

Uncertain Supply Chain Management

homepage: www.GrowingScience.com/uscm

A robust optimization approach for scheduling a supply chain system considering preventive maintenance and emergency services using a hybrid ant colony optimization and simulated annealing algorithm

Aidin Delgoshaei^{a*}, Armin Delgoshaei^b, Aisa Khoushniat Aram^c and Ahad Ali^d

^aDepartment of Industrial Engineering, Kharazmi University, Tehran, Iran

^bKhaje Nasir Toosi University of Technology, Tehran, Iran

^cUniversity Putra Malaysia, Malaysia

^dLawrence Technological University, United States

CHRONICLE

Article history:

Received June 22, 2018

Accepted September 25 2018

Available online

October 3 2018

Keywords:

Facilities planning and design

Supply Chain Scheduling

Machine Failure

Preventive Maintenance

ABSTRACT

Machine failures during production period may impose thousands to millions of dollars to a manufacturing system. In this paper, the impact of machine failures on production lines in a closed-loop supply chain systems is examined. For this purpose, a new method is proposed for scheduling manufacturing workshops in a supply chain systems. The aim is to determine the best production plans in a manufacturing system by considering alternative preventive maintenance programs while machine failures can affect system performance. To solve the model, a hybrid Ant Colony and Simulated Annealing algorithms is developed and the results are compared with branch and bound method. Our findings show that the condition of emerging machine failure affects machines' capacity which yields to lost sale. The impacts of using appropriate preventive maintenance on reducing lost sale is also examined. The results indicate that the proposed method can significantly reduce the level of sale variation in supply chain systems.

© 2018 by the authors; licensee Growing Science, Canada

1. Introduction

In industrial world, supply chain management (SCM) is referred to the systematic and strategic management of value added flows of goods and services from purchasing the raw materials until delivery and after sale services (Azmi et al., 2018). Such strategies require a comprehensive view of the links in the chain that work together efficiently to create customer satisfaction at the end point of delivery to the consumer. Fig. 1 shows a classic supply chain. A supply chain can be divided into 3 main categories; namely upstream, manufacturer and downstream. In each phase, scientists focused on various types of objectives. Fig. 2 shows a graphical view of supply chain problem's classification.

* Corresponding author

E-mail address: delgoshaei.aidin@gmail.com (A. Delgoshaei)

© 2019 by the authors; licensee Growing Science, Canada

doi: 10.5267/j.uscm.2018.10.001

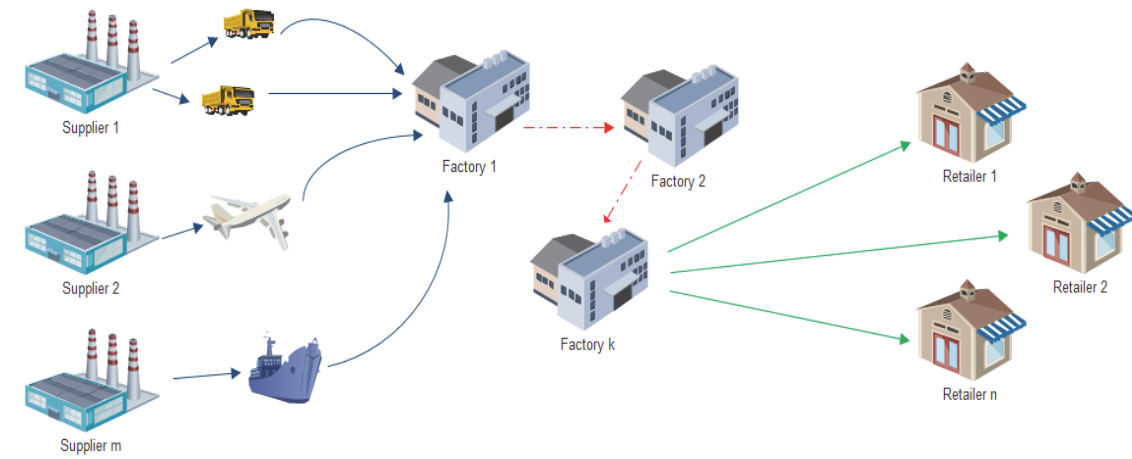


Fig. 1. A Classic Supply Chain

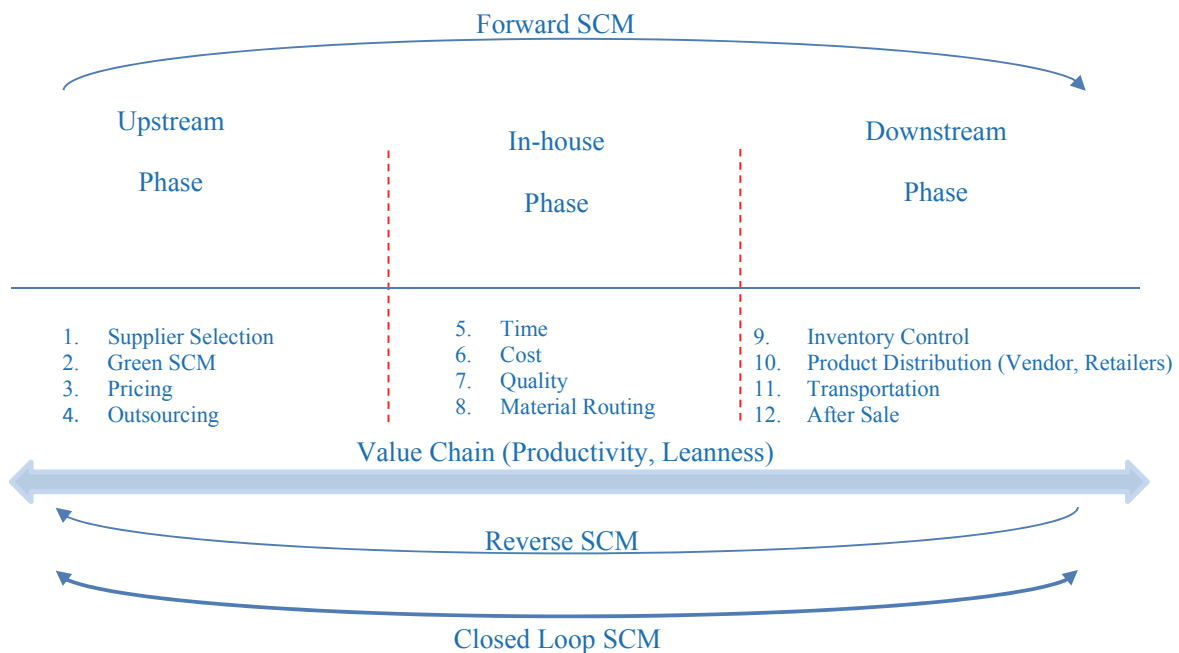


Fig. 2. Classification of Supply Chain Problems

During last 2 decades, supply chain management has been taken into consideration by scientists due to its high advantage. Brandenburg et al. (2014) reviewed mathematical models that are published in sustainable SCM. In continue, in each section, important drawbacks and problems in supply chain management studies will be discussed and solutions that are offered by scientists will be explained. In the classification that is shown in Fig. 1, part routing is an important objective of in-house processes. The aim of part routing is to find the best way of transferring materials through a chain of processes to minimize transferring cost of in-process materials inside a system, delivery time, or maximize productivity and leanness of a SCM. Delgoshaei et al. (2016a) compared different material transferring models that are developed by scientists in the CMS problem so far. Part routing is alternating the best sets of machines that use to perform the consecutive operations that are needed to complete product(s) while confronting with series of parallel machines to be selected. Choosing different sets of machines can yield various part routes with inter and intracellular material transferring costs. In continue a number of related problems will be reviewed which can help address the problem statement of this research (Arora et al., 2017).

2. Leanness and Productivity of SCM

Zhao et al. (2010) focused on dynamic changes and uncertainties such as machine breakdown, hot orders and other kinds of disturbances in holonic manufacturing systems. They argued that holonic systems require robust coordination and collaboration mechanisms to allocate available resources to achieve the production goals. In continue an integrated process planning and scheduling system is proposed to select suitable machining sequences of machining features and suitable operation sequences of machining equipment considering restricted capacities of machines. The proposed method is worked based on a fuzzy inference system which helped choosing alternative machines for integrated process planning. Vinodh and Balaji (2011) assessed the leanness level of a manufacturing organizations in lean manufacturing system. To solve the experiments they used a decision support system for fuzzy logic based leanness assessment. Miller and John (2010) proposed an interval type-2 fuzzy logic model of a multiple echelon supply chain which fallows better representation of the uncertainty and vagueness present in resource planning models. In continue, a Genetic Algorithm (GA) was employed for the proposed model to search for a near-optimal plan for the scenario.

Wong and Lai (2011) divided fuzzy techniques, that are used for scheduling and production operation management problems, into 5 categories. They found that most popular applications are capacity planning, scheduling, inventory control, and product design. Besides some application areas make more use of particular types of fuzzy techniques. Meanwhile the percentage of applications that address semi or unstructured types of POM problems is increasing. Moreover the most common technologies integrated with the fuzzy set theory technique are genetic-evolutionary algorithms and neural networks and finally the most popular development tool is C Language and its extension. Azadegan et al. (2011) proposed fuzzy linear programming to product mix prioritization problem. Their method was flexible enough to use in practice. Figueroa-García et al. (2012) used a mixed production planning problem in the presence of fuzzy demands which enables scientists to fuzzy sets in mathematical programming methods.

The product distributions is also considered among important keys to have a successful SCM (Janaki et al., 2018). In many cases products are delivered to retailers using different transporting systems and via various routes. Mula et al. (2010) presented a review of mathematical programming models for supply chain production and transport planning. Qin et al. (2011) proposed an alternative control system to describe a dynamic system with fuzzy white noise using a linear quadratic model. Jia and Bai (2011) proposed an approach for manufacturing strategy development based on quality function deployment. In continue, the authors also integrated fuzzy set theory and house of quality in order to provide a structural tool to capture the inherent imprecision and vagueness of decision-relevant inputs and to facilitate the analysis of decision-relevant quality function deployment (QFD) information. Delgoshaei et al. (2016b) proposed a new method for increasing the productivity of a manufacturing system by decreasing the work load variations.

Baykasoglu and Gocken (2010) proposed a hybrid fuzzy based ranking and Tabu search method to solve fuzzy multi-objective aggregate production planning problem. Liang et al. (2011) proposed a fuzzy mathematical programming method to solve aggregate production planning (APP) decision problems that involve multi-products and multi-periods in a fuzzy environment which aims to minimize total cost with respect to inventory carrying levels, available labor levels, machine capacity and warehouse space, and the constraint of available budget. Olugu and Wong (2012) proposed an expert fuzzy rule-based system for evaluating closed-loop supply chain management in terms of efficiency, effectiveness and economical strategies towards environmental sustainable practices in manufacturing companies. Delgoshaei et al. (2016c) proposed a new method for scheduling D-CMS while system costs are not fixed and can be varied from period to period. In many cases, the irrelevant location of machines has been observed to increase material transferring costs. Peidro et al. (2010) dealt with developing a fuzzy linear programming model for tactical supply chain planning in a multi-echelon,

multi-product, multi-level, multi-period uncertain supply chain network. The aim is to centralize multi-node decisions simultaneously to achieve the best use of the available resources along the time horizon so that customer demands are met at a minimum cost. El-Baz (2011) developed a decision making approach to deal with the performance measurement in supply chain systems using fuzzy set theory and the pair-wise comparison of analytical hierarchy process (AHP), which ensures the consistency of the designer's assignments of importance of one factor over another to find the weight of each of the manufacturing activity in the departmental organization. Venkata Rao and Patel (2010) proposed a new integrated AHP and the fuzzy logic based MCDM method named Preference Ranking Organization for Enrichment Evaluations (PROMETHEE) to select best manufacturing alternative option.

2. Dynamic and Uncertainty

In most real cases, part demands are different from one planning horizon to another. Such a criterion is known as dynamic part demand. Market changes, changes in product designs, and the manufacture of new products are some of the reasons for the change in part demands through different time periods. These conditions may cause emerging imbalances in part routings and bottleneck machines. They will be explained in a separate section because of their importance. Jeon and Leep (2006) presented a model for scheduling dynamic cells where machine failures can cause waiting times and reduce system capacity accordingly. Tavakkoli-Moghaddam et al. (2007a) considered dynamic part demands and parts mixed for a reconfigurable part routing problem; minimizing operating (constant and variable), machine relocating, and intercellular WIP transferring costs was considered as the objective of the proposed model. Tavakkoli-Moghaddam et al. (2007b) considered the normal distribution function to estimate the part demands in a stochastic model; minimizing material transferring movements was the main objective of the method. During the scheduling of a dynamic manufacturing system, the system capacity may be inadequate to meet customer demand at a specific period. Hence, Safaei and Tavakkoli-Moghaddam (2009) addressed a dynamic scheduling problem to find the tradeoff values between in-house production and outsourcing while cells are supposed to be reconfigurable. This time, they considered intercellular movements in addition to intracellular ones. The other solution to address part uncertainties is forming new cells as a result of market changes. Tavakkoli-Moghaddam et al. (2005) minimized material transferring costs in the dynamic condition of part demands by using alternative process plans and machine relocation and replications. Egilmez et al. (2012) focused on uncertain operation times in D-CMS. The contribution of their model is to consider risk level in process of designing cells in dynamic environment. A few years later, Egilmez and Sürer (2014) evaluated the impact of risk level in an integrated cell forming and scheduling problem using Monte Carlo Simulation. Sürer et al. (2010) proposed a new model which could determine the dedicated, shared and reminder cells in D-CMS. One important conclusion of their research is that in the average flow time and total WIP are not always the lowest when additional machines are used. Delgoshaei et al. (2016d) proposed a new method for scheduling dynamic CMS using a hybrid Ant Colony Optimization and Simulation Annealing Algorithms. Delgoshaei and Gomes (2016) used artificial neural networks for scheduling cellular layouts while preventive maintenance and periodic services are taken into consideration. Afterward, Renna and Ambrico (2015) also proposed three models for designing, reconfiguring and scheduling cells in dynamic condition of product demands. In their models, they considered minimizing system costs including intercellular movements, machining and reconfiguring costs as well as maximizing net-profit. While the dynamic costs through the time is taken into consideration, the present value of money must be considered. Inflation is defined as a sustained increase in the general level of prices for goods and services¹. The same resource provided many reasons why inflation rate must be considered but perhaps the most important is that “the entire economy must absorb repricing costs (“menu costs”) as price lists, labels, menus and more have to be updated”. An intensive review of literature reveals that problem scheduling supply chains to determine the ideal tradeoff between the values of in-house manufacturing and of outsourcing when the cost of manufacturing systems is not the same through planning periods and backorder of products between

¹. <http://www.investopedia.com/university/inflation/inflation1.asp>

planning periods is restricted, is less developed. Hence in this paper, a non-linear mixed integer programming method is developed which will be helpful to schedule supply chains in the presence of uncertain product demands and dynamic costs in short-term periods (one to three months).

3. Research methodology

In this section a new mathematical programming model will be presented in order to show the impacts of preventive maintenance and emergency services on part routing. In this regard Fig. 3 can show the logic of the proposing model:

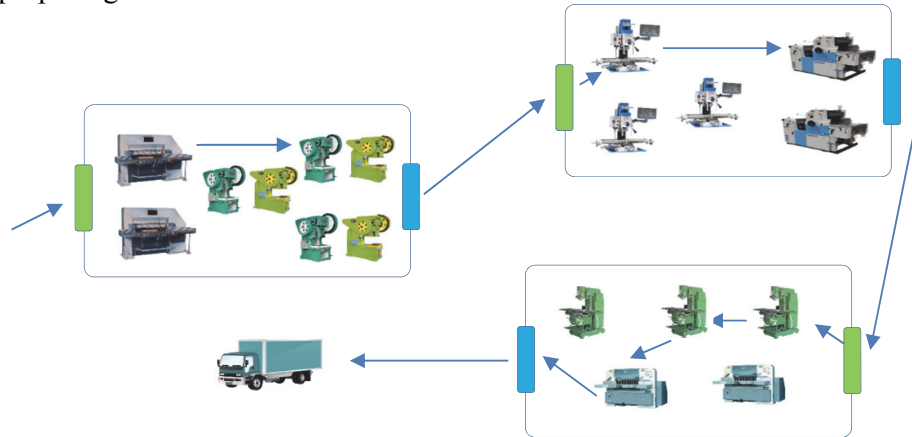


Fig. 3. Graphical View of the proposing model

The main contribution of this model can be listed as following:

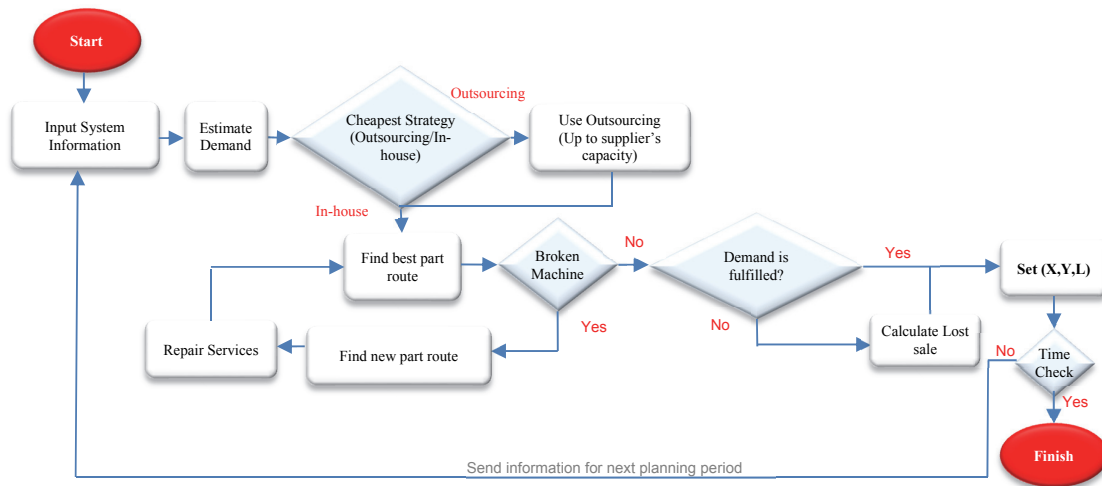
1. To find best amount of in-house manufacturing,
2. To use outsourcing as an effective strategy for SCM,
3. To find the best part routes for in-process materials,
4. To provide a mechanism for quick responding in the conditions of confronting with machine breakdown

To formulate the problem, some assumptions are considered which are:

1. Products have a number of operations that must be performed in a consecutive manner.
2. The product demands are uncertain and can be varied from period to period.
3. Machines breakdown may happen during production period. The failure rate will be expressed by normal function distribution.
4. Using outsource is allowed.
5. Machines must receive periodic services according to preventive maintenance plans.
6. Machine Purchasing is not allowed during the production horizon.
7. The performance of machines is not constant and will be affected by depreciation rate.
8. The machines' capacities must be considered while scheduling.
9. Backorders are not allowed. The beginning inventory is considered zero and last period backorder is not allowed.

In formulating the model, all system costs are accounted for, including group setup, operating, machine purchasing, outsourcing, and backorder costs. The available processing time of any workstation in a manufacturing period depends on the capacity of the machine inside the workstation. The goal is to survey production specifications under dynamic cost and to demonstrate how system characteristics can influence system performance.

To study the function of the proposing method, a flowchart is developed which shows the steps of determining appropriate product schedules by in-house manufacturing and using outsource services in a supply chain as shown by Fig. 4.



X: in-house Manufacturing; Y is using outsources and L shows lost sale

Fig. 4. Flowchart for the proposed method

The algorithm starts by importing production information in each of the period, then the amount of the product demand will be estimated using triangular probability function. In continue the preventive actions will be carried out for machines according to preventive maintenance service plans. Then the algorithm determines to use in-house strategy or outsourcing. This strategy will be taken by comparing manufacturing cost to outsourcing costs. In continue if the algorithm choose the in-house manufacturing, the production process, considering the capacity of machines, will carried out until all product demands are fulfilled. During the production process, the algorithm finds a new part route while a machine breakdown happens and then emergency repair activities will then carried out for the broken machine.

While the system and outsource capacities are not sufficient for fulfilling the product demands, the remained demands are considered as lost sale. These steps will be carried out again in next periods until all products are scheduled.

4. Mathematical model

In order to develop a model with mentioned features, the proposed NL-MIP model in the previous section is developed more by adding preventive activities and emergency maintenance services during optimizing process. The aim is to survey how trading off values between in-house manufacturing, outsourcing and backorders will be affected by machine failures and preventive maintenance. Again the problem circumstance is considered dynamic where product demands and all system costs are not deterministic and may be varied from period to period. The other important goal is evaluating how part-routings process is sensitive to condition of machine unreliability and how in-process materials change their production routes due to machines' broken.

As a brief, the advantages of the proposed model can be summarizes as: considering uncertain costs in sub periods, dynamic product demands, machine failures, preventive activities, machine depreciation rate, internal and external material movements inside and between workshops with the varied batch size, existence of parallel machines, alternative process routes for part types considering operation sequence. To find the best set of in-house manufacturing (using the system capacity) and outsource services, the following assumptions are taken into consideration:

Inputs:

i: number of parts
j: type of machines
k: number of workshops
l: number of sub – contractors
t: planning periods (time slots)

Parameters:

$d_{i,t}$: demand of product *i* in period *t*

$$d_{i,t} \sim N(\mu_{i,t}, \sigma_{i,t}) \quad (1)$$

S_j : Setup cost of machine *j*

$OP_{i,j}$: cost of processing part *i* using machine *j*

OS_l : cost of performing one product by subcontractor *l*

PM_j : preventive maintenance cost of machine *j*

EM_j : emergency repair cost of machine *j*

α_i : Material movements cost inside a workshop for part *i*

β_i : Material movements cost between workshops for part *i*

length: is a length of the work shops

width: is a width of the work shops

Input Matrixes:

Product Demand ($D_{i,t} [\cdot]_{i,t}$)

Batch Size ($BS_i [\cdot]_i$)

Machine Component Incidence Matrix ($MCIM_{i,j} [\cdot]_{i,j}$)

Machine Capacity ($MA_j [\cdot]_j$)

Sub-contractor Capability ($SC_l [\cdot]_l$)

Initial Number of Machines ($NOM_j [\cdot]_j$)

Operation Cost ($OP_i [\cdot]_i$)

Setup Cost ($S_j [\cdot]_j$)

Internal Movement Cost ($\alpha_i [\cdot]_i$)

External Movement Cost ($\beta_i [\cdot]_i$)

Depreciation Rate ($DR_j [\cdot]_j$)

Preventive List ($PL_{j,t} [\cdot]_{j,t}$)

Preventive Service Time ($PST_j [\cdot]_j$)

Emergency Maintenance Time ($EMT_j [\cdot]_j$)

Failure Rate ($FR_j [\cdot]_j$)

Preventive Maintenance Cost ($PMC_j [\cdot]_j$)

Emergency Services Cost ($ESC_j [\cdot]_j$)

Outsourcing Cost ($OS_l [\cdot]_l$)

Variables:

$X_{i,f,j,k,t}$: number of part *i* which is performing by *f*th of machine type *j* in workshop *k* in period *t* (int.)

$Z_{i,f,j,k,t}$ if part *i* performs using *f*th of machine type *j* in workshop *k* in period *t* (bin.)

$Y_{l,t}$: number of products which manufactured by sub – contractor *l* in period *t* (int.)²

$L_{i,t}$: number of products *i* which are considered as lost sale (int.)

$W_{f,j,k,t}$ = if *f*th machine type *j*, which is located in workshop *k*, is broken in period *t* (bin.)

². Noted that in order to track the model easier it is supposed that part 1 can be performed by subcontractor type 1 and so on.

Mathematical Model:

$$\text{Min: } \sum_t \sum_k \sum_j \sum_f \sum_i S_j \cdot Z_{ifjkt} \cdot (X_{ifjkt} / BS_i) + \sum_t \sum_k \sum_j \sum_f \sum_i MCIM_{i,j} \cdot OP_{i,j} \cdot X_{ifjkt} \cdot Z_{ifjkt} \quad (2)$$

$$+ \sum_t \sum_k \sum_j PM_j \cdot PL_{j,t} + \sum_t \sum_k \sum_j \sum_f (EM_j + S_j) \cdot W_{f,j,k,t} \quad (3)$$

$$+ \sum_t \sum_l OS_l \cdot Y_{l,t} \quad (4)$$

$$+ \sum_t \sum_k \sum_i \sum_j \sum_f \left[d_{i,t} - \left(X_{i,j,f,k,t} / BS_i \right) \right] \cdot BC_i \quad (5)$$

$$+ \sum_{t=1}^{TH} \sum_{k=1}^K \sum_{l=1}^L \sum_{m=1}^{m-1} \sum_{j=1}^J \sum_{f=1}^F \left(\alpha_t \cdot Z_{m,i,f,j,k,t} \cdot \left(\left(\sum_{j=1}^J Z_{m+1,i,f,j,k,t} \cdot X_{m+1,i,f,j,k,t} \right) - X_{m+1,i,f,j,k,t} \right) \right) \quad (6)$$

$$+ \sum_{t=1}^{TH} \sum_{k=1}^K \sum_{l=1}^L \sum_{m=1}^{m-1} \sum_{j=1}^J \sum_{f=1}^F \left(\beta_t \cdot Z_{m,i,f,j,k,t} \cdot \left(\left(\sum_{k=1}^K \sum_{j=1}^M Z_{m+1,i,f,j,k,t} \cdot X_{m+1,i,f,j,k,t} \right) - \sum_{j=1}^M X_{m+1,i,f,j,k,t} \right) \right) \quad (7)$$

s.t:

$$\sum_k \sum_j \sum_f \sum_i \left[\left(X_{i,f,j,k,t} / M \right) \right] + \sum_l Y_l + \sum_t L_{i,t} \geq d_{i,t} \quad \forall t, i; \quad (8)$$

$$X_{i,f,j,k,t} \cdot (1 - MCIM_{i,j}) \leq 0 \quad \forall t, k, j, i; \quad (9)$$

$$X_{i,f,j,k,t} \leq MA_j \cdot DR_j - PST_j; \quad \forall t, k, j, i; \quad (10)$$

$$X_{i,f,j,k,t} \cdot (1 - W_{f,j,k,t}) = 0 \quad \forall i, f, j, k, t; \quad (11)$$

$$\sum_t Y_l \leq SC_l \quad \forall l, t; \quad (12)$$

$$\sum_k \sum_i \sum_j X_{i,f,j,k,t} \leq \sum_k MA_j \quad \forall t, j; \quad (13)$$

$$X_{i,f,j,k,t}; Y_{l,t}; L_{i,t} : \text{integer} \quad (14)$$

$$Z_{i,f,j,k,t}; W_{f,j,k,t} : \text{binary} \quad (15)$$

The first sentence of objective function represents the setup cost for each machine that may be different from period to period. The next sentence shows the operating cost including machinery of each part. The third and fourth sentences are developed to calculate preventive maintenance and emergency activity services costs, respectively. As seen, after each emergency service, the machine is needed to be setup again. Then, the fifth sentence shows the outsourcing cost. The eighth and ninth sentences show internal and external material transferring costs respectively. The first set of constraints guarantees that demands for each part will be satisfied by in-house manufacturing, using outsource services or lost sale. The second constraint is developed for ensuring that operations on parts will be performed based on MCIM information. The third constraint ensures that each machine will be allocated based on its capacity. It should be mentioned that in this model, machine capacities are affected by depreciation rate in each time slot. The next constraint is developed to show that while a machine is broken it cannot perform any operation. The fifth constraint is to guarantee that using subcontractor services will not be more than their announced capacity. The sixth set of operations shows that the amount of producing each part should not be more than the available capacity of the related machine. The next 2 set of constraints are used to control domain of the variables. This research is in continue of some other researches that found in the literature. Table 1 compares the features and novelties of this research to similar researches in the literature:

Table 1
Novelties of this research comparing to the literature

Reference	Idea	Objectives	Model Type	Product Demand	Planning Period	Solving Algorithm	Novelties
This Research	Part Routing	Minimizing system cost	NL-MIP	Stochastic	Multi-Period	Hybrid ACO-SA	Preventive Plans/Emergency Services
Peidro et al. (2010)	Production Planning	Minimizing system cost	FLP	Stochastic	Multi-Period	Fuzzy Programming	Utilizing Resource Usage
Delgoshaei et al. (2016a)	Production Planning	Minimizing system cost	MIP	Stochastic	Multi-Period	Genetic Algorithm	Minimizing work-load Variation
Liang et al. (2011)		Minimizing system cost	APP	-	Multi-Period	Fuzzy Programming	Inventory/ Labor Levels/ Machine Capacity/ Warehouse Space/ Available Budget

5. A hybrid Ant Colony Optimization and Simulated Annealing

Ant Colony Optimization is inspired from swarm intelligence of real ants that live in big colonies (hundred thousand of ants). The main aim of classic version of ACO, which is designed to solve discrete optimization problems, is to find smaller path in a graph, but other versions were promoted to solve continuous problems with various objectives. Fig. 5 shows how ACO finds the better path with higher pheromone which directs ants to optimum solution area. Since Simulated annealing (SA) has strong mechanism to escape from local optimum points, in this research, the ant colony optimization will be promoted by using this feature.

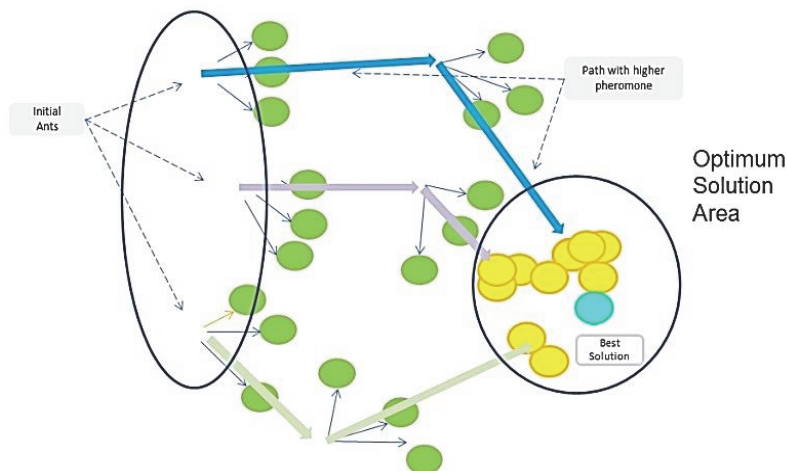


Fig. 5. Solving process scheme of Ant Colony Optimization algorithm

The algorithm starts by importing the period data (such as product demand, manufacturing sequences, preventive plans, setup costs, operating cost, emergency service costs, internal and external material transferring costs) (Fig. 6). Then the algorithm will find the best series of machines that can serve operations process. This part route is alternated according to remained capacity of each machine, distance between machines and material transferring costs. This process will be repeated until all product demands are scheduled. Whenever needed, outsource services will be used. While all products are scheduled, the system will calculate objective function as a threshold value to accept or reject the solution. The amount of the objective function is then used for calculating the level of pheromone. This pheromone will provides the chance of using similar sets of machines in coming solutions. To escape local optimum rates the algorithm will use local escaping operator by giving small chance to accepting worse solution.

Table 2 summarized the steps of the proposed hybrid ACO-SA.

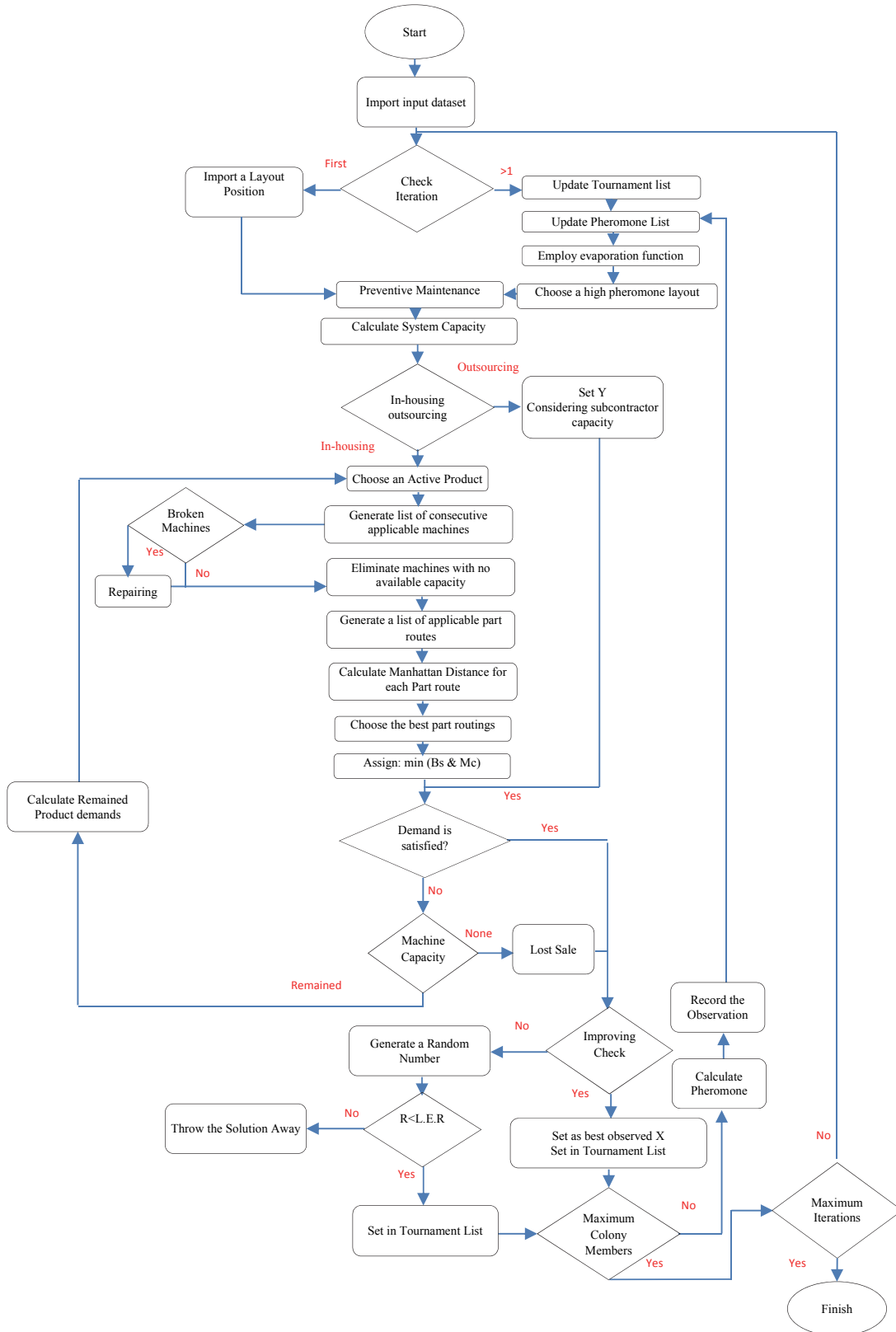


Fig. 6. Mechanism of the proposed hybrid ACO-SA for solving the developed model

Table 2
Pseudo code for the proposed ACO-SA algorithm

Insert Dataset
<ul style="list-style-type: none"> Input data sets ($D_{i,t}$; BS_i; $MCIM_{i,j}$; MA_j; SC_i; $AB_{i,t}$; NOM_j; OP_i; S_j; α_i; β_i; OC_j; SR_n; $TB_{j,t}$; CC_n; H_j; F_j; CS_j; W_i; K_j; DR_j; $PL_{j,t}$; PST_j; EMT_j; FR_j; PMC_j; ESC_j; BC_i; OS_i; lr_i) Initialize CSz, k, itr, τ_0, EVR & LEP (CSz: colony size; k= neighbourhood radius; itr: number of iterations; τ_0: initial level of pheromone, EVR: Evaporating rate; LEP: local escape probability)
While n<itr
for $m \in cs$
Step 1)
<ul style="list-style-type: none"> If $\tau_m - \max_{n \in cs}^{itr}(\tau_n) \geq EVR$ <ul style="list-style-type: none"> $\tau_m = \tau_m - EVR$
Step 2)
<ul style="list-style-type: none"> if n=1 Generate a layout else Call Selecting operator (tournament list)
Step 3)
<ul style="list-style-type: none"> for $i \in I$ if $OS_i < OP_i$ $O_i = \min(D_{i,t}; \text{Out source capacity } l)$ calculate Remained $D_{i,t}$
Step 4)
<ul style="list-style-type: none"> find $MCIM_{i,j}$ Generate Neighbor list considering neighbor radius (n^m) <ul style="list-style-type: none"> for $k \in n$ <ul style="list-style-type: none"> Part. route$_k = \sum internal\ travel + external\ travel$ (for consecutive machines) Find the Best Part routes in the X $x_i = \min(MA_j, \text{Remained } D_{i,t}, BS_i)$ calculate Remained $D_{i,t}$
Step 5)
<ul style="list-style-type: none"> for $i \in I$ if Remained $D_{i,t} > 0$ <ul style="list-style-type: none"> if Machine Capacity=0 Set $L_{i,t}$ Go step 3
Step 6)
<ul style="list-style-type: none"> else Set $b_i = \text{Remained } D_{i,t}$
Step 7)
<ul style="list-style-type: none"> Calculate OFV_m if $OFV_m \leq \min_{itr}(OFV)$ Tournament list$_i = X_m$ $OFV^{best} = OFV_m$ $X^{best} = X_k^{itr}$ call Observation.operator $\tau_m = 1 - \left(\frac{\text{Tournament}_{obj(m)}}{\sum_{itr}(\text{Tournament}_{obj})} \right) \left(\frac{\text{Observation}(m)}{\sum_{itr}(\text{Observation})} \right)$
Step 8)
<ul style="list-style-type: none"> else R=rand(1) if $R \leq LEP$ Tournament list$_i = X_m$ call Observation.operator $\tau_m = 1 - \left(\frac{\text{Tournament}_{obj(m)}}{\sum_{itr}(\text{Tournament}_{obj})} \right) \left(\frac{\text{Observation}(m)}{\sum_{itr}(\text{Observation})} \right)$
Step 9)
<ul style="list-style-type: none"> Check Stopping Criteria
End

There are some parameters which must be set before using the ACO-SA algorithm which are neighborhood radius, colony size, number of iterations, local escaping parameter, initial pheromone and evaporating rate.

5.1 Neighborhood radius

While searching the solution space, the proposed algorithm will search the parallel machines close to the last machine which served the operation process. Such approach will guarantee reducing material transferring costs. The size of neighborhood searching plays key role in both speed of the algorithm and quality of the solutions. Thus the neighborhood size for considering part routes will be k^m where k shows the number of neighborhood radius which can be determined by decision maker and m is number of applicable machines according to MCIM. For example if neighborhood radius is set 3 by a decision maker and for completing a part (let's say i) required 4 machines, then algorithm will generate 81 neighbors (part routes) in candidate list.

$$k^m \rightarrow 3^4 = 81 \quad (16)$$

Fig. 7 shows the approach of searching the neighborhood radius to find parallel machines which are candidate to serve the next operation.

The figure consists of two side-by-side screenshots from a MATLAB environment. The left screenshot shows MATLAB code for creating a candidate list of feasible part routes. The right screenshot shows the results of creating a part route for an example with 2 machines and 3 neighborhood size.

```

465 -     eee;
466 -     b=nnz(eee)/neighbour_radius;
467 -     eeee=zeros(neighbour_radius*b,b);
468 -     candidate_list=zeros(neighbour_radius*b,1);
469 -     for bb=1:1:neighbour_radius*b
470 -         for b=1:1:(nnz(eee)/neighbour_radius)
471 -             cc=ceil(rand(1)*neighbour_radius);
472 -             eeee(bb,b)=eee(b,cc);
473 -         end
474 -     end
475 -     eeee;
476 -     %for each route, find the intercellular and intercellular
477 -     %geometrical path of consecutive machines which shows the
478 -     %material transferring costs that must be spend for that

```

The right screenshot shows the results of creating a part route for an example with 2 machines and 3 neighborhood size. The results are displayed as follows:

```

eee =
    69    51    69
    76    76    21

eeee =
    69    21
    69    76
    51    21
    51    76
    69    21
    51    76

```

Fig. 7. Scripts for creating candidate list of feasible part routes in ACO-SA (left image)/results of creating a part route for an example with 2 machines and 3 neighborhood size (right image)

As shown in Fig. 7, the algorithm calculates the Manhattan Distance formula for sets of machines that can serve the next operations. Then the algorithm chooses the best machine for the candidate list.

5.2. Colony size

One dominant feature of the proposing algorithm is to start search approach with more than one point. For this purpose number of colony members must be specified by decision maker as an input value. It is obvious that choosing large number of colony members will increase the chance of searching the solution space but at the same time it increases the searching time, significantly. Therefore, the appropriate value for colony size will be selected by Taguchi method in section 4.

5.3. Pheromone sprinkle

While the layout of a new ant is generated, the operators are allocated and the product demands are assigned; the total cost of the system will be calculated using the objective function. Then, the amount of pheromone for the ant will be calculated by selecting operator in future as shown by Fig. 8:

$$\tau_u = \begin{cases} 1 - \left(\frac{Tournament_{obj}(u)}{\sum_u(Tournament_{obj})} \right) \left(\frac{Observation(u)}{\sum_u(Observation)} \right); & \text{if } OFV_u \leq \sum_i(OFV_{c.m_i,itr-1}) \text{ or } Y \leq LEP \\ 0; & \text{Otherwise} \end{cases} \quad (17)$$

where τ_u is the pheromone of u^{th} ant in an iteration, $Tournament_{obj}(u)$ is the objective function of the u^{th} colony member, $Observation(u)$ is the number of times that the similar layout is observed, Y is a random number between (0,1) and LEP is the local escape parameter that is set by decision maker. Eq. (25) uses the value of objective function (total system costs) for calculating the amount of pheromone sprinkle of a colony member. In addition, the number of observant of a layout is considered as part of the formula to emphasis on the elite layouts. As a result, those layouts that provide better system costs and are observed more than the other layouts in tournament lists, have the more chance to be chosen by the selecting operator. Therefore, the more objective function value saving occurs in a path, the more pheromone will be sprinkled.

5.3. Pheromone update

While processing on an ant completed, the pheromone list will be updated to find the summation of pheromones in the related iteration and to update the observation list as shown by Fig. 8. The pheromone list, observation list and tournament lists are the inputs of next round of the algorithm.

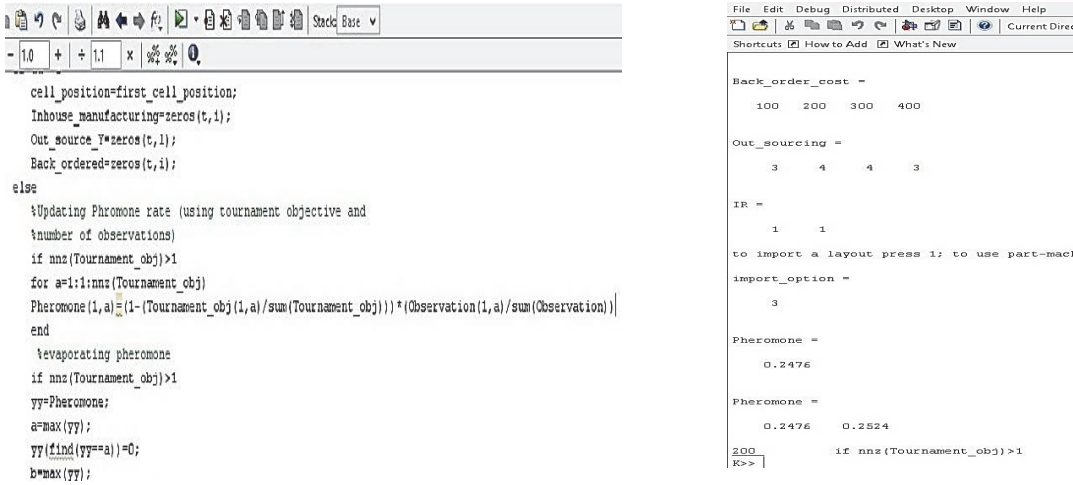


Fig. 8. Scripts of the proposed ACO for calculating pheromone sprinkle (left image)/results of calculating pheromone in a ACO with 2 colony members (right image)

5.4 Pheromone evaporation

During the optimizing process, some solutions may be too strong and cause other solutions to have small chance to be chosen in next iterations. Such condition will lead early stage convergence. To avoid it, scientists normally use a mechanism to control the level of pheromones. Fig. 9 shows evaporating pheromone process in the proposed ACO-SA which will be carried out using an evaporation rate in a way that if the pheromone of a specific layout increases too much, the algorithm will reduce it gradually. Suppose $X_k^m(x_1, x_2, \dots, x_i; o_1, o_2, \dots, o_i; b_1, b_2, \dots, b_i)$ is the k^{th} ant in m^{th} iteration.

$$\text{If } \tau_k - \max_{n \in m}(\tau_n) \geq EVR \tag{18}$$

$$\text{Then } \tau_k = \tau_k - EVR \tag{19}$$

where τ_k is pheromone of the k^{th} ant in m^{th} iteration, $\max_{n \in m}(\tau_n)$ is the maximum pheromone that observed in m^{th} iteration so far and EVR is evaporation rate.

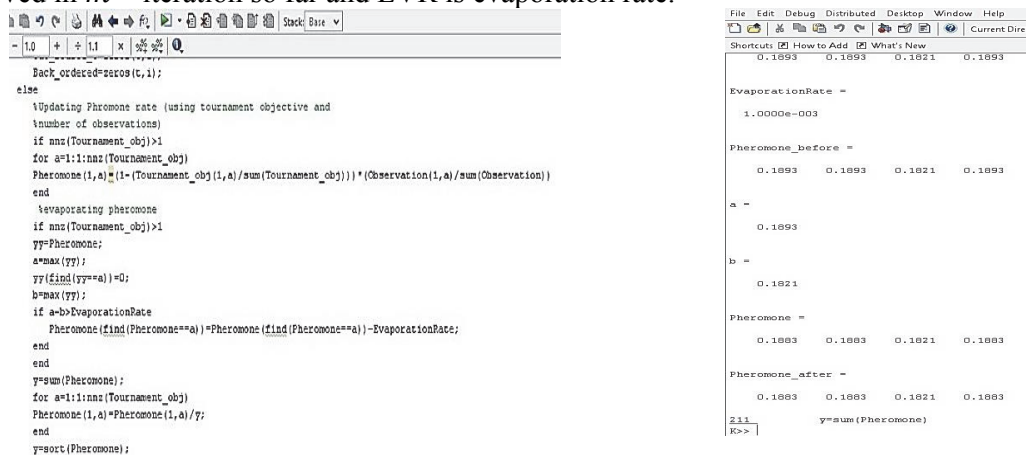


Fig. 9. Scripts of the ACO for evaporating the extreme pheromones (left image)/ results of decreasing the extremely high pheromones in a sample

During each iteration, the ACO uses a tournament list where each member inside is a worthy layout chosen before. Those members of tournament list who have better objective function will have greater pheromone and consequently have more chance to be selected for new ants. The selecting operator normalizes the pheromones in pheromone list. Then, it sorts the pheromones in a cumulative ascending order. Hence, solutions with higher pheromones have more chance to be chosen as shown by Fig. 9.

a) Suppose $X_k^m(x_1, x_2, \dots, x_i; o_1, o_2, \dots, o_i; b_1, b_2, \dots, b_i)$ is the k^{th} ant in m^{th} iteration.
 b) Normalize $\tau_n \in m^{th}$ iteration (20)

c) $Y: \text{Sort}^m(\tau_n)$ in ascending order (21)

d) $YY: \text{Rand}$ a number between (0,1) (22)

e) Find the nearest number to YY in the vector Y

5.5 Fitness function operator

Fitness function value is a measuring index to evaluate the quality of the solutions. For this purpose the objective functions that are explained in section 2.1 is used.

$$\begin{aligned} \sum \text{Group Setup Cost} &+ \sum \text{Operating Cost} + \\ \sum \text{Outsourcing Cost} &+ \sum \text{Preventive Maintenance Cost} + \\ \sum \text{Repair Services Cost} &+ \sum \text{Warehousing Cost} + \\ \sum \text{Internal Material Transferring Cost} &+ \sum \text{External Material Transferring Costs} \end{aligned} \quad (23)$$

While a solution is generated the algorithm will calculate the fitness function for it. If the amount of fitness function is less or equal to what obtained from other ants so far then the algorithm accept the ant as a member of tournament list.

a) Suppose $X_k^{\text{itr}}(x_1, x_2, \dots, x_i; o_1, o_2, \dots, o_i; b_1, b_2, \dots, b_i)$ is the k^{th} solution in itr^{th} iteration.
 b) If $F(X_k^{\text{itr}}) \leq \min(F(X_1^{\text{itr}}), F(X_2^{\text{itr}}), \dots, F(X_{k-1}^{\text{itr}}); F(X_{\text{best}}^{\text{itr}-1})) ; \forall k \in i$ (24)

c) Then, $\text{Tournament.list}_{m+1} = X_k^{\text{itr}}$ (suppose the tournament list already has m members) (25)

5.6 Local optimum escaping operator

Since the model is complicated, the chance of falling into local optimum raises.

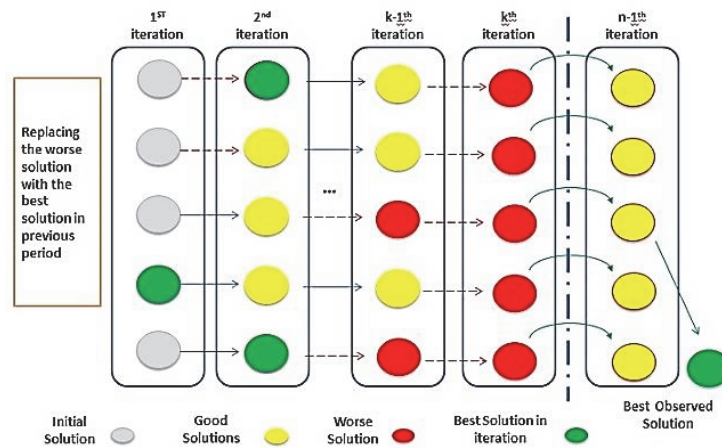


Fig. 10. Solving process scheme of ACO-SA with the ability of escaping from local optimum traps

In such situations, the algorithm must be equipped with powerful mechanism to escape from local optimum tales, otherwise the searching process will soon be interrupted by falling into such tales. The proposed algorithm uses local escaping mechanism of SA as an effective way to escape from local

optimum traps (See Fig. 10). After calculating the fitness function for the developed solution in iterations (say X_k^{itr}), the ACO-SA algorithm checks them with the best observed value achieved so far; $F(X_{best}^{itr-1})$. If the fitness function value, $F(X_k^{itr})$, is less than the $F(X_{best}^{itr-1})$, ACO-SA replaces the X_k^{itr} with $X_{best,itr-1}$. Such approach allows the algorithm to search other directions, by small chance, to find better solutions in future (Fig. 11).

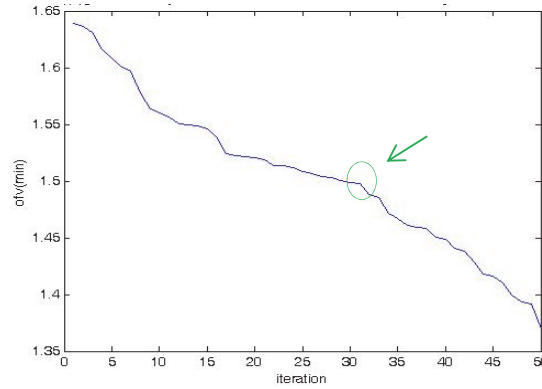


Fig. 11. Sample of escaping from local optimum traps by using local optimum escaping operator

Such local escaping operator is added to the algorithm by using a function which described:

$$F(X_{best}^{itr}) = \begin{cases} F(X_k^{itr}); & \text{if } F(X_k^{itr}) \leq \min \{F(X_i^{itr}), F(X_{best}^{itr-1})\} \forall i \in itr \\ F(X_k^{itr}); & \text{if } R \leq LEP \\ F(X_{best}^{itr-1}); & \text{otherwise} \end{cases} \quad (26)$$

where R is a normal random number between (0,1) and LEP is a local escaping parameter which is defined by decision maker. Note that the exact amount of LEP cannot be determined and may be different from case to case but it can be approximately estimated using DOE.

5.7. Number of iterations

Number of iterations shows the number of time the algorithm repeats the searching process. The large number of iterations increases the accuracy of the solving algorithm but at the same time it increases the time of computations. Therefore it must be chosen in a logic manner which will be estimated in DOE section.

5.8. Stopping criteria

The stopping criteria in the ACO-SA algorithm are set as:

- 1) Reaching to maximum number of pre-defined iterations,
- 2) If there is no choice in a tournament list of iteration which means none of the solutions in the iteration can improve the fitness function so there would be no choice for improving the algorithm.

a) Suppose $X_k^{itr} (p_1, p_2, \dots, p_n; s_1, s_2, \dots, s_n; x_1, x_2, \dots, x_n; o_1, o_2, \dots, o_n)$ is the k^{th} solution in itr^{th} iteration.

$$b) \text{ If } F(X_k^{itr}) > \min (F(X_1^{itr}), F(X_2^{itr}), \dots, F(X_{k-1}^{itr}); F(X_{best}^{itr-1})) ; \forall k \in i \quad (27)$$

$$\text{Then Tournament. list}^{itr} = \emptyset \text{ for next iteration} \quad (28)$$

6. Design of Experiments

The experimental design helps estimate the significance of each setting parameter and also determine the interactions between these parameters. In this research, Taguchi method designs are used using Minitab® 17. For DOE experiments, the levels of each parameter must be estimated. In order to estimate levels of an input parameter a range of data for a parameter can be used and then appropriate value(s) can be estimated considering other condition of the problem like, the scale of the problem and complexity of the problem. The best estimated values for each parameter of metaheuristics are shown by the Table 3:

Table 3
Initial estimation for levels of factors

Algorithm	Factor	Small scale			Medium scale			Large scale		
		L1	L2	L3	L1	L2	L3	L1	L2	L3
BnB	Number of iteration	30	50	100	50	100	150	50	100	200
	Number of Solutions	20	50	100	30	50	80	50	80	100
	Local Escaping Rate	0	0.1	0.3	0	0.1	0.3	0	0.1	0.3
Hybrid ACO and SA	Number of Iterations	30	50	100	50	100	150	50	100	200
	Colony Size	20	50	100	30	50	80	50	80	100
	Evaporation rate	0.001	0.002	0.005	0.001	0.002	0.005	0.001	0.002	0.005
	Local Escaping rate	0	0.1	0.3	0	0.1	0.3	0	0.1	0.3

The levels of factors, which are shown by Table 3, are then used for DOE in order to find the best estimating value for each parameter, significant parameters and interactions between them. Upon implementing the experiments designed for the Taguchi method, the obtained results show that the algorithm is sensitive to the number of iterations, colony size, and the local escaping rate (Fig. 12).

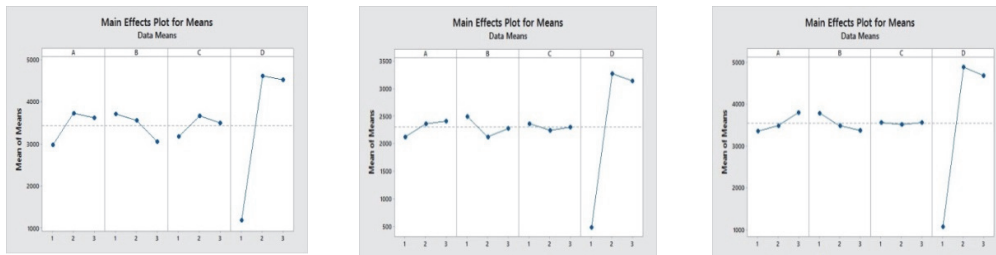


Fig. 12. The main effects plot of input parameters for small, medium and large scale problems while using the proposed hybrid ACO-SA (left to right)

The high slope of the curves related to the local escaping rate shows that while using ACO-SA for small, medium and large scale problems, L₂ provides better results. With the same logic, number of iterations should not have more than 2 members for small scale problems. For medium and large scale problems, L₂ must be considered for number of the iterations. The numbers of iterations and colony size have moderate impact on minimizing the objective function. Fig. 12 also shows that the local escaping rate has the highest impact on maximizing the algorithm power (minimizing total system costs). Therefore, the appropriate ranges for number of iterations and number of solutions in iterations can be set as what shown by Table 4:

Table 4
Estimated values for input parameters of ACO-SA algorithm

Algorithm	Factor	SMALL SCALE	MEDIUM SCALE	LARGE SCALE
BnB	Number of iteration	100	150	200
	Number of Solutions	50	50	100
	Local Escaping Rate	0.1	0.1	0.3
Hybrid ACO and SA	Number of Iterations	50	150	200
	Colony Size	20	30	50
	Evaporation rate	0.002	0.001	0.001
	Local Escaping rate	0.1	0.1	0.1

7. Results and Discussion

7.1. A small size example

To determine and verify the performance of the solving algorithm in scheduling CMS-SCM in the presence of product demands uncertainty, a small scale experiment is described in details. Then, results will be analyzed and discussed thoroughly. In this example, it is assumed that four products must be processed in a manufacturing system containing three different types of workers and two workstations. Two planning periods are also considered. Rest of the related information is presented in Table 5.

Table 5
Dataset for small scale experiment

		Product P1	P2	P3	P4
Product dataset	Demand $N\sim(\mu,\sigma^2)$	$N\sim(100,12)$	$N\sim(60,8)$	$N\sim(75,14)$	$N\sim(200,4)$
	Supplier	L1	L2	L3	L4
	Outsource Capacity	100	50	60	120
	Batch size	15	10	8	15
	Allowed Backorders	[20,0]	[30,0]	[45,0]	[60,0]
	Outsourcing cost (per product)	50	14	20	18
	Backorder cost (per product)	4	3.5	3	2.4
	Internal transferring cost	3	5	4	7
	External transferring cost	4	8	6	8
	Operator dataset	Operator	W1	W2	W3
Skill Level		[L1, L2, L3]	[L1, L2]	[L1, L2, L3]	
Course Cost		[50 35 40]	[35 40]	[25 30 50]	
Increasing Capacity		[0.04 0.1 0.015]	[0.03 0.07]	[0.03 0.1 0.12]	
Machine Data set	Machine	M1	M2	M3	
	Initial Number	5	7	4	
	Capacity	100	150	75	
	Setup Cost	100	140	200	
	Machine Purchasing cost	8000	12000	7500	

Table 6 depicts the operation process for completing each product. For example, product 1 needs to visit machine types 1, 2 and 3 respectively.

Table 6
MCIM matrix for small scale experiment

MCIM	M1	M2	M3
P1	1	1	1
P2	0	0	1
P3	0	1	1
P4	1	0	0

To be more comprehensive, all results are presented using the result view of Matlab. The result of best workstations is displayed in Fig. 13.

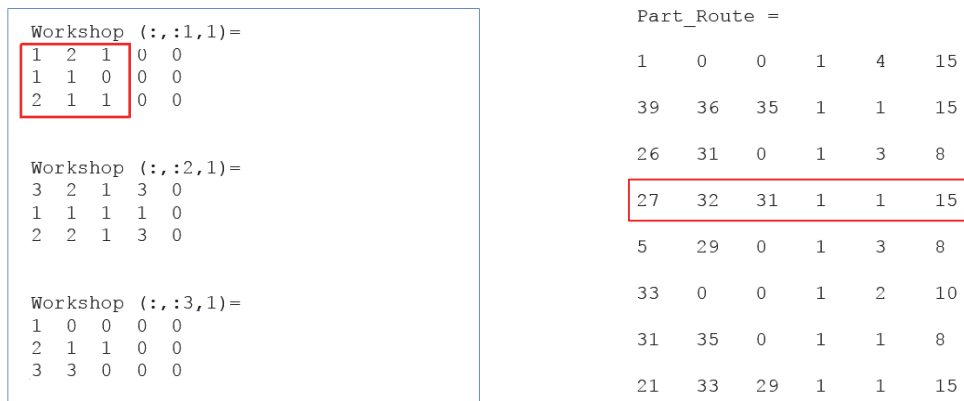


Fig. 13. Outputs of the ACO Algorithm for the Solved Experiment

Fig. 13 indicates that the ACO-SA generated three block diagonals through agglomeration. These blocks are maintained until the end of the searching process. Such agglomerative blocks are generated in this example primarily because the similarities in using machine services. Hence the algorithm considers only one center point and attempts to locate other machine groups on the basis of their similarity scores around the center point. After forming workshops and locating the machines and operators inside, the next step is to identify the best parts routes for allocating the operation sequence of products (part allocation). For this purpose, a new matrix is developed that determines the ideal number of part allocations for production (Fig. 13). The fourth row in Fig. 13 shows that machines that are located in positions 27, 32, and 31 in the right image employed to output 15 units of product 1 in the first period of planning. Generating one unit of product type 1 requires machine types 1, 2, and 3, consecutively. The algorithm is developed such that it can search the nearest set of the required machines in the layout to minimize material transfer costs. Therefore, machines that are located in positions 27, 32, and 31 are chosen to complete product 1 at the lowest internal WIP transfer costs (Fig. 13). Scheduling 15 units of product type 1 is based on calculating of the minimum value in the remained product demands, the capacity of the selected machines, skill of the operators that are working on machines and the batch size for product 1 (15).

$$\text{work load} = \min(\text{remained product demands}; \text{capacity of the selected machines}; \text{skill of the operators}; \text{batch size}) \quad (29)$$

After scheduling products we need to survey the number of workstation loads because this number represents work-load variations. This calculation can assist decision makers in determining the efficiency of the generated layout and parts routes in practice. The “*Work-loads*” matrix in Fig. 14 indicates the number of workstation loads in each workstations. Workstations with less or more load than other parallel workstations can thus be recognized. For example, the machine located in the second column of the first row in the first workstation of Fig. 14 is allocated only eight times, whereas similar machine types in the same workstation are allocated more often (the position of similar machine types can be identified using Fig. 13). Hence this event can be considered is an example of workstation load variation that may lead to work load variation as well.

Work_loads (:, :, 1, 1)=	Finished_Products =
35 8 15 0 0	94 92 24 145
35 35 0 0	117 85 8 110
15 20 15 0 0	Outsourcing =
Work_loads (:, :, 2, 1)=	100 28 60 0
24 20 15 15 0	3 5 52 0
24 15 10 10 0	Lost_sale =
15 20 15 20 0	6 0 2 0
Work_loads (:, :, 3, 1)=	0 0 0 0
20 0 0 0 0	
24 15 10 0 0	
15 20 0 0 0	

Fig. 14. The best observed workstation loads (left image) and Results of in-house manufacturing, outsourcing and backorders (right image) achieved in final solution

The total number of in-house manufacturing processes in each period is shown by the “In-house manufacturing” matrix depicted in Fig. 14. For example, the amount of product 1 that must be manufactured in the first period is 94 units in this matrix. The capacity of machines in workstations is inadequate; therefore, the algorithm set some units to be completed through subcontractor services. The “best-observed-out-source” matrix indicates the number of units that are scheduled for outsourcing. For example, 28 units of product 2 are scheduled to be completed through subcontractor services in the first period. As mentioned previously, the remained product demands that cannot be completed through in-house manufacturing and outsourcing services are regarded as backorders. In this experiment, six units of product 1 and two units of product 3 are postponed to the second period.

The final layout can be analyzed further. Table 7 presents the analysis of the solved experiment. The numbers of internal material movements are 3 and 5 in first and second planning periods, respectively; these numbers represent the number of bottleneck products. The amount of backorders in the second period is zero, which is in accordance with the problem assumptions of the proposed model.

Table 7
Results of analyzing best observed layout

Metrics	t:1	t:2
Number of Machine Loads	567	562
Percentage of existed exceptional elements	0	0
Number of observed voids after scheduling	1	1
Amount of outsourcing	188	60
Amount of backorders	8	0
Number of internal movements	3	5
Number of external movements	16	14
Number of observed part routings	35	28
Number of bottleneck machines	0	0
Maximum machine load variation	0	0
Maximum work-load variation	0	0

7.2 Solving experiments derived from the literature

For solving such problem we used the strategy that offered by Li et al. (2012) as they run their metaheuristic 10 times for each experiment and then the best solution is reported. The results in Table 8 show that for small size problems of CMS model, the solving algorithms reported the same results but for the medium and large scale problems, in most of the cases ACO-SA provided better solutions than Branch and Bound algorithm.

Table 8
Solving mathematical problems with ACO-SA and Branch and Bound

	Algorithm	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	Std.	Max	Min
Small Scale	BnB Heuristic	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	0	4454	4454
	Hybrid ACO-SA	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	4454	0	4454	4454
Medium Scale	BnB Heuristic	9185	9455	9195	9185	9191	9185	9185	9195	9185	9191	9215.2	84.361392	9455	9185
	Hybrid ACO-SA	9185	9193	9185	9185	9185	9185	9191	9185	9193	9191	9187.8	3.6757463	9193	9185
Large Scale	BnB Heuristic	16735	16935	16835	16435	15635	16735	15535	16735	17035	17235	16585	568.13535	17235	15535
	Hybrid ACO-SA	17035	16735	15635	16635	16535	16435	16835	16435	16635	16435	16535	371.18429	17035	15635

In continue, to verify the proposed hybrid metaheuristics, 17 experiments are solved with data gathered from literature. The results are then compared with results of BnB algorithm as shown by Table 9. Fig. 15 shows the minimizing process for some experiments in Table 9.

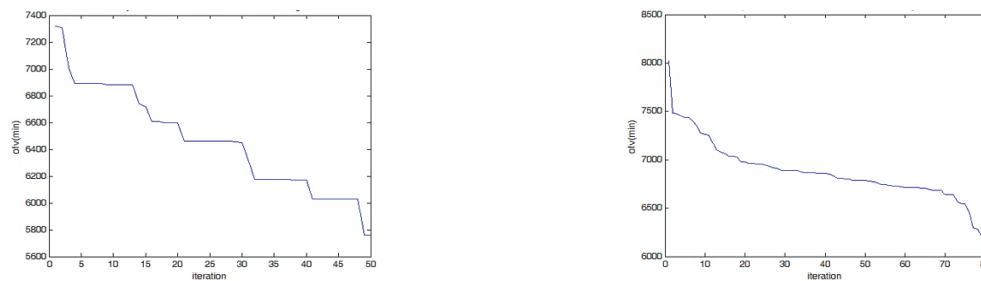


Fig. 15. The objective function graph of ACO-SA method for experiment number 16 and 17 (left to right) in table 10

Table 10 depicts the workstations that are allocated more than the average number of similar parallel machines in a workshop. While a workstation allocated more than mean of the other parallel machines inside the workshop, the workload variation emerges. Such phenomenon will be increased whenever number of broken machines increases.

Table 9
Results of solving numerical examples from the literature for CCDCMS model

Scale	No.	Problem Source	K	I	m	j	L	T	C.S	NOP	BnB	ACO-SA	TS-SA	Min	R	Average
											TC	TC	TC			
small	1	Askin & Huang (2001)	2	2	2	2	2	4	8	[2 4]	4454	4454	4454	4454	0	4454
	2	Askin & Huang (2001)	2	2	4	2	2	4	20	[3 6]	6304	6304	6304	6304	0	6304
	3	Suer & Cedeno (1996)	1	4	4	4	4	4	15	[3 2 3 2]	19550	19550	19550	19550	0	19550
	4	Askin & Huang (2001)	8	2	2	2	2	4	4	[14 6]	1245	1245	1245	1245	0	1245
	5	Askin & Huang (2001)	8	2	2	2	2	4	4	[8 8]	1355	1355	1355	1355	0	1355
	6	Mahdavi et al. (2010)	2	4	4	4	4	2	20	[4 5 3 6]	21155	21130	21680	21130	25	21142.5
	7	Mahdavi et al. (2012)	2	4	4	4	4	4	16	[4 3 6 5]	7030	6870	6978	6870	160	6950
medium	8	Aryanezhad et al. (2009)	3	3	3	3	3	3	20	[5 4 5]	6030	6030	5760	6030	0	6030
	9	Aryanezhad et al. (2009)	5	4	5	5	4	4	8	[4 3 2 3 3]	5650	5509	5560	5509	141	5579.5
	10	Aryanezhad et al. (2009)	4	5	5	5	5	4	12	[2 4 3 4 4]	5571	5627	5671	5571	-56	5599
	11	Li et al. (2012)	2	5	5	5	5	4	15	[3 5 4 2 3]	10657	10663	11039	10657	-6	10660
	12	Mahdavi et al. (2012)	2	5	5	5	5	4	15	[7 8 5 9 4]	5706	5633	5687	5633	73	5669.5
	13	Mahdavi et al. (2010)	2	6	6	6	6	3	20	[5 4 3 6 5 4]	10319	10355	10389	10319	-36	10337
	14	Norman et al. (2002)	2	6	6	6	6	5	14	[3 5 2 4 3 4]	9455	9193	9455	9193	262	9324
Large	15	Askin & Huang (2001)	2	8	6	6	8	4	30	[3 2 2 3 2 3]	15697	16385	16600	15697	-688	16041
	16	Askin & Huang (2001)	2	8	8	8	8	4	20	[2 4 3 5 4 3 2 4]	17858	17117	17125	17117	741	17487.5
	17	Aryanezhad et al. (2009)	5	5	5	5	5	5	30	[4 3 4 3 3]	6221	6131	6148	6131	90	6176

TC: Total cost

7.3 Measuring system imbalance in uncertain product demands

As shown although in all cases, the proposed method found the minimum total system cost (which contains feasible part allocations) but in many cases work load variations emerged. Fig. 16 depicts an example of work-load variation recognition in a part of the first period of the final solution of experiment 8 in Table 10. Using a macro in Matlab 2016 that works based on the set center point cubic blocks which are set around the center points and all of the recognized sets of machines, the over-allocated machines can be identified.

Table 10
Observed work load variation in the solved experiments

No.	t=1	t=2	t=3	t=4	t=5
1	-	-	-	-	-
2	(1,2)	(2,5)	(1,7)	(2,6)	-
3	-	-	-	-	-
4	(10,12)	(7,12,16)	-	-	-
5	(8,9,19)	(3,8,14,18,19)	(4,8,13,19)	(4,8,9,12,19)	-
6	(12,13,17,18)	(7,10,12,13)	-	-	-
7	(21,27,28)	-	-	-	-
8	(23,32,34,35)	(4,12); (22,28,31)	(4,13,18)	(7,12); (23,29)	-
9	(1,2,4,11); (34,36)	(1,8,19); (34,36)	(5,6); (26,28)	-	-
10	(7,11); (28,29)	(7,9,10)	(9,11); (26,29)	-	(9,10,12,14); (26,29)
11	(5,11,12)	-	(2,3)	(3,15,24)	-
12	-	-	-	-	-
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	(6,7)	(5,8,11)	-
16	-	-	-	-	-
17	(18,23,26)	(22,25)	-	-	-

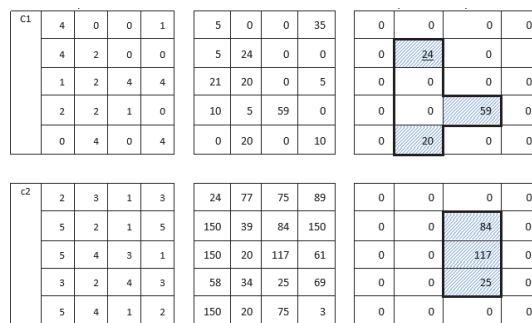


Fig. 16. Results of work-load variation for first period in final solution of experiment number 8

7.4 Improving the SCM performance using preventive maintenance plans

In this section, the effects of using preventive plans on reducing work-station load variation will be examined. During manufacturing process, machines may broke as result of depreciation, low quality raw material, etc. Therefore machine availability should not be considered fixed during scheduling of manufacturing systems as it reduces the capacity of workstation. To examine the effects of preventive maintenance plans all experiments in Table 9 were solved again by considering preventive actions (Fig. 17). The algorithm will perform preventive actions before a production period starts. Besides, emergency services were executed for machines broken to let them back to manufacturing process.

```

Editor - C:\Users\Aidin\Documents\
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
76 - Tournament_list=zeros(length,width,k,t,uuu);
77 - Tournament_obj=zeros(1,uuu);
78 - Pheromone=zeros(1,uuu);
79 - Observation=zeros(1,uuu);
80 - if nnz(Preventive_list)>0
81 -   for tt=1:t
82 -     for b=1:l:j
83 -       Preventive_list(tt,b)=PM(Preventive_list(tt,b));
84 -     end
85 -   end
86 - %exact value can be find for each machine type in real cases, we assumed 0.5
87 - failure_rate=failure_rate*0.5;
88 - end
    
```

Fig. 17. Employing applicable preventive services at the beginning of planning periods

The outcomes in Fig. 18 show that using appropriate preventive plan can cause reducing work-load variation in 71% of studied cases.

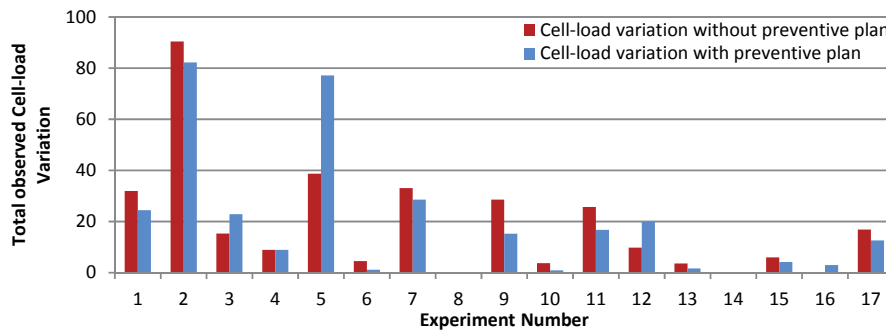


Fig. 18. Comparing work-load variation with and without using preventive services

Table 11

Observed values of work-load variation in the proposed model before and after using preventive plan

Problem	work-load variation			
	work-load variation without preventive plan	Total	work-load variation with preventive plan	Total
1	1.265; 7.5; 15.037; 8.133;	31.935	14.751; 1.884; 7.75;	24.385
2	43.52; 7.403; 12.469; 16.664; 10.399;	90.455	12.587; 16.217; 11.026; 8.419; 6.717; 5.3; 5.3;	82.23
3	6.748; 5.895; 2.386; 0.321;	15.35	2.302; 4.534; 6.965; 9.102;	22.903
4	3.551; 4.521; 0.808;	8.88	4.459; 4.461;	8.92
5	1.47; 1.47; 10.099; 4.929; 16.107; 4.623;	38.698	14.397; 14.397; 12.599; 13.604; 21.34; 0.507; 0.306;	77.15
6	2.87; 1.677;	4.547	1.183;	1.183
7	0.419; 5.752; 1.917; 3.27; 6.26; 4.36; 11.132;	33.11	5.504; 1.875; 2.08; 1.823; 0.662; 3.188; 0.188; 12.613; 0.695;	28.628
8	0	0	0	0
9	1.186; 8.769; 2.067; 0.139; 9.712; 2.357; 4.305;	28.535	0.391; 10.297; 1.774; 1.112; 1.689;	15.263
10	1.972; 1.745;	3.717	0.937;	0.937
11	0.556; 1.423; 3.206; 6.491; 0.446; 1.387; 8.045; 1.404;	25.726	0.8072; 0.603; 1.029; 0.029; 8.877; 5.252; 0.12;	16.7172
12	3.645; 0.322; 1.183; 1.006; 1.417; 1.079; 1.113;	9.765	4.933; 8.183; 6.726;	19.842
13	1.174; 0.169; 2.25;	3.593	0.803; 0.803;	1.606
14	0	0	0	0
15	0.875; 2.358; 0.315; 0.417; 0.178; 1.787;		0.625; 0.45; 1.518; 1.607;	4.2
16	0	0	1.511; 0.733; 0.733;	2.977
17	8.174; 4.396; 1.304; 2.956;	16.83	1.429; 2.25; 1.518; 4.787; 2.633;	12.617

8. Conclusion

In this research, a new method has been proposed to assess the impact of machine uncertainty through scheduling material in process in a forward supply chain while product demands are uncertain and can be varied from period to period. For this purpose, an NL-MIP mathematical programming has been developed while materials can transfer inside or between workshops. The model can provide different part routes alternatives and choose the best one using a hybrid ACO-SA. The outcomes of using the algorithm for solving problems have shown that the algorithm can successfully solve uncertain supply chains to minimize the amounts of lost sale. It has also found that while machine broken increases, the system costs and lost sale values could raise up accordingly. In this regard, using preventive maintenance plan to reduce number of machine failures plays key roles. The findings show that in 71% of studied cases, the proposed method can cause less system costs by using appropriate plan for periodic services. As an industrial application, the proposed method is suitable for scheduling of supply chains to determine the quota of in-house manufacturing and outsourcing to avoid lost sale.

Future expansion of the proposed method by developing multi-objective decision making model is suggested. It is also recommended to use a closed loop supply chain in order to use feedback to utilize the system.

Acknowledgement

The authors would like to thank Mohammad Gholami (Post-doctoral fellow, University of Calgary, CA) for his positive comments during the writing of the manuscript. We also appreciate the editor and anonymous reviewers for their constructive comments to improve the manuscript.

References

- Arora, R., Haleem, A., & Farooque, J. (2017). Impact of critical success factors on successful technology implementation in Consumer Packaged Goods (CPG) supply chain. *Management Science Letters*, 7(5), 213-224.
- Azadegan, A., Porobic, L., Ghazinoory, S., Samouei, P., & Kheirkhah, A. S. (2011). Fuzzy logic in manufacturing: A review of literature and a specialized application. *International Journal of Production Economics*, 132(2), 258-270.
- Azmi, F., Abdullah, A., Bakri, M., Musa, H., & Jayakrishnan, M. (2018). The adoption of halal food supply chain towards the performance of food manufacturing in Malaysia. *Management Science Letters*, 8(7), 755-766.
- Baykasoglu, A., & Gocken, T. (2010). Multi-objective aggregate production planning with fuzzy parameters. *Advances in Engineering Software*, 41(9), 1124-1131.
- Brandenburg, M., Govindan, K., Sarkis, J., & Seuring, S. (2014). Quantitative models for sustainable supply chain management: Developments and directions. *European Journal of Operational Research*, 233(2), 299-312.
- Delgoshaei, A., Ali, A., Ariffin, M. K. A., & Gomes, C. (2016a). A multi-period scheduling of dynamic cellular manufacturing systems in the presence of cost uncertainty. *Computers & Industrial Engineering*, 100, 110-132.
- Delgoshaei, A., Ariffin, M., Baharudin, B., & Leman, Z. (2016b). A new method for decreasing cell-load variation in dynamic cellular manufacturing systems. *International Journal of Industrial Engineering Computations*, 7(1), 83-110.
- Delgoshaei, A., Ariffin, M. K. A. M., Leman, Z., Baharudin, B. H. T. B., & Gomes, C. (2016c). Review of evolution of cellular manufacturing system's approaches: Material transferring models. *International Journal of Precision Engineering and Manufacturing*, 17(1), 131-149.

- Delgoshaei, A., Ariffin, M. K. A., & Ali, A. (2017). A multi-period scheduling method for trading-off between skilled-workers allocation and outsource service usage in dynamic CMS. *International Journal of Production Research*, 55(4), 997-1039.
- Delgoshaei, A., & Gomes, C. (2016). A multi-layer perceptron for scheduling cellular manufacturing systems in the presence of unreliable machines and uncertain cost. *Applied Soft Computing*, 49, 27-55.
- Egilmez, G., & Süer, G. (2014). The impact of risk on the integrated cellular design and control. *International Journal of Production Research*, 52(5), 1455-1478.
- Egilmez, G., Süer, G. A., & Huang, J. (2012). Stochastic cellular manufacturing system design subject to maximum acceptable risk level. *Computers & Industrial Engineering*, 63(4), 842-854.
- El-Baz, M. A. (2011). Fuzzy performance measurement of a supply chain in manufacturing companies. *Expert Systems with Applications*, 38(6), 6681-6688.
- Figueroa-García, J. C., Kalenatic, D., & Lopez-Bello, C. A. (2012). Multi-period mixed production planning with uncertain demands: fuzzy and interval fuzzy sets approach. *Fuzzy Sets and Systems*, 206, 21-38.
- Janaki, D., Izadbakhsh, H., & Hatefi, S. (2018). The evaluation of supply chain performance in the Oil Products Distribution Company, using information technology indicators and fuzzy TOPSIS technique. *Management Science Letters*, 8(8), 835-848.
- Jeon, G., & Leep, H. R. (2006). Forming part families by using genetic algorithm and designing machine cells under demand changes. *Computers & operations research*, 33(1), 263-283.
- Jia, G. Z., & Bai, M. (2011). An approach for manufacturing strategy development based on fuzzy-QFD. *Computers & Industrial Engineering*, 60(3), 445-454.
- Li, Q., Gong, J., Fung, R. Y., & Tang, J. (2012). Multi-objective optimal cross-training configuration models for an assembly cell using non-dominated sorting genetic algorithm-II. *International Journal of Computer Integrated Manufacturing*, 25(11), 981-995.
- Liang, T. F., Cheng, H. W., Chen, P. Y., & Shen, K. H. (2011). Application of fuzzy sets to aggregate production planning with multiproducts and multitime periods. *IEEE Transactions on Fuzzy Systems*, 19(3), 465-477.
- Miller, S., & John, R. (2010). An interval type-2 fuzzy multiple echelon supply chain model. In *Research and Development in Intelligent Systems XXVI* (pp. 407-420). Springer, London.
- Mula, J., Pedro, D., & Poler, R. (2010). The effectiveness of a fuzzy mathematical programming approach for supply chain production planning with fuzzy demand. *International Journal of Production Economics*, 128(1), 136-143.
- Olugu, E. U., & Wong, K. Y. (2012). An expert fuzzy rule-based system for closed-loop supply chain performance assessment in the automotive industry. *Expert Systems with Applications*, 39(1), 375-384.
- Pedro, D., Mula, J., Jiménez, M., & del Mar Botella, M. (2010). A fuzzy linear programming based approach for tactical supply chain planning in an uncertainty environment. *European Journal of Operational Research*, 205(1), 65-80.
- Qin, Z., Bai, M., & Ralescu, D. (2011). A fuzzy control system with application to production planning problems. *Information Sciences*, 181(5), 1018-1027.
- Renna, P., & Ambrico, M. (2015). Design and reconfiguration models for dynamic cellular manufacturing to handle market changes. *International Journal of Computer Integrated Manufacturing*, 28(2), 170-186.
- Safaei, N., & Tavakkoli-Moghaddam, R. (2009). Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems. *International Journal of Production Economics*, 120(2), 301-314.
- Süer, G. A., Huang, J., & Maddisetty, S. (2010). Design of dedicated, shared and remainder cells in a probabilistic demand environment. *International journal of production research*, 48(19), 5613-5646.

- Tavakkoli-Moghaddam, R., Aryanezhad, M. B., Safaei, N., & Azaron, A. (2005). Solving a dynamic cell formation problem using metaheuristics. *Applied Mathematics and Computation*, 170(2), 761-780.
- Tavakkoli-Moghaddam, R., Aryanezhad, M. B., Safaei, N., Vasei, M., & Azaron, A. (2007a). A new approach for the cellular manufacturing problem in fuzzy dynamic conditions by a genetic algorithm. *Journal of Intelligent & Fuzzy Systems*, 18(4), 363-376.
- Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., & Safaei, N. (2007b). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. *Applied Mathematics and Computation*, 184(2), 721-728.
- Venkata Rao, R., & Patel, B. K. (2010). Decision making in the manufacturing environment using an improved PROMETHEE method. *International Journal of Production Research*, 48(16), 4665-4682.
- Vinodh, S., & Balaji, S. R. (2011). Fuzzy logic based leanness assessment and its decision support system. *International Journal of Production Research*, 49(13), 4027-4041.
- Wong, B. K., & Lai, V. S. (2011). A survey of the application of fuzzy set theory in production and operations management: 1998–2009. *International Journal of Production Economics*, 129(1), 157-168.
- Zhao, F., Hong, Y., Yu, D., Yang, Y., & Zhang, Q. (2010). A hybrid particle swarm optimisation algorithm and fuzzy logic for process planning and production scheduling integration in holonic manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 23(1), 20-39.



© 2018 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).