International Journal of Applied Operational Research
Vol. 4, No. 4, pp. 87-106, Autumn 2014

Journal homepage: ijorlu.liau.ac.ir

An Integrated Aggregate Production Planning Model with Two-Phase Production System and Maintenance Costs

E. Mehdizadeh*, A. A. Atashi –Abkenar

Received: 20 February 2014 ; **Accepted:** 7 July 2014

Abstract Aggregate production planning (APP) is one of the most important issues carried out in manufacturing environments which seeks efficient planning, scheduling and coordination of all production activities that optimizes the company's objectives. In this paper, we develop a mixed integer linear programming (MILP) model for an integrated aggregate production planning system with closed loop supply chain and preventive maintenance. The goal is to minimize setup costs, production costs, labor costs and preventive maintenance (PM) costs and instabilities in the work force and inventory levels. Due to NP-hard class of APP, we implement genetic algorithm (GA), harmony search (HS) and vibration damping optimization (VDO) for solving this model. Additionally, the Taguchi method is conducted to calibrate the parameter of the meta-heuristics and select the optimal levels of the algorithm's performance influential factors. Finally, computational results on a set of randomly generated instances show the efficiency of the VDO algorithm against the other meta-heuristics, and this algorithm obtain good quality solutions for aggregate production planning with preventive maintenance and could be efficient for large scale problems.

Keywords: Aggregate production planning, Preventive maintenance, Genetic algorithm, Harmony search, Vibration damping optimization.

1 Introduction

Aggregate production planning (APP) is a medium-range planning. Aggregate production plans are necessary to maximize workforce opportunity and constitute a crucial part of operations management, and help match supply and demand while minimizing costs. Aggregate production planning applies the upper-level forecasts to lower-level, production-floor scheduling and is most effective when applied to periods 2 to 18 months in the future. Plans generally either "chase" demand, adjusting workforce accordingly, or are "level" plans, meaning that labor is relatively constant with fluctuations in demand being met by inventories and back orders. One of the first multi-objective models is presented by Masud and Hwang. They presented a multiple objective formulation of the multi-product, multi-period aggregate production planning problem. For solving this model, they used three multiple objective decision making methods [1]. Lee studies a two-machine flowshop scheduling problem with

*Corresponding Author.
Email: emehdi@qiau.ac.ir (E. Mehdizadeh)

E. Mehdizadeh
Assistant professor, Department of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

A. A. Atashi -Abkenar
M.Sc. Student, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

an availability constraint. He assumes that a machine may not always be available. Also if a machine continues to process those unfinished jobs that were scheduled in the previous planning period, then it is not available at the beginning of the period. Lee studies the problem under a deterministic environment. Namely, Lee assumes that the unavailable time is known in advance. He proves that the problem is NP-hard and he develop pseudo-polynomial dynamic programming algorithm to solve the problem optimally [2]. Leung et al propose a goal programming approach to multi-site aggregate production planning with multiple objectives such as the maximization of profit, minimization of the change of workforce level and maximization of utilization of import quota [3]. Wang and Liang presented a novel interactive possibility linear programming (PLP) approach for solving the multi-product aggregate production planning (APP) problem with imprecise forecast demand, related operating costs, and capacity [4]. Leung and Chan address the aggregate production planning problem with different operational constraints, including production capacity, workforce level, factory locations, machine utilization, storage space and other resource limitations. A pre-emptive goal programming model is developed to maximize profit, minimize repairing cost and maximize machine utilization of the Chinese production plant hierarchically [5]. Ramezanian et al develop a mixed integer linear programming (MILP) model for general two-phase aggregate production planning systems. The goal is to minimize costs and instabilities in the work force and inventory levels. They presented genetic algorithm and tabu search for solving this problem [6]. Sadeghi et al proposed a multi-objective model for aggregate planning problem in which the parameters of the model are expressed in the form of grey numbers. The suggested grey multi-objective model is solved based on a goal programming problem with fuzzy aspiration levels [7]. Wang and Yeh, present a scheme of an aggregate production planning (APP) from a manufacturer of gardening equipment. It is formulated as an integer linear programming model and optimized by PSO [8].

Preventive maintenance, where equipment is maintained before break down occurs. This type of maintenance has many different variations and is subject of various researches to determine best and most efficient way to maintain equipment. Preventive maintenance (PM) has the following meanings: (1) The care and servicing by personnel for the purpose of maintaining equipment and facilities in satisfactory operating condition by providing for systematic inspection, detection, and correction of incipient failures either before they occur or before they develop into major defects. (2) Maintenance, including tests, measurements, adjustments, and parts replacement, performed specifically to prevent faults from occurring. Production planning models seek typically to balance the costs of setting up the system with the costs of production and materials holding, while maintenance models attempt typically to balance the costs and benefits of sound maintenance plans in order to optimize the performance of the production system. In both domains, issues of production modeling and maintenance modeling have experienced an evident success both from theoretical and applied viewpoints. Paradoxically the issue of combining production and maintenance plans has received much less attention [9]. Adiri et al, presented for the first time, a production planning model for machine failure costs. They show that the single-machine scheduling problem with machine failure, even when the failure is already known from the type of NP-Hard problems [10]. Wienstein and Chung proposed a three-part model for evaluation of maintenance policies. In their approach, the Aggregate production planning is considered for the first time. In order to resolve often conflicting objectives of system reliability and profit maximization, an organization should establish appropriate maintenance guidelines that take into consideration (1) costs associated with performing production activities, (2) costs associated with performing maintenance activities, and (3) the various costs associated with equipment failure and the resulting interruptions to the production plan. In currently prevailing practices, maintenance policy often is determined at the operational level in a political test between production and maintenance management. The resulting policy often is not optimal for the organization's overall objectives [11]. Lee and Chen study the problem of processing a set of n jobs on m parallel machines where each machine must be maintained once during the planning horizon. Their objective is to schedule jobs and maintenance activities so that the total weighted completion time of jobs is minimized [12]. Aghezzaf and Najid discuss the issue of integrating production planning and preventive maintenance in manufacturing production systems. In particular, it tackles the problem of integrating production and preventive maintenance in a system composed of parallel failure-prone production lines. It is assumed that when a production line fails, a minimal repair is carried out to restore it to an 'as-bad-as-old' status. Preventive maintenance is carried out, periodically at the discretion of the decision maker, to restore the production line to an 'as-good-as-new' status. It is also assumed that any maintenance action, performed on a production line in a given period, reduces the available production capacity on the line during that period [13]. Pan et al. suggested an integrated scheduling model incorporating both production scheduling and preventive maintenance planning for a single machine in order to minimize the maximum weighted tardiness [14]. Nourelfath and Chatelet paper deals with the problem of integrating preventive maintenance and tactical production planning, for a production system composed of a set of parallel components, in the presence of economic dependence and common cause failures. Economic dependence means that performing maintenance on several components jointly costs less money and time than on each component separately. Common cause failures correspond to events that lead to simultaneous failure of multiple components due to a common cause [15]. Yalaoui et al propose an extended linear programming model as a hybrid approach for computing the optimum production plan with minimum total cost. The dual objective problem of production planning and maintenance is treated into a mixed integer linear program. This program is not only considering cases of multi-lines, multi-periods and multi-items but also taking into account the deterioration of the lines. This deterioration is represented in the model as a reduction of production lines capacities in function of the time evolution. Maintenance operations are supposed to provide lines in an operational state as good as new, i.e. with a maximum capacity [16].

The large part of the production planning models assumes that the system will function at its maximum performance during the planning horizon, and the large part of the maintenance planning models disregards the impact of maintenance on the production capacity and does not explicitly consider the production requirements [9]. It is therefore crucial that both production and maintenance aspects related to a production system are concurrently considered during the elaboration of optimal production and maintenance plans. The purpose of this paper is to develop a combined production planning model for two phase production systems and preventive maintenance in an aggregate production planning. The main objective of the proposed model is to determine an integrated production and maintenance plan that minimizes the expected total production and maintenance costs over a planning horizon.

The remaining of this paper is organized as follows: Section 2 describes a MILP formulation of the aggregate production planning model with preventive maintenance. The solution approaches genetic algorithm (GA), harmony search (HS) and vibration damping optimization (VDO) are presented in Sections 3, 4 and 5. Section 6 presents computational experiments. The conclusions and suggestions for future studies are included in Section 7.

2 Problem formulation

In this section, we present a MILP formulation of the problem. This model is relevant to multi-period, multi-product, multi-machine, two-phase production systems.

2.1 Assumptions

- The quantity shortage at the beginning of the planning horizon is zero.
- The quantity shortage at the end of the planning horizon is zero.
- Maintenance decision variable, if maintenance to be performed, the decision variable is equal to one, and otherwise it is zero.
- There is a setup cost of producing a product only once at the beginning of a period, And the setup cost after a failure is not considered.
- If maintenance is not performed in period t, the time and cost of maintenance will not apply to the model, the failure costs will be considered in period t+1 instead, and downtime will be deducted from available machine capacity.

2.2 Decision variables

P_{i2t}: Regular time production of second-phase product i in period t (units).
O_{i2t}: Over time production of second-phase product i in period t (units).
C_{i2t}: Subcontracting volume of second-phase product i in period t (units).
B_{i2t}: Backorder level of second-phase product i in period t (units).
I_{i2t}: The inventory of second-phase product i in period t (units).
H_t: The number of second group workers hired in period t (man-days).
L_t: The number of second group workers laid off in period t (man-days).
W_t: Second workforce level in period t (man-days).
Y_{i2t}: The setup decision variable of second-phase product i in period t, a binary integer variable.
XR_{i2t}: The number of second-phase returned products of product i that remanufactured in period t.
XRI_{i2t}: The number of second-phase returned products of product i held that in inventory at the end of period t.
XD_{i2t}: The number of second-phase returned products of product i that disposed in period t.
P_{k1t}: Regular time production of first-phase product k in period t (units).
O_{k1t}: Over time production of first-phase product k in period t (units).
C_{k1t}: Subcontracting volume of first-phase product k in period t (units).
B_{k1t}: Backorder level of first-phase product k in period t (units).
I_{k1t}: The inventory of first-phase product k in period t (units).
H'_t: The number of first group workers hired in period t (man-days).
L'_t: The number of first group workers laid off in period t (man-days).
W'_t: First workforce level in period t (man-days).
Y_{k1t}: The setup decision variable of first-phase product k in period t, a binary integer variable.
PMF_{lt}: The preventive maintenance decision variable of first-phase machine l in period t, a binary integer variable.

PMS_{jt}: The preventive maintenance decision variable of second-phase machine j in period t, a binary integer variable.

2.3 Parameters

p_{k1t}: Regular time production cost of first-phase product k in period t ($/units).
o_{k1t}: Over time production cost of first -phase product k in period t ($/units).
c_{k1t}: Subcontracting cost of first-phase product k in period t ($/units).
h_{k1t}: Inventory cost of first-phase product k in period t ($/units).
a_{k1l}: Hours of machine l per unit of first-phase product k (machine-days/unit).
u_{k1l}: The setup time for first-phase product k on machine l (hours).
r_{k1lt}: The setup cost of first-phase product k on machine l in period t ($/machine-hours).
R'_{kt}: The regular time capacity of machine l in period t (machine-hours).
hr'_t: Cost to hire one worker in period t for first group labor ($/man-days).
l'_t: Cost to layoff one worker of first group in period t ($/man-days).
w'_t: The first group labor cost in period t ($/man-days).
I_{k10}: The initial inventory level of first-phase product k in period t (units).
w'_0: The initial first group workforce level (man-days).
B_{k10}: The initial first group backorder level (man-days).
e_{k1}: Hours of labor per unit of first-phase product k (man-days/unit).
α'_t: The ratio of regular-time of first group workforce available for use in overtime in period t.
β'_{lt}: The ratio of regular time capacity of machine l available for use in overtime in period t.
$w'_{max\,t}$: Maximum level of first group labor available in period t (man-days).
D_{i2t}: Forecasted demand of second-phase product i in period t (units).
p_{i2t}: Regular time production cost of second-phase product i in period t ($/units).
o_{i2t}: Over time production cost of second-phase product i in period t ($/units).
c_{i2t}: Subcontracting cost of second-phase product i in period t ($/units).
h_{i2t}: Inventory cost of second-phase product i in period t ($/units).
a_{i2j}: Hours of machine j per unit of second-phase product i (machine-days/unit).
u_{i2j}: The setup time for second-phase product i on machine j (hours).
r_{i2jt}: The setup cost of second-phase product i on machine j in period t ($/machine-hours).
R_{jt}: The regular time capacity of machine j in period t (machine-hours).
hr_t: Cost to hire one worker in period t for second group labor ($/man-days).
l_t: Cost to layoff one worker of second group in period t ($/man-days).
w_t: The first group labor cost in period t ($/man-days).
I_{i20}: The initial inventory level of second-phase product i in period t (units).
w_0: The initial second group workforce level (man-days).
B_{i20}: The initial second group backorder level (man-days).
e_{i2}: Hours of labor per unit of second-phase product i (man-days/unit).
α_t: The ratio of regular-time of second group workforce available for use in overtime in period t.
β_{jt}: The ratio of regular time capacity of machine j available for use in overtime in period t.
f: The working hours of labor in each period (man-hour/man-day).
$w_{max\,t}$: Maximum level of second group labor available in period t (man-days).
$C_{max\ it}$: Maximum subcontracted volume available of second-phase product i in period t (units).
f_{ik}: The number of unit of first-phase product k required per unit of first-phase product i.

TR_{i2t}: The number of second-phase returned products of product i in period t.
$XD_{max\ i2t}$: The maximum number of second-phase returned products of product i that could be disposed in period t.
$XR_{max\ i2t}$: The maximum number of second-phase returned products of product i that could be remanufactured in period t.
hX_{i2t}: Inventory cost of second-phase returned products of product i in period t (\$/units).
MTS_{jt}: The preventive maintenance time of second-phase machine j in period t (minutes).
MTF_{lt}: The preventive maintenance time of first-phase machine j in period t (minutes).
$C1_{l1t}$: Failure cost of first-phase machine l in period t (\$).
$C2_{l1t}$: Maintenance cost of first-phase machine l in period t (\$).
$C3_{j2t}$: Failure cost of second-phase machine j in period t (\$).
$C4_{j2t}$: Maintenance cost of second-phase machine j in period t (\$).
$C5_{i2t}$:The cost of returned products of second-phase product i that disposed in period t (\$).
$C6_{i2t}$: The cost of returned products of second-phase product i that remanufactured in period t (\$).
m : Percentage of machine capacity in each period (due to lack of maintenance in the previous period) is lost due to Failure.
LT :Lead time.
M: A large number.

2.4 The proposed model

$$\begin{aligned}
MinZ =& \sum_{i=1}^{N}\sum_{t=1}^{T}(p_{i2t}P_{i2t}+o_{i2t}O_{i2t}+c_{i2t}C_{i2t})+\sum_{k=1}^{K}\sum_{t=1}^{T}(p_{k1t}P_{k1t}+o_{k1t}O_{k1t}+c_{k1t}C_{k1t})+\\
&\sum_{t=1}^{T}\sum_{i=1}^{N}h_{i2t}I_{i2t}+\sum_{t=1}^{T}\sum_{k=1}^{K}h_{k1t}I_{k1t}+\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{J}r_{i2jt}Y_{i2t}+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{l=1}^{L}r_{k1lt}Y_{k1t}+\sum_{i=1}^{N}\sum_{t=1}^{T}b_{i2t}B_{i2t}+\\
&\sum_{i=1}^{N}\sum_{t=1}^{T}b_{k1t}B_{k1t}+\sum_{t=1}^{T}(hr_tH_t+l_tL_t)+\sum_{t=1}^{T}w_tW_t+\sum_{t=1}^{T}(hr'_tH'_t+l'_tL'_t)+\sum_{t=1}^{T}w'_tW'_t+\\
&\sum_{l=1}^{L}\sum_{t=1}^{T}C1_{l1t}(1-PMF_{l,t-1})+\sum_{l=1}^{L}\sum_{t=0}^{T-1}C2_{l1t}PMF_{lt}+\sum_{j=1}^{J}\sum_{t=1}^{T}C3_{j2t}(1-PMS_{j,t-1})+\\
&\sum_{l=1}^{J}\sum_{t=0}^{T-1}C4_{j2t}PMS_{jt}+\sum_{i=1}^{N}\sum_{t=1}^{T}C5_{i2t}XD_{i2t}+\sum_{i=1}^{N}\sum_{t=1}^{T}C6_{i2t}XR_{i2t}+\sum_{i=1}^{N}\sum_{t=1}^{T}hX_{i2t}XRI_{i2t}
\end{aligned} \tag{1}$$

$$P_{i2t}+O_{i2t}+C_{i2t}+XR_{i2t}+B_{i2t}-B_{i2,t-1}+I_{i2,t-1}-I_{i2t}=D_{i2t}; \tag{2}$$
$$i=1,2,...,N \qquad t=1,2,...,T$$

$$P_{k1t}+O_{k1t}+C_{k1t}+B_{k1t}-B_{k1,t-1}+I_{k1,t-1}-I_{k1t}=\sum_{i=1}^{N}f_{ik}(P_{i2,t+LT}+O_{i2,t+LT}); \tag{3}$$
$$k=1,2,...,K \qquad t=1,2,...,T$$

$$C_{k10}+I_{k10}=\sum_{i=1}^{N}f_{ik}(P_{i2,LT}+O_{i2,LT}); \qquad k=1,...,K \quad \text{LT}=1 \tag{4}$$

$$\sum_{i=1}^{N}(a_{i2j}P_{i2t}+U_{i2j}Y_{i2t})+PMS_{jt}MTS_{jt}+(1-PMS_{j,t-1})m\ R_{jt}\le R_{jt}; \tag{5}$$
$$t=1,2,...,T \qquad j=1,2,...,J$$

$$\sum_{i=1}^{N}(a_{i2j}O_{i2t})+(1-PMS_{j,t-1})m\ \beta_{jt}\ R_{jt}\le\beta_{jt}R_{jt}; \quad t=1,2,...,T \qquad j=1,2,...,J \tag{6}$$

$$\sum_{i=1}^{N}(a_{k1l}P_{k1t}+U_{k1l}Y_{k1t})+PMF_{lt}MTF_{lt}+(1-PMF_{l,t-1})m\ R'_{lt}\le R'_{lt};\quad t=1,...,T\quad l=1,...,L \tag{7}$$

$$\sum_{i=1}^{N}(a_{k1j}O_{k1t})+(1-PMF_{l,t-1})m\ \beta'_{lt}\ R'_{lt}\le \beta'_{lt}\ R'_{lt};\quad t=1,2,...,T\quad l=1,2,...,L \tag{8}$$

$$P_{k1t}+O_{k1t}\le MY_{k1t};\qquad k=1,2,...,K\quad t=1,2,...,T \tag{9}$$

$$P_{i2t}+O_{i2t}\le MY_{i2t};\qquad i=1,2,...,N\quad t=1,2,...,T \tag{10}$$

$$W_t=W_{t-1}+H_t-L_t;\qquad t=1,2,...,T \tag{11}$$

$$W'_t=W'_{t-1}+H'_t-L'_t;\qquad t=1,2,...,T \tag{12}$$

$$\sum_{k=1}^{K}e_{k1}P_{k1t}\le fw'_t;\qquad t=1,2,...,T \tag{13}$$

$$\sum_{k=1}^{K}e_{k1}O_{k1t}\le \alpha'_t fw'_t;\qquad t=1,2,...,T \tag{14}$$

$$\sum_{i=1}^{N}e_{i2}P_{i2t}\le fw_t;\qquad t=1,2,...,T \tag{15}$$

$$\sum_{i=1}^{N}e_{i2}O_{i2t}\le \alpha_t fw_t;\qquad t=1,2,...,T \tag{16}$$

$$w_t\le w_{\max t};\qquad t=1,2,...,T \tag{17}$$

$$w'_t\le w'_{\max t};\qquad t=1,2,...,T \tag{18}$$

$$C_{i2t}\le C_{\max i2t};\qquad i=1,2,...,N\quad t=1,2,...,T \tag{19}$$

$$B_{i2t}.I_{i2t}=0;\qquad i=1,2,...,N\quad t=1,2,...,T \tag{20}$$

$$B_{k1t}.I_{k1t}=0;\qquad k=1,2,...,K\quad t=1,2,...,T \tag{21}$$

$$XR_{i2t}=XRI_{i2,t-1}-XD_{i2t}-XR_{i2t}+TR_{i2t};\quad i=1,2,...,N\quad t=1,2,...,T \tag{22}$$

$$XD_{i2t}\le XD_{\max i2t};\qquad i=1,2,...,N\quad t=1,2,...,T \tag{23}$$

$$XR_{i2t}\le XR_{\max i2t};\qquad i=1,2,...,N\quad t=1,2,...,T \tag{24}$$

$$Y_{i2t}=\{0,1\};\qquad i=1,2,...,N\quad t=1,2,...,T \tag{25}$$

$$Y_{k2t}=\{0,1\};\qquad k=1,2,...,K\quad t=1,2,...,T \tag{26}$$

$$PMF_{lt}=\{0,1\};\qquad l=1,2,...,L\quad t=1,2,...,T \tag{27}$$

$$PMS_{jt}=\{0,1\};\qquad j=1,2,...,J\quad t=1,2,...,T \tag{28}$$

$$B_{i2T}=0;\qquad i=1,2,...,N \tag{29}$$

$$B_{k1T}=0;\qquad k=1,2,...,k \tag{30}$$

$$PMF_{l0}=1;\qquad l=1,2,...,L \tag{31}$$

$$PMS_{j0}=1;\qquad j=1,2,...,J \tag{32}$$

3 The genetic algorithm

In the computer science field of artificial intelligence, genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a meta-heuristic) is routinely used to generate useful solutions to optimization and search problems [17]. The GA proposed by Holland (1975) to encode the features of a problem by chromosomes, where each gene represents a feature of the problem. In general, GA consists of the following steps:

Step 1: Initialize a population of chromosomes.
Step 2: Evaluate the fitness of each chromosome.
Step 3: Create new chromosomes by applying genetic operators such as reproduction, crossover and mutation to current chromosomes.
Step 4: Evaluate the fitness of the new population of chromosomes.
Step 5: If the termination condition is satisfied, stop and return the best chromosome; otherwise, go to Step 3.
Our implementation of genetic algorithm is presented as follow:

3.1. Representation schema

To design genetic algorithm for mentioned problem, a suitable representation scheme that shows the solution characteristics is needed. In this paper, each gene is total aggregate production (X) of second-phase products and a chromosome is a production plan. The X is decomposed to the regular time production, overtime production, returned products that could be remanufactured and subcontracting volume. The general structure of the solution representation performed for running the genetic algorithm on second-phase with six periods and two products is shown in Fig 1.

Total aggregate production for second-phase product 1	X_{121}	X_{122}	X_{123}	X_{124}	X_{125}	X_{126}
Total aggregate production for second-phase product 2	X_{221}	X_{222}	X_{223}	X_{224}	X_{225}	X_{226}

Fig.1 Chromosome representation

3.2 Selection

The selection provides the opportunity to deliver the gene of a good solution to next generation. There are various selection operators available that can be used to select the parents. In this study, the tournament selection is employed.

3.3 Crossover

Crossover is a process in which chromosomes exchange genes through the breakage and reunion of two chromosomes to generate a number of children. Crossover's offspring should represent solutions that combine substructures of their parents. In this study, crossover generates an offspring by combining two selective parents as shown in Fig 2 and Fig 3.

Parent1	X_{121}	X_{122}	X_{123}	X_{124}	X_{125}	X_{126}
	X_{221}	X_{222}	X_{223}	X_{224}	X_{225}	X_{226}

Offspring	X'_{121}	X'_{122}	X'_{123}	X_{124}	X_{125}	X_{126}
	X_{221}	X_{222}	X_{223}	X'_{224}	X'_{225}	X'_{226}

Parent2	X'_{121}	X'_{122}	X'_{123}	X'_{124}	X'_{125}	X'_{126}
	X'_{221}	X'_{222}	X'_{223}	X'_{224}	X'_{225}	X'_{226}

Fig.2 Illustration of the One-point crossover structure

Parent1	X_{121}	X_{122}	X_{123}	X_{124}	X_{125}	X_{126}
	X_{221}	X_{222}	X_{223}	X_{224}	X_{225}	X_{226}

Offspring	X_{121}	X'_{122}	X'_{123}	X_{124}	X'_{125}	X'_{126}
	X_{221}	X'_{222}	X'_{223}	X_{224}	X'_{225}	X'_{226}

Parent2	X'_{121}	X'_{122}	X'_{123}	X'_{124}	X'_{125}	X'_{126}
	X'_{221}	X'_{222}	X'_{223}	X'_{224}	X'_{225}	X'_{226}

Fig.3 Illustration of the two-point crossover structure

3.4 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This mutation operator takes the chosen genome and reduces the total aggregate production level for a random selective period by the amount of β and then it is added to other selective period at each row of current solution as shown in Fig 4 illustrates this operation.

Parent	X_{121}	X_{122}	X_{123}	X_{124}	X_{125}	X_{126}
	X_{221}	X_{222}	X_{223}	X_{224}	X_{225}	X_{226}

Offspring	X_{121}	X_{122}-β	X_{123}	X_{124}	X_{125}+β	X_{126}
	X_{221}-β	X_{222}	X_{223}	X_{224}+β	X_{225}	X_{226}

Fig.4. Illustration of the mutation structure

The principle of this operator is based on the following equation, shown for X_{122}:

$$\beta = X_{122} \times \varphi; \qquad \varphi \in [0.1,1] \qquad (33)$$

3.5 Fitness function

The fitness function is the same as the objective function which is defined in Section 2.

3.6 Termination condition

The search process stops if the some specified number of generations without improvement of the best known solution is reached. In our experiments we accepted stop= 100**.**

4 Harmony search

Harmony search (HS) algorithm was developed in an analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony [18]. The steps in the procedure of HS are as follows [19]:
Step 1. Initialize the problem and algorithm parameters.
Step 2. Initialize the harmony memory.
Step 3. New harmony improvisation.
Step 4. Update the harmony memory.
Step 5. Check the stopping criterion.
The pseudo-code of the original harmony search algorithm for the problem is shown in Fig 5:

Harmony search
Objective function $f(x_i)$, i=1 to N
Define HS parameters: HMS, HMCR, PAR, and BW
Generate initial harmonics (for i=1 to HMS)
Evaluate $f(x_i)$
While (t<Max number of iterations)
Create a new harmony: x_i^{new}, i=1 to N
If(U(0,1)>HMCR),
$x_i^{new}=x_j^{old}$, where x_j^{old} is a random from {1, …, HMS}
Else if If (U (0,1)>PAR),
$x_i^{new}=x_j^{old}$ + BW [(2×U (0,1))-1], where x_j^{old} is a random from {1,…,HMS}
Else
$x_i^{new}=x_L(i)+ U(0,1)\times[x_U(i) - x_L(i)]$
End if
Evaluation $f(x_i^{new})$
Accept the new harmonics (solutions) if better
End while
Fine the current best estimates

Fig. 5 Pseudo-code of the original harmony search

The search process stops if the some specified number of generations without improvement of the best known solution is reached. In our experiments we accepted Stop= 100.

5 Vibration damping optimization

Recently, a new heuristic optimization technique based on the concept of the vibration damping in mechanical vibration was introduced by Mehdizadeh and Tavakkoli-Moghaddam named vibration damping optimization (VDO) algorithm [20]. The VDO algorithm is illustrated in the following steps:
Step 1. Generating feasible initial solution.
Step 2. Initializing the algorithm parameters which consist of: initial amplitude (A_0), maximum Number of Sub-iteration (sub-it), number of generations without improvement (Stop), damping coefficient (γ), and standard deviation (σ =1). Finally, parameter S is set in one (S=1)
Step 3. Calculating the objective value U_0 for initial solution.
Step 4. Initializing the internal loop
In this step, the internal loop is carried out for l =1 and repeat while $l \leq$ sub-it.
Step 5. Neighborhood generation.

Step 6. Accepting the new solution
Set $\Delta = U - U_0$ Now, if $\Delta < 0$, accept the new solution, else if $\Delta > 0$ generate a random number between (0, 1);
If $r < 1 - \exp\left(\dfrac{-A_S^2}{2\sigma^2}\right)$, then accept a new solution;
Otherwise, reject the new solution and accept the previous solution.
If $l >$ sub-it, then $S+1 \rightarrow S$ and go to step 7; otherwise $l+1 \rightarrow l$ and go back to step 5.
Step 7. Adjusting the amplitude
In this step, $A_S = A_0 \exp(\frac{-\gamma S}{2})$ is used for reducing amplitude at each iteration of the outer cycle of the algorithm. If S>Stop return to step 8; otherwise, go back to step 4.
Step 8. Stopping criteria

In this step, the algorithm will be stopped after number of generations without improvement, we accepted Stop= 100. At the end, best solution is obtained.

5.1 Representation schema

In this paper, each chromosome is a production plan and each chromosome formed shown in Fig 1.

5.2 Neighborhood scheme

In this paper we use swap and insertion scheme, Fig 6 and Fig 7 illustrates this operation on second-phase with the six periods and two products. Swap and insertion are selected Roulette Wheel method.

Parent	X_{121}	X_{122}	X_{123}	X_{124}	X_{125}	X_{126}
	X_{221}	X_{222}	X_{223}	X_{224}	X_{225}	X_{226}

Offspring	X_{121}	X_{125}	X_{123}	X_{124}	X_{122}	X_{126}
	X_{224}	X_{222}	X_{223}	X_{221}	X_{225}	X_{226}

Fig.6 Illustration of the swap structure

Parent	X_{121}	X_{122}	X_{123}	X_{124}	X_{125}	X_{126}
	X_{221}	X_{222}	X_{223}	X_{224}	X_{225}	X_{226}

Offspring	X_{121}	X_{123}	X_{124}	X_{125}	X_{122}	X_{126}
	X_{222}	X_{223}	X_{221}	X_{224}	X_{225}	X_{226}

Fig.7 Illustration of insertion structure

6 Excremental Results

In order to evaluate the performance of the meta-heuristic algorithms, 30 test problems with different sizes are randomly generated. The proposed model is coded with LINGO 8 software and using LINGO solver for solving the instances. Furthermore, for the small and medium sized instances of two phases APP with PM, LINGO optimization solver is used to figure out the optimal solution and compared with GA, HS and VDO results.
The genetic algorithm, harmony search and Vibration Damping optimization are coded in MATLAB R2011a and all tests are conducted on a not book at Intel Core 2 Duo Processor 2.00 GHz and 2 GB of RAM.

6.1 Parameter calibration

Appropriate design of parameters has significant impact on efficiency of meta-heuristics. In this paper the Taguchi method applied to calibrate the parameters of the proposed methods namely GA, VDO and HS algorithms. The Taguchi method was developed by Taguchi [21]. This method is based on maximizing performance measures called signal-to-noise ratios in order to find the optimized levels of the effective factors in the experiments. The signal-to-noise ratio refers to the mean-square deviation of the objective function that minimizes the mean and variance of quality characteristics to make them closer to the expected values. For the factors that have significant impact on signal-to-noise ratio, the highest signal-to-noise ratio provides the optimum level for that factor. As mentioned before, the purpose of Taguchi method is to maximize the signal-to-noise ratio. In this subsection, the parameters for experimental analysis are determined. Table 1 lists different levels of the factors for GA, HS and VDO. In this paper according to the levels and the number of the factors, the Taguchi method L_{25} is used for the adjustment of the parameters.

Table 1 Factors and their levels

Factors	Algorithms	Notations	Levels	Values
Population Size		npop	5	25,50,75,100,125
Crossover Percentage	GA	Pc	5	0.3,0.45,0.6,0.75,0.9
Mutation Percentage		Pm	5	0.35,0.5,0.65,0.8,0.95
Strongly mutation Rate		mu	5	0.001,0.026,0.5,0.075,0.1
Harmony memory size		HMS	5	5,10,15,20,25
Harmony memory considering rate	HS	HMCR	5	0.7,0.75,0.8,0.85,0.9
Pitch-adjusting rate		PAR	5	0.1,0.15,0.2,0,25,0.3
Bandwidth		BW	5	0.2,0.5,0.8,0.9,0.99
Max of iteration at each amplitude		sub-it	5	5,10,15,20,25
Damping coefficient	VDO	γ	5	0.01,0.05,0.1,0.5,0.9
Initial amplitude		A_0	5	4,5,6,7,8

Figures 8, 9 and 10 show signal-to-noise ratios. Best level of the factor for each algorithm is shown in table 2.

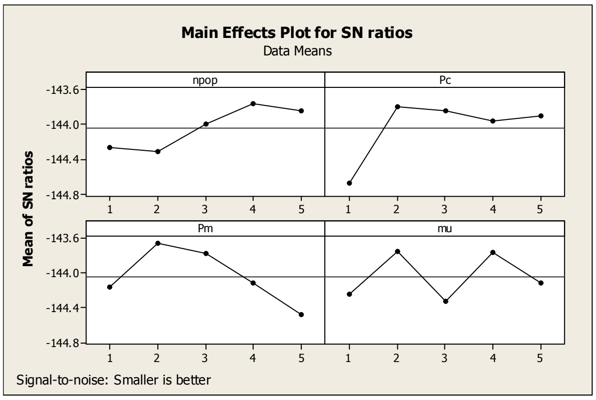

Fig. 8 The signal-to-noise ratios for Genetic Algorithm

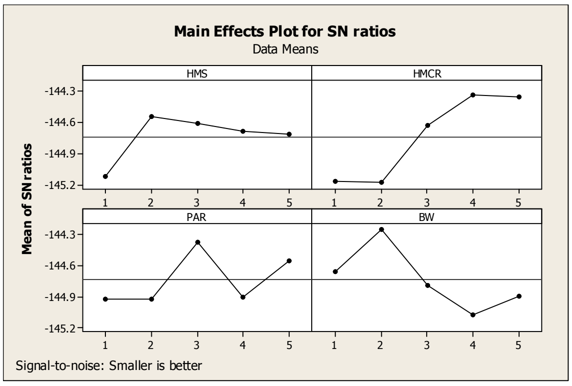

Fig. 9 The signal-to-noise ratios for Harmony search

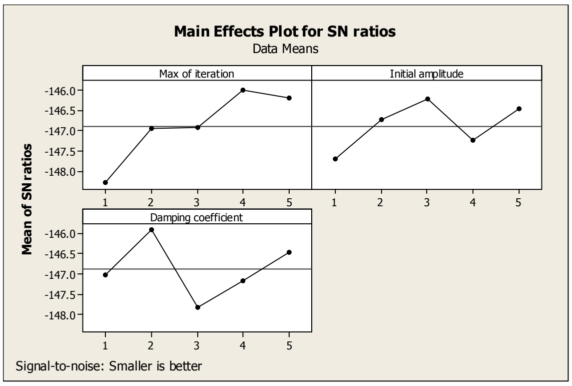

Fig. 10 The signal-to-noise ratios for Vibration damping optimization

Table 2 Best level for parameters

Factors	Algorithms	Notations	Values
Population Size		npop	100
Crossover Percentage	GA	Pc	0.45
Mutation		Pm	0.5

Factors	Algorithms	Notations	Values
Percentage Strongly mutation Rate		mu	0.026
Harmony memory size		HMS	10
Harmony memory considering rate	HS	HMCR	0.85
Pitch-adjusting rate		PAR	0.2
Bandwidth		BW	0.5
Max of iteration at each amplitude		sub-it	20
Damping coefficient	VDO	γ	6
Initial amplitude		A_0	0.05

6.2 Computational results

Computational experiments are conducted to validate and verify the behavior and the performance of the genetic algorithm, harmony search algorithm and vibration damping optimization algorithm to solve the aggregate production planning model with preventive maintenance. In order to evaluate the performance of the meta-heuristic algorithms, 30 test problems with different sizes are randomly generated. These test problems are classified into three classes: small size, medium size and large size aggregate production planning problems. The number of products, machines and periods has the most impact on problem hardness. The approaches are implemented to solve each instance in five times to obtain more reliable data. Table 3 shows details of computational results obtained by solution methods for all test problems.

Table 3 Details of computational results for all test problems

NO	Prob. size	i.j.k.l.t	Lingo	Time(s)	GA	Time(s)	HS	Time(s)	VDO	Time(s)
1		2.1.2.1.3	6720124	1	6720124	59.4	6720124	10.6	6720124	12.7
2		2.1.2.2.3	7093831	1	7093831	148.4	7093831	18.2	7093831	29.2
3		2.1.3.2.3	7345570	1	7345570	527.8	7345570	32.6	7345570	1013
4		2.1.4.1.3	7585061	1	7585061	1269.6	7585061	824	7585061	1987
5	**Small**	2.2.2.1.3	7594855	3	7594855	96.3	7594855	13.7	7594855	18.8
6		2.1.2.1.4	8522935	3	8522935	72	8522935	6.5	8522935	23.7
7		2.2.2.1.4	9939956	4	9939956	62.9	9939956	15.4	9939956	48.6
8		2.1.2.1.6	13931320	6	15898457.8	76.8	14142746	31.2	14119881	32.4
9		2.1.3.1.4	10185920	7	10599753	146.7	10525717.8	75.1	10386292.2	152.5
10		2.2.2.1.5	11858890	28	12836809.6	113.2	12088009.2	25.8	12022934.6	54
11		2.1.3.2.4	11042530	31	11577865.2	153.5	11420786	392.5	11210416	24.6
12		2.1.2.2.5	12824550	172	15151918.4	72.3	13627122.4	74	13531169.4	63.8
13		2.1.2.2.6	15105320	1035	16870590	110.7	16394869.4	37.7	15176125.8	96
14	**Medium**	2.2.2.2.6	16202530	2002	21703160.4	213.1	17340630.2	76.2	16560223.4	61.9
15		4.1.2.1.3	---	---	16684098.2	57.8	13750161	51	12435647	113.4
16		3.1.2.1.5	---	---	22298122	150.9	17321010	108.5	15010929	94.4
17		4.1.2.1.5	---	---	32939551.2	187.8	24351300.8	357.1	21996169.2	159.4
18		2.1.4.1.5	---	---	17290025.4	1217.3	13170435.2	2213	11823412	2830
19		3.1.2.1.6	---	---	33937574.8	199.1	27788070.6	105.9	24912403.4	345.6
20		4.1.2.1.6	---	---	46863855	185.9	36965194.4	493.5	30281600.2	197.6
21		2.1.3.2.6	---	---	22836542	779.7	19765159.2	1439.7	11412775.2	2681.2
22		2.1.2.1.8	---	---	34025305.8	139.8	30847836.2	149.8	18458592	235.2

NO	Prob. size	i.j.k.l.t	Lingo	Time(s)	GA	Time(s)	HS	Time(s)	VDO	Time(s)
23		2.1.2.2.8	---	---	31403459.8	124.5	29067560.4	88.8	15299469.4	81.1
24	**Large**	2.2.2.1.8	---	---	35099474.2	138.3	31525137.6	114.5	19876372.6	42
25		2.1.2.1.12	---	---	72024715	475.5	63658096	118.3	36973604.8	89.7
26		2.1.2.2.12	---	---	74636612.8	547.9	63231299.6	312.5	42977358.8	112.6
27		3.1.2.1.12	---	---	112933333	780.1	91273845	676.1	68783215.8	601.3
28		2.1.2.1.16	---	---	107898484.4	493.5	83846588.6	120.2	50431013.2	703.5
29		2.1.2.2.16	---	---	103650367.4	605.8	76633081	214	56096091.6	811.4
30		2.2.2.1.16	---	---	104470567	600	85097596.6	226.1	62294712.8	947.1

--- Means that a feasible and optimum solution has not been found after 3600 s of computing time.

$$p_{k1t} \in [20,24], o_{k1t} \in [22,27], c_{k1t} \in [70,77], h_{k1t} \in [40,45], h_{k1t} \in [40,45], a_{k1l} = 1, u_{k1l} = 0.1,$$
$$r_{k1lt} \in [4,7], R'_{lt} \in [21000,40000], hr'_t \in [200,480], l'_t \in [200,480], w'_t \in [60,65], I_{k10} = 500,$$
$$w'_0 = 3500, B_{k10} = 0, e_{kl} = 0.2, \alpha'_t = 0.2, \beta'_{lt} = 0.5, w'_{\max t} \in [3000,7000], D_{i2t} \in [6000,24000],$$
$$p_{i2t} \in [20,25], o_{i2t} \in [22,27], c_{i2t} \in [100,106], h_{i2t} \in [60,67], a_{i2j} \in [0.4,0.5], u_{i2j} = 0.2,$$
$$r_{i2jt} \in [10,15], R_{jt} \in [21000,40000], hr_t \in [200,460], l_t \in [200,460], w_t \in [61,64], I_{i20} = 500,$$
$$w_0 = 3500, B_{i20} = 0, e_{i2} = 0.4, \alpha_t = 0.2, \beta_{jt} \in [0.4,0.5], f \in [120,190], w_{\max t} \in [3000,7000],$$
$$C_{\max it} \in [2000,9500], f_{ik} = 2, C1_{l1t} \in [100000,220000], C2_{l1t} \in [10000,50000],$$
$$C3_{j2t} \in [100000,220000], C4_{j2t} \in [10000,50000], C5_{i2t} \in [11,14], C6_{i2t} \in [4,7],$$
$$MTS_{jt} \in [1500,5000], MTF_{lt} \in [1500,5000], TR_{i2t} \in [300,800], XD_{\max i2t} \in [300,600],$$
$$XR_{\max i2t} \in [400,650], hX_{i2t} \in [60,65];, m = 0.1, LT = 1.$$

The presented statistical analysis (the variance analysis outcome) were reported for problems with small, medium, and large dimensions between algorithms, in tables 4 to12 and figures 11 to 19 , which according to the values of the survey (or *P-Value*), we can chose the better algorithm with use ANOVA related:

- ✓ The objective values obtained by GA, HS and VDO are close to each other for small dimensions problems.
- ✓ The objective values obtained by GA, HS, VDO are no different from each other in the medium dimensions test problems.
- ✓ The objective values obtained by GA and HS are no different from each other in the large dimensions test problems.
- ✓ The objective values obtained by VDO are better from GA and HS results for large dimensions test problems.

Also Figure 20, depict comparison between solution quality of the GA, HS and VDO of the instances:

- ✓ The GA, HS and VDO can find good quality solutions for small dimensions problems.
- ✓ The GA, HS and VDO algorithms can solve all the test problems.
- ✓ The objective values obtained by VDO and HS are close to each other for medium size problems.
- ✓ For small dimensions test problems, the GA and HS can find good quality solutions but, its results will be worse when the problem size increases.

We can reach the conclusion that the VDO has shown its usefulness in large dimensions problems as compared to the GA and HS.

Table 4 Analysis of variance for test problems with small size, between GA and HS

Source	SS	DF	MS	F_0	P
Small size	3.32445E+11	1	3.32445E+11	0.04	0.835
Error	1.34538E+14	18	7.47432E+12		
Total	1.34870E+14	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N    Mean    StDev    ---+---------+---------+---------+------
GA     10  9413735  2977943      (--------------*--------------)
HS     10  9155881  2465867    (--------------*--------------)
                              ---+---------+---------+---------+------
                            8000000   9000000  10000000  11000000
```

Fig. 11 The output of analysis of variance with small size, between GA and HS

Table 5 Analysis of variance for test problems with small size, between GA and VDO

Source	SS	DF	MS	F_0	P
Small size	3.93657E+11	1	3.93657E+11	0.05	0.820
Error	1.33565E+14	18	7.42029E+12		
Total	1.33959E+14	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N    Mean    StDev    ----+---------+---------+---------+-----
GA     10  9413735  2977943       (--------------*--------------)
VDO    10  9133144  2443856    (--------------*--------------)
                              ----+---------+---------+---------+-----
                            8000000   9000000  10000000  11000000
```

Fig. 12 The output of analysis of variance with small size, between GA and VDO

Table 6 Analysis of variance for test problems with small size, between HS and VDO

Source	SS	DF	MS	F_0	P
Small size	2584746709	1	2584746709	0.00	0.984
Error	1.08476E+14	18	6.02647E+12		
Total	1.08479E+14	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N    Mean    StDev   ---------+---------+---------+---------+
HS     10  9155881  2465867   (-----------------*-----------------)
VDO    10  9133144  2443856  (------------------*------------------)
                             ---------+---------+---------+---------+
                               8400000   9100000   9800000  10500000
```

Fig.13. The output of analysis of variance with small size, between HS and VDO

Table 7 Analysis of variance for test problems with medium size, between GA and HS

Source	SS	DF	MS	F_0	P
Medium size	9.32566E+13	1	9.32566E+13	1.00	0.330
Error	1.67776E+15	18	9.32086E+13		
Total	1.77101E+15	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N      Mean     StDev  -----+---------+---------+---------+----
GA     10  23531676  11001909             (------------*------------)
HS     10  19212958   8085498    (------------*------------)
                               -----+---------+---------+---------+----
                             16000000  20000000  24000000  28000000
```

Fig. 14 The output of analysis of variance with medium size, between GA and HS

Table 8 Analysis of variance for test problems with medium size, between GA and VDO

Source	SS	DF	MS	F_0	P
Medium size	1.94555E+14	1	1.94555E+14	2.41	0.138
Error	1.45314E+15	18	8.07301E+13		
Total	1.64770E+15	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N      Mean     StDev  -----+---------+---------+---------+----
GA     10  23531676  11001909              (---------*---------)
VDO    10  17293810   6357535    (---------*--------)
                               -----+---------+---------+---------+----
                             15000000  20000000  25000000  30000000
```

Fig. 15 The output of analysis of variance with medium size, between GA and VDO

Table 9 Analysis of variance for test problems with medium size, between HS and VDO

Source	SS	DF	MS	F_0	P
Medium size	1.84157E+13	1	1.84157E+13	0.35	0.563
Error	9.52142E+14	18	5.28968E+13		
Total	9.70557E+14	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N      Mean     StDev  -------+---------+---------+---------+--
HS     10  19212958  8085498          (---------------*---------------)
VDO    10  17293810  6357535   (---------------*---------------)
                               -------+---------+---------+---------+--
                               15000000  17500000  20000000  22500000
```

Fig.16 The output of analysis of variance with medium size, between HS and VDO

Table 10 Analysis of variance for test problems with large size, between GA and HS

Source	SS	DF	MS	F_0	P
Large size	7.69205E+14	1	7.69205E+14	0.75	0.398
Error	1.85026E+16	18	1.02792E+15		
Total	1.92718E+16	19			

Individual 90% CIs For Mean Based on Pooled StDev

```
Level   N      Mean     StDev  -------+---------+---------+---------+--
GA     10  69897886  36288867           (--------------*--------------)
HS     10  57494620  27183819  (--------------*--------------)
                               -------+---------+---------+---------+--
                               48000000  60000000  72000000  84000000
```

Fig.17 The output of analysis of variance with large size, between GA and HS

Table 11 Analysis of variance for test problems with large size, between GA and VDO

Source	SS	DF	MS	F_0	P
Large size	5.00468E+15	1	5.00468E+15	5.69	0.028
Error	1.58295E+16	18	8.79414E+14		
Total	2.08341E+16	19			

```
Individual 90% CIs For Mean Based on Pooled StDev
Level   N      Mean     StDev   --+---------+---------+---------+-------
GA     10  69897886  36288867                      (-------*-------)
VDO    10  32360321  16701891     (-------*-------)
                                --+---------+---------+---------+-------
                          20000000  40000000  60000000  80000000
```

Fig. 18 The output of analysis of variance with large size, between GA and VDO

Table 12 Analysis of variance for test problems with large size, between HS and VDO

Source	SS	DF	MS	F_0	P
Large size	1.84979E+15	1	1.84979E+15	3.13	0.094
Error	1.06282E+16	18	5.90453E+14		
Total	1.24779E+16	19			

```
Individual 90% CIs For Mean Based on Pooled StDev
Level   N      Mean     StDev  -------+---------+---------+---------+--
HS     10  57494620  27183819                    (-------*--------)
VDO    10  32360321  16701891  (--------*-------)
                               -------+---------+---------+---------+--
                               30000000  45000000  60000000  75000000
```

Fig. 19 The output of analysis of variance with large size, between HS and VDO

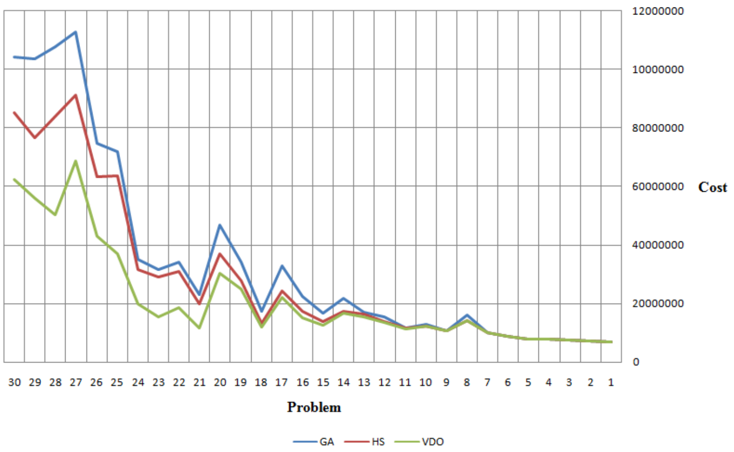

Fig. 20 Comparison between solution quality of the GA, HS and VDO

7 Conclusion

This paper is concentrated on multi-period, multi-product, multi-machine, two stage systems, setup decisions, return products and preventive maintenance. We have developed a mixed integer linear programming model that can be used to compute optimal solutions for the problems by an operation research solver. Due to the complexity of the problem, three meta-heuristics algorithms named genetic algorithm (GA), harmony search (HS) algorithm and vibration damping optimization (VDO) algorithm were used to solve the problem. Moreover, an extensive parameter setting with performing the Taguchi method was conducted for selecting the optimal levels of the factors that impactalgorithm's performance. The computational results show that VDO the algorithm obtain good solutions for APP with PM problem. One straightforward opportunity for future research is extending the assumption of the proposed model for including real conditions of production systems such as uncertainty return products, uncertainty PM, etc. Also, developing new meta-heuristic algorithms to make better solutions can be suggested.

References

1. Masud, A. S. M., Hwang, C. L., (1980). An aggregate production planning model and application of three multiple objective decision methods. International Journal of Production Research, Vol. 18, pp. 741-752.
2. Lee, C. Y., (1997). Minimizing the makespan in the two machines flowshop scheduling problem with an availability constraint. Operations Research Letters, Vol. 20, pp. 129-139.
3. Leung, S. C. H., Wu, Y., Lai, K. K,. (2003). Multi-Site Aggregate Production Planning with Multiple Objectives: a Goal Programming Approach. Production Planning & Control, Vol. 14, pp. 425-436.
4. Wang, R. C., Liang, T. F., (2005). Applying possibilistic linear programming to aggregate production planning", International Journal of Production Economics, Vol. 98, pp.328–341.
5. Leung, S. C. H., Chan, S. S. W., (2009). A goal programming model for aggregate production planning with resource utilization constraint. Computers & Industrial Engineering, Vol. 56, pp. 1053-1064.
6. Ramezanian, R., Rahmani, D., Barzinpour, F., (2012). An aggregate production planning model for two phase production systems: Solving with genetic algorithm and tabu search. Expert Systems with Applications, pp. 1256–1263.
7. Sadeghi, M. R., Razavi Hajiagha, S. H., Hashemi, S. S., (2013). A fuzzy grey goal programming approach for aggregate production planning. Int J Adv Manuf Technol, Vol. 64. pp. 1715–1727.
8. Wang, S. C., Yeh, M. F. (2014). A modified particle swarm optimization for aggregate production Planning. Expert Systems with Applications, Vol. 41, pp. 3069-3077.
9. Aghezzaf, E. H., Jamali, M. A., Ait-Kadi, D., (2007). An Integrated Production and Preventive Maintenance Planning Model. European Journal of Operational Research, Vol. 181, pp. 679-685.
10. Adiri, I., Bruno, J., Frostig, E., Rinnooy Kan, A. H. G., (1989). Single machine down-time scheduling with a single breakdown. Acta Informatica, Vol. 26, pp. 679-696.
11. Wienstein, L., Chung, C. H. (1999). Integrated maintenance and production decisions in a hierarchical production planning environment. Computer and Operations Research, Vol. 26, pp. 1059-1074.
12. Lee, C. Y., Chen, Z. L., (2000). Scheduling jobs and maintenance activities on parallel machines. Naval Research Logistics, Vol. 47, pp. 145-165.
13. Aghezzaf, E. H., Najid, N. M., (2008). Integrated Production Planning and Preventive Maintenance in Deteriorating Production Systems. Information Sciences, Vol. 178, pp. 3382-3392.
14. Pan, E., Liao, W., Xi, L., (2010). Single-machine-based production scheduling model integrated preventive maintenance planning. The International Journal of Advanced Manufacturing Technology, Vol. 50. pp. 365–75.
15. Nourelfath, M., Chatelet, E., (2012). Integrating production, inventory and maintenance planning for a parallel system with dependent components. Reliability Engineering and System Safety, Vol. 101, pp. 59-66.
16. Yalaoui, A., Chaabi, C., Yalaoui, F., (2014). Integrated production planning and preventive maintenance in deteriorating production systems, Information Sciences. In Press Corrected Proof.
17. Mitchell, M., (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press.
18. Lee, K. S., Geem, Z. W., (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanical Engineering, Vol. 194. pp. 3902-3933.
19. Mahdavi, M., Fesanghary, M., Damangir, E., (2007). An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation, Vol. 188, pp. 1567-1579.

20. Mehdizadeh, E., Tavakkoli-Moghaddam, R., (2009). Vibration damping optimization algorithm for an identical parallel machine scheduling problem. Proceeding of the 2nd International Conference of Iranian Operations Research Society, Babolsar, Iran.
21. Taguchi, G., Chowdhury, S., (2000). Taguchi, Robust Engineering. McGraw-Hill, New York.