

Using Electromagnetism Algorithm for Determining the Number of kanbans in a Multi-stage Supply Chain System

Majid Khalili^{a,*}, Mohammad Jafar Tarokh^b, Bahman Naderi^c

^a Islamic Azad University, Karaj branch, Department of Industrial Engineering, Alborz, Iran

^b Department of Industrial Engineering, K.N.Toosi University, Tehran, Iran

^c Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Received 13 Jul., 2010; Revised 22 Jul., 2010; Accepted 15 Aug., 2010

Abstract

This paper studies the multi-stage supply chain system (MSSCM) controlled by the kanban mechanism. In the kanban system, decision making is based on the number of kanbans as well as batch sizes. A kanban mechanism is employed to assist in linking different production processes in a supply chain system in order to implement the scope of just-in-time (JIT) philosophy. For a MSSCM, a mixed-integer nonlinear programming (MINLP) problem is formulated from the perspective of JIT delivery policy where a kanban may reflect to a transporter. Since the adopted model is of MINLP type and solving it by branch and bound (B&B) takes time, a metaheuristic is presented. This metaheuristic is an electromagnetic algorithm (EA). The EA is compared against an existing algorithm and also B&B results to evaluate the proposed metaheuristic. Extensive experiments and statistical analyses demonstrate that our proposed EM is more efficient than B&B with regard to the objective functions considered in this paper.

Keywords: Kanban; Multi-stage supply chain system (MSSCS); Mixed integer non-linear programming (MINLP); Electromagnetism algorithm (EA).

1. Introduction

A supply chain system (SCS) is a set of subsystems utilized to establish an effective relation between suppliers, manufactures, warehouses, distribution centers, retailers and finally customers in order to produce products in right quantities and to distribute them to right location and at right time, with the aim of minimizing the total system's costs while satisfying the service level requirements [1].

An SCS operating with excess inventory is not effective as excess inventory denotes poor planning, poor purchasing practices, poor communication and poor quality levels. In order to remain competitive and to experience economic success, every organization focuses on increasing productivity, improving the quality of its products and setting high standards of efficiency within its company. Improvements in reduction of inventory, wasted labor and customer service are usually accomplished through the kanban operations. The material flow and information flow between two adjacent plants form a kanban stage. If an SCS

consists of only two plants, it is called single-stage supply chain system (SSSCS). If it consists of more than two plants, and they are in series, it is called multi-stage supply chain system (MSSCS) but assembly type supply chain system (ATSCS) consists of more than two plants, and they form an assembly tree. This research aims at increasing the degree of efficiency within the production processes and reducing the level of inventory and wasted materials, time, and effort involved in each production stage. The objective is to build a logistics system for a general SCS, the MSSCS, controlled by the kanban technique.

For the SCS, the number of kanbans, the batch size and the number of batches in each stage that is to be shipped by kanbans and the total quantity over one period are to be determined optimally. The kanban operation at each stage is to be configured based on the optimal results of the model. Efficiently controlling the production and reducing the work in process (WIP) will be the outcome of the SCS, which leads to minimizing the total cost of SCS. In the kanban system, each plant sends signals to the preceding plant for needed parts and the kanban system

* Corresponding author email: Khalili.mj@kiaau.ac.ir

acts with customer demand on the last plant. The workstations are located along the production lines. Empty transporters show that more parts will be needed for production. Each workstation only produces enough components to fill transporters and then stops [22].

An SCS is generally viewed as a network of facilities that performs the procurement of raw material, its transformation into intermediate and end-products for the customers with the use of two major control philosophies: the “push” system and the “pull” system [3,6]. In an SC that is controlled by the kanban, the flow of components is from preceding plant to succeeding plant, but the flow of information is downward and from succeeding plant to

preceding plant as shown in Fig. 1. Just-in-time (JIT) production system is the manufacturing philosophy of production of what is needed at the right time and in the right quantity [8, 9]. The kanban coupled with pull system of production is used as a means of implementing JIT. Determining the kanban numbers and batch sizes between two plants to reach JIT production philosophy are important factors in this paper. If p_i is the production rate of plant i , the assumptions considered in this paper are: (a) Demand rate is less than bottleneck plant, (b) Demand is known and (c) in each stage $\bar{P}_i > P_{i+1}$

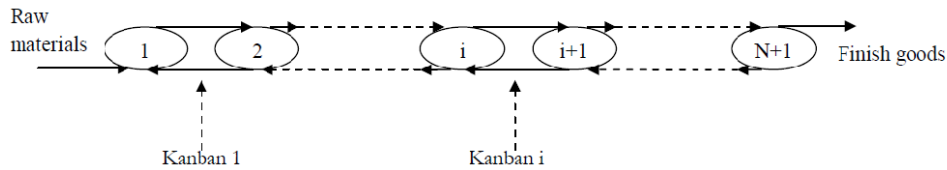


Fig. 1. A Configuration of th N-stage kanban system (N+1 workstations)

With respect to the above explanations, the aim of this study is to determine the number of kanbans in a MSSCS. In this paper, by considering the advantages of EA, an efficient algorithm called EA is proposed.

Wang and Sarker [23] solved large MINLP problems; the method was helpful to obtain a close to optimal solution when the optimal solution is hard to reach. At least, the solution can serve as an upper bound of the optimal solution.

In the supply chain system controlled by the kanbans, the total quantity of products produced in each stage over a period ties the individual stages together. Without this variable approach, the inventory model of a supply chain system with a series of stages is hard to establish [23]. MSSCM is of MINLP type and solving it by branch and bound (B&B) takes time [23].

The remainder of the paper is as follows: In Section 2, some previous studies about the kanban are reviewed. In Section 3, the problem is described. In Section 4, the problem assumptions are presented. In Section 5, the problem is modeled and a formulation is presented for the problem and in Section 6 the proposed model is analyzed. At last, in Section 7 the final conclusions are presented.

2. Literature Review

The kanban technique has attracted many researchers' attention since it was first brought to light by Monden [12] who summarized the Toyota approach to determine the appropriate number of kanbans at each workstation.

Berkly [2] reviewed many papers in the field of production control via kanban and categorized them at the base of their systems. He listed 24 design parameters for the kanban performance as well. Sarker and Balan in [18,

19, and 20] determined the number of kanbans to transport the material between two workstations for SSSCS and MSSCS. Rees et al. [17] extended the Toyota approach to fluctuate product-mix problem by using the next period's forecast demand and the last period's observed lead times. Co and Sharafali [4] considered the over-planning factor in Toyota's formula for computing the number of kanbans for several production inventory control models.

Sue et al. [18] provided further insights into the adoption of ERP systems and the impacts on firm competence in SCM and proposed a model featuring ERP benefits to firm competences in supply chain management. Sengupta et al. [21] analyzed manufacturing and service supply chain performance. Chang et al. [5] introduced a neural network evaluation model for ERP performance from SCM perspective to enhance enterprise competitive advantage.

Altioek and Ranjan [14] studied a multi-stage pull system dealing with production inventory system. Gupta and Alturki [7] introduced a systematic methodology to manipulate the number of kanbans in a JIT system, where an algorithm to minimize the backlog and WIP was developed for stochastic processing times and variable demand environment. Miyazaki et al. [11] derived two formulae to calculate the average inventory yielded by the fixed-interval withdrawal kanbans and supplier kanbans in a JIT production system, and the minimum number of kanbans required for this system was determined by two formulae. Parija and Sarker [15] analyzed a supply chain system by modeling the raw material ordering policy and finished goods delivering policy. As a result, an economic batch size for a product with a fixed time-interval was developed. Mascolo et al. [10] developed a general-purpose analytical method for performance evaluation of

the multi-stage kanban controlled production systems. In their method, a given number of kanbans was associated with each stage. Nori and Sarkar [14] determined the delivery policy and the number of kanbans between two workstations. In Nori and Sarker's models, the total cost was expressed as the function of the number of kanbans, shortage cost of materials, and holding cost of containers. Wang et al. determined the number of kanbans and transported batch size in SCS and also Rabbani et al. [16]

studied the number of kanbans in an SCS via Memetic algorithms.

3. Problem Description

The function of kanban is described through the use of an N-stage production system as shown in Fig. 2:

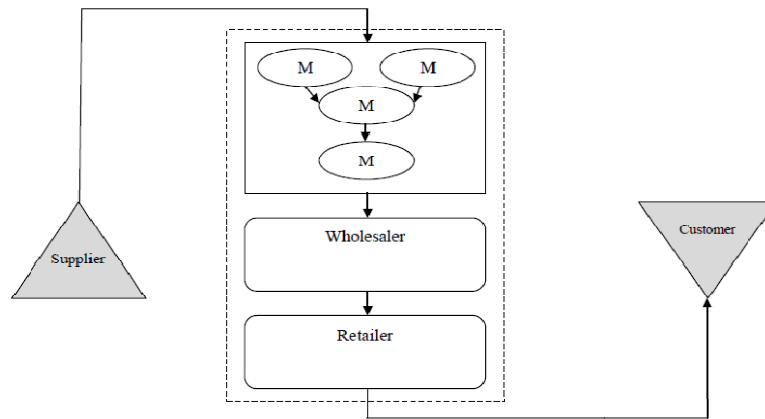


Fig. 2. A typical supply chain system.

In this problem, each plant has got two stores: one is located before the plant where the parts are received, and another store is located after the plant for storing the parts processed in the plant. Two plants j and $j + 1$ in Fig. 2 are isolated for illustration as indicated in Fig. 3. Store γ is a store after plant j and store λ is a store before plant $j + 1$ and when plant $j + 1$ takes and uses a container, the departure kanban is detached and put in the kanban post. Thus, in fixed or non-fixed intervals, the withdrawal kanbans are collected from the kanban post, and

accompanied with empty transporters, transshipped to the preceding plant. Finally, all collected withdrawal kanbans and empty transporters are put in their particular places. Each detached withdrawal kanban acts as a trigger of the preceding plant and commands operators to produce and fill empty transporters. Now transporters filled in store γ should wait and then any filled transporter transshipped with its kanban, is carried to the succeeding plant and put in store λ .

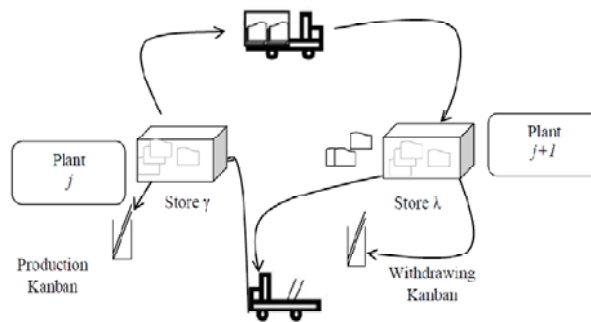


Fig. 3. Operation of kanban production system.

When customer orders, each plant is actuated by the preceding plant to reach the first plant. When the detached kanban arrives at the first plant, empty transports and detached withdrawal kanbans are transshipped to supplier to get filled again and then full transporters and their attached kanbans are transshipped to the factory together. The present study aims at calculating the minimum total cost of the system. So, a system of costs is developed to

obtain the minimum value of the total cost through finding the optimal number of kanbans and batch sizes.

4. Problem Definition

The MSSCS model gives the parameters, e.g. the delivery policy to the retailers, the ordering policy to the suppliers, the batch size and number of batches to deliver

WIPs, and the total quantity of products in one period in manufacturing stages. The cost of each plant is a function of transportation of each transporter, holding costs of inventory, setup. The notations used in model are:

4.1. Cost of raw material

As mentioned in section 1, the demand rate of raw material inventory for the products at the first plant is p_1 , the production rate of plant 1. Many orders arrive on time when an order is placed and also shortage is not allowed. So, the input rate is considered as infinite. In this SCS, the economic order quantity (EOQ) is divided into a number of equal batches. When the production starts, the shipment is set at a fixed interval during one period. The inventory level is illustrated in Fig. 4.

$$I_{max} = T_{P0}(P_0 - P_1) \quad (1)$$

$$T_{P0} = \frac{Q_r}{P_0} \quad (2)$$

$$I_{max} = Q_r \left(1 - \frac{P_1}{P_0}\right) \quad (3)$$

$$I_{min} = 0$$

$$I_{ave} = \frac{I_{max} + I_{min}}{2} = \frac{Q_r}{2} \left(1 - \frac{P_1}{P_0}\right) \quad (4)$$

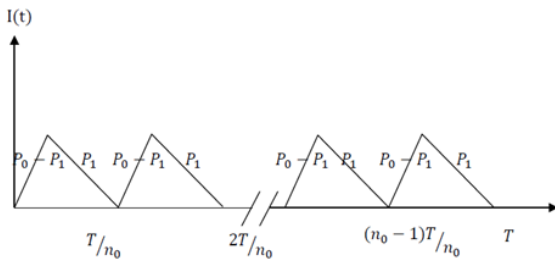


Fig. 4. Raw material [23]

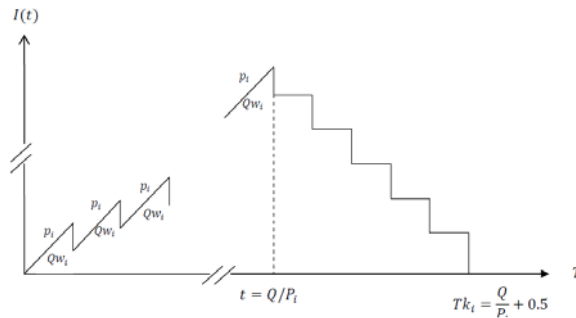


Fig. 5. The WIP of an intermediate stage (store γ) [23]

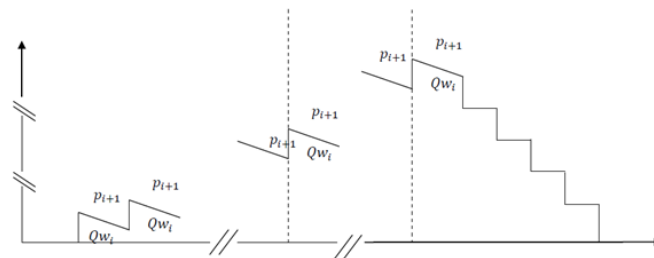


Fig. 6. The WIP of an intermediate stage (store λ) [23]

4.2. Cost of WIP inventory

Cost of WIP inventory consists of holding inventory cost, container transshipment cost, and setup cost of product 3 [23]. At first, plant i , produces at the rate of p_i and puts full container in store γ . Then filled containers are transshipped to the next plant and stay in store B to be used by the succeeding plant at the rate of p_{i+1} . Thus holding inventory cost occurs in both stores γ and λ . To facilitate formulation of this problem, some parameters and variables are needed as follows:

- COI_{Ai} Store A holding cost of inventory at stage i
- COI_{Bi} Store B holding cost of inventory at stage i
- IAi_{max} Store A maximum inventory level at stage i
- IBi_{max} Store B maximum inventory level at stage i
- IAi_{ave} Store A average inventory at stage i
- IBi_{ave} Store B average inventory at stage i
- COI_i Holding inventory cost at stage i

As mentioned before, holding inventory cost of stage i is the sum of λ and γ inventory holding costs:

$$COI_i : COI_{Ai} + COI_{Bi}$$

Figs. 5 and 6 respectively show the inventory levels of stores λ and γ at stage i . In the base of presented figures, it could be seen clearly that the exported goods, Q_{wi} from store γ , are added to store λ and used at the rate of p_{j+1} . It is better to obtain the time of the last container transshipment from store γ to store λ (T_{ki}).

As shown in Fig. 6, and as plant i , and $i + 1$ product cycle time, and since full containers transshipment is done in a fixed time interval, then:

$$T_{ki} = \frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \quad (5)$$

During the rest of the discussion, by implicated T_{ki} both I_{ave} and I_{max} will never be negative. I_{Aimax} , I_{Aiave} , COI_{Ai} can be calculated by:

$$I_{Aimax} = Q_{wi}(K_j - m_j + 1) \quad (6)$$

$$I_{Aiave} = \frac{1}{2} Q_{wj}(K_j - m_j + 1) \quad (7)$$

$$COI_{Aj} = COI_{Aj} \times \frac{1}{2} Q_{wi}(K_j - m_j + 1) \quad (8)$$

By defining α_{1i} as follows: α_{1i} is the ratio of total time that goods stayed in store A to total cycle time:

$$\alpha_{1i} = \frac{\frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right)}{\frac{D}{2} \left(\frac{1}{P_{i+1}} + \frac{1}{P_i} \right)} \quad (9)$$

These relations yield

$$COI_{Ai} = \frac{1}{2} \left(\frac{H_{wi} \times \alpha_{1i}}{2} \times Q_{wi}(K_i - m_i + 1) + \frac{H_{wi} \times \alpha_{1i}}{2} \times Q_{wi} + \frac{H_{wi} \times \alpha_{1i}}{2} \times Q \left(1 - \frac{m_i}{k_i} \right) \right) \quad (10)$$

$$\frac{m_i}{k_i} = \frac{T_{mi}}{T_{ki}} = \frac{\frac{Q}{P_i}}{\frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right)} \quad (11)$$

$$COI_{Ai} = \frac{H_{wi} \times \alpha_{1i}}{2} \times Q_{wi} + \frac{H_{wi} \times \alpha_{1i}}{2} \times Q \left(1 - \frac{2}{1 + \frac{P_i}{P_{i+1}}} \right) \quad (12)$$

To calculate the inventory holding cost in store B, we can write

$$I_{Bimax} = (K_i \times Q_{wi}) + P_{i+1} \quad (13)$$

$$\frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) = Q - Q \left[\frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right] \quad (14)$$

$$I_{Biave} = \frac{1}{2} Q \left[1 - \left[\frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right] \right] \quad (15)$$

And α_{2i} can be defined as the ratio of total time that goods stayed in store B to the total cycle time. So, we can write

$$\alpha_{2i} = \frac{\frac{Q}{P_{i+1}}}{\frac{Q}{Q_i}} = \frac{Q}{P_{i+1}} \quad (16)$$

$$COI_{Bi} = \frac{H_{wi} \times \alpha_{2i}}{2} \times Q \times \left(1 - \frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) \quad (17)$$

$$TC_{wi} = A_{si} \frac{D}{Q} + A_{wi} \frac{D}{Q_{wi}} + \frac{H_{wi} \times \alpha_{1i}}{2} \times Q \left(1 - \frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) + \frac{H_{wi} \times \alpha_{1i}}{2} \times Q_{wi} + \frac{H_{wi} \times \alpha_{1i}}{2} \times Q \left(1 - \frac{2}{1 + \frac{P_i}{P_{i+1}}} \right) \quad (18)$$

With regard to

$$Q = Q_f \times K_N$$

$$Q = Q_r \times n_0$$

$$Q = Q_{wi} \times K_i$$

So:

$$TC_{wi} = [A_{si} + A_{wi} \times k_i] + Q \left[\frac{H_{wi} \times \alpha_{1i}}{2 \times K_i} + \frac{H_{wi} \times \alpha_{2i}}{2} \times \left(1 - \frac{1}{2} \left(\frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) \right] + \frac{H_{wi} \times \alpha_{1i}}{2} \times \frac{H_{wi} \times \alpha_{1i}}{2} \times \left(1 - \frac{2}{1 + \frac{P_i}{P_{i+1}}} \right) \quad (19)$$

By the following definition:

$$B_{1i} = \frac{H_{wi} \times \alpha_{1i}}{2} \quad (20)$$

$$B_{2i} = \frac{H_{wi} \times \alpha_{2i}}{2} \times \left(1 - \frac{1}{2} \left(\frac{Q}{P_{i+1}} + Q P_i \right) \right) + \frac{H_{wi} \times \alpha_{1i}}{2} \times \frac{H_{wi} \times \alpha_{1i}}{2} \times \left(1 - \frac{2}{1 + \frac{P_i}{P_{i+1}}} \right) \quad (21)$$

So

$$TC_{wi} = \frac{D}{Q} [A_{si} + A_{wi} \times k_i] + Q \left[\frac{B_{1i}}{K_i} + B_{2i} \right] \quad (22)$$

4.3. Cost of finished goods

The cost of finished goods is related to the N -stage, i.e. the cost obtained from plant N , and plant $N + 1$ (customer) rate of customer consumption is D and in attention to P_{N+1} is also customer consumption rate, in this station it can be written $P_{N+1} = D$. Whereas the cost of finished goods is similar to WIP cost and $D = P_{N+1}$, $i = N$ in this stage. The cost of finished goods is as follows:

$$TC_{wf} = \frac{D}{Q} [A_{sf} + A_{wff} \times k_i] + Q \left[\frac{B_{1N}}{K_N} + B_{2N} \right] \quad (23)$$

In order to calculate the total SCS cost, let $x_0, x_1, \dots, x_N, A_{SN}, A_{WW}$ replace $n_0, k_1, \dots, k_N, A_{sf}, A_{wff}$, so the general form of MSSCS is given by

$$TC = \frac{D}{Q} \left[\sum_{i=1}^N A_{si} + \left(\sum_{i=1}^N A_{si} \times x_i \right) + \sum A_r \times x_0 \right] + Q \left[\sum_{i=1}^N \frac{B_{1i}}{x_i} + \sum_{i=1}^N B_{2i} + \frac{H_r(1 - \frac{P_1}{P_0})}{2 \times x_0} \right] \quad (24)$$

x_i : integer

$$\alpha_{2i} = \frac{\frac{Q}{P_{i+1}}}{\frac{Q}{Q_i}} = \frac{Q}{P_{i+1}} \quad (25)$$

$$B_{1i} = \frac{H_{wi} \times \alpha_{1i}}{2} \quad (26)$$

$$B_{2i} = \frac{H_{wi} \times \alpha_{2i}}{2} \times \left(1 - \frac{1}{2} \left(\frac{Q}{P_{i+1}} + Q P_i \right) \right) + \frac{H_{wi} \times \alpha_{1i}}{2} \times \frac{H_{wi} \times \alpha_{1i}}{2} \times \left(1 - \frac{2}{1 + \frac{P_i}{P_{i+1}}} \right) \quad (27)$$

By relaxed mixed integer non-linear programming (MINLP), the optimum total cost of MSSCS is obtained as follows:

$$TC^* = \sqrt{2D} \left(\sqrt{2 \sum_{i=1}^N B_{2i} \times \sum_{i=1}^N A_{si}} + \sum_{i=1}^N \sqrt{2B_{1i} \times A_{wi}} + \sqrt{A_r \times H_r \left(1 - \frac{P_1}{P_0} \right)} \right) \quad (28)$$

$$Q^* = \sqrt{\frac{D \sum_{i=1}^N A_{si}}{\sum_{i=1}^N B_{2i}}} \quad (29)$$

$$x_i^* = Q^* \sqrt{\frac{B_{1i}}{D \times A_{wi}}} \quad (30)$$

$$x_0^* = Q^* \sqrt{\frac{H_r(1 - \frac{p_1}{p_0})}{2D \times A_{ri}}} \quad (31)$$

This optimum TC^* is an ideal solution of the problem. The real feasible optimum is always worse than this ideal optimum. The ideal optimum can be assumed as the lower bound.

5. Proposed algorithm

Combinatorial optimization problems are complex and hard to be solved; some exact methods such as B&B are used to obtain the solution [16]. Due to the complexity of such problems, some powerful approximation methods are needed that although they do not ensure an exact solution, they can find a solution close to the exact one in a shorter period of time. In order to analyze the complexity of the B&B method used to find the exact solutions, the number of considered and evaluated nodes are measured and shown as a function of (n) that is almost $O(n^n)$ even if the worst case of B&B may be exponential [3]. In order to do so, we first need to present our encoding scheme which makes a solution recognizable for the algorithms. Then, the electromagnetic algorithm is reviewed.

Electromagnetism EA is a population-based meta-heuristic proposed to solve continuous problems effectively. This method originates from the electromagnetism theory of physics by considering potential solutions as electrically charged particles spread around the solution space. Birbil and Fang [3] proposed the EA as a flexible and effective population-based algorithm to search for the optimal solution of global optimization problems. This meta-heuristic utilizes an attraction-repulsion mechanism to move the particles towards optimality. EA is useable for a particular set of optimization problems with bounded variables in the form of:

Min $f(x)$

st: $x \in [l, u]$,

where $[l, u] := \{x \in R^n \mid l_k \leq x_k \leq u_k; k = 1, 2, \dots, n\}$

Each candidate solution as a charged particle is considered. The charge of each candidate solution is related to objective function. The size of attraction or repulsion over candidate solutions in the population is calculated by this charge. The direction of this charge for candidate solution i is determined by adding vectorally the forces from each of other solutions on candidate solution i . In this mechanism, a candidate solution with good objective function value attracts the other ones; candidate solutions with worse objective function repel the other population members; and a better objective function value result in a higher size of attraction. As shown in Fig. 7, EA has four phases including

Initialization of algorithm, computation of total force exerted on each particle, movement along the direction of the force and local search.

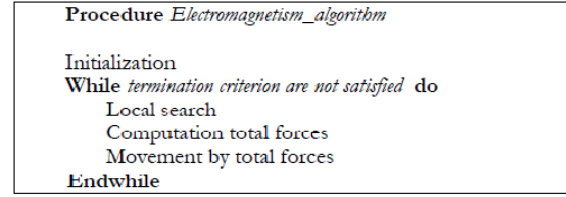


Fig. 7: Pseudo code of an EA

Procedure Electromagnetism algorithm

Step-1: Initialize

Step-2: Compute total forces

Step-3: Move by total forces

Step-4: Apply local search

Step-5: If the stopping criterion is not met, go to step-2; otherwise Stop.

5.1. Initialization

The first procedure, Initialization, is used for sampling m points from the feasible region and assigning them their initial TC . The initialization procedure in such a hard combinatorial problem that has to be made with great care, to ensure convergence to desirable, better objective functions in a reasonable amount of time. In order to generate an initial TC , number of k initial solutions are generated, some of which should be close to the solutions gained by Eq. (21). In this paper, this initial solution by calculating floors and ceilings of gens of the infeasible optimal solution obtained from Eq. (21) is generated. Hence a new feasible solution is generated. Now this action to construct other initial TC is repeated. The TC of solutions is calculated and the best one is recorded as x_{best} .

5.2. Local search engine

The proposed EA is hybridized with a local search in order to improve the performance of the algorithm. The procedure of this local search can be described as follows: The random key of the first solution (x_{i1}) in the sequence of candidate solution i (x_i) is randomly regenerated. If this new sequence (v) results in a better TC , the current solution (x_i) is replaced by the new sequence (v). If improvement in $kth < g$ has, the local search for the current solution terminates. After all, the best solution is updated. The procedure of the local search is shown in Fig. 8.


```

Procedure Local_search
  k = 1
  while k < n+1 do
    v = xik is randomly regenerated
    if f(v) < f(xi) then
      xi = v
      k = n
    endif
    k = k + 1
  endwhile
  
```

Fig. 8: procedure of the local search

5.3. Computation of total forces

In order to compute the force between two points, a charge-like value q_i , is assigned to each point. The charge of the point is calculated according to the relative efficiency of the TCs in the current population, i.e.

$$q_i = \exp\left(-N \cdot \frac{f(x_i) - f(x_{best})}{\sum_{j=1}^{popsize} (f(x_j) - f(x_{best}))}\right), \forall i \quad i = 1, 2, \dots, popsize \quad (32)$$

where x_{best} represents the point that has the best TC among the points at the current iteration. In this way the points that have better TCs possess higher charges. Note that, unlike electrical charges, no signs are attached to the charge of an individual point in the Eq. (33) Instead, the direction of a particular force between two points will be determined after comparing their TCs. The total force F_i exerted on the candidate solution i is also calculated by the following formula:

$$F_i = \begin{cases} \sum_{j \neq i}^p (x_j - x_i) \frac{q_i \cdot q_j}{\|x_j - x_i\|^2} ; & f(x_j) < f(x_i) \\ \sum_{j \neq i}^p (x_i - x_j) \frac{q_i \cdot q_j}{\|x_j - x_i\|^2} ; & f(x_i) < f(x_j) \end{cases} \quad (33)$$

The two-dimensional example total force vector F_i exerted on the candidate solutions is shown in Fig. 9. The force exerted by x_1 on x_3 is F_{13} (repulsion: the TC of x_1 is worse than that of x_3) and the force exerted by x_2 on x_3 is F_{23} (attraction: the TC of x_2 is better than that of x_3). F_3 is the total force exerted on x_3 by x_1 and x_2 . The procedure of the computation of total forces is shown in Fig. 10.

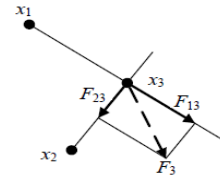


Fig. 9: Example of exertion of forces

```

Procedure computation_of_total_forces
  for i = 1 to popsize do
    calculate (qi)
    Fi = 0
  endfor
  for i = 1 to popsize do
    for j = 1 to popsize do
      if f(xj) < f(xi) then
        Fi = Fi + (xj - xi) * (qi * qj) / \|xj - xi\|^2 (attraction)
      else
        Fi = Fi - (xj - xi) * (qi * qj) / \|xj - xi\|^2 (repulsion)
      endif
    endfor
  endfor
  endfor
  
```

Fig. 10. The procedure of the computation of total forces

5.4. Movement by total forces

The total force exerted on each point by all other points is calculated in the CalcF() procedure. All the candidate solutions are moved with the exception of the current best solution. The move for each candidate solution is in the direction of total force exerted on it by a random step length. This length is generated from the uniform distribution between (0, 1). The candidate solutions have a nonzero probability to move to the unvisited solution along this direction by selecting a random length and by normalizing total force exerted on each candidate solution as Fig. 11 depicts.

```

Procedure Movement_by_total_forces
  for i = 1 to popsize do
    if i ≠ best then
      β = random [0, 1]
      Fi = Fi / \|Fi\|
      for k = 1 to n do
        if Fik > 0 then
          xik = xik + β * Fik (1 - xik)
        else
          xik = xik + β * Fik (xik)
        endif
      endfor
    endif
  endfor
  endfor
  
```

Fig. 11. Movement procedure by total forces

6. Algorithm's Calibration

It is well known that the quality of algorithms is significantly influenced by the values of parameters. In order to tune the algorithms, a full factorial design is applied in the design of experiment (DOE) approach [13].

One of the advantages of EA is that it has only one parameter, *popsize* (number of population). The results are analyzed by the means of the analysis of variance (ANOVA) method. The means plot and least significant differences (LSD) intervals for the levels of *popsize* parameter factor are shown in Fig. 12.

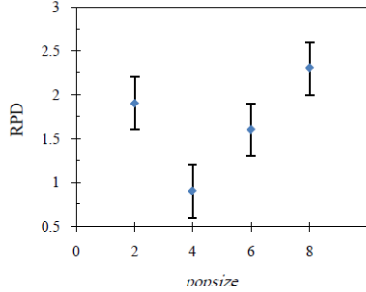


Fig. 12. Means plot and LSD intervals for number of population in EA

This figure depicts that there are statistical differences between different values of *popsize*, and this implies that *popsize* factor is an important parameter in EA. In our case, *popsize* of 4 provides statistically better results than the other values of *popsize* = 2, 6, 8.

7. Computational Evaluation

In this section, the proposed EA is compared with other existing methods. From the literature, memetic algorithm (MA) proposed by Rabbani et al. [16], and

B&B are brought into the comparison. The algorithms are implemented in MATLAB 7.0 and run on a PC with 2.0 GHz Intel Core 2 Duo and 1 GB of RAM memory. We use relative percentage deviation (RPD) as the performance measure to compare the methods. When the TC of each algorithm is obtained for a given instance, the best solution obtained (which is named L_B) is calculated by any of the algorithms. RPD is obtained by the formula

$$RPD = 100 * \left(\frac{A_{sol} - L_B}{L_B} \right) \tag{35}$$

where A_{sol} is the TC botained for a given algorithm and instance. Obviously, the lower values of RPD are preferable. The stopping criterion is 2N seconds computational time. This stopping criterion permits for more time as the number of stages increases.

The algorithms are evaluated by a set of instances taken from the literature. The largest size of these instances is $N = 30$, but in the real world, industries usually encounter problems up to $N = 100$. Therefore, in addition to the instances taken from the literature, we generate some other instances based on that benchmark. The results of the experiments are shown in Table 1. EA provides better results compared with the MA proposed by Rabbani et al. [16] with RPD of 0.10%. Considering the results, there is a clear trend that MA, like EA, obtains the optimal solution in small-sized instances up to $N = 30$; however, comparatively speaking, its performances gets worse in larger instances.

Table 1
The total cost obtained by proposed model and algorithms

Number of Plants	Ideal solutions	Algorithm				
		B&B	MA		EA	
			TC	TC	RPD	TC
N = 3	25774.87	25802.60	25802.60	0.00	25802.60	0.00
N = 4	29330.33	29355.08	29355.08	0.00	29355.08	0.00
N = 5	34125.74	34172.61	34172.61	0.00	34172.61	0.00
N = 6	38627.23	38665.64	38665.64	0.00	38665.64	0.00
N = 7	43335.03	43368.28	43368.28	0.00	43368.28	0.00
N = 8	50142.95	50169.60	50193.07	0.05	50169.60	0.00
N = 9	56264.61	56286.75	56286.75	0.00	56286.75	0.00
N = 10	58107.89	58132.31	58132.31	0.00	58132.31	0.00
N = 16	113180.44	113214.60	113217.05	0.00	113214.60	0.00
N = 30	144863.05	144933.30	144958.31	0.02	144933.30	0.00
N = 42	173263.42	174026.36	178931.48	2.82	174026.36	0.00
N = 50	219257.09	222439.44	225643.02	1.44	222439.44	0.00
N = 58	256359.12	260243.89	264332.31	1.57	262117.81	0.72
N = 72	294197.63	300843.21	311291.22	3.47	301452.93	0.20
N = 90	406263.34	411234.16	416118.68	1.19	412365.64	0.28
N = 100	488869.15	495637.02	499520.74	0.78	497348.23	0.35
Average				0.71%		0.10%

Table 2
ANOVA table for RPDs

Factor	df	SS	MS	F	P-Value
Algorithm	1	2.995	2.995	4.70	0.038
Error	30	19.131	0.638		
Total	31	22.126			

Figure 13 shows the mean plots and LSD intervals for EA and MA. According to the figure, at the confidence level of 95%, there is a significant difference between EA and MA, and EA statistically outperforms MA.

8. Conclusion and Future Study

To conclude, we recall that, in this work, we investigated the MSSCM which is controlled by the kanban mechanism. Since our problem is NP-Hard, we have adopted the electromagnetic algorithm (EA) to solve it. The encoding scheme was adapted to facilitate the use of EA to boost the performance of the proposed algorithm. Then, we presented a mathematical model in the form of MILP to produce optimal solutions. Afterwards, to evaluate the competitiveness of the proposed EA, we conducted an experiment in which we compared the comparative performance of EA against a high performing metaheuristic

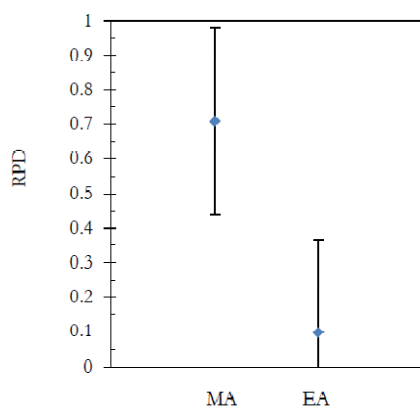


Fig. 13. Means plots and LSD intervals fo EA and MA

algorithm in the literature. The results clearly manifest the excellent and high performance of our proposed algorithm. It is interesting to indicate that EA outperformed the other algorithms.

As a direction for future studies, it can be interesting to apply EA to other NP-hard problems such as supply chain problems with various practical assumptions. Another orientation for future research can be the multi-objective consideration of different objective functions.

9. References

- [1] T. Altioek, R. Ranjan, Multi-stage, pull-type production/inventory systems. *IIE Transactions*, 27, 190-200, 1995.
- [2] B. Berkley, A review of the kanban production control research literature. *Production and Operations Management*, 4, 393-411, 1992.
- [3] I. Birbil, S. Fang, An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25, 263-282, 2003.
- [4] H. Co, M. Sharafali, Over planning factor in Toyota formula for computing the number of kanban. *IIE Transactions*, 29, 409-415, 1997.
- [5] I. Chang, H. Hwang, H. Liaw, M. Hung, S.Chen, D. Yen, A neural network evaluation model for ERP performance from SCM perspective to enhance enterprise competitive advantage. *Expert Systems with Applications*, 35, 1809-1816, 2008.
- [6] R. Ganeshan, Managing supply chain inventories: A multiple retailer, one warehouse, multiple supplier model. *International Journal of Production Economics*, 59, 341-354, 1999.
- [7] S. Gupta, Y. AlTurki, An algorithm to dynamically adjust the number of kanbans in stochastic processing times and variable demand environment. *Production Planning and Control*, 8, 133-141, 1997.
- [8] D. Hutchins, *Just in time*, Madras: Productivity Press Ltd, 1993.
- [9] G. Hult, D. Ketchen, S. Cavusgil, R. Calantone, Knowledge as a strategic resource in supply chains. *Journal of Operations Management*, 24, 458-475, 2005.
- [10] M. Mascolo, Y. Frein, Y. Dallery, An analytical method for performance evaluation of kanban controlled production systems. *Operations Research*, 44, 50-64, 1996.
- [11] S. Miyazaki, H. Ohta, N. Nishiyama, The optimal operation planning of kanban to minimize the total operation cost. *International Journal of Production Research*, 26, 1605-1611, 1988.
- [12] Y. Monden, *The Toyota Production System*. Industrial Engineering and Management Press. Norcross, GA, 1983.
- [13] D. Montgomery, *Design and Analysis of Experiments*, 5th edition, John Wiley and Sons, USA, 2000.
- [14] V. Nori, B. Sarker, Cyclic scheduling for a multi-product, single-facility production system operating under a just-in-time delivery policy. *Journal of the Operational Research Society*, 47, 930-935, 1996.
- [15] G. Parija, B. Sarker, Operations planning in a supply chain system with fixed-interval deliveries of finished goods to multiple customers. *IIE Transactions*, 31, 1075-1082, 1999.
- [16] M. Rabbani, J. Layegh, R. Mohammad Ebrahim, Determination of number of kanbans in a supply chain system via Memetic algorithm. *Advances in Engineering Software*, 40, 431-437, 2009.
- [17] L. Rees, P. Philipoom, B. Taylor, P. Huang, Dynamically adjusting the number of kanbans in a just-in-time production system using estimated values of lead time. *IIE Transactions*, 19, 199-207, 2006.
- [18] B. Sarker, C. Balan, Operations planning for kanbans between two adjacent workstations. In: Randhawa, Sabah U. (Ed.), *Proceedings of the 19th International Conference on Computers and Industrial*

Engineering, Miami, FL Elsevier Science Ltd, 1996.

- [19] B. Sarkar, C. Balan, Operations planning for a single-stage kanban system operating under linear demand. *International Journal of Production Research*, 36, 357-375, 1998.
- [20] B. Sarkar, C. Balan , Operations planning for a multi-stage kanban system. *European Journal of Operational Research*, 112, 284-303, 1999.
- [21] K. Sengupta, D. Heiser, L. Cook, Manufacturing and service supply chain performance: a comparative Analysis. *Journal of Supply Chain Management*, 42, 4-15, 2006.
- [22] P. Shahabudeen,R. Gopinath, K. Krishnaiah, Design of bi-criteria kanban system using simulated annealing technique. *Computers and Industrial Engineering*, 41, 355-370, 2002.
- [23] S. Wang, B. Sarker, Optimal model for supply chain system controlled by kanban under just-in-time philosophy. *European Journal of Operational Research*, 172:179-200, 2006.