

(preprint)

Organization-oriented Chemical Programming of Distributed Artifacts

Naoki Matsumaru, Thomas Hinze, Peter Dittrich (2009)

Bio Systems Analysis Group, Department of Mathematics and Computer Science,
Friedrich Schiller University Jena, 07743 Jena, Germany
peter.dittrich@uni-jena.de

Published as:

Matsumaru, N., Hinze, T., & Dittrich, P. (2009). Organization-oriented Chemical Programming of Distributed Artifacts. *International Journal of Nanotechnology and Molecular Computation (IJNMC)*, 1(4), 1-19.

DOI: 10.4018/jnmc.2009120901

Organization-Oriented Chemical Programming of Distributed Artifacts

Naoki Matsumaru, Thomas Hinze, and Peter Dittrich

Bio Systems Analysis Group, Jena Centre for Bioinformatics (JCB) and Institute of Computer Science, Friedrich Schiller University Jena, 07743 Jena, Germany,

<http://www.minet.uni-jena.de/csb>,
{naoki,hinze,dittrich}@minet.uni-jena.de

Abstract

The construction of molecular-scale machines requires novel paradigms for their programming. Here, we assume a scenario of distributed devices that process information by chemical reactions and that communicate by exchanging molecules. Programming such a distributed system requires to specify reaction rules as well as exchange rules. Here, we present an approach that helps to guide the manual construction of distributed chemical programs. We show, how chemical organization theory can assist a programmer in predicting the behavior the program. The basic idea is that a computation should be understood as a movement between chemical organizations, which are closed and self-maintaining sets of molecular species. When sticking to that design principle, fine-tuning of kinetic laws becomes less important. We demonstrate the approach by a novel chemical program that solves the maximal independent set problem on a distributed system without any central control – a typical situation in ad-hoc networks. We show that the computational result, which emerges from many local reaction events, can be explained in terms of chemical organizations, which assures robustness and low sensitivity to the choice of kinetic parameters.

Nanotechnology and molecular computation are a great match since those share the same scale medium: nanoscale molecules. Under the achievements of nanotechnology, lots of examples including logic gates using multiple nanotube transistors (Bachtold, Hadley, Nakanishi, & Dekkerdagger, 2001) have been reported. Wide varieties of nanoparticle applications (Salata, 2004), for example, ultrasensitive biosensors (Wang, 2005) using gold nanoparticles coupled with enzymes (Willner, Basnar, & Willner, 2007), attribute to nanotechnological techniques of manipulating nano-scale objects. Synthesizing molecular machinery out of DNA molecules seems promising (Bath & Turber-

The authors would like to thank our colleagues of our group for their support and useful advice. We are very grateful to Gerd Grünert and Christoph Kaleta for their help with the software development and Elisabeth zu Erbach-Schoenberg for reviewing nanoparticle research activities. We acknowledge financial supports by the German Research Foundation (DFG) Grant DI 852/4 and by the European Union, NEST-project ESIGNET no. 12789.

field, 2007) even though the lack of stiffness of biomolecules in comparison with 'dry' nanotechnology materials has been argued (Merkle, 2000) to be the drawback. Despite the rapid development of nanotechnology, wet-lab experiments are exclusively exercised to operate the boolean logics (de Silva & Uchiyama, 2007).

DNA computing demonstrated by Adleman already addressed that limitation of imperative, logic computation paradigms by utilizing particular operation modes of DNA molecules. Assuming DNA as data carrier, up to 10^{21} bytes can be saved and operated simultaneously within one liter of liquid providing a storage density of $1\text{bit}/\text{nm}^3$ (Păun, Rozenberg, & Salomaa, 1998). One Joule allows up to 10^{19} molecular operations on DNA (Pisanti, 1998). This highly parallelized operation on DNA strands with high data density is the key characteristics of the DNA computation approach. The significance, we note here, is that the computation model exploited is in concert with computation medium.

In nano-scale world, molecules are regarded to constitute medium, and chemical reactions play an important role in biological information processing principles (). Employing molecules and reaction rules as a metaphor, thus, novel computation paradigms have been explored (Banâtre & Métayer, 1986; Păun, 2002; Banâtre, Fradet, & Radenac, 2004; Tschudin, 2003; Berry & Boudol, 1992). Essentially, those chemical computing models refer the elementary units as molecules, and the operations are described in the form of reactions among those molecules. Given the inputs of the computation as the initial configuration of reaction vessels or reactors, the outputs emerge from local interactions in accordance with the reaction rules given (Banzhaf, Dittrich, & Rauhe, 1996). In these chemical computing models, programming corresponds to designing the reaction rules at the microscopic levels, and the desired computational result emerges at the macroscopic levels as a global systems' state. The relation between those two levels is highly non-linear, and thus the question for effective programming techniques arises. It seems scarcely possible in this context to predict the macro behavior from the micro rules because of the parallel operations of the reaction rules that are possibly tangled in a complex manner. A common approach to this difficulty is to find a mapping from a known computation model like a Turing machine or a finite state automaton (Păun, 2002; Rothmund, 1996).

Our programming approach, on the other hand, does not refer to those computation models. As argued by Conrad, the conventional computers differ from natural molecular systems, such as brain or enzymes, with respect to the level of achievement of programmability. The conventional digital computers are designed to achieve high programmability by restricting the behaviors of computational entities, and the natural molecular systems operate to exploit the useful properties of the medium. As a result of respecting the medium, the natural molecular systems are not highly programmable, but are evolvable (or adaptable) and computationally efficient (?, ?). In this paper, we present a programming technique utilizing a notion of chemical organization (Dittrich & Speroni di Fenizio, 2007) as a guide for predicting the behavior of the chemical program. Our case story is the maximal independent set (MIS) problem particularly in a distributed computing environment, since there are efficient algorithms to solve the MIS problem (Luby, 1986). We first start with introducing the theory of chemical organizations in the next section. Then, our programming techniques are summarized under the name of organization-oriented design principles and demonstrated on chemical programming for the MIS problem. Results of analyzing the constructed chemical programs using a chemical organization theory is shown. Finally, dynamical simulation is employed to validate the chemical programs.

Approaches to Chemical Programming

In general, there are several approaches to obtain a chemical program capable of solving a predefined computational problem. Here, we distinguish optimization versus construction. Optimization subsumes heuristically driven techniques: A more or less randomly chosen reaction network becomes successively improved e.g. by evolutionary methods (Ziegler & Banzhaf, 2001), learning inspired by neural networks, simulated annealing, or tabu search (L.F. Landweber, 2002). Within the optimization process, the reaction network topology as well as reaction parameters are fitted according to the desired behavior (Deckard & Sauro, 2004a). Although pure optimization often generates solutions of astonishing efficiency, the absolute correctness of chemical programs is not guaranteed. In contrast, the roots of manual construction of reaction systems lie in engineering (Alon, 2006). This approach is based on some elementary computational units represented by predefined well-understood reaction network motifs (Aoki, Kameyama, & Higuchi, 1992). Equipped with specified interfaces, these motifs then can be combined towards interconnected networks of more complex functionality. Using construction principles like hierarchical modularization, malfunction of the final reaction network can be avoided (Papin, Reed, & Palsson, 2004).

Chemical Organization Theory

Inspired by Fontana and Buss, a theory of chemical organizations has been developed by ?. A chemical organization is defined as a set of molecular species that is closed and self-maintaining. These properties are only dependent on stoichiometry of reaction network, and a relation of the organizations to dynamical behaviors of reaction systems based on the given reaction network was proven. Given a fixed point, the set of species with positive concentrations is an organization (?, ?). Thus, chemical organization theory allows to predict which species and reactions can persist in a long-term simulation of the reaction system (see (Kaleta, Centler, Fenizio, & Dittrich, 2008) for a practical application). This prediction does not only encompass steady-states where the concentration of no molecular species changes, but also periodic attractors and long-term behaviors with an unbounded increase of some concentrations (Peter, 2008). Since that theory is dependent on an algebraic analysis of the reaction network, only dynamical behaviors induced by the reaction network topology are concentrated, independent of reactor dynamics and reaction kinetics. It is a natural confinement that the theory operates on a relatively high abstraction level, sets of molecular species, and neglects, so far, quantitative aspects such as concentration level. The definition of chemical organizations described in this section is adopted from (Dittrich & Speroni di Fenizio, 2007).

Formally, a reaction network is a tuple $\langle \mathcal{M}, \mathcal{R} \rangle$ where \mathcal{M} is a set of molecular species and \mathcal{R} is a set of reaction rules among those species. A reaction rule $\rho \in \mathcal{R}$ is defined by the stoichiometric coefficients $l_{i,\rho} \geq 0$ and $r_{i,\rho} \geq 0$ for $i \in \mathcal{M}$, corresponding to the left-hand side and right-hand side, respectively. Two mappings are defined, given a reaction rule $\rho \in \mathcal{R}$, $\text{LHS}(\rho) := \{i \in \mathcal{M} | l_{i,\rho} > 0\}$ and $\text{RHS}(\rho) := \{i \in \mathcal{M} | r_{i,\rho} > 0\}$, representing the set of reactant species and product species, respectively.

At this point, the first property *closure* can be defined. A set $A \subseteq \mathcal{M}$ is closed if, for all reaction rules that can happen in A , their products are also contained in A ; $\forall \rho \in \mathcal{R}_A$ where $\mathcal{R}_A = \{\rho \in \mathcal{R} | \text{LHS}(\rho) \subseteq A\}$, $\text{RHS}(\rho) \subseteq A$. The closure condition assures that reactions among molecules in a closed set cannot produce molecular species outside this set.

A definition of the second property *self-maintenance* involves production rates of species described as the multiplication of stoichiometric matrix \mathbf{S} and flux vector \mathbf{v} . The dynamics of a reaction system is often specified in terms of a differential equation of the form $\dot{\mathbf{x}}_t = \mathbf{S}\mathbf{v}$ where $\mathbf{S} = (r_{i,\rho} - l_{i,\rho})$ is the stoichiometric matrix and $\mathbf{v} = (v_{\rho \in \mathcal{R}})$ is a flux vector (kinetic laws) depending on the current concentration vector $\mathbf{x}_t \in \mathbb{R}_{\geq 0}^{|\mathcal{M}|}$. Without loss of generality, we assume $v_{\rho} \geq 0$ so that a reversible reaction needs two entries in \mathbf{v} .

Although the kinetic laws can be an arbitrary functions, they are constrained by topology of the reaction network: Obviously, the flux v_{ρ} of reaction ρ can only be positive, if all reactant species ($\text{LHS}(\rho)$) are present. It also makes sense to assume the reverse. Then we obtain the *chemical ODE constraint*:

$$v_{\rho} > 0 \quad \Leftrightarrow \quad \text{for all } i \in \text{LHS}(\rho), x_i > 0. \quad (1)$$

There is a couple of theoretical approaches that use this constraint, already formulated by (Feinberg & Horn, 1973), to relate the algebraic structure of the underlying reaction network to the dynamical behavior of the reaction system (e.g., (Gatermann, Eiswirtha, & Sensse, 2005)). Under this constraint, the self-maintenance property is defined. A set $A \in \mathcal{M}$ is self-maintaining if there exists a strictly positive flux vector $\mathbf{v}' \in \mathbb{R}_{> 0}^{|\mathcal{R}_A|}$ such that all species in A are produced at a non-negative rate. That is, $\mathbf{S}_A \mathbf{v}' \geq 0$ (Dittrich & Speroni di Fenizio, 2007) where \mathbf{S}_A is a part of the stoichiometric matrix regarding the sub-network $\mathcal{R}_A \subseteq \mathcal{R}$. The self-maintenance property assures that all species consumed by reactions in A can be reproduced by the some reaction pathways in the whole network of A .

Using that notion of chemical organizations, the given reaction network is explored which species combination is a chemical organization and is decomposed into hierarchical, overlapping sub-networks, organizations. That hierarchy is an overview of persisting set of species in the reaction systems after molecules are processed according to the reactions specified. When employing the reaction processes for computation and choosing a proper coding scheme, the hierarchical organizational structure provides us with an overview of computation outputs. This view has motivated organization-oriented chemical programming techniques (Dittrich & Matsumaru, 2007), where the theory of chemical organizations is utilized as a tool for programming chemical reaction systems.

Organization-oriented Chemical Programming

Organization-oriented chemical programming was described naively in (Dittrich & Matsumaru, 2007). Here, we elaborate those principles in more detail and through an example of chemical programs for the maximal independent set (MIS) problem.

A Chemical Program for the MIS Problem

The MIS problem is, given an undirected graph, to find a MIS as illustrated in Figure 1. Let $G = \langle V, E \rangle$ be an undirected graph where $V = \{v_1, \dots, v_N\}$ is a set of N vertices and $E \subseteq V \times V$ is a set of edges. An edge is represented by a pair $(v_p, v_q) \in E$ of vertices that are connected. Note that the order of the pair is insignificant, that is, $(v_p, v_q) = (v_q, v_p)$. A set of vertices $I \subseteq V$ is *independent* if no two vertices in the set are directly connected by an edge: $\forall v_p, v_q (p \neq q, v_p \in I, v_q \in I, (v_p, v_q) \notin E)$. An independent set is *maximal* if there is no larger independent set containing it. No vertex can be added to a maximal independent set without violating its independence property. The MIS problem can be efficiently solved while finding the largest MIS (denoted as *maximum* independent set problem) is NP complete.

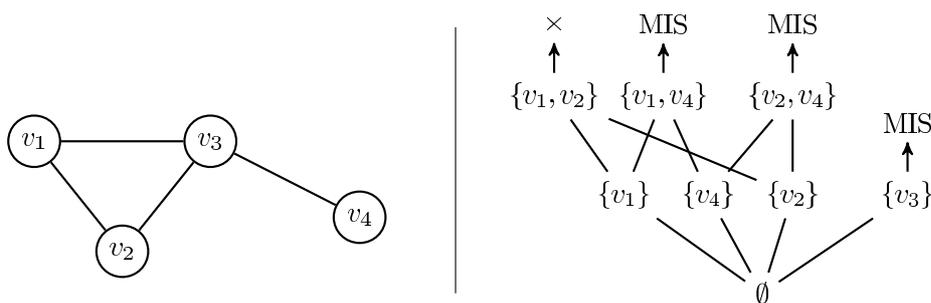


Figure 1. Illustration of the maximal independent set (MIS) problem. (Left) The given undirected graph $G = \langle V, E \rangle$ consists of four vertices. (Right) Schematic representation of solving the MIS problem by adding vertices to the empty set. Adding vertex v_1 , v_2 or v_4 satisfies the property of independence in the set, but these sets are not maximal, because it is possible to add a vertex without violating the independence property. Adding vertex v_3 , on the other hand, results in a solution of the MIS problem. Another solution to the MIS problem is $\{v_1, v_4\}$ and $\{v_2, v_4\}$. The set $\{v_1, v_2\}$ is not an independent set. The solutions to the *maximum* independent set problem is $\{v_1, v_4\}$ and $\{v_2, v_4\}$, because of the largest size 2 while the other MIS $\{v_3\}$ has the size of 1.

To solve the MIS problem in a distributed computing environment without central control, the following two predicates applied locally in every vertex v_i are suggested: (Herman, 2003; Shukla, Rosenkrantz, & Ravi, 1995; Ikeda, Kamei, & Kakugawa, 2002): (I) If a neighboring vertex v_j ($(v_j, v_i) \in E$) of v_i is included in the set I , then the vertex v_i should be excluded from I . (II) If no neighboring vertex v_j of v_i is included in I , the vertex v_i should be included in I . Formally:

$$\begin{aligned} \text{(I)} \quad & \exists v_j [(v_j, v_i) \in E, v_j \in I] \quad \Rightarrow \quad v_i \notin I, \\ \text{(II)} \quad & \forall v_j [(v_j, v_i) \in E, v_j \notin I] \quad \Rightarrow \quad v_i \in I \end{aligned} \quad (2)$$

Our chemical program is derived from these rules.

We developed a chemical program for the MIS problem on a distributed computing environment. Each distributed device corresponds to a vertex of the given graph. Each device consists of a reactor populated by molecules, copies or instances of species, and a chemical program is a chemical reaction network defining the reaction rules among these molecular species. The reaction rules are common in every device, and the devices are internally differ in terms of the molecules' concentrations. Communication between the devices is achieved through the exchange of molecules by undirected diffusion or directed transport (Abelson et al., 2000; Siehs & Mayer, 1999; Hiyama et al., 2005). The exchange rules are, from a formal point of view, indistinguishable with reaction rules when spatial localizations, to which device the molecules belong, are taken into account.

A chemical program is a reaction network $\langle \mathcal{M}, \mathcal{R} \cup \mathcal{T} \rangle$ where \mathcal{M} is a set of molecular species and \mathcal{R} is a set of reaction rules among those species. For the MIS problem, there are four species $\mathcal{M} = \{s^1, s^0, f^0, f^1\}$ and three reaction rules in \mathcal{R} and four exchange rules in \mathcal{T} as depicted in Figure 3. There are two kinds of reactions: reaction rules \mathcal{R} within one vertex and exchange rules \mathcal{T} between two vertices For any vertices, the reaction rules are common:

$$\mathcal{R} = \{s^0 + s^1 \rightarrow \emptyset, f^0 \rightarrow \emptyset, f^1 \rightarrow s^0\}. \quad (3)$$

Exchange rules $\mathcal{T}_{i,j}$ are defined for each edge $(v_i, v_j) \in E$:

$$\mathcal{T}_{i,j} = \{s_i^1 \rightarrow s_i^1 + f_j^1, s_i^0 \rightarrow f_j^0, \quad s_j^1 \rightarrow s_j^1 + f_i^1, s_j^0 \rightarrow f_i^0\}. \quad (4)$$

Note that the vertex number is attached to the species names in order to distinguish identical species based on locations. These rules transport the information whether a vertex is included or not in the MIS and establish communication between two devices. For example, with $s_i^1 \rightarrow s_i^1 + f_j^1$, vertex v_i informs a neighbor vertex v_j that v_i is included in the MIS. For bidirectional communication, there are also flows from vertex v_j to v_i . Rationals of these reactions are explained in the latter section.

In a former work (Matsumaru, Lenser, Hinze, & Dittrich, 2007), we developed another chemical program for the same problem. The difference is that the former program involves an irrational reaction such that the number of reactants depends strictly on the number of neighbors. If a vertex has five neighbors, a fifth order reaction must be defined. Moreover, that reaction itself must be modified whenever the neighbor list is changed, and it becomes critical in a dynamically changing environment. These are overcome in this work. The order of the chemical reactions are restricted to at most two regardless of the graph topology. The neighboring vertices are presumed to be indistinguishable such that all neighbors are categorized as a foreign vertex. Even in the wireless network, maintaining neighbor list is feasible with the help of neighbor discovery algorithms, but the improved chemical program does not require the list, saving energy and memory resources.

Taking these principles into consideration, we construct a chemical program for a MIS problem instance, given a two-vertex graph: $G = \langle \{v_i, v_j\}, \{(v_i, v_j)\} \rangle$. The solutions to this MIS problem is $\{v_i\}$ and $\{v_j\}$. The constructed reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ consists of eight species: $\mathcal{M} = \{s_i^1, s_i^0, f_i^1, f_i^0, s_j^1, s_j^0, f_j^1, f_j^0\}$. Species name s stands for *self* and f for *foreign*. The superscript is a binary number, indicating the membership of vertices in the MIS, and the subscript is the vertex number where the species belong. If there would be no reaction rules, we would have 256 ($= 2^8$) organizations. The resulting program depicted in Figure 3 contains only three organizations as shown in Figure 2: $\{\emptyset\}$, $\{s_i^0, f_i^1, s_j^1, f_j^0\}$, $\{s_i^1, f_i^0, s_j^0, f_j^1\}$. In the following, we sketch the principles we followed in order to achieve this final structure.

Organization-oriented Design Principles

Here seven design principles of organization-oriented programming are listed. The first principle (P1) describes constraint on mainly coding schemes to insure the applicability of chemical organization theory. Considering principles P2-P6, reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ is designed. The basis of these principles is to arrange reaction networks to conform the embedded organizational structures with the desired ones. Then, the kinetics including kinetic parameters is specified for fine-tuning the computation as stated in P7.

P1: There should be one organization for each output behavior class Dynamical behaviors of reaction systems are time series of concentration profiles $\{\vec{x}(t) \in X | t_0 < t < t_1\}$ where X is the systems' state space. By some features, the dynamical behaviors are categorized in classes, and those behavior classes are interpreted as outputs of computation. This first programming principle states that this output behavior classification should be arranged such that there should be one organization for each of that class. In other words, if there are two distinguishable behavior classes, then the corresponding organizations should be different from each other.

In our MIS example, the features are whether species s^1 or s^0 is present in the reactor at a certain time point. The time point is determined such that the reaction systems will reach to a steady state. With this classification arrangement, we have four behavior classes for each vertex depending on species present: neither (c_\emptyset), s^1 (c_{s^1}), s^0 (c_{s^0}), and both (c_s). Out of these, two (c_{s^1} , c_{s^0}) are interpreted as an output behavior class: either s^1 or s^0 is exclusively present. An output behavior

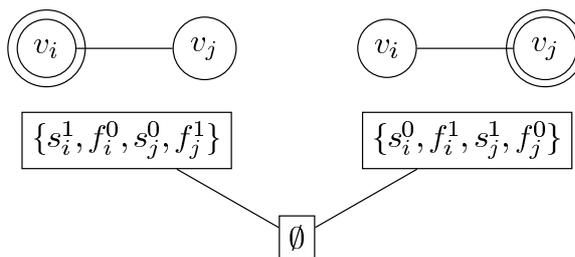


Figure 2. Organizational structure within the reaction network for the MIS problem. Assuming two vertices connected and 10 reactions defined as shown in Figure 3, there are three organizations within that reaction network. Except for the empty set, two organizations corresponds to two solutions to the MIS problem. Vertices with double circles are in the MIS.

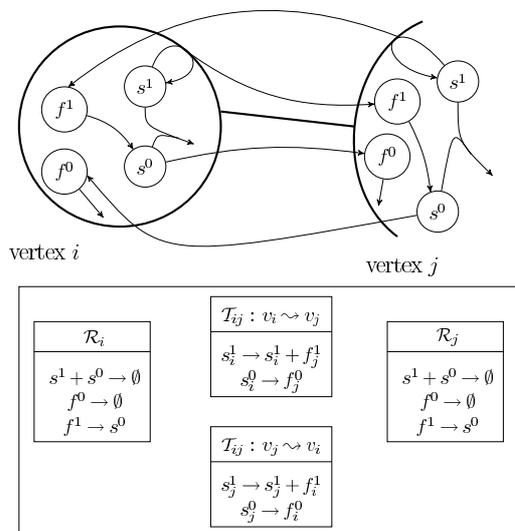


Figure 3. Chemical program for the MIS problem. Bottom: Reaction \mathcal{R} and exchange \mathcal{T} rules of a chemical program for the MIS problem. Top: Illustration of the reaction network for two vertices. See text for details.

class when only s^1 species is present, for example, is mapped to the computational output such that the vertex belongs to the MIS. Class c_{s^0} is interpreted such that the vertex is excluded from the MIS. The other two behavior classes (c_0, c_s) indicate uncompleted computations.

This behavior classification and coding scheme are a simple way to fulfill the first principle. The output behavior classes c_{s^1} and c_{s^0} are distinguishable at the abstraction level where chemical organization theory operate. Two kinds of organizations at least are necessary within the reaction network to be constructed: organization with only either species s^1 or s^0 . This programming principle claims that the constructed networks should keep these two sorts of organizations.

Organizations contain both s^1 and s^0 are associated with a class of uncompleted computation, c_s . To avoid the organizations, reactions $s^1 + s^0 \rightarrow \emptyset$ for both vertices are added. With these two reactions, sets containing both s^1 and s^0 simultaneously are no longer self-maintaining. For example, set $A = \{s_i^1, s_i^1\}$ is not self-maintaining. Production rate of s_i^1 is $\dot{x}_{s_i^1} = -v_{(s_i^1 + s_i^0 \rightarrow \emptyset)}$, and flux $v_{(s_i^1 + s_i^0 \rightarrow \emptyset)} > 0$ due to the chemical ODE constraint. Species s_i^1 cannot be produced at a non-negative

rate, and thus the set $\{s_i^1, s_i^0\}$ is not self-maintaining. Hundred twelve sets are not an organization: 16 ($= 2^4$) sets containing only s_i^1, s_i^0 ; 16 sets with s_i^1, s_i^0 , and s_j^1 ; 16 sets with s_i^1, s_i^0 , and s_j^0 ; 16 sets with s_j^1 and s_j^0 ; 16 sets with s_j^1, s_j^0 , and s_i^1 ; 16 sets with s_j^1, s_j^0 , and s_i^0 ; 16 sets with $s_i^1, s_i^0, s_j^1, s_j^0$.

In passing, there are approaches of chemical computation exploiting quantitative values of species concentration. Examples are enzymatic computation (Zauner & Conrad, 2001), where the concentration level (high or low) of reaction products catalyzed are chosen for features to classify output behaviors. Another example is a chemical reaction system evolved by Deckard and Sauro to compute the square-root. The final concentration x_T of a molecular species is the square root of the initial concentration x_0 , $x_T = \sqrt{x_0}$. These coding schemes to map dynamical behaviors to computational output do not cooperate with this programming technique because of the violation of this principles. Quantitative aspects are indistinguishable from the organizational point of view. On the other hands, the classical DNA computing (Adleman, 1994) or a prime number artificial chemistry (Banzhaf et al., 1996) may be combined with chemical organization theory since the species present characterize computational outputs.

P2: The set of molecular species (and the organization) representing a result should be in the closure of the species representing the initial input The closure denotes a set of molecular species that is generated by adding all possible reaction products until no more new species can be produced. This principle assures that there is a reaction path from the initial input configuration to the desired output species. Otherwise, the desired output will not appear as a result of the computation.

Furthermore, it is expected that the desired output set is contained in a self-maintaining set within that closure. The self-maintenance property of the set of molecular species indicates theoretical possibilities to sustain all the species in the dynamical reaction systems, so the desired output species may be sustained in the reaction system until the outcomes of the computation is observed. The ideal case is that the desired output is represented by a largest self-maintaining set within that closure. In case that there exists a larger self-maintaining set than the desired output set, the dynamics may settle above the desired one. This argument leads to the next principle.

P3: The set of molecular species representing an input should generate the organization representing the desired output To generate the organization from a set of species, by definition, the closure of the given set is taken at first. Then we remove species until we reach a largest self-maintaining set contained in the closure. This principle will be fulfilled on the following two conditions: the desired output is contained within the closure of the input (P2 is fulfilled), and the largest self-maintaining set contained in the closure corresponds to the desired output.

The largest self-maintaining set within a closure is not always unique in general although it is uniquely generated in a specific class of reaction networks, called semi-consistent (Dittrich & Speroni di Fenizio, 2007) In chemical computing, the uniqueness is not required. It can be even beneficial, on the contrary.

In our example, the initial input configuration for each vertex is represented by the self species. Following the principle 2, we added four reactions $s_i^1 \rightarrow s_i^1 + f_j^1$, $f_j^1 \rightarrow s_j^0$, $s_i^1 \rightarrow s_j^1 + f_i^1$, and $f_i^1 \rightarrow s_i^0$. in order to create a reaction path from input configuration to desired output. This pathway is a reaction form of the first predicate (I) in Equation (2). There are now twenty four organizations: sixteen organizations of any combination including empty set of four species s_i^0, f_i^0, s_j^0 , and f_j^0

because of no reactions among those, $\{s_i^1, f_j^1, s_j^0\}$, $\{s_i^1, f_j^1, s_j^0, f_i^0\}$, $\{s_i^1, f_j^1, s_j^0, f_j^0\}$, $\{s_i^1, f_j^1, s_j^0, f_i^0, f_j^0\}$, and the four rules with exchanging vertex numbers i and j in species names due to the symmetric structure of the graph. The reactions of exchanging species between vertices are catalytic in order to follow the principle 3. Otherwise any set with s^1 species including a desired output $\{s_i^0, f_i^1, s_j^1, f_j^0\}$ cannot be self-maintaining and an organization because production rate of s_i^1 would be always negative $\dot{x}_{s_i^1} = -v_{(s_i^1+s_i^0 \rightarrow \emptyset)} - v_{s_i^1 \rightarrow f_j^1}$.

P4: Eliminate organizations not representing a desired output Since each organization potentially includes fixed points, the reaction system's dynamics may converge to one of the organizations. Hence, it makes sense to eliminate organizations not representing an output in order to avoid premature termination of a computation or even false computational outputs. This can be achieved by destroying either its closure property or its self-maintenance.

For our example, the collaborative decays $s^1 + s^0 \rightarrow \emptyset$ previously described also attribute to this principle. In addition, the current reaction network still contains the organization representing an invalid output: $\{s_i^0, f_i^0, s_j^0, f_j^0\}$. Reactions $s_i^0 \rightarrow f_j^0$ and $s_j^0 \rightarrow f_i^0$ are added in order to destroy its self-maintenance property. Production rate $\dot{x}_{s_i^0} = -v_{s_i^1+s_i^0 \rightarrow \emptyset} + v_{f_j^1 \rightarrow s_i^0} - v_{s_i^0 \rightarrow f_j^0}$ can be positive only when s_i^0 is present together with f_j^1 to apply positive constraint on flux $v_{f_j^1 \rightarrow s_i^0} > 0$. Following is the list of eight organizations: $\{\emptyset\}$, $\{f_i^0\}$, $\{f_j^0\}$, $\{f_i^0, f_j^0\}$, $\{s_i^1, f_i^0, s_j^0, f_j^1\}$, $\{s_i^0, f_i^1, s_j^1, f_j^0\}$, $\{s_i^1, f_i^0, s_j^0, f_j^1, f_j^0\}$, and $\{s_i^0, f_i^1, f_i^0, s_j^1, f_j^0\}$.

P5: An output organization should have no organization below The dynamics of the reaction system that moves from one organization O_1 to another O_2 below (*i.e.*, $O_2 \subset O_1$) is called a downward movement. This dynamical move can be theoretically prevented by the self-maintenance property with the right kinetics. Practically speaking, this move may occur spontaneously due to, *e.g.*, stochastic effects because the self-maintenance property only ensures the possibilities to sustain all species. Following this principle, a downward movement can be restricted.

We added outflows of f^0 species, $f^0 \rightarrow \emptyset$ to eliminate organizations below output organizations. Namely, $\{f_i^0\}$, $\{f_j^0\}$, and $\{f_i^0, f_j^0\}$ are not self-maintaining. Additionally, these two sets are not self-maintaining: $\{s_i^1, f_i^0, s_j^0, f_j^1, f_j^0\}$, and $\{s_i^0, f_i^1, f_i^0, s_j^1, f_j^0\}$. Production rate of f_j^0 is $\dot{x}_{f_j^0} = -v_{f_j^0} + v_{s_i^0 \rightarrow f_j^0}$ so that s_i^0 must coexist to apply positive constraint on the positive term. Then, we reach to the reaction network containing only three organizations as shown in Figure 2.

P6: Assure, if possible, stoichiometrically the stability of an output organization Instead of eliminating organizations below the desired output as in the previous principle P5, the downward movement can be ruled out by purely stoichiometric argument. It may be possible to design the reaction network such that the organization representing the desired output is stable for any kinetic law. As a simple example consider the system $\mathcal{R} = \{a \rightarrow b, b \rightarrow a\}$, which has two organizations: $\{\emptyset\}$ and $\{a, b\}$. Due to mass-conservation, the system can never move spontaneously from the organization with two species to the empty one. In the MIS example, this principle is implicitly conformed.

P7: Use kinetic laws for fine tuning The kinetic laws determines the systems' behavior within an organization and the transition dynamics between organizations. One of rationals for the right

kinetics is to assure that the dynamical reaction systems are stable in the output organizations, restricting mainly the downward movement. Finding the right kinetic laws is in general a non-trivial task. However, the existence of such laws is ensured by chemical organization theory to a certain extent, and we have seen that following principles P1-P6 simplify this tasks significantly. Classical dynamical systems theory is certainly reliable for this task, and it is even possible to derive at least in some cases rigorously dynamical stability from network structure (Clarke, 1980; Feinberg & Horn, 1974). Another point of consideration is a trade-off between that stability and the speed of computation since chemical reaction systems may compute by moving amongst organizations.

Organizational Analysis

Organizational analysis is an algebraic analysis on a reaction network to explore which species combination is a chemical organization. The results are a hierarchical organizational structure of the reaction network. In this section, we analyze the constructed MIS chemistry described above with respect to the organizational structure, using the theory of chemical organizations. We further present the dynamical behaviors of the program for validation purpose.

Organizational Analysis of Distributed Systems

When analyzing reaction networks, it is significant how large spatial regions are covered because the size of regions also affect on the network structure. For instance, an enzymatic reaction, which is mediated by a certain molecular species, is only defined when that specific enzyme species is present. The catalysts may be isolated by membrane or immobilized to a specific location. If that regions is not covered, the reactions should not be considered. As argued in (Fenizio & Dittrich, 2007), the organizational analysis can also provide valuable insights into the spatial structure of the system and information about the best spatial scale to consider. An important difference in their approach is that diffusion played only a minor role. Here, on the other hand, the continuous exchange of molecular species is essential.

On a distributed system modeled as a set of nodes and links, there are two regional perspectives: local and global. Global perspective considers the whole system, and local perspective focuses on a local node neglecting the network topology.

Global Analysis of the MIS Chemistry

For a global analysis, a particular graph structure has to be chosen, so the undirected graph $G = \langle V, E \rangle$ in Figure 1 is regarded:

$$V = \{v_1, v_2, v_3, v_4\}, \quad E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_3, v_4)\}. \quad (5)$$

The chemical program $\langle \mathcal{M}_{global}, \mathcal{R}_{global} \cup \mathcal{T}_{global} \rangle$ for the MIS problem consists of 16 species

$$\mathcal{M}_{global} = \{s_i^1, s_i^0, f_i^1, f_i^0 \mid i = 1, \dots, 4\} \quad (6)$$

The species names are associated with the vertex numbers such that those species are distinguishable even in a non-spatial “well-stirred” reactor. For each vertex, there are three reactions:

$$\mathcal{R}_{global} = \bigcup_{i=1}^4 \mathcal{R}_i = \left\{ \begin{array}{cccc} s_1^0 + s_1^1 \rightarrow \emptyset, & s_2^0 + s_2^1 \rightarrow \emptyset, & s_3^0 + s_3^1 \rightarrow \emptyset, & s_4^0 + s_4^1 \rightarrow \emptyset, \\ f_1^0 \rightarrow \emptyset, & f_2^0 \rightarrow \emptyset, & f_3^0 \rightarrow \emptyset, & f_4^0 \rightarrow \emptyset, \\ f_1^1 \rightarrow s_1^0, & f_2^1 \rightarrow s_2^0, & f_3^1 \rightarrow s_3^0, & f_4^1 \rightarrow s_4^0 \end{array} \right\} \quad (7)$$

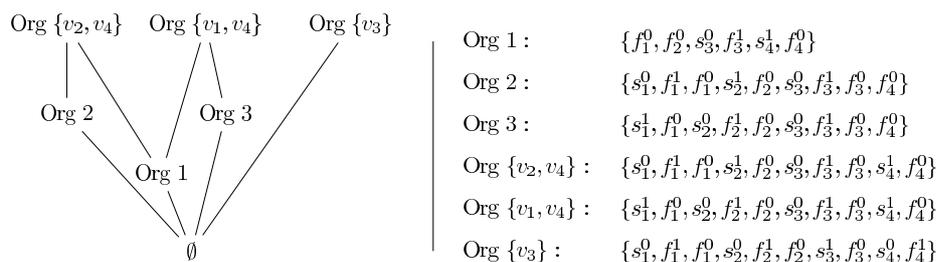


Figure 4. Global analysis of a distributed chemical program for the four-vertex topology shown in Figure 1. The Hasse-diagram (left) shows all organizations of the global reaction network $\langle \mathcal{M}_{global}, \mathcal{R}_{global} \cup \mathcal{T}_{global} \rangle$. Contents of each organization is listed on the right. The largest organizations corresponds to correct results of the computation. The four organizations below represent states of the chemical system that cannot lead to a computational result.

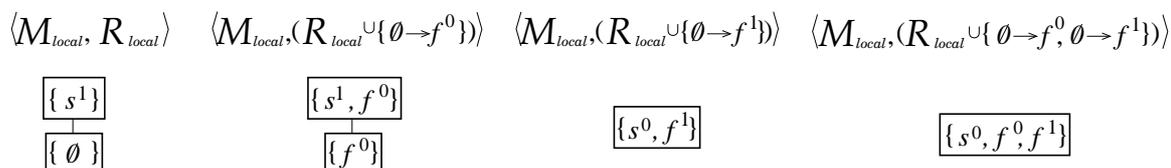


Figure 5. Analysis of chemical programs optimized for distributed systems with chemical organization theory. Four reaction networks are given at the top, and the organizational structure embedded within the network is listed below. From the left, base reaction network with no inflow, with inflow of f^0 , inflow of f^1 , and both. Components of the organizations agree to the desired behavior of the vertex.

Four exchange reactions are attached to each edge:

$$\mathcal{T}_{global} = \bigcup_{(v_i, v_j) \in E} \mathcal{T}_{i,j} = \bigcup_{(v_i, v_j) \in E} \{s_i^1 \rightarrow s_i^1 + f_j^1, s_i^0 \rightarrow f_j^0, s_j^1 \rightarrow s_j^1 + f_i^1, s_j^0 \rightarrow f_i^0\}. \quad (8)$$

The organizational structure within the reaction network $\langle \mathcal{M}_{global}, \mathcal{R}_{global} \cup \mathcal{T}_{global} \rangle$ is shown in Figure 4. The three largest organizations correspond to the three solutions of the MIS problem, and there is no organization with an incorrect solution. Implications of this analysis are chemical reaction systems based on that MIS chemistry behaves in accordance with the solutions to the MIS problem, and the correctness is theoretically guaranteed. The analysis also indicates possibilities of the uncompleted computations with Org 1, Org 2, Org 3, and the empty set. When the system is “caught” in one of these organizations, the outcome of the computation is inconclusive. This happens when there are not enough molecules, especially s^1 , in the system. This situation was circumvented in the ODE system by introducing inflow of s^1 species.

Local Analysis of the MIS Chemistry

To take a local perspective, a vertex is focused and the other parts are identified as environments. No vertex number is necessary for the species name. The exchange rules are modeled as an influx and an outflux because the neighboring vertices are out of scope of the local perspective. The chemical program $\langle \mathcal{M}_{local}, \mathcal{R}_{local} \cup \mathcal{T}_{local} \rangle$ to analyze is:

$$\mathcal{M}_{local} = \{s^1, s^0, f^1, f^0\}, \quad \mathcal{R}_{local} = \{s^1 + s^0 \rightarrow \emptyset, f^0 \rightarrow \emptyset, f^1 \rightarrow s^0, \}, \quad \mathcal{T}_{local} = \{s^0 \rightarrow \emptyset\}. \quad (9)$$

Note that there is no outflow of s^1 because that species is not consumed by an exchange rule. Environmental conditions are modeled by inflows.

In the MIS problem, there are four environmental cases as shown in Figure 5: no neighbors (without any inflows), no neighbors are included in the MIS (append influx of f^0), all neighbors are included in the MIS (append influx of f^1), and some neighbors are included and some are not (append both influxes of f^0 and f^1). In all cases, incorrect organizations do not exist, and thus, there cannot be a wrong result of computation. If the vertex has decided whether it belongs to the MIS, we can be sure that it is a correct solution. This result is consistent with the global analysis.

Undesired organizations are $\{\emptyset\}$ in the leftmost case and $\{f^0\}$ in the case with f^0 influx. As we will see later, these organizations cannot be avoided in principle, so that we have either to start with a sufficiently large set of molecules of s^1 or add a mechanism that can move the system from the organizations up to the “solution” organizations above. With the former method, the system cannot manage a dynamically changing environment where a vertex may be desired to alter the membership state of the MIS from negative (s^0) to positive (s^1) because s^1 species will be completely consumed in the case with f^1 influx. Therefore, we adopted the latter. The appended inflow of s^1 is designed to be inhibited by s^0 species. Unless, no organization without s^1 becomes possible. In the case with f^1 influx, for example, the only organization becomes $\{s^1, s^0, f^1\}$.

Dynamical Simulation

In order to validate the chemical program presented, we constructed an ordinary differential equation (ODE) system based on that reaction network and analyzed dynamical behaviors. For that purpose, we chose as a particular problem instance the graph $G = \langle V, E \rangle$ depicted in Figure 1 formulated in Equation (5). The ODE system consists of 16 equations because of four equations for each vertex, and there are four vertices in the problem instance. Mass action kinetics, for each reaction, is assumed.

Concentration dynamics of species s^1 and s^0 with respect to v_i are expressed as follows:

$$[s_i^1] = \frac{d[s_i^1]}{dt} = -k_1[s_i^1][s_i^0] \quad (10)$$

$$[s_i^0] = \frac{d[s_i^0]}{dt} = k_2[f_i^1] - k_1[s_i^1][s_i^0] - \left(\sum_{(v_i, v_j) \in E} k_3[s_i^0] \right) \quad (11)$$

where k_1 and k_2 are kinetic constants for reactions $s^1 + s^0 \xrightarrow{k_1} \emptyset$ and $f^1 \xrightarrow{k_2} s^0$, respectively, and set to 0.1. Exchange rule $s_i^0 \xrightarrow{k_3} f_j^0$ with a kinetic constant $k_3 = 0.1$ is considered for every neighboring vertex v_j with $(v_i, v_j) \in E$. Because of the catalytic nature of the exchange rule for s^1 : $s_i^1 \xrightarrow{k_4} s_i^1 + f_j^1$, species s_i^1 is not flowed out to the neighbor vertices. This exchange rule, associated with kinetic constant $k_4 = 0.1$, does not consume s^1 but does produce f^1 . Concentration dynamics of species f^1 and f^0 with respect to vertex v_i are given:

$$[f_i^1] = \frac{d[f_i^1]}{dt} = -k_2[f_i^1] + \left(\sum_{(v_i, v_j) \in E} k_4[s_j^1] \right) \quad (12)$$

$$[f_i^0] = \frac{d[f_i^0]}{dt} = -k_5[f_i^0] + \left(\sum_{(v_i, v_j) \in E} k_3[s_j^0] \right). \quad (13)$$

For both species, the equations are composed of one term for an outflow and multiple terms for inflows from neighboring vertices. Kinetic constant k_5 for outflow $f^0 \xrightarrow{k_5} \emptyset$ is set to 0.1.

The equation system listed in Equations (10) – (13) is the reaction system based on our chemical program to solve the MIS problem. However, we introduced a modification on the dynamics of s^1 concentration:

$$[s_i^1] = -k_1[s_i^1][s_i^0] - k_6[s_i^1] + \frac{0.1}{[s_i^0] + 10}. \quad (14)$$

Species s^1 is incorporated with an inflow that is inhibited by species s^0 , expressed by the last term of Equation (14). The production rate of s^1 is low when s^0 exists in a high concentration. Looking at the reaction network, there is no production rule of species s^1 . This becomes problematic when a vertex is desired to alter the membership state from negative (s^0) to positive (s^1). In order to cope with that condition, an inflow of s^1 is added to compensate disappearance of s^1 species. The inhibitory effect to the inflow from s^0 is necessary because that inflow should be activated only when species s^0 is vanished. Otherwise, there will be no steady state composed of only s^0 and f^1 . Outflow term $-k_6[s^1]$ ($k_6 = 0.001$) is also added in order to avoid explosive increase of s^1 concentration ($s^1 \xrightarrow{k_6} \emptyset$).

Figure 6 shows a dynamical behavior of the chemical program for a MIS problem instance, implemented as an ODE system listed in Equations (11) – (14). The reaction system is stochastically simulated using *Copasi* (Sahle et al., 2006) and compartmentalized (Amos, 2004) in order to emulate distributed system settings, in which each compartment equals a vertex. The compartment size is set to 20 ml. Initially, we disconnected every vertex so that no exchange of species between vertices is possible, and no molecules are present in the reactors. Then, each compartment comes to be filled with ≈ 200 s^1 molecules (10 # / ml) due to the appended inflow of s^1 . This state is interpreted such that the MIS consists of every vertex, and the set with every vertex included is certainly the MIS. At $t = 5000$, we connected the four vertices and continued the simulation. Due to the exchange rules, species f^1 is generated in each compartment. Instantaneously, species s^0 is produced by the reaction $f^1 \rightarrow s^0$, and s^1 is consumed by the reaction $s^1 + s^0 \rightarrow \emptyset$. The generated s^0 species are also exchanged to the neighboring vertices as f^0 . After ≈ 500 steps, the reaction systems already came to a steady state. In this run, compartments for v_2 and v_3 are populated with s^0 so that those vertices are excluded from the MIS. Vertex v_1 and v_4 , populated with s^1 , remain in the MIS. For this graph structure, there are three possible MISs. MIS $\{v_3\}$ rarely appears because of the fast generation of s^0 , resulted by the introduction of f^1 from the three neighboring vertices.

We next study the behaviors of the chemical program on dynamically changing topology, and a result is illustrated in Figure 7. The concentration graph at the bottom only shows that of s^1 for each vertex. We checked that s^0 species concentration became high when s^1 concentration became low, and vice versa. Before the simulation, the reaction system settled in the state where the reactors in vertex v_2 and v_4 are filled with s^1 . In the time range of [30000 : 35000], every vertex is disconnected, and all compartments are populated with s^1 species. Then, at $t = 35000$, v_1 and v_3 are connected so that the communication between those vertices becomes established. A rapid drop of s^1 species concentration in v_1 is observed so that vertex v_1 is now excluded to maintain the MIS property. At $t = 40000$, vertex v_2 is further connected, and s^1 in v_2 is consumed.

When the edge (v_3, v_4) is further enabled at $t = 45000$, set $\{v_2, v_4\}$, excluding v_3 and including v_2 , is computed to be the MIS. Disabling and then enabling the edges to v_2 , the outcomes of the computation is changed from $\{v_2, v_4\}$ to $\{v_1, v_4\}$. During this simulation run, the property of the MIS is constantly maintained.

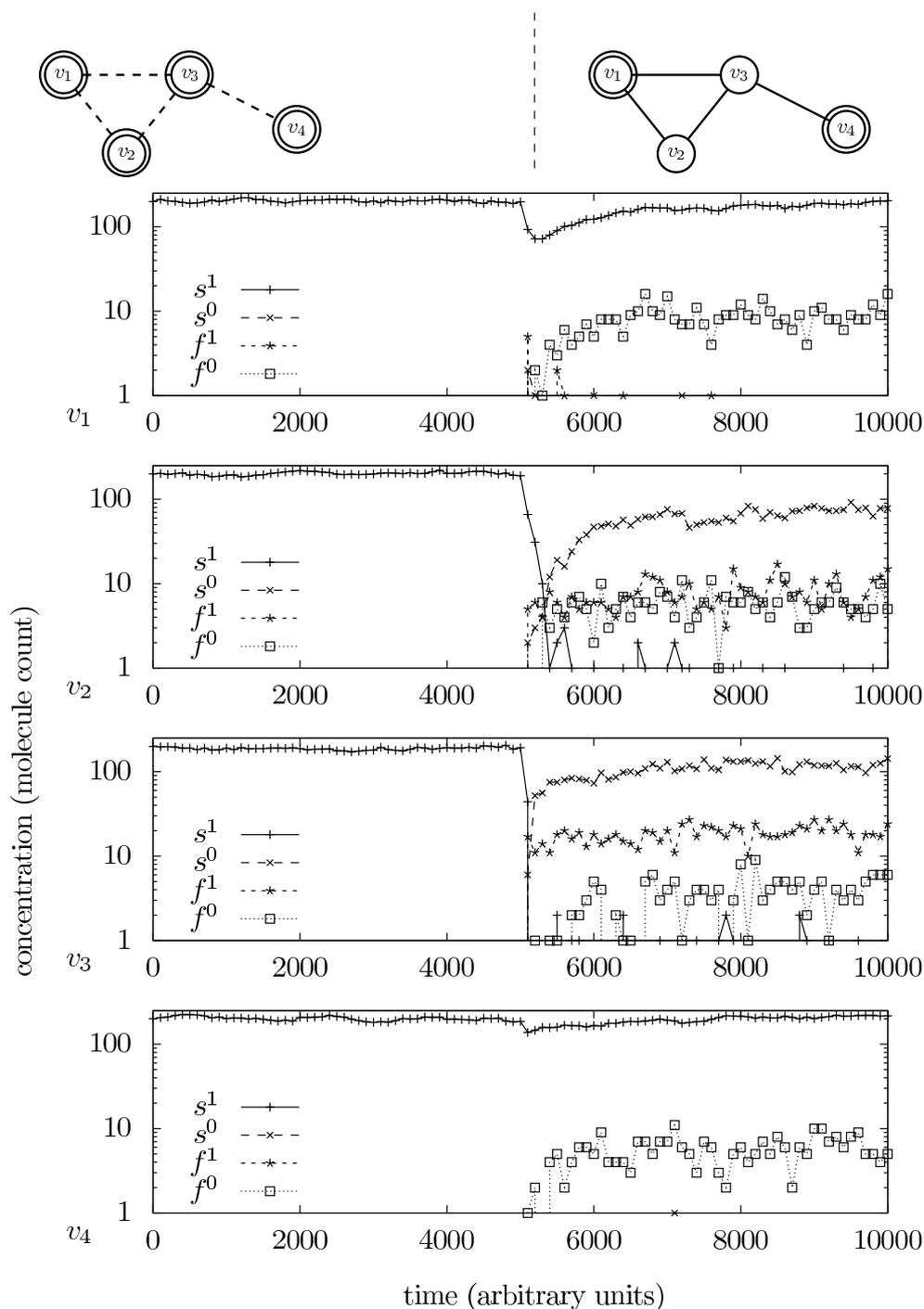


Figure 6. Dynamical behavior of a chemical program for the MIS problem. The graph structure is shown at the top, and dashed lines represent no connection. Double circled vertices are computed to be in the MIS with the chemical reaction systems. When every vertex is disconnected at the initial time point, only s^1 species has a positive concentration for every vertex. At $t = 5000$, every vertex is connected. In vertex v_1 and v_4 , s^1 continue to have a positive high concentration. Vertex v_2 and v_3 is populated with s^0 species, and s^1 species is vanished. The output of the chemical computation is $\{v_1, v_4\}$, which is a MIS. Species f^1 and f^0 are present accordingly. In v_1 , for instance, only f^0 species is with a positive concentration because that vertex is linked only with vertices excluded from the set. Vertex v_2 contains both f^1 and f^0 species because of the connections with v_1 and v_3 transporting f^1 and f^0 , respectively.

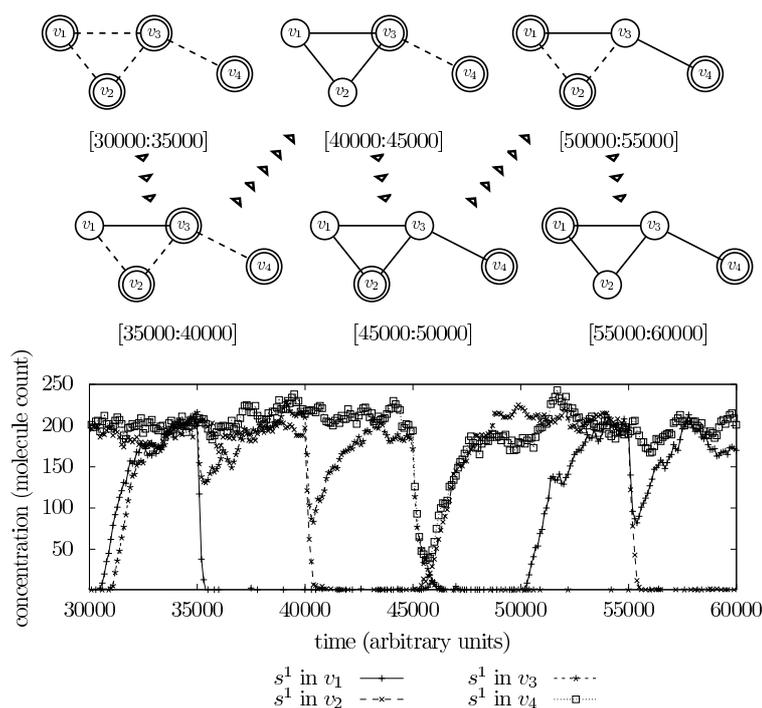


Figure 7. Dynamical behaviors of the graph structure and those of a chemical program for the MIS problem. The dashed lines represent no connection, and double circled vertices indicate the inclusion in the MIS. On dynamically changing topology, the chemical reaction systems compute so that the property of the MIS is always maintained. See text for details.

Conclusion and Outlook

We have presented a theoretical base for the design of distributed chemical computing systems, namely organization-oriented chemical programming. Our techniques were exemplified on the maximal independent set problem, which could be regarded as a prototype of differentiation and morphogenesis. Applying our approach has led to a solution chemistry of that problem, and robustness against dynamical changes of the base graph topology has been also demonstrated using simulations. Considering implementation beyond *in silico*, simplicity of our solution is also valuable. There are only four molecular species necessary, and three reaction rules among those and two exchange rules between reactor compartments are demanded. We anticipate that the implementation is feasible with methods from nanotechnology and synthetic biology already available today. Furthermore, our programming approach appears promising, especially, for synthetic biology because our organizational analysis only depends on stoichiometry of chemical reactions. Manipulating kinetic laws for fine-tuning is quite complicated in biological systems.

We have also introduced new analysis techniques using chemical organization theory for reaction systems such that computing devices are distributed in space. The artifice is to append spatial coordinates to the species name so that an identical species becomes distinguishable based on residing locations. Here, locations are represented by the device numbers where molecules are situated. Then, the expanded reaction networks are analyzed for the organizational structure within in order to take the global perspectives of the reaction systems' behaviors, but for a specific spatial graph

topology. While being independent of the topology, the organizational analysis is concentrated on each device for local perspectives. The device index is omitted from the species names because molecules are situated at the same or indistinguishable place, and molecular exchanges between devices, mediated through such as diffusion and transportation process, are modeled as inflows and outflows. Considering all possible environmental conditions the device encounters, the local analysis reveals all possible qualitative attractors of a local device.

The insight gained from the local and global analysis allowed us to understand how the system copes with qualitative perturbations and with a changing base graph topology. Consequently, we add a stochastic inflow of a particular species, which should initiate upward movements in the space of organizations after a topological change. This modification led to a robust system that can adapt to a dynamically changing topology.

When multiple devices are employed in a distributed environment, communication between devices plays an important role. Our assumption was that the communication is established by the exchange of molecules through directed diffusion or transportation along communication links. This sort of communication paradigm has been provoked and investigated for biological, nano-scale devices, instead of communication with electrons or electromagnetic waves, because of the power and size limitations (?). However, the traditional communication methods, e.g., by radio links are also suggested for nano-scale machines (Demoustier, Minoux, Le Baillif, Charles, & Ziaei, 2008). We also envision that the molecular computing metaphor can be used as a programming approach for that system. A molecule will be exchanged via virtual data packages transmitted by electromagnetic waves.

Here, our focus was on the manual construction of chemical programs such that, with the help of chemical organization theory, reaction networks are designed from scratch. It will be practical when manual construction methods are combined with optimization. One method can supply building blocks or network motifs for the other. We have seen benefits especially when an optimization process is guided with chemical organization theory (Lenser, Matsumaru, Hinze, & Dittrich, 2008).

Molecular computing paradigms Chemical programs that respect the medium, molecules.

In this paper, we proposed that chemical computing is the computation model for nano-scale device.

References

- Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T. F., et al. (2000). Amorphous computing. *Commun. ACM*, 43(5), 74–82.
- Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266, 1021–1024.
- Alon, U. (2006). *An introduction to systems biology: Design principles of biological circuits*. Chapman and Hall.
- Amos, M. (2004). *Cellular computing*. Oxford University Press.
- Aoki, T., Kameyama, M., & Higuchi, T. (1992). Interconnection-free biomolecular computing. *IEEE Computer*, 25(1), 41–50.
- Bachtold, A., Hadley, P., Nakanishi, T., & Dekkerdagger, C. (2001). Logic circuits with carbon nanotube transistors. *Science*, 294(5545), 1317 – 1320.
- Banâtre, J.-P., Fradet, P., & Radenac, Y. (2004). Principles of chemical programming. In S. Abdennadher & C. Ringeissen (Eds.), *Rule'04 fifth international workshop on rule-based programming* (pp. 98–108). Tech. Rep. AIB-2004-04, Dept. of Comp. Sci., RWTH Aachen, Germany.

- Banâtre, J.-P., & Métayer, D. L. (1986). *A new computational model and its discipline of programming* (Tech. Rep. No. RR-0566). INRIA.
- Banzhaf, W., Dittrich, P., & Rauhe, H. (1996). Emergent computation by catalytic reactions. *Nanotechnology*, 7(1), 307-314.
- Bath, J., & Turberfield, A. J. (2007). Dna nanomachines. *Nature Nanotechnology*, 2, 275-284.
- Berry, G., & Boudol, G. (1992). The chemical abstract machine. *Theor. Comput. Sci.*, 96(1), 217-248.
- Clarke, B. L. (1980). Stability of complex reaction networks. *Advances in Chemical Physics*, 42, 1-213.
- Conrad, M. (1985). On design principles for a molecular computer. *Commun. ACM*, 28(5), 464-480.
- de Silva, A. P., & Uchiyama, S. (2007). Molecular logic and computing. *Nature Nanotechnology*, 2, 399-410.
- Deckard, A., & Sauro, H. (2004a). Preliminary studies on the in silico evolution of biochemical networks. *ChemBioChem*, 5(10), 1423-1431.
- Deckard, A., & Sauro, H. (2004b). Preliminary studies on the in silico evolution of biochemical networks. *ChemBioChem*, 5(10), 1423-1431.
- Demoustier, S., Minoux, E., Le Baillif, M., Charles, M., & Ziaei, A. (2008). Review of two microwave applications of carbon nanotubes: nano-antennas and nano-switches. *Comptes Rendus Physique*, 9(1), 53-66.
- Dittrich, P., & Matsumaru, N. (2007, 17-19 Sept. 2007). Organization-oriented chemical programming. In *7th international conference on hybrid intelligent systems (his)* (pp. 18-23). IEEE. Available from papers/2007-DM-his.pdf
- Dittrich, P., & Speroni di Fenizio, P. (2007, May). Chemical organisation theory. *Bull Math Biol*, 69(4), 1199-1231. Available from <http://dx.doi.org/10.1007/s11538-006-9130-8>
- Feinberg, M., & Horn, F. J. M. (1973). Dynamics of open chemical systems and the algebraic structure of the underlying reaction network. *Chem. Eng. Sci.*, 29(3), 775-787.
- Feinberg, M., & Horn, F. J. M. (1974, March). Dynamics of open chemical systems and the algebraic structure of the underlying reaction network. *Chemical Engineering Science*, 29, 775-787.
- Fenizio, P. Speroni di, & Dittrich, P. (2007). Chemical organizations at different spatial scales. In F. A. e Costa, L. M. Rocha, E. Costa, I. Harvey, & A. Coutinho (Eds.), *Proc. european conference on artificial life (ecal07)* (Vol. 4648, p. 1-11). Springer, Berlin. Available from <http://dblp.uni-trier.de/db/conf/ecal/ecal2007.html#FenizioD07>
- Fontana, W., & Buss, L. W. (1994). 'The arrival of the fittest': Toward a theory of biological organization. *Bull Math Biol*, 56, 1-64.
- Gatermann, K., Eiswirtha, M., & Senssea, A. (2005). Toric ideals and graph theory to analyze hopf bifurcations in mass action systems. *Journal of Symbolic Computation*, 40(6), 1361-1382.
- Herman, T. (2003). Models of self-stabilization and sensor networks. In S. R. Das & S. K. Das (Eds.), *Iwdc* (Vol. 2918, p. 205-214). Springer, Berlin.
- Hiyama, S., Moritani, Y., Suda, T., Egashira, R., Enomoto, A., Moore, M., et al. (2005). Molecular communication. In *Proc. of the 2005 nsti nanotechnology conference*.
- Ikeda, M., Kamei, S., & Kakugawa, H. (2002). A space-optimal self-stabilizing algorithm for the maximal independent set problem. In *Proceedings of the third international conference on parallel and distributed computing, applications and technologies (pdcat)* (pp. 70-74).
- Kaleta, C., Centler, F., Fenizio, P. S. di, & Dittrich, P. (2008, Apr). Phenotype prediction in regulated metabolic networks. *BMC Syst Biol*, 2(1), 37. Available from <http://dx.doi.org/10.1186/1752-0509-2-37>
- Lenser, T., Matsumaru, N., Hinze, T., & Dittrich, P. (2008). Tracking the evolution of chemical computing networks. In S. Bullock, J. Noble, R. A. Watson, & M. A. Bedau (Eds.), *Proceedings of the eleventh international conference on artificial life*. MIT Press, Cambridge, MA.
- L.F. Landweber, E. W. (2002). *Evolution as computation*. Springer.
- Luby, M. (1986). A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15, 1036 - 1055.
- Matsumaru, N., Lenser, T., Hinze, T., & Dittrich, P. (2007). Toward organization-oriented chemical program-

- ming: A case study with the maximal independent set problem. In F. Dressler & I. Carreras (Eds.), *Advances in biologically inspired information systems* (Vol. 69, p. 147-163). Springer.
- Merkle, R. C. (2000). Molecular building blocks and development strategies for molecular nanotechnology. *Nanotechnology*, 11, 89–99.
- Papin, J. A., Reed, J. L., & Palsson, B. O. (2004). Hierarchical thinking in network biology: the unbiased modularization of biochemical networks. *Trends in Biochemical Sciences*, 29(12), 641-647.
- Păun, G., Rozenberg, G., & Salomaa, A. (1998). *Dna computing. new computing paradigms*. Springer.
- Peter, S. (2008). *Chemical organizations and continuous dynamics (in german)*. Unpublished master's thesis, Department of Mathematics and Computer Science, Friedrich Schiller University Jena.
- Pisanti, N. (1998). Dna computing: A survey. *Bulletin of the European Association for Theoretical Computer Science*, 64, 171–187.
- Păun, G. (2002). *Membrane computing: An introduction*. Berlin: Springer.
- Rothmund, P. W. K. (1996). A dna and restriction enzyme implementation of turing machines. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 27(12), 75-113.
- Sahle, S., Gauges, R., Pahle, J., Simus, N., Kummer, U., Hoops, S., et al. (2006, 3-6 Dec.). Simulation of biochemical networks using copasi - a complex pathway simulator. In *Winter simulation conference, 2006. wsc 06. proceedings of the* (pp. 1698–1706).
- Salata, O. V. (2004, Apr). Applications of nanoparticles in biology and medicine. *J Nanobiotechnology*, 2(1), 3. Available from <http://dx.doi.org/10.1186/1477-3155-2-3>
- Shukla, S. K., Rosenkrantz, D. J., & Ravi, S. S. (1995). Observations on self-stabilizing graph algorithms for anonymous networks. In *Proceedings of the second workshop on self-stabilizing systems* (p. 7.1-7.15).
- Siehs, C., & Mayer, B. (1999). Dynamical hierarchies of structure and control in chemical reaction networks. *Nanotechnology*, 10(4), 464–471.
- Tschudin, C. (2003). Fraglets - a metabolic execution model for communication protocols. In *Proc. ains'03*.
- Wang, J. (2005, Nov). Nanomaterial-based amplified transduction of biomolecular interactions. *Small*, 1(11), 1036–1043. Available from <http://dx.doi.org/10.1002/sml1.200500214>
- Willner, I., Basnar, B., & Willner, B. (2007, Jan). Nanoparticle-enzyme hybrid systems for nanobiotechnology. *FEBS J*, 274(2), 302–309. Available from <http://dx.doi.org/10.1111/j.1742-4658.2006.05602.x>
- Zauner, K.-P., & Conrad, M. (2001). Enzymatic computing. *Biotechnology Progress*, 17(3), 553–559.
- Ziegler, J., & Banzhaf, W. (2001). Evolving control metabolisms for a robot. *Artif. Life*, 7(2), 171-190.