Steepest Descent Algorithm to Determine the Proper Levels of Bees Parameters on Dynamic Multi-Zone Dispatching Problems

P. Mongkolkosol* and P. Luangpaiboon, *Member, IAENG*

Abstract — **Effective methods for increasing the efficiency of transportation management system is a use of dynamic multi-zone dispatching problem. This problem concentrates on the quantities of inbound and outbound in each area and it is modified from the multi-zone dispatching. The factors of the rearrangement penalty of the area, in each zone, including time periods are also included. In this research there are various levels of areas, zones, inbound, outbound and time periods. The objective of this research is to manage zones with minimal imbalance scenario via an application of the Bees algorithm. Bees algorithm is an optimisation algorithm inspired by the natural foraging behaviour of honey bees. However, the performance of the algorithm depends on its parameters levels and need to be determined and analysed before its implementation. BEES parameters are determined through the Steepest Descent method based on the statistically significant regression analysis. Experimental results were analysed in terms of best solutions found so far, mean and standard deviation on both the imbalance and execution time to converge to the optimum. Finally a recommendation of proper level settings of BEES parameters for some selected problem sizes that can be used as a guideline for future applications of the BEES. This is to promote ease of use of the BEES in real life problems. This study also found that number of zones affect iterations toward the optimum. Number of areas affects the imbalance. The parameters of zone and area are then the important variables for these multi-zone dispatching systems.**

Index Terms — **Meta-Heuristic, Bees Algorithm, Steepest Descent Algorithm, Dynamic Multi-Zone Dispatching**

I. Introduction

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].

Manuscript received December 8, 2010; revised January 20, 2011. This work was supported in part the Thailand Research Fund (TRF), the National Research Council of Thailand (NRCT) and the Commission on Higher Education of Thailand. The authors wish to thank the Faculty of Engineering, Thammasat University, THAILAND for the financial support.

*P. Mongkolkosol is an undergraduate student with 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: tor_tp28@hotmail.com, lpongch@engr.tu.ac.th].

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

Multi-Zone Dispatching (MZD)

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 inbound goods and outbound 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].

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:

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

Subject to:

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

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

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

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

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

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

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 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.

The objective of this paper is to investigate the performance of the algorithmic approach on the conventional and dynamic natures 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. Conclusions are drawn, and practical recommendations are made.

II. Bees Algorithm (BEES)

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.

Bees 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 (*n*), the number of patches selected out of *n* visited points (*m*), the number of elite patches out of *m* selected patches (*e*), the number of bees recruited for the best *e* patches (*nep*) and the number of bees recruited for the other (*m-e*) selected patches (*nsp*) including the preset values of the iterations (*i*).

The algorithm starts with the *n* 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 *e* 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 BEES in its simplest form.

Procedure ***BEES Meta-heuristic()***
Begin;
  Initialise algorithm parameters:
    i: *the number of iterations*
    n: *the number of scout bees*
    m: *the number of sites selected out of n visited sites*
    e: *the number of the best sites out of m selected sites*
    nep: *the number of bees recruited for the best e sites,*
    nsp: *the number of bees recruited for the other m-e selected sites*
  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 e sites;
    Evaluate the fitnesses;
  End while;
End procedure;

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

III. Steepest Descent Algorithm (SDA)

The procedure of SDA 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 is then determined by using principles of least squares and experimental designs. 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 SDA ()***
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;
  Schedule activities
    Move along the steepest descent's path 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 2^6*design;*
      end if;
    end if;
  end schedule activities;
  end schedule activities;
end while;
end procedure;

Fig. 2. Pseudo Code of SDA.

IV. Tested Models

In this paper, the study was conducted by applying the SDA to determine the proper levels of BEES 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. For the computational procedures described above a computer simulation was implemented in a Visual Basic program. There are three problem sizes as described in Table I. Experimental results in each run will show the effectiveness of the algorithm in terms of total imbalance and the multi-

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

zone pattern arrangement. There are five replicates in each case.

TABLE I
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

Iterative strategy of SDA has the imbalance value as a moving trigger. Parameters are 2^6 unit^6 of the volume of the factorial design; ±3 and ±5 of factorial design ranges; 1 and 2 units of the step length; 5% and 10% of the significance levels (α) for tests of significance of slopes; *n*, *m*, *e*, *nep*, *nsp* and *i*. There are four scenarios to be tested in this study (Table II). The BEES parameters and their initial level from the literatures are given in Table III.

TABLE II
FOUR SCENARIOS TO BE TESTED BASED ON THE INFLUENTIAL FACTORS OF THE SDA

Scenario	Design Range	Step Length
S1	±5	2
S2	±5	1
S3	±3	2
S4	±3	1

TABLE III
BEES PARAMETERS AND THEIR INITIAL LEVELS

Parameters	Symbols	Initial Values
The number of scout bees	*n*	40
The number of sites selected out of n visited sites	*m*	20
The number of the best sites out of m selected sites	*e*	10
The number of bees recruited for the best *e* sites	*nep*	40
The number of bees recruited for the other *m-e* selected sites	*nsp*	20
The number of iterations	*i*	20

The iterations replicate until the termination criteria is at the satisfaction state. Whilst continually checking stopping criteria, following steps below would be carried out;

Step 1: Perform a 2^6 design at an initial centre point.
Step 2: Fit a regression plane to the imbalance so that the fitted model has the form,

$$\hat{y} = \beta_0 + \beta_1 n + \beta_2 m + \beta_3 e + \beta_4 nep + \beta_5 nsp + \beta_6 i.$$

Step 3: Test whether there is evidence that either β_1, β_2, β_3 β_4, β_5 or β_6 is different from zero at the α% level of significance.
Step 4: If the result is significant, move one step along the path of steepest descent (the fitted regression line).

Stopping Criteria for the BEES;

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

V. COMPUTATIONAL RESULTS AND ANALYSES

In this work, for the computational procedures described above a computer simulation program was implemented in a Visual C#2008 computer program. A Laptop computer ASUS with Microsoft Windows version 5.1 (Build 2600.xpsp_sp2_gdr.070227-2254: service pack 2) was used for computational experiments throughout.

Based on SDA, 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 *m* selected patches (*e*), the number of bees recruited for the other *m-e* selected sites (*nsp*) and the number of iterations (*i*) were statistically significant (Fig. 3).

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 reviewed by looking at each of 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 IV). 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.

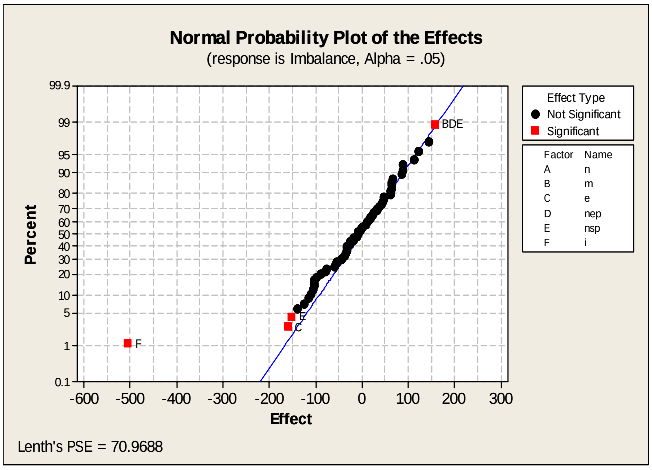


Fig. 3. Normal Probability Plot of Effects on the L1 problem (5 Zones and 50 areas) with the ±5 Factorial Design Range.

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

Source	DF	SS	MS	F	P-value
Regression	6	5191774	865296	11.47	0.000*
Residual Error	57	4301825	75471		
Total	63	9493599			

Parameters	Coef	T-Stat	P-value
Constant	2709938	5865.72	0.000
n	-10.159	-1.48	0.145
m	3.803	0.55	0.582
e	-15.934	-2.32	**0.024***
nep	-9.766	-1.42	0.161
nsp	-15.141	-2.20	**0.032***
i	-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 V. If the algorithm does proceed to the next design and the only chosen one will be attributed to the

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

prior-best- calculation. On the L2 problem and the 5% significance level, the proper levels of *nep* and *i* were 46 and 22, respectively when the chosen step length was set at 1 (Fig. 4). On the experimental results of the medium size problem (M1) with the range of the ±3 experimental design, 1 and 2 step length and α of 0.05, the proper levels were determined by the best design point from the 2^k factorial design.

TABLE V
SIGNIFICANT PARAMETERS CATEGORISED BY DESIGN RANGES

Problem Size	*n*	*m*	*e*	*nep*	*nsp*	*i*
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

Note: Factorial Design Range of ±5

Problem Size	*n*	*m*	*e*	*nep*	*nsp*	*i*
S	0.700	0.767	0.496	0.882	0.722	0.010
M1	0.997	0.063	0.443	0.122	0.253	0.056
M2	0.644	0.932	0.072	0.665	0.999	0.000
L1	0.674	0.107	0.071	0.738	0.300	0.005
L2	0.910	0.249	0.695	0.001	0.224	0.000

Note: Factorial Design Range of ±3

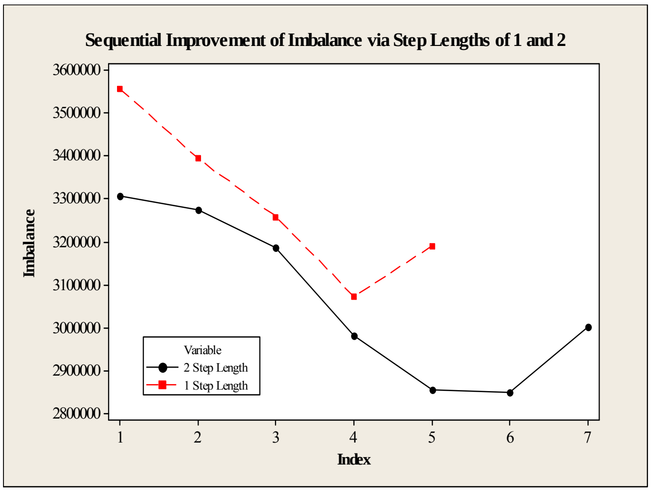


Fig. 4. Imbalance Improvement via the Path of Steepest Descent Categorised by the Levels of Step Length of 1 and 2 on the L2 Problem.

It is also stated that BEES' 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 BEES algorithm. The recommended levels of BEES parameters are summarised in Table VI.

On the experimental results of all the problems, the range of the ±5 experimental design and the step length of one seem to work well to determine the proper levels of BEES's parameters on both significance levels (α). This could be affected by the integer property of the parameters on the factorial experiments and also the restriction of stepping at longer step length on the response. Recommended levels of parameters found by the SDA are determined and are set to be suggested levels for BEES' parameters, to promote an ease of use in all classes of problems. However, the SDA drifted at some problems and there is no directly recommended level from the path for a practical use. The best design points from the experimental design were selected as the proper levels. The summary on the imbalance were given in Table VII.

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

Numerical results (Table VIII) revealed that the BEES with the proper levels on related parameters was able to obtain good solutions for all the tested cases when compared with the dynamic programming with Rosenblatt's sub-procedure, especially the speed of convergence. On the MZD the total imbalance from all problem sizes were at the optimum. The evolution via the imbalance values in each iteration was shown in Fig. 5 and 6. The convergence to the global optimum was quite rapid, 20 iterations on average.

TABLE VI
RECOMMENDED LEVELS OF PARAMETER SETTINGS

5% of the Significance Level					
Problem Size	Recommended Levels on (*n, m, e, nep, nsp, i*)	S1	S2	S3	S4
S	(40, 20, 10, 40, 20, **22**)		✓	✓	✓
M1	(40, 20, 10, 40, 20, **21**)		✓		
M2	(40, 20, 10, **44**, 20, **21**)		✓		
L1	(40, 20, **16**, 46, **26**, 22)		✓		
L2	(40, 20, 10, **46**, 20, 22)		✓		✓
10% of the Significance Level					
Problem Size	Recommended Levels on (*n, m, e, nep, nsp, i*)	S1	S2	S3	S4
S	(**46**, 20, 10, **46**, 20, 22)		✓		✓
M1	(**46**, 20, 10, **46**, 20, **20**)		✓		
M2	(40, 20, 10, **48**, 20, 22)		✓		
	(40, 20, **12**, 44, 20, 22)				✓
L1	(40, 20, **16**, 40, **26**, 22)		✓		
	(40, 20, **12**, 44, 20, **21**)				✓
L2	(40, 20, 10, **46**, 20, 22)				✓

TABLE VII
EXPERIMENTAL RESULTS OF THE TOTAL IMBALANCE CATEGORISED BY THE FACTORIAL DESIGN RANGES

±5 Factorial Design Range				
Problem Size	α = 0.05		α = 0.1	
	2 steps	1 step	2 steps	1 step
S	4,711,318	4,711,318	4,711,312	4,711,318
M1	3,897,588	3,898,570	3,904,980	3,898,132
M2	4,417,522	4,398,016	4,223,788	4,450,278
L1	2,707,348	2,707,127	2,707,348	2,707,127
L2	2,850,473	3,073,620	2,850,473	3,073,620
±3 Factorial Design Range				
Problem Size	α = 0.05		α = 0.1	
	2 steps	1 step	2 steps	1 step
S	4,711,318	4,711,318	4,711,318	4,711,318
M1	3,897,805	3,898,132	3,910,012	3,903,958
M2	4,442,191	4,627,747	4,359,359	4,266,965
L1	2,707,855	2,707,235	2,706,906	2,707,644
L2	3,115,978	3,261,933	3,115,978	3,251,933

TABLE VIII
MINIMAL IMBALANCE RESULTS

Problem Size	Zone	Area	MZD	DMZD
S	3	10	2,369,022	4,711,312
M1	5	30	1,240,214	3,921,208
M2	10	30	1,240,214	4,123,082
L1	5	50	487,741	2,706,391
L2	10	50	483,741	2,740,483

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

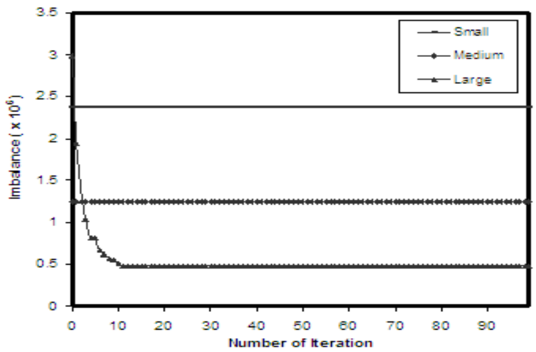
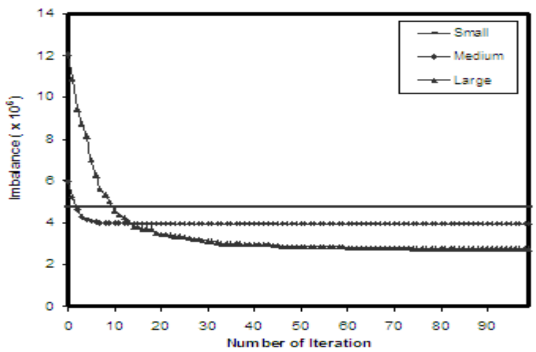
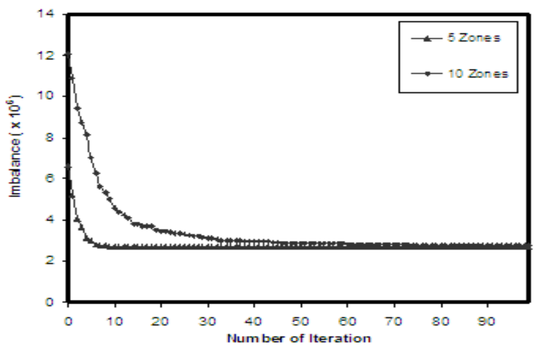


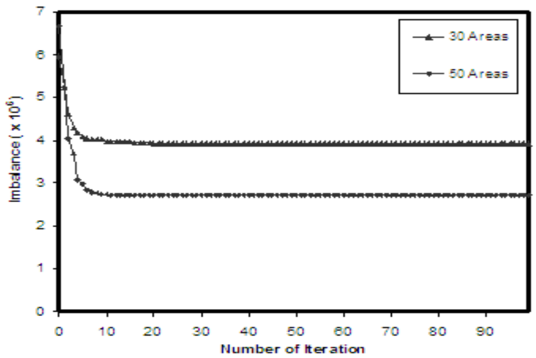
Fig.5. Imbalance Improvement on the MZD Categorised by Problem Sizes.



(a) Imbalance Improvement on the DMZD Categorised by Problem Sizes {Small: 3 zones 10 areas, Medium: 5 zones 30 areas, Large: 10 zones 50 areas}.



(b) Imbalance Improvement on the DMZD Categorised by Number of Zones {5 zones 50 areas VS 10 zones 50 areas}.



(c) Imbalance Improvement on the DMZD Categorised by Number of Areas {5 zones 30 areas VS 5 zones 50 areas}.

Fig. 6. Imbalance Improvement on the DMZD Categorised by Problem Sizes.

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, 2002, pp. 257-276.

[2] G.D. Taylor and T.S. Meinert, "Improving the Quality of Operation in Truckload Trucking," *IIE Transaction*, vol. 32, no. 6, 2000, pp. 551-562.

[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, 1999, pp. 191-205.

[4] G.D. Taylor, G.L. Whicker and J.S. Usher, "Multi-zone Dispatching in Truckload Trucking," *Transportation Research, Part E*, vol. 37, 2001, pp. 375-390.

[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, 2004, pp. 3902–3933.

[6] P. Muller and D.R. Insua, "Issues in Bayesian Analysis of Neural Network Models," *Neural Computation*, vol. 10, 1995, pp. 571–592.

[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, 1996, pp. 29-41.

[8] E. Emad, H. Tarek and G. Donald, "Comparison among Five Evolutionary-based Optimisation Algorithms," *Advanced Engineering Informatics*, vol. 19, 2005, pp. 43-53.

[9] J. Y. Jeon, J.H. Kim and K. Koh, "Experimental Evolutionary Programming-based High-Precision Control," *IEEE Control Sys. Tech.*, vol. 17, 1997, pp. 66-74.

[10] R. Storn, "System Design by Constraint Adaptation and Differential Evolution," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 1, 1999, pp. 22-34.

[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, 2002, pp.58-73.

[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, 1994, pp. 305-316.

[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.), 2010, S232–S237.

[15] D.T. Pham, A.J. Soroka, A. Ghanbarzadeh, E. Koç, S. Otri and M. Packianather, "Optimising Neural 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, 2010, pp. 3782-3795.