

SELF-ORGANIZING AGENT-BASED GRADE TRANSITION IN DISTRIBUTED CHEMICAL REACTOR NETWORKS

M. Derya Tetiker* Eric Tatara** Michael North**
Fouad Teymour* Ali Cinar*

* *Illinois Institute of Technology, Chicago, IL 60616, USA ,*
Email: {tetimeh, tataeri, teymour, cinar}@iit.edu
** *Argonne National Lab, Argonne, IL, USA*
Email: north@anl.gov

Abstract: Agent-based control structures provide flexible and emergent solutions to complex nonlinear problems benefiting from properties such as modularity, adaptability, scalability and robustness. One such problem is product grade transitions in distributed process. The framework proposed earlier (Tetiker, 2006a) is extended by adding several layers of agents to control species percentage distribution in autocatalytic reactor networks. A deadlock detection layer is implemented to detect and solve conflicting cases between local controller agents. An auctioning mechanism is employed to promote competition between local controller agents leading to emergent solutions satisfying global constraints. The proposed architecture performed successfully to change the species percentage distributions without specifying the final configuration. *Copyright © 2007 IFAC*

Keywords: Grade transitions, Reactor networks, Agent-based systems, Self-organizing systems.

1. INTRODUCTION

Networked processes producing high value-added specialty products (pharmaceuticals, specialty polymers etc.) generate complex and challenging control problems. Usually, one knows the desired end-product qualities such as density and molecular weight distribution, however, the network configuration(s) that yields the desired product properties and the reconfiguration strategy to be applied to reach that final desired state may not be known a priori. Additionally, there are usually multiple system configurations that lead to the specified end-product qualities. It is a common practice to use the same production line for producing different grades of a product. Generating production campaigns of different grade products may usually result in off-spec products during transition from

one product grade to another. Minimizing such transients is a challenging task and may require reconfiguration of the whole process.

The problem of grade transition can be cast into two sub problems. The first one is to identify the final system configuration that produces the desired grade. Often there is more than one configuration that satisfies the final product requirements and the mechanism for finding the optimum configuration becomes the important factor. Once the final configuration is found, the second sub problem is to decide on the actions to be taken in the correct order to move the system from its current state to the final state. Earlier works demonstrated autocatalytic reactions in CSTR networks to simulate population dynamics, multiple species of organisms that compete on same

resources, or chemical manufacturing problems (Tatara *et al.*, 2004; Tatara *et al.*, 2005a; Tatara *et al.*, 2005b). Controlling the spatial distribution of autocatalytic species that compete for the same resources in a network of reactors can be achieved by simultaneous manipulation of interconnection flow rates within the system. Numerical experiments suggest that individual CSTRs in networks are capable of hosting only a single dominant species, while other competing species may be present only in trace quantities (Birol *et al.*, 2002).

Tetiker *et al.* proposed two different hierarchical agent-based frameworks for the problem of changing the spatial distribution of autocatalytic species in a CSTR network, given the initial and the final configurations of the system (Tetiker, 2006a; Tetiker, 2006b). The first framework uses a centralized approach where a global planner agent observes the whole network, calculates the optimum path to be followed to move the system from one spatial distribution to another and assigns local objectives to local agents. Each local agent tries to reach its assigned local objective which is to maximize a certain species in the reactor being controlled by manipulating interaction flow rates with its neighboring reactors. In the second framework, the path to be followed is not calculated by a central unit; instead, every local agent searches for the target species in the network, calculates the optimum path from that reactor and requests it. This approach makes every local agent a local planner, increasing the robustness and flexibility of the system. Since the framework is decentralized, failure of one local agent does not collapse the whole system. The agent-based system adapts to the changing network conditions and modifies its strategy on-the-fly, if a more feasible solution appears or the solution being executed becomes unreachable. The system is in that sense, self organizing and flexible to incoming disturbances and failures of local units by creating alternative solutions dynamically.

This paper focuses on both problems, by calculating a feasible final configuration yielding desired final product properties and the sequence of moves to achieve it. In addition to the decentralized framework proposed earlier, a deadlock detection and resolution structure is developed to make the framework operate more effectively. Finally, an auctioning mechanism is implemented into the local controller agents to identify the final system configuration that produces the desired grade. The percentage distribution of autocatalytic species the network produces is taken as the final product property desired (e.g.: Species 1 25%, Species 2 40% and Species 3 35%) and the goal is to find the optimum configuration that produces the targeted species distribution. By incorporating the decentralized framework proposed

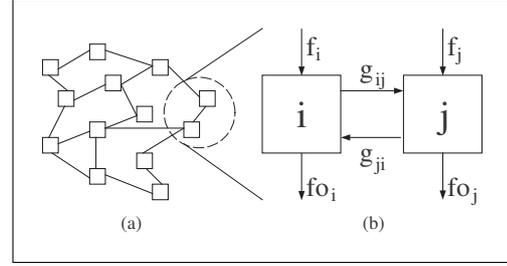
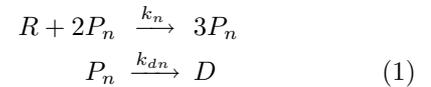


Fig. 1. (a) Network of interconnected reactors. Feed and exit streams are not shown. (b) Detail view of interconnection flows between reactors with feed and exit streams.

by Tetiker *et al.* (Tetiker, 2006b), the proposed method finds a reachable configuration that meets the final product requirements and moves the network to that configuration in a self-organizing manner.

2. CSTR NETWORKS

A network of I interconnected isothermal CSTRs (Fig. 1a) is modeled by specifying the material balance for each reactor i (Fig. 1b) in the network, $i = 1..I$. The cubic autocatalytic reaction for N autocatalytic species is,



where R is the resource, P_n the n^{th} species, and D a dead (inert) species. Reaction rate constants k_n and k_{dn} characterize the growth and death rates of the n^{th} species. The rates of change of the resource and species concentrations for reactor i in a network of I identical reactors of constant volume are

$$\begin{aligned} \frac{dr_i}{dt} = & - \sum_{n=1}^N k_n r_i p_{ni}^2 + f_i - r_i \left[f_{o_i} + \sum_{j=1}^I G(i, j) \right] \\ & + \sum_{j=1}^I r_j G(j, i) \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{dp_{ni}}{dt} = & k_n r_i p_{ni}^2 - p_{ni} d_n + f_i p_{fni} \\ & - p_{ni} \left[f_{o_i} + \sum_{j=1}^I G(i, j) \right] + \sum_{j=1}^I p_{jn} G(j, i) \end{aligned} \quad (3)$$

by defining the variables as $r_i = R_i/R_0$, $p_{in} = P_{ni}/R_0$, $f = F/(VR_0^2)$, $f_o = F_o/(VR_0^2)$, $d_n = k_{dn}/R_0^2$, and $t = R_0^2 t'$, where subscript $_0$ denotes the feed and subscript $_i$ denotes reactor i . P_{ni} is

the n^{th} species concentration in reactor i , F is the feed flow rate, F_o is the exit flow rate and V is the reactor volume. The interconnection matrix G defines the strength of the interconnection flow rates between networked reactors, such that $G(i, j)$ is the interconnection flow rate from reactor i to reactor j .

In the case study, the interconnection flow rates are used as manipulated variables. The system is operated with constant volume, thus, constraint equations are formulated on the reactor flow rates to ensure that material is conserved. The reactor flow inputs include the reactor feed and the interconnection flows from the neighboring reactors. Outflow rates from each reactor include the interaction outflows to neighboring reactors as well as the drain.

3. SIMULATION ENVIRONMENT AND AGENT LAYOUT

The agent-based framework proposed in this work addresses the problem of reconfiguration of a CSTR network when transitioning between different grade products. The objective of the framework is to be able to move the network from its current configuration to another configuration that yields the output satisfying the product quality requirements using the most feasible path possible. Supervision of such transition requires concurrent manipulation of variables such as interconnection flow rates and feed flow rates. The architecture described in this paper incorporates the work by Tetiker et al. (Tetiker, 2006b).

The hierarchical agent-based system consists of several layers containing multiple types of agents operating in coordination (Fig. 2). The architecture is modular so that addition or removal of layers can be made. It also allows using multiple techniques for the same task and makes the system more robust. The lowest layer in the framework is the process layer where, network elements such as reactors and valves are modeled, each unit containing variable information specific to that unit such as concentrations, feed flow rates and valve openings. This layer is a process simulator and provides sensor and actuator information. This layer can be replaced with the simulator of any specific process or actual plant data and interfaced with the layers above it.

The layer above the process layer contains local controller agents where each reactor in the network is assigned a local controller agent. Local controller agents have their local objectives in the form of maximizing the concentration of certain species and control the dominant species in the reactor. Local controller agents manipulate the

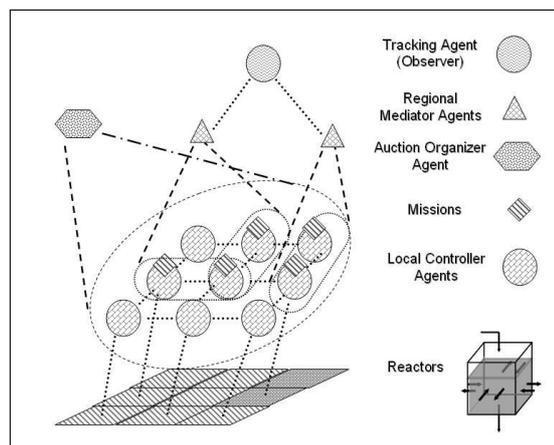


Fig. 2. General organization scheme displaying different layers of agents in the framework.

feed flow rates coming into the reactor and interconnection flow rates with its neighbors (Fig. 3). Depending on its objective every local controller agent searches for the species it is trying to maximize in the network. A well-known algorithm in graph theory, Dijkstra's shortest path algorithm, is implemented to calculate the distance between reactors (Dijkstra, 1959). Two different approaches are used in defining the distance between two reactors. The first approach assumes that the distances between two neighboring reactors are the same. A drawback of this approach is that there exist multiple paths having the same distance between two reactors which may result in selecting an infeasible path. The second approach looks at the species concentrations of two connected reactors for defining a distance between them. The distance between two neighboring reactors is calculated by summing the differences between the concentrations of undesired species and desired species for both reactors. This quantity may be viewed as a measure of difficulty to invade the connected reactor. An object called "Mission" is created when the desired species is found in the network and that mission is requested from the closest reactor that contains the species. Missions contain information about the source and the destination reactors, the distance between the source and the destination, the next reactor on its current path and the species that is requested from the source.

Since the framework is decentralized, every local controller agent carries out the species search concurrently. A queuing algorithm is implemented to avoid deadlocks in the agents receiving multiple missions at the same. Missions are collected in mission lists and prioritized according to the distances between the source and the destination reactors. This mechanism allows self-organization of the network to the desired configuration in a robust and efficient way. Every time the requested species moves to the next reactor on the path cal-

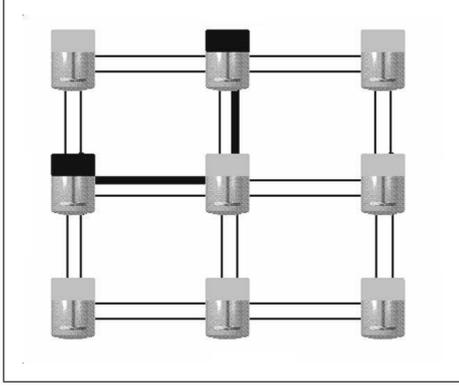


Fig. 3. The dominant species in the center reactor is Species 2 (gray). The goal of the reactor is to maximize Species 1 (black). Thicknesses of interconnections represent the magnitudes of the flow rates.

culated by Dijkstra’s algorithm, the distances for the mission objects are updated and the shortest path is recalculated, because the network conditions might have changed since that mission is created and a more feasible path may be available at that moment. Whenever a certain species is transferred from one reactor to a neighboring reactor, the relevant mission is removed from that local controller agent’s mission list and that agent moves to the next mission in its mission list. The network reaches its final configuration when all the missions are completed.

There are also cases where conflicting missions block each other’s way and none of the missions can reach to their destination reactors. A deadlock detection layer is implemented where every time a mission object is created/destroyed, that object is registered/unregistered to a global observer agent. The goal of the global observer agent is to detect possible deadlocks in the network. It detects possible deadlocks by creating directed graphs as the conditions in the network changes (as the mission objects move). In the directed graphs constructed, every mission object and the local controller agents sending and receiving missions are considered a node. If a local controller is sending the mission, a directed link from the mission to the local controller is drawn. If a local controller is receiving a mission, a directed link from the local controller to the mission is drawn. Whenever the constructed graph contains a cycle, a deadlock is detected. Detection of the deadlock is followed by creation of a mediator agent assigned for supervising the deadlocked region. Conflicting missions are removed from the relevant local controllers mission lists and the mediation agent computes a solution to a portion of the overall problem and recommends local strategies to the agents involved in the mediation section.

Because the main problem is grade transition of the product and property being controlled is the

final weight distribution of species in the network, the decision on which reactors should produce what species can be adjusted to optimize the system as long as final weight distributions are matched with the product specifications. An auctioning mechanism is implemented to promote self-organizing behavior in the system, which allows local controllers to compete with each other to be able to produce certain species in its reactor. At the beginning of the simulation, an auction organizer agent calculates how many reactors should produce each species according to the final product specifications and starts a separate auction for every species that should be produced. The auction organizer agent’s job is to take bids from local controller agents, identify the winners and assign the objectives (which species to produce) of local controllers that won the auction. There is a physical constraint in this auction. A local controller can only win a single auction. In other words, a reactor cannot produce multiple species dominantly at the same time. Every local control agent bids for each species available in the network. The bid values for the auctions are determined by a cost-reward function specific for every system being controlled.

In this case study, the cost-reward function measures how much a reactor is willing to produce certain species. It calculates how strong that species is with respect to the other species in the reactor in terms of concentration differences and it sums it with the availability of that species in the neighboring reactors. A higher value of the cost-reward function means a better chance for that reactor to succeed in producing that species. The weights in the cost-reward function can be adjusted so that different system behaviors can be observed by increasing the importance of neighboring reactors or the reactor itself. The cost-reward function used in the case study is given in 4. The first term in the equation calculates the strength of the species in the reactor itself. The second term in the equation is the part where the availability of certain species in the neighboring reactors is calculated. W_1 and W_2 are weight parameters. Using W_1 values bigger than W_2 prioritizes the reactor itself over the neighbors and results in more scattered final distributions, whereas the opposite yields more clustered final configurations.

$$R = W_1 \left[C_{dom} - \sum_{n=1, n \neq dom}^{NumSpecies} C_n \right] + W_2 \left[\sum_{m=1}^{Neighbors} \left(C_{m, dom} - \sum_{n=1, n \neq dom}^{NumSpecies} C_{m, n} \right) \right] \quad (4)$$

The reactor network model and agent-based control system is implemented with the open source agent modeling and simulation environment RePast (Collier *et al.*, 2003). The RePast toolkit is a Java-based framework for agent simulation and provides features such as an event scheduler and visualization tools. The control agents created with RePast interact with virtual representations of the physical reactor network. The virtual network objects map the states of the physical system to objects that can be manipulated by control agents. The interface between the physical network and the agent environment can take the form of a data acquisition system in the case of a real process, or in this case, a simulator of a chemical reactor network. The ordinary differential equations that describe the autocatalytic reactions in each CSTR are solved numerically using the CVODE solver (Cohen and Hindmarsh, 1994). The solver code is written in C and linked with RePast via the Java Native Interface (JNI).

4. RECONFIGURATION OF THE REACTOR NETWORK

The agent-based framework is tested on a network that has twenty five (5x5 grid) chemical reactors hosting two autocatalytic species competing with each other. The objective is to change the percentage distribution of the two species without specifying the final configuration to demonstrate self-organizing product grade transition. Parameters in the model are kept the same with the earlier works of Tetiker *et al.* (Tetiker, 2006b). The growth and death rate parameters for the species are chosen to be equal ($k_g = 25$, $k_d = 0$) in order to reduce the possibility of one species dominating the other. Initially all feed flow rates ($f = 0.0125$) and all interconnection flow rates ($g = 0.001$) are uniform for all reactors. Initial species concentrations in the reactors are selected randomly with the constraint of initial species percentage distributions are matched.

The screen capture of the framework developed (Fig. 4) displays several different views of the physical system and different layers of agents. In the physical system view, reactors and their interconnections are displayed. The thickness of the lines between reactors signifies the value of interconnection flow rates. The color of each reactor shows the dominant species in the reactor. The local controller agent's view displays the current missions each local controller is working on. The first and second numbers display the source and destination reactor of the current mission respectively, and the last number shows the number of steps calculated to reach from the source reactor

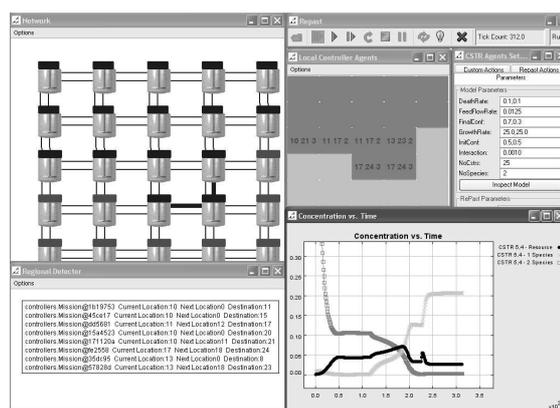


Fig. 4. Screen capture of the software developed in Java based RePast environment.



Fig. 5. Initial network configuration. Species 1 produced in 20% of the reactors (top left). Species 2 produced in 80% of the reactors.

to the destination reactor. The observer agent's view displays the list of ongoing missions throughout the whole network. Species concentration profiles of the desired reactors are also available as a separate display.

The test case is to move the system from the initial state producing 20% species 1 and 80% species 2 to a final state of 70% species 1 and 30% species 2. Every local controller agent uses the same weight parameters in the cost-reward function. Several different combinations are tested to study the effects of cost-reward function parameters on the final distribution of species. Fig. 5 shows the initial species distribution in the network where the species 1 (dark gray) is produced in the upper left corner of the network. Fig.6 displays the final configuration reached with cost function parameters $W_1=5$ and $W_2=0.5$. In the cost function, assigning W_1 bigger than W_2 increases the weight of the species strength in the reactor itself over its neighbors. The initial concentrations of species in the reactors were chosen randomly. This leads to a scattered from configuration since the bid values calculated using cost function is affected the most from the species concentration in the reactor. When the weight parameters are changed to $W_1=0.5$ and $W_2=5$, the final configuration expected is more clustered groups of species in the

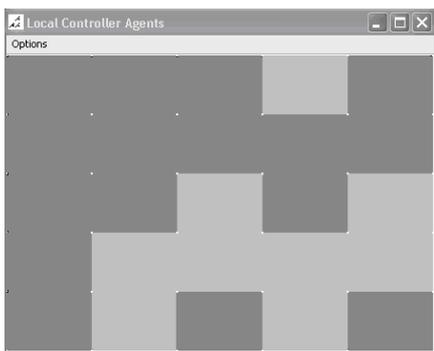


Fig. 6. Final network configuration with the cost-reward function parameters $W1=5$, $W2=0.5$. Species 1 produced in 70% of the reactors. Species 2 produced in 30% of the reactors.



Fig. 7. Final network configuration with the cost-reward function parameters $W1=0.5$, $W2=5$. Species 1 produced in 70% of the reactors. Species 2 produced in 30% of the reactors.

network since the weight assigned to the availability of species in the neighboring reactors affects most the final distribution (Fig.7). The configuration obtained in each run is different but consistent with final product specifications (70% species 1 and 30% species 2). Every run of the simulation with the same parameters may produce a unique final configuration because of the differences in initial species concentrations in the network.

5. SUMMARY AND CONCLUSIONS

The decentralized agent-based system proposed earlier (Tetiker, 2006b) is extended to simulate self-organizing product grade transition in an autocatalytic chemical reactor network. Additional layers of agents are implemented to improve detection and solution of conflicting situations. An auctioning mechanism is proposed to promote competition between local controller agents leading to emergent solutions satisfying physical system constraints. A modular cost-reward function is implemented in local controller agents so that they can calculate their bids. The weights for the parameters of the cost-reward function determine the pattern of the final network configuration.

Higher weights for availability of the species in the neighboring reactors result in more clusters of species whereas higher weights in the species concentration in the reactor itself leads to more distributed solutions. The design of the cost-reward function is problem specific. The agent-based control framework developed is able to change the species percentage distribution in the reactor network by using distributed artificial intelligence without specifying the final configuration by executing the right combination of strategies.

6. ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. CTS-0325378 of the ITR program.

REFERENCES

- Birol, I., S.J. Parulekar and F. Teymour (2002). Effect of environment partitioning on the survival and coexistence of autocatalytic replicators. *Physical Review E* **66**, 051916.
- Cohen, S.D. and A.C. Hindmarsh (1994). *CVODE User Guide*. LLNL Report UCRL-MA-118618.
- Collier, N., T. Howe, and M. North (2003). Onward and upward: The transition to repast 2.0. In: *First Annual North American Association for Computational Social and Organizational Science Conference, Pittsburgh, PA*.
- Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Math.* (1), 269–271.
- Tatara, E., I. Birol, F. Teymour and A. Cinar (2004). Static and dynamic behavior of autocatalytic replicators in reactor networks. *Industrial and Engineering Chemistry Research* **43**, 3972–3993.
- Tatara, E., I. Birol, F. Teymour and A. Cinar (2005a). Agent-based control of autocatalytic replicators in networks of reactors. *Computers & Chemical Engineering* **29**, 807–815.
- Tatara, E., I. Birol, F. Teymour and A. Cinar (2005b). Measuring complexity in reactor networks with cubic autocatalytic reactions. *Industrial and Engineering Chemistry Research* **44**, 2781–2791.
- Tetiker, M. D.; Artel, A.; Tatara E.; Teymour F.; North M.; Hood C.; Cinar A. (2006a). Agent-based system for reconfiguration of distributed chemical reactor network operation. In: *American Control Conference, Minneapolis, MN*. pp. –.
- Tetiker, M. D.; Cinar, A; Teymour F.; North M. (2006b). Decentralized multi-agent control of distributed reactor networks. *AIChE Annual Meeting*. San Francisco, CA, November 13-17.