CHAPTER 3

HYBRID PARTICLE SWARM OPTIMIZATION

3.1 Particle Swarm Optimization

PSO has been developed by Eberhart and Kenedy in 1995 [34]. It is a form of swarm intelligence in which the behavior of a biological social system like a flock of birds or a school of fish is simulated. The PSO provides a population-based search procedure in which called particles change their position. The position of each particle is represented in X-Y plane. Each particle physically moves to the new position using velocity according to its own experience, called $Pbest$, and according to the experience of a neighboring particle, called $Gbest$, which use of the best position encountered by itself and its neighbor. The modification of searching point is shown in Figure 3.1.

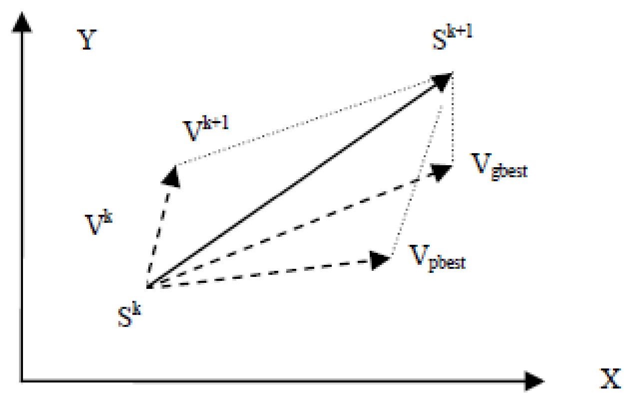

Figure 3.1 Concept of searching point by PSO.

The flowchart of PSO is shown in Figure 3.1, which can be described as follows.

Step 1: Generation of initial condition of each particle. Initial searching point (s_i^0) and velocity (v_i^0) of each particle are usually random selected within the allowable range. The current searching point is set to $Pbest$ for each particle. The best evaluated value of $Pbest$ is set to $Gbest$, and the best value is stored.

Step 2: Evaluation of searching point of each particle. The objective function value is calculated for each particle. $Pbest$ and $Gbest$ are assigned.

Step 3: Update particle velocity and position. Velocity of each particle can be modified by (3.1).

$$v_i^{k+1} = w^k \times v_i^k + c_1 \times rand_1 \times (p_{besti} - s_i^k) + c_2 \times rand_2 \times (g_{best} - s_i^k) \quad (3.1)$$

where

v_i^k	velocity of particle i at iteration *k*,
w^k	weight factor,
c_1 and c_2	weighting coefficients
$rand_1$ and $rand_2$	random number between 0 and 1,
s_i^k	current positions of particle i at iteration k,
p_{besti}	best position of *i*th particle up to the current iteration, and
	best overall position found by the particles up to the current iteration.

Weight function is given by (3.2).

$$w^k = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (3.2)$$

where

$w_{\max}$	max weight, it is equal 0.9,
$w_{\min}$	min weight, it is equal 0.4,
$iter_{\max}$	maximum iteration number, and
$iter$	current iteration number.

The first term of (3.1) is the previous velocity of the particle. The second and the third terms are utilized to change the velocity of the particle. Without the second and third terms, the particle will keep on “searching” in the same direction until it hits the boundary. On the other hand, without the first term, the velocity of the “searching” particle is only determined by using its current position and its best positions in history. The new position can be modified by (3.3).

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{3.3}$$

The current searching point of each particle is changed using (3.1), (3.2), and (3.3).

Step 3: Evaluation of searching point of each particle. The objective function value is calculated for each particle.

Step 4: Checking the exit condition. The current iteration number reaches the pre-determined maximum iteration number, then exits. Otherwise the process proceeds to step 2.

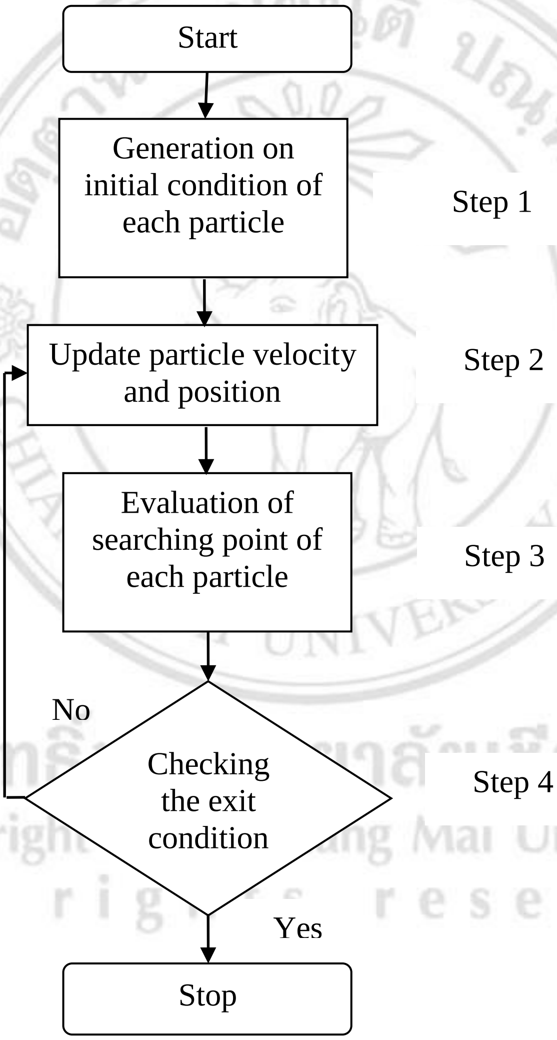

Figure 3.2 The flowchart of PSO.

3.2 Evolutionary Programming

Evolutionary Programming (EP) is presented by Fogel *et. al.*[35]. EP is probably the oldest evolutionary algorithms used to solve optimization problems. EP adapts the concept of natural evolution or survival of the fittest founded in Darwinian Theory of evolution to enhance the population or "solutions" quality. The EP paradigm emphasizes the relationship between ancestors (parents) and their descendants (offspring) and it relies exclusively on a mutation operator to produce offspring, in additional, there is no recombination operator. The flowchart of EP is shown in Figure 3.3. The main components of the algorithm are briefly explained as follows.

Step 1: The initial population is initialized randomly using sets of uniform random number distribution ranging over the limitation of each control variable as (3.4).

$$x_i = x_i^{\min} + u\left(x_i^{\max} - x_i^{\min}\right) \tag{3.4}$$

where

x_i	value of the *i*th element,
$x_i^{\min}, x_i^{\max}$	lower and upper limits of the *i*th element, and
u	uniform random number in the interval [0,1].

Step 2: A new population is generated by using Gaussian mutation operator. Each element of the new trial solution vector *k*, $V_k^{'}$ is computed by using 3.5 and 3.6.

$$x'_{k,i} = x_{k,i} + N\left(0, \sigma_{k,i}^2\right) \tag{3.5}$$

$$\sigma_{k,i} = \left(x_i^{\max} - x_i^{\min}\right)\left(\frac{f_{\max} - f_k}{f_{\max}} + a^g\right) \tag{3.6}$$

where

$x'_{k,i}$	value of the *i*th element of the *k*th offsping individual,
$x_{k,i}$	value of the *i*th element of *k*th parent individual,
$N\left(0, \sigma_{k,i}^2\right)$	Gaussian random number with a mean of zero and standard deviation of $\sigma_{k,i}$,

$x_i^{\min}, x_i^{\max}$	lower and upper limits of the *i*th element,
f_k	fitness value of the *k*th individual,
$f_{\max}$	maximum fitness of the parent population,
a	positive constant slightly less than one, and
g	generation counter.

Step 3: Evaluation of searching point of each individual. The objective function value is calculated for each individual.

Step 4: There are wellknown selection technique such as tournament scheme, roulette wheel selection.

Step 5: Checking the exit condition. The current iteration number reaches the pre-determined maximum iteration number, then exits. Otherwise the process proceeds to step 2.

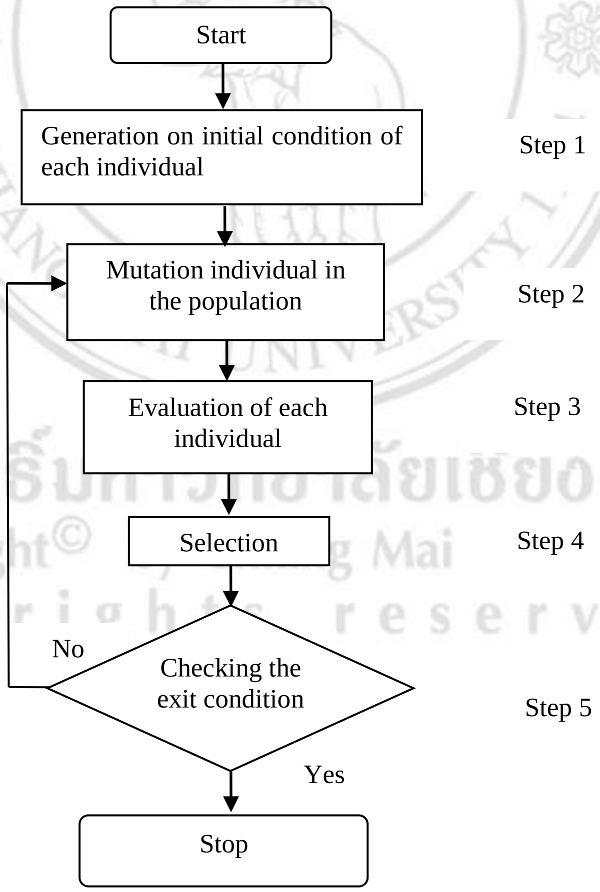

Figure 3.3 The flowchart of EP.

3.3 Tabu Search

Tabu search (TS) was originally proposed by Glover and Laguna in 1980s [36, 37] and has ever since been applied with success to a number of complex problems in science and engineering. The name tabu is related to the fact that in order to avoid revisiting certain areas of the search space that have already explored, TS algorithm turns these area to tabu.

The basic concept of TS is to escape from the optimal local value while TS evaluates the neighbor value. If the better value appeared, TS will move to that area of search space and TS will avoid the optimal loop value by updating "tabulist" which is the non-revisiting path to evaluate. The flowchart generic of TS is shown in Figure 3.4 [38]. The basic TS algorithm is as follow:

Step 1: Generate an initial solution

Step 2: Checking the stop criteria satisfied? If yes, stop. Else, go to step3

Step 3: Generate neighbors of the current seed solution by a neighborhood

Step 4: Is the aspiration criterion satisfied? If yes, Store the aspiration solution as the new seed and go to step 3. Else, go to step5.

Step 5: The best neighbor which is not tabu is selected as new seed.

Step 6: Update tabulist and go to step2

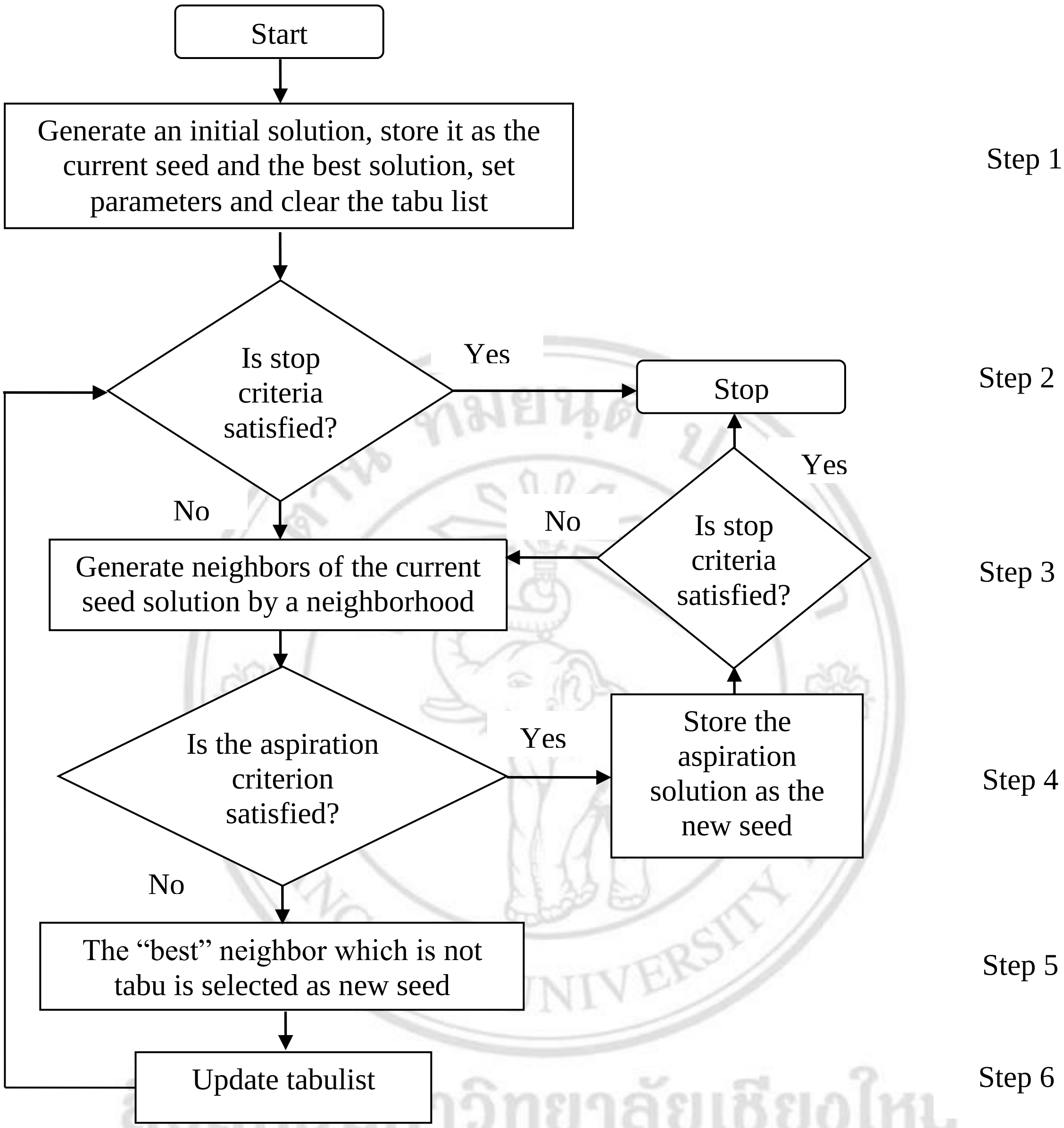

Figure 3.4 The generic flowchart of TS

3.4 Hybrid Particle Swarm Optimization

Hybrid methods are the method which made by merging of two or more single different methods. Example, hybrid solutions between the heuristics method and other modeling approaches which based on other branches of computational intelligence. Hybrid methods are proposed to merge all advantages, thereby reducing the limitation of each single method which is shown in Table 3.1.

Table 3.1 The advantages and limitation of each method

	Advantage	Limitation
PSO	- Less time consuming - give better answer	- More chance to stuck in local loop
EP	- Selection mechanism is used - Variety mutation process	- More chance to stuck in local loop - More time consuming
TS	- Less chance to stuck in local loop	- No individual function to determine parameters

The proposed method used in this study is made by merging the general PSO with EP and TS. It is based on general mix integer number of PSO where the selection from EP and the tabulist concept from TS will be used. This proposed method aimed to solve the complex system, to give optimal global answer value and to yield better efficiency than using each single method.

Therefore, this proposed method is produced as the hybrid Particle Swarm Optimization (hybrid PSO). The hybrid PSO is an integrated approach between PSO, EP, and TS, where PSO is used as a main algorithm. The general flowchart of hybrid PSO is presented in Figure 3.5

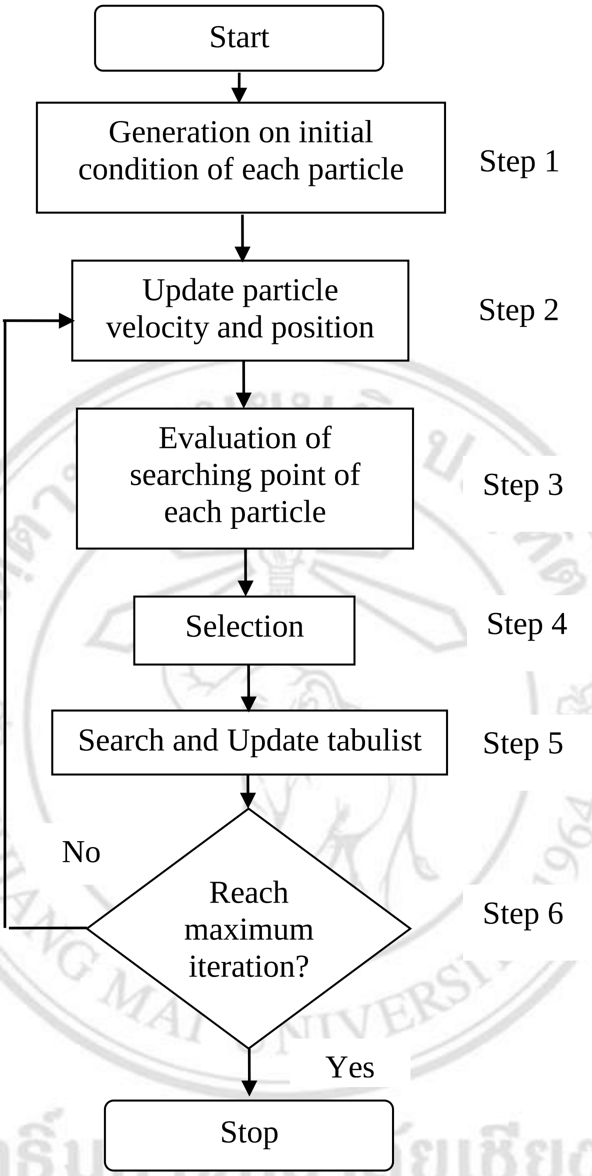

Figure 3.5 A flowchart of hybrid PSO

The main components of the algorithm are briefly explained as follows.

Step 1: Generation of initial condition of each particle. Initial searching points (s_i^0) and velocity (v_i^0) of each particle are usually random within the allowable range. The current searching point is set to $Pbest$ for each particle. The best evaluated value of $Pbest$ is set to $Gbest$, and the best value is stored.

Step 2: Update particle velocity and position. The current searching point of each particle is changed using general velocity equation of PSO in (3.7).

$$v_i^{k+1} = w^k \times v_i^k + c_1 \times rand_1 \times (p_{besti} - s_i^k) + c_2 \times rand_2 \times (g_{best} - s_i^k) \tag{3.7}$$

where

v_i^k	velocity of particle i at iteration k,
w^k	weight function,
c_1 and c_2	weighting coefficients both equal to 2,
$rand_1$ and $rand_2$	random number between 0 and 1,
s_i^k	current positions of particle i at iteration k,
p_{besti}	best position of particle i th up to the current iteration, and
g_{best}	best overall position found by the particles up to the current iteration.

Weight function is given by (3.8)

$$w^k = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{3.8}$$

where

$w_{\max}$	max weight, it is equal to 0.9,
$w_{\min}$	min weight, it is equal to 0.4,
$iter_{\max}$	maximum iteration number, and
$iter$	current iteration number.

The particle is moved to the new position by (3.9).

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{3.9}$$

Step 3: Evaluation of searching point of each particle. The objective function value is calculated for each particle. If the value is better than the current $Pbest$ of the particle, the $Pbest$ value is replaced by the current value. If the best value of $Pbest$ is better than the current $Gbest$, $Gbest$ is replaced by the best value and the best value is stored.

Step 4: Selection. The utilization technique is a tournament scheme, which can be computed by using (3.11) and (3.12).

$$w_t = \begin{cases} 1 & if \quad f_k > f_r \\ 0 & otherwise \end{cases} \tag{3.11}$$

$$s_k = \sum_{t=1}^{Nt} w_t \tag{3.12}$$

where

w_t weight value of each opponent,

f_k objective value of the *k*th particle,

f_r the *r*th opponent randomly selected from the combined particle group based on $r = \lfloor 2 * P * u + 1 \rfloor$,

$\lfloor x \rfloor$ the greatest integer less than or equal x,

u uniform random in the interval [0,1],

P particle group size,

s_k total score of each *k*th particle, and

Nt number of the opponents.

Step 5: Tabu list. This step may be viewed as a "meta-heuristic" superimposed on another heuristic method. It is designed to jump out of the local optimal and prevent the cycling movement. It stores movement of solution and forbids backtracking to previous movement [39].

Step 6: Checking the exit condition. The current iteration number reaches the pre-determined maximum iteration number, then stop. Otherwise the process proceeds to step 2.

Special features and merits of the proposed hybrid PSO can be described as follows:

1. Hybrid PSO has a balanced mechanism and adaptation to the global and local exploration abilities. This is the ability of PSO which is the base of developed method.
2. Hybrid PSO integrated with tournament scheme which is used for competition between the parent and offspring particles.
3. A probabilistic updating strategy is integrated in to hybrid PSO. TS is applied to avoid dependency on fitness function and to avoid being trapped in local optimal solutions.