New Trends in Mathematical Sciences
http://dx.doi.org/10.20852/ntmsci.2023.515

Iterative Approach for Traffic Assignment Problem with Capacity Limits

Hasan Dalman

Batman University, Department of Mathematics, Batman, Turkey

Received: 2 September 2023, Accepted: 2 November 2023
Published online: 20 November 2023.

Abstract: This study presents a method for reaching user equilibrium in network traffic assignment problems with capacity constraints. The presented approach minimizes each user's own travel time using a gradient-based algorithm based on the Taylor series. The algorithm is shown to converge efficiently to user equilibrium after a limited number of iterations. A numerical example is provided to demonstrate the effectiveness of the presented approach, and comparisons are made with other algorithms available in the literature. The obtained results show that the presented method is capable of reaching user equilibrium for capacity network traffic assignment problems.

Keywords: Network Traffic Assigment, Nonlinear Optimization, Algorithms

1 Introduction

The Traffic Assignment Problem (TAP) plays a pivotal role in the field of transportation planning, aiming to accurately predict drivers' route preferences and resulting traffic flow patterns. This problem is closely linked to Wardrop's User Equilibrium (UE) condition, originally introduced by Wardrop in 1952 [1]. Wardrop's first principle, as it is commonly known, explains the behavior of travelers within a transportation network. According to this principle, individuals with the same origin and destination (OD) must experience identical travel durations and cannot independently minimize their travel times by choosing alternative routes [2]. Beckmann and colleagues made significant progress by transforming the UE condition into a nonlinear convex programming problem. This transformation paved the way for efficient algorithms designed to address the UE condition ([13], [4], [12], [26], [27], [9], [16], [31]). Researchers have approached the Traffic Assignment Problem (TAP) from various angles, employing different solution methods categorized broadly as link-based, path-based, and bush-based approaches. Link-based methodologies, exemplified by well-known algorithms like the Frank-Wolfe algorithm [8], the LeBlanc-Morlok algorithm [9], the Mitradjieva-Lindberg algorithm [10], and the Chen-Liu-Kim algorithm [14], prioritize optimizing link flows. One notable advantage of these approaches is their lower operational memory requirements when solving the optimization problem. On the other hand, path-based methods, such as the Florian-Constantin algorithm [11] and the Jayakrishnan-Tsai algorithm [12], work with path flows and tend to converge faster compared to link-based methods. However, they typically require more memory. In contrast, bush-based approaches, illustrated by algorithms like the Bar-Gera algorithm [13], the Dial algorithm [5], the Nie algorithm [15], and the Gentile algorithm [7], leverage the acyclic network structure created by the set of utilized routes originating from each source point in an equilibrium scenario. These methodologies address the problem by solving a sequence of node-centric subproblems defined within the "bush" structure emanating from each source point. They have made significant strides in algorithmic development, offering notable gains in computational speed and memory efficiency, especially for large networks. However, applying these algorithms to address equilibrium concerns in exceedingly

* Corresponding author e-mail: hasan.dalman@batman.edu.tr

large-scale networks may still present certain challenges. Javani and Babazadeh introduced an algorithm that identifies descent directions by iteratively solving a series of quadratic programming subproblems within the framework of truncated quadratic programming. This method, formally referred to as OD-based Frank-Wolfe truncated quadratic programming, systematically addresses each quadratic programming subproblem related to Origin-Destination (OD) pairs. These subproblems are efficiently resolved using the Frank-Wolfe optimization technique, considering only the active paths [39]. Recently, Galligari and Sciandrone introduced a path equilibration algorithm. In each iteration of this algorithm, the flows of only two paths associated with each Origin-Destination (OD) pair are adjusted. This is achieved through an inexact line search method combined with an adaptive column generation technique, as detailed in their work referenced as Galligari and Sciandrone [40]. For static TAP, Babazadeh et al. devised a path-based algorithm that decomposes the problem into origin-destination (OD) pairs and utilizes the Wolfe reduced gradient method to solve each subproblem. To model the impact of traffic congestion on arc connections, various travel cost functions have been proposed. Polynomial functions are commonly used in practice. These functions estimate finite travel times for all arc flows, assuming that roads can handle unlimited traffic by default. However, in reality, arc connections have specific capacity limits that restrict traffic flows ([21], [22]). To improve the accuracy of assignment models and address the constraints associated with assuming infinite traffic capacity, it would be advantageous to incorporate maximum limits on arc flows. This can be achieved explicitly by introducing arc capacities or implicitly by utilizing asymptotic travel time functions. The latter concept suggests that as the flow on an arc approaches its upper bound, the travel time on that arc tends towards infinity ([23], [24]). Arc flow capacity constraints commonly arise due to traffic control measures or congestion. Factors such as speed limits and traffic signal cycle times establish predefined capacities for arc flows, which are widely recognized and must not be exceeded by drivers violating traffic regulations [25]. Capacity constraints on arc flows can also vary depending on current traffic conditions. During peak hours, arc flows may become imbalanced and exceed capacity, while under normal circumstances, they usually remain below capacity. Therefore, a traffic model should consider different capacity levels for different traffic situations, such as specific time intervals. It is worth noting that approximate weak capacities may slightly exceed the capacity limits in certain cases [28].

The consideration of capacity constraints is of paramount importance as it enables the control of arc flows to approach their maximum limits, while asymptotic travel time functions generally remain below these thresholds. Nevertheless, there have been instances where asymptotic travel time functions have led to excessively high travel times, resulting in misleading route redirections. Conversely, the explicit integration of link capacities poses the challenge of compromising the structured nature of capacity-unconstrained problems, as discussed by Boyce, Janson, and Eash (1981) [29]. This presents computational complexities, especially when employing techniques such as the Frank-Wolfe method and simplicial decomposition, as outlined by Klessig (1974) [30]. Under rigorous assumptions regarding the characteristics of travel time functions and initial conditions, the subproblem of multicommodity flow within the Frank-Wolfe method can be transformed into a series of shortest-route subproblems, ensuring convergence toward an optimal flow pattern, as described by Daganzo (1977) [23] and Hearn and Ribera (1981) [17]. While a capacitated user equilibrium assignment problem may deviate from strict adherence to Wardrop's first principle, it still upholds a modified version of the principle, wherein generalized travel costs with well-defined attributes replace traditional travel costs. From a computational perspective, the utilization of asymptotic travel time functions entails certain drawbacks due to potential numerical challenges. Moreover, when employing feasible-direction algorithms like the Frank-Wolfe method, meticulous initialization of the algorithm is imperative to compute a flow pattern that strictly adheres to the implicit upper bounds on link flows, as highlighted by Daganzo (1977) [24]. Hence, we propose an iterative algorithm to address the Traffic Assignment Problem (TAP) with capacity constraints. There exists a substantial body of literature on the application of Taylor series in optimization problems (refer to Bertsekas et al. [33], Nocedal and Wright [34], and Bazaraa et al. [32]). The algorithm's objective is to determine an optimal flow pattern that minimizes travel times while respecting the capacity limits of the arcs within the transportation network. Furthermore, the user equilibrium conditions

are mathematically formulated as a linearly constrained convex program. The algorithm takes into consideration the arc performance functions, which quantify the cost of utilizing an arc based on its flow and congestion effects. By incorporating capacity constraints, the algorithm aims to strike a balance between travel efficiency and network capacity utilization. Finally, this paper examines a numerical example previously utilized by Barton and Hearn (1979) [18] and compares the results obtained through Moiseev's direct search method implemented in Maple (2011) [37] and the Powell method (1964) [36]. This comparison showcases the superior performance of the proposed algorithms in terms of convergence rate and solution quality. The paper is structured as follows:
Section 2 presents the model formulation of the optimization problem for the network TAP with capacities. Section 3 provides the proposed algorithm for obtaining the UE of TAP with capacities. Section 4 applies the proposed algorithm to a numerical example. Section 5 presents the conclusions and findings of the study.

2 Capacitated User Equilibrium Traffic Assignment Problem

A traffic network can be formally represented as a directed graph denoted by $G(\mathscr{N},\mathscr{A})$, where $\mathscr{N}$ represents the set of nodes, and $\mathscr{A}$ represents the set of connections. In this context, the nodes correspond to sources denoted as S, and destinations denoted as D.

In transportation networks, each arc $a \in \mathscr{A}$ is associated with an arc performance function $t_a : \mathbb{R}_+ \to \mathbb{R}_{++}$. This function measures the cost of utilizing the arc based on its flow, taking into account congestion effects. The primary objective is to identify a traffic flow pattern that can meet travel demands while simultaneously achieving user equilibrium.

The principle of user equilibrium route choice, first introduced by Wardrop [1], stipulates that the travel times for all utilized routes are equal and lower than the travel time for any unused route taken by a single vehicle.

Let c^{rs} represent the travel time (or travel cost) on route k for the origin-destination pair $(r,s) \in \mathscr{C}$ under a feasible flow pattern. Assuming that k routes are used and carry positive flows, a user equilibrium flow pattern is attained if the following condition is satisfied:

$$c_1^{rs} = c_2^{rs} = \ldots = c_k^{rs},$$

and the travel times for unused routes within each origin-destination pair are equal to or greater than those of utilized routes.

Addressing the challenge of achieving user equilibrium traffic assignment while accounting for arc capacity constraints gives rise to a captivating problem known as the Capacitated User Equilibrium Traffic Assignment Problem (capacitated UE TAP). This intricate mathematical puzzle was initially introduced by Dafermos and Sparrow in 1969 [38].

At its core, the problem aims to minimize the total travel cost or travel time, denoted by $z(f)$, which emerges as the integral of the arc cost or travel time function $t_a(\tau)$ over the flow quantity f_a on each arc a. The optimization can be mathematically expressed as follows:

$$\min \quad z(f) = \sum_{a\in\mathscr{A}} \int_0^{f_a} t_a(\tau)\, d\tau \tag{1}$$

subject to:

$$\sum_{k\in\mathscr{R}_{rs}} h_k^{rs} = d_{rs}, \quad \forall (r,s)\in\mathscr{D} \tag{2}$$

$$h_k^{rs} \geq 0, \quad \forall k\in\mathscr{R}_{rs}, \forall (r,s)\in\mathscr{D} \tag{3}$$

$$\sum_{(r,s)\in\mathscr{C}} \sum_{k\in\mathscr{R}_{rs}} \delta_{a,k}^{rs} h_k^{rs} = f_a, \quad \forall a\in\mathscr{A} \tag{4}$$

$$f_a \leq u_a, \quad \forall a\in\mathscr{A} \tag{5}$$

where:

–$z(f)$ is the objective function representing the total travel time to be minimized by integrating travel times over all routes.
–f_a denotes the flow quantity on each arc.
–$t_a(\cdot)$ represents the travel time function on link a at a certain flow level.
–h_k^{rs} indicates the flow quantity on each link for a specific route and origin-destination pair.
–d_{rs} represents the demand quantity for a given origin-destination pair.
–$\delta_{a,k}^{rs}$ is an indicator variable that takes the value of 1 if arc a belongs to route k between nodes r and s, and 0 otherwise.
–$\mathscr{R}_{rs}$ represents the set of routes between nodes r and s.
–$\mathscr{C}$ is the set of all arcs.
–u_a is the capacity constraint on arc a.

3 KKT Conditions for Network Traffic Assignment Problem

The first-order optimality conditions for this optimization problem are given by the Karush-Kuhn-Tucker (KKT) conditions:

$$\nabla z(f) - \sum_{(r,s)\in\mathscr{D}} \lambda_{rs}\nabla\left(\sum_{k\in\mathscr{R}_{rs}} h_k^{rs} - d_{rs}\right) - \sum_{a\in\mathscr{A}} \eta_a \nabla\left(f_a - u_a\right) = 0 \tag{6}$$

$$\eta_a(f_a - u_a) = 0, \quad \forall a\in\mathscr{A} \tag{7}$$

$$\lambda_{rs}\left(\sum_{k\in\mathscr{R}_{rs}} h_k^{rs} - d_{rs}\right) = 0, \quad \forall (r,s)\in\mathscr{D} \tag{8}$$

$$\eta_a \geq 0, \quad \lambda_{rs} \geq 0, \tag{9}$$

where ∇ denotes the gradient operator. λ_{rs} and η_a are the Lagrange multipliers associated with the constraints, and f_a is the predefined arc flows.
To derive the second-order optimality conditions, we analyze the Hessian matrix of the Lagrangian function, defined as follows:

$$H(f,\lambda,\eta) = \nabla^2 z(f) - \sum_{(r,s)\in\mathscr{D}} \lambda_{rs}\nabla^2\left(\sum_{k\in\mathscr{R}_{rs}} h_k^{rs} - d_{rs}\right) - \sum_{a\in\mathscr{A}} \eta_a \nabla^2\left(f_a - u_a\right) \tag{10}$$

After calculating the Hessian matrix, we evaluate its definiteness at the optimal solution to determine the second-order optimality conditions. The conditions are as follows:

1. If **H** is positive definite, then the solution is a strict local minimum.
2. If **H** is positive semidefinite, then the solution is a local minimum.
3. If **H** is negative definite, then the solution is a strict local maximum.
4. If **H** is negative semidefinite, then the solution is a local maximum.

4 Iterative Approach for TAP with Capacities

In this section, we present an iterative solution approach for the TAP with Capacities. To facilitate understanding, we will use mathematical symbols commonly used in optimization.

Table 1: List of Notations and Definitions

$z(f)$	Objective function
$f^{(k)}$	Value of the arc flow variable f at iteration k
$z(f^*)$	User optimal value of the objective function
$\Delta f^{(k)}$	Change in the arc flow variable f between consecutive iterations
$H(f^*)$	Hessian matrix of the objective function evaluated at the user optimal solution point f^*
$(\Delta f^{(k)})^\top$	Transpose of the vector $\Delta f^{(k)}$
$(f^{(k)} - f^*)^\top$	Transpose of the vector $f^{(k)} - f^*$
"Positive definite"	Property of the Hessian matrix $H(f^*)$ implying it is nonsingular and all of its eigenvalues are positive
"Positive semidefinite"	Property of the Hessian matrix $H(f^*)$ implying it is nonnegative and all of its eigenvalues are nonnegative

The second-order Taylor series approximation of the objective function $z(f)$ around the optimal point f^* is given by the following equation:

$$z(f) \approx z(f^*) + \nabla z(f^*) \cdot (f - f^*) + \frac{1}{2}(f - f^*)^\intercal H(f^*) \cdot (f - f^*) \tag{11}$$

This expression provides us with an estimation of the objective function value $z(f)$ near the optimal point f^* in an indirect manner, utilizing information from the gradient vector $\nabla z(f^*)$ and the Hessian matrix $H(f^*)$.
To demonstrate the convergence of the proposed algorithm to the user equilibrium of TAP with capacities, we can examine the following:
For the arc flow f^*, we have $\nabla z(f^*) = 0$, which simplifies the approximation as:

$$z(f) \approx z(f^*) + \frac{1}{2}(f - f^*)^\intercal H(f^*) \cdot (f - f^*) \tag{12}$$

Next, let's bound the expression $z(f) - z(f^*)$:

$$|z(f) - z(f^*)| \leq \frac{1}{2}\|f - f^*\|^2 \|H(f^*)\| \tag{13}$$

where $\|\cdot\|$ represents the norm of a vector.

To show the convergence, we need to demonstrate that the expression above approaches a bounded value. For this, we need to control the value of $\|f - f^*\|$.

If f approaches f^*, $\|f - f^*\|$ will tend to zero, which implies that $z(f) - z(f^*)$ will approach a bounded value. This indicates the convergence of the proposed procedure.

Therefore, as the points f approach the point f^*, the function $z(f)$ will also converge to the value $z(f^*)$.

Theorem 1.*The objective function $z(f)$ converges to $z(f^*)$ as the number of iterations tends to infinity, provided that the following conditions hold:*

1.The descent property holds, i.e., the objective function decreases or remains the same with each iteration.
2.The Hessian matrix $H(f^*)$ is positive definite (or positive semidefinite).

The descent property requires that the objective function decreases (or remains the same) with each iteration. Let's consider the change in the objective function after one iteration:

$$\begin{aligned} z(f^{(k+1)}) - z(f^{(k)}) = &\frac{1}{2}(\Delta f^{(k)})^\top \cdot H(f^*) \cdot (\Delta f^{(k)}) \\ &- \frac{1}{2}(f^{(k)} - f^*)^\top \cdot H(f^*) \cdot (f^{(k)} - f^*) \end{aligned}$$

Expanding the terms and rearranging, we obtain:

$$\begin{aligned} z(f^{(k+1)}) - z(f^{(k)}) = &-\frac{1}{2}(f^{(k)} - f^*)^\top \cdot H(f^*) \cdot (f^{(k)} - f^*) \\ &+ (\Delta f^{(k)})^\top \cdot H(f^*) \cdot (\Delta f^{(k)}) \end{aligned}$$

Since $H(f^*)$ is positive definite (or positive semidefinite), we can write:

$$(f^{(k)} - f^*)^\top \cdot H(f^*) \cdot (f^{(k)} - f^*) \geq 0$$

Thus, we have:

$$z(f^{(k+1)}) - z(f^{(k)}) \leq (\Delta f^{(k)})^\top \cdot H(f^*) \cdot (\Delta f^{(k)})$$

Since $(\Delta f^{(k)})^\top \cdot H(f^*) \cdot (\Delta f^{(k)})$ is nonnegative, it follows that $z(f^{(k+1)}) - z(f^{(k)}) \leq 0$. Hence, the objective function either decreases or remains the same with each iteration, satisfying the descent property.

*Remark.*The positive definiteness (or positive semidefiniteness) of the Hessian matrix $H(f^*)$ depends on the specific properties and characteristics of the objective function $z(f)$, as well as the problem itself. In general, the Hessian matrix is positive definite if the objective function is convex and smooth.

Without further information about the objective function $z(f)$ and its specific properties, it is challenging to provide a general proof of positive definiteness (or positive semidefiniteness) for all cases.

However, in practice, the positive definiteness (or positive semidefiniteness) of the Hessian matrix can be ensured by considering appropriate conditions and constraints in the problem formulation. These conditions may include convexity properties, smoothness of the objective function, and convexity of the feasible region.

Overall, if the descent property holds (objective function decreases or remains the same with each iteration), and the Hessian matrix is positive definite (or positive semidefinite), the given algorithm will converge to the user optimal solution.

Using the second-order Taylor series approximation (4), we reduce the original problem (1)-(2) to a subproblem that approximates the user equilibrium. The subproblem is defined as follows:

$$\min \quad z(f^*) + \nabla z(f^*) \cdot (f - f^*) + \frac{1}{2}(f - f^*)^\top \nabla^2 z(f^*)(f - f^*) \tag{14}$$

where:

–$z(f^*)$ represents the objective function value at the arc flow point f^*.
–$\nabla z(f^*)$ denotes the gradient vector evaluated at f^*.
–$\nabla^2 z(f^*)$ represents the Hessian matrix evaluated at f^*.

The constraints remain the same as in equation (2), specifying the additional conditions to be satisfied. The subproblem aims to approximate f^* of the original problem.
The algorithm is illustrated with Algorithm 1.

Algorithm 1 Iterative Algorithm

1: Initialize the current point as $f = f^*$
2: Compute $z(f^*)$, gradient vector $\nabla z(f^*)$, and Hessian matrix $\nabla^2 z(f^*)$
3: Calculate $z(f)$ for the current point
4: Build and solve the subproblem subject to the constraints
5: Obtain a new solution f
6: Compute $|z(f) - z(f^*)|$
7: **if** $|z(f) - z(f^*)| < \varepsilon$ **then**
8: Terminate the iteration as convergence is achieved
9: **else**
10: Update f^* with the current f and go back to step 2
11: **end if**

*Remark.*It is important to note that the initial point selected can be arbitrary and may not satisfy the constraints of the TAP with capacities. Additionally, it is important to note that f is dependent on chain (route) flows, and when f is substituted into $z(f)$, the problem becomes entirely dependent on chain (route) flows h_k^{rs}.

$$f = f(h_k^{rs}) z(f) = z(f(h_k^{rs}))$$

As a result, the problem can be solved by iteratively adjusting the chain (route) flows h_k^{rs} until the constraints of the problem are satisfied.

5 A Numerical Example

Consider a directed network ([18]) with nodes N and links A, where $N = \{1,2,3,4,5,6,7,8,9\}$ and A represents a set of directed links as shown in Table 2.
Let C be the set of center nodes, where $C = \{1,2,3,4\}$. Let T be the set of transfer point nodes, where $T = \{5,6,7,8,9\}$. We can determine that the number of center nodes and transfer point nodes is $\|C\| = 4$ and $\|T\| = 5$, respectively.

The network A consists of several arcs, and each arc is assigned a specific capacity value. We define set u_a to represent practical capacities corresponding to arcs in A, as presented in Table 2.

Table 2: Details of Directed Network

Arc Set A	Capacity Set u_a	Travel Time T_a
$(1,5)$	10	5
$(1,6)$	16	6
$(2,5)$	35	3
$(2,6)$	18	9
$(5,6)$	20	9
$(5,7)$	11	2
$(5,9)$	26	8
$(6,5)$	11	4
$(6,8)$	33	6
$(6,9)$	32	7
$(7,3)$	25	3
$(7,4)$	24	6
$(7,8)$	19	9
$(8,3)$	39	8
$(8,4)$	43	6
$(8,7)$	36	4
$(9,7)$	26	4
$(9,8)$	30	8

The cost of reaching the destination nodes from the source nodes is given as in Table 3.

Table 3: Cost of Reaching Destination Nodes from Source Nodes

Source Node	Destination Node	Cost
1	3	10
1	4	20
2	3	30
2	4	40

The set of start nodes S and end nodes D can be identified. In this case, start nodes are represented by $S = \{1,2\}$, and end nodes are represented by $D = \{3,4\}$ in the network.
The given network is depicted in Figure 1.
In this example, the test problem within the network employs the travel time formula specified by the Bureau of Public Roads. The formula used to compute the travel time on arc a, denoted as $t_a(f_a)$, is given by:

$$t_a(f_a) = T_a \left(1 + \theta \left(\frac{f_a}{u_a}\right)^{\beta}\right)$$

Here, T_a represents the free-flow travel time on arc a, as shown in Table 2. θ is a fixed parameter set to 0.15, and β is a constant exponent greater than 1. In these calculations, we selected $\beta = 4$. The practical capacity of arc a is denoted as u_a.

To ensure flexibility and scalability, arc capacities u_a are uniformly scaled by a factor of θ, denoted as $\theta \cdot u_a$. It is important to note that θ is a positive constant applied uniformly across all arcs in set A, facilitating consistent scaling of practical capacities.

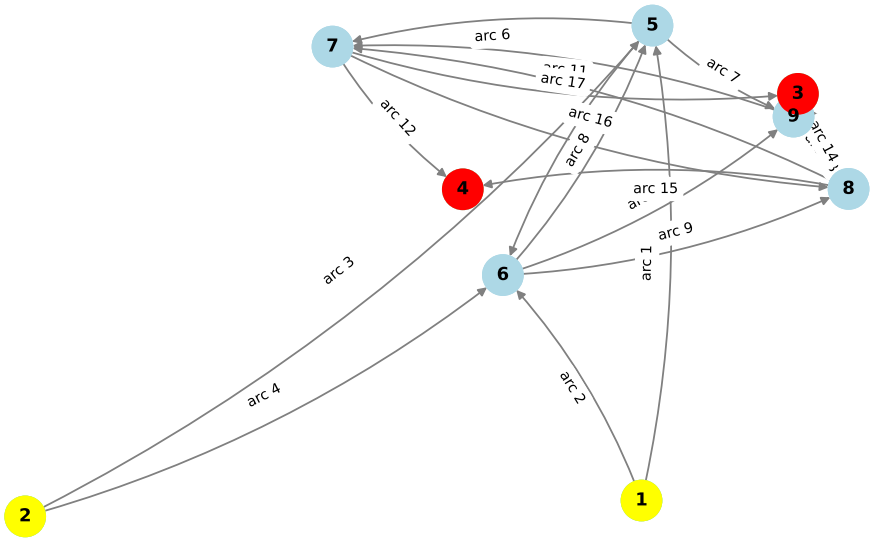

Fig. 1: Transportation network with capacities

For each specific problem instance, the capacity scaling factor θ is carefully chosen to be as small as possible, while still preserving feasibility. This deliberate selection of a small θ value, such as $\theta = 1.5$, contributes to the creation of challenging instances for the TAP with Capacities, thereby generating more intricate and demanding scenarios.

The transportation network under investigation consists of 96 chain flows or routes, which are subject to 4 demand constraints and 18 arc flow constraints. These constraints and chain flows have been determined using the Python 3 programming language within the Jupyter notebook.

Subsequently, the proposed algorithm was implemented based on these constraints and executed using the Maple 2023 software. The obtained results for each iteration of the algorithm are presented in Table 4. Furthermore, a visual representation of the obtained results is displayed in Figure 2 and Figure 3.

Table 4 shows results obtained from the iterative optimization process. It provides valuable insights into the progress of the presented algorithm and the corresponding values at each iteration.

The ”Iteration” column enumerates the iterations performed during the optimization process. ”The subproblem (z value)” column represents values obtained from optimizing the subproblem (7)-(8) within each iteration. The “Primal Problem (z value)” column displays the values obtained from optimizing the main problem (1)-(2) itself. These values represent the overall objective or performance measure of the optimization problem.

Table 4: The obtained results for each iteration

Iteration	Subproblem (z value)	Primal Problem (z value)	Tolerance Value
1	1139.888345599630	1132.758369384533	-
2	1130.266246175331	1129.960421053934	2.80×10^{-3}
3	1129.742574488579	1129.703495875214	2.57×10^{-1}
4	1129.665822397805	1129.658097868443	4.54×10^{-2}
5	1129.650511108232	1129.648939306953	9.16×10^{-3}
6	1129.647365299124	1129.647030158681	1.91×10^{-3}
7	1129.646675140486	1129.646593810956	4.36×10^{-4}
8	1129.646516108661	1129.646507267934	8.65×10^{-5}
9	1129.646500084046	1129.646498622683	8.65×10^{-6}
10	1129.646497205562	1129.646496916876	1.71×10^{-6}
11	1129.646496636925	1129.646496579896	3.37×10^{-7}

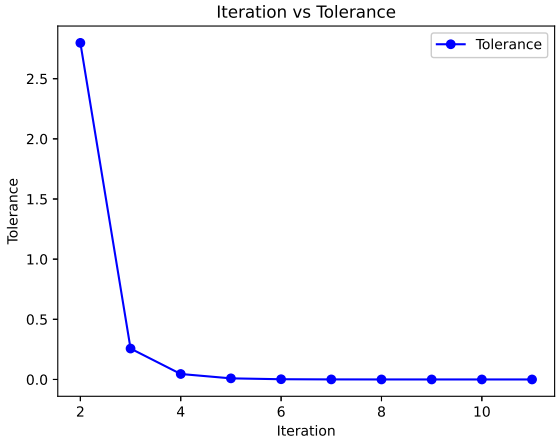

Fig. 2: Change in tolerance at each iteration

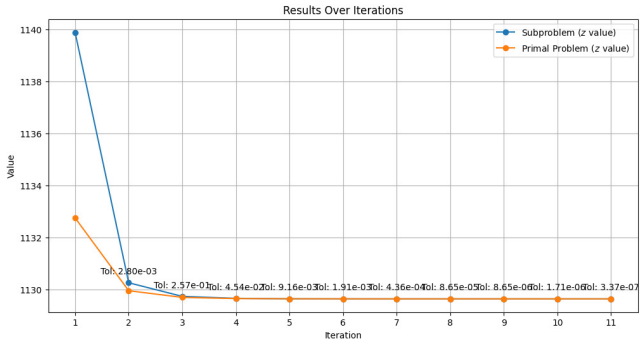

Fig. 3: Value of objective functions at each iteration

The ”Tolerance Values” column quantifies the convergence of the optimization algorithm by measuring the discrepancy between consecutive iterations. It indicates the level of acceptable deviation or error tolerated between iterations. A smaller tolerance value suggests a closer approximation to the solution.

Fig. 2 depicts the change of tolerance values at each iteration. It provides a graphical representation of the analysis of the proposed algorithm.

The table is arranged in ascending order of iterations, allowing for a clear understanding of the optimization progress. Each row presents the values achieved at a specific iteration, including the subproblem value, primal problem value, and tolerance value.

In addition, the obtained results of the proposed algorithm were compared with a derivative-free optimization [37] and Powell method [36], employing identical initial conditions and a tolerance value of 10^{-6}. This comparison, conducted using Maple 2023 software, revealed that the proposed method outperformed alternative methods in terms of iteration count and objective function values. As shown in Table 5, the proposed algorithm demonstrated superior convergence, achieving better results and requiring fewer iterations to reach the solution compared to Moiseev's and Powell's methods.

Table 5: Comparison of Methods

Method	Objective Function Value (z)	Iteration Count
Moiseev's Method [37]	1138.334035437554	7652
Powell's Method [36]	1133.133.008693671905	10000
Proposed Method	1129.646496579896	11

6 Conclusion

In this paper, an iterative solution method for the capacitated TAP is introduced, which is a nonlinear programming challenge aimed at minimizing total travel cost while adhering to arc capacity constraints. The approach involves using the second-order Taylor series approximation of the objective function centered around the user optimal solution. It is demonstrated that, as the number of iterations approaches infinity, the objective function converges to its optimal value, given that the descent property is upheld and the Hessian matrix is positive definite (or positive semidefinite).

The primary contributions of this study are as follows:

An iterative solution method for the capacitated TAP is proposed. The convergence of the proposed approach is proven, subject to the satisfaction of the descent property and positive definiteness (or positive semidefiniteness) of the Hessian matrix. A detailed analysis of the convergence properties of the approach is conducted. It is believed that the proposed method shows promise as an effective and straightforward alternative for solving the capacitated TAP. The simplicity of implementation, along with its proven convergence under reasonable conditions, makes it a valuable tool for transportation planners and researchers involved in addressing capacitated TAPs.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors have contributed to all parts of the article. All authors read and approved the final manuscript.

References

[1] J. G. Wardrop, *Some theoretical aspects of road traffic research*, *Proc. Institute of Civil Engineers, Part II*, pp. 325-378, 1952.

[2] M. J. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*, New Haven: Yale University Press, 1956.

[3] M. Patriksson, *Sensitivity Analysis of Traffic Equilibria*, *Transportation Science*, vol. 38, no. 3, pp. 258-281, 2004.

[4] A. Chen, X. Xu, S. Ryu, and Z. Zhou, *A self-adaptive Armijo stepsize strategy with application to traffic assignment models and algorithms*, *Transportmetrica A: Transport Science*, vol. 9, no. 8, pp. 695-712, 2013.

[5] R. B. Dial, *A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration*, *Transportation Research Part B: Methodological*, vol. 40, no. 10, pp. 917-936, 2006.

[6] H. Bar-Gera, *Traffic assignment by paired alternative segments*, *Transportation Research Part B: Methodological*, vol. 44, no. 8, pp. 1022-1046, 2010.

[7] G. Gentile, *Local user cost equilibrium: a bush-based algorithm for traffic assignment*, *Transportmetrica A: Transport Science*, vol. 10, no. 1, pp. 15-54, 2014.

[8] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95-110, 1956.

[9] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla, *An efficient approach to solving the road network equilibrium traffic assignment problem*, *Transportation Research*, vol. 9, no. 5, pp. 309-318, 1975.

[10] M. Mitradjieva and P. O. Lindberg, *The stiff is moving-conjugate direction frank-wolfe methods with applications to traffic assignment*, *Transportation Science*, vol. 47, no. 2, pp. 280-293, 2013.

[11] M. Florian, I. Constantin, and D. Florian, *A new look at projected gradient method for equilibrium assignment*, *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2090, pp. 10-16, 2009.

[12] R. Jayakrishnan, W. Tsai, J. Prasker, and S. Rajadhyakfchsha, *A faster path-based algorithm for traffic assignment*, *Transportation Research Record*, vol. 1443, pp. 75-83, 1994.

[13] H. Bar-Gera, *Origin-based algorithm for the traffic assignment problem*, *Transportation Science*, vol. 36, no. 4, pp. 398-417, 2002.

[14] X. Chen, Z. Liu, and I. Kim, *A parallel computing framework for solving user equilibrium problem on computer clusters*, *Transportmetrica A: Transport Science*, vol. 16, no. 3, pp. 550-573, 2020.

[15] Y. Nie, *A class of bush-based algorithms for the traffic assignment problem*, *Transportation Research Part B: Methodological*, vol. 44, no. 1, pp. 73-89, 2010.

[16] Y. Nie, *A note on Bar-Gera's algorithm for the origin-based traffic assignment Problem*, *Transportation Science*, vol. 46, no. 1, pp. 27-38, 2012.

[17] D. W. Hearn and J. Ribera, *Convergence of the Frank-Wolfe method for certain bounded variable traffic assignment problems*, *Transportation Research Part B: Methodological*, vol. 15, no. 3, pp. 223-232, 1981.

[18] Barton, R. R., and Hearn, D. W., *Network aggregation in transportation planning models* (No. DOT-TSC-RSPA-79-18). United States. Dept. of Transportation. Research and Special Programs Administration, 1979

[19] D. W. Hearn, *Practical and theoretical aspects of aggregation problems in transportation planning*, *Transportation Planning Models*, pp. 257-287, 1984.

[20] R. R. Barton, D. W. Hearn, and S. Lawphongpanich, *The equivalence of transfer and generalized benders decomposition methods for traffic assignment*, *Transportation Research Part B: Methodological*, vol. 23, no. 1, pp. 61-73, 1989.

[21] D. Branston, *Link capacity functions: A review*, *Transportation Research*, vol. 10, no. 4, pp. 223-236, 1976.

[22] D. W. Hearn, *Bounding flows in traffic assignment models*, Research Report, 80-4, 1980.

[23] C. F. Daganzo, *On the traffic assignment problem with flow dependent costs?I*, *Transportation Research*, vol. 11, no. 6, pp. 433-437, 1977.

[24] C. F. Daganzo, *On the traffic assignment problem with flow dependent costs?II*, *Transportation Research*, vol. 11, no. 6, pp. 439-441, 1977.

[25] H. Yang and S. Yagar, *Traffic assignment and traffic control in general freeway-arterial corridor systems*, *Transportation Research Part B: Methodological*, vol. 28, no. 6, pp. 463-486, 1994.

[26] T. Larsson and M. Patriksson, *Simplicial decomposition with disaggregated representation for the traffic assignment problem*, *Transportation Science*, vol. 26, no. 1, pp. 4-17, 1992.

[27] T. Larsson, A. Migdalas, and M. Patriksson, *A partial linearization method for the traffic assignment problem*, *Optimization*, vol. 28, no. 1, pp. 47-61, 1993.

[28] T. Larsson and M. Patriksson, *An augmented Lagrangean dual algorithm for link capacity side constrained traffic assignment problems*, *Transportation Research Part B: Methodological*, vol. 29, no. 6, pp. 433-455, 1995.

[29] D. E. Boyce, B. N. Janson, and R. W. Eash, *The effect on equilibrium trip assignment of different link congestion functions*, *Transportation Research Part A: General*, vol. 15, no. 3, pp. 223-232, 1981.

[30] R. W. Klessig, *An algorithm for nonlinear multicommodity flow problems*, *Networks*, vol. 4, no. 4, pp. 343-355, 1974.

[31] J. Xie and Y. Nie, *A new algorithm for achieving proportionality in user equilibrium traffic assignment*, *Transportation Science*, vol. 53, no. 2, pp. 566-584, 2019.

[32] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, 2013.

[33] D. P. Bertsekas, *Nonlinear Programming*, Journal of the Operational Research Society, 48(3), 334-334, 1997.

[34] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer New York, 1999.

[35] P. E. Fishback, *Linear and Nonlinear Programming with Maple: An Interactive, Applications-Based Approach*, CRC Press, 2009.

[36] M. J. D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Computer J., 7, 155-162, 1964.

[37] S. Moiseev, *Universal derivative-free optimization method with quadratic convergence*, arXiv preprint arXiv:1102.1347, 2011.

[38] S. C. Dafermos and F. T. Sparrow, *The traffic assignment problem for a general network*, *Journal of Research of the National Bureau of Standards B*, vol. 73, no. 2, pp. 91-118, 1969.

[39] B. Javani and A. Babazadeh, Origin-destination-based truncated quadratic programming algorithm for traffic assignment problem. *Transportation Letters*, 9(3), 166-176, 2017.

[40] A. Galligari and M. Sciandrone, A Convergent and Fast Path Equilibration Algorithm for the Traffic Assignment Problem. *Optimization Methods and Software*, 33(2), 354-371, 2018.

[41] A. Babazadeh, B. Javani, G. Gentile and M. Florian, Reduced gradient algorithm for user equilibrium traffic assignment problem. *Transportmetrica A: Transport Science*, 16(3), 1111-1135, 2020.