Research Article

Enhanced Simulated Annealing for Solving Aggregate Production Planning

**Mohd Rizam Abu Bakar,¹ Abdul Jabbar Khudhur Bakheet,² Farah Kamil,³
Bayda Atiya Kalaf,¹ Iraq T. Abbas,¹ and Lee Lai Soon¹**

¹*Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Malaysia*

²*Department of Statistics, Faculty of Administration and Economics, University of Baghdad, Baghdad, Iraq*

³*Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, Malaysia*

Correspondence should be addressed to Bayda Atiya Kalaf; hbama75@yahoo.com

Received 28 March 2016; Accepted 11 July 2016

Academic Editor: Sergii V. Kavun

Copyright © 2016 Mohd Rizam Abu Bakar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Simulated annealing (SA) has been an effective means that can address difficulties related to optimisation problems. SA is now a common discipline for research with several productive applications such as production planning. Due to the fact that aggregate production planning (APP) is one of the most considerable problems in production planning, in this paper, we present multiobjective linear programming model for APP and optimised by SA. During the course of optimising for the APP problem, it uncovered that the capability of SA was inadequate and its performance was substandard, particularly for a sizable controlled APP problem with many decision variables and plenty of constraints. Since this algorithm works sequentially then the current state will generate only one in next state that will make the search slower and the drawback is that the search may fall in local minimum which represents the best solution in only part of the solution space. In order to enhance its performance and alleviate the deficiencies in the problem solving, a modified SA (MSA) is proposed. We attempt to augment the search space by starting with $N + 1$ solutions, instead of one solution. To analyse and investigate the operations of the MSA with the standard SA and harmony search (HS), the real performance of an industrial company and simulation are made for evaluation. The results show that, compared to SA and HS, MSA offers better quality solutions with regard to convergence and accuracy.

1. Introduction

Aggregate production planning (APP) is considered as an important technique in operations management. Other elemental methods that are closely related to this included capacity requirements planning (CRP), master production scheduling (MPS), and material requirements planning (MRP). APP refers to a medium-term resource and capacity planning that ascertains optimum levels of the inventory, workforce, subcontracting, production, and backlog to meet the inconsistent demands and requirements with limited capacity and supply over a definite period of time that spans from 3 to 18 months. Also, APP is implemented to resolve problems that involve collective decisions. It finds the total capacity level in industries for a certain period, while not calculating the amount of produce from each individual stock-keeping unit. The extent of details estimated makes

APP a useful instrument for analysing decisions with a transitional time frame where determining levels of production by stock-keeping unit becomes too early, while arranging for additional capacity becomes too late. The objective of APP is to fulfill the expected changing requirements and demands in a cost-effective way through a definite period. The usual expenses include the costs of inventory, production, backlog, subcontracting, hiring, layoff, backorder, over time, and regular time.

The APP problem has been researched widely since it was first conceptualised in the 1955 study by Holt et al. [1]. They proposed the HMMS rule. Bergstrom and Smith [2] generalized the HMMS approach to a multiproduct formulation. Bitran and Yanasse [3] investigated the problems of ascertaining production plans under stochastic demands over a set of time periods. Nam and Logendran [4] evaluated APP models and categorised them into near optimal and optimal.

APP along with the other mathematical optimisation models indicated that linear programming has the most pervasive acknowledgment. Silva Filho [5] developed a stochastic optimisation model with the limitations of production, stock, and workforce to explain a multiproduct and multiperiod APP problem. APP was described by Fung et al. (2003) [6] as a method to ascertain inventory, production, and workforce levels required to meet all market demands. Mirzapour Al-E-Hashem et al. [7] formulated a robust multiproduct, multiperiod, multiobjective, and multisite APP that handles contradictions in relation to total expenses of supply chain, customer assistance levels, and efficiency of workers during medium-term planning in an uncertain environment.

A mixed-integer linear programming model was proposed by Zhang et al. [8] for APP problems related to the expansion of production capacity. Jamalnia and Feili [9] introduced a distinct hybrid event simulation and system dynamics method to simulate and replicate APP problems. The prime purpose of their study was to find out the efficiency of APP strategies with regard to the total profit. Tonelli et al. [10] suggested an optimisation method to confront aggregate planning difficulties of a hybrid model production setting. They assessed the planning problem of a real-world assembly manufacturing system and tackled the model flexibility challenge. A mixed-integer, multiobjective, nonlinear programming model was proposed by Gholamian et al. [11] for scheduling aggregate production in an SC under uncertainty demand. The model resolves the problem with a fuzzy multiobjective optimisation approach. The suggested approach changes the nonlinear model to a linear one and then alters the fuzzy model to the parametric deterministic model.

In recent times, with the influence of sophisticated modeling methods invented and increasing the number of assumptions, the APP problem has turned into a complicated and large-scale problem. In the research community, there is a tendency to resolve large and complex problems by the use of modern heuristic optimisation methods. This is mainly due to the time-consuming and unsuitability of classical techniques in many circumstances.

A fuzzy multiobjective APP model was introduced by Baykasoglu and Gocken [12] along with a direct solution technique based on ranking orders of fuzzy numbers and tabu search. Ramazanian and Modares [13] presented a goal programming multiobjective model for a multiperiod, multistep, and multiproduct APP problem in the cement sector. The model was modified as a nonlinear programming model with a single objective. An extended objective function method and a suggested PSO variant whose inertia weight was established as a function were used to resolve it. A comparison of the simulation with GA and PSO confirmed that PSO shows more acceptable results than GA. To simulate the process of human decision-making, Wang and Fang [14] implemented a genetic algorithm (GA) based technique with fuzzy logic. In place of finding the most optimal solution, this procedure looks for a group of inaccurate solutions that yields results within satisfactory levels. Subsequently, a final solution was selected by examining a convex combination of these solutions.

Kumar and Haq [15] resolved an APP problem by utilising genetic algorithm (GA), ant colony algorithm (AGA), and hybrid genetic-ant colony algorithm (HGA). Based on the results, HGA and GA exhibited relatively good performance.

Aungkulanon et al. [16] considered various hybridisation techniques of the harmony search algorithm to explain the fuzzy programming method for several objectives to the APP problems. To resolve the multiobjective APP problem, Luangpaiboon and Aungkulanon [17] recommended hybrid metaheuristics of the hunting search (HuSIHSA) and firefly (FAIHSA) procedures based on the enhanced harmony search algorithm. A mathematical model was suggested by Kaveh and Dalfard [18] that took into consideration the time and expenses for preventative upkeep of APP problems and used a simulated annealing algorithm to solve this. In spite of this, they manage single objective only. Wang and Yeh [19] introduced the concept of subparticles to the update rules of PSO to relieve the lack of resources to change the PSO plan so that the restrained integer linear programming model can be implemented to solve real-world APP problems. However, all these presented approaches are generality concentrated on the solution algorithm but not on a general model. The majority of models in the APP are relevant to single product, single stage systems, and even single objective; they are not compatible with production systems [20].

A simulated annealing algorithm (SAA) has not been functional in solving multiobjective APP problems, even though most applications made use of metaheuristic algorithms on APP. In this paper, we first suggest a conventional multiobjective linear programming model for APP. Throughout the implementation of SA to the APP problems, it was observed that SA has an inadequacy with respect to big APP problems that have plenty of decision variables. To improve the SA efficiency for the production planning system, a modified SA is introduced that can resolve APP problems with multiobjective linear programming model, since a large number of companies intend to fulfill more than one objective while creating a response and flexibility. The objective of the model is to decrease the aggregate expenses of production and workforce, and simulation and real-life data were used to verify the effectiveness of this method.

The organisation of the paper is as follows. Section 2 gives an overview of harmony search. Section 3 includes some essential theories of simulated annealing algorithm. Section 4 introduces the scheme of modifying the simulated annealing algorithm. Section 5 explains the multiobjective linear programming model of the APP. Section 6 discusses the trials and comparisons for the altered SA with standard HS and SA. Section 7 provides a few conclusions and offers some discussions.

2. Multiobjective Linear Programming Model

We proposed mathematical model for APP problem and assumed that an industrial company manufacturing produces n types of products to fulfill market demand over planning time horizon T . We considered two objective functions in this paper: to minimize production costs and minimize workforce costs. We will describe them below.

2.1. Objective Function

Minimize Production Costs. Consider the following:

$$\min Z_1 = \sum_{n=1}^N \sum_{t=1}^T c_{nt} P_{nt} + i_{nt} I_{nt}. \quad (1)$$

Minimize Workforce Costs. Consider the following:

$$\min Z_2 = \sum_{t=1}^T w_t W_t + h_t H_t + f_t F_t + o_t O_t. \quad (2)$$

2.2. Subject to Constraint. Consider the following:

$$\begin{aligned} P_{nt} + I_{n(t-1)} - I_{nt} &= D_{nt}, \quad \forall n, \forall t \\ F_t - H_t + W_t - W_{t-1} &= 0 \quad \forall t \\ O_t - A_O \times W_t &\leq 0 \quad \forall t \\ \sum_{n=1}^N M_n P_{nt} - A_R \times W_t - O_t &\leq 0 \quad \forall t. \\ H_t &\leq H_{\max} \\ F_t &\leq F_{\max} \\ H_t, F_t, W_t &\text{ are integer, } P_{nt}, I_{nt}, O_t \geq 0. \end{aligned} \quad (3)$$

3. Harmony Search Algorithm

Geem et al. [21] first introduced the harmony search algorithm HSA. HSA is believed to be an inspiration algorithm based socially on local search attributes. The concept of HSA has originated from the natural pattern of musicians conduct when they play their musical instruments or improvise the music together.

This creates a perfect state of harmony or a pleasurable harmony as governed by an aesthetic quality with the pitch of every musical device. In a similar manner, the optimisation technique searches for a globule solution as ascertained by an objective function through a series of values designated to every decision variable. In a musical orchestration, the set of pitches from every musical instrument accomplishes the aesthetic evaluation. The quality of harmony gets enriched with every practice. Each style of music is composed by musicians out of specific instruments. If all musical pitches create a perfect harmony, then that musical experience remains in every instrument player's memory, and the likelihood to create a good harmony the next time increases manifold. If every player plays together with dissimilar notes, then a new musical harmony is composed. There are three rules of musical improvisation playing a completely random pitch from the workable sound range, playing a pitch from memory, or playing an adjoining pitch of a pitch from memory. These techniques rules are implemented in HSA and the same was explained to describe the process of HSA.

The HSA process can be summed up as follows.

Step 1. Initialize the harmony memory (HM) which contains HMS vectors generated randomly, where $x_i = x_{ij}$, $i = 1, 2, \dots, \text{HMS}$, and $j = 1, 2, \dots, n$, where n is the size of the problem, and the harmony memory (HM) matrix is filled with HMS and f represented the fitness function as follows:

HM

$$= \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} & f(x_1) \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} & f(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{\text{HMS}_1} & x_{\text{HMS}_2} & x_{\text{HMS}_3} & \cdots & x_{\text{HMS}_n} & f(x_{\text{HMS}_n}) \end{bmatrix}. \quad (4)$$

Step 2. Improvise a new harmony. There are three rules for this:

- (1) Harmony memory considering (HMC) rule: for this rule, a new random number r_1 is generated within the range $[0, 1]$.

If $r_1 < \text{HMCR}$, then the first decision variable in the new vector x'_{ij} is chosen randomly from the values in the current HM as follows:

$$x_{ij}^{\text{new}} = x_{ij}, \quad x_{ij} \in \{x_{1j}, x_{2j}, \dots, x_{\text{HMS}_j}\}. \quad (5)$$

- (2) Pitch adjusting rate (PAR): a new random number r_2 is generated within the range $[0, 1]$.

If $r_2 < \text{PAR}$, then the pitch adjustment decision variable is calculated as follows:

$$x_{ij}^{\text{new}} = x_{ij} \pm r_3 \cdot \text{BW}, \quad (6)$$

where BW is a bandwidth factor, which is used to control the local search around the selected decision variable in the new vector.

- (3) Random initialization rule is as follows.

If the condition $r_1 < \text{HMCR}$ fails, the new first decision variable in the new vector x_{ij}^{new} is generated randomly as follows:

$$x_{ij}^{\text{new}} = l_{ij} + (u_{ij} - l_{ij}) r_4, \quad (7)$$

where l, u are the lower and upper bound for the given problem and $r_3, r_4 \in (0, 1)$

Step 3. Update harmony memory. If improvised harmony vector is better than the worst harmony, replace the worst harmony in the HM form:

$$\text{if } f(x'^{\text{new}}) < f(x^{\text{worst}}) \text{ then } x^{\text{worst}} = x^{\text{new}} \quad (8)$$

Step 4. Check the stopping criterion. If the stopping criterion is satisfied, the computation is terminated. Otherwise, repeat Steps 3 and 4.

The steps of HSA are elucidated in Figure 1.

FIGURE 1: Flowchart for HS procedure.

4. Simulated Annealing Algorithm

For complicated optimisation problems, simulated annealing (SA) has been regarded as an effective measure. Kirkpatrick et al. [22] introduced SA for the first time. SA refers to a random search technique that makes use of a similarity between the method through which a metal cools down and freezes to a crystalline structure (the annealing process) with minimum energy and the search process for a minimum in a more undefined system. This is used as the base of an optimisation method that can be used for combinatorial problems. The method has been extensively used to solve various problems. The foremost benefits and strengths of SA over other search techniques are its flexibility and capability to achieve overall optimality. Conversely, the main disadvantage of the method is its extremely slow convergence in big problems. In SA,

first a primary solution is randomly created, and a neighbour is found and this is accepted with a probability of $\min(1, e^{(-\delta f/T)})$, where T is the control parameter corresponding to the temperature of the physical analogy and is called temperature, and δf is the cost difference [23]. On gradual decrease of temperature, the algorithm congregates to the global minimum. The operations of the SA are depicted in Figure 2.

5. Modified Simulated Annealing Algorithm

Practitioners make extensive use of SA to solve NP-hard problems. But SA can get stuck at a local minimum and not attain the global optimum because of its inadequate capability and substandard performance, specifically with a large number of decision variables in APP problems. It is also

FIGURE 2: Operations for SA.

programmed to function sequentially and hence the current state will generate only one subsequent state. This will slow down the search. In order to improve the problem solving by the aforementioned SA, a modified SA (MSA) is proposed to augment its performance and address the problem-solving insufficiencies. We attempt to augment the search space by starting with $N + 1$ solutions, instead of one solution. These solutions work in a parallel way for utilising the overall search area in short time. The details are described in the following steps, and the operations of the MSA are also illustrated in Figure 3.

Step 1. Generate $n + 1$ initial solutions, namely, $x_1, x_2, \dots, x_n, x_{n+1}$, and set initial high temperature T .

Step 2. Find the objective function for each $x_i, i = 1, 2, \dots, n, n + 1$.

Step 3. Sort the solution such as $f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq f(x_{n+1})$, where $f(x_1)$ is the best solution (current solution).

Step 4. Repeat this step m times:

- (i) Generate a new solution x'_i for each x_i .
- (ii) If $f(x'_i) \leq f(x_i)$ then $x_i = x'_i$ and $f(x_i) = f(x'_i)$.
- (iii) Else if $p = e^{-(|f(x'_i) - f(x_i)|)/T}$ and $P < Z$ then $x_i = x'_i$, $f(x) = f(x'_i)$, where Z is a random number $\in (0 \text{ to } 1)$ and p is Boltzmann probability factor.

Step 5. Reduce the parameter T according to $T_{k+1} = \alpha T$, $\alpha = 0.95$. Convergence criteria are satisfied and if not then go to Step 3. Concerning the stopping condition, theory suggests a final temperature equal to 0.

6. Computational Study and Results

To assess the functioning of the MSA algorithm for the APP problem, two criteria are used: APP problem from real-world (case study) and simulation. Standard SA, MSA, and HS are coded by MATLAB R2015a; the prevailing algorithms (SA, HS) are compared with the proposed algorithm.

FIGURE 3: Operation for modified SA.

6.1. Case Study. In this section, the General Company for Vegetable Oils is used as a case study to demonstrate the proposed model. This company produces ten types of products. Each product is represented by a letter (A, B, C, D, E, F, G, H, J, and K). The time horizon of APP decision is six months. Tables 1–3 represent costs of production and inventory, hours required to produce one ton for each product, and forecast demand for each product, respectively. The initial inventory for product A is 105 tons, E is 333 tons, H is 0.25 ton, and J is 1.8 ton. The initial worker level is 3313 workers. The cost of regular worker per month is 500 \$/man, the working hours in one month are 140 hours. 5.357 \$ (dollar) is overtime costs per worker per hour. The costs associated with hiring and firing are 774.910 \$ and 581.182 \$ per worker, respectively. Hours of overtime, allowed during the period, are 60 hours per period t .

The authors select HS because it provides better results than standard SA in terms of costs especially in the second objective Z_2 , while HS gives slower runtime than standard SA.

TABLE 1: Production and inventory costs in dollars.

	A	B	C	D	E	F	G	H	J	K
c_{nt}	3285	385	451	1006	801	487	449	1007	496	739
i_{nt}	38	47	53.6	35.511	35.415	24.6	37.75	37.666	58.666	37

TABLE 2: Hours required to produce one ton of product.

Product	A	B	C	D	E	F	G	H	J	K
Hours	92	525	69	64	50	121	42	607	172	692

After we used the MSA to solve the case study, it is clear from the results that the proposed MSA was effective and faster than the standard SA and HS as shown in Table 4.

Due to the fact that MSA provided better results than the SA and HS in terms of cost and time, consequently, MSA was used to solve the model in the company as it is shown in Tables 5–7.

Tables 5 and 6 contain the amount of the production yield and inventory levels, respectively, to fulfill the forecast

TABLE 3: Forecast demand for all products.

Period	A	B	C	D	E	F	G	H	J	K
1	3049.1	539	340.6	100	606.4	23.1	1.7	1.2	3.1	0/74
2	1664.1	509	708.1	152	482.7	265	3.3	2	1.8	1.1
3	1236.4	35.4	700	138	496.8	14.8	7.4	1.7	2.3	0.47
4	782.5	40.8	650	77	429.9	25	8.7	2.5	2.9	0.76
5	914.4	275	439	56	324.7	15	215	2.4	2.1	2.3
6	652.9	379	619.1	50	652.9	12.4	29.1	1.3	2.7	0.71

TABLE 4: Results for each algorithm.

Algorithm	Z_1	Z_2	Time
HS	7043102	5737053	157 s.
SA	7159657	5918976	17 s.
MSA	6993111	5814778	11 s.

TABLE 5: Production yield.

Product	P1	P2	P3	P4	P5	P6
A	2944.82	1664.1	1236.4	782.5	914.4	652.9
B	53.9	50.9	35.4	40.8	27.5	37.9
C	340.6	708.1	700	650.2	439	619.1
D	100	152	138	77	56	50
E	144.071	482.7	496.8	429.9	324.7	652.9
F	23.101	26.501	14.801	25	14.5671	11.7534
G	1.7	3.3	7.4	8.7	21.5	29.1
H	1.19	2	1.7	2.5	3.69	0.7
J	1.258	1.8	2.3	2.9	2.1	2.7
K	0.741	1.1669	0.4614	0.722	3.0498	0.068

TABLE 6: Inventory levels.

Product	P1	P2	P3	P4	P5	P6
A	36.50167	36.5017	36.5017	36.502	36.502	36.502
B	1.29	2.59	3.63	4.15	4.15	5.27
C	0	0	0	0	0	0
D	0	0	0	0	0	0
E	0.1612	0.1612	0.1612	0.1612	0.1612	0.1612
F	0	0	0	0	0.2308	0
G	0	0	0	0	0	0
H	0.1434	0.1434	0.1434	1.1434	0.1946	0.1263
J	0	0	0	0	0	0
K	0.1	0.6	0.06	0.02	0.77	0

TABLE 7: The rate of workforce level.

Workforce level	P1	P2	P3	P4	P5	P6
W_t	1915	1831	1602	1296	1190	1191
H_t	0	0	0	0	0	1
F_t	1400	84	229	306	106	0
O_t	31079	0	0	0	0	0

demand, while Table 7 shows the results in the first column which represent the first period. Number 1915 explains the

FIGURE 4: Optimal cost comparison MSA, SA, and HS.

number of regular workers that should be hired during that period. Also, number 0 refers to the fact that the company does not need to hire any worker in that period. In addition, the result ($F_t = 1400$) represents the number of workers that should be fired from the factory, while number 31079 refers to the hours of overtime that should be adopted in that period.

6.2. Simulation. Simulation is another criterion for analysing the effectiveness of the proposed modified SA. During simulation, based on the same preliminary data set for the case study, the algorithm was used to resolve various problems. To compare the performances of these algorithms, SA, MSA, and HS, 9 cases of problems have been used depending on regular working hours (A_R), overtime hours (A_O), and different number of products and periods. The algorithms have been implemented 1000 times for every problem. The average time and the total costs for each objective of MSA, SA, and HS are depicted in Table 8. According to this table, the MSA provided better results by minimizing the objective function through utilising a different number of products and period. Besides that, we changed the regular A_R and A_O to enlarge and change our problem, where N and P represent the product number and period number, respectively, while minimizing production costs and minimizing the workforce costs are represented by Z_1 and Z_2 , respectively.

From Figure 4, we can see that it is evident that the modified algorithm is superior even for complex and large problems in minimizing the total cost of production and workforce levels. As illustrated in Figure 5, the quality of solutions obtained by MSA is better in runtime than SA and HS.

7. Conclusion

Simulated annealing provides a mechanism to escape local optima by allowing hill-climbing moves in hopes of finding

TABLE 8: Optimal costs and time for each problem when $A_R = 160$, $A_O = 100$ hours, respectively.

			SA	HS	MSA
$N = 15$	$P = 6$	Z_1	96231219.04	95850473.04	93276091.81
		Z_2	44743742.22	44686669.41	44327446.38
		Time	4.8048	9.8436	0.6864
$N = 15$	$P = 12$	Z_1	250053722.9	247445920	241796947.2
		Z_2	61096198.47	61096198.47	61096198.47
		Time	8.4552	17.4097	0.7956
$N = 15$	$P = 18$	Z_1	364563840.4	363660493.9	350843833.1
		Z_2	48838028.24	48838028.24	48838028.24
		Time	13.1352	44.3042	0.4212
$N = 25$	$P = 6$	Z_1	178871907.1	176467812.5	173263161
		Z_2	37988545.97	37988545.97	37988545.97
		Time	5.0856	14.821	0.3276
$N = 25$	$P = 12$	Z_1	272811050.9	271276691.3	260926577.2
		Z_2	54821702.95	47805043.77	45726695.02
		Time	8.081	68.999	0.4056
$N = 25$	$P = 18$	Z_1	489020932.2	486493908	467749414.1
		Z_2	55478086.78	50555865.02	50163753.84
		Time	16.9105	135.1904	0.4524
$N = 35$	$P = 6$	Z_1	209736151.9	208824558.5	202629622.2
		Z_2	42890759.51	41396915.52	41396915.52
		Time	8.346	18.33	0.3276
$N = 35$	$P = 12$	Z_1	400060878.1	399731639.9	384906240.5
		Z_2	62516501.51	54712694.41	46909627.02
		Time	11.4817	126.0956	0.421
$N = 35$	$P = 18$	Z_1	769395323.5	767172581.9	740635030.5
		Z_2	87815572.15	85013218.68	85013218.68
		Time	16.3801	165.478	0.4683

FIGURE 5: Comparison time running for each algorithm.

a global optimum. However, it has a disadvantage which is the imperfections ability and unacceptable performance, especially a large constrained APP issue. Therefore, this

algorithm works sequentially that the current solution will generate only one solution. To improve its performance and lessen its insufficiencies to problem-solving, a modified SA (MSA) is introduced that will expand the search space by generating $N + 1$ solutions to create as many numbers as possible of the neighbour states. The study effectively explains the applicability of the MSA algorithm in solving the APP problem in industries. The results indicated that the use of the modified SA resolved numerous problems, including the complex and large ones.

Definition of Notations

- n : Number of products, $n = 1, 2, \dots, N$
- t : Number of periods in the planning horizon,
 $t = 1, 2, \dots, T$
- c_{nt} : Production cost per ton of product n per period t (\$/ton)
- i_{nt} : Inventory carrying cost per ton of product n per period t (\$/ton)
- h_t : Hiring cost per worker in period t (\$, worker)
- f_t : Firing cost per worker in period t (\$, worker)
- o_t : Cost per man-hour of overtime labor per period t (\$, worker)

w_t :	Cost of regular labor per period t (\$, worker)
D_{nt} :	Forecasted demand for product n per period t (tons)
P_{nt} :	Production of product n per period t (tons)
I_{nt} :	Inventory level of product n per period t (tons)
O_t :	Man-hours of overtime labor per period t
W_t :	Workforce level per period t (workers)
H_t :	Hired workers per period t (workers)
F_t :	Fired workers per period t (workers)
$H_{\max}$:	Maximum hiring in each period
$F_{\max}$:	Maximum firing in each period
M_n :	Hours required to produce one ton of product n
A_R :	Working regular hours per period t
A_O :	Working overtime hours, which are allowed during per period t
K_n :	Hours required to produce one ton for product n per worker.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

- [1] C. C. Holt, F. Modigliani, and H. A. Simon, "A linear decision rule for production and employment scheduling," *Management Science*, vol. 2, no. 1, pp. 1–30, 1955.
- [2] G. L. Bergstrom and B. E. Smith, "Multi-item production planning—an extension of the hmms rules," *Management Science*, vol. 16, no. 10, pp. 614–629, 1970.
- [3] G. R. Bitran and H. H. Yanasse, "Deterministic approximations to stochastic production problems," *Operations Research*, vol. 32, no. 5, pp. 999–1018, 1984.
- [4] S.-J. Nam and R. Logendran, "Aggregate production planning—a survey of models and methodologies," *European Journal of Operational Research*, vol. 61, no. 3, pp. 255–272, 1992.
- [5] O. S. Silva Filho, "An aggregate production planning model with demand under uncertainty," *Production Planning & Control*, vol. 10, no. 8, pp. 745–756, 1999.
- [6] R. Y. Fung, J. Tang, and D. Wang, "Multiproduct aggregate production planning with fuzzy demands and fuzzy capacities," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 33, no. 3, pp. 302–313, 2003.
- [7] S. M. J. Mirzapour Al-E-Hashem, H. Malekly, and M. B. Aryanezhad, "A multi-objective robust optimization model for multi-product multi-site aggregate production planning in a supply chain under uncertainty," *International Journal of Production Economics*, vol. 134, no. 1, pp. 28–42, 2011.
- [8] R. Zhang, L. Zhang, Y. Xiao, and I. Kaku, "The activity-based aggregate production planning with capacity expansion in manufacturing systems," *Computers and Industrial Engineering*, vol. 62, no. 2, pp. 491–503, 2012.
- [9] A. Jamalnia and A. Feili, "A simulation testing and analysis of aggregate production planning strategies," *Production Planning and Control*, vol. 24, no. 6, pp. 423–448, 2013.
- [10] F. Tonelli, M. Paolucci, D. Anghinolfi, and P. Taticchi, "Production planning of mixed-model assembly lines: a heuristic mixed integer programming based approach," *Production Planning and Control*, vol. 24, no. 1, pp. 110–127, 2013.
- [11] N. Gholamian, I. Mahdavi, and R. Tavakkoli-Moghaddam, "Multi-objective multi-product multi-site aggregate production planning in a supply chain under uncertainty: fuzzy multi-objective optimisation," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 2, pp. 149–165, 2016.
- [12] A. Baykasoglu and T. Gocken, "Multi-objective aggregate production planning with fuzzy parameters," *Advances in Engineering Software*, vol. 41, no. 9, pp. 1124–1131, 2010.
- [13] M. Ramazanian and A. Modares, "Application of particle swarm optimization algorithm to aggregate production planning," *Asian Journal of Business Management Studies*, vol. 2, no. 2, pp. 44–54, 2011.
- [14] D. Wang and S.-C. Fang, "A genetics-based approach for aggregated production planning in a fuzzy environment," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 27, no. 5, pp. 636–645, 1997.
- [15] G. M. Kumar and A. N. Haq, "Hybrid genetic—ant colony algorithms for solving aggregate production plan," *Journal of Advanced Manufacturing Systems*, vol. 4, no. 1, pp. 103–111, 2005.
- [16] P. Aungkulanon, B. Phruksaphanrat, and P. Luangpaiboon, "Harmony search algorithm with various evolutionary elements for fuzzy aggregate production planning," in *Intelligent Control and Innovative Computing*, S. I. Ao, O. Castillo, and X. Huang, Eds., vol. 110 of *Lecture Notes in Electrical Engineering*, pp. 189–201, Springer, Berlin, Germany, 2012.
- [17] P. Luangpaiboon and P. Aungkulanon, "Integrated approaches to enhance aggregate production planning with inventory uncertainty based on improved harmony search algorithm," *Proceedings of World Academy of Science, Engineering and Technology*, no. 73, p. 243, 2013.
- [18] M. Kaveh and V. M. Dalfard, "A simulated annealing algorithm for aggregate production planning with considering of ancillary costs," *International Journal of Mathematics in Operational Research*, vol. 6, no. 4, pp. 474–490, 2014.
- [19] S.-C. Wang and M.-F. Yeh, "A modified particle swarm optimization for aggregate production planning," *Expert Systems with Applications*, vol. 41, no. 6, pp. 3069–3077, 2014.
- [20] R. Ramezani, D. Rahmani, and F. Barzinpour, "An aggregate production planning model for two phase production systems: solving with genetic algorithm and tabu search," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1256–1263, 2012.
- [21] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [22] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [23] V. M. Dalfard and G. Mohammadi, "Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints," *Computers and Mathematics with Applications*, vol. 64, no. 6, pp. 2111–2117, 2012.