



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Co-evolutionary analysis: a policy exploration method for system dynamics models
Author(s)	Hongliang, Liu; Howley, Enda; Duggan, Jim
Publication Date	2012-10-22
Publication Information	Liu, HL,Howley, E,Duggan, J (2012) 'Co-evolutionary analysis: a policy exploration method for system dynamics models'. System Dynamics Review, 28 :361-369.
Publisher	System Dynamics Society / Wiley
Link to publisher's version	<a href="http://dx.doi.org/10.1002/sdr.1482">http://dx.doi.org/10.1002/sdr.1482</a>
Item record	<a href="http://hdl.handle.net/10379/3932">http://hdl.handle.net/10379/3932</a>
DOI	<a href="http://dx.doi.org/DOI%2010.1002/sdr.1482">http://dx.doi.org/DOI 10.1002/sdr.1482</a>

Downloaded 2024-04-17T21:16:59Z

Some rights reserved. For more information, please see the item record link above.



# Co-evolutionary Analysis: A Policy Exploration Method for System Dynamics Models

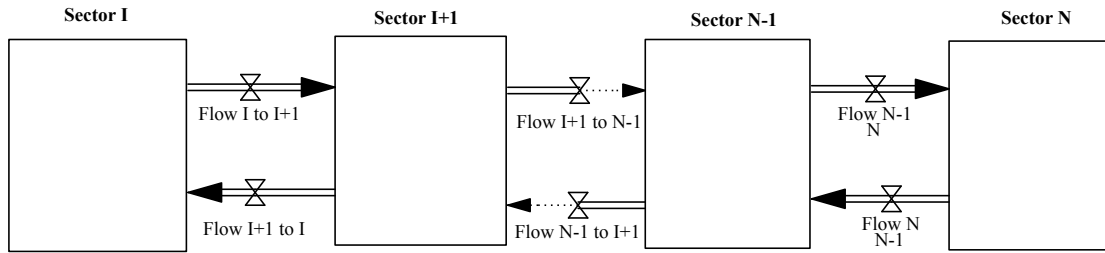
## Abstract

In system dynamics (SD), complex nonlinear systems can generate a wide range of possible behaviours that frequently require search and optimization algorithms in order to explore optimal policies. Within the SD literature, the conventional approach to optimization is the formulation of a single objective function, with a targeted parameter list, and the entire model is simulated repeatedly in order to arrive at optimum values. However, many sector-based SD models contain heuristics of ‘intended rationality’, and a desired outcome for modellers to be able to explore the policy implications of locally optimal behaviours. This can now be achieved through a method known as coevolution, which allows modelers to divide an unsolved problem into constituent parts, where each part can be solved with respect to its own fitness function. In this paper, we specify a solution for evolving locally rational strategies across a multi-sector SD structure. Using the beer distribution game (BDG) as an illustration, we demonstrate the utility of this approach in terms of the impact of two different order management strategies on the policy space of the BDG.

## Introduction

Complex nonlinear systems can generate a wide range of possible behaviours, and developing insights into the dynamics of a complex system has often been difficult (Sterman 2000). A key aspect that drives model complexity are those interactions and feedbacks that occur across organisational boundaries, for example, the production-distribution system models in industrial dynamics (Forrester 1961), the market growth model (Forrester 1968; Morecroft 1985), and the body of empirical and experimental work on the beer distribution game (BDG) (Mosekilde and Larsen 1988; Mosekilde and Laugesen 2007; Sterman 1989; Strozzi, Bosch, and Zaldívar 2007). In terms of policy analysis, these nonlinear and feedback-rich characteristics frequently surpass the capabilities of traditional analytical approaches, and therefore there is a clear need for automated and efficient search algorithms to support policy analysis (Yücel and Barlas 2011). Recent examples of computer science and technology based additions to the SD literature include: a pattern-based approach utilising qualitative features of a desired behavior pattern (Yücel and Barlas 2011); a policy design method for system dynamics models based on recurrent neural networks Chen, Tu, and Jeng (2011); and statistical screening which helps identify high-leverage model parameters and structures for further analysis (Taylor, Ford, and Ford 2010).

A common feature of inter-organisation structures is that they are made up of multiple actors who must coordinate a diverse set of decision policies. The resulting SD models contain multiple decision rules, which involve sensing the state of the system, and combining this with goal-based information cues, in order to control the rate of flows. Furthermore, SD models such as the BDG and the market growth model contain independent actors whose decisions are *intendedly rational* – i.e. their decisions “... produce reasonable and sensible results if the actual environment were as simple as the decision maker assumes it to be” (Sterman 2000). In summary, many of these multi-sector models can be viewed as an interconnected system of physical and information flows, where each sector is autonomous, in the sense of making its own decision based on information sources, yet these decisions can also impact on the future decisions of its neighbours (see Figure 1).



**Figure 1: A Generic structure for SD multi-sector models**

Within this context, a key requirement for modellers is to have access to efficient and reliable methods for thoroughly exploring a model's policy space. The SD optimization approach focuses on the attainment of an overall optimal value across an entire model, for example, finding the minimum costs for actors in the BDG. However, the coevolutionary approach offers an additional, powerful dimension to policy exploration, that can be viewed as a computational extension of the ideas of partial model testing (Morecroft 1985). The distinction from normal optimization methods is that with coevolutionary optimization, individual sectors in the model can be optimized to their own fitness functions, and because of this, a fuller range of policy responses can be investigated.

Coevolutionary algorithms usually employ genetic algorithms (GA) to model the evolution of each species. GAs are inspired by Darwin's theory of evolution and were developed by (Holland 1992) and have been applied to a range of SD models (Duggan 2008; Grossman 2002). The key operators for the GA are selection, crossover, and mutation, and these transform the solution information that is stored in a 'chromosome'. Therefore each chromosome has a fitness that captures the overall solution quality of the SD model.

As part of the optimization process, the *selection operator* is used to select two solutions (parents) from the population, using methods such as roulette wheel selection or rank selection. The *crossover operator* selects "genes" from parent chromosomes and creates two new offspring. It involves firstly randomly choosing some crossover point and then swapping the values according to this point. Finally, the *mutation operator* changes randomly the values of offspring, in order to promote diversity in the overall solution space. These models of evolutionary processes are found to be effective analogues of economic agent strategic learning (Teshfatsion 2002). A strategy can be represented as a chromosome, and the GA processes are models of learning. In the GA, the reproduction operator can be interpreted as learning by imitation, the crossover operator can be interpreted as learning through communication, and the mutation operator is interpreted as learning by experiment (Riechmann 2001).

A coevolutionary algorithm is a model that can be sub-divided into inter-related sectors with either individual or shared fitness functions. As each sector only encodes part of the solution information, the fitness evaluation procedure involves choosing representatives from other sectors to form a collaboration (i.e. a complete solution). This maps very well onto the requirements SD models of organisations, as we can combine a number of *intendedly rational* actors in order to form a complete model. After selecting the representatives, the agent's fitness can be determined through simulating this shared domain model. There are many methods to choose representatives, such as choosing the current best individual agent from each species to be the representative, or randomly selecting an individual agent from each species to be the representative (Potter and De Jong 2000).

The algorithmic design for the coevolutionary approach is shown in Figure 2. It shows that when an overall SD model can be disaggregated into a number of *sectors*, these form the building blocks for creating a *shared domain model*. This shared domain model is a combination of individuals from each sector, where each individual will have a different set of parameter values for key decision heuristics. The solution process is the conventional simulation by repeated optimization (Coyle 1996), and the key advantage of this approach is that individual sectors can also have their own fitness functions, which increases the exploration space for policy analysis. In summary, the contribution of our approach is two-fold. First, we provide an architecture to demonstrate the potential for the use of coevolution to explore the policy space in SD models. Second, we use this method to formally examine, and experiment with, the differences between individual and group oriented behaviors in the BDG.

```

// Create M Sector populations
for Sector  $s \leftarrow 1$  to Sector  $M$  do
    Initialise  $N$  sector individuals;
end for

for  $i \leftarrow 1$  to Max_Generations  $M$  do
    // Measure each sector's fitness for all individuals
    for Sector  $s \leftarrow 1$  to Sector  $M$  do
        for Individual  $k \leftarrow 1$  to Individual  $N$  do
            Select representatives from other sectors;
            Form new collaboration;
            Simulate shared domain model;
            Evaluate fitness (through objective function);
        end for
    end for
    // Evolve each sector population through the GA
    for Sector  $s \leftarrow 1$  to Sector  $M$  do
        Apply selection operator;
        Apply crossover operator;
        Apply mutation operator;
    end for
end for

```

**Figure 2: The coevolutionary algorithm for multi-sector optimisation**

## A Coevolutionary Approach for the Beer Distribution Game

The BDG game (Croson and Donohue 2006; Forrester 1961; Sterman 1989, 2010) offers a simplified implementation of common real world production and distribution systems, where each participant has control and responsibility for its own inventory, and individual sectors strive to maintain their inventory levels as low as possible, while also avoiding out-of-inventory conditions which cause backlogs. There are two main flow channels: information and physical goods. Orders originate at the customer, and flow upstream from sector to sector. Shipments travel downstream and represent the fulfillment of orders. The presence of delays, multiple sectors and potential stock-outs increase the complexity of inventory management challenges for all sectors.

In the BDG, repeated interactions between sectors are dynamic and can be viewed as coevolution processes. For example, a sector may find a better strategy during the play, which might affect the performance of other sectors. Subsequently, other sectors may also decide to change their strategies in order to improve performance, which in turn may affect other sectors and lead them to try alternative strategies. Each strategy's performance or fitness can be evaluated by returns generated such as profits or costs. Over successive generations each sector will select the fittest strategies and use these to evolve new strategies that will replace the least fit strategies. Furthermore, it is also possible that all sectors in the supply chain have a common goal to minimise the total costs.

In using the BDG to validate the coevolutionary approach, and in line with previous research (Sterman (1989)), we apply the anchoring and adjustment heuristic for the ordering policy. The ordering decision equation is:

$$O_t = \text{MAX}(0, \hat{L}_t + \alpha (S^* - S_t - \beta SL_t))$$
$$\hat{L}_t = \theta \hat{L}_{t-1} + (1-\theta) \hat{L}_{t-1}$$

Where:

- $O_t$ , the order rate for supplies, which is anchored on the expected demand;
- $\hat{L}_t$ , the expected loss rate (or demand forecast), modeled by the exponential smoothing of incoming orders;
- $SL_t$ , the number of goods in the supply line.
- $S_t$ , the number of goods currently in stock.

The associated parameters for each sector include:

- $\alpha$ , the fraction of the inventory shortfall or surplus ordered each period, which is the adjustment rate that parameter is in the range  $(0 \leq \alpha \leq 1)$ ;
- $\beta$ , which models the fraction of the supply line taken into account when ordering, and is in the range  $(0 \leq \beta \leq 1)$ ;
- $\theta$  is used in the calculation of the expected demand, captures the weight assigned to the most recent observation, and this parameter is also in the range  $(0 \leq \theta \leq 1)$ ;
- $S^*$  refers to the target value of the effective stock (desired stock and desired supply line) that decision makers should maintain.

The parameters ( $\alpha$ ,  $\beta$ ,  $\theta$ ,  $S^*$ ) are used to represent decision makers' strategies, and these can be tested using two different decision approaches:

1. *Individually Oriented Strategies (IOS)*, which allows us to test outcomes that are evaluated solely against the target of an individual sector's costs. For this, each agent chooses to minimise their own cost regardless of the whole supply chain cost in the beer game, and therefore all agents are intendedly rational.
2. *Group Oriented Strategies (GOS)*, to analyze the outcome whereby fitnesses are determined based on collective supply chain costs. This models an approach in which all agents cooperate to achieve a common goal.

In order to facilitate this requirement, two separate objective functions are defined, and these variables are summed over H weeks of the simulation. These fitness functions can be either individually-oriented (1) or group oriented (2).

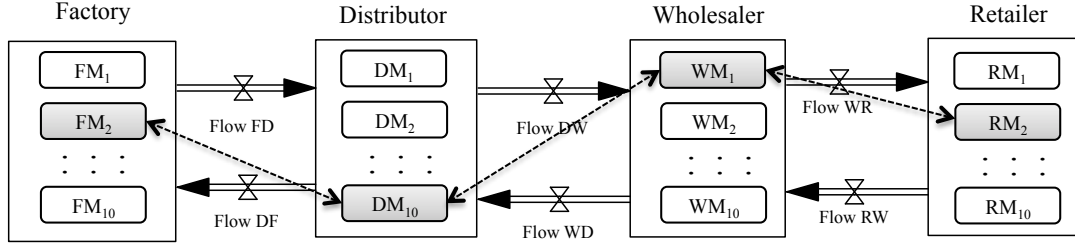
$$COST_{SECTOR} = \sum^H (0.5 * INV_{SECTOR} + 1.0 * B'LOG_{SECTOR}) \quad (1)$$

$$COST_{ALL} = \sum (COST_{SECTOR}) \quad (2)$$

For the model, the cost of inventory holding is 0.5 for each case of beer per week, and the cost of backlogs is 1.0 for each case of beer per week. The key stages of the coevolutionary approach for the BDG are:

1. Initialise the four sectors (Factory, Distributor, Wholesaler and Retailer), and create 10 individual models for each sector (e.g. FM<sub>1</sub>.. FM<sub>10</sub> for the Factory sector). Each of these models will have a randomly allocated value for  $\alpha$ ,  $\beta$ ,  $\theta$ , and  $S^*$ .
2. Iterate until the number of generations equals *Max\_Generations*
  - a. Iterate through each sector in sequence
    - i. For each individual (e.g. FM<sub>2</sub>), select participants from other sectors (e.g. DM<sub>10</sub>, WM<sub>1</sub> and RM<sub>2</sub>) to form a shared model (see Figure 3). In the design, in order to obtain the average performance, we allow each individual from a sector to interact with all participants from other sectors. This forms 10<sup>3</sup> (1000) shared models.
    - ii. Simulate all shared models. The fitness value for the individual is the average fitness value over 10<sup>3</sup> interactions.
  - b. Evolve the solution population by iterating through each sector in sequence, and for each individual:
    - i. Apply selection operator
    - ii. Apply crossover operator
    - iii. Apply mutation operator

The results from this algorithm yield interesting and novel insights into the policy space for the BDG, and these are now presented.



**Figure 3: Creating a shared model for each simulation iteration**

### Policy Exploration using the Coevolutionary Approach

For experimental runs across these two strategy scenarios, the following values are used:

- Total simulation length is 50 weeks.
- Parameters  $\alpha$ ,  $\beta$  and  $\theta$  values are all in the range of  $[0,1]$ .
- The  $S^*$  value is in the range  $[0, 50]$ , where the shipment delay is 2 weeks, and the ordering delay is also 2 weeks.
- The customer demand is initially four cases per week and increases to eight cases per week in week 5 and remains at that level thereafter.
- Each population size  $N$  is set to 10 and the total number of iterations (generations) is 150. Typically either a larger population or a greater number of generations can lead to a better result, as more of the solution space is searched. However, a larger population or a greater generation number also increases the computational cost, and it is recommended that the generation number is much bigger than the population number. For GAs, the population size is usually set in range of 50 to 100, and the generation number is usually set in  $[200, 2000]$ . However, as we are using a coevolutionary algorithm, which requires much more computation resources, and because of this we use smaller values.
- The value for  $M$  (number of “species”) is set to 4, as this represents the number of actors in the BDG.
- A real value encoding method to construct the solution “chromosomes”. Each chromosome is a sequence of real values ( $\alpha$ ,  $\beta$ ,  $\theta$  and  $S^*$ ).
- The fitness functions are straightforward: for the individual strategy, the local costs are used, for the group strategy, the overall supply chain cost is used.

The following experimental results are from the agents that have the best fitness among each sector's agent population. Furthermore, because of the stochastic nature of the solution process, each result set is averaged from 50 individual runs of the coevolutionary algorithm. Each sector member and overall supply chain costs are shown in **Table 1**, where all agents are individually or group oriented under the one step demand pattern.

<b>Role</b>	<b>Individually Oriented</b>		<b>Group Oriented</b>	
	$\mu$ ( $\sigma$ )		$\mu$ ( $\sigma$ )	
Retailer (R)	101.93	(23.54)	95.88	(17.69)
Wholesaler (W)	141.13	(25.92)	117.89	(25.46)
Distributor (D)	161.42	(36.16)	124.69	(34.03)
Manufacturer (M)	143.97	(35.54)	98.99	(23.66)
<b>Entire Chain</b>	<b>620.54</b>	<b>(45.37)</b>	<b>521.69</b>	<b>(56.69)</b>

**Table 1: Supply chains costs based on step demand pattern**

Inventory costs increase as one moves further upstream when agents are individually oriented, except that the manufacturer's cost is lower than the distributor's in the one step change demand. The upstream agents get most benefit from group oriented strategies. This highlights a further benefit from the coevolution approach, as the information generated from the optimization process can be utilized to formally test aspects of model behaviour, and confirm whether or not one set of policies consistently outperform another set (see Table 2).

<b>Role</b>	<b>Cost Reduction</b>	<b>t Score</b>	<b>Two-tailed P Value</b>
R	5.93%	1.45	0.15
W	16.46%	4.52	< 0.001
D	22.75%	5.23	< 0.001
M	31.24%	7.44	< 0.001
<b>Entire Chain</b>	<b>15.92%</b>	<b>9.62</b>	<b>&lt; 0.001</b>

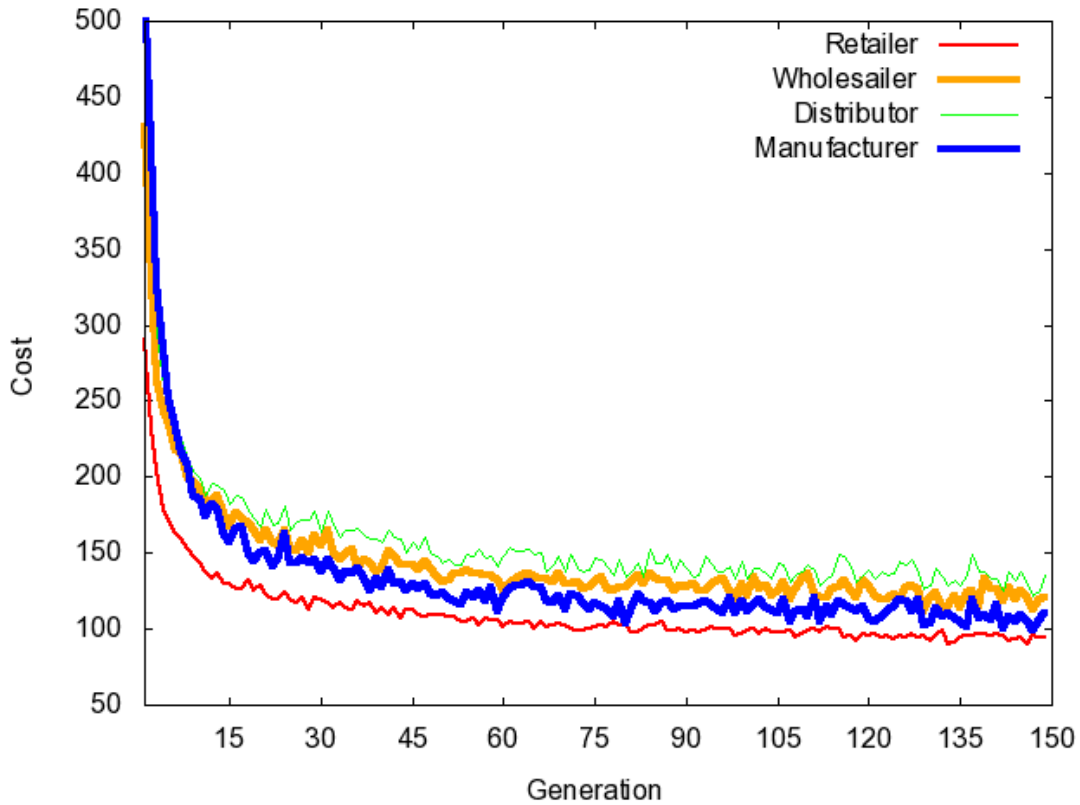
**Table 2: Benchmark of improvements of GOS vs IOS approaches**

The optimization process also generates a wealth of data on the model performance across both fitness functions. For example, Table 3 captures the range of values for the sector parameters ( $\alpha$ ,  $\beta$ ,  $\theta$  and  $S^*$ ) across the two strategies, run across 50 simulations with average ( $\mu$ ) and standard deviation ( $\sigma$ ). The benefit of this analysis is that it supports a greater degree of experimentation when compared with convention optimization approaches that only support the optimization of a single objective function. Figure 4 shows the convergence properties of the algorithm under the *GOS scenario*, whereby the costs of each sector diminish over time before settling at their final values.



Role	$\alpha$		$\beta$		$\theta$		$S^*$	
	IOS	GOS	IOS	GOS	IOS	GOS	IOS	GOS
R	0.56 (0.34)	0.45 (0.33)	0.92 (0.14)	0.85 (0.17)	0.61 (0.30)	0.48 (0.39)	22.8 (3.88)	24.6 (5.09)
W	0.82 (0.20)	0.38 (0.32)	0.94 (0.06)	0.90 (0.13)	0.58 (0.32)	0.49 (0.35)	21.8 (3.08)	27.3 (7.48)
D	0.85 (0.18)	0.70 (0.30)	0.93 (0.05)	0.94 (0.06)	0.67 (0.30)	0.65 (0.30)	20.7 (2.67)	24.3 (6.69)
M	0.75 (0.20)	0.87 (0.12)	0.91 (0.07)	0.89 (0.08)	0.71 (0.20)	0.70 (0.21)	18.9 (1.75)	19.0 (1.72)

**Table 3: Mean (std. dev.) of parameter values in response to a step demand pattern. Individual and Group Oriented Strategies.**



**Figure 4: Convergence of costs for model sectors (GOS)**

## Discussion

Our research can be framed in the context of existing optimization approaches for system dynamics (Coyle 1996; Dangerfield and Roberts 1996), and to studies and analysis of the BDG, for example the work of (Mosekilde and Larsen 1988; Sterman 1989). In particular, we identify two contributions:

First, we propose a new optimization framework that is ideally suited to exploring policy options for inter-organisational models. In particular, where the model comprises the interaction of sectors with distinctly rational decision rules.

The coevolutionary framework allows for individual fitness functions for each sector to be explored, and for the impact of these policies to be compared with scenarios of a single shared fitness function.

Second, we use this method to formally examine, and experiment with, the differences between individual and group oriented behaviors in the BDG. Our overall optimization results compare favourably with earlier published work on the BDG (Sterman 1989). Furthermore, statistical analysis of the model results demonstrate that group oriented strategies improve the overall performance of the supply chain, when compared with individually oriented strategies.

Future work with the coevolutionary framework will include developing a full analysis of SD models such as the market growth model (Forrester 1968), and finalising the optimization component so that it can be used by the wider SD community.

## Acknowledgement

The support of Science Foundation Ireland is gratefully acknowledged.

## References

- Chen Y-T, Tu Y-M, Jeng B. 2011. A Machine Learning Approach to Policy Optimization in System Dynamics Models. *Systems Research and Behavioral Science* **28**(4): 369-390.
- Coyle R. 1996. *System Dynamics: A Practical Approach*. Chapman and Hall: London.
- Crosby R, Donohue K. 2006. Behavioral Causes of the Bullwhip Effect and the Observed Value of Inventory Information. *Management Science* **52**(3): 336.
- Dangerfield B, Roberts C. 1996. An Overview of Strategy and Tactics in System Dynamics Optimization. *The Journal of the Operational Research Society* **47**(3): 405-423.
- Duggan J. 2008. Equation-based policy optimization for agent-oriented system dynamics models. *System Dynamics Review* **24**(1): 97-118.
- Forrester JW. 1961. *Industrial Dynamics*. Pegasus Communications: Waltham MA.
- Forrester JW. 1968. Market growth as influenced by capital investment. *Industrial Management Review* **9**(2): 83-105.
- Grossman B (2002) Policy Optimization in Dynamic Models with Genetic Algorithms. *Proceedings of the 20th International Conference of the System Dynamics Society*. Palermo, Italy.
- Holland JH. 1992. *Adaptation in natural and artificial systems*. MIT Press.
- Morecroft JDW. 1985. Rationality in the Analysis of Behavioral Simulation Models. *Management Science* **31**(7): 900-916.
- Mosekilde E, Larsen ER. 1988. Deterministic chaos in the beer production-distribution model. *System Dynamics Review* **4**(1-2): 131-147.
- Mosekilde E, Laugesen JL. 2007. Nonlinear dynamic phenomena in the beer model. *System Dynamics Review* **23**(2-3): 229-252.
- Potter MA, De Jong KA. 2000. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evol. Comput.* **8**(1): 1-29.
- Riechmann T. 2001. Genetic algorithm learning and evolutionary games. *Journal of Economic Dynamics and Control* **25**(6-7): 1037.
- Sterman JD. 1989. Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. *Management Science* **35**(3): 331-339.
- Sterman JD. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill: New York.
- Sterman JD. 2010. Does formal system dynamics training improve people's understanding of accumulation? *System Dynamics Review* **26**(4): 316-334.

- Strozzi F, Bosch J, Zaldívar JM. 2007. Beer game order policy optimization under changing customer demand. *Decision Support Systems* **42**(4): 2153-2163.
- Taylor TRB, Ford DN, Ford A. 2010. Improving model understanding using statistical screening. *System Dynamics Review* **26**(1): 73-87.
- Tesfatsion L. 2002. Agent-Based Computational Economics: Growing Economies From the Bottom Up. *Artificial Life* **8**(1): 55-82.
- Yücel G, Barlas Y. 2011. Automated parameter specification in dynamic feedback models based on behavior pattern features. *System Dynamics Review* **27**(2): 195-215.