Expert Systems with Applications 39 (2012) 1256–1263

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An aggregate production planning model for two phase production systems: Solving with genetic algorithm and tabu search

Reza Ramezanian *, Donya Rahmani, Farnaz Barzinpour

Department of Industrial Engineering, Iran University of Science & Technology, P.C. 1684613114, Tehran, Iran

ARTICLE INFO

Keywords:
Aggregate production planning
Mathematical programming
Genetic algorithm
Tabu search

ABSTRACT

Aggregate production planning (APP) is a medium-term capacity planning to determine the quantity of production, inventory and work force levels to satisfy fluctuating demand over a planning horizon. The goal is to minimize costs and instabilities in the work force and inventory levels. This paper is concentrated on multi-period, multi-product and multi-machine systems with setup decisions. In this study, we develop a mixed integer linear programming (MILP) model for general two-phase aggregate production planning systems. Due to NP-hard class of APP, we implement a genetic algorithm and tabu search for solving this problem. The computational results show that these proposed algorithms obtain good-quality solutions for APP and could be efficient for large scale problems.

1. Introduction

Aggregate production planning is medium-term capacity planning often from 3 to 18 months ahead. It is concerned with the lowest-cost method of production planning to meet customer's requirements and to satisfy fluctuating demand over the planning horizon.

A survey of models and methodologies for APP has been represented by Nam and Ogendar (1992). Some researchers have used a hierarchical approach for production planning that called hierarchical production planning (HPP) (Ari & Axsater, 1988; Axsater, 1986; Bitran, Haas, & Hax, 1982). Also, the multi criteria decision making (MCDM) approach has been used for production planning (Masud & Hwang, 1980; Tabucanon & Majumdar, 1989).

Nowadays, meta-heuristic methods are used to solve NP-hard problems and due to NP-hard class of aggregate production planning, these approaches have been used for solving APP (Fahimnia, Luong, & Marian, 2006; Jiang, Kong, & Li, 2008). Researchers have used fuzzy approach with genetic algorithm to formulate and solve APP (Aliev, Fazlollahi, Guirimov, & Aliev, 2007; Hsu & Lin, 1999). Other methods such as hybrid algorithms (Ganesh & Punniyamoorthy, 2005; Mohan Kumar & Noorul Haq, 2005) and tabu search algorithm (Baykasogluy, 2006; Pradenas & Pe~nailillo, 2004) have been implemented to solve APP. But these presented methods are generality concentrated on the solution algorithm but not on a general model. On the other hand, the consideration of the all parameters in an APP model makes it more difficult. So researchers have not presented a comprehensive and general model to formulate real production environments. The majority of models in the APP are relevant to single product and single stage systems and they are not compatible to real production systems. In this paper a general and comprehensive aggregate production planning model is represented and is solved by meta-heuristic approaches.

This paper considers a multi-period, multi-product multi-machine and two-phase system in which involves setup costs and setup times. If a specific product is produced in a period then each required machine must be set up exactly once in that period. Since there is setup decisions in this system so we must formulate this model as a mixed integer programming (MIP) problem (Hung & Hu, 1998).

The rest of this paper is organized as follows. In Section 2, the proposed aggregate production planning model is demonstrated. In Sections 3 and 4, genetic algorithm and tabu search for solving the problem are described. In Section 5, the computational results are given and in last section we present our conclusion.

2. Aggregate production planning model

In this section, a proposed MILP for APP is presented. This model is relevant to multi-period, multi-product, multi-machine and two-phase production systems. At first phase, the individual pieces are produced by first groups of workers and machines; we call these pieces, first-phase products. At the next stage, the first-phase products and other purchased products are assembled into aggregate products by second groups of workers and machines. We call them second-phase products (Fig. 1).

There are several features which are involved in the model such as setup decisions and lead time. If a specific product is produced

* Corresponding author. Tel.: +98 9125397764.
E-mail addresses: Ramezanian@iust.ac.ir (R. Ramezanian), drahmani@iust.ac.ir (D. Rahmani), Barzinpour@iust.ac.ir (F. Barzinpour).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.07.134

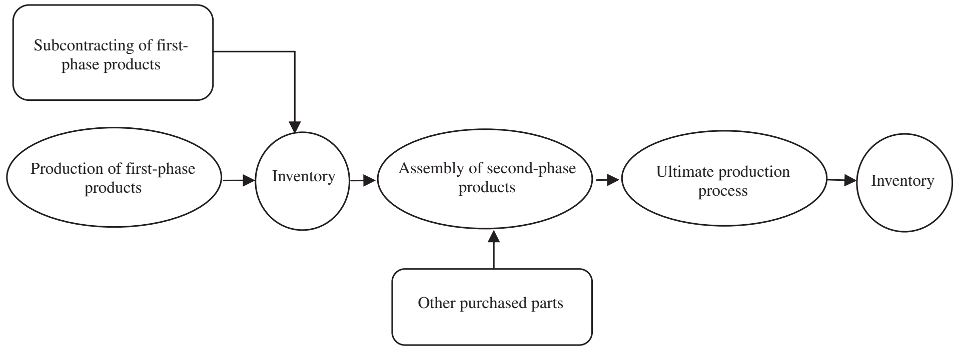

Fig. 1. The two phase production system.

in a period then each required machine must be set up exactly once in that period. The assumptions of the model are as follows:

- Setup times and setup costs are considered.
- Setup times are independent on jobs sequence.
- Machines are available at all times.
- All programming parameters have deterministic value and there is no randomness.

2.1. Model variables

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).
I_{k1t}: The inventory of first-phase product k in period t (units).
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 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).
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).
C_{k10}: Subcontracting volume of first-phase product k in the beginning of planning horizon (units).
y_{k1t}: The setup decision variable of first-phase product k in period t, a binary integer variable.
y_{i2t}: The setup decision variable of second-phase product i in period t, a binary integer variable.

2.2. Model parameters

D_{i2t}: Forecasted demand of product i in period t (units).
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).
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 ($/units).
b_{i2t}: Backorder cost of second-phase product i in period t ($/units).
h_{i2t}: Inventory cost of second-phase product i ($/units).
a_{i2j}: Hours of machine j per unit of second-phase product i (machine-days/unit).
a_{k1l}: Hours of machine l per unit of first-phase product i (machine-days/unit).
u_{i2j}: The setup time for second-phase product i on machine j (hours).
u_{k1l}: The setup time for first-phase product i on machine l (hours).
r_{i2jt}: The setup cost of second-phase product i on machine j in period t ($/machine-hours).
r_{k1lt}: The setup cost of first-phase product i on machine l in period t ($/machine-hours).
R_{jt}: The regular time capacity of machine j in period t (machine-hours).
R'_{lt}: The regular time capacity of machine l in period t (machine-hours).
hr_t: Cost to hire one worker in period t for second group labor ($/man-days).
l_t: Cost to layoff one worker of second group in period t ($/man-days).
hr'_t: Cost to hire one worker in period t for first group labor ($/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).
w'_t: The second group labor cost in period t ($/man-days).
I_{i20}: The initial inventory level of second-phase product i in period t (units).
I'_{k10}: The initial inventory level of first-phase product i in period t (units).
w_0: The initial first group workforce level (man-days).
w'_0: The initial first group workforce level (man-days).
B_{i20}: The initial first group workforce level (man-days).
f_i: The number of unit of first-phase product k required per unit of first-phase product i.

e_{k1}: Hours of labor per unit of second-phase product i (man-days/unit).
e_{i2}: 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.
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.
β'_{llt}: The ratio of regular time capacity of machine l available for use in overtime in period t.
f: The working hours of labor in each period (man-hour/man-day).
W_{tmax}: Maximum level of first group labor available in period t (man-days).
W'_{tmax}: Maximum level of second group labor available in period t (man-days).
C_{i2tmax}: Maximum subcontracted volume available of second-phase product i in period t (units).
L: Lead time.
M: A large number.

2.3. Proposed mixed integer linear programming model

At first we define the parts of objective function:

Production cost and subcontracting cost of first-phase and second-phase products:

$$\sum_{i=1}^{N}\sum_{t=1}^{T}(p_{k1t}P_{k1t}+o_{k1t}O_{k1t}+c_{k1t}C_{k1t})+\sum_{i=1}^{N}\sum_{t=1}^{T}(p_{i2t}P_{i2t}+o_{i2t}O_{i2t}+c_{i2t}C_{i2t}) \tag{1}$$

Set up cost for first-phase and second-phase products on corresponding machines:

$$\sum_{l=1}^{L}\sum_{t=1}^{T}\sum_{k=1}^{K}r_{k1lt}y_{k1t}+\sum_{\mathbf{t}=1}^{\mathbf{T}}\sum_{\mathbf{i}=1}^{\mathbf{N}}\sum_{\mathbf{j}=1}^{\mathbf{J}}\mathbf{r_{i2jt}y_{i2t}} \tag{2}$$

Inventory cost of first-phase and second-phase products:

$$\sum_{t=1}^{T}\sum_{k=1}^{K}h_{k1t}I_{k1t}+\sum_{t=1}^{T}\sum_{i=1}^{N}h_{i2t}I_{i2t} \tag{3}$$

Back order cost of second-phase products:

$$\sum_{i=1}^{N}\sum_{t=1}^{T}b_{i2t}B_{i2t} \tag{4}$$

Total Work force and Hiring and Layoff cost:

$$\sum_{t=1}^{T}hr_tH_t+l_tL_t+w_tW_t+hr'_tH'_t+l'_tL'_t+w'_tW'_t \tag{5}$$

Then, the proposed MILP model is as follow:

$$\begin{aligned}\text{Min}&\sum_{i=1}^{N}\sum_{t=1}^{T}(p_{i2t}P_{i2t}+o_{i2t}O_{i2t}+c_{i2t}C_{i2t})+\sum_{k=1}^{K}\sum_{t=1}^{T}(p_{k1t}P_{k1t}+o_{k1t}O_{k1t}\\&+c_{k1t}C_{k1t})+\sum_{t=1}^{T}\sum_{k=1}^{K}h_{k1t}I_{k1t}+\sum_{t=1}^{T}\sum_{i=1}^{N}h_{i2t}I_{i2t}+\sum_{l=1}^{L}\sum_{t=1}^{T}\sum_{k=1}^{N}r_{k1lt}y_{k1t}\\&+\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{J}r_{i2jt}y_{i2t}+\sum_{i=1}^{N}\sum_{t=1}^{T}b_{i2t}B_{i2t}+\sum_{t=1}^{T}(hr_tH_t+l_tL_t)\\&+\sum_{t=1}^{T}w_tW_t+\sum_{t=1}^{T}(hr'_tH'_t+l'_tL'_t)+\sum_{t=1}^{T}w'_tW'_t\end{aligned} \tag{6}$$

$$P_{i2t}+O_{i2t}+C_{i2t}+B_{i2t}-B_{i2t-1}+I_{i2t-1}-I_{i2t}=D_{i2t};\quad i=1,\ldots,N,\quad t=1,2,\ldots,T \tag{7}$$

$$P_{k1t}+O_{k1t}+C_{k1t}+I_{k1t-1}-I_{k1t}=\sum_{i=1}^{N}f_{ik}(P_{i2t+rmL}+O_{i2t+L});\quad k=1,2,\ldots,K,\quad t=1,2,\ldots,T \tag{8}$$

$$C_{k10}+I'_{k10}-I_{k10}=\sum_{i=1}^{N}\sum_{t=1}^{L}f_{ik}(P_{i2t+L}+O_{i2t+L});\quad k=1,2,\ldots,K \tag{9}$$

$$\sum_{i=1}^{N}(a_{i2j}P_{i2t}+U_{i2j}y_{i2t})\leqslant R_{jt};\quad j=1,2,\ldots,J,\quad t=1,2,\ldots,T \tag{10}$$

$$\sum_{i=1}^{N}(a_{i2j}O_{i2t})\leqslant \beta_{jt}\cdot R_{jt};\quad j=1,2,\ldots,J,\quad t=1,2,\ldots,T \tag{11}$$

$$\sum_{k=1}^{K}(a_{k1l}P_{k1t}+U_{k1l}\cdot y_{k1t})\leqslant R'_{lt};\quad l=1,2,\ldots,L,\quad t=1,2,\ldots,T \tag{12}$$

$$\sum_{k=1}^{K}(a_{k1l}O_{k1t})\leqslant \beta_{lt}\cdot R'_{lt};\quad l=1,2,\ldots,L,\quad t=1,2,\ldots,T \tag{13}$$

$$P_{k1t}+O_{k1t}\leqslant My_{k1t};\quad k=1,2,\ldots,K,\quad t=1,2,\ldots,T \tag{14}$$

$$P_{i2t}+O_{i2t}\leqslant My_{i2t};\quad i=1,2,\ldots,N,\quad t=1,2,\ldots,T \tag{15}$$

$$W_t=W_{t-1}+H_t-L_t;\quad t=1,2,\ldots,T \tag{16}$$

$$W'_t=W'_{t-1}+H'_t-L'_t;\quad t=1,2,\ldots,T \tag{17}$$

$$\sum_{i=1}^{N}e_{k1}P_{k1t}\leqslant fW'_t;\quad t=1,2,\ldots,T \tag{18}$$

$$\sum_{i=1}^{N}e_{k1}O_{k1t}\leqslant \alpha'_tfW'_t;\quad t=1,2,\ldots,T \tag{19}$$

$$\sum_{i=1}^{N}e_{i2}P_{i2t}\leqslant fW_t;\quad t=1,2,\ldots,T \tag{20}$$

$$\sum_{i=1}^{N}e_{i2}O_{i2t}\leqslant \alpha_tfW_t;\quad t=1,2,\ldots,T \tag{21}$$

$$W_t\leqslant W_{tmax};\quad t=1,2,\ldots,T \tag{22}$$

$$W'_t\leqslant W'_{tmax};\quad t=1,2,\ldots,T \tag{23}$$

$$C_{i2t}\leqslant C_{i2tmax};\quad i=1,2,\ldots,N,\quad t=1,2,\ldots,T \tag{24}$$

$$B_{i2t}\cdot I_{i2t}=0;\quad i=1,2,\ldots,N,\quad t=1,2,\ldots,T \tag{25}$$

$$y_{i2t}=\{0,1\};\quad i=1,2,\ldots,N,\quad t=1,2,\ldots,T \tag{26}$$

$$y_{k1t}=\{0,1\};\quad k=1,2,\ldots,K,\quad t=1,2,\ldots,T \tag{27}$$

Constraint (7) is relevant to market demand for second phase products. Constraint (8) is relevant to production and subcontract of first phase products that associated to total production of second phase products. Constraint (9) certifies that the initial inventory level and the Subcontracting volume of first-phase products in the beginning of planning horizon should be equal or greater than the total production of second phase products at the first L periods to satisfy the market demand. Constraint (10) ensures that the quantity of regular time production do not exceed the available first group machines capacity, we also include setup times in this machine capacity constraint. Constraint (11) is relevant to over time machine capacity constraint. Also Constraints (12) and (13) are relevant to first group machines capacity. We mention that the first group machines are necessary for production of first-phase products and second group are necessary to produce second-phase products. Constraints (14) and (15) are relevant to the setup costs and setup times in this model for first-phase products and second-phase products respectively.

Constraints (16) and (17) are relevant to workforce level for the both groups of workers. Constraints (18)–(21) imply workforce capacity constraints at regular time and overtime at each period for the both groups of workers. Naturally in order to minimizing the objective function, the constraint (25) is not necessary and we can ignore it.

3. The genetic algorithm

Genetic algorithm has been proven to be powerful method for combinatorial optimization problems. 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

In this paper, each gene is total aggregate production (TAP) of second-phase products and a chromosome is a production plan and each row of the chromosome formed as an integer vector with T genes as shown in Fig. 2, where T is the number of periods.

To decompose the TAP to the regular time production, overtime production and subcontracting volume, we propose a heuristic method. This method is detailed as follow.

3.1.1. Calculation of production, subcontracting, hiring and layoff variables

The regular time production, overtime production and subcontracting variables for each product at each period and hiring and layoff variables at each period are obtained as the following procedure.

3.1.1.1. Production amount of second-phase products. For each period, the second-phase products are sorted ascending based on regular time production cost of them:

$$\begin{aligned} &L_i = p_{it} \\ &L_{[1]} \leqslant L_{[2]} \leqslant L_{[3]} \leqslant \cdots \leqslant L_{[N]} \end{aligned} \tag{28}$$

The regular time production amount of second-phase products is calculated by the following equation based on obtained sorted products.

$$\begin{aligned} P_{[i]t} = \min\Big\{ &\max\Big\{0, \frac{W_{tmax}\cdot f - \sum_{l<i} e_{[l]}\cdot P_{[l]t}}{e_{[i]}}\Big\}, \\ &\times \max\Big\{0, \frac{R_{Jtmax} - \sum_{l<i} a_{[l]j}\cdot P_{[l]t}}{a_{[i]j}}\Big\}, \max\{0, R_{[i]t}\}\Big\} \end{aligned} \tag{29}$$

Thereafter, for each period, the second-phase products are sorted ascending based on overtime production cost of them:

$$\begin{aligned} &Q_i = o_{it} \\ &Q_{[1]} \leqslant Q_{[2]} \leqslant Q_{[3]} \leqslant \cdots \leqslant Q_{[N]} \end{aligned} \tag{30}$$

The overtime production amount of second-phase products is calculated by the following equation based on obtained sorted products.

$$\begin{aligned} O_{[i]t} = \min\left\{ \vphantom{\sum_{l<i}} \right. &\max\left\{0, \frac{W_{tmax}\cdot f\cdot \alpha_t - \sum\limits_{l<i} e_{[l]}\cdot O_{[l]t}}{e_{[i]}}\right\}, \\ &\left. \times \max\left\{0, \frac{R_{Jtmax}\cdot \beta_{Jt} - \sum\limits_{l<i} a_{[l]j}\cdot O_{[l]t}}{a_{[i]j}}\right\}, \max\{0, R_{[i]t} - P_{[i]t}\}\right\} \end{aligned} \tag{31}$$

Finally, subcontracting volume of second-phase products is obtained by using the following equation:

$$C_{[i]t} = \min\{C_{itmax}, \max\{0, R_{[i]t} - (P_{[i]t} + O_{[i]t})\}\} \tag{32}$$

Now, total aggregate production for second-phase products are:

$$TP_t = P_{it} + O_{it} + C_{it} \tag{33}$$

3.1.1.2. Production amount of first-phase products. For each period, the first-phase products are sorted ascending based on regular time production cost of them:

$$\begin{aligned} &M_k = p_{kt} \\ &M_{[1]} \leqslant M_{[2]} \leqslant M_{[3]} \leqslant \cdots \leqslant M_{[K]} \end{aligned} \tag{34}$$

The regular time production amount of first-phase products is calculated from the following equation based on obtained sorted products.

$$\begin{aligned} P_{[k]t} = \min\left\{ \vphantom{\sum_{l<k}} \right. &\max\left\{0, \frac{W'_{tmax}\cdot f - \sum\limits_{l<k} e'_{[l]}\cdot P_{[l]t}}{e'_{[k]}}\right\}, \\ &\left. \times \max\left\{0, \frac{R'_{Jtmax} - \sum\limits_{l<i} a'_{[l]j}\cdot P_{[l]t}}{a'_{[k]j}}\right\}, \sum_{i=1}^{N} f_{i[k]}(P_{it} + O_{it})\right\} \end{aligned} \tag{35}$$

Period (t)	1	2	...	T-1	T
Total aggregate production for second-phase product 1: V_1(t)	TP_{11}	TP_{12}	...	$TP_{1(T-1)}$	TP_{1T}
Total aggregate production for second-phase product 2: V_2(t)	TP_{21}	TP_{22}	...	$TP_{2(T-1)}$	TP_{2T}
...			...		
Total aggregate production for second-phase product N: V_N(t)	TP_{N1}	TP_{N2}	...	$TP_{N(T-1)}$	TP_{NT}

Fig. 2. Chromosome representation (production plan for second-phase products) for GA.

Thereafter, for each period, the second-phase products are sorted ascending based on overtime production cost of them:

$$\begin{aligned} & N_k = o_{kt} \\ & N_{[1]} \leqslant N_{[3]} \leqslant N_{[3]} \leqslant \cdots \leqslant N_{[K]} \end{aligned} \tag{36}$$

The overtime production amount of first-phase products is calculated from the following equation based on obtained sorted products.

$$\begin{aligned} O_{[k]t} = \min \Bigg\{ & \max \left\{ 0, \frac{W'_{tmax} \cdot f \cdot \alpha'_t - \sum_{l<k} e'_{[l]} \cdot O_{[l]t}}{e'_{[k]}} \right\} \\ & \times \max \left\{ 0, \frac{R'_{Jtmax} \cdot \beta'_t - \sum_{l<k} a_{[l]j} \cdot O_{[l]t}}{a_{[k]j}} \right\} \\ & \times \max \left\{ 0, \left(\sum_{i=1}^{N} f_{i[k]} (P_{it} + O_{it}) - P_{[k]t} \right) \right\} \Bigg\} \end{aligned} \tag{37}$$

Finally, Subcontracting volume of first-phase products are obtained by using the following equation:

$$C_{[i]t} = \max \left\{ 0, \sum_{i=1}^{N} f_{i[k]} (P_{it} + O_{it}) - (O_{[k]t} + P_{[k]t}) \right\} \tag{38}$$

To obtain other variables we use the following equations set:

$$\begin{aligned} & W_t = \sum_{i=1}^{N} e_{[i]} \cdot P_{it} \\ & W'_t = \sum_{k=1}^{K} e'_{[k]} \cdot P_{kt} \\ & W_t = W_t + H_t - L_t \\ & W'_t = W'_{t-1} + H'_t - L'_t \\ & TP_t + B^s_{it} - B^s_{it-1} + I^s_{it-1} - I^s_{it} = D^s_{it} \\ & P_{kt} + O_{kt} + I'_{kt-1} - I'_{kt} = f_{ik} * (P_{it} + O_{it}) \end{aligned} \tag{39}$$

3.2. Initial population

The initial solutions are generated by using two procedures. First total aggregate production for each period is considered equal demand of that period. Second procedure is based on the following equation:

$$TP_{it} \in [\lambda D_{it}, (1+\lambda) D_{it}]; \quad \lambda \in (0,1) \tag{40}$$

The diversity of the initial population can be maintained because some production plans are generated randomly.

3.3. 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 roulette wheel selection is employed.

3.4. The genetic operators

3.4.1. Reproduction

The best chromosomes which have a lower fitness function are chosen. This mechanism just copies the chosen chromosomes to the next generation.

3.4.2. 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, to explore solution space an arithmetic crossover is chosen.

- **Arithmetic crossover (AC)**: Arithmetic crossover generates an offspring by linear combining two selective parents as shown

Parent 1	TAP for second-phase product 1	TP_{11}	TP_{12}	TP_{13}	TP_{14}	TP_{15}	TP_{16}	TP_{17}	TP_{18}
	TAP for second-phase product 2	TP_{21}	TP_{22}	TP_{23}	TP_{24}	TP_{25}	TP_{26}	TP_{27}	TP_{28}
Offspring	TAP for second-phase product 1	η_{11}	η_{12}	η_{13}	η_{14}	η_{15}	η_{16}	η_{17}	η_{18}
	TAP for second-phase product 2	η_{21}	η_{22}	η_{23}	η_{24}	η_{25}	η_{26}	η_{27}	η_{28}
Parent 2	TAP for second-phase product 1	TP'_{11}	TP'_{12}	TP'_{13}	TP'_{14}	TP'_{15}	TP'_{16}	TP'_{17}	TP'_{18}
	TAP for second-phase product 2	TP'_{21}	TP'_{22}	TP'_{23}	TP'_{24}	TP'_{25}	TP'_{26}	TP'_{27}	TP'_{28}

Fig. 3. Illustration of the arithmetic crossover.

Parent	TAP for second-phase product 1	TP_{11}	TP_{12}	TP_{13}	TP_{14}	TP_{15}	TP_{16}	TP_{17}	TP_{18}
	TAP for second-phase product 2	TP_{21}	TP_{22}	TP_{23}	TP_{24}	TP_{25}	TP_{26}	TP_{27}	TP_{28}
Offspring	TAP for second-phase product 1	TP_{11}	TP_{12}-Δ	TP_{13}+Δ	TP_{14}	TP_{15}	TP_{16}	TP_{17}	TP_{18}
	TAP for second-phase product 2	TP_{21}	TP_{22}	TP_{23}	TP_{24}-Δ	TP_{25}	TP_{26}+Δ	TP_{27}	TP_{28}

Fig. 4. Illustration of the arithmetic mutation.

Parent	TAP for second-phase product 1	TP_{11}	TP_{12}	TP_{13}	TP_{14}	TP_{15}	TP_{16}	TP_{17}	TP_{18}
	TAP for second-phase product 2	TP_{21}	TP_{22}	TP_{23}	TP_{24}	TP_{25}	TP_{26}	TP_{27}	TP_{28}
Offspring	TAP for second-phase product 1	TP_{11}	TP_{12}	TP_{17}	TP_{14}	TP_{15}	TP_{16}	TP_{13}	TP_{18}
	TAP for second-phase product 2	TP_{21}	TP_{22}	TP_{23}	TP_{28}	TP_{25}	TP_{26}	TP_{27}	TP_{24}

Fig. 5. Illustration of the exchange mutation.

in Fig. 3. The principle of this operator is based on the following equation:

$$\eta_{it} = \lambda TP_{it} + (1-\lambda)TP'_{it}; \quad \lambda \in (0,1) \tag{41}$$

3.4.3. Mutation

Mutation generates an offspring solution by randomly modifying the parent's feature. It helps to keep a reasonable level of diversity in population, and serves the search by jumping out of local optimal solutions. In this research an arithmetic mutation is chosen and discussed in detail as follows:

- **Arithmetic mutation (AM)**: A mutation scheme that reduces the total aggregate production level for a random selective period by the amount of Δ and then it is added to other selective period at each row of current solution as shown in Fig. 4.
- **Exchange mutation (EM)**: A mutation scheme that swaps the value of the two random selected genes of current solution together as shown in Fig. 5.

In our implemented GA, crossover and mutation operators are used each with the given probabilities.

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 number of generations is greater than maximum number of generations, a priori fixed constant or the some specified number of generations without improvement of best known solution is reached.

A sensitivity analysis has been done by varying the parameters and the values were fixed and listed in Table 1.

4. Tabu search (TS)

The TS method is iterative, neighborhood based search method. In this technique, at each iteration, a move is performed to the best solution in the neighborhood of actual one. To avoid cycling and to escape from local optimum, the short-term memory of visited solutions is introduced and called the tabu list. In the tabu list attributes of some number of recently visited solutions are stored. The move is tabu (omitted in the search process) if it arrives to the solution already visited (Glover, 1989a; Glover, 1989b). In the following, we present our implementation of TS.

4.1. Initial solution

The solution that via it the total aggregate production for each period is equal demand of that period is considered as an initial solution.

Table 1
Parameters in implemented GA.

Parameter	Value
Population size	100
Probability of arithmetic crossover	0.8
Probability of exchange mutation	0.04
Probability of arithmetic mutation	0.09
Probability of reproduction	0.07
Number of generations	100

Table 2
Comparison of LINGO8, GA and TS results.

Prob.	N	J	K	L	T	LINGO			GA				TS			
						LB	Obj. func.	Time (S)	Obj. func.	% Dev. LB	% Dev. obj.	Time (S)	Obj. func.	% Dev. LB	% Dev. obj.	Time (S)
1	2	1	2	1	3	1,152,780	1,156,460	1800	1,241,159	7.7	7.3	0.9	1,243,959	7.9	7.6	0.5
2	2	1	3	1	3	2,622,010	2,622,010	617*	2,874,897	9.6	9.6	1.1	2,875,108	9.7	9.6	0.6
3	2	1	4	1	3	6,097,720	6,104,010	1800	6,535,294	7.2	7.1	1.2	6,614,062	8.5	8.4	0.79
4	2	1	2	1	6	12,910,500	13,302,600	1800	13,033,916	1.0	−2.0	1.9	13,040,681	1.0	−2.0	2.1
5	2	1	3	1	6	15,626,400	15,638,400	1800	15,667,076	0.3	0.2	2.3	15,720,746	0.6	0.5	2.8
6	2	1	3	2	3	3,724,630	3,728,280	1800	3,743,059	0.5	0.4	1.5	3,789,197	1.7	1.6	0.9
7	2	1	3	2	6	9,774,750	12,188,100	1800	10,286,664	5.2	−15.6	2.5	10,575,351	8.2	−13.2	3.2

4.2. Tabu list

In the tabu list, criterion values of maximum number of recently visited solutions are stored. The list is initialized with empty elements. Newly added element replaces the oldest one. Index of mutated periods of each solution is stored in the tabu list. The move is tabu if the criterion value of solution is equal to the one of the stored in the tabu list. For our experiments we accepted the tabu list size with $TL = min\{N \times T, 50\}$.

4.3. The aspiration level criteria

It is possible to overcome the tabu restriction for a pair at a given iteration if the exchange given an objective function value strictly better than the best obtained so far.

4.4. Neighborhood search scheme

Since the time required for evaluating the entire neighborhood increases very fast with the increase of the problem size, we examine only part of the neighborhood. In our algorithm we evaluate only $min\{N \times T, 50\}$ neighbors. The neighborhood scheme is the same as mutation operators which defined previously for the GA algorithm.

4.5. Stopping condition

Two criteria are used as stoppage rules: First the search process stops if the number of iterations is greater than maximum number of iterations, a priori fixed constant. Second 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 $Max - Iteration = 100$.

5. Computational results

For the two phase aggregate production planning we did not find benchmarks in the literature. In order to evaluate the performance of the meta-heuristic algorithms, 17 test problems with different sizes are randomly generated. Furthermore, for the small-sized instances of two phases APP, LINGO optimization solver is used to figure out the optimal solution and compared with the corresponding GA and TS results.

The genetic algorithm and tabu search are coded in MATLAB R2007(b) and all tests are conducted on a Laptop Computer at Core Due 2 GHz with 1.0 GB of RAM.

To compare the lower bound (LB) and objective values obtained by LINGO with the results of the GA and TS, a quality measure, the percent deviation of solution, is defined according to the following equations set:

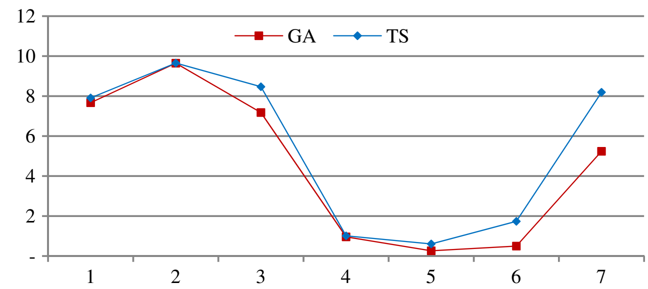

Fig. 6. Percent deviance of objective values obtained by GA and TS with lower bound of LINGO.

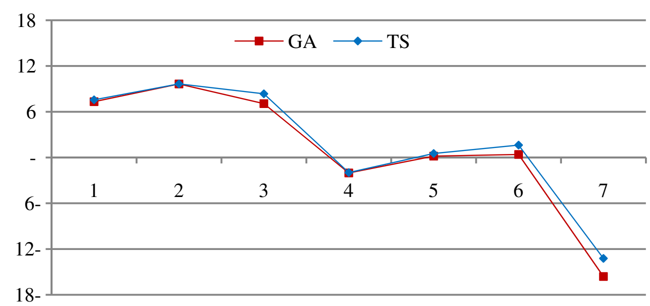

Fig. 7. Percent deviance of objective values obtained by GA and TS with LINGO.

$$\begin{aligned} \%Deviation_{LB} &= \frac{OF_{GA\ \ or\ \ TS} - LB_{LINGO}}{LB_{LINGO}} \times 100 \\ \%Deviation_{OF} &= \frac{OF_{GA\ \ or\ \ TS} - OF_{LINGO}}{OF_{LINGO}} \times 100 \end{aligned} \tag{42}$$

The comparison for small-sized instances between implemented GA, TS and LINGO is presented in Table 2. LINGO solver is run for 30 min. As the results show, the solution gaps with LB vary from 0.3% to 9.7% for GA and TS. The presented GA and TS provide result close to solution obtained through LINGO. The computational times also are much less than LINGO.

The performance of the GA and TS are illustrated in Figs. 6 and 7, with respect to the solution deviance with the lower bound (LB) and objective values obtained by LINGO. As is illustrated in the figures, the quality of solutions obtained by genetic algorithm is better than the tabu search results for small-sized problems.

For large-sized instances, Table 3 gives the best, the mean objective function and average relative percent deviation (ARPD)

Table 3
Results of 10 independent runs for the GA and TS.

Prob.	N	J	K	L	T	GA				TS			
						Best (10^6)	Ave (10^6)	Time (S)	ARPD	Best (10^6)	Ave (10^6)	Time (S)	ARPD
1	5	2	5	2	6	71.34	71.59	41.12	0.36	71.60	71.81	27.76	0.66
2	5	2	5	2	9	156.67	157.14	69.65	0.32	156.64	157.16	43.57	0.34
3	5	2	5	5	6	77.50	77.68	49.74	0.23	77.85	77.91	36.08	0.53
4	5	2	10	2	6	81.35	81.68	96.77	0.40	81.92	82.00	46.16	0.79
5	5	2	10	2	9	174.48	175.10	143.78	0.35	174.57	175.24	70.93	0.43
6	5	2	10	5	6	71.62	72.05	134.17	0.6	72.16	72.40	65.12	1.09
7	5	2	5	2	12	246.28	246.87	104.83	0.35	246.02	246.95	71.28	0.38
8	5	2	5	5	12	241.59	242.95	116.68	0.56	242.11	243.02	86.26	0.59
9	5	2	10	2	12	269.99	270.84	188.22	0.55	269.36	270.55	111.23	0.44
10	5	2	10	5	12	273.24	274.37	250.87	0.41	273.69	274.42	156.49	0.43
Average						166.41	167.03	119.58	0.413	166.59	167.15	71.49	0.568

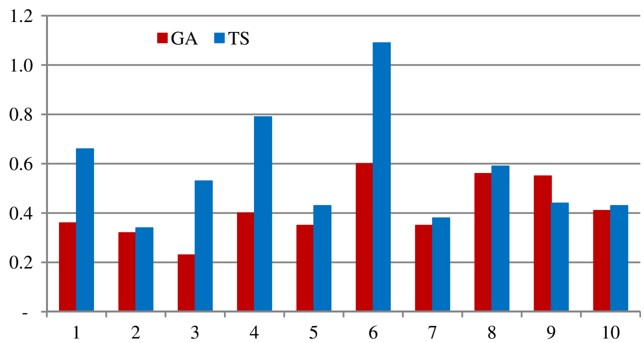

Fig. 8. The average relative percent deviation (ARPD) for the GA and TS.

of 10 independent runs for each instance, obtained by the GA and TS.

Let OF_{min} be the minimum objective function (total cost) among two algorithms. The relative percentage deviation (RPD) is defined according to the following equation:

$$RPD = \frac{OF_{algorithm} - OF_{best}}{OF_{best}} \times 100 \tag{43}$$

where, $OF_{\text{algorithm}}$ is the expected total cost attained for a given algorithm and instance. RPD helps us compare the performance of the algorithms because the RPD values denote the relative distance of the solution from the best solution obtained for special instance. As is illustrated in the Table 3 and Fig. 8, the quality of solutions obtained by GA is better than the TS results, but the computational times are greater than TS.

6. Conclusion

The purpose of this paper is to formulate and solve aggregate production planning model for two phase production systems in which the objective function is to minimize the costs of production over the planning horizon. This paper is concentrated on multi-period, multi-product, multi-machine and two stage systems with setup decisions. We develop a mixed integer linear programming model that can be used to compute optimal solution for the problems by an operation research solver. We presented genetic algorithm and tabu search for solving this problem. To verify the effectiveness of the presented approaches, computational experiments are performed on a set of random small-sized instances by LINGO. Due to NP-hard class of APP, the computational results show that these implemented algorithms obtain good solutions for APP within a reasonable computational time.

References

Aliev, R. A., Fazlollahi, B., Guirimov, B. G., & Aliev, R. R. (2007). Fuzzy-genetic approach to aggregate production–distribution planning in supply chain management. *Information Sciences, 177*, 4241–4255.

Ari, E. A., & Axsater, S. (1988). Disaggregation under uncertainty in hierarchical production planning. *European Journal of Operational Research, 35*, 182–186.

Axsater, S. (1986). On the feasibility of aggregate production plans. *Operations Research, 34*, 796–800.

Baykasogluy, A. (2006). MOAPPS 1.0: Aggregate production planning using the multipleobjective tabu search. *International Journal of production Research, 39*(16), 3685–3702.

Bitran, G. R., Haas, E. A., & Hax, A. C. (1982). Hierarchical production planning: A two stage system. *Operations Research, 29*, 232–251.

Fahimnia, B., Luong, L.H.S., & Marian, R.M. (2006). Modeling and optimization of aggregate production planning – A genetic algorithm. In:*Proceedings of world academy of science, Engineering and technology*.

Ganesh, K., & Punniyamoorthy, M. (2005). Optimization of continuous-time production planning using hybrid genetic algorithms-simulated annealing. *International Journal of Advanced Manufacturing Technology, 26*, 148–154.

Glover, F. (1989a). Tabu search. Part I. *ORSA Journal on Computing, 1*, 190–206.

Glover, F. (1989b). Tabu search. Part II. *ORSA Journal on Computing, 2*, 4–32.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Hsu, H-M., & Lin, Y-J. (1999). Fuzzy logic control of aggregate production planning using genetic algorithms. *International Journal of Industrial Engineering: Theory Applications and Practice, 6*(4), 325–333.

Hung, Y. F., & Hu, Y. C. (1998). Solving mixed integer programming production planning problems with setups by shadow price information. *Computers and Operations Research, 25*, 1027–1042.

Jiang, G., Kong, J., & Li, G. (2008). Aggregate Production Planning Model of Production Line in Iron and Steel Enterprise Based on Genetic Algorithm. Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China.

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, 18*, 741–752.

Mohan Kumar, G., & Noorul Haq, A. (2005). Hybrid genetic-ant colony algorithms for solving aggregate production plan. *Journal of Advanced Manufacturing Systems (JAMS), 1*, 103–111.

Nam, S. J., & Ogendar, N. R. (1992). Aggregate production planning–a survey of models and methodologies. *European Journal of Operational Research, 61*, 255–272.

Pradenas, L., & Peñailillo, F. (2004). Aggregate production planning problem: A new algorithm. *Electronic Notes in Discrete Mathematics, 18*, 193–199.

Tabucanon, M. T., & Majumdar, S. (1989). *Production planning in a ship repair company in Bangladesh. Proceedings of the International Conference on MCDM: Application in Industry and Service*. Bangkok: Asian Institute of Technology.