



# Journal of Applied Sciences

ISSN 1812-5654

**science**  
alert

**ANSI***net*  
an open access publisher  
<http://ansinet.com>

## An Agent-Based Framework for Building Decision Support System in Supply Chain Management

A. Kazemi and M.H. Fazel Zarandi  
Department of Industrial Engineering, Amirkabir University of Technology,  
P.O. Box 15875-4413, Tehran, Iran

---

**Abstract:** In this study, two scenarios are presented for solving Production-Distribution Planning Problem (PDPP) in a Decision Support System (DSS) framework. In the first scenario, a Traditional Decision Support System (TDSS) is presented for PDPP and a Genetic Algorithm (GA) is used for solving it. In the second scenario, a Multi-agent Decision Support System (MADSS) is considered for PDPP and three algorithms are used for solving it: Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA). Then an algorithm is suggested by using multi-agent system and A Teams concept. The obtained results reveal that the use of MADSS delivers better solutions to us.

**Key words:** Agent-based systems, production-distribution planning, decision support systems, supply chain, A Teams

---

### INTRODUCTION

Supply Chain Management (SCM) is the management of material and information flows both in and between facilities, such as vendors, manufacturing and assembly and distribution centers in a network format (Thomas and Griffin, 1996).

The main objective of SCM is to integrate all the organizations in a supply chain, so as to deliver products to customers with minimal total cost of the whole system (Chan *et al.*, 2006)

Production-distribution in supply chains can take many forms. In general, there are two distinctive models: production and distribution. They must be designed such that to be linked together and considered as a production-distribution model in supply chain. These models are operationally connected and closely related with each other (Lee and Kim, 2000).

The Production-Distribution Planning Problem (PDPP) involves the determination of the best configuration regarding location, size, technology content and product range to achieve a firm's long-term goals (Dasci and Verter, 2001).

Ideally, a good distribution network design can help companies to have better value-addition, reduced costs and increased customer service level by determining optimal links between each node and the traffic flow

routine (Lumsden *et al.*, 1999; Milgate, 2001; Stank and Goldsby, 2000; Chan and Chung, 2005).

In recent years providing effective decision support has become more important due to: increased employee empowerment, heightened requirements for speed and quality in managerial decisions and increased accessibility to a vast amount of information through electronic networks (Lerch and Harter, 2001).

Research in Decision Support System (DSS) design has been shifting from a traditional toolbox system organization towards a more collaborative and active paradigm (Vahidov and Fazlollahi, 2004). In particular, researchers have stressed the importance of providing higher cognitive level support (Radermacher, 1994) including encouragement of divergent processes, generation of decision alternatives (Fazlollahi and Vahidov, 2001) and automated critiquing of alternatives (Fischer and Mastaglio, 1991). However, past researches have not addressed the issue of the integration of piece-meal prescriptions for building more effective DSSs under a unified approach (Vahidov and Fazlollahi, 2004).

The rapidly growing areas of intelligent agents (IAs) and multi-agent systems (MASs) offer the opportunity for building more effective systems using a unified approach based on sound theories. The promise that agent-based technologies holds for enhancing DSS capabilities is recognized by many researchers (Vahidov and Fazlollahi,

2004). IAs possess attractive features including: autonomy, proactiveness, reactivity, reasoning capability, social ability (interaction with the environment, user and other agents) and incorporation of human-like features e.g., beliefs, desires, intentions, commitments, motivations, etc. (Vahidov and Fazlollahi, 2004).

In this study, we propose an agent-based framework for building decision support system to solve the Production-Distribution Planning Problem (PDPP) for a supply chain management. This problem consists of three types of multi-stage, multi-product, production and distribution planning sub-models. We use two scenarios for solving this model. In the first scenario, the problem is as a Traditional Decision Support System (TDSS). Also in the second scenario, the problem takes form an agent-based framework for building decision support system in which teams of autonomous agents (A Teams) are used such that each agent acts as a search method to handle its tasks.

#### **PRODUCTION-DISTRIBUTION SYSTEMS IN SUPPLY CHAIN MANAGEMENT**

Modeling and analysis of production-distribution systems in supply chain has been an active area of research for many years. Chon and Lee (1998) study production-distribution integrated systems under stochastic demands. They present a supply chain model that incorporates raw materials, intermediate and final product plants, distribution centers, warehouses and customers. Their model shows interactions of multi-stage production-distribution systems. At each stage, sub-models are defined and control policies are implemented. Then, a heuristic optimization procedure is introduced and some results are discussed. Thomas and Griffin (1996) define three categories of operational coordination: buyer and vendor, production and distribution and inventory and distribution. Vidal and Goetschalckx (1997) review the strategic production-distribution models. They focus on global supply chain models with emphasis on mixed integer programming models. Beamon (1998) provides a focused review of literature in the area of multi-shop supply chain design and analysis and suggests four categories: deterministic analytic models, stochastic analytic models, economic models and simulation models. Mohamed (1999) incorporates the production-distribution planning and logistics decisions for multi-national companies (MNC's) operating under varying inflation and scanty exchange rates. Lee and Kim (2002) propose an integrated multi-period, multi-product, multi-shop production and

distribution model in supply chain to satisfy the retailer's demand in the given periods of time. The model is formulated as an analytic model which minimizes the overall costs of production, distribution, inventory holding and shortage costs, subject to various kinds of inventory and operation time constraints.

Jang *et al.* (2002) propose supply network with a global bill of material (BOM). They apply four modules for this supply network: supply network design optimization module, planning module for production and distribution operations from raw material suppliers to customer, model management module and data management module. The first two modules are solved by a Lagrangian heuristic and a generic algorithm, respectively. Yan *et al.* (2003) add logical constraints to the production-distribution problem. Their main contribution is adding BOM limitations as logical constraints to the mixed integer representation of the problem. The results of a small-scale problem are presented to show solution validity.

Chan *et al.* (2005) develop a hybrid genetic algorithm for production and distribution problems in multi-factory supply chain models and solve a hypothetical production-distribution problem by the proposed algorithm. Barnes-Schuster *et al.* (2006) study a system composed of a supplier and buyer(s). They assume that the buyer faces random demand with a known distribution function. The supplier faces a known production lead time. The main objective is to determine the optimal delivery lead time and the resulting location of the system inventory. Rizk *et al.* (2006) examine a multi-item dynamic production-distribution planning problem between a manufacturing location and a distribution center. Transportation costs between the manufacturing location and the distribution center offer economies of scale and can be represented by general piecewise linear functions. They proposed a tight mixed-integer programming model of the production process, as well as three different formulations to represent general piecewise linear functions.

Keskin and Uster (2007) consider a multi-product two-stage production-distribution system design problem (PDSD) where a fixed number of capacitated distribution centers are to be located with respect to capacitated suppliers (plants) and retail locations (customers) while minimizing the total costs in the system. They present a mixed-integer problem formulation that facilitates the development of efficient heuristic procedures. They provide meta-heuristic procedures, including a population-based scattered search with path relinking and trajectory-based local and tabu search, for the solution of the problem. They also develop efficient construction

heuristics and transshipment heuristics that are incorporated into the heuristic procedures for the solution of subproblems.

In all above studies, the considered Production-Distribution Planning Problem (PDPP) has been solved using one of the heuristic, meta heuristic or operation research methods. But in this study, we consider PDPP as both Traditional Decision Support System (TDSS) and agent-based decision support system and present the algorithm for each.

The objective of this study is to develop an agent-based framework for building decision support system to cope with the difficulties in modeling Production-Distribution Planning Problems (PDPP) in supply chain and to provide better solution plans using teams of autonomous agent (A Teams).

### **AN AGENT-BASED APPROACH TO INFORMATION MANAGEMENT**

Fox *et al.* (2000) propose some agent-oriented methods for handling information in dynamic supply chains. In general, there is no universal agreement on what an agent is, but common aspects to most definitions seem to be that an agent should be autonomous, social, reactive and pro-active (Wooldridge and Jennings, 1995; Jennings and Wooldridge, 2002).

Autonomy signifies that agents operate without direct intervention of humans or others. Social ability means that agents interact with other agents via a communication language. In order to be reactive, agents perceive their environment and respond in a timely fashion to the changes that occur in it. Finally, agents do not simply act in response to their environment; they are also able to exhibit goal-directed behavior by taking the initiative (pro-activity).

In the following parts of this section, we will first address the definition of software agents and then present the basics of Asynchronous teams (A Teams), Decision Support Systems (DSSs) incorporating agents and the way A Teams can be used to solve large combinatorial optimization problems.

**Software agents:** Software agents can be defined in different ways depending on the way they are implemented and the tasks they perform. Wooldridge and Jennings (1995) suggest any computer system (software or hardware) should have the following properties to be termed as an agent:

- **Autonomy:** It should have some control over its actions and should work without human intervention.

- **Social ability:** It should be able to communicate with other agents and/or with human operators.
- **Reactivity:** It should be able to react to changes in its environment.
- **Pro-activeness:** It should also be able to take initiative based on pre-specified goals.

The above-mentioned properties are generic for an agent. An agent may exhibit more of one property than another based on its architecture and embedded intelligence.

**Asynchronous teams (A Teams):** Talukdar (1993) and Talukdar *et al.* (1996) proposed the teams of autonomous agent (A Teams) to solve large combinatorial optimization problems using a multi-agent-based distributed problem solving method, where, the agent asynchronously build shared solution. This method allows the system either to be centrally controlled or decentralized. In an ATeam, a collection of agents co-operates by sharing solutions through a common memory. The architecture is asynchronous and the agents are autonomous, each agent decides when and how to act. Some agents may operate solely to keep the population in check, destroying selected inferior solutions.

Talukdar (1993) proposes a basic architecture to autonomous agent operating asynchronously on a shared population of solution attempts, which they call A Teams. In the basic architecture, each agent is completely independent from the rest and operates by selecting a solution from the memory, carrying out some operations on that it and then placing it back in the memory. Thus Co-operation is achieved by sharing solutions. The population of solution is controlled by a subset of destroyer agents, which evaluate solutions according to certain criteria and remove unwanted solutions. The organization of the agents is such that loose-agents may appear and disappear from the team without penalty, or may be widely distributed and do not communicate directly with other agents. An instance of ATeam architecture is shown in Fig. 1. Here, the system consists of a team of agents and a single memory, which has particular communications with each agent (Aydin and Fogarty, 2004).

A Teams are, in many respects, similar to blackboard system, in that a collection of processes co-operates to solve problems by posting the results of actions to a shared memory (or blackboard). However, there are some differences. In the blackboard systems, the problem solving process is typically centrally controlled, with a control process deciding which of the available

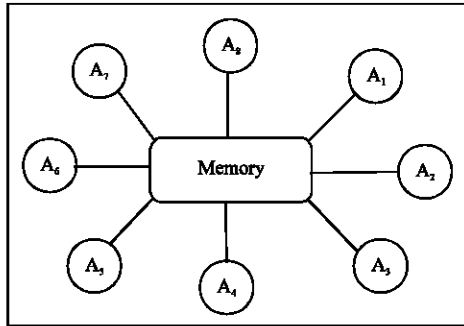


Fig. 1: An instance of A Teams architecture (Aydin and Fogarty, 2004)

knowledge source should be activated at which point. Blackboards are typically structured to suit a particular problem, being hierarchically sub-divided, with problems also being sub-divided and sub-problems combined in pre-determined ways. In a basic ATeam, there is no control and each agent operates without knowledge of the others. The memory is typically on a single level (Aydin and Fogarty, 2004).

SCM problems are both distributive in nature and require extensive intelligent decision-making. Thus, in the last few years, multi-agent systems have been a preferred tool for solving supply chain problems.

**DSSs incorporating agents:** A recent analysis of research in IS revealed that DSSs had been one of the most popular research topics (Claver *et al.*, 2000). However, traditional DSSs offer a passive form of support, where users needed to have full knowledge on how to use the relevant models, data sources and other tools and take initiative to perform all necessary operations effectively (Vahidov and Fazlollahi, 2004).

Recently researchers have argued in favor of making DSS a more active participant in the decision making process (Raghavan, 1991). The vision has favored a higher degree of system involvement and collaboration with the human in the decision process. More recently, some researchers emphasized the importance of a holistic approach (Mirchandani and Pakath, 1999).

It has been recognized that the potential contributions of IAs to DSSs is enormous. This has been re-emphasized in the recent special issue of the DSS journal on the next decade of DSS (Carlsson and Turban, 2002; Shim *et al.*, 2002). Autonomy, reactivity, social ability and proactiveness of agents can facilitate active decision support. Mentalistic characterization and reasoning capability can promote high-level cognitive processes, including qualitative reasoning, handling soft information and alternative generation. Furthermore, the

artificial Persona view of agents can contribute towards stronger collaborative relationships between a human and a DSS (Vahidov and Fazlollahi, 2004).

**PROBLEM DEFINITION**

The proposed system can generate production and distribution plans of the supply network over a planning horizon. As stated by Jang *et al.* (2002), an integrated model consists of three types of multi-stage, multi-product, production and distribution planning sub-model, as shown in Fig. 2. The three sub-models are called the P-P-P [plant-plant-plant] model, P-P-W [plant-plant-warehouse] model and W-D-C [warehouse-distribution center-customer] model, respectively. The solution designates the production and/or distribution quantities of each item for each period at each site subject to multi-level global Bill of Material (BOM) and capacity constraints (Jang *et al.*, 2002).

The integrated production-distribution problem is difficult to be solved due to the great number of integer variables representing facilities, items and periods and the binary variables indicating major setups (Jang *et al.*, 2002). Thus, a multi-agent system is adopted as a solution methodology considering the problem complexity and time requirements. In this section, only the P-P-P model is presented since it incorporates all the characteristics of other two models. Details of the Solution methodology and experimental result are presented in the next sections.

**P-P-P model:** This model covers the problem area ranging from the suppliers to final assembly plants. The model includes a two-level BOM to generate a production plan at each plant and a distribution plan among the plants. The transportation lead-time was also considered in this model (Jang *et al.*, 2002). The indices, parameters and decision variables retain their meanings in this section.

**Mathematical formulation:** The indices, parameters, decision variables, objective function and constraints are as follows:

**Indices**

- V = {1, 2, ..., v} Plant at first stage of Bill of Material (BOM)
- S = {1, 2, ..., s} Plant at second stage of BOM
- P = {1, 2, ..., p} Plant at final stage that assembles end item of BOM
- R = {1, 2, ..., r} Child item at level 2 of BOM
- C = {1, 2, ..., c} Child item at level 1 of BOM
- I = {1, 2, ..., i} Child item at level 0 of BOM
- T = {1, 2, ..., t} Period

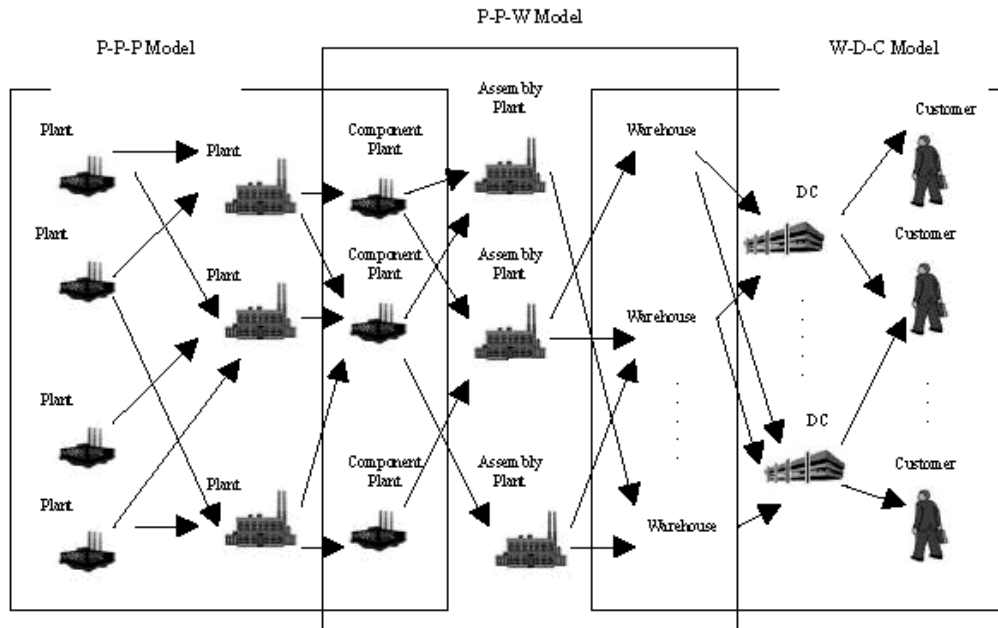


Fig. 2: Network fragmentation for integrated production and distribution planning (Jang *et al.*, 2002)

**Parameters**

- $s_{ipt}$  : Fixed producing cost for  $i$  in  $p$  in  $t$
- $h_{ipt}$  : Unit holding cost of  $i$  in  $p$  in  $t$
- $a_{ipt}$  : Unit variable cost of producing  $i$  in  $p$  in  $t$
- $s_{rvt}$  : Fixed producing cost for  $r$  in  $v$  in  $t$
- $ho_{rvt}$  : Unit holding cost of  $r$  in  $v$  in  $t$
- $a_{rvt}$  : Unit variable cost of producing  $r$  in  $v$  in  $t$
- $s_{cst}$  : Fixed producing cost for  $c$  in  $s$  in  $t$
- $ho_{cst}$  : Unit holding cost of  $r$  in  $v$  in  $t$
- $a_{cst}$  : Unit variable cost of producing  $r$  in  $v$  in  $t$
- $fo_{rvst}$  : Fixed cost of transporting  $r$  from  $v$  to  $s$  in  $t$
- $co_{rvst}$  : Unit variable cost of transporting  $r$  from  $v$  to  $s$  in  $t$
- $f_{cst}$  : Fixed cost of transporting  $c$  from  $s$  to  $p$  in  $t$
- $c_{cst}$  : Unit variable cost of transporting  $c$  from  $s$  to  $p$  in  $t$
- $d_{ipt}$  : Demand of  $p$  for  $i$  in  $t$
- $p_{ip}$  : Processing time of  $i$  in  $p$
- $p_{rv}$  : Processing time of  $r$  in  $v$
- $pc_s$  : Processing time of  $c$  in  $s$
- $A_{pt}$  : Total available production capacity of  $p$  in  $t$
- $A_{vt}$  : Total available production capacity of  $v$  in  $t$
- $A_{st}$  : Total available production capacity of  $s$  in  $t$
- $Q_{ci}$  :  $c$ 's quantity required for  $i$  (quantity per); this is awkward
- $Q_{rc}$  :  $r$ 's quantity required for  $c$  (quantity per); this is awkward
- $L_{rv}$  : Lead time from  $r$  to  $v$  for  $s$
- $L_{csp}$  : Lead time from  $c$  to  $s$  for  $p$
- $ho_{rvt}$  : Unit holding cost of  $r$  in  $v$  in  $t$
- $h_{cpt}$  : Unit holding cost of  $c$  in  $p$  in  $t$

**Decision variables**

- $X_{ipt}$  : Production amount of  $i$  in  $p$  in  $t$
- $I_{ipt}$  : Inventory amount of  $i$  in  $p$  in  $t$
- $I_{cpt}$  : Inventory amount of  $c$  in  $p$  in  $t$
- $Z_{ipt}$  : Setup variable for  $i$  in  $p$  in  $t$
- $YO_{sp}^i$  : Amount of  $c$  transported from  $s$  to  $p$  in  $t$  intended for  $p$
- $W_{sp}$  : Link variable from  $s$  to  $p$  in  $t$  for  $t$  with respect to  $YO_{sp}^i$
- $XO_{sv}^c$  : Production amount of  $c$  in  $s$  in  $t$  for  $p$
- $IO_{cst}$  : Ending inventory of  $c$  in  $s$  in  $t$
- $IO_{rvt}$  : Ending inventory of  $r$  in  $v$  in  $t$
- $ZO_{cst}$  : Setup variable for  $c$  in  $s$  in  $t$
- $YO_{vs}^r$  : Amount of  $r$  transported from  $v$  to  $s$  in  $t$  intended for  $p$
- $W_{vs}$  : Link variable from  $v$  to  $s$  in  $t$  for  $p$  with respect to  $YO_{vs}^r$
- $XO_{vs}^r$  : Production amount of  $r$  in  $v$  in  $t$  for  $s$
- $IO_{rvt}$  : Ending inventory of  $r$  in  $v$  in  $t$
- $ZO_{rvt}$  : Setup variable for  $r$  in  $v$  in  $t$

**Objective function and constraints**

$$\begin{aligned}
 & \sum_{i=1}^R \sum_{p=1}^V \sum_{t=1}^T \left( S_{ip} ZO_{ip} + ho_{ip} + a_{ip} \sum_{i=1}^R ZO_{ip}^i \right) \\
 \text{Min} & \\
 & + \sum_{c=1}^C \sum_{s=1}^S \sum_{t=1}^T \left( S_{cs} ZO_{cs} + ho_{cs} + a_{cs} \sum_{i=1}^R XO_{sv}^c \right) + \sum_{i=1}^R \sum_{p=1}^V \sum_{t=1}^T ho_{ip} IO_{ip}
 \end{aligned} \tag{1}$$

$$\sum_{i=1}^I \sum_{p=1}^P \sum_{t=1}^T (s_{ipt} Z_{ipt} + h_{ipt} + a_{ipt} X_{ipt}) + \sum_{c=1}^C \sum_{p=1}^P \sum_{t=1}^T h_{cpt} I_{cpt}$$

$$+ \sum_{r=1}^R \sum_{v=1}^V \sum_{t=1}^T \left( f_{rvt} W_{rvt} + c_{ovt} \sum_{p=1}^P Y_{ovt}^p \right) + \sum_{c=1}^C \sum_{r=1}^R \sum_{p=1}^P \sum_{t=1}^T (f_{cpt} W_{cpt} + c_{cpt} Y_{cpt}^p)$$

s. t.

$$\sum_{s=1}^S \sum_{r=1}^R p_{rvs} X_{ovs}^s \leq A_{vt} \quad \forall v, t \quad (2)$$

$$\sum_{p=1}^P \sum_{c=1}^C p_{cst} X_{ovs}^p \leq A_{st} \quad \forall s, t \quad (3)$$

$$\sum_{i=1}^I p_{ipt} X_{ipt} \leq A_{pt} \quad \forall p, t \quad (4)$$

$$\sum_{s=1}^S X_{ovs}^s \leq M_{(ZO_{vt})} \quad \forall r, v, t \quad (5)$$

$$\sum_{p=1}^P X_{ovs}^p \leq M_{(ZO_{st})} \quad \forall c, s, t \quad (6)$$

$$X_{ipt} \leq M_{(Z_{pt})} \quad \forall i, p, t \quad (7)$$

$$\sum_{p=1}^P Y_{ovt}^p \leq M_{(W_{va})} \quad \forall r, v, s, t \quad (8)$$

$$Y_{ovt}^p \leq M_{(W_{opt})} \quad \forall c, s, p, t \quad (9)$$

$$\sum_{s=1}^S X_{ovs}^s + IO_{rv,t-1} - \sum_{p=1}^P \sum_{t=1}^T Y_{ovt}^p - IO_{rvt} = 0 \quad \forall r, v, t \quad (10)$$

$$\sum_{p=1}^P X_{ovs}^p + IO_{cs,t-1} - \sum_{p=1}^P Y_{ovt}^p - IO_{cst} = 0 \quad \forall c, s, t \quad (11)$$

$$X_{ipt} + I_{ip,t-1} - d_{ipt} - I_{ipt} = 0 \quad \forall i, p, t \quad (12)$$

$$\sum_{s=1}^S Y_{ovt}^p - L_{cp} + I_{cpt-1} - \sum_{i=1}^I (X_{ipt} Q_{ia}) - I_{cpt} = 0 \quad \forall c, p, t \quad (13)$$

$$\sum_{v=1}^V \sum_{p=1}^P Y_{ovt}^p - L_{na} + IO_{vt-1} - \sum_{c=1}^C \sum_{p=1}^P (X_{ovs}^p Q_{ca}) - IO_{vt} = 0 \quad \forall r, s, t \quad (14)$$

$$\sum_{v=1}^V X_{ovs}^s - L_{mv} \geq \sum_{p=1}^P \sum_{c=1}^C (X_{ovs}^p Q_{ca}) - IO_{rs,t-1} \quad \forall r, s, t \quad (15)$$

$$\sum_{s=1}^S X_{ovs}^s - L_{cp} \geq \sum_{i=1}^I (X_{ipt} Q_{ia}) - IO_{cp,t-1} \quad \forall c, p, t \quad (16)$$

$$X_{ipt} \geq d_{ipt} - I_{ipt-1} \quad \forall i, p, t \quad (17)$$

$$X_{ipt}, I_{ipt}, Z_{ipt}, I_{cpt}, IO_{cst}, IO_{rst}, IO_{rvt} \geq 0, \quad \forall i, p, t, c, s, r, v \quad (18)$$

$$Y_{ovt}^p, X_{ovs}^p, Y_{ovt}^p, X_{ovs}^p \geq 0 \quad \forall c, s, p, t, r, v \quad (19)$$

$$Y_{ovt}^p, X_{ovs}^p, Y_{ovt}^p, X_{ovs}^p \geq 0 \quad \forall c, s, p, t, r, v \quad (20)$$

Here, the objective function is to minimize total production cost, total ending inventory cost and total cost of transporting. The constraints (2)-(4) are the capacity restrictions at each stage. Constraints (5)-(7) ensure that a setup occurs when a plant manufacturing an item. Constraints (8) and (9), where is an arbitrary large number, imply that a link between plants is connected if the transportation quantity is non-zero. Constraints (10)-(12) represent inventory balance equations that define the inventory levels for items and at the end of period at each plant resulting from the production and transportation. Constraints (13) and (14) balance inventory available against successor's production quantities subject to BOM relationships that define the single-level gozinto structure between successors and their predecessors and Constraints (15)-(17) ensure that the external demands can be satisfied through stages.

### SOLUTION METHODOLOGY FOR DSS

Here, we provide a solution methodology for DSS of the P-P-P problem. To this aim, two scenarios are addressed. In the first scenario, a Traditional Decision Support System (TDSS) is considered for the P-P-P problem. In this scenario, a Genetic Algorithm (GA) is used for solving it. In the second scenario, an agent-based framework for building decision support system is considered in the P-P-P problem. Three algorithms are used in the scenario: Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA). Then an algorithm is suggested by using the agent-based system and A Teams concept.

**First scenario: Traditional decision support system (TSS):** A traditional DSS is designed for the P-P-P problem as shown in Fig. 3. This DSS involves the following three modules:

**Model management module:** The primary objective of the model management module is to relieve the burden of specifying models from the decision maker. It provides an environment for strong, retrieving and manipulating models. This module serves as a bridge, linking the environment of a decision maker to appropriate models. This module develops the production and distribution model in mathematical format. The data management module supplies the data required to write these models and is explained below.

**Data management module:** The data management module, which performs data collection, reduction and validation activities, plays a very important role in the modeling process. First, it separates the database and application programs to make them logically independent. The

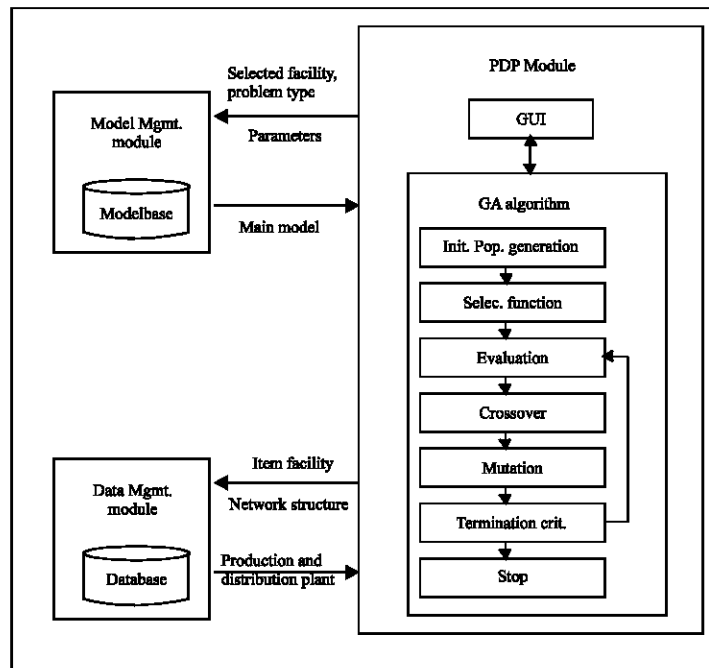


Fig. 3: A traditional decision support system (TSS) for P-P-P problem

architecture makes it possible to hide the complex database structure from the application point of view and still maintain the database integrity. Second, it manages the versions of the plans so that alternative plans can be compared and the best one selected. This function enables the decision maker to manage plans historically and perform a what-if analysis.

**Production and distribution planning module:** This module generates real time production and distribution plans for distributed facilities over the supply network. These include Master Production Schedule (MPS) level plans for each plant in the Material Requirement Planning (MRP) environment and these plans minimize total costs while satisfying the global BOM constraints and capacity constraints. The integrated production-distribution problem (P-P-P problem) is difficult to solve due to the number of integer variables representing the facilities, items and the number of periods and binary variables indicating major setups. Considering problem complexity and time requirement, a Genetic Algorithm (GA) is used as a solution methodology for integrated production and distribution planning problems. As said before, the P-P-P problem can be solved by using GA.

Genetic Algorithms (GAs) have been used successfully to find optimal or near-optimal solutions for a wide variety of optimization problems (Gen and Cheng, 1997; Goldberg, 1989) since its introduction by Holland

(1992). GAs are intelligent stochastic optimization techniques based on the mechanism of natural selection and genetics. GAs start with an initial set of solutions, called population. Each solution in the population is called a chromosome (or individual), which represents a point in the search space. The chromosomes are evolved through successive iterations, called generations, by genetic operators (selection, crossover and mutation) that mimic the principles of natural evolution. In a GA, a fitness value is assigned to each individual according to a problem-specific objective function. Generation by generation, the new individuals, called offspring, are created and survive with chromosomes in the current population, called parents, to form a new population.

The following GA modules are used for solving the P-P-P problem:

- **Initial population generation:** We use the random method for generating the initial population and the uniform distribution for reaching this goal.
- **Fitness function:** We considered the objective function of P-P-P problem as the fitness function.
- **Selection function:** The selection function chooses parents for the next generation. We apply roulette wheel [This is a way of choosing members from the population of chromosomes in a way that is proportional to their fitness] selection procedure for the selection function in the problem.



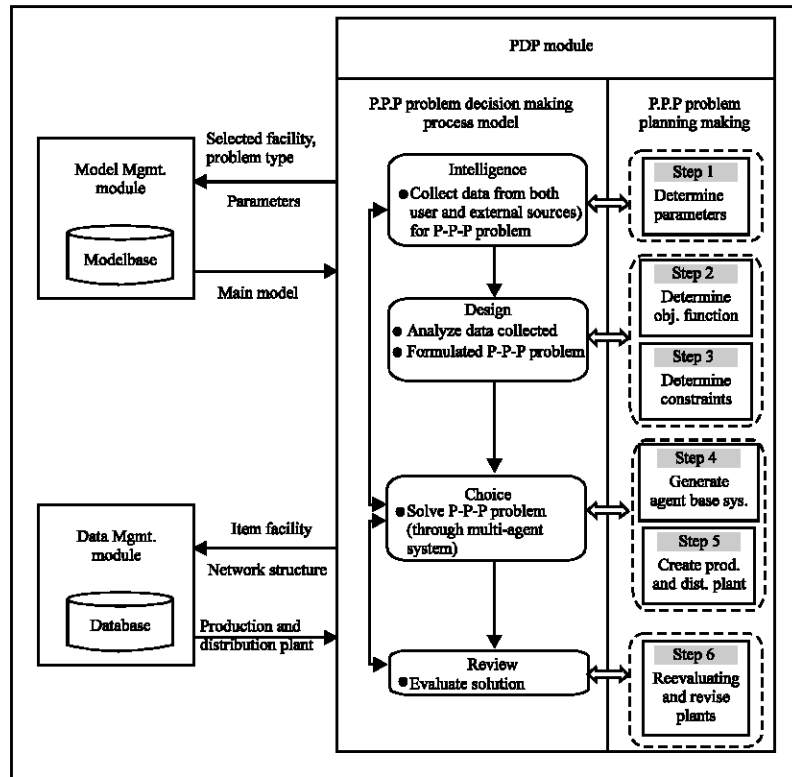


Fig. 4: An agent-based framework for building decision support system in P-P-P problem

- **Crossover:** We apply two-point crossover in the P-P-P problem.
- **Mutation:** We apply a uniform selection of new values in the developed algorithm in the mutation procedure.
- **Termination criterion:** In the P-P-P problem, termination criterion is considered to be 1000 generations and if there is no improvement in the best fitness value for the 20 generations, the algorithm stops.

**Second scenario: Agent-based framework for building decision support system:** An agent-based framework DSS is presented for the P-P-P problem as shown in Fig. 4. This DSS is designed based on the decision process proposed by Simon (1977).

Herbert A. Simon proposed a decision process comprising four distinct phases: intelligence, design, choice and review. In the intelligence phase, the decision maker recognizes the problem at hand and gathers information about the situation. The design phase is marked with structuring the problematic situation, developing criteria and identifying the various alternatives through which the problem can be solved. In the choice phase, the decision maker chooses the best

alternative that meets the criteria and makes the final decision. Following these three phases, the decision maker uses the feedback from the results of the decision to review how well the process was executed. Such reflection on past processes can form a basis of the intelligence phase for future decisions. Although generic and simple in nature, Simon’s decision-making process model has been applied and validated in a wide array of situations (Gao *et al.*, 2007).

Referring to Simon’s model and the P-P-P problem, we formulated the decision-making process model for P-P-P problem, which includes gathering information about the P-P-P problem situation (intelligence), identifying various alternatives (i.e., formulating models) through which the problem can be solved (design), choosing the best alternative that meets the criteria (choice) and evaluating and revising the alternative (review).

Figure 4 shows that the main engine of an agent-based system is Step 4 that is Generate agent-based system which is described in more details later. This engine involves an agent-based system which solves the P-P-P problem considering A Teams concept using three algorithms: Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA). The modules of each algorithm are as follows:

**Genetic algorithm:** The modules of this algorithm is as those of GA which was introduced in the case of TDSS. So in this section, the problem is solved using the GA which was introduced in the case of TDSS.

**Tabu search:** Tabu search (TS) is a metaheuristic, which is initially introduced by Glover (1987). Independently, Hansen (1986) proposed a steepest ascent/deepest descent technique, based on similar ideas. In TS, the process of searching for a good or optimal solution is based on applying a subordinate heuristic in each step, which has to be designed for each particular problem type. Beginning with an initial feasible solution in each iteration, it tries to minimize the objective function during the search procedure by applying a local transmission to the current solution. It has a particular mechanism in order to prevent the search procedure from trapping in local minimums. One of the most important key elements of TS is the use of a short-term memory function, called Tabu List, which provides strategic forgetting. It may include the information of solutions obtained or moves associated to the last iterations. Any element of iteration, existing in this list is forbidden, except if it satisfies some aspiration criteria.

Here, the P-P-P problem is solved using TS. The following TS modules are used for solving the P-P-P problem:

- **Initial solution:** A good and yet not optimal solution is to consider all variables as small as possible. So we equal all variables to 0. However this may overrule constraints, so we increase them incrementally until they meet constraints.
- **Neighborhood structure and search strategy:** A neighborhood to a given solution is obtained adding an increment () to one and only one of the variables.
- **Tabu list:** In order to guide the search process we need to use memory to store all recently visited solutions. This prevents search process from cycling in a small region of solution space. This is implemented by establishing a Tabu List which is updated at some stages to store visited solution for a limited time.
- **Aspiration criterion:** Sometimes Tabu Structure and its rules may be such that prevent some attractive moves and forbid some good solution, although there may be no danger of cycling. To solve this problem, sometimes it is better to consider tabu solutions. This may be when Tabu solutions result in objective function values which are better than best-known solutions.
- **Diversification:** Sometimes exploration process may be trapped in a local minimum and no improvement occurs. In such a condition we need a mechanism to

drive search process into new region or explore areas which have been explored less than other regions. This can be implemented by increasing the increment ( $\Delta$ ) for a limited time. This is only implemented when no improvement is reached for a pre-specified number of iterations. Once any improvement is detected, diversification is disabled and normal increment is used.

- **Termination criteria:** The algorithm terminates if a pre-specified number of iterations is reached.

**Simulated annealing:** Simulated Annealing (SA) is a stochastic neighborhood search method that is developed for the combinatorial optimization problems. It was initially proposed by Kirkpatrick *et al.* (1983) based on the analogy between the process of annealing of solids and the solution methodology of combinatorial optimization problems. The SA methodology draws its analogy from the annealing process of solids. In the annealing process, a solid is heated to a high temperature and gradually cooled to allow it to crystallize. As the heating process allows the atoms to move randomly, if the cooling is done too rapidly, it gives the atoms enough time to align themselves in order to reach a minimum energy state. This analogy can be used in combinatorial optimization with the state of solid corresponding to the feasible solution, the energy at each state corresponding to the improvement in the objective function and the minimum energy state being the optimal solution.

Here, the P-P-P problem is solved using SA. The following SA modules are used for solving the P-P-P problem:

- **Initial solution:** The algorithm for finding initial solution is same as that of TS.
- **Neighborhood structure and search strategy:** The neighborhood at each step is any solution obtained by a small perturbation in system variables. This may be obtained by adding a small increment to only one of the variable of the system. The neighborhood solution is selected if it improves the energy of the system (decreases the system energy), otherwise it is selected with a suitable probability. This probability is proportional to its closeness to current minimum value.
- **Termination criteria:** The algorithm terminates if it meets any of the following criteria:
  - A pre-defined iterations is reached.
  - There is no improvement in the solution for last pre-specified number of iterations.
  - Fraction of neighbor solutions tried that is accepted at any temperature reaches a pre-specified minimum.

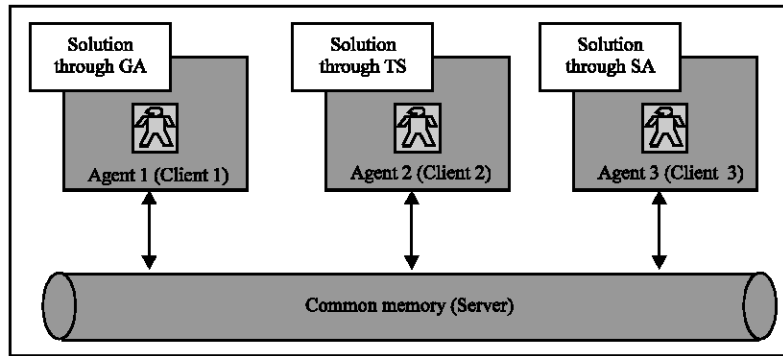


Fig. 5: System architecture

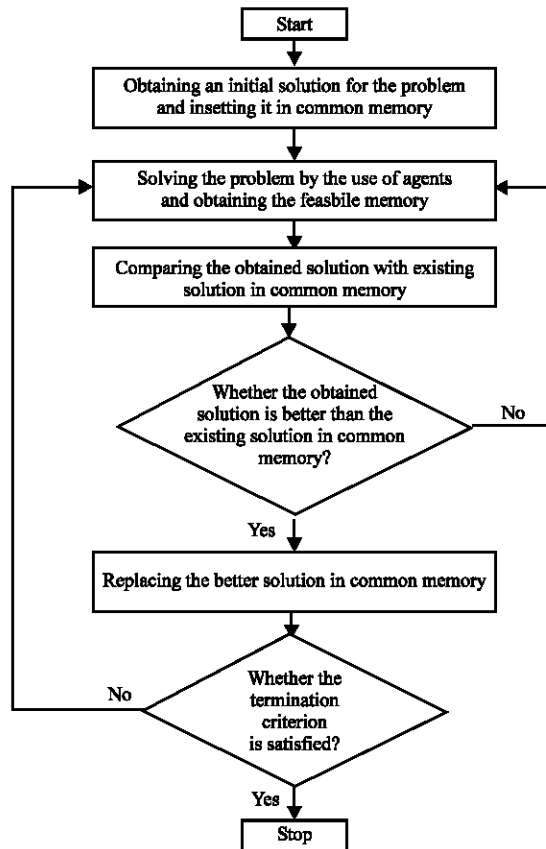


Fig. 6: Proposed algorithm of each agent

**An agent-based system:** In this section an agent-based system is used for solving the P-P-P problem considering A Teams concept. The suggested architecture is presented in Fig. 5.

In this architecture, each algorithm which was introduced in the previous section is considered as an agent. Each agent acts as a client and after solving the problem, it delivers the obtained solution to common memory which plays the role of a server.

In this architecture, each agent solves the problem as a parallel problem according to its mechanism and inserts the obtained result into common memory after solving the problem. The suggested algorithm in this architecture is presented in Fig. 6.

The procedure of implementation of this algorithm is as follows: By considering the modules of one of the three agents, an initial solution is obtained randomly. This initial solution is inserted into common memory. Hereafter, each agent solves the problem independently and compares its obtained solution with existing solution in common memory. If the obtained solution through each agent is better than the existing solution in common memory, the existing solution is replaced with it. Otherwise, the problem is resolved through that agent. This process continues until the termination criterion is satisfied.

Two important characteristics of agents namely cooperation and parallel processing are used in this algorithm which will increase synergy and thereby coordination in the system.

Figure 7 shows the diagram of interaction between agents and common memory and the way messages are exchanged in each one. In Fig. 7, the interactions are based on the procedure proposed. The arrowed lines show interactions between agents and common memory.

**EXPERIMENTAL RESULTS**

Here, the results of a series of numerical experiments that were carried out for a production-distribution system. To demonstrate the effectiveness of the suggested methods, both TDSS and MADSS methods were run using MATLAB® 7 (R14).

To examine the results, the data of an auto company in Asia has been used. According to this company’s data, 10 problems have been generated randomly yet systematically to capture a wide range of problem

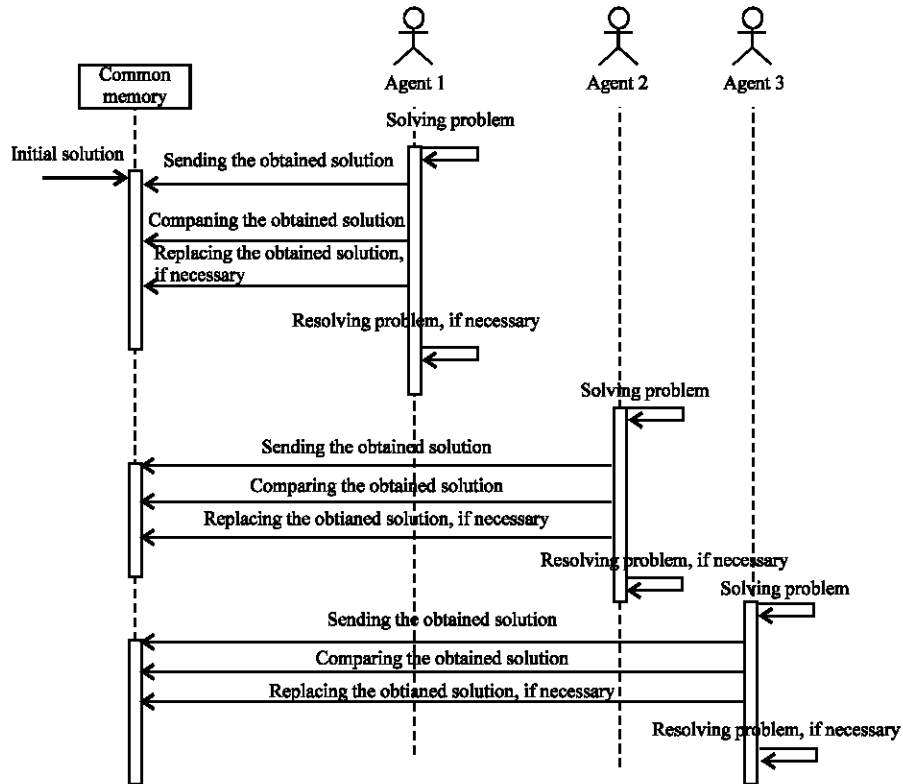


Fig. 7: Interaction diagram for the second scenario

Table 1: Comparison results of Agent-based system DSS with TDSS

No. problem	V	S	P	R	C	I	TDSS		Agent-based system DSS	
							Time (sec)	Obj.	Time (sec)	Obj.
1	1	2	2	3	2	1	53	1253	93	1372
2	2	1	2	2	3	1	124	45372	298	45551
3	2	2	1	3	3	2	152	51251	479	51472
4	3	2	2	2	3	2	355	1982	625	2003
5	3	3	3	2	3	2	697	23796	882	23851
6	3	4	4	3	4	2	831	7329	917	7418
7	4	3	5	3	4	3	1052	828103	988	828324
8	5	4	4	4	5	3	1548	22534	993	22794
9	6	5	6	5	4	3	1923	931024	1024	931975
10	5	6	5	4	6	3	2281	77538	1135	77582

structure. Each problem has been solved using both TDSS and agent-based system DSS methods and is shown in Table 1. In Table 1, the objective function values of P-P-P problem and CPU time of each problem with different indices and parameters which solved by using TDSS method and agent-based system DSS methods are provided. The obtained data in this table shows that, for large scale, the CPU time for agent-based system DSS method is less than that of the TDSS method.

In large scale, the results show that the proposed agent-based system DSS method provides better solutions than that of TDSS method. Since parallel processing is used by multi-agent system in agent-based system DSS method better results are obtained compared to other solutions.

## CONCLUSIONS

In this study, a Decision Support System (DSS) was developed in Production-Distribution Planning Problem (PDPP) for a supply chain management. For this purpose, we presented two scenarios. In the first scenario, a Traditional Decision Support System (TDSS) was considered for the PDPP and a genetic algorithm (GA) has been used for solving it. In the second scenario, an agent-based framework decision support system is considered for PDPP considering Simon process and three algorithms were used for solving it: Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA). Then an algorithm was suggested by using the agent-based system and teams of autonomous agents (A Teams)

concept. The results show that the use of agent-based system DSS in large scale delivers better solutions to us.

#### REFERENCES

- Aydin, M.E. and T.C. Fogarty, 2004. Teams of autonomous agents for job-shop scheduling problems: An experimental study. *J. Intel. Manuf.*, 15: 455-462.
- Barnes-Schuster, D., Y. Bassok and R. Anupindi, 2006. Optimizing delivery lead time-inventory placement in a two-stage production-distribution system *Eur. J. Operat. Res.*, 174: 1664-1684.
- Beamon, B.M., 1998. Supply chain design and analysis: models and methods. *Int. J. Prod. Eco.*, 55: 281-294.
- Carlsson, C. and E. Turban, 2002. DSS: Directions for the next decade. *Dec. Sup. Syst.*, 33 (2): 105-110.
- Chan, C.C.H., C.B. Cheng and S.W. Huang, 2006. Formulating ordering policies in a supply chain by genetic algorithm. *Int. J. Model Sim.*, 26 (2): 129-136.
- Chan, F.T.S. and S.H. Chung, 2005. Multicriterion genetic optimization for due date assigned distribution network problems. *Dec. Sup. Syst.*, 39: 661-675.
- Chan, F.T.S., S.H. Chung and S. Wadhwa, 2005. A hybrid genetic algorithm for production and distribution. *Omega*, 33: 345-355.
- Chon, M.A. and H.L. Lee, 1988. Strategic analysis of integrated production-distribution systems: Models and methods. *Operat. Res.*, 36 (2): 216-228.
- Claver, E., R. Gonzales and J. Llopis, 2000. An analysis of research in information systems (1981-1997). *Inform. Manage.*, 37: 181-195.
- Dasci, A. and V. Verter, 2001. A continuous model for production-distribution system design. *Eur. J. Operat. Res.*, 129: 287-298.
- Fazlollahi, B. and R. Vahidov, 2001. A method for generation of alternatives by decision support systems. *J. Manage. Inform. Syst.*, 18 (2): 229-250.
- Fischer, G. and T. Mastaglio, 1991. A conceptual framework for knowledge-based critic systems. *Dec. Sup. Syst.*, 7 (4): 355-378.
- Fox, M.S., M. Barbuceanu and A.R. Teigen, 2000. Agent-oriented supply chain management. *Int. J. Flex. Manuf. Syst.*, 12: 165-188.
- Gao, S., H. Wang, D. Xu and Y. Wang, 2007. An intelligent agent-assisted decision support system for family financial planning. *Dec. Sup. Syst.*, 44 (1): 60-78.
- Gen, M. and R. Cheng, 1997. *Genetic Algorithms and Engineering Design*. Wiley, New York.
- Glover F., 1987. Tabu search methods in artificial intelligence and operations research. *ORSA Artificial Intelligence Newsletter*, 1 (2): 4-6.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, MA.
- Hansen, P., 1986. The steepest ascent mildest descent heuristic for combinatorial Programming. Presented at the Congress on Numerical Methods in Combinational Programming, Capri, Italy.
- Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems*. 2nd Edn. University of Michigan Press, MIT Press.
- Jang, Y.J., S.Y. Jang, B.M. Chang and J. Park, 2002. A combined model of network design and production distribution planning for a supply network. *Comput. Ind. Eng.*, 43: 263-281.
- Jennings, N.R. and M. Wooldridge, 2002. *Agent Technology: Foundations, Applications and Markets*, Springer.
- Keskin, B.B. and H. Uster, 2007. Meta-heuristic approaches with memory and evolution for a multi-product production/distribution system design problem. *Eur. J. Operat. Res.*, 182: 663-682.
- Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, 1983. Optimisation by simulated annealing. *Science*, 220 (4598): 671-680.
- Lee, Y.H. and S.H. Kim, 2000. Optimal production-distribution planning in supply chain management using a hybrid simulation-analytic approach. *Proceedings of the 2000 Winter Simulation Conference*.
- Lee, Y.H. and S.H. Kim, 2002. Production-distribution planning in supply chain considering capacity constraints. *Comput. Ind. Eng.*, 43: 169-190.
- Lerch, F.J. and D.E. Harter, 2001. Cognitive support for real-time dynamic decision making. *Inform. Syst. Res.*, 12 (1): 63-82.
- Lumsden, K., F. Dallari and R. Ruggeri, 1999. Improving the efficiency of the hub and spoke system for the SKF European distribution network. *Int. J. Phys. Dist. Logis. Manage.*, 29 (1): 60-64.
- Milgate, M., 2001. Supply chain complexity and delivery performance: An international exploratory study. *Supply Chain Manage. An Int. J.*, 6 (3): 106-118.
- Mirchandani, D. and R. Pakath, 1999. Four models for a decision support system. *Inform. Manage.*, 35: 31-42.
- Mohamed, Z.M., 1999. An integrated production-distribution model for a multi-national company operating under varying exchange rates. *Int. J. Prod. Eco.*, 58: 81-92.
- Radermacher, F.J., 1994. Decision support systems: Scope and potential. *Dec. Sup. Syst.*, 7: 315-328.
- Raghavan, S.A., 1991. JANUS: A paradigm for active decision support. *Dec. Sup. Syst.*, 7: 379-395.

- Rizk, N., A. Martel and S. D'Amours, 2006. Multi-item dynamic production-distribution planning in process industries with divergent finishing stages. *Comput. Operat. Res.*, 33: 3600-3623.
- Shim, J.P., M. Warkentin, J.F. Courtney, D.J. Power, R. Sharda and C. Carlsson, 2002. Past, present and future of decision support technology. *Dec. Sup. Syst.*, 33 (2): 111-126.
- Simon, H.A., 1977. *The New Science of Management Decision*. Englewood Cliffs. N.J., Prentice-Hall.
- Stank, T.P. and T.F. Goldsby, 2000. A framework for transportation decision making in an integrated supply chain. *Logis. Inform. Manage.*, 5 (2): 71-77.
- Talukdar, S., 1993. Asynchronous teams. *Proceedings of the 4th International Symposium on Expert Systems Applications to Power Systems*, LaTrobe University, Melbourne, Australia.
- Talukdar, S., L. Baerentzen, A. Gove and P. de Souza, 1996. Asynchronous teams: Co-operation schemes for autonomous. *Computer-Based Agents*, Technical Report EDRC 18-59-96, Engineering Design Research Center, Carnegie Mellon University.
- Thomas, D.J. and P.M. Griffin, 1996. Coordinated supply chain management. *Eur. J. Operat. Res.*, 94: 1-15.
- Vahidov, R. and B. Fazlollahi, 2004. Pluralistic multi-agent decision support system: A framework and an empirical test. *Inform. Manage.*, 41: 883-898.
- Vidal, C.J. and M. Goetschalckx, 1997. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *Eur. J. Operat. Res.*, 98: 1-18.
- Wooldridge, M. and N.R. Jennings, 1995. Intelligent agents: Theory and practice. *Knowl. Eng. Rev.*, 10 (2): 115-152.
- Yan, H., Z. Yu and T.C.E. Cheng, 2003. A strategic model for supply chain design with logical constraints: Formulation and solution. *Comput. Operat. Res.*, 30: 2135-2155.