A Study on Reducing Convergence Iterations of a Timetabling Transportation Algorithm

Bassam Nakhle[1]

Abstract

Some computational results are presented in this work trying to show how the convergence iterations needed for a particular type of matrices resulting from a least costs scheduling transportation algorithm can be reduced. More in details, a scheduling or timetabling problem could be considered as an assignment problem which could be solved by a transportation algorithm instead of using the Hungarian method. These facts permitted to us to solve a scheduling problem by the transportation (U-V method) algorithm with some particular constraints. Supplementary constraints are tested wishing to reduce the iterations number needed for convergence.

Keywords: Transportation problem, least cost method, assignment, scheduling.

1. Introduction

Years ago till now, many methods have been applied to find the optimal solution of an important problem: time organization or scheduling. Here, it is not an evidence to find always the ideal solution to be applied. Then the best existing solution or the optimal one is searched. A heuristic approach has been used to solve this type of problem (Nanda, Pai, & Gole, 2012; Silver, Vidal, & Werra, 1980), graph theory has been applied too (Cauvery, 2011; Werra (1985); Leighton, 1979).

Nowadays, with the continuous growth of computers power, tools are more helpful. The use of assignment concept mentioned in Michalewicz and

[1] Associate Professor, Common Trunk Department, Faculty of Economics and Business Administration, Lebanese University, Ashrafieh – Lebanon.

Email Address: bassam.nakhle@ul.edu.lb

Fogel (2000) to solve a scheduling problem seems to be more practical than before. A transportation (U-V) software has been developed to this study and efforts have been concentrated on the modeling of scheduling problem by an assignment problem. Efforts are deployed too to reduce hard computations on large costs matrices resulting from this modeling process by adding constraints on the least cost and the solution optimization algorithms. Results on two sets of costs matrices are presented in this work. The first set consists of 100 matrices of 120 rows and 120 columns each and 100 matrices of 160 rows and 160 columns each. The second set consists of 13 groups of 30 matrices each. All these matrices are written with respect to matrix form obtained by assignment problem modeling. Results show that the number of iterations needed for convergence is reduced.

2. Transportation problem

Transportation problem consists of minimizing a cost function under transportation constraints (Saad, 2004).

We must find:

$$Min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} q_{ij}$$

Such that:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} q_{ij} = \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

Where:

- m: Number of sources.
- n: Destinations number.
- a_i: Quantity to be shipped from source i. A is a vector containing offered quantities $A = [a_i]$.
- b_j: Quantity received by destination j. B is a vector containing received quantities $B = [b_j]$.
- c_{ij}: Transportation cost for one unit of goods shipped from source *i* to destination *j*.

  q_{ij}: Quantity to be shipped from source *i* to destination *j*.

Problem solution starts by finding an initial feasible solution using one of Northwest corner method, least costs or Vogel method. Initial feasible solution is generally optimized by a MODI (U-V) algorithm. In general, starting with the Northwest Corner method demands hereafter more iterations to converge. We see in Loch and Silva (2014) that for some larger cost matrices, the best is to start with least cost method for initial feasible solution. That is why we choose this method to start with our transportation program named "OR Transport".

Once we obtain the initial solution, parameters u, v and $\bar{c}_{ij}$ are computed at each iteration by writing for each element in the base:

$$u_i + v_j = c_{ij}$$

And for each element out of the base:

$$\bar{c}_{ij} = u_i + v_j - c_{ij}$$

Convergence is reached when all $\bar{c}_{ij}$ are non positive.

Tables 1 to 7 show computation results of MatLab "OR Transport" code. From six sources $O_1 \dots . O_6$, thirty two units of goods are shipped to six destinations $D_1 \dots . D_6$. Supply vector is A(1 2 3 7 9 10) and demand vector is B(2 5 2 4 15 4). Solution consists of seven tables, first one shows the initial solution provided by the least cost method and six other tables correspond to six iterations needed for convergence.

The second and third iteration provide same transportation cost. Actually, the base element $q_{6,2}$ is equal to 0 in iteration two. It is replaced by $q_{5,2}$ in third iteration. Minimal transportation cost is $1 \times 12 + 2 \times 13 + 3 \times 15 + 5 \times 14 + 2 \times 5 + 5 \times 32 + 4 \times 16 + 2 \times 21 + 4 \times 27 + 4 \times 25 = 637$ units.

Table 1: Solution achieved by Least Costs Method

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12	12 **1**	1
O_2	15	30	10 **2**	35	13	17	2
O_3	20	15	20	45	15	15 **3**	3
O_4	5 **2**	10 **5**	50	30	14	13	7
O_5	34	32	36	40 **4**	38 **5**	16 **0**	9
O_6	22	23 **0**	21 **0**	27	25 **10**	28	10
	2	5	2	4	15	4	

Table 2: First iteration

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12 **1**	12	1
O_2	15	30	10 **2**	35	13	17	2
O_3	20	15	20	45	15	15 **3**	3
O_4	5 **2**	10 **5**	50	30	14	13	7
O_5	34	32	36	40 **4**	38 **4**	16 **1**	9
O_6	22	23 **0**	21 **0**	27	25 **10**	28	10
	2	5	2	4	15	4	

Table 3: Second iteration

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12 **1**	12	1
O_2	15	30	10 **2**	35	13	17	2
O_3	20	15	20	45	15 **3**	15	3
O_4	5 **2**	10 **5**	50	30	14	13	7
O_5	34	32	36	40 **4**	38 **1**	16 **4**	9
O_6	22	23 **0**	21 **0**	27	25 **10**	28	10
	2	5	2	4	15	4	

Table 4: Third iteration

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12 **1**	12	1
O_2	15	30	10 **2**	35	13	17	2
O_3	20	15	20	45	15 **3**	15	3
O_4	5 **2**	10 **5**	50	30	14	13	7
O_5	34	32 **0**	36	40 **4**	38 **1**	16 **4**	9
O_6	22	23	21 **0**	27	25 **10**	28	10
	2	5	2	4	15	4	

Table 5: Fourth iteration

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12 **1**	12	1
O_2	15	30	10 **2**	35	13	17	2
O_3	20	15	20	45	15 **3**	15	3
O_4	5 **2**	10 **4**	50	30	14 **1**	13	7
O_5	34	32 **1**	36	40 **4**	38	16 **4**	9
O_6	22	23	21 **0**	27	25 **10**	28	10
	2	5	2	4	15	4	

Table 6: Fifth iteration

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12 **1**	12	1
O_2	15	30	10 **2**	35	13	17	2
O_3	20	15	20	45	15 **3**	15	3
O_4	5 **2**	10	50	30	14 **5**	13	7
O_5	34	32 **5**	36	40 **0**	38	16 **4**	9
O_6	22	23	21 **0**	27 **4**	25 **6**	28	10
	2	5	2	4	15	4	

Table 7: Sixth iteration and optimal solution

	D_1	D_2	D_3	D_4	D_5	D_6	
O_1	10	15	40	25	12 **1**	12	1
O_2	15	30	10	35	13 **2**	17	2
O_3	20	15	20	45	15 **3**	15	3
O_4	5 **2**	10	50	30	14 **5**	13	7
O_5	34	32 **5**	36	40 **0**	38	16 **4**	9
O_6	22	23	21 **2**	27 **4**	25 **4**	28	10
	2	5	2	4	15	4	

3. Assignment and scheduling

An assignment problem consists of a set of workers and a set of tasks. We need to assign tasks to workers with the least possible cost. This problem is generally solved by the Hungarian method, but it could be solved also by the transportation algorithm where $a_i = 1$ and $b_j = 1$ for all i and j.

Scheduling is an up to date problem in which we search for the optimal timetable affectation. It could be modeled as an assignment problem, as in the following example computed by the "OR Transport" program. Instructors availability is assigned to groups availability. Actually, suppose that 10 instructors offer training sessions to 5 groups of workers in a week.

Consider the following constraints:

- Groups availability: A group may be involved in other tasks and will not be available at the same time with the corresponding instructor.
- Instructors availability: Each instructor suggests different periods per week for his training session.
- Working hours: Assume that the total number of working hours is 15 per week, 5 days and 3 training sessions per day. Period number 3 indicates the third hour on Monday, period 5 is the second period on Tuesday and period number 13 is the first period on Friday. Tables 8 and 9 illustrate a given situation.

Table 8: Instructors availability

Instructor	**Availability hours: 1,..,15**	**Affiliated groups**
Instructor 1	3,6,9,13,15	1,3,5
Instructor 2	1,4,7,10,12,15	1,2,4,5
Instructor 3	3,5,7,10,14	1,3,5
Instructor 4	8,14	2,3,4,5
Instructor 5	7,14	1,3,4,5
Instructor 6	8,12	2,3,5
Instructor 7	10,12	1,2
Instructor 8	5,9	2,3,5
Instructor 9	2,10	1,3,4,5
Instructor 10	5,10	2,3,4,5

Table 9: Groups availability

Group	**Availability**
1	1,2,4,6,9,11,14
2	1,3,6,8,11,13
3	7,10,15
4	4,8,11,13
5	3,5,10

Suppose that each group needs to follow only one training session, the transportation program suggests the following assignments given by table 10.

Table 10: Suggested assignment solution

Group	Period	Instructor
1	***2***	***Instructor 9***
1	6	Instructor 1
1	9	Instructor 1
1	14	Instructor 5
2	***1***	***Instructor 2***
2	8	Instructor 6
3	7	Instructor 5
3	10	Instructor 3
3	***15***	***Instructor 1***
4	4	Instructor 2
4	***8***	***Instructor 4***
5	3	Instructor 3
5	***5***	***Instructor 3***
5	10	Instructor 2

4. Reducing computations

Assignments results in Table 10 are obtained by a transportation code "OR Transport". Problem modeling generates a costs matrix with 150 rows and 76 columns (11400) elements equal to a big number 1000000 or 1. Vectors A and B contain 1 and 0 only. As one value of unitary cost may be repeated many times in the costs matrix, more hypotheses are suggested in the least coast algorithm. For a set of computations on one hundred costs matrices, the choice is made at each step between all equal unitary costs, the one that has the greater sum for all active elements belonging to its row and to its column. For instance:

$$C = \begin{pmatrix} 3 & 5 & 4 & 2 \\ 1 & 6 & 6 & 9 \\ 8 & 10 & 1 & 11 \\ 7 & 8 & 15 & 16 \end{pmatrix} \qquad c_{21} = 1 \text{ and } c_{33} = 1$$

The sum for c_{21} is 3 + 8 + 7 +6 + 6 + 9 = 39. The sum for c_{33} is 8 + 10 + 11 + 4 + 6 + 15 = 54. We choose then c_{33} .

After, when this element has equal offer and demand quantities in the least cost algorithm, the saturation between the line or column with highest unitary transportation cost between them trying to eliminate elements with highest costs from the base is the objective.

In addition, when the variable quitting the base at each base renewal in the (U- V) algorithm is chosen, it will not be chosen randomly anymore between other variables allowed to go out the base, but it must have the greatest unitary cost between them for the same reason as before.

As in costs matrices obtained from a scheduling modeling elements are repeated many times, these criteria are applied to series of cost matrices with m = n = 10 till m = n = 160 (25600 elements) such that:

$$c_{ii} = 1 \; for \; i = 1, \dots ,120$$

$$c_{ii+1} = 10 \; for \; i = 1, \dots ,120$$

$$c_{i+1\,i} = 20 \; for \; i = 1, \dots ,120$$

$$c_{ii+2} = 5 \; for \; i = 1, \dots ,120$$

$$c_{i+2\,i} = 15 \; for \; i = 1, \dots ,120$$

And all other elements of the matrix are random costs between 0 and 2000 (A and B contain only 1). Figure 1 is an example of C when m = n = 8.

Figure 1: Matrix form

1	10	5	1357.5	553.85	877.49	1418.7	1919.5
20	1	10	5	92.343	763.12	1509.4	680.77
15	20	1	10	5	1531	552.05	1170.5
1826.8	15	20	1	10	5	1359.4	447.62
1264.7	1914.3	15	20	1	10	5	1502.5
195.08	970.75	71.423	15	20	1	10	5
557	1600.6	1698.3	1412.1	15	20	1	10
1093	283.77	1868	63.666	68.892	15	20	1

Figure 2 shows the iterations number needed to converge for one hundred matrices (120 rows 120 columns) by choosing the greatest elements sum on row and column each time. For each matrix, iterations number are written with and without supplementary restrictions on the transportation algorithm, which corresponds to four curves. In other words, four computations are done to each matrix between the one hundred (120 rows 120 columns) costs matrices and only random costs between 0 and 2000 change from a matrix to another between these hundred matrices. Figure 3 shows same results on a set of one hundred (160 rows 160 columns) matrices.

Figure 2: (m = n =120)

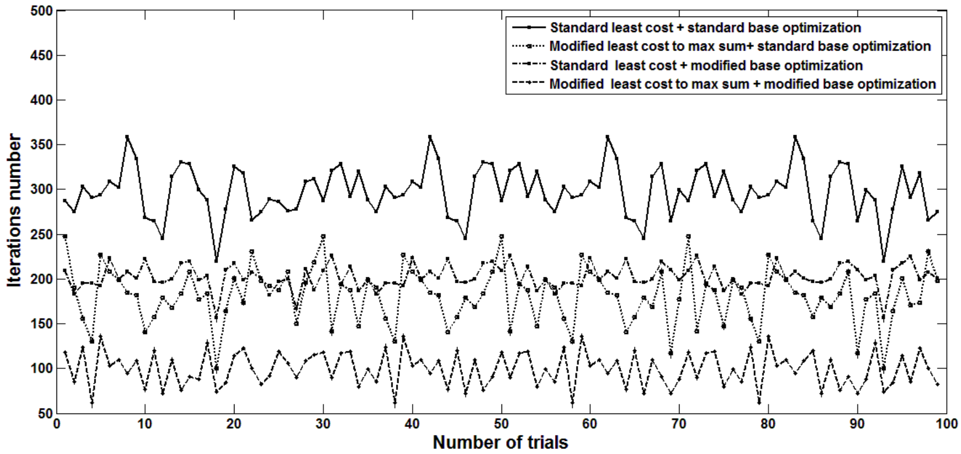


Figure 3: (m=n=160)

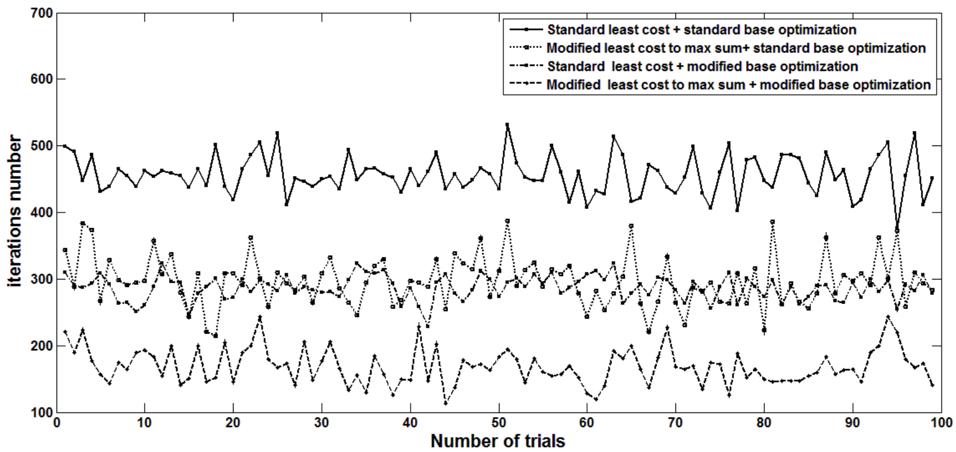
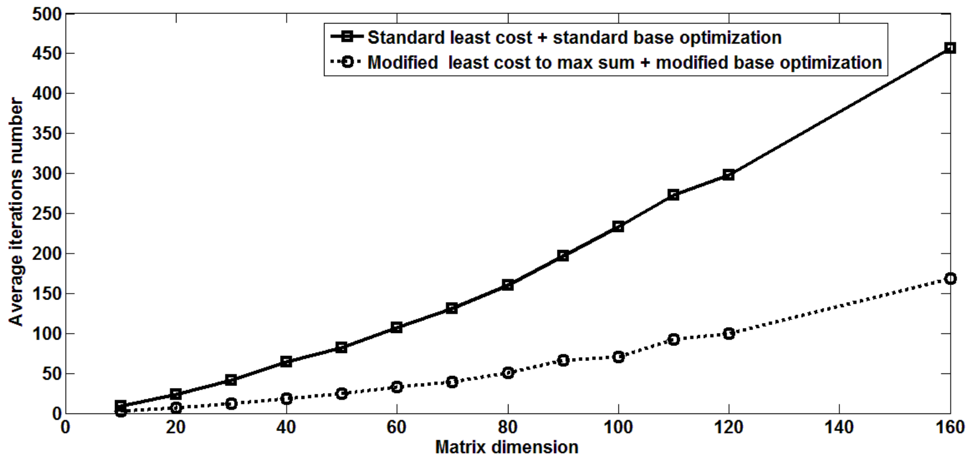


Figure 4 and Table 11 show same results in two different ways. They correspond to averages obtained for 13 tests on costs matrices by adding each time 10 rows and 10 columns. Each test between the 13 ones consists of 30 computations by changing randomly each time the costs matrix but keeping the size of this matrix and the matrix form given in Figure 1.

Figure 4: Averages before and after optimization



Adding constraints on algorithm gives better results. Figures 2 and 3 show that choosing elements with greatest sum on row and column reduces computations to 38%.

Table 11: Averages before and after optimization

m = n =	Standard least cost + standard base optimization	Modified least cost to max sum + modified base optimization
10	8.7	2.8
20	23.7	7
30	41.7	11.8
40	63.7	17.8
50	82.2	24.8
60	106.6	32.6
70	131	39.5
80	160.6	50.4
90	197.2	65.8
100	233	70.2
110	272.8	92.7
120	297.8	99.47
160	456.45	169

More in details, variations of the curves shown in Figure 2 and in Figure 3 are nearly homogeneous for all computations. The constraint in the base renewal is improving the iterations number for approximately 62 %. Another 62 % is guaranteed by the supplementary constraint on the least cost algorithm. This explain the resulting 38 % $(0.62 \times 0.62 = 0.38)$ mentioned before. It is obvious in Figure 4 that the proposed amelioration reduces in all cases iterations number to less than the half. It is about the third in most small sizes matrices.

5. Conclusion

If we suppose in a real case study which resembles to the timetable problem solved in Table 10 that all instructors give the same time availability, it is obvious that the ideal solution of this problem does not exist. The assignment approach is important because it finds the best existing solution, but with a considerable amount of computations on a particular form of costs matrix, where 0 and 1 are the only values in offer and demand vectors as mentioned before. The computational scheme related to these facts showed efficiency too on the suggested assignment solution, but many tests are to be done once the assignment-modeling algorithm is optimized to handle complicated time constraints.

Reference List

Cauvery, N. K. (2011). Timetable scheduling using graph coloring. *International Journal of P2P Network Trends and Technology, 1,* (2), 57 - 62.

Leighton, T. (1979). A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards, 84,* 489-506.

Loch, G. & Da Silva, A. (2014). A computational study on the number of iterations to solve the transportation problem, *Applied Mathematical Sciences, 8, (92)*, 4579 - 4583.

Michalewicz, Z. & Fogel, D. (2000). *How to solve it: Modern Heuristics*. Springer, 554.

Nanda, A., Pai, M., & Gole, A. (2012). An algorithm to automatically generate schedule for school lectures using a heuristic approach. *International Journal of Machine Learning and Computing, 2, (4)*, 492-495

Saad, W. (2004). Introduction à la recherche opérationnelle, Prana publishers.

Silver E., Vidal, I. & Werra, D. (1980). A tutorial on heuristic methods. *European Journal of Operational Research, 5,* 153-162.

Werra, D. (1985). *An introduction to timetabling*. Swiss Federal Institute of Tech-nology. Lausanne, Switzerland.