Bee Parameter Determination via Weighted Centroid 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

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 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:

\[ M \sum_{i=1}^{F_i} ZP_{jt} - \sum_{i=1}^{F_i} ZN_{jt} + \sum_{i=1}^{F_i} \sum_{j=1}^{k} R_{ijkt} - X_{ijkt} \delta_{ik} \quad (1) \]

Subject to:

\[ \sum_{i \in F_j} X_{ijt} + I_{jt} - ZP_{jt} - ZN_{jt} = 0 \quad \forall j, t \quad (2) \]

\[ \sum_{j \in F_i} X_{ijt} = 1 \quad \forall i, t \quad (3) \]

\[ ZP_{jt} \geq 0 \quad \forall j, t \quad (4) \]

\[ ZN_{jt} \leq 0 \quad \forall j, t \quad (5) \]

\[ ZP_{jt} + ZN_{jt} = \text{integer} \quad \forall j, t \quad (6) \]

\[ X_{ijt} = \text{binary}(0,1)\text{integer} \quad \forall j, t \quad (7) \]

\[ R_{ijkt} \] is 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 \( ZP_j \) and \( ZN_j \) in each zone with an
additional of the rearrangement penalty in consecutive time
periods. \( ZP_j \) 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. \( ZN_j \) 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. \( ZN_j \) equals zero when in-bound goods in
that zone equal the sum of out-bound goods in that zone.

Besides, \( I_i \) is imbalance value of area \( i \) which came from
in-bound goods of area \( i \) minus with out-bound goods of
area \( i \). \( I_j \) is imbalance value of zone \( j \) that came from in-bound
goods of zone \( j \) minus with out-bound goods of zone \( j \). \( F_i \) is a
set of feasible areas for area \( i \). \( F_j \) is a set of feasible areas for
zone \( j \). Finally, \( X_{ij} \) 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 \( j \).

Bringing 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 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 Bee 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 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 other \( x_2-x_3 \) selected sites
- \( x_5 \): 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 \( x_2 \) 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
\[ x_2 < x_1 \]
\[ \text{ILB} \leq x \leq \text{IUB} \]

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)*

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;
2. **Schedule activities**
   - Move along the estimated parameter levels with a step length (\( \Delta \));
   - Compute imbalance values;
   - If new imbalance design point is smaller than the preceding then move ahead with another \( \Delta \);
   - 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 \( 2^k \) design;
     - End if;
   - End if;
   - End schedule activities;
   - End schedule activities;
3. End while;
4. 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. MSN 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 shorter 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.

![Diagram](image)

**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 MSN 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

\[
WC = \frac{\sum_{i} |f(i) - f(W)|}{\sum_{i} |f(i) - f(W)|}.
\]

**Procedure of the WCMSM Meta-heuristic()**

*While (termination criterion not satisfied) – (line 1)*

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;
2. 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.

Iterative strategies of the LCRSOM and WCMSM have the imbalance value as a moving trigger. LCRSOM parameters are $2^n$ unit of the volume of the factorial design; ±5 of factorial design ranges; 1 unit of the step length; and 5% of the significance levels ($\alpha$) for tests of significance of slopes; $x_1$, $x_2$, $x_3$, $x_6$, $x_7$ 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 $\alpha$.

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.

Based on the LCRSOM, if P-Value exceeds the 5% preset value of significance level ($\alpha$), 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 ($x_1$), the number of bees recruited for the other $x_2$-$x_3$ selected sites ($x_4$) and the number of iterations ($x_5$) were statistically significant (Fig. 5).

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, $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$, $\beta_5$ or $\beta_6$. 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 than termination.
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.

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 Problem Sizes](Image)

![Fig. 7 Imbalance Comparison between LCRSOM and WCMSM on the L2 Problem](Image)

### TABLE IV

**SIGNIFICANT BEE PARAMETERS**

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0.094</td>
<td>0.356</td>
<td>0.588</td>
<td>0.052</td>
<td>0.453</td>
<td>0.000*</td>
</tr>
<tr>
<td>M1</td>
<td>0.201</td>
<td>0.463</td>
<td>0.801</td>
<td>0.819</td>
<td>0.440</td>
<td>0.000*</td>
</tr>
<tr>
<td>M2</td>
<td>0.454</td>
<td>0.267</td>
<td>0.297</td>
<td>0.021*</td>
<td>0.332</td>
<td>0.000*</td>
</tr>
<tr>
<td>L1</td>
<td>0.145</td>
<td>0.582</td>
<td>0.024*</td>
<td>0.161</td>
<td>0.032*</td>
<td>0.000*</td>
</tr>
<tr>
<td>L2</td>
<td>0.276</td>
<td>0.873</td>
<td>0.360</td>
<td>0.000*</td>
<td>0.914</td>
<td>0.000*</td>
</tr>
</tbody>
</table>

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 V

**ONE-WAY ANOVA:**

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Degree of Freedom</th>
<th>Sum of Square</th>
<th>Mean Square</th>
<th>F</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor</td>
<td>1</td>
<td>756</td>
<td>756</td>
<td>0.003</td>
<td>0.96</td>
</tr>
<tr>
<td>Residual</td>
<td>98</td>
<td>28 799,853</td>
<td>298 376</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>99</td>
<td>28 800,609</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TABLE VI

**EXPERIMENTAL RESULTS OBTAINED FROM RELATED METHODS IN EACH PROBLEM SIZE**

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>LCRSOM</th>
<th>WCMSM</th>
<th>Imbalance</th>
<th>Imbalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>(40, 20, 10, 40, 20, 22)</td>
<td>(80, 40, 20, 40, 20, 50)</td>
<td>4,711,312</td>
<td>4,711,318</td>
</tr>
<tr>
<td>M1</td>
<td>(40, 20, 10, 40, 20, 21)</td>
<td>(50, 30, 10, 40, 20, 50)</td>
<td>3,921,208</td>
<td>3,897,397</td>
</tr>
<tr>
<td>M2</td>
<td>(40, 20, 10, 44, 20, 21)</td>
<td>(78, 50, 36, 100, 59, 100)</td>
<td>4,123,082</td>
<td>4,022,609</td>
</tr>
<tr>
<td>L1</td>
<td>(40, 20, 16, 46, 26, 22)</td>
<td>(60, 40, 20, 40, 20, 30)</td>
<td>2,706,391</td>
<td>2,707,234</td>
</tr>
<tr>
<td>L2</td>
<td>(40, 20, 10, 46, 20, 22)</td>
<td>(100, 50, 49, 100, 95, 100)</td>
<td>2,740,483</td>
<td>2,708,962</td>
</tr>
</tbody>
</table>

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 also 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**

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>Zone</th>
<th>Area</th>
<th>Imbalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>3</td>
<td>10</td>
<td>4,711,312</td>
</tr>
<tr>
<td>M1</td>
<td>5</td>
<td>30</td>
<td>3,921,208</td>
</tr>
<tr>
<td>M2</td>
<td>10</td>
<td>30</td>
<td>4,123,982</td>
</tr>
<tr>
<td>L1</td>
<td>5</td>
<td>50</td>
<td>2,706,391</td>
</tr>
<tr>
<td>L2</td>
<td>10</td>
<td>50</td>
<td>2,740,483</td>
</tr>
</tbody>
</table>

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


