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

Majid Khalili, Mohammad Jafar Tarkh, Bahman Naderi

Islamic Azad University, Karaj branch, Department of Industrial Engineering, Alborz, Iran
Department of Industrial Engineering, K.N.Toosi University, Tehran, Iran
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, manufacturers, 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
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

\[
P_i = P_{i+1} \]

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.

Altiok 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

---

**Fig. 1. A Configuration of the N-stage kanban system (N + 1 workstations)**
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:

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 \( \gamma \) is a store after plant \( j \) and store \( \lambda \) 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 \( \gamma \) should wait and then any filled transporter transshipped with its kanban, is carried to the succeeding plant and put in store \( \lambda \).

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.

\[ I_{\text{max}} = T_{po}(P_0 - P_1) \]  
\[ T_{po} = \frac{Q_0}{P_0} \]  
\[ I_{\text{max}} = Q_r(1 - \frac{P_1}{P_0}) \]  
\[ I_{\text{min}} = 0 \]  
\[ I_{\text{ave}} = \frac{I_{\text{max}} + I_{\text{min}}}{2} = \frac{Q_0}{2}(1 - \frac{P_1}{P_0}) \]  

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 $\gamma$. 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 $\gamma$ and $\lambda$. To facilitate formulation of this problem, some parameters and variables are needed as follows:

- $COI_{\lambda i}$: Store A holding cost of inventory at stage $i$
- $COI_{B i}$: Store B holding cost of inventory at stage $i$
- $IA_{\lambda \text{max}}$: Store A maximum inventory level at stage $i$
- $IB_{\lambda \text{max}}$: Store B maximum inventory level at stage $i$
- $IA_{\lambda \text{ave}}$: Store A average inventory at stage $i$
- $IB_{\lambda \text{ave}}$: Store B average inventory at stage $i$
- $COI_{\lambda i}$: Holding inventory cost at stage $i$

As mentioned before, holding inventory cost of stage $i$ is the sum of $\lambda$ and $\gamma$ inventory holding costs:

\[ COI_{\lambda i} = COI_{\lambda i} + COI_{B i} \]

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

\[
T_{ki} = \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right)
\]

During the rest of the discussion, by implicated \(T_{ki}\) both \(I_{low}\) and \(I_{max}\) will never be negative. \(I_{max}, I_{low}\), \(COI_{dl}\) can be calculated by:

\[
I_{max} = Q_{wi}(K_i - m_i + 1)
\]

\[
I_{low} = \frac{1}{2} Q_{wi}(K_i - m_i + 1)
\]

\[
COI_{dl} = COI_{dl} \times \frac{1}{2} Q_{wi}(K_i - m_i + 1)
\]

By defining \(\alpha_{l1}\) as follows: \(\alpha_{l1}\) is the ratio of total time that goods stayed in store A to total cycle time:

\[
\alpha_{l1} = \frac{\frac{Q}{P_{i+1}} + \frac{Q}{P_i}}{2}
\]

These relations yield

\[
COI_{dl} = \frac{H_{wi} \times \alpha_{l1}}{2} \times Q_{wi} (K_i - m_i + 1) + \frac{H_{wi} \times \alpha_{l1}}{2} \times Q_{wi} (1 - \frac{m_i}{K_i})
\]

\[
m_i = \frac{T_{mi}}{T_{ki}} = \frac{\frac{Q}{P_i}}{\frac{Q}{P_{i+1}} + \frac{Q}{P_i}}
\]

\[
COI_{dl} = \frac{H_{wi} \times \alpha_{l1}}{2} \times Q_{wi} + \frac{H_{wi} \times \alpha_{l1}}{2} \times Q (1 - \frac{m_i}{1 + \frac{Q}{P_{i+1}}})
\]

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

\[
I_{max} = (K_i \times Q_{wi}) + P_{i+1}
\]

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

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

And \(\alpha_{l2}\) can be defined as the ratio of total time that goods stayed in store \(B\) to the total cycle time. So, we can write

\[
\alpha_{l2} = \frac{\frac{Q}{P_{i+1}} + \frac{Q}{P_i}}{\frac{Q}{P_{i+1}} + \frac{Q}{P_i}}
\]

\[
COI_{bl} = \frac{H_{wi} \times \alpha_{l2}}{2} \times Q \left( 1 - \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right)
\]

\[
TC_{uli} = A_i \left[ \frac{Q}{4} + A_{wi} \frac{D}{x_0} + \frac{HN_{wi}k_{i+1}}{2} \right] + Q \left( 1 - \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) + \frac{HN_{wi}k_i}{2}
\]

\[
Q = Q_{i} \times K_i
\]

\[
Q = Q_{i} \times N
\]

\[
Q = Q_{wi} \times K_i
\]

So:

\[
TC_{uli} = \left[ A_i \left[ \frac{Q}{4} + A_{wi} \frac{D}{x_0} + \frac{HN_{wi}k_{i+1}}{2} \right] + Q \left( 1 - \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) + \frac{HN_{wi}k_i}{2} \right]
\]

By the following definition:

\[
B_{i1} = \frac{H_{wi} \times \alpha_{l1}}{2}
\]

\[
B_{i2} = \frac{H_{wi} \times \alpha_{l2}}{2} \times \left( 1 - \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) + \frac{HN_{wi}k_i}{2}
\]

So

\[
TC_{uli} = \frac{D}{Q} \left[ A_i + A_{wi} \times k_i + Q \left( \frac{HN_{wi}k_i}{2} \times \left( 1 - \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) \right) \right]
\]

\[
4.3. Cost of finished goods
\]

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 \(P_{N + i}\) is also customer consumption rate, in this station it can be written \(P_{N + i} = D\). Whereas the cost of finished goods is similar to WIP cost and \(D = P_{N + i}, i = N\) in this stage. The cost of finished goods is as follows:

\[
TC_{uli} = \frac{D}{Q} \left[ A_i + A_{wi} \times k_i + Q \left( \frac{B_{i1} + B_{i2}}{K_i} \right) \right]
\]

To calculate the total cost of MSSCS, let \(x_0, x_1, \ldots, x_N, A_{wi}, A_{wi}w\) replace \(n_x, K_i, \ldots, k_N, A_{wi}w\), so the general form of MSSCS is given by

\[
TC = \frac{D}{Q} \left[ \sum_{i=1}^{N} A_i \left( \frac{Q}{4} + \frac{D}{x_0} + \frac{HN_{wi}k_{i+1}}{2} \right) + Q \sum_{i=1}^{N} B_{i1} + \sum_{i=1}^{N} B_{i2} + \frac{HN_{wi}k_i}{2} \right]
\]

\[
x_{i\text{: integer}}
\]

\[
\alpha_{l2} = \frac{\frac{Q}{P_{i+1}} + \frac{Q}{P_i}}{\frac{P_{i+1}}{P_i}}
\]

\[
B_{i1} = \frac{H_{wi} \times \alpha_{l1}}{2}
\]

\[
B_{i2} = \frac{H_{wi} \times \alpha_{l2}}{2} \times \left( 1 - \frac{1}{2} \left( \frac{Q}{P_{i+1}} + \frac{Q}{P_i} \right) \right) + \frac{HN_{wi}k_i}{2}
\]

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

\[
TC^* = \sqrt{2D} \left( 2 \sum_{i=1}^{N} B_{i2} \times \sum_{i=1}^{N} A_{wi} + \sum_{i=1}^{N} B_{i1} \times A_{wi} \right)
\]

\[
Q^* = \frac{Q}{\sqrt{2D}}
\]
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 \( O(n^2) \) 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:

\[
\text{Min } f(x) \quad \text{st: } x \in [l, u],
\]

where \( [l, u] = \{x \in \mathbb{R}^n | l_k \leq x_k \leq u_k \; ; k = 1,2,\ldots,n\} \)

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 each of other solutions on 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

\[
x^* = Q^* \frac{F(x)}{2 \pi \Delta x^*}
\]

\[
x^0 = Q^* \frac{[x_l(1-x\Delta x^*)]}{4 D x^*}
\]

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.

5.1. 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 \( (x_{i,1}) \) 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 \( k^{th} < g \) 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.
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.

$$q_i = \exp \left( -\sum_{j=1}^{\text{popsize}} \frac{f(x_i) - f(x_{\text{best}})}{f(x_j) - f(x_{\text{best}})} \right), \forall i$$  \hspace{1cm} (32)

where $x_{\text{best}}$ 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:

$$F_i = \begin{cases} \sum_{j=1}^{\text{popsize}} q_i q_j \|x_j - x_i\|^2 ; & f(x_j) < f(x_i) \\ \sum_{j=1}^{\text{popsize}} q_i q_j \|x_j - x_i\|^2 ; & f(x_i) < f(x_j) \end{cases}$$  \hspace{1cm} (33)

The two-dimensional example total force vector $F_i$ exerted on the candidate solution 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.

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.

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.

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 given below:

$$ RPD = 100 \times \left( \frac{A_{sol} - L_{B}}{L_{B}} \right) $$

where $A_{sol}$ is the TC obtained 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>
<thead>
<tr>
<th>Number of Plants</th>
<th>Ideal solutions</th>
<th>B&amp;B</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 3</td>
<td>25774.87</td>
<td>25802.60</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>N = 4</td>
<td>29330.33</td>
<td>29355.08</td>
<td>29355.08</td>
</tr>
<tr>
<td>N = 5</td>
<td>34125.74</td>
<td>34172.61</td>
<td>34172.61</td>
</tr>
<tr>
<td>N = 6</td>
<td>38627.23</td>
<td>38665.64</td>
<td>38665.64</td>
</tr>
<tr>
<td>N = 7</td>
<td>43335.03</td>
<td>43368.28</td>
<td>43368.28</td>
</tr>
<tr>
<td>N = 8</td>
<td>50142.95</td>
<td>50169.60</td>
<td>50193.07</td>
</tr>
<tr>
<td>N = 9</td>
<td>56264.61</td>
<td>56286.75</td>
<td>56286.75</td>
</tr>
<tr>
<td>N = 10</td>
<td>58107.89</td>
<td>58132.31</td>
<td>58132.31</td>
</tr>
<tr>
<td>N = 11</td>
<td>113180.44</td>
<td>113214.60</td>
<td>113217.05</td>
</tr>
<tr>
<td>N = 12</td>
<td>144863.05</td>
<td>144958.31</td>
<td>144958.31</td>
</tr>
<tr>
<td>N = 13</td>
<td>173263.42</td>
<td>174026.36</td>
<td>174026.36</td>
</tr>
<tr>
<td>N = 14</td>
<td>219257.09</td>
<td>222439.44</td>
<td>222439.44</td>
</tr>
<tr>
<td>N = 15</td>
<td>256359.12</td>
<td>260243.89</td>
<td>260243.89</td>
</tr>
<tr>
<td>N = 16</td>
<td>294197.63</td>
<td>300843.21</td>
<td>301129.22</td>
</tr>
<tr>
<td>N = 17</td>
<td>406263.34</td>
<td>411234.16</td>
<td>411234.16</td>
</tr>
<tr>
<td>N = 18</td>
<td>488869.15</td>
<td>495637.02</td>
<td>499520.74</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>0.71%</td>
</tr>
</tbody>
</table>
Table 2

ANOVA table for RPDs

<table>
<thead>
<tr>
<th>Factor</th>
<th>df</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>1</td>
<td>2.995</td>
<td>2.995</td>
<td>4.70</td>
<td>0.038</td>
</tr>
<tr>
<td>Error</td>
<td>30</td>
<td>19.131</td>
<td>0.638</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>31</td>
<td>22.126</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 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

Engineering, Miami, FL Elsevier Science Ltd, 1996.


Surf and download all data from SID.ir: www.SID.ir

Translate via STRS.ir: www.STRS.ir

Follow our scientific posts via our Blog: www.sid.ir/blog

Use our educational service (Courses, Workshops, Videos and etc.) via Workshop: www.sid.ir/workshop