ELSEVIER

Decision Support Systems 33 (2002) 323–333

Decision Support Systems

www.elsevier.com/locate/dsw

Computers play the beer game: can artificial agents manage supply chains?

Steven O. Kimbrough [a,1], D.J. Wu [b,*], Fang Zhong [b]

[a]The Wharton School, University of Pennsylvania, 3620 Locust Walk, Philadelphia, PA 19104 6366, USA
[b]LeBow College of Business, Drexel University, Philadelphia, PA 19104, USA

Abstract

We model an electronic supply chain managed by artificial agents. We investigate whether artificial agents do better than humans when playing the MIT Beer Game. Can the artificial agents discover good and effective business strategies in supply chains both in stationary and non-stationary environments? Can the artificial agents discover policies that mitigate the Bullwhip effect? In particular, we study the following questions: Can agents learn reasonably good policies in the face of deterministic demand with fixed lead time? Can agents cope reasonably well in the face of stochastic demand with stochastic lead time? Can agents learn and adapt in various contexts to play the game? Can agents cooperate across the supply chain? © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Artificial agents; Automated supply chains; Beer game; Bullwhip effect; Genetic algorithms

1. Introduction

We consider the case in which an automated enterprise has a supply chain ‘manned’ by various artificial agents. In particular, we view such a virtual enterprise as a multi-agent system [10]. Our example of supply chain management is the MIT Beer Game [11], which has attracted much attention from supply chain management practitioners as well as academic researchers. There are four types of agents (or roles) in this chain, sequentially arranged: Retailer, Wholesaler, Distributor, and Manufacturer. In the MIT Beer Game, as played with human subjects, each agent tries to achieve the goal of minimizing long-term system-wide total inventory cost in ordering from its immediate supplier. The decision of how much to order for each agent may depend on its own prediction of future demand by its customer, which could be based on its own observations. Other rules might as well be used by the agents in making decisions on how much to order from a supplier.

The observed performance of human beings operating supply chains, whether in the field or in laboratory settings, is usually far from optimal from a system-wide point of view [6,11,13]. This may be due to lack of incentives for information sharing, bounded rationality, or possibly the consequence of individually rational behavior that works against the interests of the group. It would thus be interesting to see if we can gain

[*] Corresponding author. Tel.: +1-215-895-2121; fax: +1-215-895-2891.
E-mail addresses: kimbrough@wharton.upenn.edu (S.O. Kimbrough), wudj@drexel.edu (D.J. Wu), fz23@drexel.edu (F. Zhong).
URL: http://www.grace.wharton.upenn.edu/~sok.
[1] Tel.: +1-215-898-5133; fax: +1-215-898-3664.

PII: S0167-9236(02)00019-2

insights into the operations and dynamics of such supply chains by using artificial agents instead of humans. This paper makes a contribution in that direction. We differ from current research on supply chains in the OM/OR area in that the approach taken here is an agent-based, information-processing model in an automated marketplace. In the OM/OR literature, the focus is usually on finding optimal solutions assuming a fixed environment such as: fixed linear supply chain structure, known demand distribution to all players, and fixed lead time in information delay as well physical delay [1,6,13]. It is generally very difficult to derive and to generalize such optimal policies for changing environments.

This paper can be viewed as a contribution to the understanding of how to design learning agents to discover insights for complicated systems, such as supply chains, which are intractable when using analytic methods. We believe our initial findings vindicate the promise of the agent-based approach. We do not essay here to contribute to management science or operations research analytic modeling, or to the theory of genetic algorithms. However, throughout this study, we use findings from the management science literature to benchmark the performance of our agent-based approach. The purpose of the comparison is to assess the effectiveness of an adaptable or dynamic order policy that is automatically managed by computer programs — artificial agents.

We have organized the paper as follows. Section 2 provides a brief literature review. Section 3 describes our methodology and implementation of the Beer Game using an agent-based approach. Section 4 summarizes results from various experiments, mainly on the MIT Beer Game. Section 5 provides a discussion of our findings in light of relevant literature. Section 6 summarizes our findings, offers some comments, and points to future research.

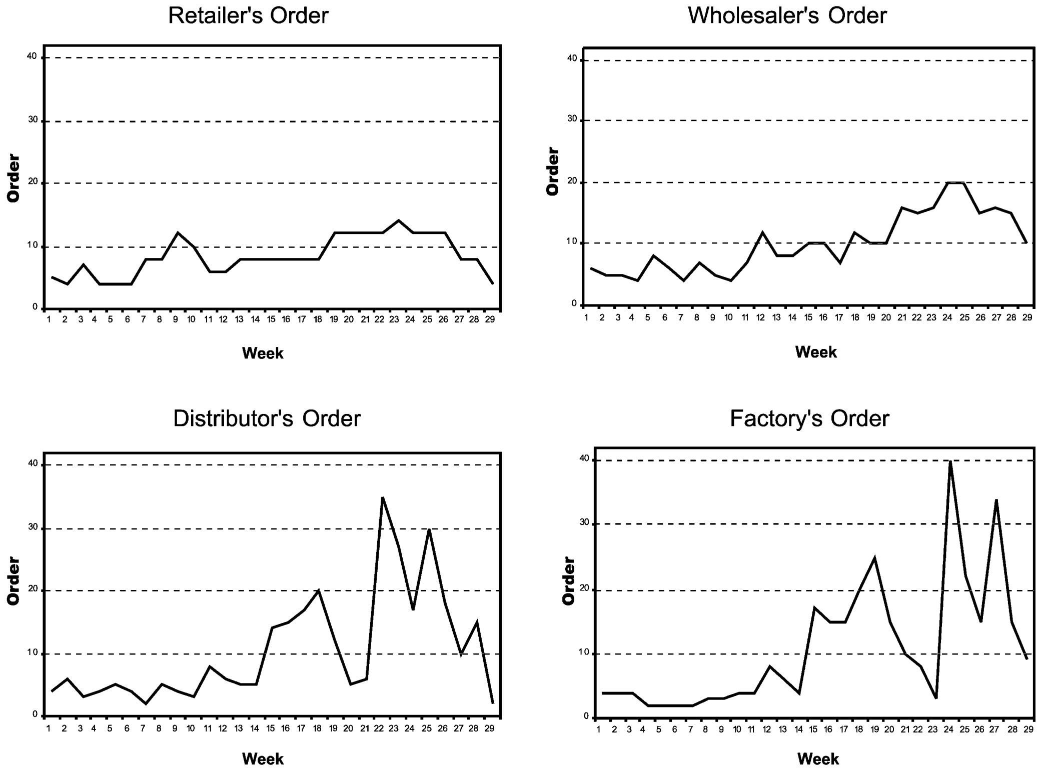


Fig. 1. Bullwhip effect using real data from a group of undergraduates playing the Beer Game.

2. Literature review

A well-known phenomenon, both in industrial practice and when humans play the Beer Game, is the bullwhip or whiplash effect—the variance of orders amplifies upstream in the supply chain [4,5]. An example of the bullwhip effect is shown in Fig. 1, using data from a group of undergraduates playing the MIT Beer Game (date played May 8, 2000).

These results are typical and have been often replicated. As a consequence, much interest has been generated among researchers regarding how to eliminate or minimize the bullwhip effect in the supply chain. In the management science literature, sources of the bullwhip effect have been identified and counteractions have been offered [5]. It is shown in Ref. [1] that the bullwhip effect can be eliminated under a base-stock installation policy given the assumption that all divisions of the supply chain work as a team [8]. Base-stock is optimal when facing stochastic demand with fixed information and physical lead time [1]. When facing deterministic demand with penalty costs for every player (The MIT Beer Game), the optimal order for every player is the so-called “pass order,” or “one for one” (1 – 1) policy—order whatever is ordered from your own customer.

In the MIS literature, the idea of combining multi-agent systems and supply chain management has been proposed in Refs. [7,9,10,12,14]. Actual results, however, have mostly been limited to conceptualizing. Our approach differs from previous work in that we focus on quantitative and agent computation methodologies, with agents learning via genetic algorithms.

It is well known that the optimality of 1 – 1 depends on the initial base stock level, and it is optimal only for limited cases. In particular, 1 – 1 requires the system to be stationary. We shall see that artificial agents not only can handle the stationary case by discovering the optimal policy, they also perform well in the non-stationary case.

In the next section, we describe in detail our methodology and implementation of the Beer Game.

3. Methodology and implementation

We adopt an evolutionary computation approach. Our main idea is to replace human players with artificial agents playing the Beer Game and to see if artificial agents can learn and discover good and efficient order policies in supply chains. We are also interested in knowing whether the agents can discover policies that mitigate the bullwhip effect.

The basic setup and temporal ordering of events in the MIT Beer Game [11] are as follows. New shipments arrive from the upstream player (in the case of the Manufacturer, from the factory floor); orders are received from the downstream player (for the Retailer, exogenously, from the end customer); the orders just received plus any backlog orders are filled if possible; the agent decides how much to order to replenish stock; inventory costs (holding and penalty costs) are calculated.[2]

3.1. Cost function

In the MIT Beer Game, each player incurs both inventory holding costs, and penalty costs if the player has a backlog. We now derive the total inventory cost function of the whole supply chain. As noted above, we depart from previous literature in that we are using an agent-based inductive approach, rather than a deductive/analytical approach.

We begin with needed notation. N = number of players, $i = 1,\ldots,N$. In the MIT Beer Game, $N = 4$. $\mathrm{IN}_i(t)$ = net inventory of player i at the beginning of period t. $C_i(t)$ = cost of player i at period t. H_i = inventory holding cost of player i, per unit per period (e.g., in the MIT Beer Game, US\$1 per case per week). P_i = penalty/backorder cost of player i, per unit per period (e.g., in the MIT Beer Game, US\$2 per case per week). $S_i(t)$ = new shipment player i received in period t. $D_i(t)$ = demand received from the downstream player in week t (for the Retailer, the demand from customers).

According to the temporal ordering of the MIT Beer Game, each player’s cost for a given time period, e.g., a week, can be calculated as following: If $\mathrm{IN}_i(t) \geq 0$, then $C_i(t) = \mathrm{IN}_i(t) \times H_i$ else $C_i(t) = |\mathrm{IN}_i(t)|$

[2] The initial/starting inventories for the MIT Beer Game are 12 cases of beer in the warehouse, 8 cases in the pipeline with 4 cases in the “truck” to be delivered to the warehouse in 1 week and 4 cases in the “train” to be delivered in the warehouse in 2 weeks [11].

$\times P_i$, where $\mathrm{IN}_i(t)=\mathrm{IN}_i(t-1)+S_i(t)-D_i(t)$ and $S_i(t)$ is a function of both information lead time and physical lead time. The total cost for the supply chain (the whole team) after M periods (e.g., weeks) is $\sum_{i=1}^{N}\sum_{t=1}^{M} C_i(t)$.

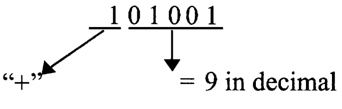
We implemented a multi-agent system for the Beer Game, and used it as a vehicle to perform a series of investigations that aim to test the performance of our agents. The following algorithm defines the procedure for our multi-agent system to search for the optimal order policy.

3.2. Algorithm

(1) Initialization. A certain number of rules are randomly generated to form generation 0.
(2) Pick the first rule from the current generation.
(3) Agents play the Beer Game according to their current rules.
(4) Repeat step (3), until the game period (say 35 weeks) is finished.
(5) Calculate the total average cost for the whole team and assign fitness value to the current rule.
(6) Pick the next rule from the current generation and repeat steps (2), (3) and (4) until the performance of all the rules in the current generation have been evaluated.
(7) Use a genetic algorithm to generate a new generation of rules and repeat steps (2)–(6) until the maximum number of generation is reached.

3.3. Rule representation

Now we describe our coding strategy for the rules. Each agent's rule is represented with a 6-bit binary string. The leftmost bit represents the sign ("+" or "−") and the next five bits represent (in base 2) how much to order. For example, rule 101001 can be interpreted as "$x+9$":



That is, if demand is x then order $x+9$. Technically, our agents are searching in a simple function space. Notice that if a rule suggests a negative order, then it is truncated to 0 since negative orders are infeasible. For example, the rule $x-15$ means that if demand is x then order Max[0, $x-15$]. Thus, the size of the search space is 2^6 for each agent.

When all four agents are allowed to make their own decisions simultaneously, we will have a "Über" rule for the team, and the length of binary string becomes $6\times4=24$. The size of search space is considerably enlarged, but still fairly small at 2^{24}. Below is an example of such an Über rule: $\underline{110001}$ $\underline{001001}$ $\underline{101011}$ $\underline{000000}$, where 110001 is the rule for Retailer; 001001 is the rule for Wholesaler; 101011 is the rule for Distributor; 000000 is the rule for Manufacturer.

Why use the "$x+y$" rule? Recent literature has identified the main cause of the bullwhip effect as due to the agents' overreaction on their immediate orders (see, e.g., Ref. [5]). While the optimality of this rule (the "$x+y$" rule) depends on the initial inventories, so do the optimality of other inventory control policies such as the (s, S) policy or base-stock policy [2], which are only optimal for the stationary case. The "$x+y$" rule is more general in the sense that it handles both the stationary and the non-stationary case. We leave for future research exploration of alternative coding strategies and function spaces. Why not let everybody to use the "order-up-to" policy? This is essentially the "$1-1$" policy our agents discover. It, however, is only optimal for the stationary case with the optimal initial configuration. There are three main reasons that preclude the usefulness of this rule (every body uses the "order-up-to" policy) in the general case where an adaptable rule is more useful. First, there is the coordination problem, the key issue we are trying to address here. It is well established in the literature (see, e.g., Ref. [1]) that if everyone uses the (s, S) policy, (which is optimal in the stationary case), the total system performance need not be optimal, and indeed could be very bad. Second, we are interested to see if the agent can discover an optimal policy when it exists. We will see that a simple GA learning agent, or a simple rule language, does discover the "$1-1$" policy (the winning rule), which is essentially equivalent to the base-stock policy. Third, the supply chain management literature has only been able to provide analytic solutions in very limited — some would argue unrealistic — cases. For the real world problems we are interested in, the literature has little to offer directly, and we believe our agent-based approach may well be a good alternative.

3.4. Rule learning

Agents learn rules via a genetic algorithm [3], where the absolute fitness function is the negative of total cost, TC(r), under (Über) rule r. We use a standard selection mechanism (proportional to fitness), as well as standard crossover and mutation operators.

4. Experimental results

In the first round of experiments, we tested agent performance under deterministic demand as set up in the classroom MIT Beer Game. In the second round, we explored the case of stochastic demand where demand is randomly generated from a known distribution, e.g., uniformly distributed between [0, 15]. In the third round, we changed the lead time from a fixed 2 weeks to being uniformly distributed between 0 to 4. In the fourth round, we conducted two more experiments with eight players under both deterministic and stochastic demands. Finally, we tested the agents' performance for a variation on the MIT Beer Game (we call it the Columbia Beer Game). We now briefly describe each round of experiment and our findings.

4.1. Experiment 1

In the first round, we tested agent performance under deterministic demand as in the classroom MIT Beer Game. The customer demands four cases of beer in the first 4 weeks, and then demands eight cases of beer per week starting from week 5 and continuing until the end of the game (35 weeks). We fixed the policies of three players (e.g., the Wholesaler, Distributor, and Manufacturer) to be $1-1$ rules, i.e., "if current demand is x, then order x from my supplier", while letting the Retailer agent adapt and learn. The goal of this experiment is to see whether the Retailer agent can match the other players' strategy and learn the $1-1$ policy.

We found that the artificial agent can learn the $1-1$ policy consistently. With a population size of 20, the Retailer agent finds and converges to the $1-1$ policy very quickly.

We then allowed all four agents make their own decisions simultaneously (but jointly, in keeping with the literature's focus on teams). In the evaluation function, each agent places orders according to its own rule for the 35 weeks and current cost is calculated for each agent separately. However, for the Wholesaler, Distributor and Manufacturer, their current demand will be the order from its downstream partner, which are Retailer, Wholesaler, and Distributor. The four agents act as a team and their goal is to minimize their total cost over a period of 35 weeks. The result is that each agent can find and converge to the $1-1$ rule separately, as shown in Fig. 2. This is certainly much better than human agents (MBA and undergraduate students; however, undergraduates tend to do better than MBAs), as shown in Fig. 3.

We conducted further experiments to test whether one for one is a Nash equilibrium. To this end, we define our supply chain game as follows. The players are, again, Retailer, Wholesaler, Distributor, and Manufacturer. The payoff is own total inventory cost. The strategies are all feasible order policies for the player. What we mean by the Nash equilibrium is the following. Fix the order policies of the other three players, say the Wholesaler, Distributor, and Manufacturer; if the fourth player, say the Retailer, has no

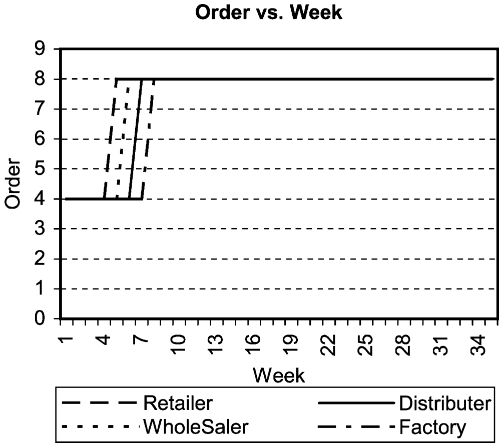


Fig. 2. Artificial agents are able to find the optimal "pass order" policy when playing the MIT Beer Game.

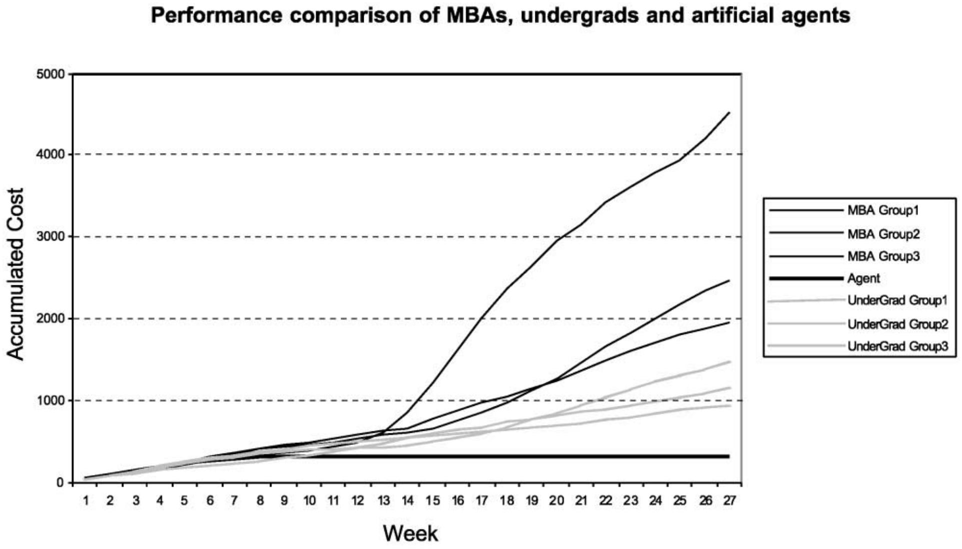


Fig. 3. Artificial agents perform much better than MBA and undergraduate students in the MIT Beer Game.

incentive to deviate from the $1-1$ policy, then we say that the $1-1$ policy constitutes a Nash Equilibrium. Notice here that when we fix the order policies of the other three, then the goal of minimizing the team cost (the total cost of the supply chain) is consistent with the goal of minimizing its own inventory cost. For example, in the case of the manufacturer, these two goals are equivalent when all down stream players are using the "$1-1$" rule.

Each agent finds one for one and it is stable, therefore it apparently constitutes a Nash equilibrium. We also introduced more agents into the supply chain.[3] In these experiments, all the agents discover the optimal one for one policy, but it takes longer to find it when the number of agents increases. Our experiment shows that the convergence generations for 4, 5 and 6 agents are 5, 11, and 21 correspondingly (one sample run).

4.2. Experiment 2

In the second round, we tested the case of stochastic demand where demand is randomly generated from a known distribution, e.g., uniformly distributed between [0, 15].[4] The goal was to find out whether artificial agents track the trend of demand, whether artificial agents discover good dynamic order policies. We used $1-1$ as a heuristic to benchmark. Our experiments show that the artificial agents track the stochastic customer demand reasonably well as shown in Fig. 4 (for the Retailer agent only). Each artificial agent discovers a dynamic order strategy (for example, each agent uses the $x+1$ rule) that outperforms $1-1$, as shown in Fig. 5.

As in experiment 1, we conducted a few more experiments to test the robustness of the results. First, the (apparent) Nash property remains under stochastic demand. Second, we increased the game period from 35 to 100 weeks, to see if agents beat one for one over a longer time horizon. The customer demand remained uniformly distributed between [0, 15]. This time, agents found new strategies $(x+1, x, x+1, x)$ that outperform the one for one policy. This is so, because the agents seem to be smart enough to adopt this new strategy to build up inventory for the base stock (more

[3] What we meant by the word "apparent" is that we have been able to demonstrate the Nash property in our experiment via a "proof by simulation" but not closed-form solution.

[4] One realization of the customer demands draw from this distribution for 35 weeks are [15, 10, 8, 14, 9, 3, 13, 2, 13, 11, 3, 4, 6, 11, 15, 12, 15, 4, 12, 3, 13, 10, 15, 15, 3, 11, 1, 13, 10, 10, 0, 0, 8, 0, 14], and this same demand has been used for subsequent experiments in order to compare various order policies.

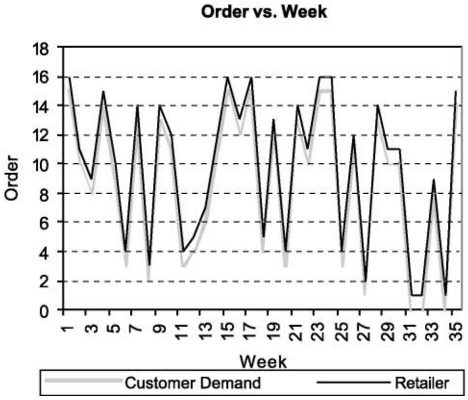


Fig. 4. Artificial Retailer agent is able to track customer demand reasonably well when facing stochastic customer demand (parameters for GA: number of generations = 30, population size = 1000, crossover rate = 0.87, mutation rate = 0.03).

on this in the concluding section). Our contribution is, again, to *explore* the agent-based platform for an adaptive and dynamic policy for efficient supply chain management. It is not surprising that adaptive and dynamic policies would be more efficient but very difficult to find, characterize and implement. This work provides a first yet promising step in this direction.

We also tested for statistical validity by rerunning the experiment with different random number seeds. The results were robust.

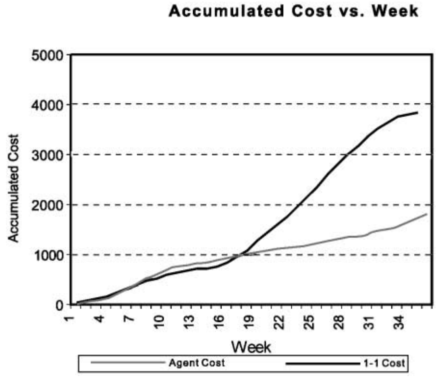


Fig. 5. When facing stochastic demand and penalty costs for all players as in the MIT Beer Game, artificial agents seem to discover a dynamic order policy that outperforms "$1-1$" (one sample run; parameters given in footnote 4).

Table 1
The evolution of effective adaptive supply chain management policies when facing stochastic customer demand and stochastic lead time

Generation	Strategies of agents				Total cost
	I	II	III	IV	
0	$x-0$	$x-1$	$x+4$	$x+2$	7380
1	$x+3$	$x-2$	$x+2$	$x+5$	7856
2	$x-0$	$x+5$	$x+6$	$x+3$	6987
3	$x-1$	$x+5$	$x+2$	$x+3$	6137
4	$x+0$	$x+5$	$x-0$	$x-2$	6129
5	$x+3$	$x+1$	$x+2$	$x+3$	3886
6	$x-0$	$x+1$	$x+2$	$x+0$	3071
7	$x+2$	$x+1$	$x+2$	$x+1$	2694
8	$x+1$	$x+1$	$x+2$	$x+1$	2555
9	$x+1$	$x+1$	$x+2$	$x+1$	2555
10	$x+1$	$x+1$	$x+2$	$x+1$	2555
11	$x+1$	$x+1$	$x+2$	$x+1$	2555
12	$x+1$	$x+1$	$x+2$	$x+1$	2555
13	$x+1$	$x+1$	$x+2$	$x+1$	2555
14	$x+1$	$x+1$	$x+2$	$x+1$	2555
15	$x+1$	$x+1$	$x+2$	$x+1$	2555
16	$x+1$	$x+1$	$x+2$	$x+1$	2555
17	$x+1$	$x+1$	$x+2$	$x+1$	2555
18	$x+1$	$x+1$	$x+2$	$x+1$	2555
19	$x+1$	$x+1$	$x+2$	$x+1$	2555
20	$x+1$	$x+1$	$x+2$	$x+1$	2555

4.3. Experiment 3

In the third round of experiments, we changed the lead time from a fixed 2 weeks to uniformly distributed from 0 to 4.[5] So now we are facing both stochastic demand and stochastic lead time. Following the steps we described in the first experiment and using the stochastic customer demand and lead time as we mentioned previously, the agents find strategies better than $1-1$ within 30 generations. The best rule agents found so far is ($x+1$, $x+1$, $x+2$, $x+1$), with a total cost for 35 weeks of 2555 (one sample run). This is much less than 7463 obtained using the $1-1$ policy. Table 1 shows the evolution of effective adaptive supply chain management policies when facing stochastic customer demand and stochastic lead time in the context of our experiments. Artificial agents seem to be able to discover dynamic ordering policies that

[5] One realization of the lead time draw from the same distribution for the 35 weeks in our experiment are [2, 0, 2, 4, 4, 4, 0, 2, 4, 1, 1, 0, 0, 1, 1, 0, 1, 1, 2, 1, 1, 1, 4, 2, 2, 1, 4, 3, 4, 1, 4, 0, 3, 3, 4].

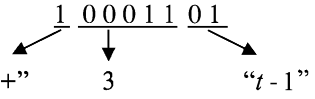
outperform the 1−1 policy when facing stochastic demand and stochastic lead time, as shown in Fig. 6. We further tested the stability of these strategies discovered by agents, and found that they are stable, and thus constitute an apparent Nash equilibrium.

4.4. Further experiments

The main challenge to computational investigation is the complexity of the search spaces when we use more realistic distribution networks. In most of above experiments, we have a small search space: 2^{24}. A simple extension to eight players, however, quickly produces a nontrivial search space of 2^{48}. So we conducted two further experiments with eight players under both deterministic and stochastic demands. For the deterministic demand, all the agents find the known optimal policy of 1−1 at around the 89th generation with a population size of 2000. For stochastic demand, in a one-run experiment, agents found a better policy (x, x, x+1, x, x+1, x+1, x+1, x+1) (total cost under this policy is 4494) than 1−1 (total cost under 1−1 is 12225) at the 182nd generation, with a population size of 3500. The agents are able to search very efficiently and the space searched compared with the entire rule space is negligible.

4.5. The Columbia Beer Game

We experimented with the Columbia Beer Game along the same settings as in Ref. [1], except that none of our agents has knowledge about the distribution of customer demand.[6] As an example, we set information lead time to be (2, 2, 2, 0), physical lead time to be (2, 2, 2, 3) and customer demand as normally distributed with mean of 50 and standard deviation of 10. This set up is borrowed from Ref. [1]. Since different players might have different information delays in the Columbia Beer Game, we modified our string representation to accommodate this. To do so, we added two bits to the bit-string. For example, if the Retailer has a rule of 10001101, it means "if demand is x in week t−1, then order x+3", where t is the current week:



Notice that the master rule for the whole team now has the length of 8×4=32. Under the same learning mechanism as described in the previous sections, agents found the optimal policy: order whatever amount is ordered with time shift, this is consistent with the MS/OR literature (e.g., see Ref. [1]).

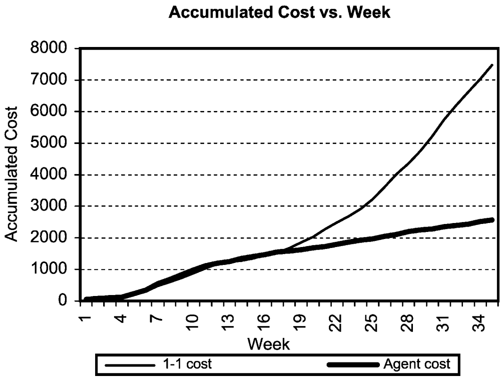


Fig. 6. Artificial agents seem to be able to discover dynamic ordering policies that outperform the "1−1" policy when facing stochastic demand and stochastic lead time (one sample run; parameters given in footnotes 4 and 5).

5. Discussion

The agent-based approach studied in this paper replicates the results of the Beer Game as reported in the MS/OR/OM literature, under the assumptions made for the models in question. The rationale of using this as a starting point is for benchmarking: if we are to field intelligent, (semi-) autonomous agents in E-Business such as in supply chains, much remains to be learned about the principles of their design and management, especially how to measure/control the performance of such agents. We think a good program of research is to start with some well-defined supply chain problems having well-known properties; to train

[6] The "Columbia Beer Game" [1] is a modified version of the "MIT Beer Game". The former assumes only the Retailer incurs penalty cost; downstream players incur higher holding costs; all players know the demand distribution, while the latter assumes all players incur the same penalty and holding costs. A unique advantage of an agent-based system like ours is to test and simulate system performance under various assumptions like the above and obtain insights.

and test agents on these problems; and then move to more complicated arrangements in which traditional analytic methods are unavailable. These latter are the most promising domains for artificial agents. Whether the promise will be fulfilled will be determined by actual computational experience with algorithms, some of which remain to be discovered. We now want to discuss several worries about the approach, which if not addressed convincingly might preclude further investigation.

In the experiments we described above, the agents were individually able to find fully cooperative solutions. Is this inevitable? The answer is no. In fact, the majority of the rules examined by the agents were not cooperative. It is the underlying structure of the problem that caused the cooperative solutions to emerge. The agent-based approach helps us discover this aspect of the underlying problem. The larger challenge is to characterize other patterns or underlying structures in supply chains. So far in the literature, it has been conjectured that there might be other fully cooperative rules [1]. The approach we have developed here can be used to investigate such conjectures.

Can the proposed "x+y" rule produce the bullwhip effect? As noted above, the main cause of bullwhip effect (characterized in Ref. [5]) is that the players in the supply chain order based on their immediate customers' orders, rather than the actual end customer's demand or the actual consumption. The "x+y" rule space used above could well produce the bullwhip effect if y changes over time, since every agent is making its decisions independently, just as when human beings are playing the Beer Game. The winning rules (when y is fixed over time) get rid of the bullwhip effect, both in the stationary and the non-stationary cases. Thus, the structure of our particular experiment precludes the bullwhip effect. We leave it to future research to investigate cases in which y can vary with time.

In summary, we are interested in the following questions:[7] (1) whether a simple strategy space is sufficient to encode optimal (or near-optimal) control policies for the supply chains, and (2) whether a simple genetic learning method can converge to a constrained-optimal control given a restricted strategy space. The first question is interesting in an environment where known effective control policies are complicated, and we want to see if a simple rule language is adequate. The second question is interesting in an environment where effective control polices are unavailable analytically or using conventional search methods. Such an environment, for example, includes the case where the agents are considered separately (and not as a team). Our agents are searching in rather simple function spaces. The functions in the spaces available to our agents are functions only of the demands seen by the agents (either in the present period or in some previous period). Consequently, the agents are not allowed to choose rules that are, for example, sensitive to the initial inventory position. Clearly, in these circumstances and in the long run, only some version of a $1-1$ rule can be optimal; otherwise, the agents will either build up inventory without bound or will essentially eliminate it, thereby incurring excessive penalties. How long is long? Under the assumptions of the Beer Game, we have seen that 35 weeks is not long enough. Extending the play to 100 weeks is, our simulations indicate, also not long enough. For practical purposes, it may well be that, say because of inevitable changes in the business environment, only the short run characteristics of the system are of real interest. If so, then real agents may well find the discoveries of artificial agents such as ours most interesting. One way of interpreting why, in the short run, our agents deviate from a $1-1$ policy is this. The agents are searching for rules that yield, over the life of the simulation, the best levels of inventory. Non-$1-1$ rules emerge when the starting inventory levels are non-optimal and the time horizon is limited. As theoreticians, we leave it to others to determine whether businesses ever encounter such situations: inventory on hand that is not known to be at an optimal level and a less than infinitely long planning horizon confronts us.

6. Summary and further research

We now summarize the contributions of this paper. First and foremost, we have explored the

[7] We thank a referee for summarizing and clarifying the discussion here.

concept of an intelligent supply chain managed by artificial agents. We have found that agents are capable of playing the Beer Game effectively. They are able to track demand, eliminate the Bullwhip effect, discover the optimal policies (where they are known), and find good policies under complex scenarios where analytical solutions are not available. Second, we have observed that such an automated supply chain is adaptable to an ever-changing business environment. The principal results we obtained are as follows.

(1) In the cases studied for which optimal order policies are known analytically (deterministic or stochastic demand, deterministic lead times), the artificial agents rather quickly found the optimal rules.

(2) In the cases for which optimal order policies are known analytically (deterministic or stochastic demand, deterministic lead times), we performed additional experiments and found that the optimal solutions are also (apparently) Nash equilibria. That is, no single agent could find an alternative order policy for itself that was better for itself than the policy it had in the team optimum, provided the other team members also used the team optimum policies. This is a new result, one that extends what was known analytically. It has intriguing consequences for applications; development of that point, however, is beyond the scope of this paper.

(3) In certain cases for which optimal order policies are not known analytically (stochastic demand and lead times, with penalty costs for every player), the artificial agents rather quickly found rules different from rules known to be optimal in the simpler cases (deterministic lead times).

(4) In these cases (for which optimal order policies are not known analytically), we performed additional experiments and found that the apparently optimal solutions found by the agents are also apparently Nash equilibria. (Again, no single agent could find an alternative order policy for itself that was better for itself than the policy it had in the team apparent optimum, provided the other team members also used the team apparent optimum policies.) This is a new result, one that extends what was known analytically; again, there are intriguing consequences for practice.

(5) In repeated simulations with stochastic demands and stochastic lead times, we paired the apparently optimal rules found by the artificial agents with the $1-1$ rules (known to be optimal in the simpler situations). We consistently found that the agents performed better than the $1-1$ rules. Thus, we have shown that analytic optimal results do not generalize and that in the generalization the agents can find good (if not optimal) policies.

Acknowledgements

Thanks to David H. Wood for many stimulating conversations and to the participants of the FMEC 2000 workshop for their comments. An earlier version of this paper appeared in Proceedings of the Thirty-fourth Annual Hawaii International Conference on System Sciences, IEEE Computer Society Press, Los Alamitos, California, 2001. Thanks to two anonymous referees for their critical yet extremely helpful comments. This material is based upon work supported by, or in part by, NSF grant number 9709548. Partial support by a mini-Summer research grant and a research fellowship by the Safeguard Scientifics Center for Electronic Commerce Management, LeBow College of Business at Drexel University are gracefully acknowledged.

References

[1] F. Chen, Decentralized supply chains subject to information delays, Management Science 45 (8) (1999) 1076–1090, August.

[2] A. Clark, H. Scarf, Optimal policies for a multi-echelon inventory problem, Management Science 6 (1960) 475–490.

[3] J. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, The MIT Press, Cambridge, 1992.

[4] H. Lee, V. Padmanabhan, S. Whang, Information distortion in a supply chain: the bullwhip effect, Management Science 43 (4) (1997) 546–558, April.

[5] H. Lee, V. Padmanabhan, S. Whang, The Bullwhip effect in supply chains, Sloan Management Review, (1997) 93–102, Spring.

[6] H. Lee, S. Whang, Decentralized multi-echelon supply chains: incentives and information, Management Science 45 (5) (1999) 633–640, May.

[7] F. Lin, G. Tan, M. Shaw, Multi-agent enterprise modeling, Journal of Organizational Computing and Electronic Commerce 9 (1) (1999) 7–32.

[8] J. Marschak, R. Radner, Economic Theory of Teams, Yale University Press, New Haven, CT, 1972.

[9] M. Nissen, Supply chain process and agent design for e-commerce, in: R.H. Sprague Jr. (Ed.), Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (CD ROM), IEEE Computer Society Press, Los Alamitos, CA, 2000.

[10] R. Sikora, M. Shaw, A multi-agent framework for the coordination and integration of information systems, Management Science 40 (11) (1998) S65–S78, November.

[11] J. Sterman, Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment, Management Science 35 (3) (1989) 321–339, March.

[12] T. Strader, F. Lin, M. Shaw, Simulation of order fulfillment in divergent assembly supply chains, Journal of Artificial Societies and Social Simulation 1 (2) (1998).

[13] S. Tayur, R. Ganeshan, M. Magazine (Eds.), Quantitative Models for Supply Chain Management, Kluwer Academic Publishers, Boston, 1999.

[14] S. Yung, C. Yang, A new approach to solve supply chain management problem by integrating multi-agent technology and constraint network, in: R.H. Sprague Jr. (Ed.), Proceedings of the 32nd Annual Hawaii International Conference on System Sciences (CD ROM), IEEE Computer Society Press, Los Alamitos, CA, 1999.

Steven O. Kimbrough received his MS in Industrial Engineering and his PhD in Philosophy from the University of Wisconsin. He holds the rank of Professor at The Wharton School, University of Pennsylvania, where his teaching focuses on Information Technology and on the theory of artificial agents. His active research areas include: computational approaches to belief formation and revision, formal languages for business communication, bargaining and negotiation by artificial agents, and knowledge management (especially content mining). From 1985 to 1995, he was principal investigator for the U.S. Coast Guard's KSS (knowledge-based decision support systems) project. He has been co-principal investigator on the Logistics DSS project, which is part of DARPA's Advanced Logistics Program. He is also currently an investigator in an NSF-funded project exploring how artificial agents in bargaining contexts may be modeled using DNA computing.

D.J. Wu is Associate Professor of MIS at LeBow College of Business, Drexel University, U.S.A. Dr. Wu's research interest include: Models of Capacity Bidding, Contracting with Options and Applications in B2B Exchanges Design; Multi-Agent Gaming and Social Trust in E-Business; Performance Impact and Outsourcing of ERP Systems; Advertising in Electronic Markets; and Formal Models of Electronic Commerce. Dr. Wu's work has appeared in Decision Support Systems, European Journal of Operational Research, International Journal of Electronic Commerce, International Journal of Electronic Markets, among others. Dr. Wu served (or is serving) as a co-guest editor for International Journal of Electronic Commerce as well as (INFORMS) Journal of Group Decision and Negotiation. Dr. Wu is currently co-editing a book on Formal Models of Electronic Commerce, to be published by Springer-Verlag in its series of electronic commerce. Dr. Wu obtained his PhD (in Operations and Information Management) from the Wharton School, University of Pennsylvania, U.S.A.; his BE in Computer Science and BE in Management Engineering from Tsinghua University in Beijing, China. Dr. Wu has been a three-time winner of the SAP University Alliance Award (1998, 1999, 2001). Dr. Wu's most recent work can be found at http://www.lebow.drexel.edu/wu300/research.htm.

Fang Zhong is a PhD student of Management Information Systems at the LeBow College of Business at Drexel University, Philadelphia, Pennsylvania. Her research interests focus on multi-agent gaming and social trust in supply chains as well in E-Business. Fang's work has appeared in International Journal of Electronic Markets, and Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS-34, HICSS-35). Fang obtained her BE in Computer Science from Northern Jiao Tong University in Beijing, China. Prior to joining Drexel's doctoral program in MIS, she had been working in Nortel, China for 2 years.