ResearchGate

See discussions, stats, and author profiles for this publication at:
https://www.researchgate.net/publication/221912334

A Multi-Agent Model for Supply Chain Ordering Management: An Application to the Beer Game

Chapter · April 2011

DOI: 10.5772/16174 · Source: InTech

READS

30

4 authors, including:

Milad Avazbeigi
University of London

16 PUBLICATIONS **51** CITATIONS

SEE PROFILE

Ali Mohaghar
University of Tehran

20 PUBLICATIONS **111** CITATIONS

SEE PROFILE

Available from: Ali Mohaghar
Retrieved on: 05 June 2016

A Multi-Agent Model for Supply Chain Ordering Management: An Application to the Beer Game

Mohammad Hossein Fazel Zarandi[1], Mohammad Hassan Anssari[2], Milad Avazbeigi[3] and Ali Mohaghar[2]
[1]*Amirkabir University of Technology*
[2]*University of Tehran*
[3]*European Centre for Soft Computing, Supported by a Master Grant*
[1,2]*Iran*
[3]*Spain*

1. Introduction

The American Production and Inventory Control Society Dictionary defines the term supply chain (SC) as "the process from the initial raw materials to the ultimate consumption of the finished product linking across supplier–user companies." Supply chain management (SCM) literature covers wide range of areas such as logistics, production, scheduling, facility location, procurement, inventory management, ordering management, and so on. Due to the increasing competition in today's global market, business enterprises are forced to improve their supply chains to reduce inventory cost and enhance customer service levels (Wang & Shu, 2005; Giannoccaro, 2003).

Supply chain ordering management (SCOM), which is the main concern of this book chapter is an integrated approach to determine the ordering size of each actor of SC to the upstream actor aiming to minimize inventory costs of the whole supply chain. SCOM is focused on the demand of the chain aiming to reduce inventory holding costs, lower slacks, improve customer services, and increase the benefits throughout the entire supply chain (Chaharsooghi et al., 2008).

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 (Lee & Whang, 1999; Petrovic, 2008). This may be due to lack of incentives for information sharing, bounded rationality, or possibly the consequence of individually rational behaviour that works against the interests of the group. In a few cases, the researchers' focus is placed on the coordination and integration of inventory policies between more than three stages (Kimbrough et al., 2002; Mahavedan et al., 1997; Petrovic et al., 1999; Wang & Shu, 2005). When there is no coordination among supply chain partners, each entity makes decision based on its own criteria, which results in local optimization as opposed to global optimum.

So called Beer game (Sterman, 1989) is a well-known example of supply chain which has attracted much attention from practitioners as well as academic researchers. Optimal parameters of the beer game ordering policy, when customers demand increases, have been analyzed in two different situations. It has been shown that minimum cost of the chain (under conditions of the beer game environment) is obtained when the players have different ordering policies rather than a single ordering policy (Strozzi et al., 2007). Indeed, most of previous works on order policy of beer game use genetic algorithms as optimization technique (Kimbrough et al., 2002; Strozzi et al., 2007).

One ordering policy based on genetic algorithm under conditions of the Beer game environment was introduced (Kimbrough et al., 2002); we call that GA-based algorithm in this chapter. GA-based algorithm has some degrees of freedom contrary to 1-1 algorithm; In the GA-based algorithm, each actor of chain can order based on its own rule and learns its own ordering policy in coordination with other members with the aim of minimizing inventory costs of the whole supply chain.

One limitation of the GA-based algorithm is the constraint of fixed ordering rule for each member through the time. An attempt to mitigate the problem of fixed ordering rules was initiated in (Chaharsooghi et al., 2008), in this study a reinforcement learning model is applied for determining beer game ordering policy. The RL model enables agents to have different rules throughout the time. In this book chapter we try to extract multiple rules for each echelon in the supply chain using Genetic Algorithm.

This book chapter 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. In this chapter, the supply chain is considered as a combination of various multi-agent systems collaborating with each other. Thus, SCOM can be viewed as a multi-agent system, consisting of ordering agents. Each ordering agent tries to make decisions on ordering size of the relevant echelon by considering the entire supply chain. Agents interact and cooperate with each other based on a common goal. For example, in a linear supply chain with four echelons (as considered in this chapter), there are four ordering agents in SCOM system, each of which is responsible for ordering decisions in its particular echelon. The main objective of ordering agents is to minimize long-term system-wide total inventory cost of ordering from immediate supplier. This is a complex task because of the uncertainty embedded in the system parameters (e.g. customer demand and lead-times) and demand amplification effect (Forrester, 1961), known as 'bullwhip effect'(Lee & Wu, 2006; Fazel Zarandi & Avazbeigi ,2008; Fazel Zarandi et al., 2009).

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. Also the results of the proposed model are compared with two other existing methods in the literature (Chaharsooghi et al., 2008; Kimbrough et al., 2002).

The rest of the book chapter is organized as follows. In section 2, the proposed GA for multi-agent supply chain is described in detail. In section 3, the method is applied on different cases and is compared with other models in the literature. Also in this section, the results are discussed. Finally in the last section, conclusions are summarized.

2. Genetic algorithm with local search for multi-supply chain

2.1 Genetic Algorithm Pseudo Code

Genetic algorithms, originally called genetic plans, were initiated by Holland, his colleagues, and his students at the University of Michigan in the 1970s as stochastic search techniques based on the mechanism of natural selection and natural genetics, have received a great deal of attention regarding their potential as optimization techniques for solving discrete optimization problems or other hard optimization problems (Masatoshi, 2002).

2.2 Representation of ordering policies in GA

In the proposed GA, each rules set (ordering policy) is encoded using binary system. In Fig. 2, the encoding schema is demonstrated. Each echelon in the supply chain has *w* rules. All rules are represented in binary system with *NumberOfBytes* cells which *NumberOfBytes* is a parameter of the model. The first cell in each echelon rule, stores the sign of the rule. 1 is for positive and 0 is for negative. These cells are distinguished with grey colour. The next *NumberOfBytes-1* bits represent how much to order.

1. Initialization. A certain number of rules (Ordering Policies) are randomly generated to form generation 0.
2. Pick the first binary rule from the current generation and decode the chosen rule to obtain the decimal ordering rules.
3. Agents play the Beer Game according to their current decimal rules.
4. Repeat step (3), until the game period (say 35 weeks) is finished.
5. Calculate the total 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 (3), (4) and (5) until the performance of all the rules in the current generation have been evaluated.
7. Use GA with local search to generate a new generation of rules and repeat steps (2) to (6) until the maximum number of generation is reached

Fig. 1. The pseudo code of the proposed GA

W rules –instead of one rule– enable each agent to have a more adaptive and dynamic behaviour. The effect of different *W*'s on system objective function is also studied in next sections.

Window Basis (w)	*Echelon 1 (Agent 1)*	*Echelon 2 (Agent II)*	*Echelon 3 (Agent III)*	*Echelon 4 (Agent IV)*
Rule 1	*1 1 0 1 0*	*1 0 0 0 0*	*1 0 0 1 0*	*0 1 1 0 0*
Rule 2	*0 0 1 0 0*	*1 1 1 1 1*	*1 1 0 0 0*	*1 0 0 1 1*
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Rule w-1	*1 0 1 0 0*	*1 0 1 0 1*	*0 1 0 1 0*	*0 1 0 1 0*
Rule w	*1 0 0 0 1*	*0 1 0 1 0*	*1 1 0 1 0*	*0 1 0 1 0*

Fig. 2. Encoding Schema

When it is needed to run a supply chain using a specific ordering policy, first it is mandatory that the chromosome of the ordering policy –similar to that shown in Fig. 2– decoded to decimal system. Two examples of decoding procedure are shown in Fig. 3.

1	*1*	*0*	*1*	*1*	→	*+13*
0	*1*	*1*	*0*	*0*	→	*-3*

Fig. 3. Decoding Example

2.3 Objective 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. We begin with the needed notation. In the MIT Beer Game:

- N is the number of players and is 4
- $IN_i(t)$ is the net inventory of player i at the beginning of period t
- $C_i(t)$ is the cost of player i at period t
- H_i is the 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 is the penalty/backorder cost of player i, per unit per period (e.g., in the MIT Beer Game, US$2 per case per week)
- $S_i(t)$ is the new shipment player i received in period t
- $D_i(t)$ is the 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 $IN_i(t) \geq 0$, then $C_i(t)=IN_i(t) \times H_i$ else $C_i(t)=|IN_i(t)| \times P_i$, where $IN_i(t)=IN_i(t-1)+S_i(t)-D_i(t)$ and $S_i(t)$ is a function of both information lead time and physical lead time. The total cost for the supply chain after M periods is

$$\sum_{i=1}^{N}\sum_{t=1}^{M} C_i(t) \tag{1}$$

2.4 GA operators

1) Selection Operator: In the proposed GA, for selection of the chromosomes from the current population, the tournament method is chose. In this method, at each time two chromosomes are selected randomly from the current population and then the chromosome with the minimum cost will be selected as a member of the next population. This process continues until the required chromosomes are chosen for the new population.

2) Mutation Operator: Mutation in the proposed GA, includes the replacement of the zero-cells with one-cells and vice versa. The Mutation type indicates that how many cells should change.

3) Crossover Operator: Crossover operator randomly chooses $2*M$ columns (M: Crossover Type) from the randomly chosen chromosome from the current population. Then, the position of two columns changes in the selected chromosome.

4) Rearrangement Operator as Local Search of GA: Rearrangement operator, first randomly choose a chromosome from the chromosomes selected by the Selection method, then choose two cells randomly and change the positions of those cells randomly. If the new chromosome had a smaller cost function, then the operator adds the new chromosome to the new population. Otherwise, the operator repeats the process until an improvement occurs.

3. Results and conclusions

To validate the proposed system, some experiments are designed. The experiments and their results are summarized in Tables 1 and 2. In the following, each experiment is described in detail.

Experiment	Number Of Bytes	W	Best Ordering Policy	Lead Time
1	4	1	[0,0,0,0]	2 for all echelons
2	5	1	[0,1,2,2]	2 for all echelons
3	5	2	[1,0,6,0;1,8,4,9]	2 for all echelons
4	5	4	[0,2,12,4;4,8,5,8;0,4,4,8;0,9,3,2]	2 for all echelons
5	5	1	[0,0,1,0]	Unifrom [0-4]
6	5	2	[0,0,1,4;0,0,2,0]	Unifrom [0-4]
7	5	3	[0,0,1,0;0,0,2,0;0,1,2,4]	Unifrom [0-4]
8	5	4	[0,0,0,7;0,0,9,9;0,6,4,0;0,0,0,1]	Unifrom [0-4]
9	5	5	[0,0,10,1;0,0,4,8;0,0,2,2;0,1,6,5;0,01,3]	Unifrom [0-4]
10	5	4	[0,1,2,15;0,3,8,0;0,2,4,10;0,1,8,3]	Unifrom [0-4]
11	5	4	[0,0,4,0;0,0,6,8;0,0,4,4;0,0,9,0]	Unifrom [0-4]
12	5	2	[0,0,3,7;0,0,5,3]	Unifrom [0-4]
13	4	1	[1,1,1,1]	2 for all echelons
14	4	2	[0,1,4,2;0,5,2,3]	2 for all echelons
15	4	3	[0,3,0,5;0,2,5,1;0,4,5,3]	2 for all echelons
16	4	4	[0,1,3,3;1,3,5,6;0,2,6,6;0,0,7,3]	2 for all echelons

Table 1. Best ordering policies achieved by the method

In the first experiment, the performance of the multi-agent system is tested under deterministic conditions. 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). 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. As the result shows (*[0, 0, 0, 0]*) we found that the artificial agents can learn the 1–1 policy consistently.

In the second experiment, we explored the case of stochastic demand where demand is randomly generated from a known distribution, uniformly distributed between *[0, 15]*. Lead time for all echelon is a constant value through the time and is 2. In this case the model is compared with (Kimbrough et al., 2002) as the result show, the model outperforms Kimbrough's model.

In experiment 3 and 4, the influence of window basis (w) on the objective function of the problem is studied. As it can be seen, more number of rules leads to smaller values of total cost. This supports the idea that more number of rules enables the agents to be more adaptive and flexible to the environmental changes.

Experiment	Demand	Best Total Cost	Worst Total Cost	Average Total Cost	1-1 Best Total Cost	GA Best Total	RL Best Total Cost
1	All the demands are 8 except 4 first weeks which is 4	400	400	400	400	400	-
2	Uniform [0-15]	**<u>1536</u>**	1586	1561	3890	1820	-
3	Uniform [0-15]	1514	1570	1548	-	-	-
4	Uniform [0-15]	1458	1545	1487	-	-	-
5	Uniform [0-15]	**<u>2124</u>**	2124	2124	7463	2555	2417
6	Uniform [0-15]	2030	2030	2030	-	-	-
7	Uniform [0-15]	2010	2067	2030	-	-	-
8	Uniform [0-15]	1979	2010	1992	-	-	-
9	Uniform [0-15]	2056	2234	2134	-	-	-
10	Uniform [0-15]	**<u>1667</u>**	-	-	5453	3109	3169
11	Uniform [0-15]	**<u>1896</u>**	-	-	8397	4156	4038
12	Uniform [0-15]	**<u>1967</u>**	-	-	7826	4330	4205
13	*F(x) = |Max Demand*sin(x.Π/Period)|*	793.715	793.715	793.715	-	-	-
14	*F(x)= |Max Demand*sin(x.Π/Period)|*	744.826	774.237	762.079	-	-	-
15	*F(x)= |Max Demand*sin(x.Π/Period)|*	779.689	799.455	789.174	-	-	-
16	*F(x)= |Max Demand*sin(x.Π/Period)|*	644.872	699.865	668.943	-	-	-

Table 2. Comparison of models with other models in the literature

In experiments 5 to 9, the model is evaluated under more challenging conditions. The demand and lead time are both nondeterministic and have distribution function uniform *[0, 15]* and *[0, 4]* respectively. The results are compared with 1-1 ordering policy (Chaharsooghi et al., 2008; Kimbrough et al., 2002). The best objective function achieved by the model is 1979 which is much smaller than (Chaharsooghi et al., 2008) results (2417). Again the positive effect of window basis can be seen as the number of window basis increases to some extent the best objective function value decreases. A trend stops at window basis equal to 5. This can be due to the exponential growth in the search space, which makes the search process so complex for GA (with the current encoding schema 25*5*4 = 2100 possible solutions exist).

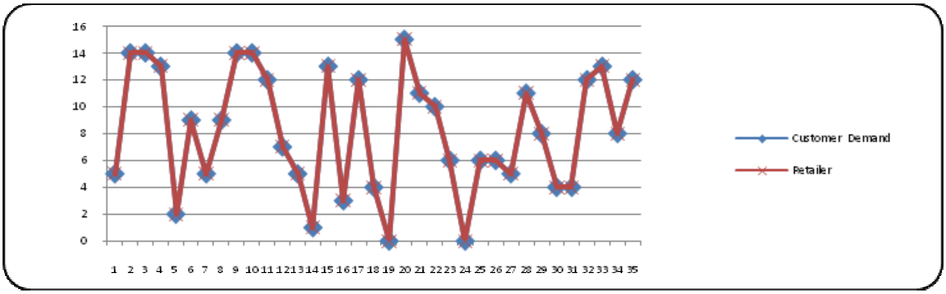

Fig. 4. Customer Demand in comparison with retailer

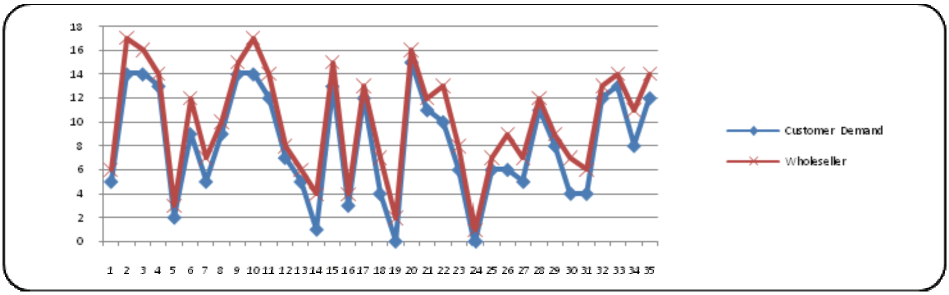

Fig. 5. Customer Demand in comparison with wholeseller

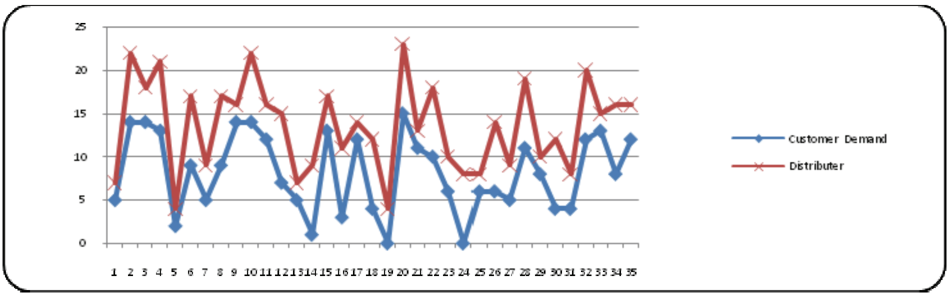

Fig. 6. Customer Demand in comparison with Distributer

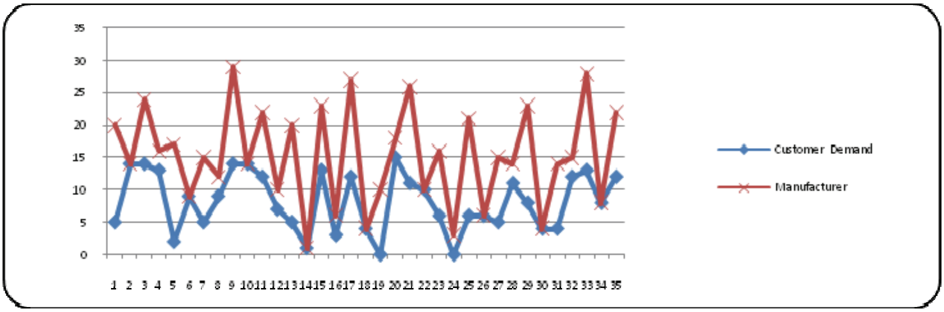

Fig. 7. Customer Demand in comparison with manufacturer

In experiments 10, 11 and 12, the proposed window basis model is again compared with 1-1 ordering policy. 1-1 ordering policy is described in (Kimbrough et al., 2002; Sterman, 1989). In all cases, the model has a better performance. The ordering values of four echelons base on the best ordering policy achieved by the model for experiment 10 are depicted in fig. 4, 5, 6 and 7.

In the last 4 experiments, the model is applied on a periodic function with the function of

$$F(x)=|MaxDemand*\sin(x.\Pi/Period)| \quad (2)$$

and the impact of different window basis is studied. in this function Max Demand is 7 and period is 8. As table 2 shows, models with window basis with the 2 multiples have a better performance.

It should be noted that in the first 12 experiments, the genetic population is 100, the number of generation is 400, the mutation, crossover and the rearrangement ration are 0.2. In the last four experiments, the genetic population is 300, the number of generation is 400, the crossover and mutation ratio are 0.3 and the rearrangement ratio is 0.2.

4. Conclusion

In this a new intelligent multi-agent system is proposed for determination of the best ordering policy in order to minimize the cost of supply chain.

The model is compared with previous models in the literature and as the results show, the model outperforms all the previous models.

The best ordering policy is obtained by a new genetic algorithm which is equipped with some local searches. One limitation of the previous presented GA-based algorithms is the constraint of fixed ordering rule for each member through the time. To resolve this problem a new concept –window- is introduced in this book chapter. Application of the window basis enables the agents to have different ordering rules throw the time. Experiment results prove that the new multi-agent system is capable of finding patterns in nondeterministic and periodic data both.

5. References

Chaharsooghi, S.K., Heydari, J. & Zegordi, S. H. (2008). A reinforcement learning model for supply chain ordering management: An application to the beer game. *Decision Support Systems*. Vol. 45, page numbers 949–959.

Fazel Zarandi, M.H. & Avazbeigi, M. (2008). A Fuzzy Logic Approach to prove Bullwhip Effect in Supply Chains, *Proceedings of AMERICA*, Toronto, ON., Canada.

Fazel Zarandi, M.H., Avazbeigi, M. & Turksen, I. B. (2009). An intelligent fuzzy Multi-Agent System for reduction of bullwhip effect in supply chains, *Proceedings of NAFIPS09*, Cincinnati, Ohio, USA.

Forrester, J. W. (1961). *Industrial Dynamics*, Cambridge, MIT Press, Massachusetts.

Giannoccaro, I., Pontrandolfo, P. & Scozzi, B. (2003). A fuzzy echelon approach for inventory management in supply chains. *European Journal of Operational Research,* Vol. 149, page numbers 185–196.

Kimbrough, S.O., Wu, D. J. & Zhong, F., 2002. Computers play the beer game: can artificial agents manage supply chains?. *Decision Support Systems*, Vol. 33, page numbers 323–333.

Lee, H. & Whang, S. (1999). Decentralized multi-echelon supply chains: incentives and information. *Management Science*, Vol. 45(5), page numbers 633– 640.

Lee, H. T. & Wu, J. C. (2006). A study on inventory replenishment policies in a two echelon supply chain system. *Computers & Industrial Engineering*, Vol. 51, page numbers 257–263.

Masatoshi, S. (2002). *Genetic Algorithms and Fuzzy Multiobjective Optimization,* (Operations Research Computer Science Interfaces Series), Massachusetts: Kluwer Academic Publishers.

Mahadevan, S., Marchalleck, N., Das, K. T. & Gosavi, A. (1997). Self-improving factory simulation using continuous-time average-reward reinforcement learning. *Proceedings of the 14th International Conference on Machine Learning*, page numbers 202–210.

Petrovic, D., Roy, R. & Petrovic, R. (1999). Modeling and simulation of a supply chain in an uncertain environment. *European Journal of Operational Research,* Vol. 109, page numbers 299–309.

Petrovic, D., Xie, Y., Burnham, K. & Petrovic, R. (2008). Coordinated control of distribution supply chains in the presence of fuzzy customer demand. *European Journal of Operational Research,* Vol. 185, page numbers 146–158.

Sterman, J. (1989). Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. *Management Science,* Vol. 35 (3), page numbers 321–339.

Strozzi, F., Bosch, J. & Zaldívar, J. M. (2007). Beer game order policy optimization under changing customer demand. *Decision Support Systems*, Vol. 42, page numbers 2153–2163.

Wang, J. & Shu, Y.-F. (2005). Fuzzy decision modeling for supply chain management. *Fuzzy Sets and Systems,* Vol. 150, page numbers 107–127.

Wang, J. & Shu, Y.-F. (2007). A possibilistic decision model for new product supply chain design, *Journal of Operational Research*, Vol. 177, page numbers 1044–1061