EMERGY SYNTHESIS 6:
Theory and Applications of the Emergy Methodology

Proceedings from the Sixth Biennial Emergy Conference,
January 14 – 16, 2010, Gainesville, Florida

Edited by
Mark T. Brown
University of Florida
Gainesville, Florida

Managing Editor
Sharlynn Sweeney
University of Florida
Gainesville, Florida

Associate Editors
Daniel E. Campbell
US EPA
Narragansett, Rhode Island

Shu-Li Huang
National Taipei University
Taipei, Taiwan

Enrique Ortega
State University of Campinas
Campinas, Brazil

Torbjörn Rydberg
Centre for Sustainable Agriculture
Uppsala, Sweden

David Tilley
University of Maryland
College Park, Maryland

Sergio Ulgiati
Parthenope University of Napoli
Napoli, Italy

December 2011

The Center for Environmental Policy
Department of Environmental Engineering Sciences
University of Florida
Gainesville, FL

47

Agent-Based Emergy Analysis: A Lagrangian Model of Energy Memory

Caner Kazanci, John R. Schramski and Ernest W. Tollner

ABSTRACT

Considering energy flow and its accumulation from an individual quanta or energy particle perspective, we describe a discrete, individual-based approach for emergy analysis. Emergy calculations are simulated with an agent-based method where discrete packets of energy are labeled and tracked in time as they flow through system model compartments. As such, to quantify energy memory, each particle has a transformity attribute derived from the premise that emergy tracks process inefficiencies thus generating a single-rule approach for an individual-based emergy computation. This agent- (or individual-) based method provides a way to compute emergy for many different processes, with multiple inputs, outputs, or even cycling systems, without assuming any other rules. We then compare the agent-based outcome with four of the current algebra rules of conventional emergy computation. We observe that the agent-based results agree with all of the current algebra rules with one exception (process by-products have total emergy assigned to each by-product pathway). We discuss potential reconciliations for this discrepancy as well as the overall potential future benefits and possibilities of this approach.

INTRODUCTION

Brown and Herendeen (1996) and Odum (1996) articulated four rules of emergy algebra using various models with increasing complexity to aid their demonstration:

1. All source EMERGY to a process is assigned to the processes' output.
2. Process by-products have the total EMERGY assigned to each by-product pathway.
3. When a pathway splits, the EMERGY is assigned to each 'leg' of the split based on the fraction of total energy on each leg.
4. EMERGY cannot be counted twice within a system.
   a. EMERGY in feedbacks cannot be double counted.
   b. Process by-products, when reunited, cannot be added to equal a sum greater than the source EMERGY from which they were derived.

Regardless of system size and complexity, emergy computation for any system is expected to obey these four rules that we subsequently refer to as the emergy algebra rules. As such, for a given system, computing all emergy values may not be a trivial task. This is especially the case if a system involves feedbacks and/or multiple inputs with different transformities.

In Figure 1, we show the two-compartment system with a single feedback and two inputs with different transformities (1 for S and 10 for F) from Brown and Herendeen (1996). Note that emergy values computed in Figure 1b do not appear to satisfy the first rule for process A, as the source emergy (400+300=700) does not equal the output emergy (460). Even for this relatively simple system, computations simultaneously satisfying Rules 1 and 4a are difficult. To understand the reason for this

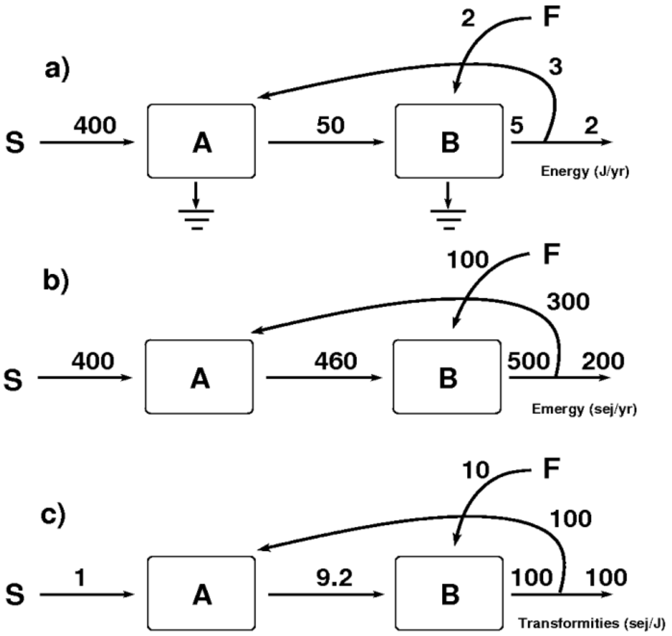


Figure 1. *Emergy flows and transformities in a 2-compartment system with feedback, from Brown and Herendeen (1996). (a) Energy flows, (b) Emergy flows, (c) Transformities.*

difficulty, we focus on the energy flow from process A to process B. This flow contains new energy that entered the system from S and F. It also contains old energy recycled between A and B. Therefore the energy that flows from A to B is not of single homogeneous transformity. Instead, depending on their pathway history, transformities of individual energy quanta will have different transformities. Therefore, the transformity of energy flow from A to B should be computed as the mean of a *transformity distribution* of energy quanta flowing from A to B.

This Lagrangian view motivates us for an agent-based emergy computation simulating individual energy quanta flowing within the system. Each individual energy quanta maintains a transformity, which depends on the processes it has experienced. In theory, such a simulation should enable us to compute the average transformity for all energy flows within the system. Accurate emergy computation regardless of system size and complexity should result, which inherently satisfies all emergy algebra rules.

Although the methodology is based on a simple idea, for complex systems the execution is rather involved. Therefore we will demonstrate various aspects of the methodology on three example systems:

1. One compartment system with single input, output, and dissipation. We will use this example to define and demonstrate the agent-based emergy rule. This single Lagrangian emergy rule effectively reproduces all Eulerian emergy rules except for rule 2. This exception is discussed later.

2. One compartment system with two inputs each with a different transformity, output, and dissipation. This example will show that the agent-based emergy computation agrees with the current methodology for systems with multiple inputs with various and potentially different transformities.

3. Two compartment system with single input, feedback, output, and dissipation. We will use an agent-based method to simulate the energy flow within the system. We then compute the emergy values based on the transformities of individual energy quanta. These values satisfy all emergy algebra rules except for rule 2 discussed later.

METHODS AND RESULTS

Agent-based emergy analysis works by separating energy into identical agents, which we refer to as quanta, or energy particles. For clarity and consistency, we will use the term *particle* to refer to energy particle, quantum, agent or individual. The size of a particle is defined as its energy content, which is user-selectable. Particle size is constant for all particles throughout the simulation, and should be small enough that any energy flow within the system can be represented by an integer number of particles. For example, if particle size is 2 joules, then 100J/day of energy flow can be represented by the movement of 50 particles over one day. Agent-based simulation software needs to be used to simulate the movement of particles.

Emergy is computed by multiplying the amount of energy with the appropriate transformity. Therefore, in this case, each particle carries a unique transformity value, which may increase after the particle completes a process. For agent-based emergy analysis to be successful, we need an agent-based rule that governs how the transformity of a particle is modified by a process. We can compute emergy values for a system when we have both the energy input and energy output of all processes and all associated transformities. However, a particle acquires or maintains no knowledge of the entire system. The only information a particle carries is its own transformity. For example, it has no knowledge of the transformity of a particle that enters a process before or after itself. This is the reason we call this an agent-based or Lagrangian emergy rule.

Agent-based Emergy Rule: When a particle leaves a process, it multiplies its transformity with the inefficiency of the process. The inefficiency of a process is defined as the multiplicative inverse of its efficiency:

$$Inefficiency = \frac{1}{Efficiency} = \frac{Energy\ in}{Energy\ out} \qquad (1)$$

This definition is motivated by the fact that the energy particles that leave a process carry the memory of the particles that dissipate, which is characterized by the inefficiency. In Figure 2, we demonstrate the agent-based rule using a single compartment model where 3 units of energy enter the system, 2 units dissipate, and 1 unit is output.

Therefore the efficiency of the process is 1/3. Using equation (1) the inefficiency is 3. The transformity of the source energy is 3. According to the agent-based emergy rule, the transformity of the output energy is computed as 3 × 3 = 9. This computation complies with the first rule of emergy algebra, that emergy in equals emergy out. A physical example representing this model would be solar cells converting sunlight to electricity. The efficiency of current solar cells is about 14%, which means that out of 100 particles entering a solar panel, only about 14 particles turn into useable electricity on average (with increased transformity), whereas the remaining 86 get dissipated.

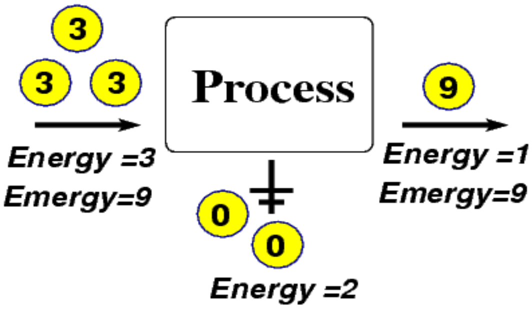


Figure 2. One-compartment system with a single input, output and transformity. Each circle represents an energy particle. The particles are labeled with their transformity. Dissipated energy particles are assigned a zero transformity. Dimensional units are ignored for simplicity.

While the agent-based emergy rule appears to work for the simple system in Figure 2, it quickly gets complicated when a process receives multiple inputs with different transformities, such as the system in Figure 3a.

Here, the transformity of the output is computed as,

$$\frac{Emergy\ out}{Energy\ out} = \frac{Emergy\ in}{Energy\ out} = \frac{18+20}{2} = 19$$

In Figure 3b, we apply the agent-based emergy rule. The inefficiency of the system is

$$Inefficiency = \frac{Energy\ out}{Energy\ in} = \frac{4+6}{2} = 5.$$

Two different energy particles enter the process with transformities of 5 and 3. If an energy particle with input transformity 3 exits the system, its output transformity will be 3 × 5 = 15. If an energy particle with input transformity 5 leaves the system, its output transformity will be 5 × 5 = 25. Neither of these values are 19 as computed by the emergy algebra. However, 40% and 60% of the particles entering the system have transformities of 5 and 3, respectively. Since these percentages remain the same for the process output, the average output transformity computed using the agent-based emergy rule is,

$$15\ \times .6 + 25\ \times .4 = 19,$$

which agrees with conventional emergy algebra. Both methods (conventional and agent-based) applied to both systems satisfy the pertinent emergy algebra rules. Therefore we have not observed an advantage of the agent-based emergy analysis. In fact, it is actually cumbersome to use. However, consider that the introduction shows a model in Figure 1 where the emergy computation does not appear to satisfy rule 1 of the emergy algebra where in Figure 1b, 400 + 100 ≠ 460. To explore this further, first we use the agent-based emergy computation for this type of model.

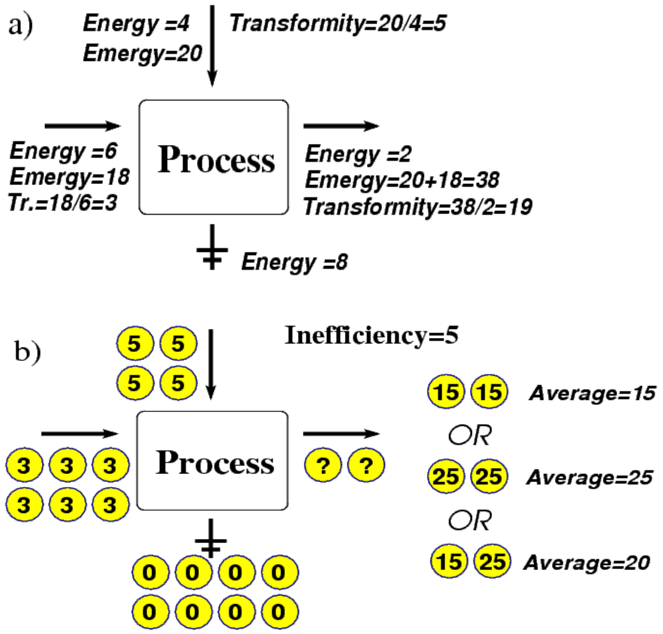


Figure 3. *One-compartment system with two inputs each with different transformities (3 and 5). Each circle represents an energy particle with a single unit of energy. (a) Transformity and emergy is computed using the four rules of emergy algebra. (b) Application of the agent-based emergy rule.*

For clarity of presentation, we simplify the numbers used in the model in Figure 1 to the model shown in Figure 4. Figure 4 shows the computed emergy and transformity values using the traditional emergy algebra. The emergy entering process A is 90 + 36 = 126, while the emergy leaving process A is 90 (Figure 4b). Before we use the agent-based emergy rule to re-compute transformity and associated emergy values, we need to step-back and discuss agent-based simulations.

An agent-based simulation is an application-specific, iterative algorithm that works by applying rules on agents changing their states according to predefined rules. Agents are application-specific entities that generally exist in high quantities in a system. Perhaps the best way to understand an agent-based model is by example. A hypothetical model for the spread of a non-fatal viral disease may be modeled as follows. First, we define agents as people. Each individual has a state of susceptibility, immune or sick. We define appropriate rules that affect individuals, like how they move (e.g., stay at home, use public transportation, work with others in an office), how often they meet, how likely is the disease transmitted to another individual by any of these relations, what is the incubation period, etc. Once the rules are established, the simulation progresses by iteratively applying these rules on all individuals in the system. During the simulation, a user observes that the people interact with each other, some healthy individuals get sick, then they get help, etc. A well-designed simulation provides an accurate prediction of how the disease spreads. Effectiveness of preventative measures, such as quarantine, can be studied using agent-based methods.

In agent-based emergy methodology, the agents are energy particles. The state of an energy particle is its transformity, which is updated as the particle passes through a process (multiplied by the inefficiency of the process). If they dissipate, their transformity is set to zero. The movements of the energy particles within the system (in and out of processes) are managed by a numerical algorithm compatible with the energy flows. For example, if the energy flow from A to B is 150 J/day and the particle represents 2 joules of energy, then the simulation will move 75 particles from process A to process B on average over one day. Various software are available for such simulation (e.g., Netlogo, Arena, and Swarm). Network Particle Tracking (NPT) (Kazanci et al., 2009, Tollner et al., 2009) was used in this study to simulate the system in Figure 4.

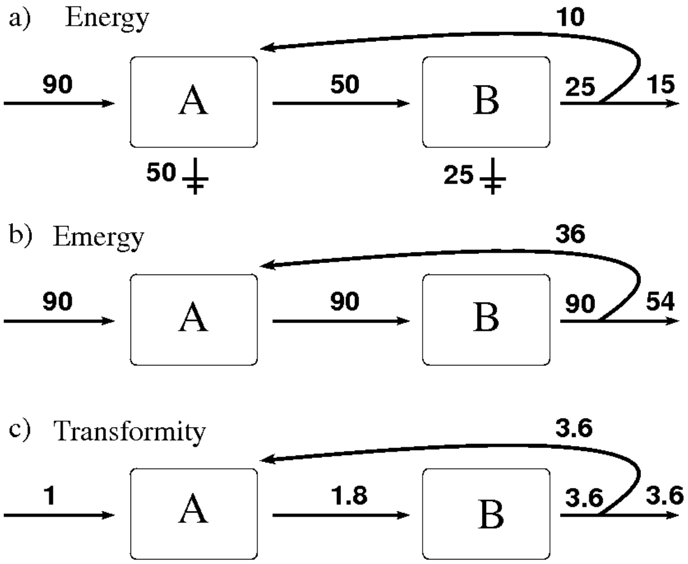


Figure 4: *Two-compartment system with single input and feedback, similar to the system in Figure 1. a) Energy, b) Emergy, and c) Transformity values are computed using the traditional emergy algebra (Brown and Herendeen 1996).*

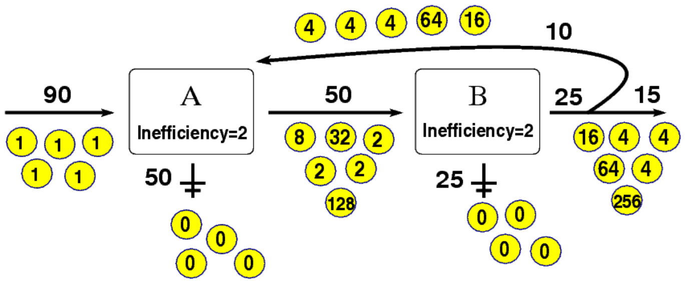
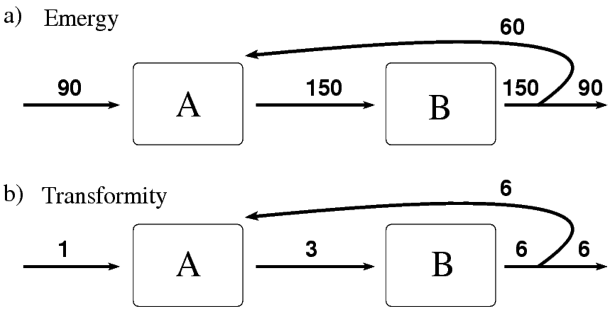


Figure 5. *Two-compartment system with internal feedback, illustrating how the transformities of individual particles change as they flow in and out of processes A and B. The transformity values presented in this figure are examples only, and do not accurately represent the actual distribution of possible transformity values, which is shown in Figure 7.*


a) Emergy

b) Transformity

Figure 6. *a) Emergy and b) transformity values as computed using the agent-based emergy methodology for the system in Figure 4.*

In Figure 5, we illustrate how an agent-based simulation works for the system in Figure 4. All particles entering system have a transformity of 1. Using equation (1), the inefficiencies of both process A and B are (90+10)/50=50/25=2. Therefore when a particle leaves process A towards B, its transformity is equal to, or larger than 2. Similarly, when a particle leaves process B, its transformity is at least equal to 4. These values are different than shown in Figure 4. Some of the output of process B is subsequently input back into process A through the feedback mechanism. These particles will either dissipate or flow back to process B, in which case, their transformity will then be 8. If we let the system progress for a period of time, we may get to observe some particles that cycle many times between process A and B. The transformity of such particles will be high; but the number of such particles will be relatively low as shown in Figure 7. We compute the agent-based emergy from A to B by finding the mean transformity of the particles that flow from A to B. In Figure 6, we show the emergy and transformity of all flows in this system computed using the agent-based emergy methodology. Note that these values satisfy all pertinent emergy algebra rules.

Agent-based emergy computation relies on NPT, which is a stochastic agent-based method based on Gillespie algorithm (Gillespie 1977). This brings out the issue of computational feasibility and accuracy. The simulation for Figure 7 ran for ten minutes on a dual-core 3 GHz desktop computer. The figure shows the transformities of energy particles leaving Process 2. For example, around 1.3 million particles with transformity 4 and 1 energy particle with transformity 16,384 were observed. If the simulation progresses longer, the number of particles observed for each transformity will increase. However, the shape of the histogram is expected to stay the same. The y-axis is logarithmic scale, so the histogram is likely to converge to an exponential distribution as the simulation time increases. The mean of this histogram is 5.98 ≈ 6, which is the transformity for the energy leaving process B shown in Figure 6b.

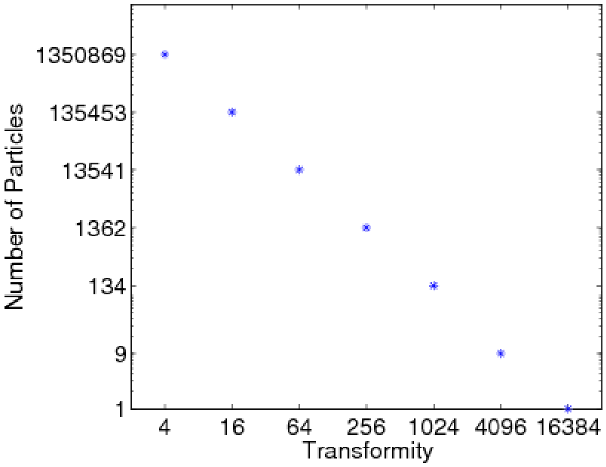


Figure 7. *Histogram of particles with different transformities for the system depicted in Figure 4. This system was simulated using Network Particle Tracking (NPT).*

The accuracy of the computed transformities will increase with the number of particles used during the simulation. The cost, as with other numerical methods, is the computation time. In our experience, the computation time has always been in the order of minutes on a common desktop computer. However, the computation time will increase with system size and complexity, which also brings the issue of scalability. NPT fares well in this regard. We have run a simulation for a model with 3000 compartments involving 9 billion particles. The very fact that we were able to run a simulation for such a model confirms its applicability to large and complex systems. The computing time will be less of an issue in near future, with the current technological advances in hardware and software.

DISCUSSION AND FUTURE WORK

We describe a new methodology for computing the transformity and emergy values for energy flow systems. Constructed from an agent-based simulation with only one rule (when a particle leaves a process, it multiplies its transformity with the inefficiency of the process), the methodology has both advantages and disadvantages. Intuitively, the premise of tracking the inefficiencies of individual energy quanta resonates well with the concept of embodied energy. A one-rule system is conceptually easier to understand and the agent-based results agree with the pertinent emergy algebra rules (Odum 1996, Brown and Herendeen 1996). However, without a direct algebraic formulation, only a full simulation model execution will compute the associated emergies. This may be impractical for researchers who are not familiar with the agent-based algorithms.

Various strengths and weaknesses aside, the principle contribution of this new methodology is its capability to compute the transformity and emergy values accurately for an energy-flow system regardless of size and complexity. Currently we have no knowledge of an alternative methodology with this capability; therefore we hope this may be a helpful development for Emergy Analysis. Although simulations may require an extended period of time for very large systems, this impracticality is mostly mitigated by the ever-increasing affordability of powerful computers. Yet, this brings out an important question for future research; can we develop a methodology that generates the same results without the need to run a simulation? The answer is currently ‘yes’ for small systems. The linear shape of the histogram shown in Figure 7 suggests we can compute the mean transformity as the sum of a converging power series. However, this non-simulation based computation will quickly become intractable as we add more compartments and feedbacks. However, the agent-based methodology will work with same accuracy and efficiency regardless of the number of feedbacks or compartments involved.

Agent-based emergy analysis appears to improve or provide clarification of the results associated with rule 1 (emergy is conserved) of the conventional emergy algebra. Figure 4b represents an example of the conventional scaling discrepancy of emergy conservation. If we consider processes A and B together (including the feedback cycle), emergy in (90) does not equal to emergy out (54). If we consider a smaller hierarchal scale, for example process A, then emergy in (90 + 36) does not equal emergy out (90). This hierarchal discrepancy is also evident in the Brown and Herendeen (1996) example of Figure 1b. However, agent-based emergy analysis appears to generate emergy conservation results that are accurate at all scales. Consider the agent-based calculation shown in Figure 6 of the system from Figure 4. The conserved emergy results in Figure 6a are accurate at all scales. If we consider process A, emergy in (150) equals emergy out (150). This is true for process B as well. If both processes A and B are combined, emergy in (90) equals emergy out (90).

Concerning the emergy algebra's Rule 2, none of the three systems that we used to demonstrate the agent-based methodology contains by-products. Essentially, by-products are treated differently than splits; therefore the agent-based methodology needs an additional programmed mechanism that accommodates this distinction. In addition to a non-simulation-based methodology, appropriate treatment of by-products is one area planned for future work.

REFERENCES

Brown, M.T. and Herendeen, R.A. 1996. Embodied energy analysis and emergy analysis: a comparative view. *Ecological Economics*, 19(3):219–235.

Gillespie, D.T. 1977. Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry*, 81(25): 2340–2361.

Herendeen, R.A. 1989. Energy intensity, residence time, exergy, and ascendency in dynamic ecosystems. *Ecological Modelling*, 48(1):19–44.

Kazanci, C., Matamba, l. and Tollner, E.W. 2009. Cycling in ecosystems: An individual-based approach. *Ecological Modelling*, 220(21):2908–2914.

Odum, H.T., 1996. Environmental Accounting: EMERGY and Decision Making. John Wiley, New York, NY.

Tollner, E.W., Schramski, J.R., Kazanci, C. and Patten, B.C. 2009. Implications of network particle tracking (NPT) for ecological model interpretation. *Ecological Modelling*, 220 (16):1904–1912.