Modular combinations of Artificial Chemistries

Penelope Faulkner Rainford[1,3], Angelika Sebald[1,3] and Susan Stepney[2,3]

[1]Department of Chemistry, University of York, UK
[2]Department of Computer Science, University of York, UK
[3]York Cross-disciplinary Centre for Systems Analysis
pf550@york.ac.uk

Abstract

We introduce a modularisation of artificial chemistries (AChems). This allows us to define a standard linking method between AChems. We illustrate the approach with a system that nests a Jordan Algebra AChem (JA AChem) inside agents of SwarmChem, and show how our modular approach allows us to define and experiment with multiple variants in a standard manner. Potential for future formalisation is discussed.

Introduction

Sub-symbolic artificial chemistries (AChems) (Faulconbridge et al., 2011; Faulconbridge, 2011; Faulkner et al., 2018) are generally AChems whose atoms and particles have internal structure that defines their behaviour. These systems so far have been analogous in their behaviour to natural chemistry viewed at the level of atoms and molecules.

Many other AChems work to reflect the properties of chemistry at the level of cells (Madina et al., 2003; Hutton, 2007) or chemical reaction systems (Soula, 2016). In natural chemistry these different levels are closely related: cells contain chemical reaction systems, and chemical reaction systems are based on individual particle and atom interactions. While attempts have been made to bridge the gaps between such levels in individual systems (Liu, 2018), so far the systems are very simple and lacking in more complex features.

We propose to take advantage of feature-rich existing AChems by combining them to give a system that can span different levels of activity and behaviour in a single AChem system. We demonstrate this approach through our own Jordan Algebra Artificial Chemistry (JA AChem) (Faulkner et al., 2016, 2017) and Sayama's SwarmChem (Sayama, 2009, 2010, 2011, 2018).

JA AChem particles are algebraic objects. It can form particles with complex structures. It is a sub-symbolic system that has probabilistic linking and decomposition. So far this system has been aspatial, working on a well-mixed tank model in which all particles can link with each other.

SwarmChem is based on Reynolds' Boids (Reynolds, 1987). Its particles or agents move and flock based on their parameter values. In existing SwarmChem instantiations the sets of parameter values are generally evolved with human interaction to generate flocking behaviours. The system agents contain all sets of values in the system and choose one to use based on a weight matrix. These are changed by collisions. In the version of SwarmChem used here we do not have a predetermined set of possible values. In our version the agents swap some number of parameter values. The only limitation is that the parameter values must make sense after the exchange (for example, normal velocity must not exceed maximum velocity).

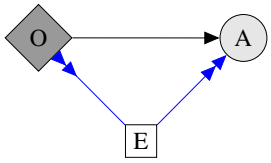

Figure 1: Communication link between first AChem (dark grey) and the second AChem (light grey). Single arrows (black) show control flow, double arrows (blue) show information flow. The observer node observes the first AChem's tank and the action node acts on the second AChem's tank.

Method

We can connect any two AChems by giving them the ability to communicate via their environment. This communication can be uni- or bi-directional. In order to talk about combining AChems we need to be able to talk about different parts of different AChems. We do this by representing information and control flow in an AChem in a graph with multiple types of nodes and edges, Table 1. The basic graph structure of communicating AChems is given in Figure 1.

We use shading to indicate the ownership of a node by a single system. A node owned by one AChem cannot directly communicate with the nodes owned by a different AChem. Instead, information is shared using an environmental node that is not owned by either AChem. So in Figure 1, the dark grey observation is of a tank in the 'dark grey' AChem, and the light grey action is on a tank in the 'light grey' AChem.

Table 1: Legend for graph notation

Element	Description
T	**Tank** containing particles.
E	**Enviroment** containing non-particle variables and information in the system.
S	Moves particles between containers by **sampling** them
O	**Observes** particles and produces summary statistics (saved in the environment)
A	Performs **actions** on particles based on state of particles and environment
	Arrow between **Sampling**, **observer** and **action** nodes to indicate control flow in system.
	Arrow from **Sample**, **observer** or **action** node to a **tank** or **enviroment** node indicating information flow out of the control node.
	Arrow from **tank** or **enviroment** node to a **sample**, **observer** or **action** node indicating information flow into the control node
	Arrow from **tank** or **enviroment** to a **sample**, **observer** or **action** node indicating information flow into and out of the control node

The figure shows uni-directional communication, in which one AChem influences the other. By adding a second link in the other direction we could establish bi-directional communication. Both the action and the observation are defined by the designer.

For example, if we wish to establish side-by-side chemistries then our observation will produce a summary statistic that is a value, or set of values, based on the whole system, which will uniformly affect the entire system in the second chemistry. Alternativly, the observer can generate statistics based on individual particles, which can then affect individual particles in the second AChem.

We give an example of a "nested", or multi-level, AChem with bi-directional communication. The observer of the lower-level AChem generates a set of values over a large number of particles in that AChem. These values are then used to influence the behaviour of a single individual in the higher-level AChem. In turn the behaviour and interaction of one or two particles in the higher-level AChem influence a large number of particles in the lower-level AChem.

Example: Nested Chemistry

We generate a new set of chemistries by converting two AChems from the literature (Faulkner et al. (2016) and Sayama (2009)) to our modular graph form, and then linking these modules in various ways to give seven distinctive systems; an eighth system is achieved through a change in system settings.

The largest of these systems is a fully nested AChem that contains all modules used in our systems, Figure 2. In this system we treat each of the agents of SwarmChem as a well-mixed tank of JA AChem particles.

We have two links in our nested system: Parameter Setting and Transfer. In Parameter Setting we generate a set of parameters values for each SwarmChem agent based on the particles in its tank. These sets are saved into the environment and used to update the Swarm. In Transfer we detect collisions between members of the Swarm. This information is saved in the environment and then used to transfer particles between the well-mixed tanks of the JA AChem system.

This provides a means of communication between the two systems. SwarmChem's spatial movement provides a limitation and control on particle exchanges in JA AChem between different tanks. Likewise, the JA AChem tanks communicate with SwarmChem by changing its parameter values, which influences the agents' spatial movement and likelihood of collision.

We divide the nested chemistry system into five sections, as shown in Figure 2:

Initialisation Initial tanks of JA AChem particles and initial swarm agents are loaded into the system and stored separately with matching indexing to allow for reference between the two.

Parameter Setting The current contents of the tanks are analysed to produce a set of parameter values for each tank. These are stored in the environmental variable "E:parameters" (2). The swarm then updates itself based on the content of this variable.

SwarmChem The agents of SwarmChem update themselves using a single SwarmChem time step.

Transfer SwarmChem assesses whether any collisions have occurred between its agents in the system. It stores out a record of these collisions in the environmental variable "E:transfers". JA AChem uses this variable to exchange parameter values between tanks based on the SwarmChem collisions.

JA AChem Updates by performing a number [1] of attempts at bonding and an equal number of attempts at decom-

[1] See "gen size" in Table 2

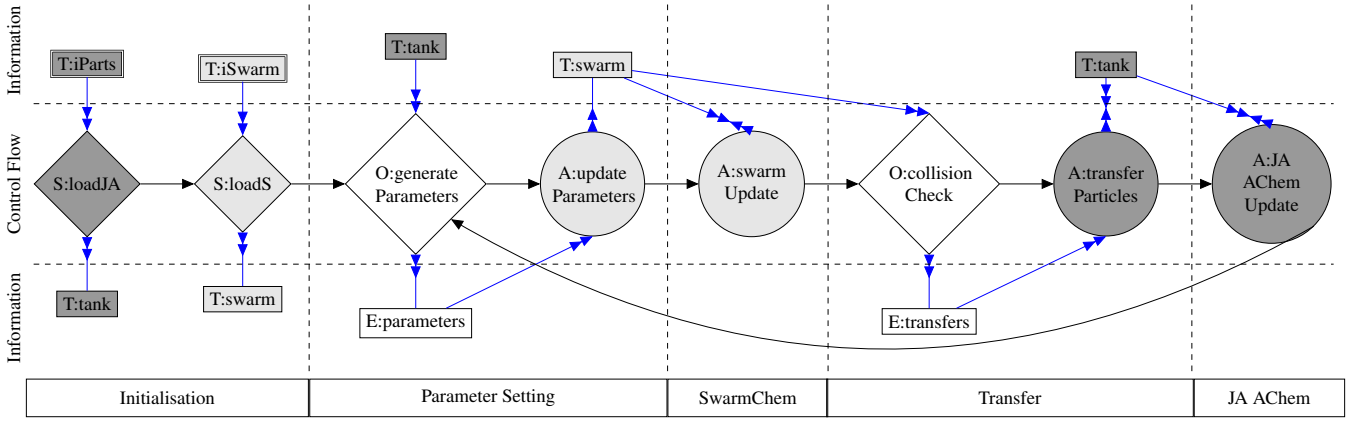

Figure 2: Graph of nested chemistry using metachem modular graph notation. JA AChem nodes are shown in dark grey, SwarmChem nodes in light grey. White nodes are either shared or not natively part of either AChem.

position. All tanks are independent and mass-conserving well-mixed tanks.

Modular Systems

From this full system we can derive eight variant systems. The control flow of these systems is shown in Figure 3.

I. Nested. The full Nested AChem system as shown in Figure 2

II. Nested without collision. JA AChem particles are not transferred between tanks, but still determine the parameter values of agents in the SwarmChem

III. SwarmChem. SwarmChem agents randomly exchange parameter values on collision; there is no communication with the JA AChem.

IV. SwarmChem without collision. A very basic form of SwarmChem in which the agents interact only through Boid like flocking behaviours.

V. JA AChem single tank. A single well-mixed tank of JA AChem. The same number of evaluations are used per generation and the same number of starting particles are also used as the other systems.

VI. JA AChem multiple tanks with no interaction. A JA AChem with the same number of tanks as in the nested version; there are fewer atoms and particles in each tank, but the same number of overall atoms and evaluations are used.

VII. JA AChem multiple tanks with random transfers. The same system as in **VI** but with tanks randomly selected to randomly transfer particles between them.

VIII. JA AChem multiple tanks with grid transfers. The same as in **VII** but transfer tanks selected based on a Moore Neighbourhood, Figure 4

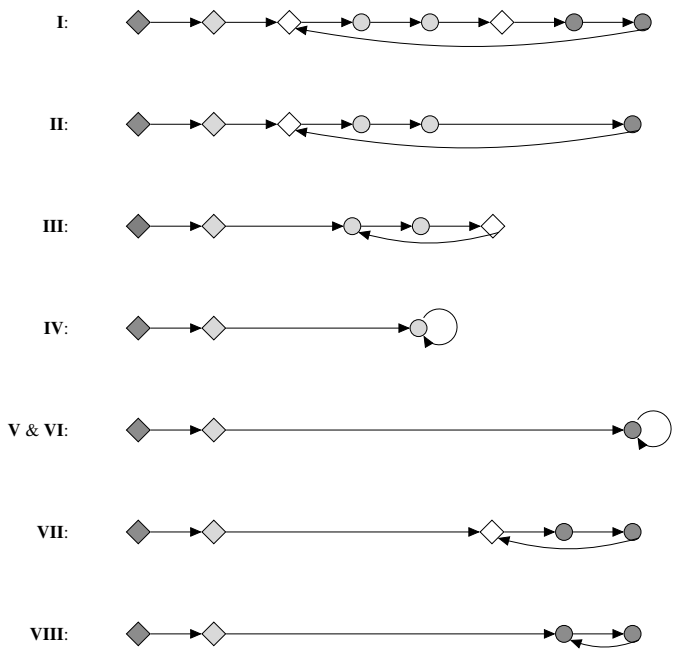

Figure 3: System Combinations

Settings: Jordan Algebra AChem

The Jordan Algebra AChem is an atom-scale sub-symbolic artificial chemistry. It has previously been presented as a single well-mixed tank.

JA AChem's properties emerge from its underlying algebraic structures. It uses 3×3 Hermitian matrices to represent both atomic and composite particles, and the Jordan matrix product (McCrimmon, 2006) to represent linked particles:

$$A \bullet B = \tfrac{1}{2}(AB + BA) \quad (1)$$

This provides a commutative non-associative product. Without these properties our product could result in our composite particles being "bags" (multi-sets) of atoms in which the

Algorithm 1 Description of a single update on a JA AChem tank

```
Tank := {69 atomic particles}
for 1 to no_of_rounds do
    LINKING PHASE
    DECOMPOSITION PHASE
LINKING PHASE
collect data from Tank

procedure LINKING PHASE
    for 1 to generation size do
        reactants := 2 random particles from Tank
        while RANDOM() < Xcoll do
            reactants += a random particle from Tank
        attempt to link reactants
        if successful then
            add new particle to Tank and remove reactants

procedure DECOMPOSITION PHASE
    for 1 to generation size do
        particle := random particle from Tank
        attempt to decompose particle
        if successful then
            remove particle from tank and add new particle(s) to Tank
```

internal structure of bonding has no effect. With them, our product allows non-trivial isomers (Faulkner et al., 2016). JA AChem uses the matrices' eigenvalues and vectors to define the probability of a link forming, and for other parts of our system.

When linked, the resultant particle can be represented as a tree, with atoms as its leaves and sub-particles at each node. The algorithm for one generation of JA AChem is given in Algorithm 1. [2]

In order to nest JA AChem into SwarmChem we need to work with multiple tanks. The algorithm is an update cycle for a single tank. We can apply this to each tank in the system.

The relevant settings for the six systems that include JA AChem here are given in Table 2. [3]

These settings give us the same number of linking and decomposition attempts in all systems. The three collision methods work as follows:

Collision: Transfers happen between two tanks when their associated SwarmChem agents collide.

Grid: Transfers happen by randomly selecting a single tank and then selecting a second tank from a grid based neighbourhood, see Figure 4.

[2] $Xcoll$ is the probablity that an additional particle will be involved in a link

[3] All systems use an additional collision probabilty of 0.2 and the Boltzmann inspired b probability from Faulkner et al. (2017)

Table 2: JA AChem Settings

System	**I**	**II**	**V**	**VI**	**VII**	**VIII**
#tanks	50	50	1	50	50	50
tank size	200	200	10000	200	200	200
gen size	200	200	10000	200	200	200
#rounds	100	100	100	100	100	100
transfers	collision	none	none	none	grid	random

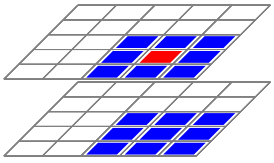

Figure 4: Grid based neighbourhood

Random: Two tanks are chosen at random and exchange materials.

Settings: SwarmChem

SwarmChem is a spatial AChem based on Boids (Reynolds, 1987). There is one set of settings relevant to SwarmChem. In all experiments we have 50 agents in a 2000×2000 space of arbitrary spatial units.

IV is the most basic version of SwarmChem, in which parameter values are fixed throughout and agents do not change their behaviour during a run.

In **III** we further develop the SwarmChem system to include changes in parameter values. In traditional SwarmChem this is done with weight matrices that assign priority to different parameter value sets. We use another method, still based on collisions being the trigger for change. For comparison purposes, we make the parameter value changes more analogous to the transfer of particles in the nested system. In the case of a collision in **III** we swap a random number of the agents' parameter values. Swaps are then reversed if they give contradictory settings (e.g. maximum velocity less than normal velocity).

In **I** and **II** the SwarmChem agents have an internal state defined by the tank of associated JA AChem particles. These particles change over time as they react, and in doing so change the parameter values of the SwarmChem agent.

In **I** we also have collisions that cause the transfer of JA AChem particles from one agent's tank to another, allowing for information transfer between agents, which may prevent the agents from becoming stable.

Analysis methods: Jordan Algebra AChem

We focus our analysis of the JA AChem level of our systems on the size of the particles and the resultant number of particles in the system or tanks over time.

These numbers should stabilise quickly in the system, but we expect the systems with transfers to be less stable than

others. Particles being transferred in and out of the tanks should disturb any equilibrium.

We also expect to see larger particles in the partitioned systems as the smaller size of the tanks limit the sampling possibilities, increasing the chances of selecting molecules which already contain multiple particles. As these are used and the number of particles in the tank decreases, these probabilities should further increase.

Analysis methods: SwarmChem

We can observe many different statistics on the agents of the swarm; here we focus on the relative position of an agent to its visible neighbours. This provides us with measure of how well our agents are clustered with each other. In homogeneous flocking this parameter's value should be very similar across agents, as a flock all have the same perception radius and tendency for avoidance. In SwarmChem these have greater variation but should be similar in sets of agents forming a swarm. Here we therefore expect to see greater variation in the nested SwarmChem where all values of perception radius and tendency for avoidance are possible.

Results and Discussion

Effect on JA AChem

There is a statistically significant difference in the sizes of particles in the system between the single tank, **III**, and everything else ($p < 10^{-33}$ ranksum test, large effect size $A = 0.99$ (Vargha and Delaney, 2000)). It also stabilises at a much higher number of particles (8824 particles) compared to the other systems (**I**, **II**, **VI**, **VII**, **VIII**, with 3750 – 3850 particles). We can therefore reject the null hypothesis that our changes to the system have no effect.

Looking at the multitank systems we can see that these systems have similar particle sizes, Figure 5. These have similar medians, but somewhat different distributions. Most distinctive is **VIII**, which is statistically significantly different (ranksum) from all the medians (except **II**), with medium or larger effect sizes.

We can predict that **II** and **VI** should be equivalent at the JA AChem level. This is because there is only a single unidirectional link in **II**. This is true in our results ($p = 0.057$, ranksum). This is particularly true as we are using Bonferroni corrections giving us a threshold for significance at 0.0032 rather than 0.05. This suggests that the number of runs and number of generations in the JA AChem has been sufficient in this example to give us the general behaviour of these systems.

Effect on SwarmChem

There are many different versions of SwarmChem, which can best be categorised in terms of their morphogenic hierarchy:

Category A: Homogeneous Swarm

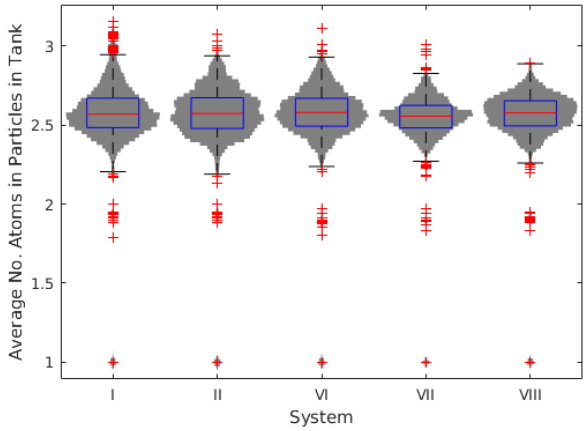

Figure 5: Average number of atoms in each particle in each Jordan Algebra AChem tank in a multitank system

Category B: Heterogeneous Swarm

Category C: Heterogeneous Swarm with redifferentiation

Category D: Heterogeneous Swarm with redifferentiation and information sharing

Here differentiation means that a particular agent can have different parameter values and redifferentiation means that those values can change.

Category A represents the original Boids work with its heterogeneous flocks of agents. Category B can be seen in the early forms of SwarmChem where we have a heterogeneous swarm. SwarmChem then moves to a Category C system by adding differentiation by changing the weighting of parameter value sets based on its local neighbours In more recent work (Sayama, 2018) SwarmChem introduces a version which could be seen as bridging the gap to Category D by allowing local information sharing.

Our full system, **I**, falls solidly into Category D with transfers and the differentiation that comes from the JA AChem. When we lose the transfers in **II**, we have a Category C. Our simpler AChems both fall mostly in Category B as heterogeneous swarms.

We get flocking behaviour in all systems, despite no control or evolution to develop it. Figure 6 shows that **I** and **II** form similar swarms to **III** and **IV**. This is contrary to our original expectation that forming swarms would be harder for nested systems. It may be because the JA AChem tanks tend to similar configurations of particles over time. In comparison the SwarmChem-only systems finish with the same parameter values as they start with. This seems to be reflected in our observations which show very little difference in our systems. However we do over all runs find a large difference in the relative position, p_r, of agents to their neigh-

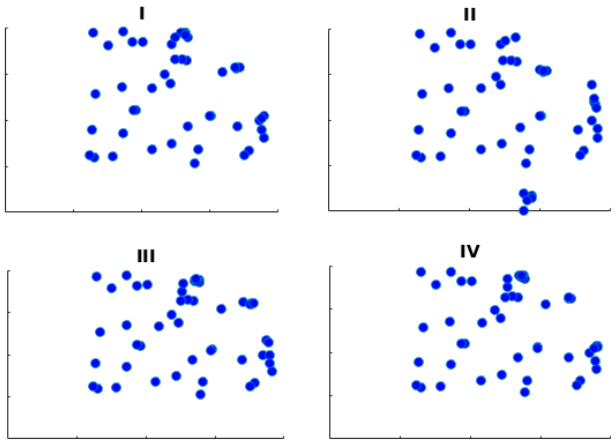

Figure 6: Image of each swarm system at the end of a run which started from the same initial conditions. Swarms are depicted in the full 2000×2000 (units arbitrary) space

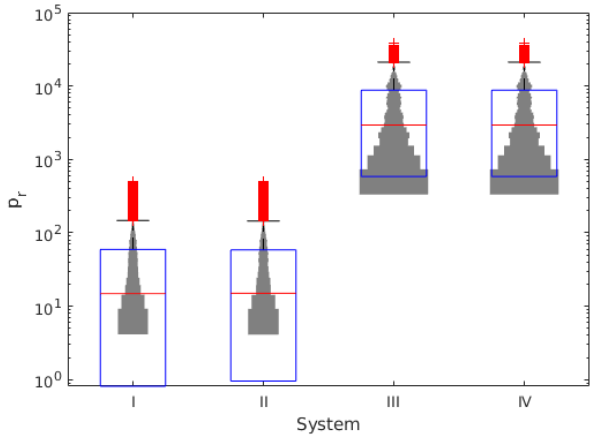

Figure 7: Log scaled final position of each agent to its neighbours relative to the agent's perception distance

bours (Figure 7)

$$p_r = |\langle x \rangle_i - x_i|^2 / R_i^2 \tag{2}$$

where $\langle x \rangle$ is the average position of an agent i's neighbours. x_i is an agents current position and R_i is the agents perception distance.

This shows we have significant differences between our nested systems (**I** and **II**) and our swarm systems (**III** and **IV**). They are statistically different ($p = 10^{-207}$, ranksum) with large effect size ($A = 0.90$). Looking to further differentiate our systems we find no difference between the swarm systems. We do not find any significant difference between **I** and **II** at the SwarmChem level. This may be the length of run though as the transfers will have very little chance to have an effect in 100 generations. We see no distinction between our two pure SwarmChem systems either. Further research should look at the effect over longer runs.

This gives us distinction between systems with redifferentiation and without. This shows that a multilevel system such as system **I** and **II** can produce distinctive behaviour in SwarmChem. While System **I** does meet all the criteria of a category D system, more work is needed to show what effect this has as opposed to **II** which is only a category C system.

Conclusions and Further Work

We have introduced the idea of modular graph based descriptions of AChems that allow 'plug and play' composition. The modular nature of these representations allows us to build and implement a wide variety of AChems from a single set of components. We can expand the number of AChems simply by expanding the number of modular components available to us. We have implemented a general method of composition using indirect communication for environment orientation (Hoverd and Stepney, 2009) at the level of our system graphs.

This approach enables us to use existing AChems to explore new questions with minimal new code required. As well as new systems, this serves to expand the implementations and capabilities of the individual AChems.

With further development the modular system used to build NestedChem can be expanded into a formal language for describing AChems. If this were done rigorously it could provide new possibilities for evolving and expanding new and existing AChems.

The "rigorous" requirement of defining a language is to enable evaluation, classification and comparison of these systems. This will require the development of new mathematical tools and analysis metrics to use the well defined language. This will allow us to push forward the development of AChems in general by preventing systems from unknowingly repeating work and ground already covered.

In this work we chose to nest JA AChem inside of SwarmChem, as this seems a natural choice of combinations. Further work could investigate different forms of nesting, such as inverse nestings and side-by-side combinations of these two AChems, or with entirely different AChems.

With the combinations of AChems now possible, and a modular description of AChem processes, we can turn to developing the ability for AChems to change their own control flow during a run. This can be done at a basic level through the introduction of decisions in the control flow. In the longer term, we wish to develop the ability for AChems to reorganise and reimplement modules while running, in response to their internal state or an external stimulus (for example, an event triggered by the designer).

Acknowledgements

PFR is funded by a University of York, Department of Chemistry Teaching Studentship.

References

Faulconbridge, A. (2011). *RBN-World: Sub-Symbolic Artificial Chemistry for Artificial Life*. PhD thesis, University of York.

Faulconbridge, A., Stepney, S., Miller, J. F., and Caves, L. S. D. (2011). RBN-World: A sub-symbolic artificial chemistry. In *ECAL 2009, Budapest, Hungary*, volume 5777 of *LNCS*, pages 377–384. Springer.

Faulkner, P., Krastev, M., Sebald, A., and Stepney, S. (2018). Sub-Symbolic artificial chemistries. In Stepney, S. and Adamatzky, A., editors, *Inspired by Nature*, pages 287–322. Springer.

Faulkner, P., Sebald, A., and Stepney, S. (2016). Jordan algebra AChems: exploiting mathematical richness for open ended design. In *ALife 2016, Cancun, Mexico*, pages 582–589. MIT Press.

Faulkner, P., Sebald, A., and Stepney, S. (2017). Tuning Jordan algebra artificial chemistries with probability spawning functions. In *ECAL 2017, Lyon, France, September 2017*, pages 497–504. MIT Press.

Hoverd, T. and Stepney, S. (2009). Environment orientation: an architecture for simulating complex systems. In *Proceedings of the 2009 Workshop on Complex Systems Modelling and Simulation*, pages 67–82. pdfs.semanticscholar.org.

Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artif. Life*, 13(1):11–30.

Liu, Y. (2018). The artificial ecosystem: number soup (part II). *ArXiv e-prints*.

Madina, D., Ono, N., and Ikegami, T. (2003). Cellular evolution in a 3D lattice artificial chemistry. In *Advances in Artificial Life, ECAL 2003*, volume 2801 of *LNCS*, pages 59–68. Springer.

McCrimmon, K. (2006). *A Taste of Jordan Algebras*. Springer.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87*, pages 25–34. ACM.

Sayama, H. (2009). Swarm chemistry. *Artif. Life*, 15(1):105–114.

Sayama, H. (2010). Swarm chemistry evolving. In *ALife 2010, Odense, Denmark*, pages 32–36. MIT Press.

Sayama, H. (2011). Seeking open-ended evolution in swarm chemistry. In *2011 IEEE Symposium on Artificial Life (ALIFE)*, pages 186–193.

Sayama, H. (2018). Complexity, development, and evolution in morphogenetic collective systems. *arXiv preprint arXiv:1801.02086*.

Soula, H. A. (2016). Generalized stochastic simulation algorithm for artificial chemistry. In *ALife 2016, Cancun, Mexico*, pages 590–597. MIT Press.

Vargha, A. and Delaney, H. D. (2000). A critique and improvement of the "CL" common language effect size statistics of McGraw and Wong. *J. Educ. Behav. Stat.*, 25(2):101–132.