Thai Journal of Mathematics
Volume 18 Number 4 (2020)
Pages 1715–1731

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Application of Element Decomposing Method for Solving Traveling Salesman Problems

Ekkaphon Jaiyen[1] **and Komgrit Leksakul**[2,*]

[1] *Ph.D.'s Degree Program in Industrial Engineering, Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University*
e-mail : bang0285@gmail.com (E. Jaiyen)

[2] *Excellence Center in Logistics and Supply Chain Management, Chiang Mai University, Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University*
e-mail : komgrit@eng.cmu.ac.th (K. Leksakul)

Abstract The objective of this research study was to solve the traveling salesman problem (TSP) in order to provide a traveling sequence for the minimum total traveling time. This paper highlights the significance of creating and developing the element decomposition method (EDCM) as a part of the finite element method for solving TSP. There are two phase of research methodology. The first phase involves simplex method. The second phase is about creating and developing the algorithm through the application of the EDCM. The results obtained using the algorithm employing the EDCM were then compared with branch and bound method (B&B) and ant colony optimization (ACO) in terms of accuracy and time consumption, Regarding the problem, it can be solved with the number of cities, that is 6 to 343. The (B&B) method has the capability of resolving problems with the limitation of 22 stations. However, between ACO and EDCM, which can resolve problems for 343 stations. It found that the EDCM provides better value than ACO with an average of 1.31 % and the time consumption of 55.00 %.

MSC: 11Y16; 05C85

Keywords: traveling salesman problem; element decomposing method (EDCM)

Submission date: 21.12.2018 / Acceptance date: 07.07.2019

1. Introduction

The main transportation modes in Thailand consist of four types, namely road, water, rail, and air transport [1]. Figure 1 demonstrates that road transport is the most important transport mode for the logistics system in Thailand. However, 49% of the total road transport cost is spent on construction of roads and related facilities, as reported by the World Bank. Compared with the rail and the water transportation modes, the cost of the road transportation mode is higher, by approximately 3.5 times and 7 times, respectively [2], as shown in Figure 1.

*Corresponding author.

Published by The Mathematical Association of Thailand.

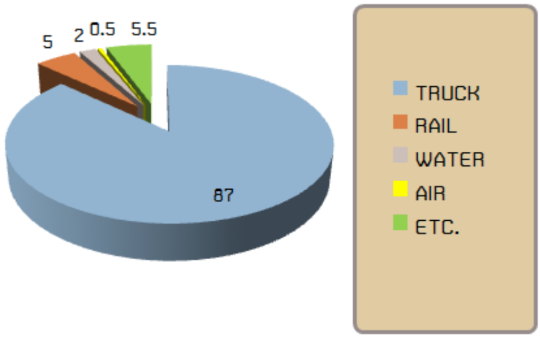


FIGURE 1. The percentages of the transportation modes in Thailand.

The traveling salesman problem (TSP) serves as a method for resolving transportation issues; it is an optimization technique that solves for the optimal transport route. This problem aims to find the shortest route answer. The principle of this technique involves delivering the product to each customer in each of the stations and coming back to the starting point [3].

The traveling salesman problem is NP-complete. It can be completed in polynomial time; thus, TSP belongs to NP. When the problem is of large size, it takes more time to solve the problem.

Operation research (OR) is the application of a model regarding mathematics, statistics, and methodology in making a decision when it is generally used in analyzing complicated systems in real situations with the purpose of developing optimal efficiency. Also, it is widely used in various fields including businesses, industries, and government sectors; hence, search for optimality is regarded as the principal duty. Operation research was initially employed during World War II when Britains military service section assigned a team of scientists to conduct research for finding out strategies and tactics to be used in national defense with regard to both land and air [4]. Consequently, Britain, after the end of World War II, was the first country to apply operation research in the fields of industry and state enterprise, and coal industry was the first industry to apply this method. Since then, the application has been extended to other industries; it is popularly used in transportation management. Algorithm is a method used in problem solving to determine the best or most approximate solution, and it can be divided into two main types including the exact method and the heuristic method. These two methods are different in both good-quality solutions and computation times. If the exact method is used in calculation, it provides the best solution with longer computation time, whereas heuristic techniques are powerful and flexible search methodologies that successfully tackle practical, difficult problems. Heuristic and metaheuristic algorithms seek to produce good-quality solutions in reasonable computation times and good enough for practical purposes.The finite element method (FEM) is a widespread method used and applied in the solid and structural mechanics of mechanical engineering [5, 6]. The FEM cuts a structure into several elements (pieces of the structure) and reconnects elements at nodes [7]. This process can cut other nodes that do not connect with the element. So, it is appropriate for application in optimization solutions which can seek answers by cutting nodes more quickly to find the answers.

Now, the problem associated with the various kinds of software that have been developed to solve mathematical problems by application algorithm is that they may find the best value but they are still limited and use a long time for processing. The present paper is an attempt to highlight the significance of creating and developing the element decomposing method as part of the finite element method for solving the optimization problem in order to quickly obtain correct answers.

2. Theory and Literature Review

2.1. Traveling Salesman Problem

The traveling salesman problem (TSP), which is an optimization technique solving for the optimal transport route, aims to find the shortest route answer. The principle of this technique involves delivering the product to each of the customers in each of the stations and coming back to the starting point.

The TSP serves as a method for resolving transportation issues, with the purpose of identifying for clients those transportation routes that will incur minimal costs. Each client receives a one-time service from the salesperson [8]. In-depth exploration of literature reviews, theories, and research identified related studies on traveling salesman problems to increase operational effectiveness, reduce traveling distance, and lower costs through integration of several techniques. Variations depend on the operations unique to each factory and business.

The TSP is one of the most popular problems among the researchers who started the development. The TSP was treated in the 1800s by the Irish mathematician Sir William Rowan Hamilton and by the British mathematician Thomas Penyngton Kirkman who solved Hamiltons Icosian Game in tours of 20 points [9]. Later, in 1930, Karl Menger, a Viennese researcher, developed the journey from the nearest neighbor heuristics during the period of 19501960, and Clarke and Wright, in 1964, also developed the method of the saving algorithm. As the TSP gained a lot of interest, development of the mathematic model and the exact method for solving the TSP also consequently started. In the beginning, Dantzig et al. (1954) developed a mathematical model and established the problem solving by the cutting plane method in order to deal with a problem consisting of 49 cities, and obtained the optimal solution. Later, Richard M. Karp (1972) showed that the TSP was NP-complete, which showed that the TSP was a difficult problem; and, there was no method that used polynomial time to successfully solve the problem. Therefore, several researchers were interested in optimal solution techniques, such as the branch-and-bound method, brute-force method, branch-and-cut method, cutting plane algorithm, column generation etc., as well as approximate solution techniques, such as nearest neighbor, greedy approach, etc. As for the metaheuristic method, one popular technique applied in solving the problem was ant colony optimization (ACO) which was started by Dorigo M. (1997) [10] and which includes local search, genetic algorithm, and particle swarm optimization. At present, the largest number of cities taken into account while solving a problem was regarded as the visit to all 24,978 cities in Sweden by David L. Applegate et al. (2004) who previously dealt with 13,509 cities in the United States in 1998, and 15,112 cities in Germany in 2001.

At present, the TSP has become more complex [11] such as the TSP with the time window which was important to the just-in-time system to deliver things punctually. In case of uncertainty of time including traveling, servicing, and awaiting [12], Tabu search is used for solving the equation [13, 14], as also in uncertainty regarding the traveling cost, etc. Sometimes, when the TSP confronts the uncertainty of anything, such as period of time for traveling, it is called the stochastic traveling salesman problem (STSP).

Additionally, the TSP can be applied in several areas such as Genome, Starlight, Scan Chains, DNA, Whizzkids, Baseball, Coin Collection, Airport Tours, USA Trip, Sonet Rings, Power Cables, etc. Obviously, the TSP has several different forms, depending on the problems occurring in those realistic situations. Consequently, this leads to a different solution to each of the different problems so that it is possible to plan a proper route and manage the reduction of the traveling cost effectively.

2.2. Integer Programming Problem

Integer programming is a branch of mathematical programming. Integer programming is optimal solution in linear programming to the integer number. There are two widely used methods for generating the special constraints that will force the optimum point of the relaxed LP problem toward the desired integer solution:

2.2.1. Branch and Bound

Branch and bound (B&B) is a widely popular method which uses tools for solving large-scale (NP-hard) combinatorial optimization problems. B&B is an algorithm paradigm which has to be filled out for each specific problem type, and there exist numerous choices for each of the components. Even then, the principles for the design of efficient B&B algorithms have emerged over the years [15]. The B&B algorithm operates according to two principles:

(1) It recursively splits the search space into smaller spaces, thus minimizing the function on these smaller spaces; the splitting is called branching.

(2) Branching alone would amount to brute-force enumeration of candidate solutions and the testing of them all. To improve the performance of the brute-force search, the B&B algorithm keeps track of the bounds on the minimum that it is trying to find, and uses these bounds to prune the search space, eliminating candidate solutions that it can prove will not contain an optimal solution.

However, the explicit enumeration is normally impossible due to the exponentially increasing number of potential solutions. The use of bounds for the function to be optimized combined with the value of the current best solution enables the algorithm to search parts of the solution space only implicitly [16]. So, when there are many variables in the solution, the time and the number of branches required are more. This method gives the performance of the best value that is accurate and very precise in the solution. This method takes a long time to process the answer.

2.2.2. Cutting Plane Method

The cutting plane method, which is an alternative to the branch-and-bound method, can also be used to solve integer programs. The fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. Of course, this method has to be carefully performed as it has constraints, and one would not want to change the problem because of the addition of the constraints.

This method will add a special type of constraint called a cut. A cut relative to a current fractional solution satisfies the following criteria [17]:

(1) Every feasible integer solution is feasible for cut, and

(2) The current fractional solution is not feasible for cut. This is illustrated in Figure 2.

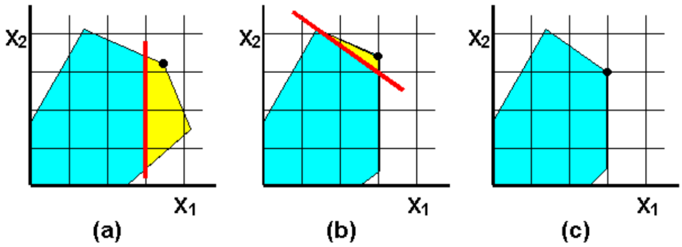


FIGURE 2. A cut of the cutting plane method. [18]

In both the methods, the added constraint will eliminate portions of the relaxed solution space, but never any of the feasible integer points. Neither of the two methods can be claimed to be uniformly more effective in solving integer linear programs (ILPs). Nevertheless, branch-and-bound methods are far more successful computationally than cutting-plane methods. For this reason, most commercial codes are based on the use of the branch-and-bound procedure.

2.2.3. Finite Element Method

The finite element method (FEM) is a numerical technique for finding approximate solutions to boundary value problems for partial differential equations; it now enjoys widespread use in solid and structural mechanics [5, 19]. It uses the subdivision of a whole problem domain into simpler parts, called finite elements, and variation methods from the calculus of variations to solve the problem by minimizing an associated error function. Analogous to the idea that connecting many tiny straight lines can approximate a large circle, FEM encompasses methods for connecting many simple element equations over many small subdomains, named finite elements, to approximate a more complex equation over a larger domain. [20].

FEM is a powerful technique originally developed for the numerical solution of complex problems in structural mechanics and it remains the method of choice for complex systems. In FEM, the structural system is modeled by a set of appropriate finite elements interconnected at points called nodes. The elements may have physical properties such as thickness, coefficient of thermal expansion, density, Youngs modulus, shear modulus, and Poissons ratio.

The beginning step is the creation of the model area (geometric construction) which defines boundaries or constraint functions to identify the areas or the shapes of the problem. Then, the model is separated into the domain of the elements, with each element consisting of a node (discretization). The next step is that of creating equations (objective functions (for solving values from the linear or the nonlinear equations. The answer consists of nodes at the edges of the area which offer the best value in objective functions. This is illustrated in Figure 3.

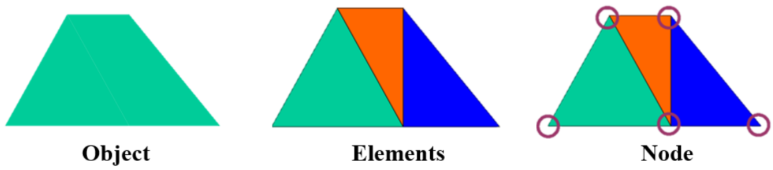


FIGURE 3. Illustration of a part in the finite element method. [20]

3. Research Methodology

The research study aims to solve an optimization problem using an algorithm that applies the decomposed element method, as shown in Figure 4.

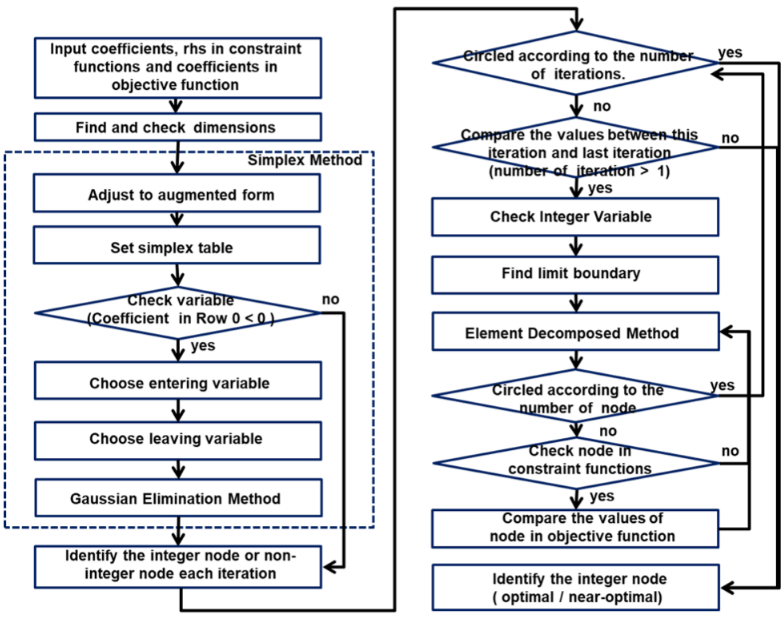


FIGURE 4. Steps of the algorithm. [21]

This research study has two phases. The first phase consists of the input data and simplex method. The second phase is about creating and developing the algorithm via application of the element decomposing method (EDCM).

Phase 1 consists of the input data and simplex method. This phase is shown at the left in Figure 4, and is performed as follows.

Step 1.1 The input coefficients, the right-hand side (RHS) in constraint functions, and coefficients in the objective function follow the mathematical equation form. In the case of the traveling salesman problem (TSP), it is the input coefficient in the objective function. The mathematical equations of the traveling salesman problem have a detailed format, as follows.

1. The objective function, which is finding the minimum total traveling distance.
2. The constraint functions which include the following equations.

- The constraint function that is the summation of the decision variables traveling from station i to station j equaling 1 (traveling from station i to station j is permitted only once).
- The constraint function that is the summation of the decision variables traveling from station j to station i equaling 1 (traveling from station j to station i is permitted only once).
- The constraint function which eliminates sub-tours.
- The constraint function which is a decision variable equal to one when traveling from any station i to any station j, and zero if that condition fails.

The equation of the traveling salesman problem can be written as follows.

$$MinZ = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\ X_{ij} \tag{3.1}$$

Equation (3.1) is the objective function to calculate the minimum total traveling distance, where d_{ij} is the mean traveling distance.

$$\sum_{j=1}^{n} X_{ij} = 1, \qquad \forall_i \ \epsilon \ n \tag{3.2}$$

Equation (3.2) is the summation of the decision variables traveling from station i to station j equaling 1.

$$\sum_{i=1}^{n} X_{ij} = 1, \qquad \forall_j \ \epsilon \ n \tag{3.3}$$

Equation (3.3) is the summation of the decision variables traveling from station j to station i equaling 1.

$$u_i - u_j + N\ X_{ij} \leq N - 1 \tag{3.4}$$

Equation (3.4) is the constraint function to eliminate sub-tours. $i \neq j$; $i, j = 2, 3, , N$, where N is the total number of stations.

$$X_{ij}\ \epsilon\ \{0, 1\} \tag{3.5}$$

Equation (3.5) is the decision variable which is equal to 1 or 0.

In the case of Example 1 in the traveling salesman problem, it has four stations, as shown in Table 1.

Table 1. Distance of Example 1

Station	**1**	**2**	**3**	**4**
1	M	6	5	13
2	6	M	17	20
3	5	17	M	27
4	13	20	27	M

$$\text{Coefficient of the objective function} = \begin{bmatrix} M & 6 & 5 & 13 & 6 & M & 17 & 20 & ... & M \end{bmatrix}$$

$$\text{Coefficient of constraint functions} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & ... & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & ... & 1 \end{bmatrix}$$

$$\text{RHS of constraint functions} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Step 1.2 Finding and checking the sizes of the coefficient of the objective function, coefficient of constraint functions, and right-hand side of constraint functions, which can be calculated using the following formulas.

Size of coefficient of objective function = 1 × 16 or 1× (number of stations * number of stations)

Size of coefficient of constraint functions = 16 × 8 or (number of stations * number of stations) × (number of stations *2)

Size of right-hand side of constraint functions = 1 × 8 or 1 × (number of stations *2)

After that, the number of columns and the number of rows in the objective function and the constraint functions are examined according to the following conditions.

1. The number of columns in the coefficients of the objective function and the coefficients of the constraint functions are equal, and equal to the number of stations *number of stations.

2. The number of rows in the coefficients of the objective function and the RHS of the constraint functions are equal, and equal to the number of stations *2.

Subsequently, write all the values of the coefficient of the objective function, coefficient of the constraint functions, and the right-hand side of the constraint functions in a simplex table adjusted to the standard form but without including the sub-tour function, as shown in Table 2. Then, adjust to the augmented form as preparation for the simplex method.

TABLE 2. Distance of Example 1

Row	X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}	X_{23}	X_{24}	X_{31}	X_{32}	X_{33}	...	RHS
0	-M	-6	-5	-13	-6	-M	-17	-20	-5	-17	-M	...	0
1	1	1	1	1	0	0	0	0	0	0	0	...	1
2	0	0	0	0	1	1	1	1	0	0	0	...	1
3	0	0	0	0	0	0	0	0	1	1	1	...	1
4	0	0	0	0	0	0	0	0	0	0	0	...	1
5	1	0	0	0	1	0	0	0	1	0	0	...	1
6	0	1	0	0	0	1	0	0	0	1	0	...	1
7	0	0	1	0	0	0	1	0	0	0	1	...	1
8	0	0	0	1	0	0	0	1	0	0	0	...	1

Step 1.3 Solving the problem using the simplex method according to the following steps.

(1) Checking the variable in row 0 and choosing the entering variable.
(2) Carrying out the ratio test and choosing the leaving variable.
(3) Using the Gaussian elimination method.

Step 1.4 Using sub-tour elimination.

From Step 1.3, while performing the step of choosing the entering variable, there is a possibility of choosing the variable that leads to the sub-tour. Therefore, blocking of the variable is done to prevent the sub-tour. For example, X_{12} consists of entering the variable, so there is blocking of the variable in X_{12} and X_{21} when the block variable value is determined to be equal to 0, as shown in Table 3. In contrast, each leaving variable determines the block variable to be equal to 1.

TABLE 3. Block Variable in Sub-tour Elimination Step

Variable	X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}	X_{23}	X_{24}	X_{31}	X_{32}	X_{33}	...	X_{44}
Block Variable	1	0	1	1	0	1	1	1	1	1	1	...	1

After Step 1.1 to Step 1.4, a node appears in each iteration when it is possible to obtain nodes as both integer nodes and non-integer nodes.

Phase 2 consists of creating and developing the algorithm via application of the element decomposing method (EDCM). This phase is shown at the right in Figure 1, and is performed as follows.

Step 2.1 Using nodes from phase 1 to check the integer number in each iteration that starts from the last iteration.

Example 2: The integer problem in the two variables solved by the graph method is shown in Figure 5, and the Simplex method gives the node in each iteration, as shown in Table 4.

Objective function: $Max\ z = 4X_1 + 3X_2$
Constraint function: $X_1 + 2X_2 \leq 9$
$3X_1 + 2X_2 \leq 16$
$X_1, X_2 \geq 0$

FIGURE 5. The feasible region of the example from the graph method.

TABLE 4. Node of Example Problem 2 from Simplex Method

Iteration	X_1	X_2	Z
Iteration 1 (N1)	0	0	0
Iteration 2 (N2)	5.33	0	21.33
Iteration 3 (N3)	3.5	2.75	22.25

According to Example no. 2, the answer is that the optimal value $(Z) = 22.25$ is at node N3 which is $(X_1, X_2) = (3.5, 2.75)$. However, if it is a pure integer problem, it is necessary to use the element decomposing method (EDCM) for solving the problem. In this method, the feasible region is divided into small sub-areas such as those of the integer unit, or it is the feasible region which has the conner node as the integer node, as shown in Figure 6.

Step 2.2 Using the nodes in each iteration by making them adjust to integer node by using the element decomposing method (EDCM) which has three cases, namely Case 1 which is the complete integer node, Case 2 which is the non-complete integer node, and Case 3 which is the non-integer node, as shown in Table 5.

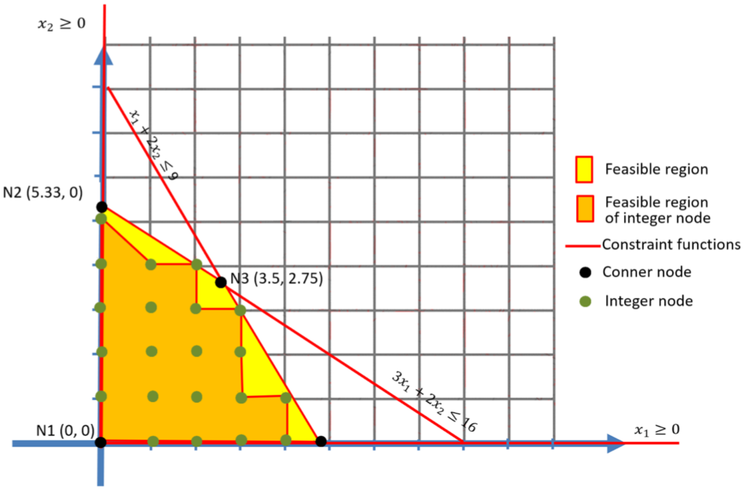


FIGURE 6. The feasible region with the integer problem.

TABLE 5. Type of Node in Example Problem 2

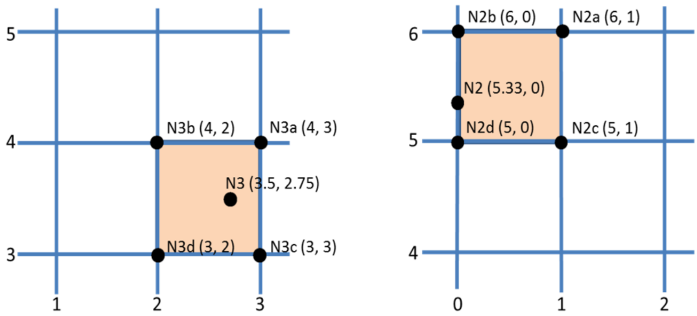
Iteration	X_1	X_2	Z	**Type**
Iteration 1 (N1)	0	0	0	Complete integer node
Iteration 2 (N2)	5.33	0	21.33	Non-complete integer node
Iteration 3 (N3)	3.5	2.75	22.25	Non-integer node

In Case 2 and Case 3, the non-integer node and the non-complete integer node adjust to the integer node by using the element decomposing method (EDCM), which is shown in Figure 7. The number of integer nodes is four, or $2^{numberofvariable}$.

The nodes are arranged in order according to the coefficient in each variable. If it is a minimum problem, arrange them in the ascending order; but, if it is a maximum problem, arrange them in the descending order, as shown in Table 6.

TABLE 6. Integer Node after Checking Constraint Functions

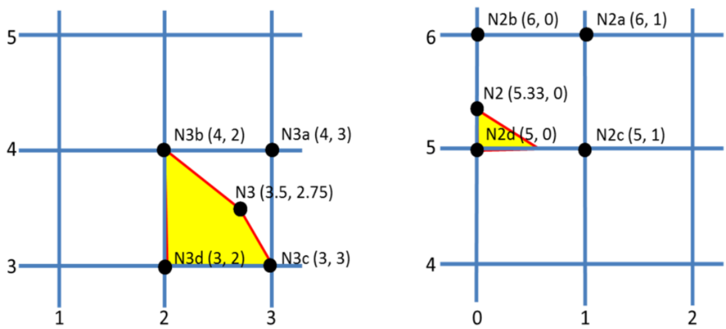
Iteration	**Original node**	**Node a**	**Node b**	**Node c**	**Node d**
Iteration 1 (N1)	(0, 0)	(0, 0)	-	-	-
Iteration 2 (N2)	(5.33, 0)	(6, 1)	(6, 0)	(5, 1)	(5, 0)
Iteration 3 (N3)	(3.5, 2.75)	(4, 3)	(4, 2)	(3, 3)	(3, 2)



(A) Node form Iteration 3 (B) Node from Iteration 2

FIGURE 7. Integer node by using the element decomposing method.

Step 2.3 Checking the integer node in each of the constraint functions. This starts from the node arranged in Step 2.2, as shown in Figure 8 and Table 7.



(A) Node form Iteration 3 (B) Node from Iteration 2

FIGURE 8. The integer node after checking the constraint function.

TABLE 7. Integer Node by Using Element Decomposing Method

Iteration	**Original node**	**Node a**	**Node b**	**Node c**	**Node d**
Iteration 1 (N1)	(0, 0)	(0, 0)	-	-	-
Iteration 2 (N2)	(5.33, 0)	~~(6, 1)~~	~~(6, 0)~~	~~(5, 1)~~	(5, 0)
Iteration 3 (N3)	(3.5, 2.75)	~~(4, 3)~~	(4, 2)	(3, 3)	(3, 2)

Step 2.4 Finding the value in the objective function using the integer node after checking the constraint function (from Step 2.3), which is shown in Table 8.

TABLE 8. Integer Node and Value

Iteration	Original node	Node a	Node b	Node c	Node d
Iteration 1 (N1)	(0, 0) Z = 0	(0, 0) Z = 0	-	-	-
Iteration 2 (N2)	(5.33, 0) Z = 21.33	~~(6, 1) Z = 27~~	~~(6, 0) Z = 24~~	~~(5, 1) Z = 23~~	(5, 0) Z = 20
Iteration 3 (N3)	(3.5, 2.75) Z = 22.25	~~(4, 3) Z = 25~~	(4, 2) Z = 22	(3, 3) Z = 21	(3, 2) Z = 18

Checking all the points may need a lot of time, so there are steps for checking and finding the answers in order to reduce length of time, as shown in Figure 9.

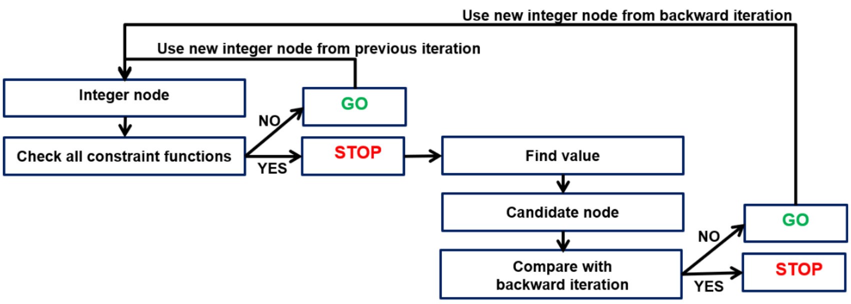


FIGURE 9. Steps for checking the node at each iteration. [21]

The node from the last iteration is firstly considered for use in checking the constraint function (checking the feasible region) under the condition that it can pass all the constraint functions. After that, the value from the objective function is calculated. When the node passes the condition, and stops at that step in the rest of the nodes in that iteration, the node from the checking acts as the candidate of that iteration.

The first node of each iteration, in the cases of both the non-integer node and the non-complete integer node, is out of the feasible region, or it cannot pass the constraint function of at least one equation, such as Node3a and Node2a.

After that, a comparison is made between the value in one iteration and that in the next iteration. If the result shows the value to be greater than that of the Maximum Problem or less than that of the Minimum Problem, examining for the answer to this problem must be stopped. If the result does not show, it is necessary to get the candidate of the new node in the next iteration of checking.

In Example 2 and Figure 10, it starts from Node3a (Iteration 3) as $(X_1, X_2) = (4, 3)$ which cannot pass the condition of the constraint function no.2; 3*(4) + 2*(3) = 18 > 16. Now we consider Node3b.

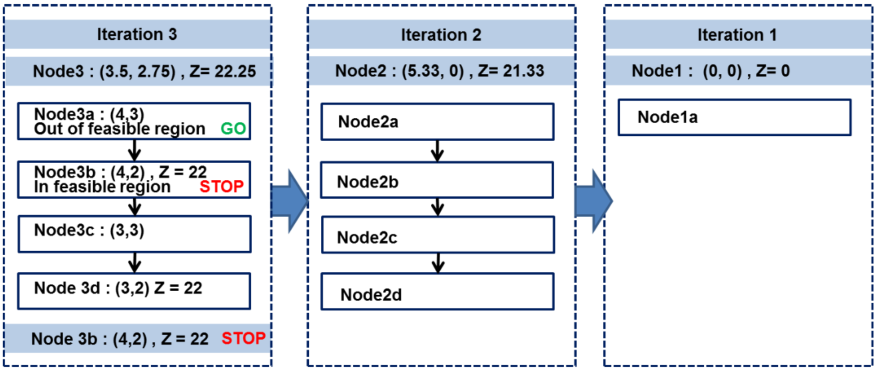


FIGURE 10. Steps for checking the node and finding the value in each iteration in Example 2.

Then, consider Node3b as $(X_1, X_2) = (4, 2)$ which can pass the conditions of both of the constraint functions and possess the value $(Z) = 22$. After that, stop the examination at iteration 3, which would make Node3b become the candidate of the iteration 3.

Subsequently, consider the value from Node3b in comparison to that from Node2 and determine that Node3b is higher than Node2 $(22 > 21.33)$. Then, stop the examination for the answer to the problem when the answer is at Node3b as $(X_1, X_2) = (4{,}2)$ and value $(Z) = 22$.

Accordingly, the element decomposing method provides a feasible region where the Conner node is the integer node, as shown in Figure 11.

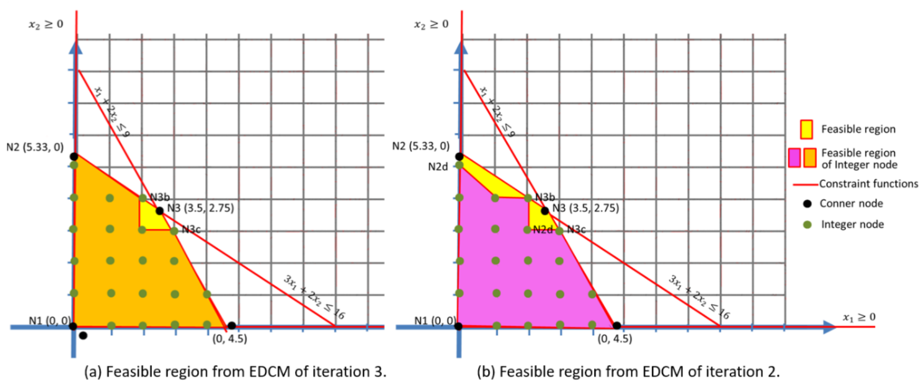


FIGURE 11. Feasible region from EDCM in each of the iterations.

4. Result and Discussion

A comparison of results was carried out between the results from the algorithm obtained by the application of the element decomposing method (EDCM) and the results from other algorithms such as ant colony optimization (ACO) and exact solution (B&B method). The results from four methods focus on the values and the time consumed, as shown in Table 9. As far as the problem is concerned, it can be solved with the number of cities, which are 6, 12, 22, 29, 30, 32, 50, 75, 100, 113, 237 and 343. The data with the problem of Symmetry-TSP are from the database OR Library and NSERC.These results are compared by percentages of difference in terms of value and time consumed, as shown in Table 10.

Table 9. Values and Time Consumed

NO. Station	**Value (Distance)**			**Time consumed**		
	B&B	**EDCM**	**ACO**	**B&B**	**EDCM**	**ACO**
6	66	66	66	1	1	1
12	132	132	132	10	1	1
22	469	476	476	1750	1	2
29	-	534	513	-	1	4
30	-	311	275	-	1	5
32	-	414	418	-	2	5
50	-	481	496	-	7	17
75	-	589	620	-	25	75
100	-	751	778	-	46	194
113	-	501	539	-	133	249
237	-	1220	1344	-	479	3656
343	-	1774	1852	-	8711	21626

Table 10. Percentages of Difference in Terms of Value and Time Consumed

NO. Station	**B&B vs EDCM**		**ACO vs EDCM**	
	Value	**Time consumed**	**Value**	**Time consumed**
6	0%	0%	0%	0%
12	0%	90%	0%	0%
22	1%	100%	0%	50%
29	-	-	4%	75%
30	-	-	13%	80%
32	-	-	1%	60%
50	-	-	3%	59%
75	-	-	5%	67%
100	-	-	3%	76%
113	-	-	7%	47%
237	-	-	9%	87%
343	-	-	4%	60%

Based on Table 9, the results are solved by ACO and EDCM that can resolve problems for 343 stations. However, the exact solution method has the capability to resolve problems for 22 stations.

Table 10 shows the percentages of difference in terms of time consumed and value when a comparison is made between the exact method, ACO, saving algorithm, and EDCM. When compared with the exact method, the values in the cases of 6 cities and 12 cities are the same when the value of the difference in the answer is at an average of 0.50% and that in the less time used for solving the problem is at an average of 63.31%. In addition, when it is compared to ACO, it can be found that EDCM provides better results in the case of the values at 1.31% and less time consumed at 55.00%.

This can be established in the following two categories:

1. Small number of stations: The values of the EDCM, ACO, and the exact method are equal or with a minor difference and the time consumed is short for problem-solving.

2. Large number of stations: The results of the EDCM and ACO are very different (not identical) and the time consumed is very short. EDCM provides better value in the case of large number of stations.

Acknowledgements

The authors would like to gratefully acknowledge the High-Technology and Advanced Computation Research Unit (Hi-TAC), Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University, for the supporting of this research work. Financial support from the Thailand Research Fund through the Research and Researchers for Industries Project (Grant NO. PHD59I0002) and Excellence Center in Logistics and Supply Chain Management, Chiang Mai University to Mr. Ekkaphon Jaiyen and Assoc. Prof. Dr. Komgrit Leksakul is also thankfully acknowledged.

References

[1] Department of Transport, Thailand, The Strategic Plan for the Department of Transport (Year 2011-2015), 2010 available at: http://www.dlt. go.th/th/attachments.

[2] T. Sorat, The Development of Logistics in year 2010/2011, 2011 available at: http://www.tanitsorat.com/view.php?id=439.

[3] L.W. Wayne, Integer Programming, Operations Research, Applications and Algorithms, 4^{th} edition, (2003) 526–530.

[4] J.F. McCloskey, OR forumBritish operational research in World War II, Operations Research 35 (1987) 453–470.

[5] J.H. Thomas, K. Wing, B. Alec, Finite element analysis of incompressible viscous flows by the penalty function formulation, Journal of Computational Physics 30 (1979) 1–60.

[6] E.K. Charles, The Finite Element Method in Mechanical Design, Virginia Polytechnic Institute and State University, Boston: PWS-KENT Publishing Company, 1993.

[7] D.W. Olivier, Y.K. Il, Finite Element Method, Lecture note of Engineering Design and Rapid Prototyping, Massachusetts Institute of Technology, USA, (2004), available at: http://web.mit.edu/16.810/www/16.810_L4_CAE.pdf.

[8] L.D. Whitley, T. Starkweather, D.A. Fuquay, Scheduling problems and traveling salesmen: The genetic edge recombination operator, ICGA 89 (1989) 133–140.

[9] T.S. Chang, Y.S. Wan, W.T. OOI, A stochastic dynamic traveling salesman problem with hard time windows, European Journal of Operational Research 98 (2009) 748–759.

[10] M. Dorigo, G. Caro and L. Gambardella, Ant colonies for the traveling salesman problem, BioSystems 43 (1997) 73–81.

[11] X. Li, M. Tian, C.S. Leung, Vehicle Routing Problems with Time Windows and Stochastic Travel and Service Times: Models and Algorithm, University of Hong Kong, 2010.

[12] A. Toriello, W.B. Haskell, M. Poremba, A Dynamic Traveling Salesman Problem with Stochastic Arc Costs, Department of Industrial and Systems Engineering, University of Southern California, 2013.

[13] R. Tadei, G. Perboli, F. Perfetti, The Muti-Path Traveling Salesman Problem with Stochastic Travel Costs, Interuniversity Research, CIRRELT, 2013.

[14] R. Pitakaso, Differential Evolution Algorithm for Logistics Transport Problem, (2015), available at: http://www.ubu.ac.th/p̃itakaso/1302476/new_doc/ ch05_s.pdf.

[15] J. Clausen, Branch and Bound Algorithms - Principles and Examples, Department of Computer Science, University of Copenhagen, 1999.

[16] C. Jens, Branch and Bound Algorithms - Principles and Examples, 1999, available at: http://janders.eecg.toronto.edu/1387/readings/b_and_b.pdf.

[17] A.T. Michael, Cutting Plane Techniques, 2014 available at: http://mat.gsia.cmu.edu/orclass/integer/node14.html.

[18] F.D. Lewis, Cutting Plane Techniques, 1999 available at: http://www.cs.uky.edu/l̃ewis/cs-heuristic/text/integer/cutting.html.

[19] E.K. Charles, The Finite Element Method in Mechanical Design, Virginia Polytechnic Institute and State University, Boston: PWS-KENT Publishing Company, 1–7.

[20] A. Tura, Z. Dong, Introduction to Finite Element Analysis (FEA) or Finite Element Method (FEM), Faculty of Engineering, University of Victory, 2014 available at: http://www.engr.uvic.ca/m̃ech410/lectures /FEA_Theory.pdf.

[21] E. Jaiyen, K. Leksakul, Application of element decomposing method for solving integer problems, MATEC Web of Conferences, 2018 available at: https://doi.org/10.1051/matecconf/201819201022.