Available online at http://scik.org
J. Math. Comput. Sci. 11 (2021), No. 6, 7511-7522
https://doi.org/10.28919/jmcs/6654
ISSN: 1927-5307

LOGISTICS NETWORKS: A SPARSE MATRIX APPLICATION FOR SOLVING THE TRANSSHIPMENT PROBLEM

TARUN KUMAR GARG[1,*], PRABHAT MITTAL[2]

[1]Department of Mathematics, Satyawati College (University of Delhi), Delhi 110052, India

[2]Department of Commerce & Management, Satyawati College (Eve.) (University of Delhi), Delhi 110052, India

Abstract: Effective and efficient supply chain management is an important concern for most companies primarily in the Retail Industry. The reduced cost and enhanced customer service levels stemming from a better inventory control approach presents opportunities for the implementation of new approaches like using Sparse Matrix for solving transshipment problems. A Sparse matrix with a significant number of zeros can better represent a transshipment model. This is represented by three parallel linear arrays viz., *row_no., col_no., value*. The present paper is an attempt to efficiently represent the lateral transshipment model in the form of three linear arrays of the Sparse Matrix. At the end of the paper, algorithms to read, display, and add sparse matrices using the linear array representations are also discussed.

Keywords: supply chain; transshipment; sparse matrix.

2010 AMS Subject Classification:15A99, 65F50, 68M20.

1. INTRODUCTION AND LITERATURE REVIEW

In the past decade, the effective and efficient management of the supply chain flows (inward/outward) has become critical for most companies especially in the retail Industry. The companies need to follow a better inventory control approach at a lower cost and attain higher

*Corresponding author
E-mail address: tkgarg@satyawati.du.ac.in
Received August 16, 2021

customer service levels. The delivery of products at lower cost and less time can be achieved with efficient transshipment models. New approaches like Sparse Matrix can help to optimize the transshipment models [1], [2]. The companies like International data corporations are investing huge amounts annually on information technology for the management of the supply chain. Most companies have not changed their supply chain policies despite the expanded scope of SCM that now encompasses strategic sourcing, supplier involvement in product development, and customer fulfillment processes in addition to the movement of materials [3], [4].

Assume that customer demands from the different retail locations in a city/district are assigned to primary stocking facilities (say, a warehouse in that city), as shown in Figure 1. Transshipment occurs when goods are shipped to the customer from a facility other than the supplier, like directly from the vendor or other retail locations. The reason for using transshipments is that when customer demand is greater than on-hand inventory at a primary stocking facility, inventory from other (secondary) locations can be used to avoid stock out. While the primary drawbacks associated with transshipments are increased transportation costs and response times, the chief benefit is increased inventory availability or, conversely, reduced safety stocks.

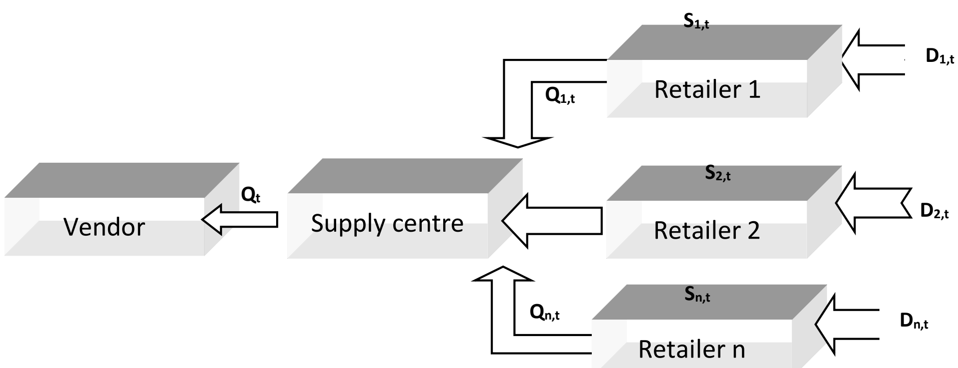

Fig. 1. Three stage-supply chain network [5]

On the downside, a transshipment policy can lead to some orders being filled from multiple locations; this situation can have a deleterious effect on customer service [6]. One possibility is that customers might receive split shipments from multiple stock-keeping locations. Holding overall volume constraint, as more shipments are made, the number of customer receipts increases, and the total transportation cost associated with these smaller shipment sizes increases. Moreover, additional paperwork is created, shipment monitoring efforts are increased, and chances of errors grow. As a result, customer service would be adversely affected. A second, and perhaps preferred, possibility is that product transshipped from secondary locations is combined with the product shipped from the primary location prior to receipt by the customer. This option gives rise to increased carrying costs as shipments must await one another at the marrying location. From the customer's perspective, this results in increased order cycle times. To counter increased cycle times, premium transportation services can be used to reduce transit times. Consequently, while customer service levels would remain unchanged, transportation costs would increase.

On the upside, according to [7], transshipments between two locations where warehouse replenishment lead times were negligible. The study found that the use of transshipments increases service levels without increasing inventory levels because of demand pooling across locations. The analysis was expanded to the case of non-negligible replenishment times by [8]. The study by [9] examines the effects of the lateral transshipment approach in three-stage supply chain network with one supply source at higher levels and multi-locations at the lower side. A series of experimental studies were carried out under different situations. The study concluded that the transshipment approach was effective compared to stock level equalization and optimization methods. Most of previous studies have considered inventory consolidations in case of transshipment and have concluded that inventory availabilities improve leading to a higher level of customer service [10], [11].

The problem of transshipment allocations can be associated with risk pooling from the different retail locations and has been the area of interest for researchers in inventory management[12]. The transshipment approach is a monitored movement of items and provides an efficient and optimal solution of transportation between the locations of demand and availability at the other centers. Thus, the implementation of transshipment policy leads to a reduction in the carrying costs and a decline in safety stock levels.

The enhanced quality of service to the customers and the inventory management has been the major concern of the managers today and have become the primary objective in the development of strategies for logistics network. The application of AI and Big data in the supply chain have also helped managers improving efficiency and managing logistics [13], [14]. [15] examined the performance of supply chain models and summarized the performances measures as a high level of efficiency, high level of customer service, and the ability to respond to a changing environment—where either of these must coincide with the firm’s strategic goals. A large number of algorithms have been developed based on various versions of network flows and optimal transshipments [16]–[18] problems. A study by [19] presented models and periodic review analysis of supply contracts with fixed ordered quantities. The model assumed no change in the order quantities at the buyers' cost. The problem was developed as an optimization of an integer linear programming problem.

Dynamic programming problems have also been very popular and have many applications such as distribution system, communication systems, and logistics scheduling. Previous research work has developed models on dynamic flow problems either as networks with discrete or continuous-time basis. The first approach uses time-expanded networks to produce theoretically or practically efficient algorithms. The second approach considers networks with time-varying capacities and costs and focuses on proving the existence of optimal solutions. [20] developed mixing the two approaches by extending some of the polynomial-time algorithms that work in the discrete-time model to solve the analogous continuous-time dynamic flow problems. Such problems include finding maximum dynamic flows and dynamic transshipments. [8] used emergency lateral transshipments to overcome the problem of shortage on a two-location periodic review system. Later [12] described three location pooling groups using emergency lateral transshipments. A stochastic approximation algorithm based on Infinitesimal Perturbation Analysis (IPA) was defined by [21] on N retailers to calculate the values of order-up-to-quantities.

Matrices are used in different branches of science and engineering, from simple mathematics to complex algorithms [22]. A matrix is said to be a sparse matrix if it has a large proportion of zeros among its elements [23]. Sparse matrices generally represented in the form of two-dimensional arrays [24] are used in different fields like circuit analysis, graph theory, deadlock handling, etc. In-circuit analysis, sparse matrices are used as incidence matrices, loop matrices, and cutset matrices [25]. Sparse matrices can also be used in Bankers' algorithm for deadlock avoidance, deadlock detection, and recovery from deadlocks [26]. The present paper is an attempt to apply the representation of Sparse Matrix to solve a transshipment problem.

2. PROBLEM CONCEPTUALIZATION

Let there be a periodic review inventory system with one supplier and $\boldsymbol{m}$ non-identical retailers. Each retailer is associated with a distinct and fixed location. Let each retailer sell any subset of the set of $\boldsymbol{n}$ distinct items. Lateral transshipments are executed to meet the observed demand at any location with low stocks. The advantage of using transshipments is in gaining a source of supply whose reaction time is shorter than that of the regular supply. The current paper develops a schematic model of such a supply chain.

The model introduced in the current paper uses two simple data structures to formally define the problem. Let there be an availability matrix, $\boldsymbol{A}$, of order $\boldsymbol{m}$x$\boldsymbol{n}$. $\boldsymbol{A_{ij}}$ represents the number of pieces of the $\boldsymbol{j}$th item available in the $\boldsymbol{i}$th store (Fig. 1). Again, let there be a transshipment cost matrix, $\boldsymbol{C}$, of order $\boldsymbol{m}$x$\boldsymbol{m}$. $\boldsymbol{C_{ij}}$ represents the cost of a typical transshipment from the $\boldsymbol{i}$th retailer to $\boldsymbol{j}$th retailer. It is assumed that the two data structures, $\boldsymbol{A}$ and $\boldsymbol{C}$, are centrally located and easily accessible by all the $\boldsymbol{m}$ retailers. Consequently, concurrency control mechanisms must be employed for the proper synchronization of the accesses by the various retailers.

The transshipment cost matrix, $\boldsymbol{C}$, can be configured at the initiation of the supply chain management system and it seldom requires any modification. If the transshipment cost matrix requires any modification, it can be made by any of the retailer or a coordinator elected by the retailers. However, the availability matrix, $\boldsymbol{A}$, requires frequent modifications. These modifications are to be carried out by various retailers. There are mainly three activities in the supply chain management system that call for modification of the availability matrix, $\boldsymbol{A}$. These activities are (i) periodic procuring of stocks form the lone supplier, (ii) selling of an item to a customer, and (iii) transshipping an item to another retailer. Out of these three activities, the first one is expected to occur least frequently. The best way to implement these activities is through a trigger mechanism. The triggers for these three activities can be defined as follows.

Periodic procuring of stocks form the supplier: If the $\boldsymbol{i}$th retailer procures $\boldsymbol{x}$ pieces of the $\boldsymbol{j}$th item, update $\boldsymbol{A_{ij}} \leftarrow \boldsymbol{A_{ij}} + \boldsymbol{x}$.

Selling of an item to a customer: Let us consider a situation where a customer arrives at the $\boldsymbol{i}$th retailer as asks for $\boldsymbol{x}$ pieces of the $\boldsymbol{i}$th item. If $\boldsymbol{A_{ij}} \geq \boldsymbol{x}$, satisfy the customer's demand and update $\boldsymbol{A_{ij}} \leftarrow \boldsymbol{A_{ij}} - \boldsymbol{x}$. Otherwise, determine the set, $\boldsymbol{S}$, of all retailers, such that, $\boldsymbol{A_{il}} \geq (\boldsymbol{x} - \boldsymbol{A_{ij}})$. Now select $\boldsymbol{k} \in \boldsymbol{S}$, such that, $\boldsymbol{C_{jk}} \leq \boldsymbol{C_{jl}}\ \forall \boldsymbol{l} \in \boldsymbol{S}$. Finally, reset $\boldsymbol{A_{il}} \leftarrow 0$. It is assumed that the $\boldsymbol{k}$th retailer updates using the trigger for the transshipment activity.

Transshipping an item to another retailer: If the $\boldsymbol{i}$th retailer transships $\boldsymbol{x}$ pieces of the $\boldsymbol{j}$th item to another retailer, update $\boldsymbol{A_{ij}} \leftarrow \boldsymbol{A_{ij}} - \boldsymbol{x}$.

3. DESIGN AND IMPLEMENTATION ISSUES

In the Section 2, the basic framework for the problem has been formalized. The basic model can be enhanced by considering varied design and implementation issues as listed next.

i. *Similar items* – Some items can be considered similar by a customer though they are considered dissimilar by others. For example, some customers can consider detergent bar as a supplement of the detergent powder.

ii. *Transshipping from multiple retailers* – If the necessary number of pieces of a particular item is not available with a retailer and no other single retailer can spare the required number of pieces, then transshipments can be made from multiple retailers.

iii. *Preemptive transshipments* – This is a conservative policy where transshipments are made by a retailer when the stock of a particular item runs lower than a preset threshold T_{ij}, i.e., $A_{ij} \geq T_{ij}$.

iv. *Bulk transshipments* – This is another conservative policy where $x + \Delta$ pieces of an item are transshipped when only x pieces of that item are required immediately.

4. OPTIMIZATION USING SPARSE MATRIX STORAGE MODELS

Each element of the transshipment cost matrix, C, will always has a valid value. Since, each of the m retailers does not sells all the n items, one or more elements of the availability matrix, A, may be empty. For large values of m and n, the degree of sparsity increases. Hence, the availability matrix, A, can be efficiently modeled using any of the many available techniques to model sparse matrices. In this paper, a simple and yet memory efficient model (Chakraborty, 2006) has been selected for representing a sparse availability matrix.

A sparse availability matrix can be represented by three parallel linear arrays. Information is stored only about the non-zero elements of the availability matrix. The linear arrays named *retailer_no*, *item_no* and *items* stores the row numbers, the column numbers and the values of all the non-zero elements of the availability matrix respectively. A sparse availability matrix of dimension mxn, stored using this model can be easily retrieved using a simple algorithm of O (mxn). Simple matrix operations which are needed by the triggers defined in Section 2, like addition and subtraction, are supported directly by this sparse matrix model.

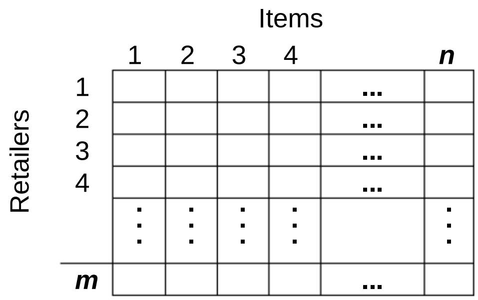

Fig. 2. The layout of a typical availability matrix A.

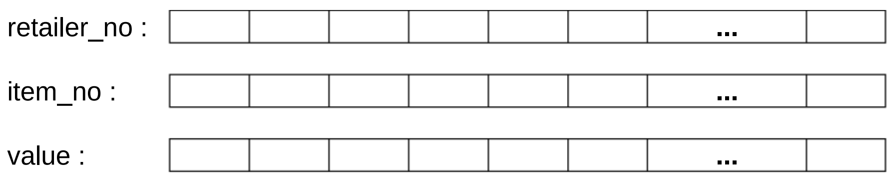

Fig. 3. Representation of a sparse availability matrix using three parallel linear arrays.

For example

Table 1: Considering availability matrix for eight different retailers with six different Articles

		Item no. (codes)					
Retailers	Store code	1 (108015133)	2 (108017243)	3 (108020368)	4 (108030061)	5 (108030066)	6 (108030113)
I	3305	0	0	0	144	0	0
II	3307	20	0	0	144	0	42
III	3309	40	0	40	0	0	0
IV	3310	20	0	0	0	0	0
V	3311	0	0	0	144	30	0
VI	3323	0	40	40	144	32	0
VII	3332	0	0	0	0	0	84
VIII	5433	20	0	40	144	60	0

Note: ‘0’ implies non-availability of the items

Table 2: Sparse availability matrix of the above representation

Retail location	I	II	II	II	III	III	IV	V	V	VI	VI	VI	VI	VII	VIII	VIII	VIII	VIII
Item No.	4	1	4	6	1	3	1	4	5	2	3	4	5	6	1	3	4	5
Item No.	144	20	144	42	40	40	20	144	30	40	40	144	32	84	20	40	144	60

The sparse matrices are assumed to have an order of 10 x 10 and 20 non-zero elements (at most) for the sake of simplicity. For a variable number of non-zero elements a dynamic memory allocation scheme should be followed.

```
typedef  sparse_matrix
        int row_no[20],col_no[20],value[20] // The three linear arrays
        int length = 0
```

Algorithm

The algorithms are kept simple to increase readability. However, they can be optimized for fast execution.

Algorithm: Reading a sparse matrix

```
int num
sparse_matrix S
for i ← 1 to 20
        read S.row_no[i], S.col_no[i], num
        if num = 0
                break   // Stop reading if value entered is 0
        else
                S.value[i]=num
                length = length + 1
return S
```

Algorithm: Displaying a sparse matrix

```
(Input: sparse_matrix S)
boolean flag
for i ← 1 to 10
        for j ← 1 to 10
```

```
        flag = false
        for k ← 1 to S.length
            if S.row_no[k] = i and S.col_no[k] = j
                print S.value[k]
                flag = true
                break
        if flag = false
            print 0
        print tab_space_character
    print new_line_character
```

Algorithm: Adding two sparse matrices

```
(Input: sparse_matrix S1, sparse_matrix S2)
sparse_matrix S
for i ← 1 to S1.length
    for j ←1 to S2.length
        if S1.row_no[i] = S2.row_no[j] and S1.col_no[i] = S2.col_no[j]
            S.row_no[S.length] = S1.row_no[i]
            S.col_no[S.length] = S1.col_no[i]
            S.value[S.length] = S1.value[i] + S2.value[j]
            S.length = S.length + 1
            S1.value[i] = 0
            S2.value[j] = 0
            break
for i ← 1 to S1.length
    if S1.value[i] != 0
        S.row_no[S.length] = S1.row_no[i]
        S.col_no[S.length] = S1.col_no[i]
        S.value[S.length] = S1.value[i]
        S.length = S.length + 1
for j ← 1 to S1.length
    if S2.value[j] != 0
```

```
        S.row_no[S.length] = S2.row_no[j]
        S.col_no[S.length] = S2.col_no[j]
        S.value[S.length] = S2.value[j]
        S.length = S.length + 1
return S
```

5. CONCLUSION

The present study concludes that linear array representation of sparse matrix for a transshipment model is more efficient since it does not require the non-available items at a retail location to present in the matrix against a two-dimensional classical array form of the matrix. Basic operations like addition and subtraction are required in the matrix, while the movement of the items during lateral transshipments is taking significantly less time and can be performed using an algorithm in section 4.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] Y. Wang, J. Cao, L. He, W. Li, L. Sun, P.S. Yu, Coupled Sparse Matrix Factorization for Response Time Prediction in Logistics Services, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, Singapore Singapore, 2017: pp. 939–947.

[2] X. Huang, H. Xiong, A. Zou, H. Li, Multi-distribution center inventory demand forecast model in regional logistics network, in: 2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management, IEEE, Xiamen, China, 2010: pp. 1360–1363.

[3] P. Mittal, E-Supply Chain & Digital Technologies: A Vision, IME J. 5(2)(2010), 9–11.

[4] S. Sharma and P. Mittal, Managing Bullwhip and Pooling Effect under Spatially-correlated Demand in FMCG Retailing, MAIMS J. Manag. 7(1)(2012), 13–18.

[5] S. Sharma and P. Mittal, Quantifying the Bullwhip in Multi-location Supply Chain Network: The Impact of Demand Forecasting, BULMIM J. Manag. Res. 1(1)(2016), 55.

[6] X. Xiao, L. Qi, Z. Fu, and X. Zhang, Monitoring method for cold chain logistics of table grape based on compressive sensing, Nongye Gongcheng Xuebao/Transactions Chinese Soc. Agric. Eng. 29(22)(2013), 259–266.

[7] G. Tagaras, Effects of pooling on the optimization and service levels of two-location inventory systems, IIE Trans. Institute Ind. Eng. 21(3)(1989), 250–257.

[8] G. Tagaras and M. A. Cohen, Pooling in two-location inventory systems with non-negligible replenishment lead times, Manage. Sci. 38(8)(1992), 1067–1083.

[9] J. Burton, A. Banerjee, Cost-parametric analysis of lateral transshipment policies in two-echelon supply chains, Int. J. Prod. Econ. 93–94(Spec.Iss.)(2005), 169–178.

[10] A. Bhatia and P. Mittal, Big Data Driven Healthcare Supply Chain: Understanding Potentials and Capabilities, SSRN Electron. J., 2019. https://doi.org/10.2139/ssrn.3464217

[11] P. Mittal, Impact of Auto-regressive (AR) Process in Bullwhip Analysis in a Multi-location Supply Chain Network, J. Bus. Manag. Inf. Syst. 6(1)(2019), 19–26.

[12] G. Tagaras, Pooling in multi-location periodic inventory distribution systems, Omega, 27(1)(1999), 39–59.

[13] P. Mittal, Impact of Digital Capabilities and Technology Skills on Effectiveness of Government in Public Services, in 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020, 2020, pp. 1–5.

[14] P. Mittal, Big data and analytics: a data management perspective in public administration, Int. J. Big Data Manag. 1(2)(2020), p. 152.

[15] B. M. Beamon, Measuring supply chain performance, Int. J. Oper. Prod. Manag. 19(3)(1999), 275–292.

[16] J. Meissner and O. V. Senicheva, Approximate dynamic programming for lateral transshipment problems in multi-location inventory systems, Eur. J. Oper. Res. 265(1)(2018), 49–64.

[17] L. Tiacci and S. Saetta, A heuristic for balancing the inventory level of different locations through lateral shipments, Int. J. Prod. Econ. 131(1)(2011), 87–95.

[18] P. M. Reyes and L. M. Meade, Improving reverse supply chain operational performance: A transshipment application study for not-for-profit organizations, J. Supply Chain Manag. 42(1)(2006), 38–48.

[19] T. L. Urban, Supply contracts with periodic, stationary commitment, Prod. Oper. Manag. 9(4)(2000), 400–413.

[20] L. Fleischer and É. Tardos, Efficient continuous-time dynamic network flow algorithms, Oper. Res. Lett. 23(3–5)(1998), 71–80.

[21] M. Madanoglu, M. Kizildag, and O. Ozdemir, Which bundles of corporate governance provisions lead to high firm performance among restaurant firms?, Int. J. Hosp. Manag. 72(2018), 98–108.

[22] M. Hershkowitz and S. B. Noble, Finding the inverse and connections of a type of large sparse matrix, Nav. Res. Logist. Q. 12(1)(1965), 119–132.

[23] S. Lipschutz, Schaum's Outline of the Theory and Problems of Data Structures. Tata McGraw-Hill. (1986).

[24] R. B. Patel and M. M. S. Rauthan, Expert Data Structures with C++. Khanna Book Publishing Company, 2019, p. 658.

[25] T. K. Soni, Impact of Ownership Structure on Dividend Yields: A Neural Network Approach, SSRN Electron. J. 2012. https://dx.doi.org/10.2139/ssrn.1841629

[26] PB Galvin, G Gagne, A Silberschatz, Operating System Concepts. John Wiley and Sons (ASIA) Pvt. Ltd. 2003.