Modified Differential Evolution Algorithm for a Transportation Software Application

Naratip Supattananon and Raknoi Akararungruangkul *

Department of Industrial Engineering, Faculty of Engineering, KhonKaen University, KhonKaen 40000, Thailand; ballnaratip@kkumail.com
* Correspondence: raxaka@kku.ac.th

Received: 16 August 2019; Accepted: 10 October 2019; Published: 12 October 2019

Abstract: This research developed a solution approach that is a combination of a web application and the modified differential evolution (MDE) algorithm, aimed at solving a real-time transportation problem. A case study involving an inbound transportation problem in a company that has to plan the direct shipping of a finished product to be collected at the depot where the vehicles are located is presented. In the newly designed transportation plan, a vehicle will go to pick up the raw material required by a certain production plant from the supplier to deliver to the production plant in a manner that aims to reduce the transportation costs for the whole system. The reoptimized routing is executed when new information is found. The information that is updated is obtained from the web application and the reoptimization process is executed using the MDE algorithm developed to provide the solution to the problem. Generally, the original DE comprises of four steps: (1) randomly building the initial set of the solution, (2) executing the mutation process, (3) executing the recombination process, and (4) executing the selection process. Originally, for the selection process in DE, the algorithm accepted only the better solution, but in this paper, four new selection formulas are presented that can accept a solution that is worse than the current best solution. The formula is used to increase the possibility of escaping from the local optimal solution. The computational results show that the MDE outperformed the original DE in all tested instances. The benefit of using real-time decision-making is that it can increase the company’s profit by 5.90% to 6.42%.

Keywords: vehicle dispatching problem; inbound transportation problem; modified differential evolution algorithm

1. Introduction

The Internet of Things, cloud technology, big data, and web and mobile applications are technologies used to increase the efficiency of the supply, production, and transportation chains. They allow for the application of information technology (IT) to current management knowledge. The data required to achieve successful decision-making are obtained from IT and used by decision-makers to make more effective real-time decisions to gain the maximum profit for the company.

Nowadays, IT is widely used in transportation management, such as for the application of the geographic information system (GIS) in truck tracking systems, automatic inventory management systems, online procurement systems, and so forth. In this research, we apply an IT system that supports a company’s transportation planning so that the depot will send vehicles to pick up finished goods from production plants, and sometimes the depot has to send trucks to pick up raw materials from suppliers and deliver them to production plants. During the working day, or the planning period, the demand from suppliers and the amount of finished goods may change, which can affect the current transportation plan. This happens because of new demands from the production plant, the capacity of trucks may be not suitable for the new demand, the current positions of trucks can be
rerouted if new information appears in the system, and so forth. The problem statement explained above is known as the vehicle dispatching problem (VDP).

Figure 1 shows the original optimal route at the beginning of the plan. After the producer or the supplier provides new information to the depot, the new optimal travel plan for the trucks will be released and the trucks will receive the information regarding changing their routes. Trucks will get the information of new routes when they arrive at their current destination. If there is no change, then drivers will continue driving to the next destination, where they get the latest information from the depot.

At the beginning of the planning period, all pickup and delivery transportation plans will be given to the truck drivers. They will start their work with the current transportation plan. The information for decision-makers to use as the data for reoptimizing the transportation plan comes during the working period (within the transportation planning horizon). The trucks will get the information for a new route if they need to change their next destination when they arrive at their current destination. The information (e.g., quantity of finished goods available to be picked up and new demand for raw material from the supplier) is updated from the production plants. This information will be transferred to the depot, which will use the proposed method to solve the problem. In the case of increased profit, the new transportation plan will be sent to drivers using the web application, but if the profit is not increased, drivers will not be informed, and they will continue driving according to their current plan. Figure 2 explains how the information flows during the reoptimization process.

Apart from the use of information technology, the depot calculates the daily optimal travel plan for drivers and lets them drive their trucks until the daily plan is finished. Using information technology, the travel plan can be reoptimized to generate better profits than would be achieved with the original version of the travel plan. The transportation plan will support the production line to have enough raw material to support the production plan. The supplier can generate more profit due to the ability to sell more raw materials, and this system will represent a significant innovation for
many supply chain businesses, thus it will become an example of supply chain 4.0 and the idea of "smart transportation" will be implemented in many organizations.

This paper is organized as follows: Section 2 presents the literature review, which describes papers related to this research. Section 3 presents the mathematical model formulated for the problem used in the reoptimization process. The proposed methods are shown in Section 4, while Sections 5 and 6 present the computational results and conclusion, respectively.

![Figure 2. Information flow during the reoptimization process.](image)

### 2. Literature Review

Transportation costs form a major proportion of the total costs incurred by any factory or production and distribution organization. They not only affect the company's overall costs and profits, but are also related to fuel use, which can affect the global warming problem. Addressing this issue can enhance the reputation of an organization, as well as increase profits. Accordingly, transportation cost is the first cost item that many organizations focus on when seeking to reduce expenditures.

The term “taxi dispatching problem” (TDP) has arisen to identify the modern transportation problem. Most of the research is focused on using the Internet of Things to manage requests for taxis from customers while maximizing profitability for drivers [1]. A driver’s profit is controlled by the idle time of the taxi during the working time. The taxi must respond immediately to passenger booking requests. Recently, most research has been focused on how to manage customer requests. Two main solution approaches have been proposed to solve the taxi problem. Customer requests can be managed by using various techniques, such as first come, first served, which has been widely used to dispatch taxis to customers [2–4], as well as “choose the closest taxi that is free,” “choose the closest taxi that is occupied,” and “choose the taxi that has been free for the longest” [5–8].

The assignment problem (AP) is a problem in which the decision-maker must match the task and the machine. Matching is required to get the lowest operating cost [9]. The extension of this assignment problem is often called the transportation problem, which not only assigns the task to the machine, but also forms the transportation sequence when there is more than one task assigned to one machine: this is called the vehicle routing problem (VRP). There are plenty of examples of the transportation problem in the literature, such as transportation mode selection (road, waterway, railway) [10,11], connecting the transport mode by assigning the inbound and outbound gates of the product [12–15], transportation planning with time windows [14,15–17], and optimal location assignment [18–21]. However, the AP aims to increase efficiency and sustainability by reducing time, cost, fuel consumption, and carbon dioxide emissions, increasing customer satisfaction levels and profits, and making other improvements.

Dantzig and Ramser [22] first introduced the term “vehicle routing problem” (VRP) in 1959. Their article tried to manage vehicles of different sizes and capacities to deliver oil to gas stations in
order to minimize the travel distance. VRP is a problem in which we have to create the travel routes that vehicles must take to reach various destinations. The number of destinations, the demand, and other information must be known in advance. Then the optimal route is planned. In the real world, there are many problems that are similar to VRP but do not quite show the same behavior. Many articles propose various types of VRPs, such as VRP with time windows [23–32], heterogeneous fleets, the so-called multi-depot heterogeneous vehicle routing problem with time windows [25–32], VRP with pickup and delivery, and the multiple-vehicle pickup and delivery problem [33–36].

At present, the VRP is being developed to be better suited to conditions or terms that are close to reality, which makes the problem more complicated; for example, differences types of vehicles used [31,32], different vehicle speeds [37,38], congestion or different routes of transportation [39], considering weight or friction [40], and taking into account the use of sustainable resources [35]. From these points, it is inevitable that the VRP must be constantly developed.

The vehicle dispatching problem (VDP) proposed in this paper is a combination of the VRP and TDP. VDP in our terms means a problem where we have a pool of vehicles located at a depot. The depot has two functions regarding vehicle management: (1) sending trucks to pick up finished products from producers; and (2) picking up raw materials, such as bottles, from suppliers to take to producers. Originally, the depot used separate trucks so that some trucks only picked up finished products while others were used to deliver raw materials to producers. The company found that some suppliers were on their way from the depot to the producers, so if the truck stopped to pick up raw materials from those suppliers and then delivered them to the producers, it was possible to reduce the total cost.

The amount of finished product available at the producers and the amount of material that the supplier needed to deliver to the producers were determined and set as the inputs of the travel plan in the morning. During the day, the amounts of raw materials and finished products that had to be delivered could be changed. Originally, at the depot, these changes would be noted to plan for the next day. In this research we integrated this information in the current planning and reoptimize the travel plan in order to accelerate the change of information, thereby reducing the total travel cost of the company.

The vehicle dispatching problem (VDP) is harder than the TDP because the parameters can be changed or can appear when the dispatch plan has already been released and execution of the truck’s travel plan has already started. For example, when depot \( a \) assigns truck \( i \) to pick up the goods from producer \( j \) and the truck starts to travel to producer \( j \) in time \( t \), it will arrive at time \( t + T \). At time \( t + E \), when \( t + E \) is less than or equal to \( t + T \), the raw material demand of producer \( j \) from supplier \( k \) arises. Because of that, depot \( a \) must send the truck to collect raw materials from supplier \( k \) and deliver them to producer \( j \) or let truck \( i \), or any other trucks that are traveling, stop at supplier \( k \) and pick up the raw materials to deliver to producer \( j \). Therefore, the involved trucks need to be rerouted to take advantage of the new optimal route. The rerouting must be achieved within a very short time so that the travel plan remains effective.

In our approach to solving the dynamic dispatching problem, we connect the fleet vehicle operation system to the production planning software of the production plant. When the company finishes production, the information is stored in the depot system. When a particular producer places an order for raw material from the supplier, the information is also stored in the depot system, and thus, the depot can redisplay the vehicle fleet. When the system receives the information, the reoptimizing process will be executed if it generates more profit than the current situation; then, the new plan is sent to the vehicles, which have to change their travel plans. This process is executed continuously. The vehicles need to contact the depot all the time via an application designed by the depot.

Open innovation is the key to success in the new business era. Firms or organizations can reduce their research and development budget by employing technology, innovations, and knowledge from other organizations. This can help firms survive with fast-changing technology. The optimization techniques used in this paper that are applicable for use in transportation web applications are open innovations for real-time transportation planning. Open innovations in the field of optimization
techniques have been proposed in the literature in recent years. For example, Praseeratasan et al. [41] proposed the adaptive large neighborhood search (ALNS) algorithm to solve real-world multilevel scheduling and assignment problems, in which ALNS is modified to solve the problem of a broiler farm. Theeraviriya et al. [42] proposed the differential evolution (DE) algorithm to solve the location and routing problem; the innovation proposed in that study is that while considering the location and routing problem, fuel consumption is taken into account and DE is proposed to solve a special case of the generalized assignment problem [43]. Praseeratasan et al. [44] proposed ALNS to solve a real-world production planning problem. The authors used pig farming as a case study. All published articles generated excellent computational results compared to the original methods.

3. Mathematical Model Formulation

The mathematical model used for reoptimization is executed when new information becomes available.

Indices

\[ T \quad \text{Aggregate number of trucks} \]
\[ I \quad \text{Aggregate number of depots} \]
\[ J \quad \text{Aggregate number of suppliers} \]
\[ K \quad \text{Aggregate number of producers} \]
\[ t \quad \text{Truck index: } t = 1, 2, \ldots, T \]
\[ i \quad \text{Depot index: } i = 1, 2, \ldots, I \]
\[ j \quad \text{Supplier index: } j = 1, 2, \ldots, J \]
\[ k \quad \text{Producer index: } k = 1, 2, \ldots, K \]

Parameters

\[ W \quad \text{Daily wages for drivers} \]
\[ C_t \quad \text{Transportation cost per hour of truck } t \]
\[ E_t^M \quad \text{Transportation cost efficiency factor depending on type of machine and age of truck } t \]
\[ F_t \quad \text{Transportation cost of truck } t \]
\[ E_{ij}^R \quad \text{Transportation cost from depot } i \text{ to supplier } j \text{ (baht/km)} \]
\[ E_{ik}^P \quad \text{Transportation cost from depot } i \text{ to producer } k \text{ (baht/km)} \]
\[ E_{jk}^D \quad \text{Transportation cost from supplier } j \text{ to producer } k \text{ (baht/km)} \]
\[ D_{ij}^1 \quad \text{Distance from } i \text{ to } j \text{ (km)} \]
\[ D_{ik}^2 \quad \text{Distance from } i \text{ to } k \text{ (km)} \]
\[ D_{jk}^3 \quad \text{Distance from } j \text{ to } k \text{ (km)} \]
\[ P_t \quad \text{Capacity of truck } t \text{ (kg)} \]
\[ Q_{i1} \quad \text{Demand of depot } i \text{ (kg)} \]
\[ Q_{jk}^2 \quad \text{Demand of } j \text{ by } k \text{ (kg)} \]
\[ V_{tik} \quad \text{Amount of product carried by truck } t \text{ from producer } k \text{ to depot } i \]
\[ H_k \quad \text{Amount of product available at producer } k \]
\[ M \quad \text{Big number} \]
\[ V_{tijk}^2 \quad \text{Amount of product carried by truck } t \text{ traveling through supplier } j; \text{ amount of product from producer } k \text{ to depot } i \]

Decision Variable

\[ X_{tik} = \begin{cases} 1 & \text{if truck } t \text{ transports from depot } i \text{ to pick up product from producer } k \\ 0 & \text{otherwise} \end{cases} \]
\[ Y_{tijk} = \begin{cases} 1 & \text{if truck } t \text{ transports from depot } i \text{ to supplier } j \text{ and moves on to producer } k \\ 0 & \text{otherwise} \end{cases} \]
The objective function (Equation (1)) of the proposed problem is the summation of all travel costs generated during the plan. Other costs incurred are the wages of the driver, who has to pick up the products on all routes, while Equation (2) shows that one truck goes directly to pick up the product from the producer or goes via the supplier. Meanwhile, Equations (3) and (4) show that the amount of product collected must not be less than the demand of the original actor. Equations (5) and (6) address the idea that the truck can pick up the product directly from the producer or can go via the supplier, while Equation (7) shows that the product can be shifted only when the route is formed. Equation (8) shows that the depot can get the product only from the truck that picks up the product
directly from the producer or goes via the supplier location; there is no other way to pick up the product. Finally, Equation (9) shows that the product delivered from producer \( k \) must not exceed its capacity.

4. Modified Differential Evolution Algorithm

The proposed algorithm generally comprises four steps: (1) generate the initial set of the solution representing the proposed problem, (2) complete the mutation process, (3) complete the recombination process, and (4) complete the selection process. After the first round of steps 1 to 4 comes the iterative repetition of steps 2 to 4 until the predefined stopping criteria have been reached, whereupon the process stops. A stepwise explanation is given in the following section.

4.1. Generating the Set of the Initial Solution

This step was executed by designing the vector representing the problem. Then the vector was decoded to be the solution of the problem. We designed two sets of vectors to represent the problem. The first set was used to represent the truck and the second set was used to represent the supplier and producer. If we have seven trucks, three suppliers, and five producers, the vectors representing the problem are as shown in Table 1.

### Table 1. Vectors representing the problem.

<table>
<thead>
<tr>
<th>Truck No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value in position of vector</td>
<td>0.25</td>
<td>0.47</td>
<td>0.8</td>
<td>0.94</td>
<td>0.71</td>
<td>0.15</td>
<td>0.32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supplier/producer</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value in position of vector</td>
<td>0.93</td>
<td>0.81</td>
<td>0.58</td>
<td>0.79</td>
<td>0.25</td>
<td>0.84</td>
<td>0.15</td>
<td>0.21</td>
<td>0.98</td>
</tr>
</tbody>
</table>

From vector 1, there are seven trucks, three suppliers, and six producers. Note that the values in the positions of all vectors are randomly generated. Trucks 1 to 7 have a capacity of 15, 12, 15, 15, 10, 12, and 15 tons, respectively. Demand for raw materials of all producers from the suppliers is shown in Table 2 (in tons).

### Table 2. Raw material demand/capacity of producer (in tons).

<table>
<thead>
<tr>
<th>Producer/Supplier</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Cap of Producer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

The DE code shown in Table 1 needed to be decoded so that the result of the proposed problem can be revealed. The decoding procedure was as shown in Algorithm 1.

### Algorithm 1. Decoding Method

Input: supplier/producer vector; truck vector  
Output: proposed problem solution  
Begin  
Sort truck list in increasing values of position in vector \( T_k \)  
Sort supplier/producer list in increasing order of value in position of supplier/producer vector \( L_p \)
While $p$ is less than the number of suppliers + number of producers
 Assign truck ($T_k$) to supplier/producer according to their list ($L_p$) and create
 the supplier route as follows:
 1. When the producer is being assigned, the closest supplier to that producer is selected to be in
    that route.
 2. When the supplier is being assigned, the closest producer to that supplier is selected to be in
    that route.
 3. The capacity of the truck must be considered so that it is not violated in the assignment.

\[
p = p + 1;
\]

end

From the decoding step shown above, we obtained the list of trucks and suppliers/producers as
follows: The truck list obtained from step 2 was \{6, 1, 7, 2, 5, 3, 4\}, which had values in the positions
of \{0.15, 0.25, 0.32, 0.46, 0.71, 0.8, 0.94\}. The supplier/producer list was \{4, 5, 2, C, 1, B, 3, A, 6\}, which
had values in the positions of \{0.15, 0.21, 0.25, 0.58, 0.79, 0.81, 0.84, 0.93, 0.98\}. The demand for raw
materials from the supplier and the capacity of the producer can be seen in Table 2. This information
in Table 3 shows that the assignment was made.

<table>
<thead>
<tr>
<th>Truck</th>
<th>Supplier</th>
<th>Tons of Raw Material Carried</th>
<th>Producer</th>
<th>Tons of Goods Carried</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>B</td>
<td>12</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>10</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>12</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>7</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>12</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>8</td>
<td>3</td>
<td>15</td>
</tr>
</tbody>
</table>

4.2. Performing the Mutation Process

Equation (10) was used to execute the mutation process. We assumed that $V_{m,n,G}$ is a mutant
vector $m$ at position $n$ in iteration $G$. $X_{r,n,G}$ is the chosen target vector $r$ at position $n$ in iteration $G$. $F$

\[
V_{m,n,G} = X_{r1,n,G} + F(X_{r2,n,G} - X_{r3,n,G})
\]  

(10)

4.3. Performing the Recombination Procedure

When $U_{m,n,G}$ is a trial vector $m$ in position $n$ for iteration $G$, $CR$ is the selected parameter, which
is normally set at a value of 0.6–0.9. $R_n$ is randomly generated number. The trail vector generated
from the recombination process can be executed using Equation (11):

\[
U_{m,n,G} = \begin{cases} 
V_{m,n,G} & \text{and } R_n \leq CR \\
X_{m,n,G} & \text{otherwise}
\end{cases}
\]  

(11)

4.4. Performing the Selection Process

The selection process was used to reveal the next iteration target vector and was performed using Equation (12):
\[ X_{m,n,G+1} = \begin{cases} U_{m,n,G} & \text{if } f(U_{m,n,G}) \leq f(X_{m,n,G}) \\ U_{m,n,G} & \text{if } f(U_{m,n,G}) \geq f(X_{m,n,G}) \text{ and } R_n \leq P^a_g \\ X_{m,n,G} & \text{otherwise} \end{cases} \] (12)

where \( f(U_{m,n,G}) \) is the objective function of the trial vector, \( f(X_{m,n,G}) \) is the objective function of the current target vector, \( R_n \) is a random number, and \( P^a_g \) is the probability of accepting the inferior solution, which can be calculated using Equations (13)–(16):

\[ P^a_g = P_r \] (13)

where \( P_r \) is the fixed probability of acceptance, which will be randomly generated and remains constant until the termination condition is met:

\[ P^a_g = \exp\left(\frac{-f(U_{m,n,G}) - f(X_{m,n,G})}{TR}\right) \] (14)

where \( T \) and \( K \) are predefined parameters;

\[ P^a_g = 1 - \frac{C^n_g}{Max} \] (15)

where \( C^n_g \) is the current iteration number and \( Max \) is the maximum predefined number of iterations. The last equation is:

\[ P^a_g = \exp\left(\frac{-C^n_g}{Max}\right) \] (16)

5. Computational Framework and Results

The algorithm was tested with 16 randomly generated datasets that have different numbers of depots, suppliers, and producers, as shown in Table 4.

<table>
<thead>
<tr>
<th>No. of Trucks</th>
<th>No. of Depots</th>
<th>No. of Suppliers</th>
<th>No. of Producers</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS01</td>
<td>10</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>PS02</td>
<td>10</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>PS03</td>
<td>28</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>PS04</td>
<td>28</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>PS05</td>
<td>28</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>PM01</td>
<td>100</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PM02</td>
<td>100</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PM03</td>
<td>140</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PM04</td>
<td>140</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PM05</td>
<td>140</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PM06</td>
<td>140</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PM07</td>
<td>140</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>PL01</td>
<td>320</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>PL02</td>
<td>320</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>PL03</td>
<td>320</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>PL04</td>
<td>320</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

The test instances were divided into three groups: (1) small (PS01–PS05), (2) medium (PM01–PM07), and (3) large (PL01–PL04) groups of test instances. The number of iterations was used as the termination condition for all groups, set to 20,000, 40,000, and 100,000 iterations for small, medium, and large sizes of test instances, respectively.

The proposed methods were tested five times for each instance. The best solution among the five tests was selected to represent the solution using the proposed methods. There were four proposed
methods based on the differential evolution (DE) algorithm using the difference in the acceptance of the current solution of the trial vector as the new target vector when it had a lower quality than the current target vector (AC). The details of the differences in DE are shown in Table 5.

**Table 5.** Details of proposed methods.

<table>
<thead>
<tr>
<th>Name of Method</th>
<th>AC Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE-AC1</td>
<td>(13)</td>
</tr>
<tr>
<td>DE-AC2</td>
<td>(14)</td>
</tr>
<tr>
<td>DE-AC3</td>
<td>(15)</td>
</tr>
<tr>
<td>DE-AC4</td>
<td>(16)</td>
</tr>
</tbody>
</table>

The results in Table 6 are the computation results of the experiment. After testing, the best of the proposed heuristics was DE, which used Equation (14). Then, we sought to discover how Equation (14) worked with this problem by plotting the best-known solution during the simulation within 100,000 iterations, a graph of which is shown in Figure 3.

**Table 6.** Result of the experiment comparing the performance of the proposed heuristics.

<table>
<thead>
<tr>
<th>Method</th>
<th>DE-AC1</th>
<th>DE-AC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Lowest Cost Found (baht)</td>
<td>Time (hours)</td>
</tr>
<tr>
<td>PS01</td>
<td>51,569.87</td>
<td>0.00389</td>
</tr>
<tr>
<td>PS02</td>
<td>102,997.93</td>
<td>0.00528</td>
</tr>
<tr>
<td>PS03</td>
<td>311,291.47</td>
<td>0.00833</td>
</tr>
<tr>
<td>PS04</td>
<td>295,165.44</td>
<td>0.00806</td>
</tr>
<tr>
<td>PS05</td>
<td>288,409.83</td>
<td>0.00917</td>
</tr>
<tr>
<td>PM01</td>
<td>300,729.15</td>
<td>0.05250</td>
</tr>
<tr>
<td>PM02</td>
<td>486,889.87</td>
<td>0.59167</td>
</tr>
<tr>
<td>PM03</td>
<td>662,163.16</td>
<td>0.05833</td>
</tr>
<tr>
<td>PM04</td>
<td>450,545.30</td>
<td>0.05583</td>
</tr>
<tr>
<td>PM05</td>
<td>520,870.74</td>
<td>0.07278</td>
</tr>
<tr>
<td>PM06</td>
<td>583,466.27</td>
<td>0.07472</td>
</tr>
<tr>
<td>PM07</td>
<td>577,565.05</td>
<td>0.07111</td>
</tr>
<tr>
<td>PL01</td>
<td>1,703,312.31</td>
<td>0.36556</td>
</tr>
<tr>
<td>PL02</td>
<td>1,913,562.37</td>
<td>0.36722</td>
</tr>
<tr>
<td>PL03</td>
<td>1,696,113.66</td>
<td>0.41278</td>
</tr>
<tr>
<td>PL04</td>
<td>1,555,553.32</td>
<td>0.42194</td>
</tr>
</tbody>
</table>

| Instance | Lowest Cost Found (baht) | Time (hours) | Lowest Cost Found (baht) | Time (hours) |
| PS01 | 51,569.87 | 0.00389 | 51,569.87 | 0.00361 |
| PS02 | 102,997.93 | 0.00500 | 102,997.93 | 0.00528 |
| PS03 | 305,770.35 | 0.00806 | 302,519.67 | 0.00806 |
| PS04 | 295,654.19 | 0.00778 | 297,626.34 | 0.00833 |
| PS05 | 294,864.69 | 0.01000 | 288,990.86 | 0.00861 |
| PM01 | 302,169.58 | 0.05000 | 311,845.29 | 0.05111 |
| PM02 | 500,850.48 | 0.05389 | 498,958.87 | 0.05639 |
| PM03 | 676,871.89 | 0.05639 | 672,189.54 | 0.05639 |
| PM04 | 474,006.05 | 0.05056 | 473,400.72 | 0.05444 |
| PM05 | 531,173.76 | 0.05722 | 524,627.62 | 0.06278 |
| PM06 | 591,229.64 | 0.06028 | 585,625.84 | 0.06194 |
| PM07 | 585,606.96 | 0.07000 | 590,662.01 | 0.06639 |
From Figure 3, we can see DE-AC2 starts with one of the poorer solutions obtained from all proposed methods, but it is able to quickly find a better solution, and the solution always improved over time, which means that Equation (14) was very effective in terms of the required behaviors of the metaheuristics, diversification and intensification, while other equations provided diversification without intensification of the search, as was the case with DE and DE-AC1, or provided intensification but finally become stuck on a local optimum, as was the case for DE-AC3.

Then, we compared the proposed method with the optimal solution, or with the best solution that Lingo v.11 found within 240 h when the approach was not able to find the optimal solution. The outcomes of the problems are displayed in Table 7.
From the solution in Table 7, we can see that DE could find an 8.47% different result than Lingo v.11, while it used 99.93% less computation time. A statistical test using the Wilcoxon sign rank test gave a \( p \)-value equal to 0.0917. This revealed that the proposed method and Lingo v.11 did not perform differently, which meant the proposed method could find the result just as effectively as Lingo v.11 while using 99.93% less computation time.

The traditional transportation plan was to plan the transportation every morning using the proposed methods without making any change when information was obtained. In this research, the traditional approach was changed by applying the idea of the taxi dispatching problem by reoptimizing the transportation plan when information from the factory was updated. We applied the traditional transportation planning method (traditional differential evolution (TDE)) and the one proposed in this paper (MDE) to PL01–PL04. The profits obtained from the TDE and MDE are shown in Table 8. The percentage difference was calculated using Equation (17), where \( P^{\text{MDE}} \) is the profit generated by using MDE and \( P^{\text{TDE}} \) is the profit generated using TDE.

\[
% \text{diff} = \frac{P^{\text{MDE}} - P^{\text{TDE}}}{P^{\text{MDE}}} \times 100\%
\] (17)

Table 8. Comparison of results of the traditional differential evolution (TDE) and modified differential evolution (MDE).

<table>
<thead>
<tr>
<th>Method</th>
<th>TDE</th>
<th>MDE (DE-AC2)</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Daily Profit (baht)</td>
<td>Daily Profit (baht)</td>
<td></td>
</tr>
<tr>
<td>PL01</td>
<td>4,145,871.00</td>
<td>4,425,178.00</td>
<td>6.31</td>
</tr>
<tr>
<td>PL02</td>
<td>4,389,594.00</td>
<td>4,664,836.00</td>
<td>5.90</td>
</tr>
<tr>
<td>PL03</td>
<td>4,241,879.00</td>
<td>4,528,747.00</td>
<td>6.33</td>
</tr>
<tr>
<td>PL04</td>
<td>4,581,436.00</td>
<td>4,895,536.00</td>
<td>6.42</td>
</tr>
</tbody>
</table>

From the computational results shown in Table 8, we can see that IT was useful in the transportation planning process since it could increase a company’s profits by 5.9% to 6.42%.

6. Conclusions

We have presented a methodology to solve a vehicle dispatching problem. The case study involved planning to deliver raw materials to producers and take finished goods from the producers to the warehouse. When there was a signal indicating a need for raw materials from the company’s production planning software, trucks were dispatched from the distribution center immediately to pick up the products at the production plant or instructed to stop receiving the raw materials as the production plant requirements were filled. The decisions for routing the cargo were carried out continuously and more efficiently.

Currently, large and complex problems are difficult to solve exactly and accurately with a general algorithm because it takes a long time to process. In addition, for most problems, it is not possible to find the solution within a limited time (240 h), and for some problems, it is not possible to find the solution at all. To solve this problem, we developed the modified differential evolution
(MDE) algorithm. MDE improves the acceptance process or the selection of the next generation by accepting all four inferior solutions: fixed probability of acceptance (DE-AC1), basic simulated annealing probability of acceptance (DE-AC2), linear probability of acceptance (DE-AC3), and nonlinear probability of acceptance (DE-AC4). Of these four forms, the most effective method was DE-AC2.

The transportation planning problem of the case study is a problem with multiple factors, including multiple producers, raw material sources, distribution centers, and types of truck. It is a basic fact that global shipping companies focus their attention and efforts on reducing transportation costs by having optimal transportation planning. The experiments indicated that the developed MDE obtained the exact solution according to all conditions or restrictions. It took an average of 0.12813 h to get the exact solution for the biggest problem. From the resulting analysis, it can be concluded that the solution from the proposed MDE was different from that of Lingo v.11 by 8.47% but took 99.93% less time.

Reducing the transportation cost meant the use of fuel was also reduced, which made for a lower impact in terms of global warming and other environmental concerns. Therefore, the proposed method not only reduces company costs but also generates sustainable transportation plans for the benefit of the company and the environment.

The new innovation we found here is the application of IT with the proposed heuristics, which is the modified different evolution algorithm. The proposed heuristics can support real-time decision-making and increase a company’s profits by up to 6.42%. This is a new innovation that can be applicable to many types of businesses.

Author Contributions: Methodology and software, N.S.; conceptualization and resources, N.S. and R.A.; writing—review and editing, R.A.

Funding: This project this funded by the Faculty of Engineering, Khonkean University, Thailand

Acknowledgments: This paper was supported by KKU-En Grad Camp 2016, Faculty of Engineering, KhonKaen University, Thailand.

Conflicts of Interest: The authors declare no conflict of interest.

References


© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).