Bee Parameter Determination via Weighted Centriod Modified Simplex and Constrained Response Surface Optimisation Methods

P. Luangpaiboon

Abstract—Various intelligences and inspirations have been adopted into the iterative searching process called as meta-heuristics. They intelligently perform the exploration and exploitation in the solution domain space aiming to efficiently seek near optimal solutions. In this work, the bee algorithm, inspired by the natural foraging behaviour of honey bees, was adapted to find the near optimal solutions of the transportation management system, dynamic multi-zone dispatching. This problem prepares for an uncertainty and changing customers' demand. In striving to remain competitive, transportation system should therefore be flexible in order to cope with the changes of customers' demand in terms of in-bound and out-bound goods and technological innovations. To remain higher service level but lower cost management via the minimal imbalance scenario, the rearrangement penalty of the area, in each zone, including time periods are also included. However, the performance of the algorithm depends on the appropriate parameters' setting and need to be determined and analysed before its implementation. BEE parameters are determined through the linear constrained response surface optimisation or LCRSOM and weighted centroid modified simplex methods or WCMSM. Experimental results were analysed in terms of best solutions found so far, mean and standard deviation on the imbalance values including the convergence of the solutions obtained. It was found that the results obtained from the LCRSOM were better than those using the WCMSM. However, the average execution time of experimental run using the LCRSOM was longer than those using the WCMSM. Finally a recommendation of proper level settings of BEE parameters for some selected problem sizes is given as a guideline for future applications.

Keywords—Meta-heuristic, Bee Algorithm, Dynamic Multi-Zone Dispatching, Linear Constrained Response Surface Optimisation Method, Weighted Centroid Modified Simplex Method

I. INTRODUCTION

METAHEURISTICS are more complicated due to constraints of the algorithm itself not of the question. These constraints or their parameters are needed to be initialised to optimise the outcome of the solution, or in other word, constraints directly affect the quality of the solution. So it is in turn inspiring an objective of this paper to examine the relation of constraints adjacent to the quality of solution of a chosen metaheuristic algorithm, Bee algorithm (BEE). Two alternatives to determine the optimum of surfaces consist of

P. Luangpaiboon is an Associate Professor, 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: lpongch@engr.tu.ac.th]

two treatments; linear constrained response surface optimisation (LCRSOM) and weighted centroid modified simplex (WCMSM) methods. Inspection and analysis are used to determine a recommendation on the proper levels of parameter settings for the transportation system of dynamic multi-zone dispatching.

Nowadays transportation systems have a significant role toward business systems and organisations; especially in the companies that operate a transportation business. They may not only operate it by themselves, but also need support from other transportation companies. This support with a proper management system could reduce the cost for business organisations abundantly. Besides, most big companies use specific operators to liberate the burden of transportation cost. Meanwhile, it is important to have further research for this matter in order to generate a procedure to bring about great efficiency in transportation and the objective of business, to gain profits [1].

The prior transportation system uses a general approach which is a single zone transportation approach from point to point. The single zone approach is found to spend more time with the long distance part of each journey and has lots of available space to travel back. Later on, Taylor and Meinert [2] conducted research to increase the efficiency of transportation. They mentioned the zone dispatching or zone expedition approach that will be easier to manage and control. Furthermore, Taylor and teams [3] proposed a multi-zone dispatching approach endeavoring to enhance the efficiency of the transportation system by adjusting the same point of products in and out to find the proper point of transportation to minimise the imbalance scenario.

In the multi-zone dispatching management, it consists of two main principles in transportation management, i.e. area and zone. Taylor and teams proposes the notion called *Minimal Imbalance Scenario* approach in terms of load which contains two parts; in-bound and out-bound goods in each area within each zone to find out the harmonious balance between goods. Among previous studies the minimal imbalance scenario is the most effective when compared to others. Then, minimal imbalance of load transferring between zones becomes an important issue for the multi-zone dispatching system [4].

The objective of this paper is to investigate the performance of the algorithmic approaches on the dynamic nature of the MZD model. A simulation study is based on the data from Thai local transportation firms. It aims to enhance the efficiency of transportation and pay more attention to the harmonious balance between cost and quantity. The algorithm

to be applied to these problems could respond to the complication of a change of internal structure. This paper is organised as follows. Section II describes the dynamic multi-zone dispatching. Sections III, IV and V are briefing about algorithms of Bee, linear constrained response surface optimisation and weighted centroid modified simplex methods, respectively. Section VI shows design and analysis of computational experiments for comparing the performance of the proposed methods. The conclusion is also summarised and it is followed by acknowledgment and references.

II. DYNAMIC MULTI-ZONE DISPATCHING (DMZD)

In fact, business conditions are constantly changing. The need of new quantity of orders, product lines, and technological advance or a dynamic nature of the multi-zone dispatching problems is proposed. There are a series of data in a static problem with its own "in-bound and out-bound freight" matrix for given finite discrete time periods. A period can be given in terms of months, quarters, or years. An additional rearrangement penalty in the objective function ties the static problems together whenever any area moves to the different zone in a consecutive time period. The multi-zone dispatching model can be extended to the dynamic nature of this problem with the following mathematical integer programming:

$$MIN \sum_{t=1}^{T}\sum_{j=1}^{F_{it}} ZP_{jt} - \sum_{t=1}^{T}\sum_{j=1}^{F_{it}} ZN_{jt} - \sum_{i=1}^{F_{jt}}\sum_{j=1}^{F_{it}}\sum_{k=1}^{F_{it}}\sum_{t=1}^{T-1} R_{ijkt}.X_{ijt}.X_{ik(t+1)} \quad (1)$$

Subject to:

$$\sum_{i\in F_{jt}} I_{it} X_{ijt} + I_{jt} - ZP_{jt} - ZN_{jt} = 0 \quad ; \forall_{j,t} \quad (2)$$

$$\sum_{j\in F_{it}} X_{ijt} = 1 \quad ; \forall_{j,t} \quad (3)$$

$$ZP_{jt} \geq 0 \quad ; \forall_{j,t} \quad (4)$$

$$ZN_{jt} \leq 0 \quad ; \forall_{j,t} \quad (5)$$

$$ZP_{jt}, ZN_{jt} = \text{integer} \quad ; \forall_{j,t} \quad (6)$$

$$X_{ijt} = \text{binary}(0,1)\text{integer} \quad ; \forall_{j,t} \quad (7)$$

R_{ijkt} = the rearrangement penalty for area i moved from zone j to k in the consecutive time period.

In each time (t) period the equations above are used to find out the number of *Minimal Imbalance* from the sum of the remainder between ZP_{jt} and ZN_{jt} in each zone with an additional of the rearrangement penalty in consecutive time periods. ZP_{jt} in each zone is the positive valued imbalance (in loads) if the sum of in-bound goods in each zone is greater than the sum of out-bound freight. ZN_{jt} in each zone and time period is the negative valued imbalance if the sum of out-bound goods in that zone is greater than the sum of in-bound goods in that zone. ZN_{jt} equals zero when in-bound goods in that zone equal the sum of out-bound goods in that zone.

Besides, I_{it} is imbalance value of area i which came from in-bound goods of area i to minus with out-bound goods of area i. I_{jt} is imbalance value of zone j that came from in-bound goods of zone j minus with out-bound goods of zone j. F_{it} is a set of feasible zones for area i. F_{jt} is a set of feasible areas for zone j. Finally, X_{ijt} is an integer that has 2 values; 1 and 0. The result comes to 1 when area i is in zone j and it equals 0 when area i is not in zone j.

Bring data in each period of time (t) to be continuously, the alteration of zone will reflect *Minimal Imbalance* as seen in the previous testimony. A direction in searching a solution of the DMZD model has applied a method in solving problems of statistical multi-zone dispatching. Firstly considering the imbalance proposed in a form of statistical multi-zone dispatching at interval and then considering a rearrangement penalty generated from time alteration in such period by finding a series of any solutions through all intervals. The objective of this approach is to minimise an imbalance with some penalty in zone dispatching planning over all the periods of time.

The dynamic multi-zone dispatching problem (DMZD) can be viewed as a generalisation of Non-deterministic Polynomial (NP) hard problems, which means that the computational time required by the conventional optimisation algorithms to solve a very large problem is reasonably expensive and impractical. Therefore, alternative nature-inspired optimisation techniques called metaheuristics is rapidly growing and applying to solve very large combinatorial optimisation problems.

III. BEE ALGORITHM (BEE)

The difficulties associated with using mathematical optimisation on large-scale engineering problems as above have contributed researchers to seek the alternatives, based on simulations, learning, adaptation, and evolution, to solve these problems. Natural intelligence-inspired approximation optimisation techniques called meta-heuristics are then introduced. Moreover, meta-heuristics have been used to avoid being trapped in local optima with a poor value. The common factor in meta-heuristics is that they combine rules and randomness to imitate natural phenomena. They widely grow and apply to solve many types of problems. The major reason is that meta-heuristic approaches can guide the stochastic search process to iteratively seek near optimal solutions in practical and desirable computational time.

These algorithms are then received more attention in the last few decades. They can be categorised into three groups: biologically-based inspiration, e.g. Genetic Algorithm or GA [5], Neural Network or NN [6], Ant Colony Optimisation or ACO [7], Memetics Algorithm or MAs [8], Evolutionary Programming or EP [9], Differential Evolution or DE [10], Particle Swarm Optimisation or PSO [11] Shuffled Frog Leaping Algorithm or SFLA [8]; socially-based inspiration, e.g. Tabu Search or TS [12]; and physically-based inspiration such as Simulated Annealing or SA [13].

Generally, meta-heuristics work as follows: a population of individuals is randomly initialised where each individual represents a potential solution to the problem [14]. The quality of each solution is then evaluated via a fitness function. A selection process is applied during the iteration of meta-heuristics in order to form a new population. The searching process is biased toward the better individuals to increase their chances of being included in the new population. This procedure is repeated until convergence rules are reached.

A colony of honey bees can be seen as a diffuse creature which can extend itself over long distances in various directions in order to simultaneously exploit a large number of food sources [15], [16]. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees.

The foraging process begins in a colony by scout bees being sent to survey for promising flower patches. Scout bees search randomly from one patch to another. A colony of honey bees can extend itself over long distances in multiple directions of a search space. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees. When they return to the hive, those scout bees that found a patch which is rated above a certain threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance".

This dance is essential for colony communication, and contains three vital pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive or energy usage and its nectar quality rating (or fitness). This information helps the bees to find the flower patches precisely, without using guides or maps.

Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it. After waggle dancing on the dance floor, the dancer bee (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. The number of follower bees assigned to a patch depends on the overall quality of the patch.

This allows the colony to gather food quickly and efficiently. While harvesting from a patch, the bees monitor its food level. This is necessary to decide upon the next waggle dance when they return to the hive. If the patch is still good enough as a food source, then it will be advertised in the waggle dance and more bees will be recruited to that source.

Bee Algorithm is an optimisation algorithm inspired by the natural foraging behaviour of honey bees [17], [18]. Fig. 1 shows the pseudo code for the algorithm in its simplest form. The algorithm requires various influential parameters to be preset, namely: the number of scout bees (x_1), the number of patches selected out of x_1 visited points (x_2), the number of elite patches out of x_2 selected patches (x_3), the number of bees recruited for the best x_3 patches (x_4) and the number of bees recruited for the other (x_2-x_3) selected patches (x_5) including the preset values of the iterations (x_6).

The algorithm starts with the x_1 scout bees being randomly placed in the search space of feasible solutions. The fitnesses of the points visited by the scout bees are evaluated in the second step. Step 3, the scout bees are classified into various groups. In step 4, bees that have the highest fitnesses are designated as "selected bees" and sites visited by them are chosen for neighbourhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighbourhood of the selected bees, assigning more bees to search near to the best x_3 bees.

The bees can be chosen directly according to the fitnesses associated with the points they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighbourhood of the best e bees which represent more promising solutions are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm. In step 6, for each site only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This constraint is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions.

These steps are repeated until a stopping criterion is met. At the end in each iteration, the colony will have two parts to its new population – representatives from each selected patch and other scout bees assigned to conduct random searches. The algorithm has been successfully applied to different problems including of neural network optimisations, training pattern recognition, scheduled jobs for a machine, data clustering and tuning the fuzzy logic controller. Fig. 1 shows the pseudo code for the BEE in its simplest form.

Procedure of the BEE Meta-heuristic()
Begin;
  Initialise algorithm parameters:
    x_1:   *the number of scout bees*
    x_2:   *the number of sites selected out of x_1 visited sites*
    x_3:   *the number of the best sites out of x_2 selected sites*
    x_4:   *the number of bees recruited for the best x_3 sites*
    x_5:   *the number of bees recruited for the other x_2-x_3 selected sites*
    x_6:   *the number of iterations*
  Randomly initialise the bee population;
  Evaluate fitnesses of the bee population;
  While *(stopping criterion not met)*
    Form the new bee population;
    Select sites for neighbourhood search;
    Recruit bees for selected sites with more bees for better x_3 sites;
    Evaluate the fitnesses;
  End while;
End procedure;

Fig. 1 Pseudo Code of the BEE Meta-heuristic

IV. LINEAR CONSTRAINED RESPONSE SURFACE OPTIMISATION METHOD (LCRSOM)

The procedure of LCRSOM is that a hyperplane is fitted to the results from the initial 2^k factorial designs. The data from these design points are analysed. If there is an evidence of main effect(s), at some chosen level of statistical significance and no evidence of curvature, at the same level of significance, the direction of steepest descent on the hyperplane of imbalance ($\hat{y}$) is then determined by using principles of least squares and experimental designs. In order to achieve the linear mathematical model of LCRSOM is then formulated with a consideration of the feasible ranges in terms of integer lower (ILB) and upper (IUB) bounds of all influential parameters (**x**). Moreover, the number of scout bees (x_1) must be larger than the number of patches selected out of x_1 visited points (x_2), namely:

Minimise $\hat{y}$

Subject to

$x_2 < x_1$

$\text{ILB} \le \mathbf{x} \le \text{IUB}$

The next run is carried out at a point, which has some fixed distance in this direction, and further runs are carried out by continuing in this direction until no further decrease in yield is noted. When the response first increases and no improvement of two more verified yields, another 2^k factorial design will be carried out, centered on the preceding design point. A new direction of steepest descent is estimated from this latest experiment. Provided at least one of the coefficients of the hyperplane is statistically significantly different from zero, the search continues in this new direction (Fig. 2). Once the first order model is determined to be inadequate, the area of optimum is identified via a second order model or a finishing strategy.

Procedure of the LCRSOM ()
While *(termination criterion not satisfied) – (line 1)*
   Schedule activities *(when regression verification criteria not satisfy)*
      Determine significant first order model from the factorial design points;
      Determine the mathematical models of LCRSOM;
   Schedule activities
      Move along the estimated parameter levels with a step length (Δ);
      Compute imbalance values;
      If *new imbalance design point is smaller than the preceding* **then**
         move ahead with another Δ;
      Else
         Calculate two more imbalance design points to verify the descending trend;
         If
            One of which imbalance design point turn out to be smaller than preceding design point's imbalance
         Then
            Use the smallest imbalance to continually move along the same path
         Else
            Use closest preceding design point as a centre for new 2^6design;
         End if;
      End if;
   End schedule activities;
   End schedule activities;
End while;
End procedure;

Fig. 2 Pseudo Code of the LCRSOM

V. WEIGHTED CENTROID MODIFIED SIMPLEX METHOD (WCMSM)

A simplex is an *m*-dimensional polyhedron with *m*+1 vertices, where *m* is the number of decision variables for optimisation or the dimension of the search space. This sequential optimum search is based on moving away from the experiment with the worst result in a simplex consisting of *m*+1 experiments. The objective of the sequential simplex method is to drive the simplex toward the region of the factor space which is of optimal response. A new symmetrical simplex consists of one new point and *m* design points from the previous simplex or discarding the worst point and replacing it with a new point. Repetition of simplex reflection and response measurement form the basis for the most elementary simplex algorithm.

Many modifications to the original simplex algorithm have been developed. Nelder and Mead modified a basic simplex method to allow various procedures to adapt to the response surface much more readily than the original method. This method is referred to modified simplex method. MSM allows the simplex to converge more rapidly towards an optimum by expansion and multiple ways of simplex contraction along the line of conventional reflection in order to speed up the convergence. When the response is more preferable than the responses of the previous vertices, expansion with a preset expansion coefficient is applied, to stretch the move beyond the simple reflection (Fig. 3).In some cases when the response is more desirable than the worst one, but still worse than all the remaining responses, contraction with a preset contraction coefficient is applied to make the move shorten in comparison to the reflection. Moreover, massive contractions are applied when the new response gets worse than any of the previous ones. In this case the size of the simplex is reduced by contracting each of its edges to one half of its previous length toward the vertex producing the best response [12]. A new simplex is thus generated with *m* new measurements, and the sequential optimisation procedures are repeated.
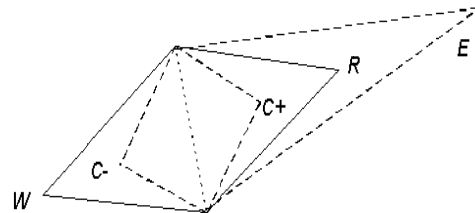

Fig. 3 Different Simplex Moves from the Rejected Trial Condition (W). R = Reflection, E = Expansion, C+ = Positive Contraction and C- = Negative contraction

The algorithmic details of the WCMSM are similar to the original MSM as above (Fig. 4). However, the subsequent vertex is projected with a preset reflection coefficient to the weighted centroid of the hyperface (*WC*) instead of the centroid. The new vertex is formed by the remaining simplex design points (*i*) with an opposite direction from the worst vertex (*W*), where

$$WC = \frac{\sum \left| f(i) - f(W) \right| i}{\sum \left| f(i) - f(W) \right|}.$$

Procedure of the WCMSM Meta-heuristic()
While *(termination criterion not satisfied) – (line 1)*
   Schedule activities
      Reflection of least yield W is processed;
      Compute WC, R and f(R);
      Compare response function;
      If *f(R) is highest* **then**
         Extension E will be processed;
      Else
         If *R and f(R) continue to be the least* **then**
            Reflect backward to prior point;
            Recalculate W and f(W);
            or
            Contraction C or shrinking S will be processed;
            Recalculate f(C) or f(S);
         Else
            Go to line 3;
         End if;
      End if;
   End schedule activities;
End while;
End procedure;

Fig. 4 Pseudo Code of the WCMSM

VI. COMPUTATIONAL RESULTS AND ANALYSES

In this paper, the study was conducted by applying the LCRSOM and WCMSM to determine the proper levels of BEE parameters on the dynamic multi-zone dispatching systems. Areas are assigned into the proper zone for the conventional and dynamic multi-zone dispatching systems under the minimal imbalance scenario. Load transfer in and out data was taken from the previous study. Total data set includes load in and out data from 50 areas within three time periods. The BEE parameters and their initial level from the literatures are given in Table I.

TABLE I
BEE PARAMETERS AND THEIR INITIAL LEVELS

Parameter	Symbol	Initial Value
The number of scout bees	x_1	40
The number of sites selected out of x_1 visited sites	x_2	20
The number of the best sites out of x_2 selected sites	x_3	10
The number of bees recruited for the best x_3 sites	x_4	40
The number of bees recruited for the other x_2- x_3 selected sites	x_5	20
The number of iterations	x_6	20

Iterative strategies of the LCRSOM and WCMSM have the imbalance value as a moving trigger. LCRSOM parameters are 2^6 unit^6 of the volume of the factorial design; ±5 of factorial design ranges; 1 unit of the step length; and 5% of the significance levels (α) for tests of significance of slopes; x_1, x_2, x_3, x_4, x_5 and x_6. For the computational procedures on both algorithms a computer simulation program was implemented in a Visual C#2008 computer program. A Laptop computer ASUS with Microsoft Windows 5.1 (Build 2600.xpsp_sp2_gdr.070227-2254: Service Pack of 2) was used for computational experiments throughout.

There are three problem sizes as described in Table II. Experimental results in each run will show the effectiveness of the algorithm in terms of total imbalance and the multi-zone pattern arrangement. There are five replicates in each case. The iterations replicate until the termination criteria is at the satisfaction state.

Stopping Criteria for the LCRSOM;
- Parameter default rule – when the coordinates escape from the upper or lower limit of BEE parameters, or,
- Second order rule – when the best imbalance deteriorates and,
- Regression verification rule – when a significance level of the regression of the LCRSOM is more than α.

Stopping Criteria for the WCMSM;
- Simplex size rule – when the size of the simplex is less than a preset value or,
- Yield standard deviation rule – when the standard deviation of imbalance values is larger than a preset value.

TABLE II
DMZD PROBLEM SIZES

Problem Size	Multi-zone Dispatching Problem		Symbol
	Zone	Area	
Small	3	10	S
Medium	5	30	M1
	10	30	M2
Large	5	50	L1
	10	50	L2

Based on the LCRSOM, if P-Value exceeds the 5% preset value of significance level (α), there is no effect of parameters. On the L1 problem (5 Zones and 50 areas), number of elite patches out of x_2 selected patches (x_3), the number of bees recruited for the other x_2-x_3 selected sites (x_5) and the number of iterations (x_6) were statistically significant (Fig. 5).
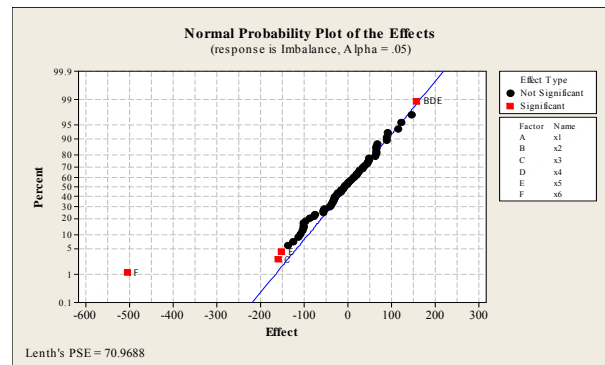

Fig. 5 Normal Probability Plot of Effects on the L1 problem (5 Zones and 50 areas)

The first order model or a linear regression is then calculated to perform the path of steepest descent via the least square method. The suitable of the first order model was determined via each linear regression coefficient, β_1, β_2, β_3 β_4, β_5 or β_6. If none of linear regression coefficient is equal to zero, all factors are significant to the model (Table III). The next step is to move a center coordinate to a new coordinate by calculating a step size and scaling with a multiplication until an imbalance could not get a better value then termination.

TABLE III
ANALYSIS OF VARIANCE (ANOVA) AND REGRESSION COEFFICIENTS AND THEIR SIGNIFICANCE FOR THE L1 PROBLEM (5 ZONES AND 50 AREAS)

Source of Variation	Degree of Freedom	Sum of Square	Mean Square	F	P-Value
Regression	6	5191774	865296	11.47	0.000*
Residual	57	4301825	75471		
Total	63	9493599			

Parameters	Coef	T-Stat	P-Value
Constant	2709938	5865.72	0.000
x_1	-10.159	-1.48	0.145
x_2	3.803	0.55	0.582
x_3	-15.934	-2.32	**0.024***
x_4	-9.766	-1.42	0.161
x_5	-15.141	-2.20	**0.032***
x_6	-50.484	-7.35	**0.000***

If P-Value exceeds the preset value of significance level, there's no effect of regression coefficients. The significant parameters which were measured by the P-Value were summarised on Table IV. If the algorithm does proceed to the next design and the only chosen one will be attributed to the prior-best- calculation. On the experimental results of the medium size problem (M1), the proper levels were determined by the best design point from the 2^k factorial design.

TABLE IV
SIGNIFICANT BEE PARAMETERS

Problem Size	x_1	x_2	x_3	x_4	x_5	x_6
S	0.094	0.356	0.588	0.052	0.453	**0.000***
M1	0.201	0.463	0.801	0.819	0.440	**0.000***
M2	0.454	0.267	0.297	**0.021***	0.332	**0.000***
L1	0.145	0.582	**0.024***	0.161	**0.032***	**0.000***
L2	0.276	0.873	0.360	**0.000***	0.914	**0.000***

On the L2 problem, the proper levels of x_4 and x_6 from the LCRSOM were 46 and 22, respectively (Fig. 6). The remaining parameters are fixed at the initial levels. The WCMSM led the parameters of (x_1, x_2, x_3, x_4, x_5, x_6) at (100**,** 50**,** 49, 100, 95, 100). The performance of both methods is not statistically significant at 95% confidence interval (Fig. 7 and Table V). The computational time and dispersion effect from the preferred BEE parameter levels from the WCMSM is taken more when compared.
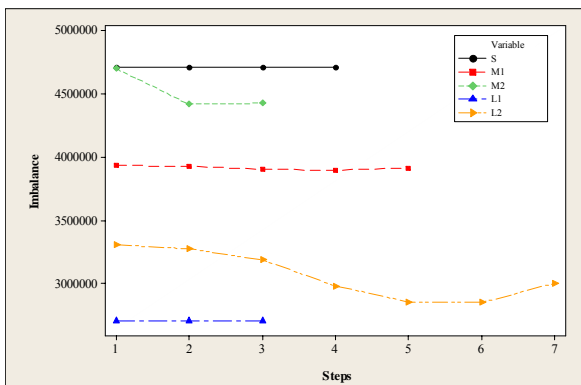

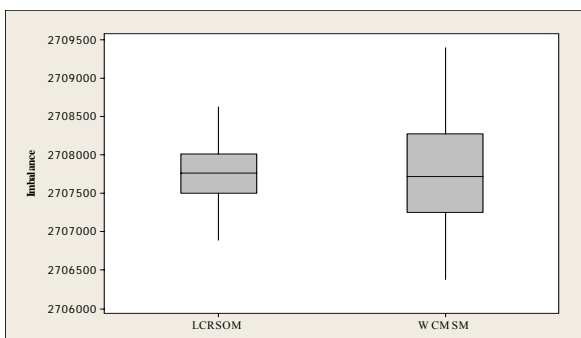Fig. 6 Imbalance Improvement via the LCRSOM Categorised by the Problem Sizes

Fig. 7 Imbalance Comparison between LCRSOM and WCMSM on the L2 Problem

TABLE V
ONE-WAY ANOVA:
IMBALANCE VERSUS TWO METHODS OF LCRSOM AND WCMSM

Source of Variation	Degree of Freedom	Sum of Square	Mean Square	F	P-Value
Factor	1	756	756	0.003	0.96
Residual	98	28799853	293876		
Total	99	28800609			

It is also stated that BEE's parameters have to only be positive integers. Consequently the process will confront with round-up error that would probably create a premature stop. When the problem sizes increase, computational time taken is also longer due to complexities of the BEE algorithm. The recommended levels of BEE parameters are summarised in Table VI. Recommended levels of parameters found by the LCRSOM seem to be more preferable and are set to be suggested levels for BEE' s parameters, to promote an ease of use in all classes of problems. However, the LCRSOM drifted at some problems and there is no directly recommended level from the path for a practical use. On the early phase, the WCMSM are more efficient for some problems. Some problems have effects of zig-zag to approach the optimum. These effects and the nature of the algorithm can terminate the final results uncertainly.

TABLE VI
EXPERIMENTAL RESULTS OBTAINED FROM RELATED METHODS IN EACH PROBLEM SIZE

Problem Size	LCRSOM		WCMSM	
	(x_1, x_2, x_3, x_4, x_5, x_6)	Imbalance	(x_1, x_2, x_3, x_4, x_5, x_6)	Imbalance
S	(40, 20, 10, 40, 20, 22)	4,711,312	(80, 40, 20, 40, 20, 50)	4,711,318
M1	(40, 20, 10, 40, 20, 21)	3,921,208	(50, 30, 10, 40, 20, 50)	3,897,397
M2	(40, 20, 10, 44, 20, 21)	4,123,082	(78, 50, 36,100, 59**,** 100)	4,022,609
L1	(40, 20, 16, 46, 26, 22)	2,706,391	(60**,** 40**,** 20**,** 40, 20, 30)	2,707,234
L2	(40, 20, 10, 46, 20, 22)	2,740,483	(100**,** 50**,** 49, 100, 95, 100)	2,708,962

Numerical results (Table VII) revealed that the bee algorithm with the proper levels on related parameters was able to obtain good solutions for all the tested cases on both the central tendency and dispersion of the imbalance values. The evolution via the imbalance values in each iteration was shown in Fig. 8. The convergence to the global optimum was quite rapid, 20 iterations on average. When the appropriate parameter settings were applied in each problem size the average execution time was approximately 55, 90 and 240 minutes, for small, medium and large testing problems respectively.

TABLE VII
MINIMAL IMBALANCE RESULTS

| Problem Size	Zone	Area	Imbalance
S	3	10	4,711,312
M1	5	30	3,921,208
M2	10	30	4,123,082
L1	5	50	2,706,391
L2	10	50	2,740,483

The additional experimental results indicated that the problem size, the number of areas and zones affect speed of

convergence. The convergences of the results obtained from the BEE were approximately 60 and 85 iterations for the medium and large problem sizes, respectively. The algorithm approached the optimum when the problem size was slightly small as appeared for the 3-zone 10-area problem (Fig. 8 (a)). Similarly, the speed of convergence increased when the number of area and zone decreased (Fig. 8 (b) and (c)).

Under a consideration of recommended levels of its parameters, those may bring the benefit to solve industrial processes via the BEE when the nature of the problems. An extension could be applied to enhance the performance of the LCRSOM and WCMSM when computational processes exceed the upper or lower limit.

REFERENCES

[1]   R.W. Holl, and V.C. Sabnani, "Control of vehicle dispatching on a cyclic route serving trucking terminals," *Transportation Research*, *Part A*, vol. 36, pp. 257-276, 2002.
[2]   G.D. Taylor and T.S. Meinert, "Improving the quality of operation in truckload trucking," *IIE Transaction*, vol. 32, no. 6, pp. 551-562, 2000.
[3]   G.D. Taylor, T.S. Meinert, R.C. Killian, and G.L. Whicker, "Development and analysis of alternative dispatching methods in truckload trucking," *Transportation Research, Part E*, vol. 35, pp. 191-205, 1999.
[4]   G.D. Taylor, G.L. Whicker and J.S. Usher, "Multi-zone dispatching in truckload trucking," *Transportation Research, Part E*, vol. 37, pp. 375-390, 2001.
[5]   K.S. Lee and Z.W. Geem, "A new meta-heuristic algorithm for continuous engineering optimisation: harmony search theory and practice," *Comput. Methods Appl. Mech. Engrg.*, vol. 194, pp. 3902–3933, 2004.
[6]   P. Muller and D.R. Insua, "Issues in bayesian analysis of neural network models," *Neural Computation*, vol. 10, pp. 571–592, 1995.
[7]   M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimisation by a colony of cooperating agents," *IEEE Transactions on Systems*, *Man, and Cybernetics Part B*, vol. 26, numéro 1, pp. 29-41, 1996.
[8]   E. Emad, H. Tarek and G. Donald, "Comparison among five evolutionary-based optimisation algorithms," *Advanced Engineering Informatics*, vol. 19, pp. 43-53, 2005.
[9]   J.Y. Jeon, J.H. Kim and K. Koh, "Experimental evolutionary programming-based high-precision control," *IEEE Control Sys. Tech.*, vol. 17, pp. 66-74, 1997.
[10] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 1, pp. 22-34, 1999.
[11] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp.58-73, 2002.
[12] A. Lokketangen, K. Jornsten and S. Storoy, "Tabu search within a pivot and complement framework," *International Transactions in Operations Research*, vol. 1, no. 3, pp. 305-316, 1994.
[13] V. Granville, M. Krivanek and J.P. Rasson, "Simulated annealing: a proof of convergence", *Pattern Analysis and Machine Intelligence*, *IEEE Transactions,* vol. 16, issue 6, pp. 652 – 656, 1994.
[14] H. Zang, S. Zhang and K. Hapeshi, "A review of nature-inspired algorithms", *Journal of Bionic Engineering*, vol. 7 (Suppl.), S232–S237, 2010.
[15] D.T. Pham, A.J. Soroka, A. Ghanbarzadeh, E. Koç, S. Otri and M. Packianather, "Optimising nNeural networks for identification of wood defects using the bees algorithm," in *Proc. 2006 IEEE International Conference on Industrial Informatics*, Singapore, 2006.
[16] D.T. Pham, E. Koç, J.Y. Lee and J. Phrueksanant, "Using the bees algorithm to schedule jobs for a machine," in *Proc. Eighth International Conference on Laser Metrology, CMM and Machine Tool Performance*, LAMDAMAP, Euspen, UK, Cardiff, 2007, pp. 430-439.
[17] D.T. Pham, S. Otri, A.A. Afify, M. Mahmuddin and H. Al-Jabbouli, "Data clustering using the bees algorithm," in *Proc. 40th CIRP Int. Manufacturing Systems Seminar*, Liverpool, 2007.
[18] L. Ozbakir, A. Baykasoglu and P. Tapkan, "Bee algorithm for generalised assignment problem," *Applied Mathematics and Computation*, vol. 215, pp. 3782-3795, 2010.
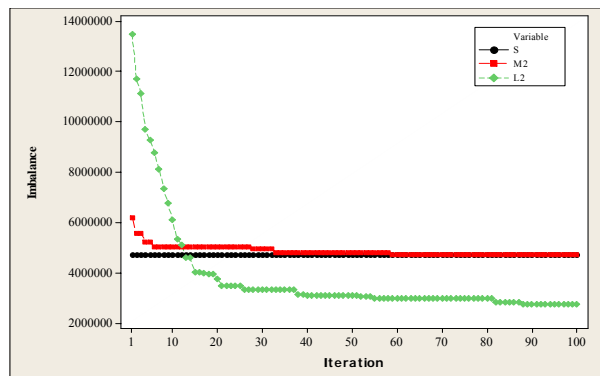
(a)
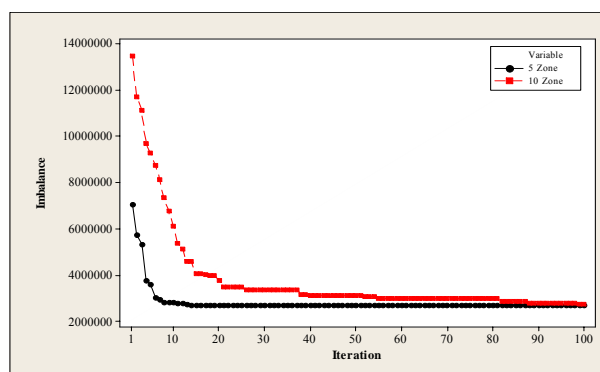
(b)
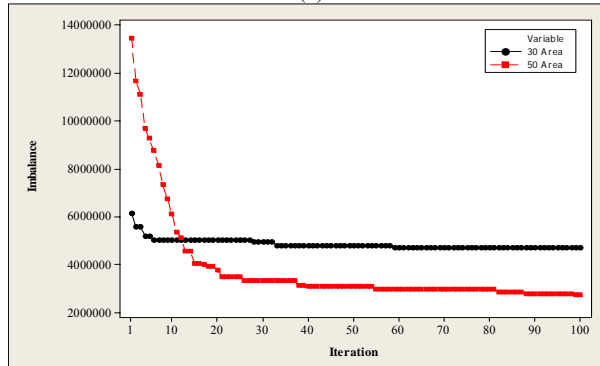
(c)

Fig. 8 Speed of Convergence of LCRSOM and WCMSM on the DMZD Categorised by Problem Sizes, Number of Zones {5 zones 50 areas VS 10 zones 50 areas} and Number of Areas {5 zones 30 areas VS 5 zones 50 areas}