \equiv [z_3 \to z_4]_2 & r_{14} \equiv [b \to \lambda]_1 \\ r_3 \equiv [c \to b^2]_2 & r_9 \equiv [z_1]_2 \to b & r_{15} \equiv [b \to c]_2 \\ r_4 \equiv [a \to bc]_2 & r_{10} \equiv [z_2]_2 \to a & r_{16} \equiv [c \to \lambda]_2 \\ r_5 \equiv [z_1 \to z_2]_2 & r_{11} \equiv [z_3]_2 \to c & r_{17} \equiv [z_4 \to z_1]_2 \\ r_6 \equiv [z_2 \to z_3]_2 & r_{12} \equiv [z_4]_2 \to a & r_{18} \equiv [z_4]_2 \to b \end{Bmatrix}$$

Experimental Results (7)

- 30 independent runs: 12 successes, 18 failures to obtain any desired P systems.

- The 12 successful runs: 7 P systems are distinct and their sets of rules are as follows:

$$R_1 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_3 \to z_4]_2 \\ [b \to bc]_2 & [a \to \lambda]_1 \\ [z_1 \to z_2]_2 & [b \to \lambda]_1 \\ [z_2 \to z_3]_2 & [z_4]_2 \to b \end{array} \right\} \qquad R_2 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_3 \to z_4]_2 \\ [b \to bc]_2 & [a \to \lambda]_1 \\ [z_1 \to z_2]_2 & [b \to \lambda]_1 \\ [z_2 \to z_3]_2 & [z_4]_2 \to a \end{array} \right\}$$

Experimental Results (8)

$$R_3 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_2 \to z_3]_2 \\ [b \to bc]_2 & [z_3 \to z_4]_2 \\ [z_1 \to z_2]_2 & [z_4]_2 \to a \end{array} \right\} \qquad R_4 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_2 \to z_3]_2 \\ [b \to bc]_2 & [z_3 \to z_4]_2 \\ [z_1 \to z_2]_2 & [z_4]_2 \to b \end{array} \right\}$$

$$R_5 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_3 \to z_4]_2 \\ [b \to bc]_2 & [a \to \lambda]_1 \\ [z_1 \to z_2]_2 & [z_4]_2 \to b \\ [z_2 \to z_3]_2 & \end{array} \right\} \qquad R_6 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_3 \to z_4]_2 \\ [b \to bc]_2 & [a \to \lambda]_1 \\ [z_1 \to z_2]_2 & [z_4]_2 \to a \\ [z_2 \to z_3]_2 & \end{array} \right\}$$

$$R_7 = \left\{ \begin{array}{ll} [a \to ab]_2 & [z_3 \to z_4]_2 \\ [b \to bc]_2 & [b \to \lambda]_1 \\ [z_1 \to z_2]_2 & [z_4]_2 \to a \\ [z_2 \to z_3]_2 & \end{array} \right\}$$

Experimental Results (8)

- Comparison

Table 2

Experiments		Successful runs/total runs	Average evolutionary generations	Time per run (s)
QIEA	4^2	12/30	20.97	3.13
[11]	4^2	1/30	--	--

Conclusions

- Presenting a clear possibility on how to use a QIEA to design a desired P system
- Characteristics

  -Using the Q-bit representation to encode P systems

  -Applying the Q-gate to guide a P system toward a desired model

  -Platform: Eclipse 3.5.1, JDK 1.6.0 and P-Lingua 2.1

  -Results better than the method in [11]

- Future work

  - Generalizing this approach to a more general or complex class of membrane systems

Thanks very much for your attention!

Any questions?