



## The Investigation of Supply Chain Optimization based on Artificial Network and Production Firms Techniques

Reza Emami

Master of Industrial Engineering (System Management), Islamic Azad University, Najafabad, Iran  
[emamireza14@yahoo.com](mailto:emamireza14@yahoo.com)

**Abstract:** The supply chain optimization is a difficult problem to solve in the context of distributed information across different members and dynamic changes in the structure and content of the information environment with multidisciplinary decisions employees' decision making at different levels. In this research work, an approach to the dynamic optimization of local decisions to assure global optimization in supply chain performance is developed within the frameworks of a Collective Intelligence and Multi-Agent Systems. As a COIN, we mean a large multi-agent system where there is no centralized control and communication, but also, there is a global task to complete: the global supply chain optimization. The proposed framework is focused on the interactions at local and global levels among the agents in order to improve the overall supply chain business process behavior. Besides, learning consists of adapting the local behavior of each agent (micro-learning) with the aim of optimizing a given global behavior (macro-learning). Reinforcement learning algorithms are used at the local level, while generalization of the Q-neural algorithm is used to optimize the global behavior. The framework is implemented over JADE agent platform. The experimental results demonstrate that this problem is a good experimental field for the investigation and application of the COIN theory.

[Reza Emami. **The Investigation of Supply Chain Optimization based on Artificial Network and Production Firms Techniques**. *N Y Sci J* 2020;13(9):48-52]. ISSN 1554-0200 (print); ISSN 2375-723X (online). <http://www.sciencepub.net/newyork>. 9. doi: [10.7537/marsnys130920.09](https://doi.org/10.7537/marsnys130920.09).

**Keywords:** Supply chain network, multi-agent system, optimization, collective intelligence, simulation.

### 1. Introduction

In order to make good decisions within a SCN, a manufacturer needs to coordinate local activities with those of upstream suppliers and downstream customers under uncertainty and imprecision in a very dynamic environment. Other entities in the chain are faced with a similar problem. The ability to manage the complete supply chain network (SCN) and to optimize decisions is increasingly being recognized as a crucial competitive factor [17]. Unfortunately, as the scope of supply chain management is extended, the underlying optimization problems grow dramatically in size. This is particularly important for operational problems, where the solution to the optimization problem specifies a short-term assignment or schedule of resource use. As a consequence, logistics gets a new focus on optimization of the production process in a very dynamic environment [12]. On the other hand, the availability of real-time status information is creating a need for 'Re -optimization' models and methods that can quickly repair a plan in response to changes in input data.

Also, most of these approaches reflect today's supply chains model, which is essentially static, relying on long-term relationships among key trading

partners. Recently several new optimization paradigms and approaches have been proposed. These new paradigms include both analytical methods (based on semi-definite optimization and computational differentiation) and simulation based optimization [5, 6, 7]. Though there are many solutions and techniques for local optimization (e.g. planning and scheduling systems, inventory management systems, market trading optimization systems, etc.), usually these algorithms spend a lot of time finding the most appropriate solution and their decisions do not assure the overall business optimization at the global level because of the conflicts between the local goals of the different entities [13, 14]. Some of the underlying mathematical issues involve development of fast algorithms to approximate the solutions to very large-scale problems, as well as methods for combining models of different types to form hybrid models. The issue of real-time updates to solutions has been addressed for a few specific applications, but no underlying theory has been developed. The size and difficulty of these problems make them impossible to solve with traditional optimization methods, and require the

development of specialized combinatorial optimization techniques.

More flexible and dynamic practices offer the prospect of better matches between suppliers and customers permitting them react to a changing environment by adapting their operations [30]. This requires an approach, which can be flexible enough to accommodate imprecise linguistic data as well as precise numerical data, and which yields solutions that will provide compromise among different parties' objectives. To this end, there are few research works on theoretical treatment and different soft computing, machine learning and agent techniques developed for dynamic supply chain modeling and optimization in vague and uncertain environments [10, 11, 18, 24, 25]. In this paper, the problem of dynamic optimization of the Supply Chain Network (SCN) is addressed within the context of the Collective Intelligence (COIN) theory as an extension of Dynamic Programming models [29] and the adaptation of the Q-neural algorithm [21]. Within this framework, a SCN is a large Multi-agent System (MAS) where:

- One of the main objectives is the decentralization of the control and communication.
- Each agent of the SCN is represented as an agent with autonomous behavior and a local utility function.
- Agents model different SCN entities, the exchange of message and commodity objects among these agents emulates the information and material flows.
- The learning process consists of adapting the local behavior of each agent with the aim of optimizing a given global SCN behavior.
- The agents execute Reinforcement Learning (RL) algorithms at the local level while generalization of Q-neural algorithm is used to optimize the global behavior.

## 2. Data and Methodology

The Dynamic Programming theory forms the base of the RL methods used within the proposed approach [3, 4]. Dynamic Programming (DP) is a technique which addresses the problems which arise in situations where decisions are taken by steps, and the result of each decision is partially foreseeable before the remaining decisions are taken. An important aspect to consider in DP is the fact that the decisions cannot be made separately [22]. For example, within the SCN the desire to obtain a reduced cost in the present must be balanced against the desire to induce low costs in the future. This constitutes a Credit Assignment Problem since one must give credit or culpability to each decision. For an optimal planning, it is necessary to have an effective

compromise between the immediate costs and the future costs. More precisely, Dynamic Programming focuses on the question: How can a system learn to sacrifice its short-term performance to improve its long-term performance? To answer this, DP relies on the application of the Bellman's principle of optimality defined as follows: An optimal strategy  $p^*$  is such as, whatever the initial state  $x(0) = i$  and the initial decision, the remaining decisions must constitute an optimal sub-strategy, with regard to the state resulting from the first decision. In other words, an optimal strategy  $p^*$  can be constituted only of optimal sub-strategies:

$$\{\mu_{x(0)}(0)^*, \mu_{x(1)}(1)^*, \dots, \mu_{x(K)}(K)^*\}$$

where  $K$  is the horizon in time (or steps number). The Bellman equation presented below permits finding the optimal value function to solve the Markov Decision process [20]:

In other words, the global optimization of the objective function is replaced by a sequential optimization that consists in optimizing each stage of decision (or period), one after the other, but by taking into account the former decisions which were made previously and the remaining decisions [9].

Reinforcement Learning (RL) extends the ideas of the DP to treat more complete and ambitious goals of Artificial Intelligence (AI). Reinforcement learning, instead of being like DP based only on the resolution of the problems of optimal control, is an aggregate of ideas from psychology, statistics, cognitive science and computer science [12]. To compute the optimal strategy, DP assumes perfect knowledge of the environment model (e.g. transition probabilities between the states of the environment and the costs (rewards/punishments) which the agent receives from this environment). The first question addressed by the DP was how to compute the optimal strategy with the minimum data-processing computation, by supposing that the environment can be perfectly simulated, without the need for direct interaction with it. The new trend in the methods of RL is the assumption of a limited (or even the absence of) knowledge about the environment model at the beginning and the prediction of rewards/punishments. Moreover, instead of moving in a mental model internal space, the agent must act in the real world and observe the consequences of its actions. In this case, we are interested in the number of real world actions the agent must carry out to move towards an optimal strategy, rather than with the number of algorithmic iterations necessary to find this optimal strategy. The Bellman's equation makes it possible to define an optimal strategy [16]. The systems which learn while interacting with a real environment and by observing their results are called online systems. On the other

hand, the systems which learn separately, without interacting with a real environment and with an internal model of this environment are called off-line systems. In this work we are developing the SCN model as an on-line system.

### 3. Results

Reinforcement Learning answers the question: how to make the mapping between the states (of an environment whose model is not known entirely) and the actions of an agent (which interacts with this environment online) so as to maximize/minimize a numerical signal of reward/punishment? In other words, within the context of the SCN this permits looking for the optimality of local decisions under the constraints of the optimal behavior of the whole SCN. The fact of learning by trial and error, and of having a delayed reward/punishment which will affect the future behavior of the agent are the two most important characteristics which differentiate it from other types of learning methods. A dilemma which arises in the field of RL and not necessarily with other types of learning is the trade-off which exists between the exploitation phase and the exploration phase. The agent must exploit the knowledge obtained until now to select the actions which brought it a high reward. But, at the same time, it must explore all the possible actions in its current state, in order to select an action which can bring it a higher reward than the actions carried out in the past. This dilemma has been studied by several mathematicians [12].

One of the most important advances in the field of RL was the development of Q-learning [28], an algorithm which follows an off-line strategy [12]. In this case, the value-action function learned,  $Q$ , approximates the optimal value-action function  $Q^*$  in a way independent of the strategy followed. In state  $x(t)$ , if the  $Q$ -values represent the environment model in an exact way, the best action will be that which has the most/less important value (according to the case) among all possible actions. The  $Q$ -values are learned by using an update rule which uses a reward  $r(t+1)$  calculated by the environment and a function of  $Q$ -values of the reachable states by taking the action  $a(x(t))$  in state  $x(t)$ . The update rule of Q-learning is defined by:

The term "supply chain" has been used since the 1980s to describe the whole spectrum of operations in almost every manufacturing industry; from purchasing of raw material, through transformation production processes, to distribution of the finished inventory to customers. As the complexity increases a supply chain is well depicted as a network of suppliers, manufacturers and customers. This work proposes a model of the SCN in the framework of the COIN theory. As defined in [8], SCN members are believed to have common functions. These common functions

are handling of incoming and outgoing flows, flow transformation and control. Materials in the SCN are represented as objects forming part of the environment. Therefore, every agent can change or influence these environment objects. The details of the objects are stored as attributes. In our approach, an agent can represent any member of the SCN. Each agent has a local utility function and handles a  $Q$ -table, which contains perceived information about the environment (both environment objects and neighbor agents). This generic scheme can be applied both to the members of the SCN and to the components of each member depending on the necessary level of decomposition. A network of agents represents the entire SCN. This representation enables dynamic SCN simulation within the COIN framework.

The relationships between the agents in the SCN are defined by:  $R = \{r_1, r_2, r_3, \dots\}$ . Agents known to the current agent form the list of his neighbors:  $N = \{n_1, n_2, n_3, \dots\}$ . In the case of the linear model, only agents from the nearest tier are included in this list. For each neighbor agent the following parameters are considered: a) its relationship to the current agent (customer, supplier), b) the nature of the agreement that governs the interaction (production guarantees) and c) the inter-agent information access rights (the agent's local state to be considered during the decision-making process).

The priorities of every agent are represented by  $Q = \{q_1, q_2, q_3, \dots\}$ . These priorities can help in sequencing incoming messages for processing.

- The local utility function (LUF) is represented as the Q-learning equation (2).
- The set of control elements:  $C = \{c_1, c_2, c_3, \dots\}$ . A control element is invoked when there is a decision to be made while processing a message. For example, in order to determine the next destination in the transport of materials, a routing-control algorithm would be utilized.
- Every agent has a message handler that is responsible for sending and receiving different messages to facilitate communication among the agents.

To address the SCN optimization problem, the adaptation of the Q-neural algorithm [21] is proposed and described. The behavior of the Q-neural was inspired by the Q-routing algorithm operation [15], the theory of COIN, including the algorithms based on the behavior of the colonies of ants. The learning is done at two levels: initially, at the agent's level locally updating the  $Q$ -values by using a RL rule, then, globally at system level by the utility function's adjustment. The control messages allow updating knowledge of the SCN entities by updating the  $Q$ -values, which are approximated by using a function

approximate (look-up table, neural network, etc.). In Q-neural, there are 5 types of control messages:

- An 'environment-message' (flag-ret=1) generated by an intermediate PA after the reception of a raw material if the interval of time  $w$  has already passed.
- An 'ant-message' (flag-ret=2) generated by the DWA according to the interval of time  $w$  ants when a final product arrives at the warehouse.
- An 'update-message' (flag-ret=3) generated in the planning phase every  $e$  update seconds to ask the neighboring PA their estimates about the operations of the product.

When an environment-message arrives at the DWA, an ant is sent in return if the period of time  $w$  ants has already passed. This ant exchanges the statistics obtained on its way and it allows the environment information communication among the agents. When it arrives at the storage of raw materials, it dies. The ant updates the Q-value of each PA through which the raw material passed before arriving at the DWA. In some cases, different agents from the same tier can have the same best estimate (prefer the same route). If they act in a greedy way, congestion occurs in the queue. To avoid congestions, an agent must sacrifice its individual utility and to use another route. In order to address this problem a punishment algorithm is developed forcing an agent who receives a punishment message to calculate the second best estimate.

## Conclusion

Today's challenge is to optimize the overall business performance of the modern enterprise. In general, the limitations of traditional approaches to solving the problem of dynamic global optimization of the SCN are due to the fact that these models do not correspond to the reality because of incomplete information, complex dynamic interactions between the elements, or the need for centralization of control and information. Most of heuristic techniques on the other hand, do not guarantee the overall system optimization. COIN optimization algorithm described above is implemented within a more general multi-agent framework for supply chain modeling and optimization. This framework represents complex dynamic interactions among supply chain members, accounts for demand uncertainty, validates and if necessary allows modification of configuration, optimization and coordination results. It can be used to improve decision-making within a wide range of problems in various supply chain scenarios. The relevant decisions can be classified into three categories: strategic, operating, and control. Strategic decisions such as selecting the supply chain participant have long-term significance. Operating

decisions refer to decisions about production to meet demand. Finally, control decisions are concerned with problems in execution. This can be classified as disruption management (like situations when a certain machine in the shop floor fails). Though being more oriented on the solution of the second type problems, the framework can handle the other categories as well. In the case study we show how simulation results can be used for the SCN configuring based on the performance analysis. On the other hand, the algorithm of dynamic optimization suits well for just-in-time decision making. At the stage of SC template configuration a user can specify three types of parameters: (i) define operational parameters and specifications, (ii) determine structural requirements to the SC, and finally (iii) define performance goals. Operational parameters include inventory control policies, production capacities, order lead-time, etc. For a selected type of a supply chain, the historical production data and BOM information can be obtained from the ERP system (at the moment the interface to the Excel based MRP is implemented). All the products composing a demand are broken down to their component parts according to the BOM. Demand parameters for each component are obtained as a result of the forecasting based on estimation of the patterns from the historical data. We use perceptual forecasting module from the Fuzzy Toolbox Library developed by the authors for time series analysis.

## Corresponding Author:

Reza Emami

Master of Industrial Engineering (System Management), Islamic Azad University, Najafabad, Iran

[emamireza14@yahoo.com](mailto:emamireza14@yahoo.com)

## References

1. Batyrshin, I., and L. Sheremetov, Perception Based Time Series Data Mining with MAP Transform. MICAI 2005: Advances in Artificial Intelligence. In A. Gelbukh, A. Albornoz, H. Terashima-Marín, eds., Lecture Notes in Computer Science, Springer Verlag, 3789: 514 – 523, 2005.
2. Bellifemine, A. Poggi, and G. Rimassa, JADE - A FIPA-compliant Agent Framework. In Proc. PAAM'99, London, UK, 1999, pp. 97-108.
3. Bellman, R., On a routing problem. Quarterly of Applied Mathematics, 1958.
4. Bertsekas, D., and J. N. Tsitsiklis, Neuro-Dynamic Programming. Athena Scientific, Belmont, Massachusetts, 1996.

5. Bertsimas, A. T hiele: A Robust Optimization Approach to Supply Chain Management. IPCO 2004: pp. 86-100.
6. Bischof, C., and A. Griewank, Computational differentiation and multidisciplinary design, in Inverse Problems and Optimal Design in Industry, H. England J. McLaughlin, eds., Stuttgart, Teubner Verlag, 1994, pp. 187--211.
7. Chandra, C., and J. Grabis, Supply Chain Configuration Using Simulation Based Optimization, Proc. of 13th Annual Industrial Engineering Research Conference, May 15-18, 2004, Houston, Texas.
8. Chandra, S. Kumar, and A. V. Smirnov, E-Management of Scalable Cooperative Supply Chains: Conceptual Modeling and Information Technologies Framework, Human Systems Management, Vol. 20, No. 2, pp. 83-94, 2001.
9. Chevalier A. La programmation dynamique. Dunod Décision, 1977.
10. Dingwei, F. Shu-Cherng, and H.L.W. Nuttle, Soft Computing for Multi-Customer Due-Date Bargaining, IEEE Transactions on Systems, Man and Cybernetics, Part (c), vol. 29(4), pp. 566-575, November 1999.
11. Eschenbacher, P. Knirsch and I. J. Timm, Demand Chain Optimization by Using Agent Technology, in Proc. of the IFIP WG 5.7 International Conference on Integrated Production Management, Tromso, Norway, June 28-30, pp. 285-292, 2000.
12. Gaither, N., and G. Frazier, Operations Management, Southwestern Thomson Learning, Cincinnati, Ohio, 2002.
13. Hoover, E. Eloranta, J. Holmström and K. Huttunen, Managing the Demand-Supply Chain: Value Innovations for Customer Satisfaction, John Wiley & Sons, New York, 2001.
14. Julka, R. Srinivasan, I. Karimi, Agent-based supply chain management-1: framework, Computers & Chemical Engineering, vol. 12, no. 12, pp. 1755 – 1769, 2002.
15. Littman, M., and J. Boyan, A distributed reinforcement learning scheme for network routing, in J. Alspector, R. Goodman, and T. X. Brown, eds., Proc. of the 1993 International Workshop on Applications of Neural Networks to Telecommunications, pp. 45--51. Lawrence Erlbaum Associates, Hillsdale NJ, 1993.
16. Mitchell, T., Machine Learning. McGraw-Hill, 1997.
17. Nah, M. Rosemann and E. Watson, Guest editorial: E -business Process Management, Business Process Management Journal, 10(1), pp. 1-15, 2004.
18. Powell, J. A. Shapiro and H. P. Simão, An Adaptive Dynamic Programming Algorithm for the Heterogeneous Resource Allocation Problem, Transportation Science, vol. 36, no. 2, pp. 231 – 249, May 2002.
19. Puterman, M.L., Markov Decision Processes. Wiley-Interscience publication, 1994.
20. Rocha Mier, L., Learning in a Neural Collective Intelligence: Internet packet routing application, PhD Dissertation, National Polytechnic Institute of Grenoble (INPG), France, 2002.
21. Sakarovitch, M., Optimisation Combinatoire. Hermann, 1984.
22. Sheremetov, L., L. Rocha, I. Batyrshin. Towards a Multi-agent Dynamic Supply Chain Simulator for Analysis and Decision Support. IEEE NAFIPS-05 Annual Conference: Soft Computing for Real World Applications Ann Arbor, Michigan, June 22-25, 2005.
23. Smirnov, A., L. Sheremetov, N. Chilov, and J. Romero-Cortes, Soft-computing Technologies for Configuration of Cooperative Supply Chain, Int. J. Applied Soft Computing. Elsevier Science, vol. 4(1), pp. 87-107, 2004.
24. Swaminathan, J.M., S.F. Smith, and N.M. Sadeh, Modeling Supply Chain Dynamics: A Multiagent Approach, Decision Sciences, 29(3), pp. 607-631, 1998.
25. Sutton, R. and A. Barto, Reinforcement Learning: An Introduction, The MIT Press, 1998.

9/25/2020