Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

GAMING SIMULATIONS WITH ENVIRONMENTAL TRAJECTORIES THAT MAXIMIZE INFORMATION GAIN

Gunnar Flötteröd

KTH Royal Institute of Technology
Teknikringen 72
11428 Stockholm, SWEDEN

Sebastiaan Meijer

KTH Royal Institute of Technology
Teknikringen 72
11428 Stockholm, SWEDEN

ABSTRACT

Gaming simulations put real actors in simulated environments. Example applications are training and scenario analysis in transport operations and disaster management. Running a single gaming simulation is an expensive endeavor and therefore must be led through interesting scenario configurations to maximize the learning or research outcomes. This article presents an approach to automatically control the simulated environment in account for the real players' behavior such that a maximum usability of the session is ensured. The approach accesses elements from discrete choice theory and provides the game designer with different options to tailor the type of learning. An archetypical application demonstrates the usefulness of the approach.

1 INTRODUCTION

Gaming simulation is rapidly becoming a regular learning method and increasingly important for research purposes (Bekebrede and Mayer, 2005; De Freitas and Martin, 2006; Kriz and Hense, 2003; Meijer et al, 2006; Meijer, 2009). Gaming simulation, here defined as 'simulating a system through gaming methods' is one of the terms in a loosely demarcated field of interactive participatory activities, aiming to involve participants, who may be the real stakeholders in an activity. Other terms used are simulation game, policy exercise and serious gaming. The word gaming will be used here as the short term for gaming simulation.

In the world of policymaking, there is half a century of history in using gaming as an intervention method. Policies can be formulated in a simulated environment to experience their effects as well as the responses of other stakeholders (Duke, 1974; Duke and Geurts, 2004; Mayer, 2010). Increasingly popular is the possibility to try out the effect of policies on a simulated system and see whether innovation in roles, rules, objectives and constraints can be made. This approach, although very relevant for policymaking, is actually a different use of gaming, namely for testing hypotheses (Peters et al, 1999). This application is less common and puts great emphasis on the verification and validation of the gaming simulation (Klabbers, 2003, 2006; Noy et al, 2006; Meijer, 2009).

Running gaming sessions is a relatively costly endeavor, as it requires (a group of) participants to spent time in usually facilitated sessions, fairly often using modern computers and with a lot of debriefing. There is therefore a need to optimize the learning or research outcomes of a session, and for that, to control the state variables that the player experiences during his time in the game. Tykhonov et al (2009) therefore developed multi agent simulations to select the most interesting variable settings. Dignum et al (2009, 2010) developed multi-agent techniques to take control over the course of action to guarantee interesting game play.

This article introduces another approach to control the game flow, accessing elements from discrete choice theory (Ben-Akiva and Lerman, 1985). Discrete choice theory is the mainstay of behavioral modeling in the transport and marketing domain. The authors set up a framework and explore a simple case study that is based upon the Trust and Tracing Game dilemma (Meijer et al, 2006; Meijer, 2009; Tykhonov et al, 2009). The purpose of the Trust and Tracing Game is to learn about the effect of checking and cheating behavior in supply chains, in a given cultural and institutional setting. The case study in this paper could be useful to learn about strategies to cope with cheating as a buyer, as a subset of the full game.

2 FRAMEWORK

This section makes the approach formally concrete. Some simplifications are adopted for notational convenience; these simplifications are indicated in the text.

2.1 Framework Elements

A gaming simulation consists of three major component that are relevant for the purposes of this text: one or more real actors (decision makers, players); a (computer) simulation of the environment; the facilitator (controls the game and the simulated environment).

The game evolves along a time line. Thinking of discrete time steps (stages), the game adheres to the following logic:

1. An initial environmental state is defined.
2. The following is repeated until the game terminates:
    a. The current environmental state is presented to all actors.
    b. All actors submit their actions for the next stage.
    c. The environmental simulation generates a new stage *under the control of the facilitator.*

The objective of this work is to investigated methods to support the facilitator in step 2c (controlling the simulated environment), such that the game evolves in an insightful way.

2.2 System Dynamics

We number the game stages $k = 0 \ldots K$ and denote the environmental state in stage k by $x^{(k)}$. The length of a stage (in terms of simulated time) is scenario-dependent; it should be long enough to allow for relevant changes in the environment but should be short enough to avoid incoherent, jump-like changes.

There are N actors (decision makers, players). Every actor n has an internal state $z_n^{(k)}$. Given a current environmental state $x^{(k)}$ and an internal state $z_n^{(k)}$, each decision maker n selects an action i from a set of alternatives. We assume that decision makers do not communicate and that their choice sets are finite. Both assumptions can be relaxed. However, assuming real-valued choice sets would leave out of the realm of *discrete* choice theory. The potential combinatorial explosion a discrete choice set brings along can be counteracted with sampling strategies (Flötteröd and Bierlaire, 2011). The environmental simulation reacts to the actions of all players according to its "laws of nature", as described further below.

2.3 Acceptability of Simulated Environment

Given either a stochastic environment or at least two non-communicating players, each single actor will acknowledge that the upcoming environmental state is not perfectly predictable. If the environment is currently in state $x^{(k)}$, its next state appears to player n, who chose action ${i_n}^{(k)}$ in stage k, like a random variable that assumes a concrete value $x^{(k+1)}$ with probability $P(x^{(k+1)} \mid {i_n}^{(k)}, z_n^{(k)}, x^{(k)})$. The conditioning of the next state only on the current state, the action of player n and that player's internal state represents truthfully the state of knowledge of player n.

This gives the facilitator, controlling the game and in particular the environmental simulator, the necessary degrees of freedom to steer the system into relevant regimes: Instead of truthfully generating a stochastic realization of the system dynamics, an *acceptable* state is deliberately selected.

To give an example, an upcoming state may be considered as acceptable if every single player believes that it can occur at least with probability δ. This would mean that the set of feasible next states given that one is currently in state $x^{(k)}$ can be written as $\{x^{(k+1)} \mid \min_n P(x^{(k+1)} \mid i_n{}^{(k)}, z_n^{(k)}, x^{(k)}) \geq \delta\}$. (Operational formulations would be somewhat more complicated, in particular to avoid situations where this set is empty.)

2.4 Objective Decision Support

Denote by X a complete realization of the game, comprising a sequence of environmental states $x^{(0)}, x^{(1)}, \dots, x^{(K)}$ and all internal states $z_n^{(0)}, z_n^{(1)}, \dots, z_n^{(K)}$ and choices $i_n{}^{(0)}, i_n{}^{(1)}, \dots, i_n{}^{(K)}$ for all players $n = 1 \dots N$.

The analyst defines a measure of information $I(X)$ that assigns to each game realization X a real-valued number representing how informative this realization is. An example would be to give all situations where some player n has to make a critical decision in less than 2 minutes an information value of one and zero otherwise. Alternatively, one could define the information value as one over the time left for player n to make a decision, reflecting that short decision time windows reveal more about the player's ability to act under time pressure.

Given a measure of plausibility of the environmental dynamics and a measure of information for complete game realizations, the objective of the decision support system is to propose to the facilitator in each stage of the game an acceptable and most informative next environmental state.

2.5 Tracking of Behavioral Models

As the game evolves through time, the choices of all actors can be observed and related to the situations in which they were made. Specifically, after stage k one has for each player n a data set $(x^{(0)}, z_n^{(0)}, i_n^{(0)}), \dots, (x^{(k)}, z_n^{(k)}, i_n^{(k)})$ of (environmental state, internal state, choice) tuples, from which a behavioral model that mimics player n's behavior can be estimated. As the game advances, more and more data becomes available, based on which the behavioral models can be continuously improved. Train (2003) presents a discrete choice framework for this model update process. Martens and Uhrmacher (2004) tackle the same problem with a rule-based approach. The presented case study does not make use of this possibility, which, however, will be accounted for in future work.

3 CASE STUDY: MODEL SPECIFICATION

Consider a market with two sellers $s = 1,2$ and one buyer. The game proceeds in rounds $k = 0,1, \dots, K$. In each round, each seller offers a good. The seller may decide to offer an intact good (this is called a "fair" offer) or to offer a defect good (this is called an "unfair" offer). The buyer selects one offer in each round of the game, without being able to check for fairness *a priori*. After the round, however, the buyer can check one or both of the offers for a fixed fee. The buyer's objective is to maximize the number of intact goods but to keep the fees paid for checking low.

The objective of this game is to learn how the buyer feels about deception and disappointment. In a real application, the buyer would be a real person. For the purposes of this study, the buyer is replaced by a behavioral model, as described further below. One stage of the game can be summarized as follows:

1. Seller 1 makes an offer (offer 1).
2. Seller 2 makes an offer (offer 2).
3. Buyer selects between offer 1 and 2.
4. Buyer decides if to check offer 1.
5. Buyer decides if to check offer 2.

3.1 Buyer's Decision Rule

The buyer assigns the following utility to the action of buying from seller s in round k:

$$V_{\text{buy}}^{(k)}(s) = \beta_{\text{trust}} \cdot \text{trust}_s^{(k)}$$

where β_{trust} is a real-valued coefficient and $\text{trust}_s^{(k)}$ represent the buyer's (real-valued) trust in seller s in stage k. The probability $P_{\text{buy}}^{(k)}(s)$ that the buyer selects the offer of a seller s is then modeled with a binomial logit model (technically a logistic regression, but based on behavioral theory; see Ben-Akiva & Lerman, 1985):

$$P_{\text{buy}}^{(k)}(s) = \frac{\exp V_{\text{buy}}^{(k)}(s)}{\exp V_{\text{buy}}^{(k)}(1) + \exp V_{\text{buy}}^{(k)}(2)}$$

The utility to check on the offer of seller s in round k is

$$V_{\text{check}}^{(k)}(s) = \gamma_{\text{trust}} \cdot \text{trust}_s^{(k)} + \gamma_{\text{posSurpr}} \cdot \text{posSurpr}_s^{(k)} + \gamma_{\text{negSurpr}} \cdot \text{negSurpr}_s^{(k)} - \text{checkCost}$$

where $\gamma_{\text{trust}}, \gamma_{\text{posSurpr}}, \gamma_{\text{posSurpr}}$ are real-valued coefficients, $\text{posSurpr}_s^{(k)}$ represents a recent (real-valued) positive surprise of the buyer about seller s, $\text{negSurpr}_s^{(k)}$ represents an accordingly negative surprise, and checkCost is the real-valued cost of checking on a player. The probability $P_{\text{check}}^{(k)}(s)$ to check on seller s in time step k is again captured with a binomial logit model:

$$P_{\text{check}}^{(k)}(s) = \frac{\exp V_{\text{check}}^{(k)}(s)}{1 + \exp V_{\text{check}}^{(k)}(s)}.$$

3.2 Buyer's learning Rule for Trust

The experience of the buyer when checking on seller s after round k is defined as

$$\text{experience}_s^{(k)} = \begin{cases} 1 & s \text{ made a fair offer in } k \\ -1 & s \text{ made an unfair offer in } k \end{cases}$$

If the buyer decides to check on a seller s after round k, the buyer updates the trust into that seller as follows:

$$\text{trust}_s^{(k+1)} = (1 - \alpha) \cdot \text{trust}_s^{(k)} + \alpha \cdot \text{experience}_s^{(k)}$$

where α is a real-valued parameter between zero and one. If no check occurs, the trust stays unchanged.

3.3 Buyer's Learning Rule for Surprise

If the buyer decides to check on a seller s after round k, the surprise about that buyer is computed as follows:

$$\begin{aligned} \text{posSurpr}_s^{(k+1)} &= \max\{0,\ \text{experience}_s^{(k)} - \text{trust}_s^{(k)} - 1\} \\ \text{negSurpr}_s^{(k+1)} &= \max\{0,\ \text{trust}_s^{(k)} - \text{experience}_s^{(k)} - 1\} \end{aligned}$$

To interpret these expressions, not that the experience is either +1 or -1 and that the trust is always in the interval $(-1, 1)$. The positive surprise only takes values larger than zero if the difference between experience and trust is larger than one, meaning that the buyer made a +1 experience and previously had a negative trust. Symmetrically, the negative surprise only takes a value larger than one if the difference between trust and experience is larger than one, meaning that the buyer had a positive trust and made a -1 experience.

If a seller is not checked upon, both corresponding surprise values are set to zero.

3.4 Mapping of Example on Formal Framework

The behavioral alternatives of each seller are to make a fair offer or to cheat:

$$i_s^{(k)} \in \{\text{fair,unfair}\}, \qquad s = 1{,}2.$$

Since "fairness" is the only relevant attribute of an offer, the offer of a player is in the following equated to the players' choices of being fair or not.

The environmental state in stage k consists of nothing but the two offers made in this stage, which in turn are characterized by the players' decisions to be fair or not:

$$x^{(k)} = \left[i_1^{(k)}, i_2^{(k)}\right].$$

The internal state of the buyer is characterized by the level of trust in and surprise of either seller:

$$z_{\text{buyer}}^{(k)} = \left[\text{trust}_1^{(k)}, \text{trust}_2^{(k)}, \text{posSurpr}_1^{(k)}, \text{posSurpr}_2^{(k)}, \text{negSurpr}_1^{(k)}, \text{negSurpr}_2^{(k)}\right].$$

Regarding the plausibility of a state, we assume that the buyer considers every possible combinations of actions of the two sellers as plausible. That is, as long as either seller produces an offer, no matter if fair or not, the buyer believes that this is a realistic outcome of the game dynamics.

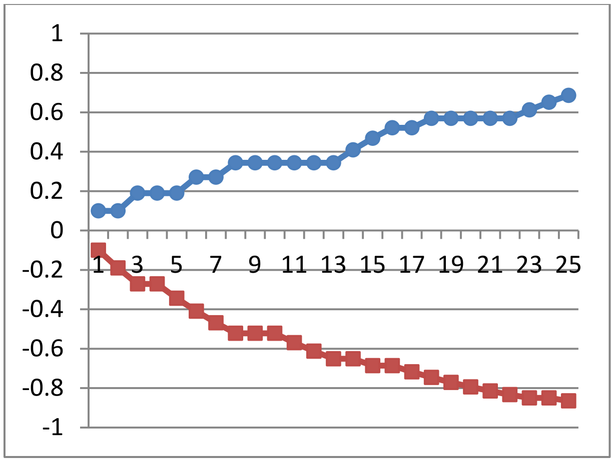
The formal framework allows for more complex environmental dynamics and player actions. The purpose of this work is a first investigation in an utmost simplified setting; more complexity would not provide more insight at this point.

4 CASE STUDY: MODEL EXPLORATION

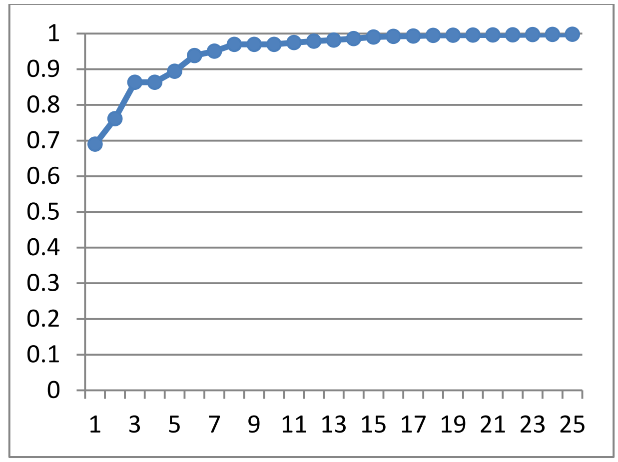
This section provides some intuition on the game dynamics of the case study. The behavioral parameters used here and in the remainder of this article are given in Table 1. The concrete values are arbitrarily selected such that strong and informative system responses are obtained.

Table 1: Behavioral parameter values for case study.

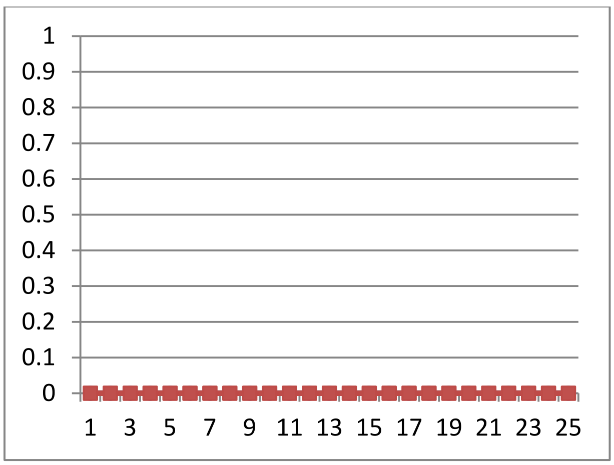
parameter	value	description
β_{trust}	4.0	coefficient for trust in utility of buying
γ_{trust}	-2.0	coefficient for trust in utility of checking
γ_{posSurpr}	2.0	coefficient for positive surprise in utility of checking
γ_{negSurpr}	8.0	coefficient for negative surprise in utility of checking
α	0.1	update weight for experience in trust learning rule
checkCost	0.0	cost of checking on a seller



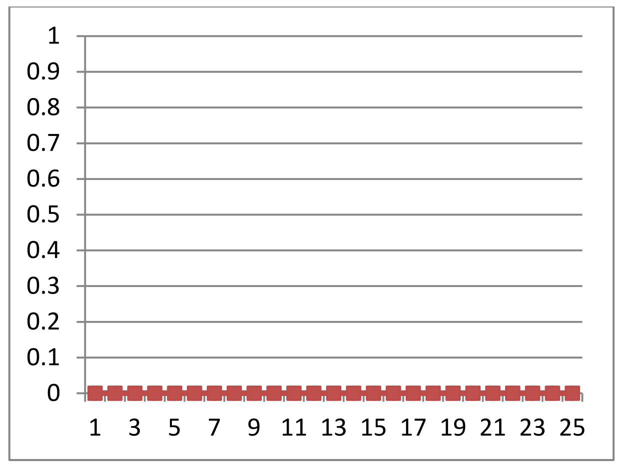
Trust (● = seller1, ■ = seller 2)



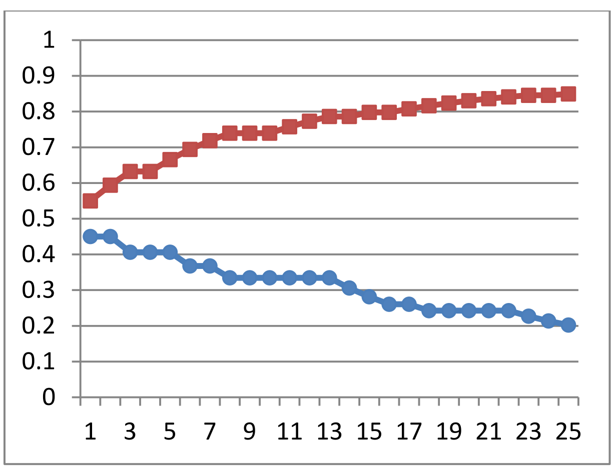
Selection probability for seller 1



Positive surprise (● = seller1, ■ = seller 2)

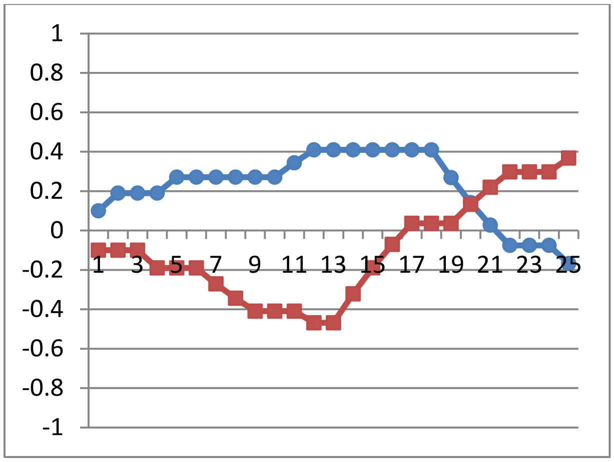


Negative surprise (● = seller1, ■ = seller 2)

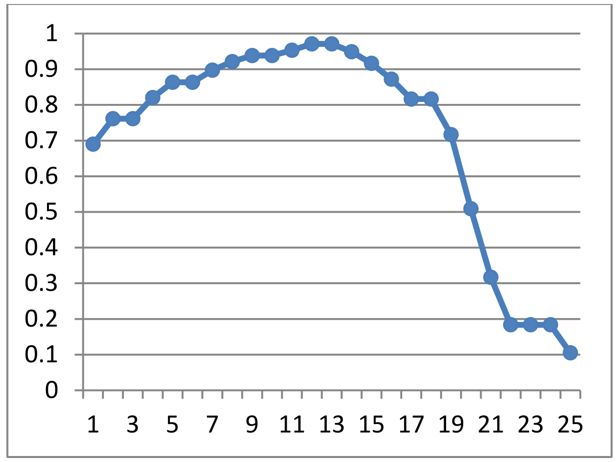


Check probability (● = seller1, ■ = seller 2)

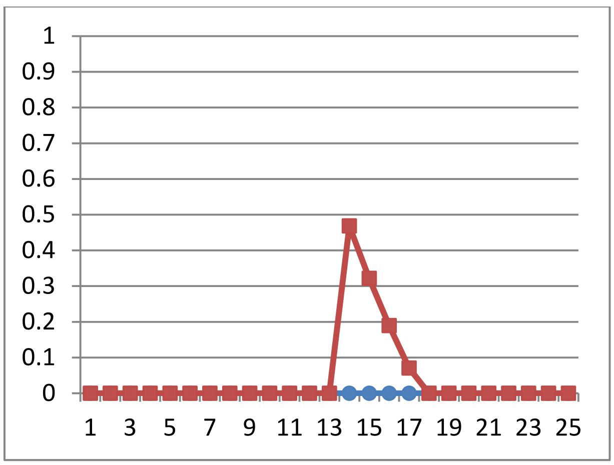
Figure 1: Seller 1 plays always fair; seller 2 plays always unfair.
(Positive and negative surprise values for seller 1 are hidden below those for seller 2.)



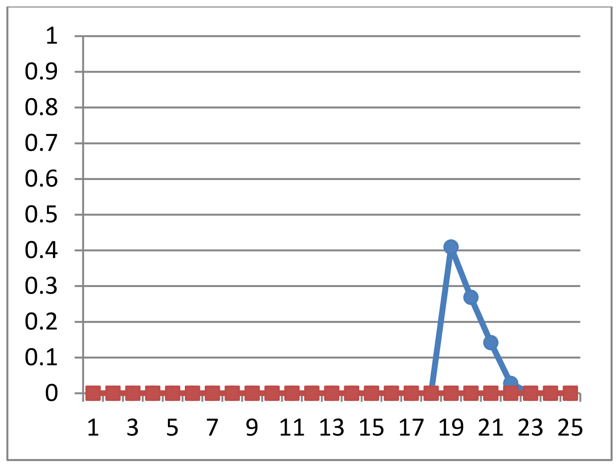
Trust (● = seller1, ■ = seller 2)



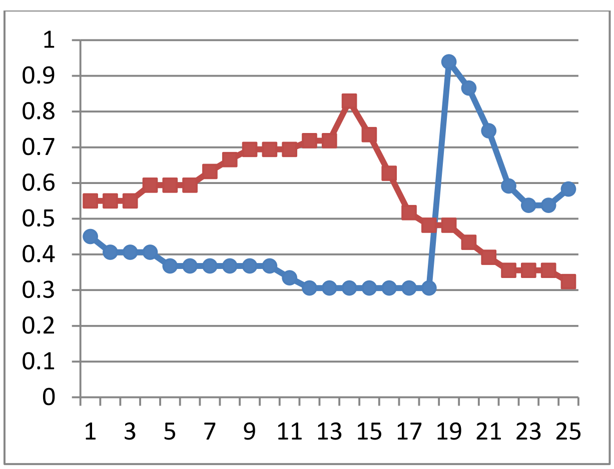
Selection probability for seller 1



Positive surprise (● = seller1, ■ = seller 2)

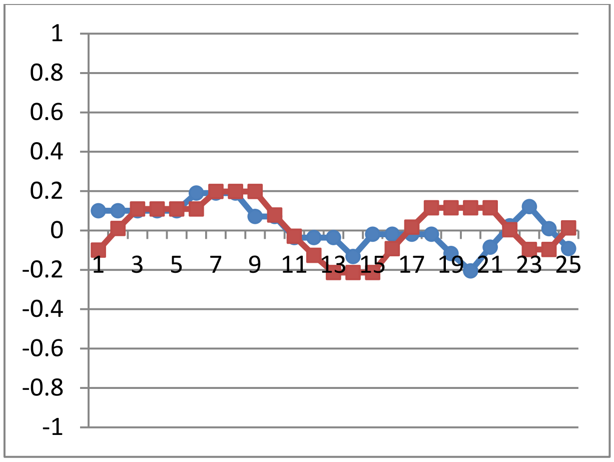


Negative surprise (● = seller1, ■ = seller 2)

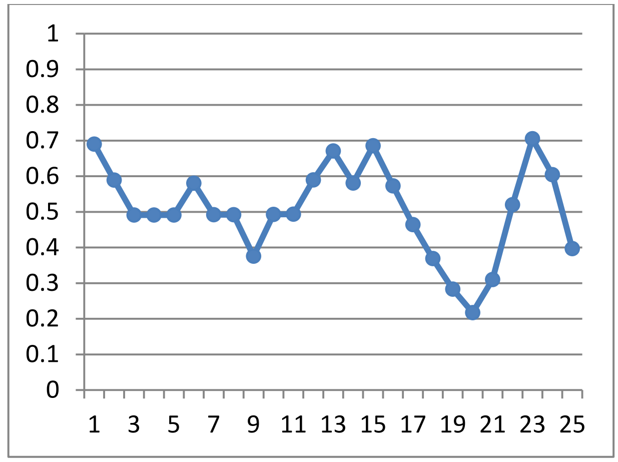


Check probability (● = seller1, ■ = seller 2)

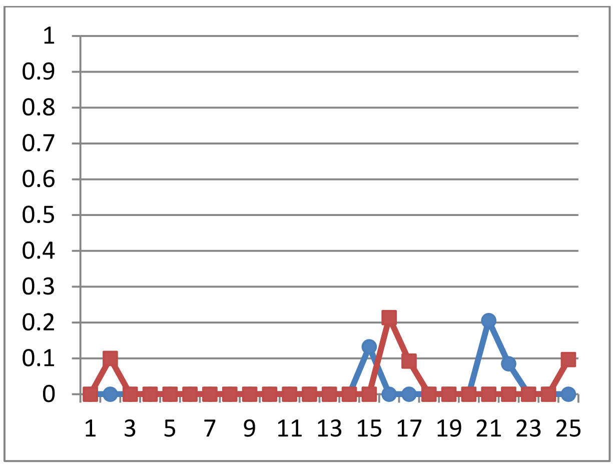
Figure 2: Seller 1 switches from fair to unfair after half the game; seller 2 does the opposite.



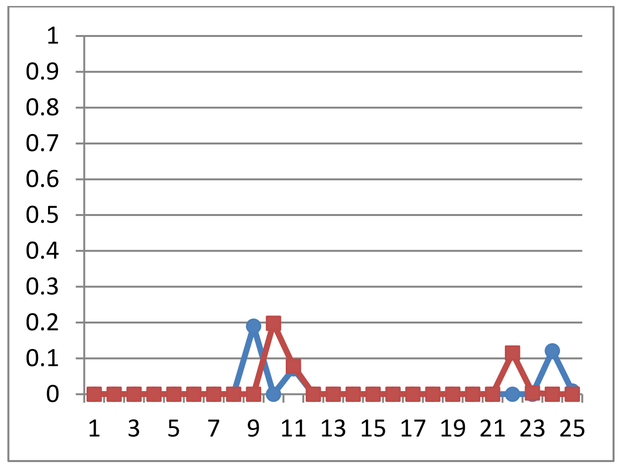
Trust (● = seller1, ■ = seller 2)



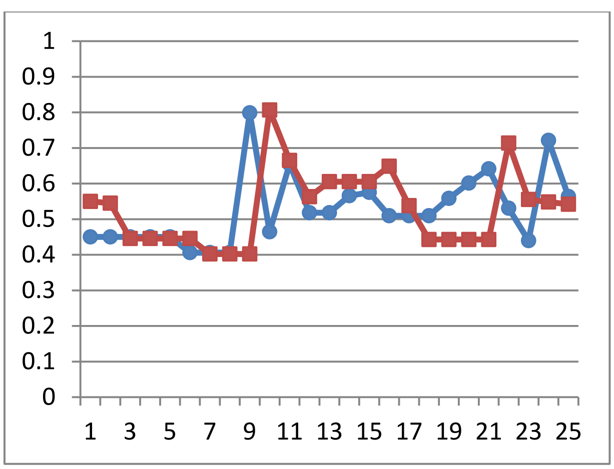
Selection probability for seller 1



Positive surprise (● = seller1, ■ = seller 2)



Negative surprise (● = seller1, ■ = seller 2)



Check probability (● = seller1, ■ = seller 2)

Figure 3: Both sellers play fair with probability 50%.

Figures 1 through 3 exemplify the dynamics of the buyer's behavioral model for given seller behavior.

Figure 1 shows a situation where seller one always plays fair and seller two always plays unfair. Consistently with this, the trust in seller one increases as the game advances, while the trust in seller two decreases. Consequently, the probability of buying from seller one eventually approaches one. The buyer experiences no surprises in this game since both sellers stick to the same strategy and hence always comply with the buyer's expectations. The probability to check on seller one decreases and the probability to check on seller two increases, consistently with the corresponding trust levels.

Figure 2 shows a situation where player one always plays fair for the first half of the game and then switches to unfair behavior, and player two does exactly the opposite. This behavior is consistently reflected by the trust levels and the corresponding probability to select an offer from seller one. This example shows well the effect of "surprise": Both the positive and negative surprise spike as the buyer observes the respective change in strategy. This in turn leads to an increased check probability on either seller during the transient phase.

Finally, Figure 3 shows a situation where both sellers randomly select to cheat or not with uniform probability. In reaction to this, the trust levels fluctuate rather unsystematically around zero and the buyer selects either seller with roughly the same probability. The random behavior of the sellers leads to the occasional occurrence of small negative and positive surprises. The probabilities to check on the sellers also stay fairly stable across the iterations.

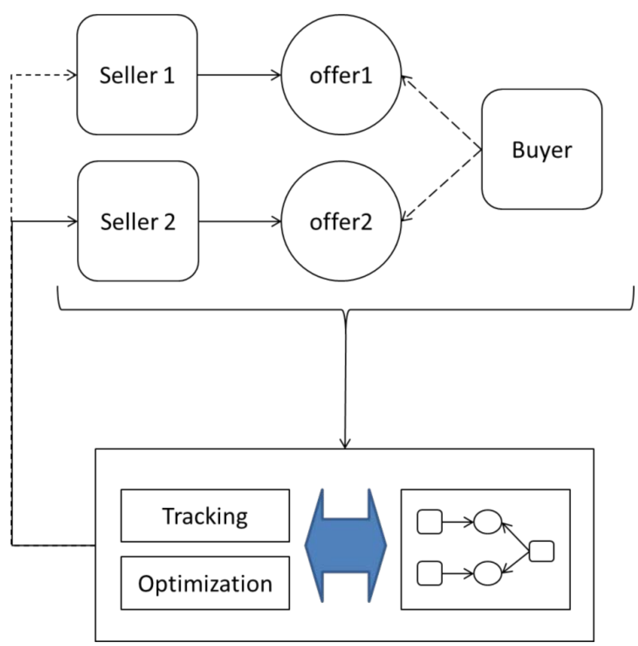


Figure 4: Controlled game flow.

5 CASE STUDY: CONTROL OF GAME FLOW

5.1 Implementation

The game flow is now controlled through the actions of one or both sellers, such that the buyer's experience of "deception" and "disappointment" is maximized. For this, a "disappointing situation" is defined as follows: The buyer selects for one or more rounds the same seller, who turns out to be fair whenever checked upon. Eventually, however, the player notices that the previously trusted seller started to make unfair offers. In reaction to this, the buyer switches to the other seller. The "level of information" $I(X)$ in a given realization X of the game is defined as the longest duration of a "trust" phase before the unfair behavior is detected and a switch to the other seller occurs. That is, $I(X)$ can take integer values 0, 1, 2, ….

The control of the game flow is implemented as follows. A simulated game is run forward through its stages. In the beginning of every stage, an identical copy of the current game state (including the state of all non-controlled players) is created. This is can be seen as an idealized tracking algorithm that, in a real application, would have to estimate the non-measurable aspects of the game, in particular the internal state of the buyer.

The future decision(s) of the controlled seller(s) is/are optimized by playing through different decision sequences in the cloned game situation (where possible future reactions of the buyer can be accounted for through the clone of the stochastic buyer model), with the objective to maximize $I(X)$ as defined above. (The optimization is performed through a simple randomized hill-climbing procedure; this heuristic approach was chosen merely for the sake of simplicity.) Once the optimization has terminated, each controlled seller selects the upcoming optimized action in the real game. (The buyer's behavior is not optimized but follows the previously defined behavioral model. Recall that in a real application of this game, the simulated buyer would be replaced by the real person under study.) This process is repeated until the game terminates. Figure 4 gives an overview.

5.2 RESULTS

Three configurations are considered. The "random" configuration does not optimize; it generates purely random behavior for each seller, selecting between fair and unfair offers with 50/50 probability. The "single control" configuration replaces one of the random sellers by an "optimal seller" that tries to steer the game into regimes of greatest information gain. In the "double control" configuration, both sellers are controlled and try jointly to steer the game into regimes of greatest information gain. 100 replications of each configuration are evaluated.

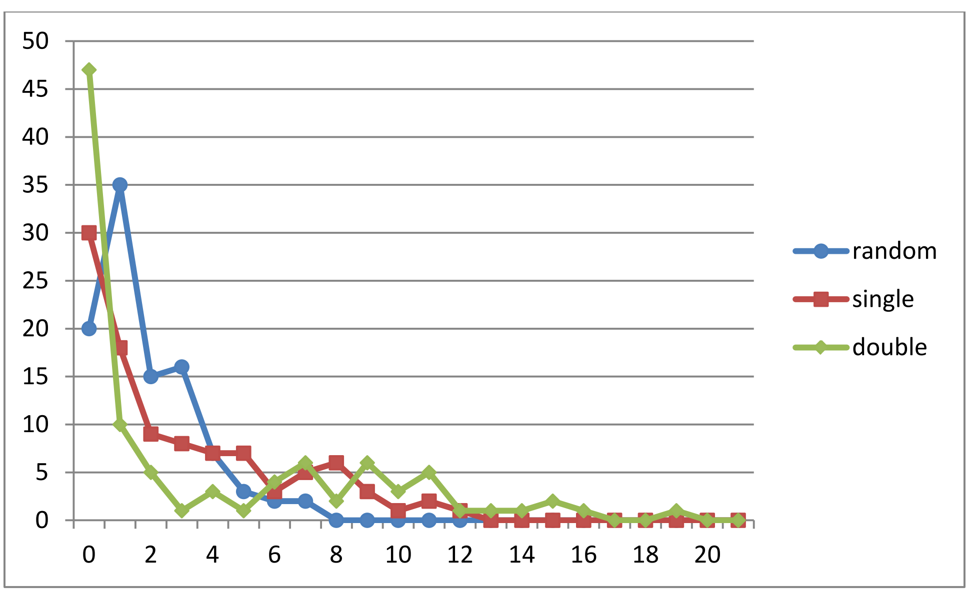


Figure 5: Histogram of $I(X)$.

Figure 5 shows a histogram of the resulting information values $I(X)$ for all three configurations. Two relevant observations can be made.

1. The "single control" histogram stretches further towards large $I(X)$ values than the "random" histogram. The "double control" histogram reaches even further towards larger values. That is, the optimization is successful in reaching more informative states. As one would expect, the coordinated control of two sellers leads to better results than the control of only one seller.
2. The probability of "failing" in the sense that zero information is extracted from a game is larger in the "single control" case than for the "random" case, and it is even larger for the "double control" case. This effect was not anticipated when specifying the objective function, but it is plausible in hindsight: In order to obtain large "disappointment" values, the sellers must treat the buyer fair for as long as possible, but they cannot ensure that the buyer actually becomes aware of that, given that they cannot control the buyer's decision to check on them. This leads to frequent failures of the game, resulting in zero values $I(X)$.

The practical relevance of these results, tentative and artificial as they may be, becomes clear in Table 2. It shows, for each configuration, the obtained mean information value, the maximum information value, the frequency of obtaining information values of at least 5, and the frequency of obtaining information values of at least 10. If, for example, relevant information can only be extracted from games with an information value of 5 or larger, the single control increases the number of useful games by a factor of four, and the double control even by a factor of five.

Table 2: Experimental results.

experiment	mean($I(X)$)	max($I(X)$)	freq($I(X) \geq 5$)	freq($I(X) \geq 10$)
random	1.82	7	7%	0%
single control	2.95	12	28%	4%
double control	3.65	19	34%	15%

6 SUMMARY AND CONCLUSIONS

This article uses basic discrete choice models, based on which it controlled a game flow to maximize the information that can be extracted from the game. The case study with a simplified version of the Trust and Tracing Game illustrates that the type of mathematical modeling is capable of producing desired results. It also showed that the approach easily gets quite complicated, especially when we think of more realistic gaming settings. With this approach it is possible to optimize the gaming environment in the loop, and to track the players based on discrete choice models. The next challenges that occur now are how to design games in such a way that the environment can actually be controlled sufficiently. The open-problem-centered games that are so popular in learning allow for a great many possible states of the environment. Future work will focus on more complex games.

REFERENCES

Bekebrede, G. and I.S. Mayer (2005). Build your seaport in a game and learn about complex systems. In: Journal of design research, 5(2): pp 273-298.

Ben-Akiva, M. and S. Lerman (1985). Discrete Choice Analysis - Theory and Application to Travel Demand. MIT Press.

De Freitas, S. and M. Oliver (2006). How can exploratory learning with games and simulations within the curriculum be most effectively evaluated? Computers and Education 46 (3) 249-264.

Dignum, Frank, Westra, Joost, Doesburg, Willem van (2009). Games and Agents: Designing Intelligent Gameplay.

International Journal of Computer Games Technology. Vol. 2009, pp. 1-18.
Dignum, F., J. Bradshaw, B. Silverman, W. van Doesburg (2010) (eds.), Agents for Games and Simulations: Trends in Techniques, Concepts and Design, LNAI 5920, Springer Verlag, 2010.Duke, R.D., 1974. Gaming: the future's language. Sage, Beverly Hills / London.
Duke, R.D. and J.L.A. Geurts, 2004. Policy games for strategic management. Dutch University Press, Amsterdam, The Netherlands.
Flötteröd, G. and M. Bierlaire (2011). Metropolis-Hastings sampling of paths. Technical Report TRANSP-OR 110606. Ecole Polytechnique Fédérale de Lausanne, Switzerland.
Klabbers, J.H.G. (2003). Simulation and gaming: introduction to the art and science of design. Simulation and Gaming, 34(4): 488-494.
Klabbers, J.H.G. (2006). Guest editorial. Artifact assessment vs. theory testing. Simulation & Gaming, 37(2): 148-154.
Kriz, W.C. and J.U. Hense (2006). Theory-oriented evaluation for the design of and research in gaming and simulation. Simulation Gaming, 37: 268 – 285.
Martens, A. and A.M. Uhrmacher (2004). A formal tutoring process model for intelligent tutoring systems. 16th European Conference on Artificial Intelligence, ECAI 2004 Valencia, Spain.
Meijer, S.A. (2009): The organisation of transactions : studying supply networks using gaming simulation. Wageningen Academic. ISBN 9789085853343.
Meijer, S.A.; Hofstede, G.J.; Beers, G.; Omta, S.W.F. (2006): Trust and Tracing game: learning about transactions and embeddedness in a trade network. Production Planning & Control 17 (2006)6. - ISSN 0953-7287 - p. 569 - 583.
Noy, A; D.R. Raban and G. Ravid, 2006. Testing social theories in computer-mediated communication through gaming and simulation. Simulation & Gaming, 37(2): 174-194.
Peters, V., G. Vissers and G. Heijne (1998). The validity of games. Simulation & Gaming, 29(1): 20 – 30.
Train, K. (2003). Discrete Choice Methods with Simulation. Cambridge University Press.

AUTHOR BIOGRAPHIES

GUNNAR FLÖTTERÖD is assistant professor at KTH Royal Institute of Technology, Department of Transport Science. His Email address is gunnar.floetteroed@abe.kth.se.

SEBASTIAAN MEIJER is associate professor at KTH Royal Institute of Technology, Department of Transport Science, and part-time assistant professor at Delft University of Technology, Faculty of Technology, Policy and Management. His Email address is smeijer@kth.se.