Simulated annealing algorithm for solving an aggregate production planning model with product returns

Hossein Alipour

Assistant Professor, Industrial Eng. Dept., Khalkhal Branch, Islamic Azad University, Khalkhal, Iran

Abstract

In this paper, we develop a mixed integer linear programming (MILP) model for aggregate production planning system with product returns. These returned products can either be disposed or be remanufactured to be sold as new ones again; hence the market demands can be satisfied by either newly produced products or remanufactured ones. The capacities of production, disposal and remanufacturing are limited. Due to NP-hard class of APP, we implement a simulated annealing (SA). Additionally, Taguchi method is conducted to calibrate the parameter of the meta-heuristics and select the optimal levels of the algorithm's performance influential factors.

Keywords: Aggregate Production Planning, Product Returns, Taguchi method

1. Introduction

The Aggregate Production Planning (APP) is a schedule of the organization's overall operations over a planning horizon to satisfy demand while minimizing costs. 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. Masud and Hwang 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(Masud and Hwang, 1980). Nam and Logendran represented a classification scheme that categorizes the literature on APP since early 1950, summarizing the various existing techniques into a framework depending upon their ability to either produce an exact optimal or near-optimal solution (Nam and Logendran, 1992). Wang and Liang presented a novel interactive possibilistic linear programming (PLP) approach for solving the multi-product aggregate production planning (APP) problem with imprecise forecast demand, related operating costs, and capacity (Wang and Liang, 2005). Silva et al. presented an aggregate production planning (APP) model applied to a Portuguese firm that produces construction materials (Silva et al., 2006). Ramezanian et al. developed a mixed integer linear programming (MILP) model for general two-phase aggregate production planning systems. They used a genetic algorithm and tabu search for solving this problem (Ramezanian et al., 2012). Karmarkar and Rajaram considered a competitive version of the traditional aggregate production planning model with capacity constraints (Karmarkar and Rajaram, 2012). Zhang et al. proposed a mixed integer linear programming (MILP) model to mathematically characterize the problem of aggregate production planning (APP) with capacity expansion in a manufacturing system including multiple activity centers. They used the heuristic based on capacity shifting with linear relaxation to solve the model (Zhang et al., 2012). Raa et al. presented a mixed integer linear programming formulations for the aggregate production–distribution problem for a manufacturer of plastic products that are produced using injection moulding, and a matheuristic solution approach based on these models (Raa et al., 2013). Golany et al. studied a production planning problem with remanufacturing. They proved the problem is NP-complete and obtain an O(T3) algorithm for solve the problem (Golany et al., 2001). Teunter and Bayındır addressed the dynamic lot-sizing problem for systems with product returns. They presented an exact, polynomial time dynamic programming algorithm (Teunter et al., 2006). Pan et al. addressed the capacitated dynamic lot-sizing problem arising in closed-loop supply chain where returned products are collected from customers. They assumed that the capacities of production, disposal and remanufacturing are limited, and backlogging is not allowed. Moreover, they proposed a pseudo-polynomial algorithm for solving the problem with both capacitated disposal and remanufacturing (Pan et al., 2009). Love and Turner used of stochastic optimal control in solving aggregate production scheduling problems and compared this approach with deterministic approaches to the problem (Love and Turner, 1993). Pradenas and Peñailillo introduced a mathematical model and a heuristic procedure based on Tabu Search for the problem of Aggregate Production Planning at a sawmill to determine the volumes of different products with different tree trunk types and using different cut schemes (Pradenas and Peñailillo, 2004). Mohankumar and Noorul haq proposed hybrid algorithm that combines genetic algorithm and ant colony algorithm for solving the aggregate production plan problem (Mohankumar and Noorul haq, 2005). Fahimnia et al. presented a methodology to model the Aggregate Production Planning problem, which is combinatorial in nature, when optimized with Genetic Algorithms (Fahimnia et al., 2008). Hashem et al. developed a stochastic programming approach to solve a multi-period multi-product multi-site aggregate production planning problem in a green supply chain for a medium-term planning horizon under the assumption of demand uncertainty (Hashem et al., 2013). Wang and Yeh presented a integer linear programming model of aggregate production planning (APP) from a manufacturer of gardening equipment. Also they proposed a modified PSO (MPSO) algorithm for solving the problem (Wang and Yeh, 2014).

2. Problem formulation

In this section, we present an MILP formulation of the problem. This model is relevant to multi-period, multi-product, multi-machine.

2.1. Assumptions

- The quantity of inventory and shortage at the beginning of the planning horizon are zero.
- The quantity of shortage at the end of the planning horizon is zero.
- Machines are available at all times.

2.2. Model Variables

P_{it}: Regular time production of product i in period t (units).

O_{it}: Over time production of product i in period t (units).

C_{it}: Subcontracting volume of product i in period t (units).

B_{it}: Backorder level of product i in period t (units).

I_{it}: The inventory of product i in period t (units).

H_t: The number of workers hired in period t (man-days).

L_t: The number of workers laid off in period t (man-days).

W_t: workforce level in period t (man-days).

y_{it}: The setup decision variable of product i in period t, a binary integer variable.

XR_{it}: The number of returned products of product i that remanufactured in period t

XRI_{it}: The number of returned products of product i held that in inventory at the end of period t

XD_{it}: The number of returned products of product i that disposed in period t

2.3. Parameters

D_{it}: Forecasted demand of product i in period t (units).

p_{it}: Regular time production cost of product i in period t (\$/units).

o_{it}: Over time production cost of product i in period t (\$/units).

c_{it}: Subcontracting cost of product i in period t (\$/units).

h_{it}: Inventory cost of product i in period t (\$/units).

a_{ij}: Hours of machine j per unit of product i (machine-days/unit).

u_{ij}: The setup time for product i on machine j (hours).

r_{ijt}: The setup cost of 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 labor (\$/man-days).

l_t: Cost to layoff one worker in period t (\$/man-days).

w_t: The labor cost in period t (\$/man-days).

I_{i0}: The initial inventory level of product i (units).

w_0: The initial workforce level (man-days).

B_{i0}: The initial backorder level of product i (man-days).

e_i: Hours of labor per unit of product i (man-days/unit).

α_t: The ratio of regular-time of workforce available for use in overtime in period t.

Q_{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/manday).

W_{maxt}: Maximum level of labor available in period t (man-days).

C_{maxit}: Maximum subcontracted volume available of product i in period t (units).

TR_{it}: the number of returned products of product i in period t.

$XDmax_{it}$: The maximum number of returned products of product i that could be disposed in period t.

$XRmax_{it}$: The maximum number of returned products of product i that could be remanufactured in period t.

hX_{it}: Inventory cost of returned products of product *i* in period *t* ($/units)

$C5_{it}$: The cost of returned products of product i that disposed in period t.

$C6_{it}$: The cost of returned products of product i that remanufactured in period t.

M: A large number.

hX_{it}: Inventory cost of returned products of product i in period t($/units).

2.4. The Proposed Model

$$MinZ=\sum_{i=1}^{N}\sum_{t=1}^{T}(p_{it}P_{it}+o_{it}O_{it}+c_{it}C_{it})+\sum_{t=1}^{T}\sum_{i=1}^{N}h_{it}I_{it}+\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{J}r_{ijt}y_{it}+\sum_{i=1}^{N}\sum_{t=1}^{T}b_{it}B_{it}+ \tag{1}$$

$$\sum_{t=1}^{T}(hr_tH_t+l_tL_t)+\sum_{t=1}^{T}w_tW_t+\sum_{i=1}^{N}\sum_{t=1}^{T}C5_{it}XD_{it}+\sum_{i=1}^{N}\sum_{t=1}^{T}C6_{it}XR_{it}+\sum_{i=1}^{N}\sum_{t=1}^{T}hX_{it}XRI_{it}$$

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

$$\sum_{i=1}^{N}(a_{ij}P_{it}+U_{ij}y_{it})\le R_{jt};\ t=1,2,...,T \quad j=1,2,...,J \tag{3}$$

$$\sum_{i=1}^{N}a_{ij}O_{it}\le Q_{jt}R_{jt};\quad t=1,2,...,T \quad j=1,2,...,J \tag{4}$$

$$P_{it}+O_{it}\le My_{it};\ i=1,2,...,N \quad t=1,2,...,T \tag{5}$$

$$W_t=W_{t-1}+H_t-L_t;\quad t=1,2,...,T \tag{6}$$

$$\sum_{i=1}^{N}e_iP_{it}\le fw_t;\ t=1,2,...,T \tag{7}$$

$$\sum_{i=1}^{N}e_iO_{it}\le \alpha_t fw_t;\ t=1,2,...,T \tag{8}$$

$$w_t\le w_{\max t};\ t=1,2,...,T \tag{9}$$

$$C_{it}\le C_{\max it};\ i=1,2,...,N \quad t=1,2,...,T \tag{10}$$

$$B_{it}.I_{it}=0;\ i=1,2,...,N \quad t=1,2,...,T \tag{11}$$

$$XR_{it}=XRI_{i,t-1}-XD_{it}-XR_{it}+TR_{it};i=1,2,...,N \quad t=1,2,...,T \tag{12}$$

$$XD_{it}\le XD\max_{it};\ i=1,2,...,N \quad t=1,2,...,T \tag{13}$$

$$XR_{it}\le XR\max_{it};\ i=1,2,...,N \quad t=1,2,...,T \tag{14}$$

$B_{iT} = 0;\ i = 1, 2, ..., N$ (15)

$y_{it} = \{0,1\};\ i = 1, 2, ..., N \quad t = 1, 2, ..., T$ (16)

$P_{it}, Y_{it}, W_t, H_t, L_t, B_{it}, XD_{it}, XR_{it}, XRI_{it} \geq 0$ *and integer* $t \in I, t \in i$ (17)

3. Simulated Annealing Algorithm

Simulated annealing (SA) was initially presented by Kirkpatrick, et al. (Kirkpatrick, et al., 1983). The SA methodology draws its analogy from the annealing process of solids. In the annealing process, a solid is heated to a high temperature and gradually cooled to a low temperature to be crystallized. As the heating process allows the atoms to move randomly, if the cooling is done too rapidly, it gives the atoms enough time to align themselves in order to reach a minimum energy state that named stability or equipment. This analogy can be used in combinatorial optimization in which the state of solid corresponds to the feasible solution, the energy at each state corresponds to the improvement in the objective function and the minimum energy state will be the optimal solution.

The steps of SA algorithm are shown in below(Mehdizadeh and Fatehi, 2014):

Step 1: Generating feasible initial solution.

$X_{best} = X_0$

Step 2: Initializing the algorithm parameters which consist of initial temperatures (T_0), rate of the decrement current temperature (α), max of iteration at each temperature (L), freezing temperature (T_f), in this paper $T_f = 0$.

Step 3: Calculating the objective value $C(X_0)$ for initial solution.

Step 4: Initializing the internal loop

In this step, the internal loop is carried out for S =1 and will be repeated while S < L.

Step 5: Neighborhood generation

Step 6: Accepting the new solution

Set $\Delta C = (C(X_n) - C(X))/C(X)$

Now,if $\Delta C \leq 0$, accept the new solution, else if $\Delta C > 0$ generate a random number r between (0, 1);

If $r < 1 - e^{\left(\frac{-\Delta C}{T_0}\right)}$, then accept a new solution; otherwise, reject the new solution and accept the previous solution.

If $S \geq L$, go to step 7; otherwise $S + 1 \rightarrow S$ and go back to step 5

Step 7: Adjusting the temperature

In this step, $T_0 = T \times \alpha$ is used for reducing temperature at each iteration of the outer cycle of the algorithm. If $T_0 = T_f$ return to step 8; otherwise, go back to step 4.

Step 8: Stopping criteria.

Three important issues that need to be defined when adopting this general algorithm to a specific problem are the procedures to generate both initial solution and neighboring solutions.

3.1. Representation Schema

To design simulated annealing optimization algorithm for mentioned problem, a suitable representation scheme that shows the solution characteristics is needed. In this paper, the general structure of the solution representation performed for running the simulated annealing for four periods with two products is shown in Figure 1.

Y11	Y12	Y13	Y14	Y21	Y22	Y23	Y24
0	1	1	0	1	1	0	0

Figure 1 Solution representation

3.2. Neighborhood Scheme

At each temperature level a search process is applied to explore the neighborhoods of the current solution. In this paper we use swap scheme, Figure 2 illustrates this operation on the four periods with two products.

Y11	Y12	Y13	Y14	Y21	Y22	Y23	Y24
0	1	1	0	1	1	0	0

Y13	Y12	Y11	Y14	Y23	Y22	Y21	Y24
1	1	0	0	0	1	1	0

Figure 2 Illustration of swap

3.3. Cooling schedule scheme

The temperature is another basic characteristic of the SA which is gradually decreased when the algorithm progressed. Initially, T is set to a high value, T_i, and it can be reduced with some patterns at each step of algorithm. The cooling schedule with $T_i = \alpha \times T_{i-1}$ (where α is the cooling factor constant $\alpha \in (0, 1)$) is considered as cooling pattern for this research.

4. RESULTS

In this paper, all tests are conducted on a not book with Intel Core 2 Duo Processor 2.00 GHz and 2 GB of RAM and the proposed algorithm namely SA are coded in MATLAB R2011(a). Moreover the proposed model are coded with LINGO 8 software and using LINGO software for solving the instances.

4.1. Parameter Calibration

Appropriate design of parameters has significant impact on efficiency of meta-heuristic. In this paper the Taguchi method applied to calibrate the parameters of the proposed method namely SA algorithm. The Taguchi method was developed by Taguchi (Taguchi and Chowdhury, 2000).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 S/N 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 S/N ratio, the highest S/N ratio provides the optimum level for that factor. As mentioned before, the purpose of Taguchi method is to maximize the S/N ratio. In this subsection, the parameters for experimental analysis are determined.

Table 1 lists different levels of the factors for SA. In this paper according to the levels and the number of the factors, respectively the Taguchi method L_9 is used for the adjustment of the parameters for the SA. Best Level of the factor for algorithm is shown in table 2.

TABLE 1 Factors and their levels

Factor	Algorithm	Level	Value
Max Number of Sub-iteration(l)			10, 15, 20
Initial Temp(T_0)	SA	3	800, 1000, 1200
rate of the decrement current temperature(α)			0.9, 0.95, 0.99

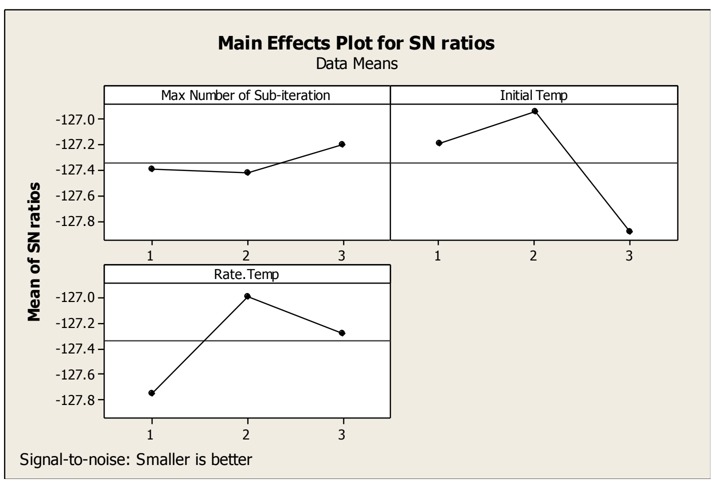

Figure 3 The SN ratios for Simulated Annealing Algorithm

TABLE 2 best level for parameters

Factor	Algorithm	Value
Max Number of Sub-iteration(l)		20
Initial Temp(T_0)	SA	1000
rate of the decrement current temperature(α)		0.95

4.2. Computational Results

Computational experiments are conducted to validate and verify the behavior and the performance of the simulated annealing algorithm to solve the aggregate production planning model. In order to evaluate the performance of the meta-heuristic algorithm, 20 test problems with different sizes are randomly generated. 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 method for all test problems.

TABLE 3 Details of computational results for all test problem

NO	Problem size (i.j.t)	LINGO		SA	
		O.F.V	T(Second)	O.F.V	T(Second)
1	2.1.6	4689067	0.46	4802139.4	52.9
2	2.1.12	8023814	0.92	8384220.4	118.2
3	3.3.6	6714547	1.56	7025394	152.2
4	3.8.8	11506630	69	11854930.8	813.1
5	3.8.12	16688330	223	16869008	1575.4
6	4.4.4	6759658	315	6767710.4	392
7	3.8.16	21934270	1287	22137458.8	2231.2
8	4.3.8	13268500	2575	13319176.6	529.6
9	4.2.12	---	---	18702552.2	708.1
10	4.2.16	---	---	24573790.2	641.1
11	4.3.12	---	---	17849891.2	1438.1
12	4.3.16	---	---	25166474	1933.2
13	4.4.12	---	---	18228164.2	1325.3
14	4.6.4	---	---	11375675.8	539.5
15	4.6.8	---	---	15351615.2	1659.2
16	6.3.8	---	---	23362537.8	3018.3
17	6.3.12	---	---	50082131.8	3097
18	6.4.4	---	---	11128577	2650
19	6.4.8	---	---	26440668.6	3156
20	8.2.5	---	---	18340220	3376

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

$D_{it} \in [6000, 24000]; p_{it} \in [20, 24]; o_{it} \in [22, 27]; c_{it} \in [100, 106]; h_{it} \in [60, 67]; a_{ij} \in [0.4, 0.5];$
$r_{ijt} \in [10, 15]; R_{jt} \in [21000, 40000]; hr_t \in [200, 460]; l_t \in [200, 460]; w_t \in [61, 64]; hX_{it} \in [60, 65];$
$Q_{jt} \in [0.4, 0.5]; f \in [120, 190]; W_{\max t} \in [3000, 7000]; C_{\max it} \in [2000, 9500]; TR_{it} \in [300, 800];$
$XD\max_{it} \in [300, 600]; XR\max_{it} \in [400, 650]; C5_{it} \in [11, 14]; C6_{it} \in [4, 7]; u_{ij} = 0.2;$
$e_i = 0.4; \alpha_t = 0.2$

5. Conclusion

In this paper, we deal with aggregate production planning problem of a multi-period, multi-product, multi-machine subject to capacity constraints in closed loop supply chain which the objective function is to minimize the costs of production over the planning horizon. We develop a mixed integer linear programming model that can be used to compute optimal solution for the problems by an operation research solver. Due to the complexity of the problem, simulated annealing algorithm used to solve the problem. Additionally, an extensive parameter setting with performing the Taguchi method was conducted for selecting the optimal levels of the factors that effect on algorithm's performance. The computational results show that increasing the number of product i and machine j have a significant impact in increasing the CPU time.

One straightforward opportunity for future research is extending the assumption of the proposed model for including real conditions of production systems such as uncertainty demands, etc.

Also, developing new heuristic or meta-heuristic algorithms to make better solutions can be suggested.

References

[1]. Fahimnia, B. Luong, L.H.S. Marian, R.M. (2008). Modeling and Optimization of Aggregate Production Planning - A Genetic Algorithm Approach. World Academy of Science, Engineering and Technology, 2, 900–905.

[2]. Golany, B. Yang, J. Yu, G. (2001). Economic lot-sizing with remanufacturing options. IIE Transactions, 33, 995-1003.

[3]. Hashem, M.A. Baboli, A. Sazvar, Z. (2013). A stochastic aggregate production planning model in a green supply chain: Considering flexible lead times, nonlinear purchase and shortage cost functions. European Journal of Operational Research, 230, 26–41.

[4]. Karmarkar, S. Rajaram, k. (2012). Aggregate production planning for process industries under oligopolistic competition", European Journal of Operational Research, 223, 680–689.

[5]. Kirkpatrick, S. Gelatt, C. Vecchi, M. (1983). Optimization by simulated annealing. Science, 220, 671-680.

[6]. Love, C.E. Turner, M. (1993). Note on utilizing stochastic optimal control in aggregate production planning. European Journal of Operational Research, 65, 199–206.

[7]. 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.

[8]. Mehdizadeh, E. Fatehi Kivi, A. (2014). Three Meta-heuristic Algorithms for the Single-item Capacitated Lot-sizing Problem. International Journal of Engineering, Transaction B: Applications, 27, 1223–1232.

[9]. Mohankumar, G. Noorul haq, A. (2005). Hybrid genetic ant colony algorithms for solving aggregate production plan. Journal of Advanced Manufacturing Systems, 4, 103–111.

[10]. Nam, S. Logendran, R. (1992). Aggregate production planning — A survey of models and methodologies. European Journal of Operational Research, 61, 255–272.

[11]. Pan, Z. Tang, J. Liu, O. (2009). Capacitated dynamic lot sizing problems in closed-loop supply chain. European Journal of Operational Research, 198, 810–821.

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

[13]. Raa, B. Dullaert, W. Aghezzaf, E.H. (2013). A matheuristic for aggregate production–distribution planning with mould sharing. International Journal of Production Economics, 145, 29–37.

[14]. 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, 39, 1256-1263.

[15]. Silva, C.G.D. Figueira, J. Lisboa, J. (2006). An interactive decision support system for an aggregate production planning model based on multiple criteria mixed integer linear programming. Omega, 34, 167–177.

[16]. Taguchi, G. Chowdhury, S. (2000). Taguchi, Robust Engineering. McGraw-Hill, New York.

[17]. Teunter, R. H. Pelin Bayındır, Z. van den Heuvel, W. (2006). Dynamic lot sizing with product returns and remanufacturing. International Journal of Production Research, 44, 4377–4400.

[18]. Wang, R.C. Liang, T.F. (2005). Applying possibilistic linear programming to aggregate production planning. International Journal of Production Economics, 98, 328–341.

[19]. Wang, S.C. Yeh, M.F. (2014). A modified particle swarm optimization for aggregate production planning. Expert Systems with Applications, 41, 3069–3077.

[20]. Zhang, R. Zhang, L. Xiao, Y. Kaku, I. (2012). The activity-based aggregate production planning with capacity expansion in manufacturing systems. Computers & Industrial Engineering, 62, 491–503.