

The transportation problem is well known and has very important applications. For this well-researched model, there are very efficient approaches for solving it that are available. These approaches include formulating the transportation problem as a linear program and then using the efficient methods such as the simplex method or interior point algorithms.

The Hungarian method is another efficient method for solving both the assignment model and the general transportation model. An assignment problem is a special case of the transportation model in which all supply and demand points are 1. Every transportation problem can be converted into an assignment problem since rows and columns can be split so that each supply and each demand point is 1.

The transportation simplex method is another method that is also used to solve the general transportation problem. This method is also called the modified distribution method (MODI). To use this approach, a starting solution is required and the closer the starting solution to the optimal solution, the fewer the iterations that are required to reach optimality.

The fourth method for transportation models is the network simplex method, which is the fastest so far. Unfortunately, all these approaches for transportation models are serial in nature and are very difficult to parallelize, which makes it difficult to efficiently use the available massively parallel technology. There is a need for an efficient approach for the transportation problem, which is easily parallelizable. This paper presents a See-Saw approach for solving the general transportation problem. This is an extension of the See-Saw approach for solving the assignment problem. The See-Saw moves can be done independently, which makes the approach proposed in this paper more promising than the available methods for transportation models

Keywords: transportation problem, Sew-Saw rule, linear programming, transportation simplex method

UDC 656

DOI: 10.15587/1729-4061.2021.243735

DEVELOPMENT OF A HEURISTIC TO SOLVE THE GENERAL TRANSPORTATION PROBLEM

Elias Munapo

PhD, Professor of Operations Research
Department of Statistics and
Operations Research
School of Economics and
Decision Sciences
North-West University
Mmabatho Unit 5, Mahikeng, Mafikeng,
South Africa, 2790
E-mail: emunapo@gmail.com

Received date 17.08.2021

How to Cite: Munapo, E. (2021). Development of a heuristic to solve the general transportation problem. *Eastern-European Journal of Enterprise Technologies*, 6 (4 (114)), 44–51. doi: <https://doi.org/10.15587/1729-4061.2021.243735>

Accepted date 01.11.2021

Published date 16.12.2021

1. Introduction

The general transportation problem is a well-researched model and there are very efficient approaches for solving this important problem that are available. These approaches include formulating the transportation problem as a linear program and then using the efficient methods such as the simplex method or interior point algorithms. The Hungarian method is another efficient method for the general transportation problem and the third one is the transportation simplex method. The fourth one is the network simplex method, which is the fastest so far. Unfortunately, all these approaches are serial in nature and are very difficult to parallelize, which makes it difficult to efficiently use the available massively parallel technology [1]. There is a need for an efficient approach for the transportation problem, which is easily parallelizable. This paper presents a See-Saw approach for solving the general transportation problem. This is an extension of the See-Saw approach for solving the assignment problem [2]. The See-Saw moves can be done independently, which makes the approach proposed in this paper more promising than the available methods. The transportation problem has applications in production planning, telecommunication, scheduling and military operations [3].

2. Literature review and problem statement

The method presented in the paper [2] has the weakness that it can only solve the transportation problem when both

supply and demand are exactly one. There is a need for a method that can solve a transportation problem with any size of supply or any demand. The general transportation can be formulated as a linear programming (LP) model. For this model, there are two main efficient algorithms, which are the simplex method and the interior point algorithm. The simplex method has proved to be effective for solving LPs even though it has an exponential worst case complexity and is affected by degeneracy [4]. Transportation models are highly degenerate problems. There are variants of the simplex method where the columns are pivoted and other variants where rows are pivoted. The simplex method follows the boundary of the convex region in its search process and this method is weak for very large LPs. On the other hand, the interior point algorithms start from the interior of the convex region and move towards the optimal solution, which lies at the boundary of the feasible region [5]. The interior point algorithms are not affected by degeneracy and the number of iterations required to reach optimality is not affected by the size of the problem. Interior point algorithms are good for large LPs.

The Hungarian method is another efficient method for solving both the assignment model and the general transportation version [6]. An assignment problem is a special case of the transportation model in which all supply and demand points are 1. Every transportation problem can be converted into an assignment problem since rows and columns can be split so that each supply and each demand point is 1.

The transportation simplex method is also used to solve the general transportation problem. This method is also called the modified distribution method (MODI). To use this approach,

a starting solution is required and the closer the starting solution to the optimal solution, the fewer the iterations that are required to reach optimality. To move from the starting solution towards the optimal solution, a Lagrangian function of the formulated LP is constructed. The Lagrangian multipliers are calculated at every stage until optimality is verified. There are three main methods that are used to find starting solutions. These are the North-West Corner (NWC) method, Least Cost (LM) method and Vogel's Approximation (VA) method [7]. In addition to these three main approximating methods, there are heuristics such as the particle swarm [8] that can be also used to accurately find a starting solution.

The network simplex method is currently the most efficient method for solving the general transportation problem. This method relies on converting the transportation into a minimum cost flow problem and then solving the problem [9]. This method is approximately 300 times faster than the transportation simplex method.

Another important feature of the transportation problem is that its cost matrix is unimodular [10]. This implies that its optimal solution is integer and this makes it easier to solve than the general linear integer problem.

The main weakness of all these known approaches is that they are serial in nature. In other words, these methods are not easily parallelizable. There is a need to come up with a method that is easily parallelizable, so as to take advantage of the ongoing developments in parallel processing.

3. The aim and objectives of the study

The aim of the study is to develop a method for the transportation problem, which is parallelizable.

To achieve this aim, the following objectives are accomplished:

- to explain the See-Saw rule;
- to apply the See-Saw rule that was developed for the assignment problem to the general transportation problem;
- to illustrate the proposed approach by a numerical illustration.

4. Materials and methods

In order to come up with a good method for the general transportation model, five known approaches for the general transportation problem were analyzed. These are the linear programming method, Hungarian method, transportation simplex method, network simplex method and See-Saw method for assignment models. The first four of these methods are serial in nature and are not easy to parallelize. The See-Saw method for assignment models was modified so that it can also solve the general transportation problem. This is a theoretical method and is easily parallelizable.

In this paper, the general transportation problem is given as shown in Table 1.

Table 1

Transportation problem

c_{11}	c_{12}	...	c_{1n}	S_1
c_{21}	c_{22}	...	c_{2n}	S_2
...
c_{m1}	c_{m2}	...	c_{mn}	S_m
D_1	D_2	...	D_n	T

Where c_{ij} is the cost of supplying from source point S_i to demand point D_j .

$$T=(S_1+S_2+\dots+S_m)=(D_1+D_2+\dots+D_n).$$

A transportation problem is called a balanced transportation problem if the total supply is equal to total demand, which is in this case equal to T . Any unbalanced transportation problem can be balanced by adding either a dummy column or dummy row.

Weakness of the available approaches for the transportation model.

The available approaches for the transportation problem are linear programming, Hungarian method, MODI and the network simplex method. The main challenge of these algorithms is that they are all serial in nature. These algorithms are very difficult to parallelize thus we cannot take advantage of the available massively parallel computing technology. There is a need for a method for the transportation models, which is easy to parallelize and take advantage of the continuous developments in parallel computing.

See-Saw algorithm.

The algorithm is based on the See-Saw game whereby two people sit at the ends of a long plank, which is balanced in the middle on a fixed support. When one person goes down the other goes up and vice versa. An important feature of this game is that both individuals cannot go up or down at the same time. This approach is not new and was proposed for the assignment model. In this paper, we slightly modify it so that it also solves the general transportation problem.

5. Results of the development of the heuristic

5.1. Solution methods

There are efficient methods that are available for the transportation model. The efficient approaches include modeling as a linear programming problem, Hungarian method, transportation simplex method and network simplex method.

5.1.1. Linear programming

The transportation problem given in Table 1 can be modeled as a linear programming model given in (1):

$$Z_{RELAX} = \text{Minimize } c_{11}x_{11} + c_{12}x_{12} + \dots + c_{mn}x_{mn},$$

Subject to:

$$\begin{aligned} x_{11} + x_{12} + \dots + x_{1n} &= S_1, \\ x_{21} + x_{22} + \dots + x_{2n} &= S_2, \\ \dots & \\ x_{m1} + x_{m2} + \dots + x_{mn} &= S_m, \end{aligned} \tag{1}$$

$$\begin{aligned} x_{11} + x_{21} + \dots + x_{m1} &= D_1, \\ x_{12} + x_{22} + \dots + x_{m2} &= 1, \\ \dots & \\ x_{m1} + x_{m2} + \dots + x_{mn} &= D_m. \end{aligned}$$

$$x_{ij} \geq 0.$$

This can be solved efficiently using interior point algorithms or some variants of the simplex method. The optimal solution of this linear programming model is guaranteed to be

integer because its coefficient matrix is unimodular. In other words, this coefficient matrix has at most two nonzero (1s) in all its columns.

5. 1. 2. Hungarian method

The Hungarian method was developed from two Hungarian theorems. The Hungarian method is not affected by degeneracy. This method is centered on the fact that subtracting or adding a constant to a row or column of Table 1 does not change the optimal solution. This is possible if Table 1 is balanced as given in (2):

$$D_1 + D_2 + \dots + D_n = S_1 + S_2 + \dots + S_m = S = D. \tag{2}$$

If the table is not balanced, then it can be balanced by the addition of a dummy column or dummy row. Let p_i be a constant subtracted from row i and q_j be a constant subtracted from column j . Thus, the cost element c_{ij} changes to \bar{c}_{ij} as given in (3):

$$\bar{c}_{ij} = c_{ij} - p_i - q_j. \tag{3}$$

Thus, the total cost can be explained as follows:

$$\text{Total Cost} = \sum_i \sum_j \bar{c}_{ij} x_{ij} = \sum_i \sum_j (c_{ij} - p_i - q_j) x_{ij};$$

$$\text{Total Cost} = \sum_i \sum_j c_{ij} x_{ij} - \sum_i p_i (1) - \sum_j q_j (1);$$

$$\text{Total Cost} = \sum_i \sum_j c_{ij} x_{ij} - \text{constant}. \tag{4}$$

The Hungarian method can solve both the assignment model and the general transportation model. The assignment model is a special transportation model in which all supplies and all demands are equal to 1. In addition, every transportation model can be expressed as an assignment problem by splitting all rows and all columns so that the amounts demanded and supplies are all equal to 1.

5. 1. 3. Transportation simplex method

The transportation simplex method is also called modified distribution method (MODI) and relies on the strategy of approximating the solution to the transportation problem and improving it until it is optimal. The transportation model given in (1) has equality constraints and its Lagrangian function (L) can be constructed as given in (5):

$$L = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + u_i \left(1 - \sum_j x_{ij} \right) + v_j \left(1 - \sum_i x_{ij} \right), \tag{5}$$

where u_i and v_j are Lagrangian multipliers and the optimality conditions are as given in (6), (7):

$$\frac{\partial L}{\partial u_i} = 1 - \sum_j x_{ij} = 0. \tag{6}$$

$$\frac{\partial L}{\partial v_j} = 1 - \sum_i x_{ij} = 0. \tag{7}$$

For this method to be efficient, the approximated starting solution must be near optimal. There are three main methods that are used to find the starting solution. These are North West Corner (NWC) method, Least Cost (LC) method and Vogel's Approximation (VA) method.

5. 1. 4. Network simplex method

This is a special-purpose simplex method, which is much more efficient than the ordinary simplex method. In this method, the transportation problem is first modeled as a minimum cost flow problem. At each stage of the task, it is solved as a minimum spanning tree resulting in very fast computations.

5. 1. 5. See-Saw rule for assignment models

To apply the See-Saw rule that was developed for the assignment problem to the general transportation problem.

The assignment rule has special features that make it easy to apply the See-Saw rule. In an assignment model, there is exactly one allocation in every column and when considering two columns, a movement up in one column implies a movement down in the other column and vice versa. In the general transportation model, there is more than one allocation in the same column. An example of an allocation in an assignment model is given in Table 2.

Table 2

Example of an allocation in an assignment model

21	17[1]	7	19	48	27	1
129	49	5[1]	65	36	25	1
89	33	39	46	30[1]	39	1
5	6	7	3[1]	2	3	1
64[1]	65	9	87	56	83	1
51	125	5	59	92	45[1]	1
1	1	1	1	1	1	6

Note that there is at least one allocation in every column and at least one allocation in every row.

5. 2. General transportation model

In the general transportation model, there is more than one allocation in the same column. The See-Saw moves for an assignment model are simple. In other words, there is a single See-Saw move for every pair of columns. For a transportation model, there is more than one allocation in a single column. An example of an allocation in the general transportation problem is given in Table 3.

Table 3

Example of an allocation in the general transportation model

23[10]	20	10	22	51	30	10
132[10]	52[80]	8[110]	68[30]	39[60]	28	290
92	35	42	49	33[5]	42	5
8	9	10	6	5[105]	5[90]	195
20	80	110	30	170	90	500

Note that there are 2 allocations in the first column and 3 allocations in the fifth column. For every pair of columns that involve the first column, we have at least 2 See-Saw moves.

5. 2. 1. See-Saw Moves

In a See-Saw move, two columns are paired. In these two paired columns, the allocations in the two columns are determined. A See-Saw move of the first allocation in the first column with all the allocations in the other paired column is done. The profit or cost of the See-Move is noted. The process is repeated on all other columns and the most profitable See-Saw move is selected. Given the element (2i) in column i paired with column j , we can have 3 possible See-Saw moves as given in Fig. 1–3.

c_{1i}	$c_{1j}[\alpha_1^j]$
$c_{2i}[\alpha_2^i]$	c_{2j}
c_{3i}	$c_{3j}[\alpha_3^j]$
c_{4i}	c_{4j}
c_{5i}	c_{5j}
$c_{6i}[\alpha_6^i]$	c_{6j}
c_{7i}	$c_{7j}[\alpha_7^j]$
...	...
$c_{mi}[\alpha_m^i]$	c_{mj}

Fig. 1. See-Saw move (a)

c_{1i}	$c_{1j}[\alpha_1^j]$
$c_{2i}[\alpha_2^i]$	c_{2j}
c_{3i}	$c_{3j}[\alpha_3^j]$
c_{4i}	c_{4j}
c_{5i}	c_{5j}
$c_{6i}[\alpha_6^i]$	c_{6j}
c_{7i}	$c_{7j}[\alpha_7^j] c_{7j}$
...	...
$c_{mi}[\alpha_m^i]$	c_{mj}

Fig. 2. See-Saw move (b)

c_{1i}	$c_{1j}[\alpha_1^j]$
$c_{2i}[\alpha_2^i]$	c_{2j}
c_{3i}	$c_{3j}[\alpha_3^j]$
c_{4i}	c_{4j}
c_{5i}	c_{5j}
$c_{6i}[\alpha_6^i]$	c_{6j}
c_{7i}	$c_{7j}[\alpha_7^j] c_{7j}$
...	...
$c_{mi}[\alpha_m^i]$	c_{mj}

Fig. 3. See-Saw move (c)

In Fig. 3, the movement is from cell $2i$ to $7i$ in column i and from cell $7j$ to $2ij$ in the other column j .

5. 2. 2. Profitability

A See-Saw move is profitable if the cost before the move is bigger than the cost after the move. From Table 3, the inequalities (8), (9) can be drawn.

Profitable See-Saw move:

$$c_{2i}[\alpha_2^i] + c_{1j}[\alpha_1^j] > c_{1j}[\alpha_1^j] + c_{2j}[\alpha_2^j]. \tag{8}$$

Costly See-Saw move:

$$c_{2i}[\alpha_2^i] + c_{1j}[\alpha_1^j] < c_{1j}[\alpha_1^j] + c_{2j}[\alpha_2^j]. \tag{9}$$

Neither profitable nor costly:

$$c_{2i}[\alpha_2^i] + c_{1j}[\alpha_1^j] = c_{1j}[\alpha_1^j] + c_{2j}[\alpha_2^j]. \tag{10}$$

The See-Saw move given in (10) is neither costly nor profitable. This move is important in determining alternate solutions.

Simplifying the See-Saw move.

In a See-Saw move, the move is profitable if the cost of keeping quantities in their current position is higher than the cost of keeping them at new positions. In other words, we can easily determine if the See-Saw move is profitable or not without the full computation of (8). Suppose we are making the See-Saw move as given in Fig. 4.

The paired columns are i and j . Profitability in a See-Saw move can be determined even if the amount being moved is not known.

$$c_{ki} - c_{ti} > c_{kj} - c_{tj}, \tag{11}$$

i.e.

$$c_{ki} + c_{tj} > c_{kj} + c_{ti}. \tag{12}$$

c_{1i}	c_{1j}
c_{2i}	c_{2j}
...	...
c_{ki}	c_{kj}
...	...
...	...
c_{6i}	c_{lj}
...	...
...	...
c_{mi}	c_{mj}

Fig. 4. Simplifying the See-Saw move

The expression in (12) is made up of cross sums and this makes the See-Saw moves very easy to compute. The See-Saw moves are made up of additions and subtractions only. If

$$c_{ki} + c_{lj} = c_{kj} + c_{li} \tag{13}$$

Then there exists an alternate solution giving the same solution.

The See-Saw algorithm that was developed for the transportation models is summarized as follows.

Step 1. Use either the LC or VA to determine a starting solution for the transportation problem.

Step 2. Pair the columns starting from the left. Select the best See-Saw move. Repeat the procedure until there is no See-Saw move possible to all the unoccupied cells and go to Step 3.

Step 3. Current solution is optimal.

The current solution is optimal if no profitable See-Saw move is possible, i.e.:

$$c_{ki} + c_{lj} \leq c_{kj} + c_{li}$$

If a See-Saw move is not profitable then it is not necessary to take such a move. This is illustrated in a numerical illustration given in Table 4.

5. 3. Numerical illustration

The transportation problem for the full numerical illustration is given in Table 4.

Using the least cost (LC) method, the starting solution becomes as given in Table 5.

Table 4

Full example of the See-Saw rule for the general transportation model

9	15	23	14	34	30
5	10	12	23	33	50
7	15	17	19	29	20
29	20	26	31	8	20
10	30	30	30	20	120

Table 5

Least Cost starting solution

9	15	23	14[30]	34	30
5[10]	10[30]	12[10]	23	33	50
7	15	17[20]	19	29	20
29	20	26	31	8[20]	20
10	30	30	30	20	120

$$\text{Total Cost (LC)} = 5(10) + 10(30) + 12(20) + 17(20) + 14(30) + 8(20) = 1,390. \tag{14}$$

See-Saw moves.

Iteration 1. The first iteration is composed of applying See-Saw moves by pivoting Columns 1–4 as given in Table 6. Pivoting Column 1 (Iteration 1).

Table 6

Pivoting column 1

9	15	23	14	34	30
5	10	12	23	33	50
7	15	17	19	29	20
29	20	26	31	8	20
10	30	30	30	20	120

See-Saw moves for pivoting Column 1.

$$c_{21} + c_{33} > c_{31} + c_{23} \Rightarrow 5 + 17 > 7 + 12 \Rightarrow 22 > 19.$$

This is profitable and 10 to be moved.

$$c_{21} + c_{14} < c_{11} + c_{24} \Rightarrow 5 + 14 < 9 + 23 \Rightarrow 19 < 31.$$

This is not profitable.

$$c_{21} + c_{45} < c_{41} + c_{25} \Rightarrow 5 + 8 < 29 + 33 \Rightarrow 13 < 62.$$

This is not profitable.

Pivoting Column 2 (Iterations 1). The pivoting iterations are shown in Tables 11–16. Pivoting of Column 2 is given in Table 7.

Table 7

Pivoting column 2

9	15	23	14	34	30
5	10	12	23	33	50
7	15	17	19	29	20
29	20	26	31	8	20
10	30	30	30	20	120

See-Saw moves for pivoting Column 2.

$$c_{22} + c_{33} = c_{32} + c_{23} \Rightarrow 10 + 17 = 15 + 12 \Rightarrow 27 = 27.$$

Alternate solution to this exists.

$$c_{22} + c_{14} < c_{12} + c_{24} \Rightarrow 10 + 14 < 15 + 23 \Rightarrow 24 < 38.$$

This is not profitable.

$$c_{22} + c_{45} < c_{42} + c_{25} \Rightarrow 10 + 8 < 20 + 33 \Rightarrow 18 < 53.$$

This is not profitable.

Pivoting Column 3 which is Iteration 1 is given in Table 8.

Table 8

Pivoting column 3

9	15	23	14	34	30
5	10	12	23	33	50
7	15	17	19	29	20
29	20	26	31	8	20
10	30	30	30	20	120

See-Saw moves for pivoting Column 3.

Note that there are two allocations in this column and we need See-Saw moves for each of the allocations.

$$c_{23} + c_{14} < c_{13} + c_{24} \Rightarrow 12 + 14 < 23 + 23 \Rightarrow 26 < 46.$$

This is not profitable.

$$c_{23} + c_{45} < c_{24} + c_{43} \Rightarrow 12 + 8 < 33 + 26 \Rightarrow 20 < 59.$$

This is not profitable.

$$c_{33} + c_{14} < c_{13} + c_{34} \Rightarrow 17 + 14 < 23 + 19 \Rightarrow 31 < 42.$$

This is not profitable.

$$c_{33} + c_{45} < c_{43} + c_{35} \Rightarrow 17 + 8 < 33 + 26 \Rightarrow 25 < 59.$$

This is not profitable.

Pivoting Column 4 which is also Iteration 1 is given in Table 9.

Table 9

Pivoting column 4

9	15	23	14	34	30
5	10	12	23	33	50
7	15	17	19	29	20
29	20	26	31	8	20
10	30	30	30	20	120

See-Saw moves for pivoting Column 4. Note that there is only one See-Saw move possible after pivoting Column 4.

$$c_{14} + c_{45} < c_{44} + c_{14} \Rightarrow 14 + 8 < 31 + 34 \Rightarrow 22 < 65.$$

This is not profitable.

The solution obtained from pivoting Columns 1–4 is given in Table 10.

Table 10

Solution after Iteration 1

9	15	23	14[30]	34	30
5	10[30]	12[20]	23	33	50
7[10]	15	17[10]	19	29	20
29	20	26	31	8[20]	20
10	30	30	30	20	120

$$\text{Total Cost (Iteration 1)} = 7(10) + 10(30) + 12(20) + 17(10) + 14(30) + 8(20) = 1,360. \quad (15)$$

Comparing this solution with the starting solution, we can notice that only one allocation in the first column has changed. The rest remains the same. So we need one pivot for Iteration 2. This is done by pivoting Column 1.

In Iteration 2, Column 1 is pivoted as given in Table 11.

Table 11

Pivoting column 1

9	15	23	14	34	30
5	10	12	23	33	50
7	15	17	19	29	20
29	20	26	31	8	20
10	30	30	30	20	120

See-Saw moves for pivoting Column 1 in Iteration 2.

$$c_{31} + c_{22} < c_{22} + c_{32} \Rightarrow 7 + 10 < 5 + 15 \Rightarrow 17 < 20.$$

This is not profitable.

$$c_{31} + c_{23} < c_{21} + c_{33} \Rightarrow 7 + 12 < 5 + 17 \Rightarrow 19 < 22.$$

This is not profitable.

$$c_{31} + c_{14} < c_{11} + c_{34} \Rightarrow 7 + 14 < 9 + 19 \Rightarrow 21 < 28.$$

This is not profitable.

$$c_{31} + c_{45} < c_{41} + c_{35} \Rightarrow 7 + 8 < 29 + 29 \Rightarrow 15 < 58.$$

This is not profitable.

This means the current solution (Table 10) is optimal. Also from Iteration 1, when pivoting Column 2 we have evidence of an alternate solution. From Table 10, the alternate optimal solution becomes as given in Table 12.

Table 12

Alternate optimal solution

9	15	23	14[30]	34	30
5	10[20]	12[30]	23	33	50
7[10]	15[10]	17	19	29	20
29	20	26	31	8[20]	20
10	30	30	30	20	120

$$\text{Total Cost(Iteration 1)} = 7(10) + 10(20) + 15(10) + 12(30) + 14(30) + 8(20) = 1,360. \quad (16)$$

The proposed approach has the advantage that pivoting of the four columns in Iteration 1 can be done independently allowing the use of parallel processors.

6. Discussion of the results of the development of the heuristic

The numerical illustration in Table 4 has 3 rows and 5 columns and required 6 allocations to come up with a feasible starting solution. In this case, the LC was used to come up

with a starting solution whose total cost amounted to 1,370. This is highly accurate since the optimal solution had a total cost of 1,360. To get to this optimal solution, 2 iterations of the See-Saw approach were necessary. The first iteration required 4 column pivots to come up with a new solution. The second iteration required only 1 column pivot since all the other 4 columns did not change in the first iteration. The second iteration showed that the solution obtained in the first iteration was optimal since $c_{ki} + c_{lj} \leq c_{kj} + c_{li}$ for all cells of the transportation problem. Since there was no change in allocations for the last 4 columns and that the first iteration had $c_{22} + c_{33} = c_{32} + c_{23}$, which indicated the existence of an alternate solution, the solution obtained in the second iteration is optimal and that an alternate solution exists, which also gave a total of 1,360. The total number of pivots was $4+1=5$ for the given problem. The illustration also showed that the See-Saw moves and pivots can be done at the same time. Parallel processors can be tasked to handle the 4 pivots in iteration 1.

Before this study, there were four main approaches for the general transportation problem. These approaches are modeling the transportation problem as a linear programming problem, using the Hungarian method, using the transportation simplex method (also called MODI method) and applying the network simplex method.

Linear programming has been successively used to solve the general transportation problem. The mathematical formulation of the transportation problem is easily determined. The problem with this linear programming formulation is that it is seriously affected by the problem of degeneracy. The obvious advantage of linear programming formulation is that it does not require a starting solution.

The Hungarian method is also used to solve the general transportation problem and this method is not affected by degeneracy. It is sometimes computationally expensive to reach the optimal solution using the Hungarian method. In other words, this method sometimes takes a significant amount of computational time to find the optimal solution. This method, just like the linear programming formulation, also does not require a starting solution.

The transportation simplex method, which is also called the modified distribution method (MODI), is also used to solve the general transportation problem. In this approach, a starting solution is required. The closer the starting solution is to the optimal solution, the fewer the MODI iterations that are required to move to the optimal solution. There are methods that are used to quickly find good starting solutions. Unfortunately using any of the starting solution methods also increases the total computational time for the problem.

The fourth available method for the general transportation problem is called the network simplex method. Of these four methods that are already known, the network simplex method is the most efficient in terms of computational time. Even though this method is efficient, it is serial in nature and is not easy to split its algorithm into parallel parts and take advantage of the developments in computer parallel processing.

The numerical illustration given from Table 4 to Fig. 3, 4 showed that the See-Saw moves and pivots can be done at the same time. Computer parallel processors can be tasked to handle the 4 pivots in iteration 1 at the same time. Also note that the Saw-Saw moves within each pivot can also be done independently.

The proposed approach has its own way of detecting alternate solutions. Any alternate exists if the See-Saw move is neither profitable nor a loss. In Table 7, pivoting Column 2 results in a See-Saw move, which gives a new cost of 27, but the current position gives a cost of 27. In this case, the See-Saw move is neither profitable nor a loss and shows the existence of an alternate solution.

The only weakness of the proposed approach is that there are no computational results.

7. Conclusions

1. The See-Saw rule for the assignment model was explained. The See-Saw movement is feasible and possible if it results in a decrease in the total assignment cost. This is a quantitative measure and is used to determine the feasibility of the See-Saw move.

2. The See-Saw rule that was developed to solve the assignment model can be extended to the general transportation model. This comes from the fact that any transportation model can be expressed as a combination of assignment models. Unlike in an assignment model where we can have a single movement per column at a time, a general transportation model can have more than one movement per column. In this case, quantitative indicators are also used to determine See-Saw moves in transportation problems. The only difference with assignment See-Saw moves is that there are more than one quantitative indicator used per column.

3. The numerical illustration clearly showed how the See-Saw moves can be done independently. Parallel processors can then be tasked to handle the independent See-Saw moves. Solving the tasks in parallel can significantly reduce the computational time of the proposed heuristic.

References

- Conte, D., Grossi, G., Lanzarotti, R., Lin, J., Petrini, A. (2021). Analysis of a parallel MCMC algorithm for graph coloring with nearly uniform balancing. *Pattern Recognition Letters*, 149, 30–36. doi: <https://doi.org/10.1016/j.patrec.2021.05.014>
- Kumar, S., Munapo, E., Nyamugure, P. (2021). An Insight into the Characteristic Equation for an Integer Program. *International Journal of Mathematical, Engineering and Management Sciences*, 6 (2), 611–620. doi: <https://doi.org/10.33889/ijmems.2021.6.2.037>
- Kline, A., Ahner, D., Hill, R. (2019). The Weapon-Target Assignment Problem. *Computers & Operations Research*, 105, 226–236. doi: <https://doi.org/10.1016/j.cor.2018.10.015>
- Liu, Y., Tu, Y., Zhang, Z. (2021). The row pivoting method for linear programming. *Omega*, 100, 102354. doi: <https://doi.org/10.1016/j.omega.2020.102354>
- Castro, J., Nasini, S. (2021). A specialized interior-point algorithm for huge minimum convex cost flows in bipartite networks. *European Journal of Operational Research*, 290 (3), 857–869. doi: <https://doi.org/10.1016/j.ejor.2020.10.027>
- Rabbani, Q., Khan, A., Qudoods, A. (2019). Modified Hungarian method for unbalanced assignment problem with multiple jobs. *Applied Mathematics and Computation*, 361, 493–498. doi: <https://doi.org/10.1016/j.amc.2019.05.041>

7. Taha, H. A. (2017). Operations Research: An Introduction. Harlow, United Kingdom: Pearson.
8. Singh, G., Singh, A. (2021). Extension of particle swarm optimization algorithm for solving transportation problem in fuzzy environment. Applied Soft Computing, 110, 107619. doi: <https://doi.org/10.1016/j.asoc.2021.107619>
9. Holzhauser, M., Krumke, S. O., Thielen, C. (2017). A network simplex method for the budget-constrained minimum cost flow problem. European Journal of Operational Research, 259 (3), 864–872. doi: <https://doi.org/10.1016/j.ejor.2016.11.024>
10. Micheli, G., Weger, V. (2019). On Rectangular Unimodular Matrices over the Algebraic Integers. SIAM Journal on Discrete Mathematics, 33 (1), 425–437. doi: <https://doi.org/10.1137/18m1177093>

This paper describes the process of shooting a mobile armored combat vehicle with directed fragmentation-beam shells as a discrete-continuous random process. Based on this approach, a stochastic model has been proposed in the form of a system of Kolmogorov-Chapman differential equations.

A universal model of the process of defeating a moving armored target with directed fragmentation-beam shells has been built, which would provide preconditions for experimental studies into the effectiveness of various variants of the components of the artillery system for three-shot firing.

The execution of an artillery task is considered as a set of certain procedures characterized by the average value of its duration. They are dependent on the firing phases involving a prospective automatic gun and the explosive destruction of fragmentation-beam shells while the explosive destruction of each shell case is characterized by the self-propagation of the reaction of explosive transformations based on tabular data on the target. An indicator of the functionality of various design options for fragmentation-beam shells is the probability of causing damage by «useful fragments» in the vulnerable compartments of a combat armored vehicle.

Devising universal models for the process of shooting a moving armored vehicle forms preconditions for further full-time experiments in accordance with the design solutions defined as a result of modeling. It is possible to use the developed discrete-continuous stochastic model in other modeling tasks to determine the optimal value of defeat.

As regards the practical application of discrete-continuous stochastic models, one can argue about the possibility of reducing the cost of performing design tasks related to weapons by 25 % and decreasing the likelihood of making mistakes at the stage of system engineering design

Keywords: *discrete-continuous stochastic model, graph of states and transitions, fragmentation-beam shells*

UDC 623.418.2:355.354
DOI: 10.15587/1729-4061.2021.245703

BUILDING A MODEL OF THE PROCESS OF SHOOTING A MOBILE ARMORED TARGET WITH DIRECTED FRAGMENTATION-BEAM SHELLS IN THE FORM OF A DISCRETE-CONTINUOUS STOCHASTIC SYSTEM

Vadim Yakovenko

Doctor of Technical Sciences, Senior Researcher*

Bohdan Volochiy

Doctor of Technical Sciences, Professor

Department of Theoretical Radio Engineering and Radio Measurement

Lviv Polytechnic National University

S. Bandery str., 12, Lviv, Ukraine, 79013

Yuriy Sydorenko

Doctor of Technical Sciences, Professor, Director of the Institute

E. O. Paton Institute of Materials Science and Welding of the National

Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»

Politechnichna str., 35, Kyiv, Ukraine, 03056

Nataliia Furmanova

Corresponding author

PhD, Associate Professor**

E-mail: nfurmanova@gmail.com

Oleksandr Malyi

PhD**

Anton Tkachenko

PhD, Senior Researcher*

Yurii Olshevskiy

PhD, Senior Researcher*

*The Scientific and Methodological Center of Scientific,

Scientific and Technical Activities Organization

National Defence University of Ukraine named after Ivan Cherniakhovskiy

Povitroflotskyi ave., 28, Kyiv, Ukraine, 03049

**Department of Information Technologies of Electronic Devices

Zaporizhzhia Polytechnic National University

Zhukovskoho str., 64, Zaporizhzhia, Ukraine, 690063

Received date 04.10.2021

Accepted date 02.12.2021

Published date 16.12.2021

How to Cite: Yakovenko, V., Volochiy, B., Sydorenko, Y., Furmanova, N., Malyi, O., Tkachenko, A., Olshevskiy, Y. (2021). Building a model of the process of shooting a mobile armored target with directed fragmentation-beam shells in the form of a discrete-continuous stochastic system. Eastern-European Journal of Enterprise Technologies, 6 (4 (114)), 51–63. doi: <https://doi.org/10.15587/1729-4061.2021.245703>

1. Introduction

The current state of saturation of the world armies with armored vehicles and manpower in terms of protected means

of individual (combat) equipment [1] did not affect the change in their product range towards reducing high-explosive fragmentation (HE-FRAG) shells that reached the limit of their capabilities [2]. However, we shall focus only