Advances in Environmental Biology, 8(11) Special 2014, Pages: 1160-1167

AENSI Journals

Advances in Environmental Biology

ISSN-1995-0756 EISSN-1998-1066

Journal home page: http://www.aensiweb.com/AEB/

Using Mathematical Planning Methods for Solving the Transportation Problem

Seyede Najmeh Hosseini, Maryam Amiri, Azar Mohamadi Deloei

Department of Mathematical, Payam Noor University, Tehran, Iran

ARTICLE INFO

Article history:
Received 15 April 2014
Received in revised form 22 May 2014
Accepted 25 May 2014
Available online 15 June 2014

Keywords:
traffic assignment, balance of user, optimization, mathematical planning methods

ABSTRACT

Today, use of optimization methods is very common in solving complex issues. One of the important issues in the science of transportation is traffic assignment problem. Many researchers of mathematical and traffic science used a lot of optimization methods to solve it so far, all of them concern which method has the ability to solve this issue in minimum time and with the highest precision. With attention to the convex nature of the traffic assignment problem, many planning methods have been used to solve this problem which are introduced and explored in the study. Understanding the types of procedures to resolve this issue helps researcher to investigate different strengths and weaknesses of these methods and upgrade the current procedures via eliminating the existing shortcomings or combining these methods with each other. Also, this helps researchers to introduce a new method by identifying the issue and other methods of optimization.

To Cite This Article: Seyede Najmeh Hosseini, Maryam Amiri, Azar Mohamadi Deloei, Using Mathematical Planning Methods for Solving the Transportation Problem. ***Adv. Environ. Biol., 8(11),*** *1160-1167, 2014*

INTRODUCTION

Today, the concept of optimization has been completely accepted as a infrastructure principle in the analysis of many complex issues of decision making or assigning. This concept involves a certain philosophical elegance that is difficult to be doubts, and often makes calculations easier to the extent that is a necessity.

Since urban transport modeling system (UTMS), which is used routinely in the planning of transport, includes four steps including creating a trip, travel distribution, select a vehicle and traffic assignment, normally distinct models are used for each stage to meet the demands. At the stage of creating a trip, the goal is to estimate produced and absorbed trips for each district. The relationship between produced and absorbed trips in different areas is determined in the form of a table of origin-destination. The contribution of each one of the shipping methods to meet the demands can be determined in the distribution of travel and choice of vehicle steps, respectively. In the end, at the stage of traffic assignment, estimated travel demand might be assigned to transport supply system. Input of this stage is origin-destination travel demand for various methods of transportation and characteristics of the network and output is considered to be traffic volume and the travel time on the arc transport network.

Traffic assignment model used in this study is static and deterministic and is based on the principle of balance of user. That is the rate of travel or origin-destination demand is fixed during a certain period. Network users have full information about travel time on the network and route choice by users is in line with the minimizing travel time. Since arc-based algorithms are very applicable for solving the issue of the allocation of traffic with steady demand, a variety of different algorithms in this area is presented in this study. A lot of techniques have been suggested in this field which tried to improve the performance of Frank-Wolfe method with eliminating shortcomings of this method or used mathematical planning methods to resolve the issue of assignment. In this article, a few examples of these methods will be introduced and compared. Other methods also exist, such as reduction techniques of the gradiandembo and Linkovich and convex Simplex 2 rocks Nguyen having been regardless in this study because of the high volume of entries. Those interested to get more information can refer to reference.

Assignment problem-solving:

In this problem, the effect of the flow of traffic on travel times of network arc is expressed via travel time functions. After two plans of principle of optimized drop, i.e. user balances and system optimization, modeling

Corresponding Author: Seyede Najmeh Hosseini, Department of Mathematical, Payam Noor University, Tehran, Iran

and techniques for solving traffic assignment problem (TAP) are widely studied. First time, Bechman *et al*. expressed the issue of user balanced traffic assignment as a planning mathematical model of convex minimizing for a constant demand with the assumption that the travel time in any arc is a function of the flow of traffic in the same arc which is descending.

$$minz\left(x\right)=\sum_{a\in A}\int_{a}^{x_a}t_a\left(\omega\right)d\left(t\right)(1)$$

$$\sum_{k\in k_{rx}}f_k^{rs}=q_{rs}\quad\forall r,s$$

$$S.t\int_{k}^{rs}\geq 0\ \forall r,s,k\in k_{rs}$$

$$x_a=\sum_{r\in R}\sum_{s\in S}\sum_{k\in k_{rs}}f_k^{rs}\delta_{ak}^{rs}\quad\forall a\in A$$

In this model that is known to the ARC path formulation, A is regarded as total number of network arcs and K is the total number of nodes between the origin of r and destination of s, ta(xa) is a function of the total travel time in an arc, xa is flow in arc a and qrs represents a constant demand between origin-destination r, s. f_k^{rs} is the amount of the flow in the rout of k between the origin-destination r, s and δ_{ak}^{rs} is the matrix component of occurrence of arc-path which is equal to 1 if a arc is located at k path between the origin and the destination r, s, and is zero otherwise. With regard to the structure of problem, Beckman tried a variety of mathematical techniques to solve this issue. The recognition of the most effective problem-solving method of allocation is of importance in science of transportation.

method of Frank-Wolfe:

Frank-Wolff method, that also known as convex combinations method, is one of the oldest methods of research in operations. This method was designed in 1956 in order to solve second order problems with linear projection conditions. Frank-Wolfe method works as many of the tractable mathematical planning methods; at first one vector is generated between a hypothetical point under the name of Xa inside the acceptable range and one other point in the name of ya which is achieved by solving the linear problem of Beckman conversion. In the next step using linear search methods such as two part search or golden split, the value of motor step of a is determined in the direction of produced vector.

Frank-Wolfe algorithm for solving the issue of the balance of traffic flow is as follow:

1-determine the initial values: all or nothing assi gnment done by ta (0) and determining x_a^1 and putting n = 1

2- update: put t_a^n = ta(x_a^n)

3. Determine the direction of movement: all or nothing assignment based on t_a^n and determining y_a^n

4-Linear search: momentum of an with solving the following:

$$\min z\left(x\right)=\sum_{a}\int_{a}^{x_a+a\left(y_a^n-x_a^n\right)}t_a\left(\omega\right)d\left(t\right)$$

$$S.t\ 0\leq a\leq 1$$

$$1-movement.insert:$$

$$x_a^{n+1}=x_a^n+a_n\left(y_a^n-x_a^n\right),\forall a$$

2-Stop criteria: If the criteria be achieved, stop, the current answer of x_a^{n+1} is balanced flow, or else put n = n + 1 and go to step 1.

Despite all the simplicity of performing the Frank-Wolfe method, it is not applicable for solving large-scale problems, because the number of variables is very high and a very long time is required. on the one hand, since search directions in the Frank-Wolfe method tend to a side point (on the border of acceptable range) and generated search line tend to get perpendicular to the gradient vector of goal function (for the sharpest reduction), so the Zigzag phenomenon is observed that cause this algorithm will converge very slow when the optimum point is approaching.

Advances in Environmental Biology, 8(11) Special 2014, Pages: 1160-1167

Method of Fukushima:

This method is trying to speed up the convergence of the original Frank-Wolf method via correcting direction of movement.

In order to ensure proper performance, this method compares new direction with Wolf-Frank method direction upon implementation and finally selects the best direction.

If we assume that x * (optimal answer of the main allocation issue) is located on the surface of s, and name points on this multifaceted convex surface e1, e2,...en, then when the k (the number of each stage) is large enough (k elementary stages in Frank-Wolfe method goes really good and in the end stages, the Zigzag phenomenon being observed) following response of the linear issue in Frank-Wolfe method, yk, should matches to the point at the end of this page, ej. The point x * can be written as a convex combination of ej:

$$x^* = \sum_{j=1}^{m} \mu_j e^j \sum_{j=1}^{m} \mu_j = 1 \mu_j \geq 0,\ j = 1,\ldots,m.(2)$$

So, if we can determine points of ej and μj, x* could be achieved from the following equation:

$$x^* = x^k + a_k d^k a_k = 1\ d^k = \sum_{j=1}^{m} \mu_j \left(e^j - x^k\right)(3)$$

Of course, it is not practically impossible to understand the values of μj and ej accurately. We hope we can obtain some useful information about the ej with saving previous created answers of yk (solving linear issue of the Frank-Wolfe method). In other words, a convex combination of the values of yk from previous stages may give us an approximate of dk and produce better a search line from dk = x, k yk- formed by using only the values of yk that have been created recently. Problem-solving assignment algorithm with Fukushima is thus:

1-Select a possible answer and name it x, let k = 1 and assume a positive value for l. We will consider l >1.

2-after solving the Frank-Wolf problem, name the resulting answers as yk.

3. If f (x, k) T (yk-x, k) = 0, stop or else go to next level

4-choose a λi so that i = k-q… k and put $\sum_{i=k}^{k}$ λi ≥ 0

(Yi – x, k) = νk xkλi = λi – $\sum_{i=k-q}^{k}$

q = min {k, l}

For example in the case of q = k-1 for k ≤ 1 and q = l-1 for k = 1, put wk > yk – x, k and go to step 4 (this step helps to introduce two search lines, one for Frank-Wolfe and the other for Fukushima).

5- Calculate derivatives below:

$$\gamma_1^k = \nabla f\left(x^k\right)^T v^k / v^k\ (\gamma_l^k = 0\ if\ \ v^k = 0)$$

$$\gamma_2^k = \nabla f\left(x^k\right)^T w^k / w$$

Insert

$$d^k = \begin{cases} v^k, if\ \gamma_1^k < \gamma_2^k \\ u^k, if\ \gamma_1^k < \gamma_2^k \\ w^k, if\ \gamma_1^k < \gamma_2^k \end{cases}$$

5. With the obtained movement direction value achieve the moving step as Frank-Wolfe, add a number to k and go to the next step.

Fukushima compared his method method in 1984 with Frank-Wolfe. He used the network of Cioxfalse with 24 nodes and 76 arcs to run this procedure while used different I values to investigate all aspects of invented method. It is entirely evident from the results that Fukushima improved Frank-Wolfe method.

For example, in the 20th repeat, the value of target function is equal to 45494 whilst this value has not been achieved up to repeat of 40 in Frank-Wolfe. Based on comparisons that Fukushima was done, it turned out that problem-solving time was reduced up to about 40% for example for a network with 100 nodes and 350arcs.

Gnsals et al. method:

Gnsals worked on the motor step and tried to compensate the ineffectiveness of the Frank-Wolfe method in converge to an optimal point by modifying it.

Gnsals raised that by defining a bigger motor step, can be easily escape from zigzag phenomenon, in figure (1) it is evident that how this method fasten converge of Frank-Wolfe algorithm. The dotted line shows the movement path of Gnsals algorithm.

Algorithm of assignment problem solving with the Gnsals method is as follows:

1-consider xk as a primary allocation issue answer (like Frank-Wolfe method)

2- Solve the linear problem of Wolfe-Frank and name the answer yk, name the motor vector as dk = yk-xk such as Frank-Wolfe method.

3-name the movement step obtained from Frank-Wolfe the a0.

4-insert ak = λka0, in fact, λk is considered as motor step modifier in kth repeat and λk >1 (λk=1 is indicative of Frank-Wolfe's method).

5-put ak = max (ak, 1)

6- Put the xk+1= xk + akdk. If f(xk+1)<f(xk) then xk+1=xk+1 and if f(xk+1)>f(xk) then xk-1= xk + akdk

so if the amended movement steps does not lead to an advance in the amount of target function, normal movement step of a0 could be used.

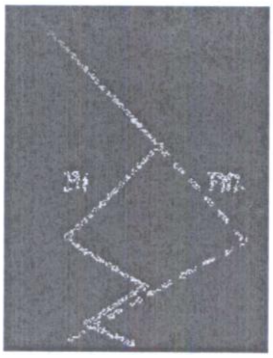

Fig. 1: Trend of movement of Gnsals algorithm against Frank-Wolfe.

Gnzans performed his proposed method in 1985 on many hypothetical networks. Results obtained from the Gnsals performances showed that his method can reduce the run time up to 55% in small LAN with 15 to 30 nodes, this run time reduction was more significant in large networks with 50 to 80 nodes and was reduced about 70%.

The most remarkable thing in Gnsals method is determining the amount of increase of moving step, so that if the numbers not be selected suitably, the performance of the proposed method might be worse than method of Frank-Wolfe.

Combinational method:

One of the most effective Frank-Wolfe modified algorithms is combinational algorithms that contain both modification of direction and step of moving. In this algorithm, Fukushima algorithm is used in order to modify the direction of motion and Gnsals algorithm is utilized for modifying movement step. How to combine these two methods is very important, a combination should be produced with the most efficiency in Frank-Wolfe method. Two combinations are introduced for combinational algorithm. in the first case, FWF is the main algorithm and step reduction techniques is used as an adjustment for this algorithm, and in the second one, FWλ is the main algorithm and Fukushima's criteria are regarded as a compensation. Since FWF operate on the direction of reduction leading to the principle of zigzag phenomenon in frank-Wolfe algorithm, then the best choice for combinational algorithm is selecting the first case. Identifying of new combinational algorithm is not difficult because of full knowledge of the FWλ and FWF algorithms.

In this algorithm we use most widescreen steps technique to accelerate the convergence of Frank-Wolfe until we have used the reduction direction of Frank-Wolfe method (in the first l repetition) and in order to prevent the phenomenon of a zigzag, we use Fukushima algorithm from the repetition of I.

FWFλ algorithm is as follows:

1-at starting, assume the x 1 is a possible point and put k = 1.

2. Determining the direction of movement: use Fukushima algorithm to determine dk.

3- Movement step: in order to find a, a linear search in the direction of dk from the previous phase should be performed. If k ≤ 1 use the FWλ technique and otherwise use the technique of FWF

4-If an ending criteria was satisfies, stop, otherwise put k = k + 1 and go to step two.

Arach and Koofi in 2008 used the network of Cioxphals city and other hypothetical network to demonstrate the efficiency of this method compared to the Frank-Wolfe ones. Results of the implementation of the algorithm shows that the response obtained in 27th repeat in this algorithm had been achieved in 55th repeat of Frank-Wolfe algorithm and this method also has many advantages upon Gnsals and Fukushima methods.

One of the most valuable works done by Arach and Koofi was implementing Gnsals and Fukushima methods on today's new computers and results of these performances is very useful for fully understanding these methods and comparison of performance of them.

Arach and Koofi showed that the running time in a network composed of 120 arcs and 40 nodes is 5.46, 3.94, 2.011 and 0.927 seconds in Frank-Wolfe, Gnsals, Fukushima, and the proposed method, respectively.

Leblank et al. method:

Leblank was one of the first people who was trying to modify the direction of movement to upgrade to the speed of convergence of Frank-Wolfe algorithm; so he utilized partan technique in order to determine the direction of movement in the in the Frank-Wolfe algorithm. partan is a technique for solving technical issues which could be used in the directional algorithms. This method is designed to increase the speed of convergence of Frank-Wolfe algorithm via addition of a linear search at each stage. This method produce its search direction from the current answer (current stage) and the answers obtained from the two previous iterations. It is evident in figure 2 that if produce a search line fromx0 to x1, we could reach to x * point faster.

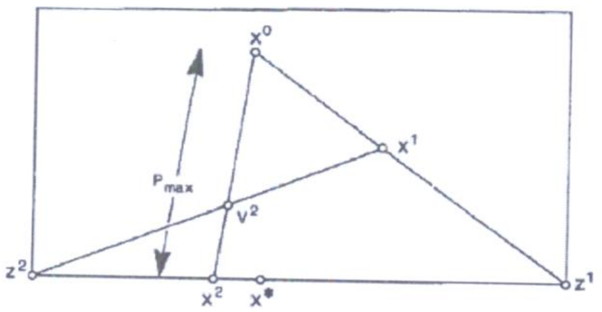

Fig. 2: trend of solving modified Frank-Wolfe algorithm with partan technique.

The modified Frank-Wolfe algorithm with partan technique is as follows:
1-1-problem: solve the standard Frank-Wolfe algorithm to each to y1as an optimal solution.
1-2-linear search in Frank-Wolfe: determine x so that we have the lowest amount of function in line with yk - vk vector.
K. 1 problem: solve the standard Frank-Wolfe algorithm at k repeat to each to yk as an optimal solution.
K. 2 linear search in Frank-Wolfe: if draw a line between xk-1 and yk, uk is a point on this line in which target function is minimum.
K. 3 partan search: determine xk point on the line passing through two points of xk-2 and vk in which target function is minimum, put k = k + 1 and go to the k stage.

When we use partan techniques, it is necessary to determine a value of pmax in each repetition which is actually the maximum of movement step length in the direction determined via partan techniques. The maximum of movement step length in the direction determined via Leblank technique could be obtained from separate formula in each stage and is only for the first 5 repetitions.

Leblank in 1985used the cioxphalse city network to confirm optimal performance of his proposed method.

In order to obtain more comprehensive results, he used a coefficient to change the values of the demand matrix. The values of the hypothetical coefficients were 0.8, 1, 1.2 and 1.4; so he resolved cioxphalse city network with 4 different demographic size using his proposed algorithms.

To achieve an answer produced by Leblank algorithm at repeat of 6, Frank Wolf needs 10, 7, 11 and13 repetitions for every 4 introduced models.

Generally, this method reduces the number of repetitions of algorithm up 53% compared to Frank-Wolfe and the more dense network, the better answers by FWP.

Hoi and Zivavi method:

Most of the methods of accelerating convergence of Frank-Wolfe algorithm have focused on determining search direction adding many computational works to the method. An innovative solution is using a linear search method other than the two parts or the golden split. There exist several well-known rules of linear search using non-uniform linear search technique such as Armiju, Goldstein and Wolf principle and it can be easily proved that Frank-Wolfe algorithm can end at a point to satisfy Kuhn-Tucker conditions or produce a set of responses which are converged to the Kuhn-Tucker point. Modified Frank-Wolfe algorithm using Armiju principle is as follow:

1-determine the initial values: all or nothing assignment done by ta (0) and determining x_a^1 and putting n = 1

2- update: put t_a^n = ta(x_a^n)

3. Determine the direction of movement: all or nothing assignment based on t_a^n and determining y_a^n . Put dk = yk – xk so that:

$$x_k = \left(x_1^k, \ldots, x_a^k, \ldots, x_A^k\right)^T \quad y_k = \left(y_1^k, \ldots, y_a^k, \ldots, y_A^k\right)^T \quad d_k = \left(d_1^k, \ldots, d_a^k, \ldots, d_A^k\right)^T$$

4-linear search: assume M is a non-negative integer default value and let that tk≤0 be limited from high to satisfy the following equation:

In fact, $\sigma^{h}{}_{1}a$ is the movement step in Frank-Wolfe method.

According to Frank-Wolfe that movement step was a number between zero and one, the upper limit is considered 1 for $\sigma^{h}{}_{1}a$ and reduce the amount of $\sigma^{h}{}_{1}a$ with γ coefficient to determine appropriate movement steps.

5-moving: put xk + 1 = x k + tkdk

6. test of convergence: stop if an ending criteria is satisfied otherwise put k = k + 1, and go to step 1.

Hoi and Zivavi in 2005 for demonstrating better performance of proposed method used a hypothetical network with two nodes and three arcs. the results obtained from the use of this algorithm in solving the issue of allocation shows that to achieve an answer with the same accuracy, 80 repetitions in Frank-Wolfe is needed while using this method, only 22 repetitions is required.

Patrikson et al. method (partial linearization):

In this method, a linear approximation to the non-separable breakdown of main goals function is created frequently. This method is actually an organized frank-Wolfe presenting better directions than frank-Wolfe method through keep non-linear main target function. The steps of solving this algorithm are as follow:

1. Write the main function using partial linearization method as following format hat is separable.

$$T(f) = \sum_{k\in C}\sum_{(i,j)\in A}\int_0^{f_{ij}} t_{ij}(s)ds + g(f),\; g(f) = \sum_{(i,j)\in A}\left\{\int_0^{f_{ij}} t_{ij}(s)ds - \sum_{k\in C}\int_0^{f_{ij}} t_{ij}(s)ds\right\}$$

A is arc set, C is pathways set, fij is total flow on (i,j) link which is equal to $\sum_{k\in C} f_{ij}^{k}$

2. Write T(f) function for each k path separately and name it PR

$$[SUB_k]\; min \sum_{(i,j)\in A}\int_0^{f_{ij}}(t_{ij}(s)+a_{ij}^k)ds$$

$$s.t \sum_{j\in w_i} f_{ij}^k - \sum_{j\in v_i} f_{ij}^k = \begin{cases} \sum_{i\in D_k} r_{ki} \; if \; i = o_k \\ -r_{ki} \; if \; i \in D_k \nabla i \in N \\ 0 \; otherwise \end{cases}$$

$$f_{ij}^k \geq 0$$

rki is optimal flow for the destination of i in k path, wi is a subset of nodes of the link started from node i and vi is the set of basic nodes of the link started from node i.

Each PR issue is a definite convex problem for the flow of each pathway and can be resolved with the mathematical planning methods.

Like convex simplex method or method of reducing the gradient, another method for solving ideal separable problems is the method of Dugan technique.

The first step is to write the Lagrange function of PR issue:

$$L(f,\pi) = \sum_{(i,j)\in A}\int_0^{f_{ij}}\left(t_{ij}(s)+a_{ij}^k\right)ds + \sum_{(i,j)\in A}(\pi_i - \pi_j)f_{ij} + \sum_{i\in D}(\pi_i - \pi_0)r_i$$

3- We write Lagrange function and introduce it to [PR-LD]

$$\theta(\pi)\sum_{i\in D}(\pi_i-\pi_0)r_i+\sum_{(i,j)\in A}min_{f_{ij}\geq 0}\int_0^{f_{ij}}\left(t_{ij}(s)+a_{ij}^k+\pi_i-\pi_j\right)ds$$

4- we write function gradient:

$$\nabla\theta(\pi)_i\sum_{f\in w_i}(\pi_i-\pi_j)f_{ij}-\sum_{j\in v_i}\left(\pi_i-\pi_j\right)f_{ij}-r\ r=\begin{cases}\sum_{i\in D}r_i\ if\ i=0\\ -r_i\ if\ i\in D\\ 0\ otherwise\end{cases}$$

5-then we run "flow balancer" algorithm to achieve the value of π *. (Π * is the optimal answer of Dugan problem).

With the obtained π *, we try to minimize L-function (f, π *) and achieve f *. This f * is the answer of PAR issue. Finally we will have optimal F for each pathway.

Flow balancer algorithm:

Primary assumptions:

π0 is a vector of N members. ε>0, L=0, a=2>0, h>0

1- choose the i point so that $\left|\nabla\theta(\pi^1)_{i_1}\right|>\varepsilon$.

2- put $\hat{\delta}_i=\pi_i, \bar{\delta}i=\pi_i$

3-If θ (π) i > 0 and if the $\underset{i}{\theta'}(\hat{\delta}_i+h)$ and $\underset{i}{\theta'}(\bar{\delta})$ have different marks.

4- = πi+1> $1/2\,(\hat{\delta}:=\bar{\delta})$.

5-If, $\left|\nabla\theta(\pi^{1+1})_{i_1}\right|\leq\varepsilon$, then $i\in N$ is the answer of algorithm, otherwise, go to the first step.

Patrilson implemented his proposed method in 1992 on a small network with 13 nodes, 19 arcs and 4 origin-destination nodes. The results of the implementation of Patrikson methods shows the answers obtained from the 40 repeating in Frank-Wolfe obtained only in 6th repetition of Patrikson methods.

As well, the results of the implementation of the algorithm on a relatively large network with 64 nodes, 112 arcs and 8 origin-destination nodes shows the answers obtained after 50 repetition in Frank-Wolfe method will be achieved in repetition of 10 in Patrikson method.

Conclusion:

The efficiency of various methods of problem solving will be determined based on the minimum time necessary for the assignment problem solving and the amount of similarity of obtained answers to optimal one.

All the introduced procedures have been tested on the hypothetical or real networks, and numerical results obtained from these experiments represent the amount of the performance of each of these methods.

Because the procedures presented in this field have been provided in several years, they did not use the same computational technology to implement their methods.

This issue shows that the results obtained from these studies are unable to be used, even if all of the methods use identical networks. In General, to use a problem-solving method of allocating several aspects should be taken into consideration to achieve an optimum result.

The characteristic of network that is assigned to the issue to be resolved is very important.

The number of nodes and arcs of network and more importantly the number of origin-destination nodes could determine which method is best to be used for such a network, the accuracy of resulting answer is also very important in determining the time of running the program because allocation issue may be resolved several times. Obviously an answer with high-precision is not applicable and a method with highest speed is better.

REFERENCES

[1] Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programing Methods, prentice-hall, newjersey, U.S.A.

[2] Mokhtar, S. Bazaraa, C.M. Shetty, 2006. Nonlinear programming: Theory and Algorithms, third edition, John Wiley & sons, New Jersey, U.S.A, pp: 853.
[3] Linear and nonlinear programming, 1378, d. J. Luenberger; translated by. d. n. Mahdavi Amiri, m. h. Poor Kazemi, the Institute of scientific publications of the Sharif University, Tehran, pp: 656.
[4] Mathematical Programing Study, 1981. R.S. Dembo, J. G. Klincevicz, A scaled resuced gradient algorithm for network flow priblems with convex seperable costs, 15: 125-147.
[5] Transportation Science, 1974. S. Nguyen, An algorithm foe the traffic assignment problem, 8(0): 203-216
[6] Ebrahimi, M., 1389. "investigating different methods of problem-solving of allocation", master's seminar, Tehran, University of science & technology; Faculty of civil engineering.
[7] Beckmann, M., C.B. McGuire, C.B. Winsten, 1956. Suydiesih the Economics of Transportation, Yale University Press New Haven.
[8] Naval Research Logistics Quarterly, 1956. M. Frank, P. Wolfe, An algorithm for quadratic programming, 3(2): 95-110.
[9] Transportation Research Part B: Methodological, 1984, M Fukushima, A modified Frank-Wolfe algorithm for solving the traffic assignment problem, 2: 169-177.
[10] Transportation Research Part B: Methodological, 1985. A. Weintraub, C. Ortiz, J. Gonzalez, Accelerating convergence of the Frank-Wolfe algorithm, 19(2): 113-122.
[11] IJCSNS International Journal of Computer Science and Network Security,2008, S. Arrache, R. Ouafi, Accelerating Convergence of the Frank-Wolfe Algorithm for Sovling the Traffic Assignmant Problem, 8(5).
[12] Transportation Science, 1985. L.J LeBlan, R.V. Helgason, and D.E. Boyce, Improved Efficiency of the Frank-Wolfe Algorithm for Convex Network Programs, 19(4): 445-462.
[13] Applied Mathematics and Computation, 2004. Z.Gao, W.H.K. Lam, S.C. Wong, H. Yang, The convergence of equilibrium algorithm with non-monotone line search technique, 184(1): 1-13.
[14] Journal of Optimization Theory and Applications, 1993. M. Patriksson, Partial linearization methods in nonlinear programming, 78(2): 227-246.
[15] A Journal of Mathematical Programming and Operations Research, 1993. T. Larsson, A. Migdalas, M. Patriksson, J.G. Wardrop, A partial linearization method for the traffic assignment problem. Optimization, 28(1): 47-61.
[16] Transportation Research Part B: Methodological, 1985. M. Carey, The dual of the traffic assignment problem with elastic demands, 19(3): 227-237.
[17] Transportation Research Part B:Methodological, 1984, M. Fukushima, On the dual approach to the traffic assignment problem, 18(3): 235-245.