Various Hybridisations of Harmony Search Algorithm for Fuzzy Programming Approach to Aggregate Production Planning

P. Aungkulanon*, B. Phruksaphanrat and P. Luangpaiboon, *Member, IAENG*

Abstract — This paper presents an application of a fuzzy programming approach for multiple objectives to the aggregate production planning (APP). The parameters levels have been applied via a case study from the SMEs company in Thailand. The proposed model attempts to minimise total production cost and minimise the subcontracting units. Conventional harmony search algorithm (HSA) with its hybridisations of the global best harmony search (GHSA) and the variable neighbourhood search of the HSA (VHSA) are applied for fine aggregate production planning. Based on the experimental results, it can be concluded that the proposed VHSA is more effective than the other approaches in terms of superiority of solution and required CPU time. The basic idea is the change of neighbourhoods during searching for a better solution. The hybridisations proceed by a descent method to a local minimum exploring then, systematically or at random, increasingly distant neighbourhoods of this local solution. The proposed model yields an efficient compromise solution and the overall levels of decision making satisfaction with the multiple objectives.

Index Terms—**Aggregate Production Planning, Fuzzy Programming, Harmony Search Algorithm, Global Best Harmony Search Algorithm, Variable Neighbourhood Search Harmony Search Algorithm.**

I. INTRODUCTION

Most manufacturing companies in Thailand [1] do not perform appropriate production planning even though it plays an important role for the companies. This may be the result from the following issues. Firstly, production planning system is not available. Secondly, manufacturing companies in Thailand are not interested in the mathematical model of production systems including related complex approaches to solve such models. Thirdly, manufacturing companies require an approach that is easy to understand and verify in order to easily convince their management team to agree with its solution. Finally, an approach should not require the additional investment on any expensive software due to the ongoing economic crisis in Thai industries. Based on those problems this paper intends to provide another solution for developing aggregate production plans to decrease the production cost.

Aggregate production planning or APP [2, 3] is concerned with the determination of production, inventory, and workforce levels to meet fluctuating demand requirements over a planning horizon that approximately ranges from six months to one year [4, 5]. Typically, the planning horizon incorporates the next seasonal peak in demand. The planning horizon is often divided into periods. Given demand forecasts for each period of a finite planning horizon, the APP specifies production levels, workforce, inventory levels, subcontracting rates and other controllable variables in each period that satisfy anticipated demand requirements while minimising the relevant cost over that production planning horizon. The fluctuations in customer's demand can be absorbed by adopting one of the following conventional strategies.

Firstly, the production rate can be altered by affecting changes in the workforce through hiring or laying off workers. Secondly, the production rate can also be altered by maintaining a constant labour force but introducing overtime or idle time. Thirdly, the production rate may be kept on a constant level and the fluctuations in demand met by altering the level of subcontracting. Finally, the production rate may be kept constant and changes in demand absorbed by changes in the inventory level. Any combination of these strategies or alternatives is possible. The concern of the APP is to select the strategy with the least cost to the manufacturing companies. This problem has been under an extensive discussion and several alternative methods for finding an optimal solution have been suggested in the literature [6, 7].

II. AGGREGATE PRODUCTION PLANNING (APP)

A. Multi-Objective Linear Programming (MOLP) Model to APP

Objective functions of the APP model are to minimise the total production cost (Z_1) and to minimise subcontracting units (Z_2). The total production cost composes of permanent worker salary, temporary worker wage, overtime cost of permanent and temporary workers, hiring and laying off cost of temporary workers and inventory holding cost over a specific number of *m* monthly planning periods in the planning horizon. In this model, the overtime costs of

Manuscript received January 7, 2011; revised January 28, 2011. This work was supported by the National Research University Project of Thailand Office of Higher Education Commission. The authors wish to thank the Faculty of Engineering, Thammasat University, THAILAND for the financial support.

*P. Aungkulanon is a Ph. D. Candidate, the Industrial Statistics and Operational Research Unit (ISO-RU), Department of Industrial Engineering, Faculty of Engineering, Thammasat University, 12120, THAILAND. [Phone: (662)564-3002-9; Fax: (662)564-3017; e-mail: pasurachacha@hotmail.com, lpongch@engr.tu.ac.th].

B. Phruksaphanrat is an Assistant Professor, ISO-RU, Department of Industrial Engineering, Faculty of Engineering, Thammasat University, 12120, THAILAND.

P. Luangpaiboon is an Associate Professor, ISO-RU, Department of Industrial Engineering, Faculty of Engineering, Thammasat University, 12120, THAILAND.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

permanent and temporary workers during workdays and holidays are different. The decision variables are as follow.

W	= the number of permanent workers
TW(t)	= the number of temporary workers in period *t*
H(t)	= the number of temporary workers to be hired at the beginning of period *t*
L(t)	= the number of temporary workers to be laid off at the end of period *t*
$OW_n(t)$	= overtime man-hours of permanent worker during normal workday in period *t*
$OTW_n(t)$	= overtime man-hours of temporary worker during holiday in period *t*
P(t)	= total production quantity in period *t*
I(t)	= inventory level in period *t*
Sub(t)	= amount of subcontracted unit in period *t*

Moreover, there are some related parameters in the model. *n(t)* and *h(t)* are the number of normal workdays and holidays that can apply OT in period *t*, respectively. R_H and OH_n are the number of regular working hours and allowable overtime hours in each normal workday, respectively. *MIN W* and *Max W* are the minimal and maximal number of permanent workers to operate the production line, respectively. K_w and K_{tw} are an average productivity rate per man-day of permanent and temporary worker, respectively. *D (t)* and *SS (t)* are forecasted demand and safety stock level in period t, respectively. $MAX\ O_n\ (t)$ and $MAX\ O_h\ (t)$ are the maximal overtime man-hours that can be applied during normal workday and holiday in period t, respectively.

MAX TW is the maximal number of temporary workers to operate the production line. *MAX I* is the maximal allowable inventory level. *MAX Sub* is the maximal allowable subcontracting units. *CW* is an average salary per month of a permanent worker. *CTW* is an average wages per day of a temporary worker. *CH* is a hiring cost per person of temporary worker. *CL* is an average inventory holding cost per month per unit of product. COW_n and COW_h are overtime cost per man-hour of permanent worker during normal workday and holiday, respectively. $COTW_n$ and $COTW_h$ are overtime cost per man-hour of temporary worker during normal workday and holiday, respectively. *CSub* is a subcontracting cost per unit.

There are six APP constraints which consist of permanent worker, inventory, production, overtime, temporary worker and subcontracting constraints. Firstly, the number of permanent workers should not be less than the minimal limit; otherwise the production line cannot function. Also, it should not be more than the maximum limit; otherwise some workers will be idle (3). Secondly, the inventory in each period is equal to the inventory from the previous period plus the production minus the demand of the period (4). Moreover, all demands must be satisfied and the inventory level cannot be less than the specified safety stock level (5). The inventory level cannot exceed the maximal allowable limit since there are limited warehouse spaces (6).

Thirdly, the production quantity in each period is equal to the sum of production quantities generated by permanent and temporary workers during regular time and overtime (both regular workdays and holidays), plus subcontracted quantities, minus a loss of production due to under time (idle time) in the period. Note that idle time can help reducing the unnecessary inventory level during low demand periods. This constraint allows different productivity rates between permanent and temporary workers (7). Fourthly, the total overtime man-hours of permanent and temporary workers must not exceed the maximum allowable limit. The limit is calculated based on the total number of permanent and temporary workers, number of days, and number of hours in each day that the overtime can be applied (8–11). Since permanent and temporary workers work as a team in the same production line, the number of overtime man-hours per person applied to both groups must be the same (12–13).

On the fifth constraint, Thai labor law states that temporary workers could not be continuously hired longer than four months (The Labor Protection Act of Thailand, 1998). After four months they must become permanent workers. Hence, Thai industries will lay off temporary workers after four months if they do not want to transfer them to permanent ones. Constraint 14 shows that the total number of temporary workers working in the current period is the sum of the numbers of temporary workers hired at the beginning of the last three periods and the current period. Constraint 15 indicates that temporary workers hired three periods ago will be laid off at the end of the current period. APP models developed from western countries do not take this constraint into consideration The total number of temporary workers in each period cannot exceed the maximum allowable limit since the production line has limited number of workstations where the temporary workers con be assigned (16). Finally, the number of subcontracting units cannot exceed the maximum allowable limit since subcontractors have limited production capacity (17).

All decision variables are nonnegative and some decision variables representing number of workers, namely, *W*, *H(t)*, *L(t)*, and *TW(t)* are integer. Since in real situations the variables *W*, *H(t)*, *L(t)* and *TW(t)* have relatively high values, the integer conditions for these variables can be relaxed in order to reduce the computation time. The solutions can be later rounded to the nearest integer. Thus, the objective function and constraints over the time period (*t* = 1, 2, …, 12) can be formulated as the APP mathematical programming model below and all parameters and the initial values, such as the forecasted demand, the number of holidays for overtime and the number of normal workdays, in each period, are given in Table I.

Min Z_1 =

$$\sum_{t=1}^{m}\left[\begin{array}{c} CW(W) + COW_n(OW_n(t)) + COW_h(OW_h(t)) + \\ CTW(TW(t)n(t)) + COTW_n(OTW_n(t)) + COTW_h(OTW_h(t) + \\ CH(H(t)) + CL(L(t)) + CI(I(t)) \end{array}\right] \quad (1)$$

$$\text{Min } Z_2 = \sum_{t=1}^{m} Sub(t) \quad (2)$$

Subject to

$$\min W \le W \le \max W \quad (3)$$

$$I(t) = I(t-1) + P(t) - D(t) \quad (4)$$

$$I(t) \ge SS(t) \quad (5)$$

$$I(t) \le MAXI \quad (6)$$

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$P(t) = W \bullet K_W \bullet n(t) + (OW_n(t) + OW_h(t)KW / RH + TW(t) \bullet K_{TW} \bullet n(t) + (OTW_n(t) + OTW_h(t))K_{TW} / RH \quad (7)$$

$$OW_n(t) + OTW_n(t) \le MAXO_n(t) \quad (8)$$

$$OW_h(t) + OTW_h(t) \le MAXO_h(t) \quad (9)$$

$$MAXO_n(t) > OH_n \bullet n(t)(W + TW(t)) \quad (10)$$

$$MAXO_h(t) > OH_h \bullet n(t)(W + TW(t)) \quad (11)$$

$$OW_n(t) / W = OTW_n(t) / TW(t) \quad (12)$$

$$OW_h(t) / W = OTW_h(t) / TW(t) \quad (13)$$

$$TW(t) = H(t-3) + H(t-2) + H(t-1) + H(t) \quad (14)$$

$$L(t) = H(t-3) \quad (15)$$

$$TW(t) \le MAXTW \quad (16)$$

$$Sub(t) \le MAXSub \quad (17)$$

Forecasted demands for each period are [146,000: 138,000: 145,000: 139,000: 165000: 145,000: 172,000: 148,000: 155,000: 141,000: 125,000: 118,000]. Normal workdays for each period are [24, 24, 26, 20, 22, 26, 25, 24, 26, 25, 26, 24].

TABLE I
PARAMETERS AND THEIR PRESET VALUES

Parameters	Preset Values
Regular working hours per day	8
Numbers of allowable overtime hours for each normal workday and holidays	2
Numbers of allowable overtime hours for each holiday	8
Minimal numbers of permanent workers to operate the production line	600
Maximal numbers of permanent workers to operate the production line	1,100
Maximal number of temporary workers to operate the production line	500
Average productivity rates of permanent workers	5
Average productivity rates of temporary workers	4.5
Maximal allowable inventory levels	unlimited
Maximal subcontracting units	unlimited
Average monthly salary per permanent workers	5,500
Average wage per day per temporary worker	162
Average cost of hiring one temporary worker	1,200
Laying off cod of temporary worker	Negligible
Average inventory holding cost per unit per month	200
Overtime costs per man-hour per permanent worker during normal workday	34.38
Overtime costs per man-hour per permanent worker during holiday	45.83
Overtime costs per man-hour per temporary worker during normal workday	30.38
Overtime costs per man-hour per temporary worker during holiday	40.50
Subcontracting cost per unit	300
Amount of Inventory at the beginning of the first period	0
Permanent workers based on the company policy to keep the number at this level	500
Temporary workers hired since three periods ago, two periods ago, and one period ago	150, 150 and 100
Holiday in each month	4

B. Fuzzy Programming Approach

Fuzzy programming approach is one of the most effective methods for solving Multiple Objective Decision Making (MODM) problem. A fuzzy set can be characterised by a membership function, usually denoted μ, which assigns to each objective [8]. In general, the non-increasing and non-decreasing linear membership functions are frequently applied for the inequalities with less than or equal to and greater than or equal to relationships, respectively. Since the solution procedure of the fuzzy mathematical programming is to satisfy the fuzzy objective, a decision in a fuzzy environment is thus defined as the intersection of those membership functions corresponding to fuzzy objectives [9, 10, 11]. Hence, the optimal decision could be any alternative in such a decision space that can maximise the minimum attainable aspiration levels, represented by those corresponding memberships.

In a general multiple objective linear programming model, all constraints are restricted to the forms of equality (=), less-than-or-equality (≤) or greater-than-or-equality (≥) as follows:

Maximise $Z_k = c_k^T x$
Subject to
$Ax \le b$, $x \ge 0$,
where $c_k \in R^n$, $x \in R^n$, $b \in R^m$ and $A \in R^{mxn}$, $k=1,\ldots,K$ (18)

In practical input data are usually fuzzy/imprecise because of incomplete or non-obtainable information. To formulate this fuzzy information, membership function is used. In the proposed model objective functions are fuzzy. Then, a symmetric fuzzy programming model can be represented by

$$c_k^T x \tilde{\ge} Z$$
$$Ax \le b,\ x \ge 0 \quad (19)$$

Here $\tilde{\ge}$ denotes the fuzzified version of ≥ and has the linguistic interpretation "essentially greater than or equal to." Using the simplest type of membership function, it is assumed to be linearly increasing over the "tolerance interval" of p_k and $k = 1, 2, \ldots, K$.

$$\mu_k(x) = \begin{cases} 1 & if \quad c_k^T x \le Z_k^{PIS} \\ 1 - \dfrac{Z_k^{PIS} - c_k^T x}{p_k} & if \quad Z_k^{PIS} < c_k^T x \le Z_k^{NIS}, \\ 0 & if \quad c_k^T x \ge Z_k^{NIS} \end{cases}$$

(21)

where $p_k = \left| Z_k^{PIS} - Z_k^{NIS} \right|, k = 1, 2, \ldots, K.$ (22)

The membership functions, $\mu_k(x)$ should increase monotonously from 0 to 1. P_k are subjectively chosen constants of admissible violations of objectives. It can be defined by the Positive-Ideal Solution (PIS) and the Negative-Ideal Solution (NIS). The Z_k^{PIS} is the best possible solution when *k*th objective is optimised. The Z_k^{NIS} is the feasible and worst value of *k*th objective.

Under the concept of min-operator, the feasible solution set is defined by interaction of the fuzzy objective set. The DM makes a decision with a maximum value in the feasible decision set. That is

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\max_{x\geq 0} \min_{k} \left(1 - \frac{Z_k^{PIS} - c_k^T x}{p_k} \right), \tag{23}$$

or equivalently

$$\begin{aligned} &\textit{Maximise } \lambda \\ &\text{subject to} \\ &\quad \mu_k(x) > \lambda, k = 1,2,..,K. \\ &\quad Ax \leq b, x \geq 0 \\ &\quad \lambda \in [0, 1]. \end{aligned} \tag{24}$$

Various hybridisations of harmony search algorithm in the following section for this fuzzy programming approach is applied to APP problem.

III. Harmony Search Algorithm

A. Harmony Search Algorithm (HSA)

Harmony search algorithm is a new meta-heuristic optimisation method proposed by Geem et al. in 2001 [12]. It is considered a population based or socially-based inspiration algorithm with local search aspects. HSA is conceptually derived from the natural phenomena of musicians' behaviour when they play or improvise their musical instruments together. This comes up with a pleasing harmony or a perfect state of harmony as determined by an aesthetic quality via the pitch of each musical instrument. Similarly, the optimisation process seeks to find a global solution as determined by an objective function via the set of values assigned to each decision variable.

In the musical improvisation, aesthetic estimation is performed by the set of pitches played by each instrument. The harmony quality is enhanced practice after practice. Each type of music composes of specific instruments played by musicians. If all pitches bring a good harmony, that experience is stored in each player's memory, and the possibility to make a good harmony is increased for the next time. Assume there are a certain number of preferable pitches in each musician's memory. Each instrument provides various notes. In music improvisation, each player sounds any pitch in the possible range, together making one a harmony vector.

If all plays together with different notes there is a new musically harmony. If this leads to a better new harmony than the existing worst harmony in their memories, a new harmony is included in their memories. In contrast, the worst harmony is excluded from their memories. Three rules of musical improvisation consist of rules of playing any one pitch from his memory, playing an adjacent pitch of one pitch from his memory, or playing totally random pitch from the possible sound range. These procedures are repeated until a fantastic harmony is found.

Similarly in engineering optimisation, harmony of the notes or pitches generated by a musician is analogous to the fitness value of the solution vector. Each musician can be replaced with each decision variable. The musician's improvisations are analogous to local and global search schemes in optimisation techniques. During searching, if all decision variable values make a good solution, that experience is stored in each variable's memory, and the possibility to make a good solution is also increased for the next time. Similarly, when each decision variable chooses one value in the HSA, it follows three rules which are to choose any one value from the harmony memory (HM) or memory considerations, choose an adjacent value of one value from the HM defined as pitch adjustments, or totally choose a random value from the possible value range defined as randomisation. These three rules in the HSA are associated with two parameters of a harmony memory considering rate (P_{HMCR}) and a pitch adjusting rate (P_{PAR}).

HSA is very successful in a wide variety of optimisation problems. It also presents several advantages with respect to conventional optimisation techniques. HSA does not require initial values for the decision variables and it imposes fewer mathematical requirements. Furthermore, instead of a gradient search like conventional algorithms, the HSA provides a stochastic search with no derivative information which is based on the harmony memory consideration rate or P_{HMCR} and the pitch adjustment rate or P_{PAR} so that it is not necessary to derive the associated function during the problem analysis. HSA generates a new vector, after considering all of the existing vectors, whereas other meta-heuristics, such as the genetic algorithm, only considers the two parent vectors. The pseudo code is used to briefly explain to all the procedures of the HSA shown in Fig. 1.

Procedure ***HSA Meta-heuristic()***
Begin;
Initialise algorithm parameters:
IM: *the preset number of improvisations*
HMS: *the size of the harmony memory*
BW: *the 'distance bandwidth' or the amount of maximal change for pitch adjustment between two neighbouring values in discrete candidate set*
P_{HMCR}: *the rate of considering from the harmony memory*
P_{PAR}: *the 'pitch adjustment rate*
Initialise the HMS harmony memories;
Evaluate the fitness values for all HMS;
For *j = 1 to IM*
Randomly select a position of [1, 2, …, HMS] to improvise;
Generate a random number in the range [0, 1] or RN1;
Check RN1 with P_{HMCR}*;*
If *RN1 <* P_{HMCR} *better, then pick the component from memory;*
Generate a random number in the range [0, 1] or RN2;
If *RN2 <* P_{PAR} *better, then adjust the harmony by a small amount BW;*
Generate a random number in the range [0, 1] or RN3;
If *RN3 > 0.5*
Pitch Adjustment Harmony vector increase;
Else
Pitch Adjustment Harmony vector decrease;
End if;
Else
Do nothing;
End if;
Else
Pick a new random value in the allowed range;
End if;
Replace a new harmony if better;
End for;
End;
End procedure;

Fig. 1. Pseudo Code of the HSA Meta-heuristic.

As concerned in the literature for the algorithm parameter levels, an *HMS* of 20-50, a P_{HMCR} of 0.7-0.95, and a P_{PAR} of 0.3-0.7 were frequently recommended in HSA applications. However, the *IM* and *BW* were determined based on the number of objective function and possible value ranges of decision variable evaluations from other competitive algorithms, respectively.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. Harmony Search Algorithm with VNS Concept of Improvement (VHSA)

Variable neighbourhood search method (VNS), initially introduced by Mladenovic and Hansen in 1997 [13], is one among meta-heuristics designed for solving combinatorial and global optimisation problems. It exploits systematically the idea of neighbourhood change within a local search method to approach a better solution. Contrary to other local search methods, VNSM proceeds by a descent method to a local minimum exploring then, systematically or randomly, increasingly distant neighbourhoods of this incumbent solution.

The first variant, called VHSA, uses maximal (or minimal) point in each iteration for the HMS improvement. This system needs improvement solutions of nearly optimal design point. Because HMS improves from an experience of solutions, these are then stored in groups of HMS. This system wants improvement in the best of the optimal points in each iteration.

C. Novel Global Best Harmony Search Algorithm (NGHSA)

A prominent characteristic of Particle Swarm Optimisation (PSO) [14] is that the individual is inclined to mimic its successful companion. Inspired by the swarm intelligence of particle swarm, a new variation of the HSA is proposed in this paper. The new approach, called NGHSA, modifies the improvisation step of the HSA such that a new harmony can mimic the global best harmony in the HM. NGHSA and HSA are different in three aspects as follows.

Firstly, in the first step the harmony memory considering rate (P_{HMCR}) and pitch adjusting rate (P_{PAR}) are excluded from the NGHSA, and genetic mutation probability (P_{m}) is included in the NGHSA. Secondly, in the third step, the NGHSA modifies the improvisation step of the HSA, and it works as shown in Fig. 2.

```
For each i ∈ [1, N] do
x_R = 2x_i^Best - x_i^Worst;
   If      x_R ≥ x_iU
           x_R = x_iU;
   Elseif  x_R < x_iL
           x_R = x_iL;
   End;
           x_i' = x_i^Worst + r(x_R - x_i^Worst);      % position updating
   If rand() ≤ p_m then
           x_i' = x_iL + rand() (x_iU - x_iL);         % genetic mutation
   End;
End;
```

Fig. 2. Pseudo Code of the NGHSA Sub-Procedure.

The most important step of the HSA is Step 3, and it includes memory consideration, pitch adjustment and random selection. P_{PAR} and BW have a profound effect on the performance of the HS. Mahdavi et al. [15] proposed a new variant of the HS, called the improved harmony search (IHS). The IHS dynamically updates P_{PAR} and BW according to (25)-(27):

$$P_{PAR}(gn) = P_{PAR(MIN)} + \frac{\left(P_{PAR(MAX)} - P_{PAR(MIN)}\right)}{Iteration(MAX)} \times Iteration(current) \quad (25)$$

$$BW(gn) = BW_{MAX}\, exp(c \times Iteration(current)) \quad (26)$$

; where
$$c = \frac{Ln(\frac{BW_{MIN}}{BW_{MAX}})}{Iteration(MAX)} \quad (27)$$

The ‘‘best” and ‘‘worst” are the indexes of the global best and the worst harmony in the HM, respectively. The terms of r and *rand()* are all uniformly generated random number in the range of [0, 1]. A new harmony, as shown in Fig. 1, is used to illustrate the principle of position updating with the step i of $\left|x_i^{Best} - x_i^{Worst}\right|$; where i is defined as adaptive step of the ith decision variable. The region between the worst and best solution spaces is defined as a trust region for the ith decision variable. The trust region is actually a region near the global best harmony. A reasonable explanation is as follows.

In the early stage of optimisation, all solution vectors are sporadic in solution space, so most adaptive steps are large and most trust regions are wide, which is beneficial to the global search of the NGHSA. While in the late stage of optimisation, all non-best solution vectors are inclined to move to the global best solution vector, most solution vectors are close to each other. In this case, most adaptive steps are small and most trust regions are narrow, which is beneficial to the local search of the NGHSA. The reasonable design for the ith step can guarantee that the proposed algorithm has strong global search ability in the early stage of optimization and also has strong local search ability in the late stage of optimisation. Dynamically the ith adjusted step keeps a balance between the global search and the local search. Genetic mutation operation with a small probability is carried out for the worst harmony of harmony memory after updating position. This could enhance the capacity of escaping from the local optimum for the proposed algorithm. In Step 4, the NGHSA replaces the worst harmony of x^{Worst} in the HM of x’ even if x’ is worse than x^{Worst}.

IV. Computational Results And Analyses

In this work with the computational procedures previously described, a computer simulation program of the fuzzy programming approach of the APP model was implemented in a Visual C#2008 computer program. The suitable linear and continuous membership function has been determined for quantifying the fuzzy aspiration levels. The corresponding linear membership functions can be defined in accordance with an analytical definition of membership functions. From the conventional harmony search algorithm, an interval of the membership calculated from all responses in both scenarios of the average minimal total production cost and the average subcontracting units in the harmony memory.

TABLE II
Initial results categorised by both objectives and individual objective with the Nominal Demand

Scenarios	Minimal Total Production Cost	Minimal Subcontracting Units
Feasible Best so far	66,343,158	95,907
Simple-HSA with the minimal cost	79,432,529	15715
Simple-HSA with the minimal Subcontracting	89,261,867	0
P_k	9,829,338	15,715

The HSA and VHSA parameters of HMS, P_{HMCR} and P_{PAR} are set at 30, 0.90 and 0.35, respectively. The NGHSA parameters of HMS, P_{HMCR}, P_{PARMAX}, P_{PARMIN}, BW_{Max}, BW_{MIN} and Pm are set at 30, 0.90, 0.99, 0.01, 0.005, 0.0000005 and 0.01 respectively. Minimal total production cost and subcontracting units scenarios are calculated from all previous data in the harmony memory with 200 iterations at the nominal demand including the demand decrease and increase at 10% (Tables II-IV). A mathematical expression of the proposed model for this APP problem with the nominal demand can be then shown as followed with the corresponding (3)-(17).

$$\max = [\lambda],$$
$$\lambda \le 1 - \left(\frac{z_1 - 79{,}432{,}529}{9{,}829{,}338} \right)$$
$$\lambda \le 1 - \left(\frac{z_2 - 0}{15{,}715} \right) \tag{28}$$

The comparisons are made for three different levels of demand in the APP. The demand deviation is taken to be independently and normally distributed with mean of the nominal level with the 10% increase and decrease. Using the fuzzy approach of the APP an aim is to simultaneously minimise total production costs and subcontracting units over a 12-month period. At the nominal demand (Table V), total production costs and subcontracting units seemed to be better at 0.960 of the overall levels of decision making satisfaction via the VHSA. The results repeated similarly when there is the demand increase at 10% as shown in Table VI. The NGHSA seemed to provide the better level of decision making satisfaction when there is the demand increase at 10% as shown in Table VII.

TABLE III
INITIAL RESULTS CATEGORISED BY BOTH OBJECTIVES AND INDIVIDUAL OBJECTIVE ON THE 10% DEMAND DECREASE

Scenarios	Minimal Total Production Cost	Minimal Subcontracting Units
Simple-HSA with the minimal cost	80,159,075	16,934
Simple-HSA with the minimal Subcontracting	89,471,737	0
P_k	9,312,662	16,934

TABLE IV
INITIAL RESULTS CATEGORISED BY BOTH OBJECTIVES AND INDIVIDUAL OBJECTIVE ON THE 10% DEMAND INCREASE

Scenario	Minimal Total Production Cost	Minimal Subcontracting Units
Simple-HSA with the minimal cost	100,030,561	156,131
Simple-HSA with the minimal Subcontracting	108,041,936	100,025
P_k	8,011,375	56,106

TABLE V
EXPERIMENTAL RESULTS CATEGORISED BY THE ALGORITHMS ON THE NOMINAL DEMAND

Algorithm	Max λ	Minimal Total Production Cost	Minimal Subcontracting Units
HSA	0.889	80,073,238	0
VHSA	0.960	80,012,432	0
NGHSA	0.949	80,071,863	0

TABLE VI
EXPERIMENTAL RESULTS CATEGORISED BY THE ALGORITHMS ON THE 10% DEMAND INCREASE

Algorithm	Max λ	Minimal Total Production Cost	Minimal Subcontracting Units
HSA	0.880	100,956,577	105,103
VHSA	0.915	100,536,630	104,038
NGHSA	0.910	100,616,218	102,163

TABLE VII
EXPERIMENTAL RESULTS CATEGORISED BY THE ALGORITHMS ON THE 10% DEMAND DECREASE

Algorithm	Max λ	Minimal Total Production Cost	Minimal Subcontracting Units
HSA	0.675	83,164,217	0
VHSA	0.629	82,859,019	0
NGHSA	0.725	81,632,893	0

When the performance of the HSA variants of HSA, VHSA and NGHSA was compared, the VHSA seems to be better in terms of speed of convergence (Fig. 3). The basic idea is the change of neighbourhoods during searching for a better solution. The hybridisations proceed by a descent method to a local minimum exploring then, systematically or at random, increasingly distance neighbourhoods of this local solution. Furthermore, in some additional experiments, small BW values with large P_{PAR} values usually cause the improvement of best solutions in final generations to converge to the optimal solution vector.

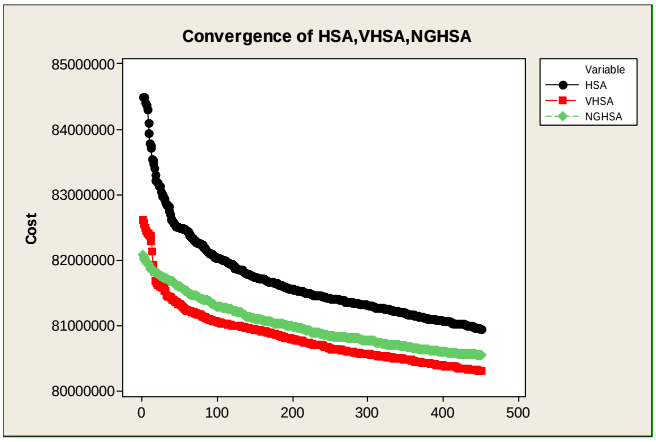

Fig. 3 Speed of Convergence on the Nominal Demand.

V. CONCLUSIONS

The APP is concerned with the determination of production, the inventory and the workforce levels of a company on a finite time horizon. Objectives are to reduce the total production cost to fulfill a non-constant demand and to reduce subcontracting units assuming fixed sale and production capacity. In this study we proposed an application of a fuzzy programming approach to the aggregate production planning with various level of the demand.

The proposed model attempts to minimise total production cost and subcontracting units so that in the end the organisation gets the optimal production plan with the overall highest levels of decision making satisfaction. The major limitations of the proposed model concern the assumptions made in determining each of the decision parameters, with reference to production cost, forecasted demand, maximal work force levels, and production resources. Hence, the proposed model must be modified to

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

make it better suited to practical applications. Future researchers may also explore the fuzzy properties of decision variables, coefficients and relevant decision parameters in the APP problems.

Various hybridisations of the HSA for solving the APP problems with a fuzzy programming approach. VHSA and NGHSA employ the variable neighbourhood search and novel global best algorithms for generating new solution vectors that enhances accuracy and convergence rate of harmony search algorithm. In this paper the impacts of algorithm parameters on the harmony search algorithm such as BW are discussed and a strategy for tuning these parameters is presented. The VHSA has been successfully applied to this benchmarking engineering optimisation problem. Numerical results reveal that the VHSA can find better solutions when compared to others and is a powerful search algorithm.

REFERENCES

[1] A. Techawiboonwong, "Aggreate Production Planning and Master Production Scheduling : Problems and Methods for Labor-Intensive Thai Industries", *Ph.D. Dissertation*, Dept. Ind. Eng., SIIT Thammasart University, Thailand, 2003.

[2] M. Belmokaddem, M. Mekidiche and A. Sahed, "Application of a Fuzzy Goal Programming Approach with Different Importance and Priorities to Aggregate Production Planning", *Journal of Applied Quantitative Method*, vol.4, no.3, 2009.

[3] B. Fahimnia, L. Luong and R. Marian, "Modeling and Optimisation of Aggregate Production Planning - A Genetic Algorithm Approach" , *International Journal of Mathematical and Computer Sciences*, vol. 1, 2005.

[4] Ö. Baykoç and Ü. Sakalli, "An Aggregate Production Planning Model for Brass Casting Industry in Fuzzy Environment", *Engineering and Technology*, vol. 52, 2009.

[5] Y. Chen, "A Production Planning Problem Solved by the Particle Swarm Optimisation", *International Multiconference of Engineers and Computer Scientist 2010.*

[6] S. Leung and Y. Wu,"A Robust Optimisation Model For Stochastic Aggregate Production Planning", *Production Planning & Control*, vol. 15, no. 5, 2004, pp.502–514.

[7] R. Aliev, B. Fazlollahi, B. Guirimov and R. Aliev," Fuzzy-genetic Approach to Aggregate Production–Distribution Planning in Supply Chain Management", *Information Sciences,* vol. 177, 2007, pp. 4241–4255.

[8] H.J. Zimmermen," *Fuzzy Set Theory and Its Application*", Kluwer Academic Publisher, 1992, pp. 336–339.

[9] H.J. Zimmermen, "Applications of Fuzzy Sets Theory to Mathematical Programming", *Information Science,* vol. 35, 1985, pp. 29-58.

[10] H.J. Zimmermen," Description and Optimisation of Fuzzy Systems", *International Journal of General Systems*, vol. 2, 1976, pp. 209–215.

[11] H. J. Zimmermen,"Fuzzy Programming and Linear Programming with Several Objective Functions", *Fuzzy Sets and Systems*, vol. 1, 1978, pp. 45–56.

[12] K.S. Lee and Z.W. Geem, "A New Meta-heuristic Algorithm for Continues Engineering Optimisation: Harmony Search Theory and Practice," *Comput. Meth. Appl. Mech. Eng.,* vol. 194, 2004, pp. 3902–3933.

[13] N. Mladenovic and P. Hansen,"Variable Neighborhood Search", *Computers and Operations Research,* vol. 24, 1997, pp 1097–1100.

[14] D. Zou, L. Gao, J. Wu, S. Li and Y. Li," A Novel Global Harmony Search Algorithm for Reliability Problems" , *Computers & Industrial Engineering*, vol.58, 2010, pp. 307–316.

[15] M. Mahdavi, M. Fesanghary and E. Damangir, "An Improved Harmony Search Algorithm for Solving Optimisation Problems", *Applied Mathematics and Computation,* vol. 188, 2007, pp. 1567–1579.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)