



Comparison of Convergence Speed among Conjugate Directions Traffic Assignment Algorithms

V. Karimi, A. Babazadeh*

School of Civil Engineering, College of Engineering, University of Tehran, Tehran, Iran

ABSTRACT: The Traffic assignment problem in the transport networks is formulated as a convex optimization problem using simplifier assumptions. Link-based, path-based and bush-based algorithms have been presented to solve this problem, among which the link-based algorithm has found more applications thanks to its low memory requirements. The link-based algorithm of Frank-Wolf (FW) is yet amongst the most popular assignment algorithms, because of its simplicity as well as high convergence speed during the initial iterations. However, the low convergence rate of this algorithm near the optimal solution has driven many studies that focus on modifying the FW search direction, resulted in newer link-based algorithms. Among them are conjugate direction algorithms which are more effective and simply implementable. These algorithms include PARTAN, conjugate FW (CFW) and bi-conjugate FW (BFW). This paper uses the Chicago Regional and Sioux-Falls test networks to make direct comparisons among these algorithms with respect to the CPU time and iteration number needed to reach various accurate solutions. The results for Chicago show that, compared with the FW algorithm, the three algorithms CFW, PARTAN increase the convergence speed to the relative error of 10-5 (i.e. a stable solution) by about 89, 72 and 63%, respectively, while only the BFW can reach to a 10-6 relative error. Comparing the results from Sioux-Fall with those of Chicago demonstrates that the performance of the conjugate directions Algorithms improves as network size decreases.

Review History:

Received: Mar. 09, 2021
Revised: Jul. 20, 2021
Accepted: Nov. 02, 2021
Available Online: Nov. 19, 2021

Keywords:

Traffic Assignment
Link-based algorithms
Frank-Wolf Algorithm
PARTAN Algorithm
Conjugate direction algorithms

1- Introduction

The issues existing in human and freight transportation are among the most important and fundamental problems of the present age. Moreover, these issues are extensively increasing due to the increase in the value of time as well as the formation of competitive markets in this area. Therefore, it is necessary to conduct continuous studies with the aim of improving or modifying existing transportation systems. The main part of such studies is modeling the traffic of transportation networks in different situations of travel demand and transportation systems.

Link flow patterns in a transportation network are obtained by solving a mathematical program called the traffic assignment problem (TAP). Given the amount of travel demands between the origin-destination (OD) pairs in the network, the TAP is aimed to answer how demands are distributed among existing paths connecting the OD pairs. This problem fundamentally obeys the used equilibrium principle [1] that is no user can reduce his own travel time by unilaterally changing his path.

Knowing that the evaluation of transportation policies (such as network expansion and congestion pricing) requires

so many times solving the TAP, providing fast convergent TAP algorithms has always been the focus of researchers in this field. Various solution methods have been proposed, differing more or less in fundamentals. They are generally divided into three categories: link-based, path-based and bush-based algorithms. The most famous link-based algorithm is the Frank-Wolf (FW) algorithm [2], which has been the most widely used TAP algorithm for the last five decades. The most important feature of this algorithm is its simplicity and low memory usage. Despite these positive features, the convergence rate of the FW algorithm is very low, especially when approaching the optimal solution.

2- Methodology

In this study, we investigate the convergence rate of the most important link-based TAP algorithms, including FW, PARTAN, conjugate FW (CFW) and Bi-conjugate FW (BFW) algorithms. The details of this algorithm can be found in [3].

Obviously, algorithms can be evaluated from different aspects, but without a doubt the convergence rate is one of the most important indicators in this type of evaluation. In this paper, to determine the convergence rate of the algorithms,

*Corresponding author's email: ababazadeh@ut.ac.ir



Table 1. Specifications of the test networks

Network	No. of Zones	No. of Nodes	No. of Links	Total Demand
Sioux Falls	24	24	76	360.6
Chicago	1790	12,982	39,018	1,360,428

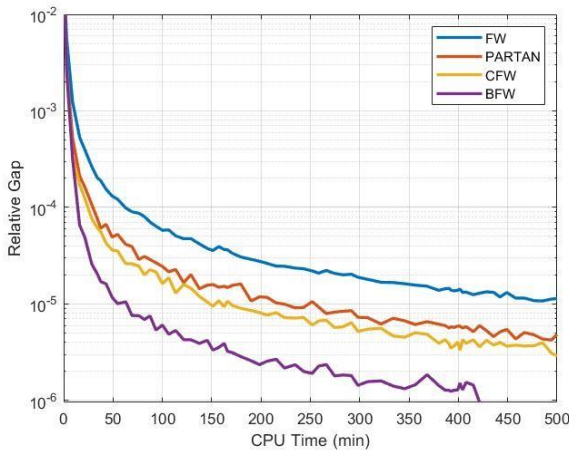


Fig. 1. Convergence rate of FW, PARTAN, CFW and BFW algorithms for Chicago network

the number of iterations and the time required to achieve a certain precision in solving the problem are used. In order to make correct comparisons between algorithms, a common structure must be used in their programming so that their results are comparable. Therefore, the computer codes of the aforementioned algorithms are written in a common structure with the same programming language, so that there is no difference between them except in the method of determining the search direction. In coding these algorithms, the bisection method [4] is used to perform linear search and the Golden algorithm [5] is used to solve the shortest path problem.

The algorithms are tested on two different networks, each of which has been used for a specific purpose. The small-sized Sioux Falls network is used to examine the solutions of these algorithms against each other as well as to control coding accuracy. Finally, the Chicago network is used as a large-scale network to compare the convergence rates of the competing algorithms. The characteristics of the test networks including the number of zones, nodes, links and sum of the elements of the demand matrix, are shown in Table 1 [6-8].

The intended algorithms were implemented in C++ programming environment by Visual Studio 2013 software on a computer with a 2.8GHz dual-core processor and 8GB of RAM. The relative gap (RGap) measure is used to set the convergence criterion, the most common criterion used in recent TAP algorithms. This measure is equal to the relative difference between the total travel time of the users and the total travel time of them as if they experienced the shortest

Table 2. Results of FW, PARTAN, CFW and BFW algorithms to the precision of 10⁻⁵ for Chicago network

Algorithm	No. of iterations	CPU Time (min)	RGap
FW	6050	504.47	0.00000971
PARTAN	2249	187.85	0.00000984
CFW	1659	140.24	0.00000978
BFW	642	53.27	0.00000950

Table 3. Required memory of the studied algorithms for Chicago network (MB)

FW	PARTAN	CFW	BFW
0.312	0.624	0.468	0.624

Table 4 Relative number of iterations of the studied algorithms to the precision of 10⁻⁵ for Chicago and Sioux Falls networks

Network	FW	PARTAN	CFW	BFW
Chicago	1	0.37	0.27	0.11
Sioux Falls	1	0.35	0.18	0.02

path travel time on the network. According to a study by Boyce et al. [9], link flows in large-scale networks stabilize after an RGap of 10⁻⁵ is reached. Therefore, in this study, for both the Sioux Falls and Chicago networks, the RG criterion is defined as reaching an RG of 10⁻⁵ or less in order to obtain adequate accurate solutions.

3- Results and Discussion

Figure 1 shows the convergence rates of the FW, PARTAN, CFW, and BFW algorithms with respect to the CPU time until RGap of 10⁻⁵ is achieved. As can be seen, only the BFW algorithm is able to hit the RGap of 10⁻⁶ in a reusable time. Moreover, Table 2 tabulates the number of iterations as well as the CPU, and Table 3 reports the memory required by each algorithm to reach this precision.

Table 4 shows the results of a direct comparison between competing algorithms in terms of the number of iterations up to the RGap of 10⁻⁵ for the Chicago and Sioux Falls networks. In this table, the number of iterations of the PARTAN, CFW and BFW algorithms are normalized with respect to that of the FW algorithm.

4- Conclusions

In this paper, the speed of convergence of FW, PARTAN, CFW, and BFW algorithms to stable solutions were investigated. Comparisons among the algorithm were performed under the same software and hardware conditions on the large-scale Chicago network. The results showed that the three algorithms BFW, CFW and PARTAN can reach relative gaps less than 10^{-5} by about 89, 72 and 63%, respectively, of the CPU time the FW algorithm needed. In other words, conjugating the search directions (even approximately) increased the efficiency of the base FW algorithm by more than 60%. The efficiency of the BFW algorithm was much higher among the studied algorithms, due to the fact that it applies the conjugation with respect to the last two directions. The CFW algorithm converges slower than the BFW algorithm, because in which the conjugation is performed only relative to the previous direction. In terms of performance, the PARTAN algorithm was placed between the CFW and FW algorithms, because it approximates the conjugation and also requires an additional linear search per iteration. By the way, among the three conjugate direction algorithms, only the BFW was able to find a solution with a relative gap of less than 10^{-6} in less than 500 minutes.

References

- [1] M.J. Beckmann, C.B. McGuire, C.B. Winsten, Studies in the economics of transportation, (1955).
- [2] L.J. LeBlanc, E.K. Morlok, W.P. Pierskalla, An efficient approach to solving the road network equilibrium traffic assignment problem, Transportation research, 9(5) (1975) 309-318.
- [3] M. Mitradjieva, P.O. Lindberg, The stiff is moving - conjugate direction Frank-Wolfe Methods with applications to traffic assignment, Transportation Science, 47(2) (2013) 280-293.
- [4] Y. Sheffi, Urban transportation networks, Prentice-Hall, Englewood Cliffs, NJ (1985).
- [5] B. Golden, Shortest-path algorithms: A comparison. Operations Research, 24(6), 1164-1168. 1976
- [6] Transportation networks for research. <https://github.com/bstabler/TransportationNetworks>. Accessed July, 20, 2019.
- [7] A. Babazadeh, B. Javani, Comparison Between Path-based Algorithms for Traffic Assignment Problem, in: Transportation Research Board 91st Annual Meeting, (2012).
- [8] H. Bar-Gera, Traffic assignment by paired alternative segments, Transportation Research Part B: Methodological, 44(8-9) (2010) 1022-1046.
- [9] D. Boyce, B. Ralevic-Dekic, H. Bar-Gera, Convergence of traffic assignments: how much is enough?, Journal of Transportation Engineering, 130(1) (2004) 49-55.

HOW TO CITE THIS ARTICLE

V. Karimi, A. Babazadeh, Comparison of Convergence Speed among Conjugate Directions Traffic Assignment Algorithms, Amirkabir J. Civil Eng., 54(6) (2022) 445-448.

DOI: 10.22060/ceej.2021.19708.7244





مقایسه سرعت همگرایی بین الگوریتم های تخصیص ترافیک جهات مزدوج

وحید کریمی، عباس بابازاده*

دانشکده مهندسی عمران، دانشگاه تهران، تهران، ایران .

تاریخچه داوری:

دریافت: ۱۳۹۹/۱۲/۱۹
بازنگری: ۱۴۰۰/۰۴/۲۹
پذیرش: ۱۴۰۰/۰۸/۱۱
ارائه آنلاین: ۱۴۰۰/۰۸/۲۸

کلمات کلیدی:

تخصیص ترافیک
الگوریتم های بر پایه کمان
الگوریتم فرانک - ولف
الگوریتم پارتان
الگوریتم های فرانک - ولف مزدوج

خلاصه: مسئله تخصیص ترافیک در شبکه های حمل و نقلی تحت فروضی ساده کننده به صورت یک مسئله بهینه سازی محدب فرمول بندی می شود. برای حل این مسئله الگوریتم های بر پایه کمان، بر پایه مسیر و بر پایه مبدا ارائه شده اند. در این بین، الگوریتم های بر پایه کمان به دلیل حافظه مصرفی کمتر، کاربرد بیشتری یافته اند. الگوریتم بر پایه کمان فرانک - ولف به دلیل سادگی و نیز سرعت همگرایی زیاد آن در تکرارهای اولیه هنوز جزو محبوب ترین الگوریتم های تخصیص ترافیک محسوب می شود. ولی، این الگوریتم در نزدیکی جواب بهینه دارای همگرایی ضعیفی است، و به همین علت تاکنون پژوهش های متعددی با هدف اصلاح جهت جست و جوی فرانک - ولف انجام شده است. الگوریتم های جهات مزدوج مؤثرترین نوع این الگوریتم ها بوده و در ضمن پیاده سازی آن ها بسیار ساده تر است. این الگوریتم ها شامل پارتان، فرانک - ولف مزدوج، فرانک - ولف دو مزدوج می باشند. در این مقاله مقایسه هایی مستقیم از نظر زمان حل و تعداد تکرار تا رسیدن به دقت های مختلف جواب بین این چهار الگوریتم روی شبکه بزرگ مقیاس شیکاگو و شبکه کوچک مقیاس سوفالز انجام می شود. نتایج نشان می دهند که سه الگوریتم فرانک - ولف دو مزدوج، فرانک - ولف مزدوج و پارتان، در مقایسه با الگوریتم فرانک - ولف، سرعت همگرایی به جوابی با خطای ۵-۱۰ (جواب پایدار) را برای شبکه شیکاگو به ترتیب در حدود ۸۹، ۷۲ و ۶۳ درصد افزایش می دهند. در ضمن، فقط الگوریتم فرانک - ولف دو مزدوج توانایی رسیدن به خطای ۶-۱۰ را دارد. مقایسه نتایج شبکه سوفالز با نتایج شبکه شیکاگو نشان می دهد که کارایی الگوریتم های جهات مزدوج با کاهش اندازه شبکه افزایش می یابد.

۱- مقدمه

سیستم را در افق های زمانی مختلف و تحت گزینه های مختلف پیش بینی کرد. مدل سازی جریان ترافیک عبوری از بخش های مختلف شبکه معابر شهری (خیابان ها، بزرگراه ها و تقاطعات) بخش مهمی از مدل سازی سیستم حمل و نقل را تشکیل می دهد، زیرا بدون آگاهی از جریان ترافیک، امکان ارزیابی گزینه های رقیب وجود ندارد.

شبکه معابر به عنوان بخش اصلی سیستم حمل و نقل وظیفه فراهم ساختن امکان جابه جایی وسایل نقلیه شخصی بین زوج های مبدا - مقصد در شبکه را بر عهده دارد. شبکه معابر در ساده ترین حالت شامل کمان ها به عنوان معابر و گره ها به عنوان تقاطعات است. نرخ تقاضای سفر با وسایل نقلیه شخصی بین زوج های مبدا - مقصد (گره های شروع و خاتمه سفر) در یک دوره زمانی مفروض به صورت یک ماتریس تحت عنوان ماتریس تقاضای مبدا - مقصد معرفی می شود. مسئله تعیین جریان در کمان های شبکه معابر تحت یک ماتریس تقاضای مفروض به عنوان مسئله تخصیص

حمل و نقل نقش پشتیبان فعالیت های اقتصادی - اجتماعی را در جوامع شهری بازی می کند. از این رو هر نوع بهبود در سیستم حمل و نقل (افزایش کارایی یا کاهش هزینه) موجب توسعه شهری در ابعاد اقتصادی و اجتماعی می شود. از طرفی، این موضوع مسائل و مشکلاتی را برای استفاده کنندگان و غیراستفاده کنندگان از سیستم حمل و نقل ایجاد می کند، که از جمله آن ها می توان به شلوغی، آلودگی هوا و تصادفات اشاره کرد. بنابراین، مطالعات برنامه ریزی و مدیریت حمل و نقل در راستای حل این مسائل کاملاً ضروری بوده، و نیز با توجه به آهنگ افزایش جمعیت و مالکیت وسایل نقلیه شخصی نیازمند استمرار است. هدف از انجام مطالعات حمل و نقل ارائه گزینه های کوتاه و بلند مدت برای اصلاح سیستم حمل و نقل شهری جهت حل مسائل آینده استفاده کنندگان و غیراستفاده کنندگان از سیستم است. لازمه این کار مدل سازی سیستم حمل و نقل است تا به کمک آن بتوان وضعیت

* نویسنده عهده دار مکاتبات: ababazadeh@ut.ac.ir



الگوریتم تانژانت‌های موازی^{۱۴} سعی می‌کند جهت فرانک - ولف را با انجام یک جست‌وجوی خطی اضافی طوری اصلاح کند که جهت حاصل (جهت پارتان) بر گرادیان تابع هدف در جریان فعلی عمود نشود، و از این طریق سرعت همگرایی را افزایش دهد. تکنیک پارتان در ابتدا برای جلوگیری از حرکت زیگزاگی در روش‌های بر پایه گرادیان برای مسائل بهینه‌سازی بدون محدودیت اعمال شده بود. لوئینگر [۶] نشان داد که برای مسائل درجه دوم بدون محدودیت، روش پارتان جهات مزدوج تولید می‌کند. یک مشکل ذاتی استفاده از این تکنیک در روش فرانک - ولف تعیین حداکثر اندازه گام در جهت پارتان برای حفظ امکان‌پذیری است. الگوریتم پارتان برای مسئله تخصیص ترافیک توسط لبلانک و همکاران [۷] معرفی، و توسط فلورین و همکاران [۸] و سپس آرزی و ون ولیت [۹] با ارائه روابطی تحلیلی برای محاسبه اندازه گام ماکزیمم پارتان توسعه داده شد. در هر تکرار این الگوریتم، ابتدا یک جست‌وجوی خطی از جواب فعلی در جهت فرانک - ولف انجام شده و یک جواب میانی (جواب فرانک - ولف) به دست می‌آید. سپس، جواب بعدی با انجام یک جست‌وجوی خطی دیگر از جواب قبلی در جهت پارتان (جواب قبلی به سمت جواب میانی) تعیین می‌شود. با این روش از شدت حرکت زیگزاگی فرانک - ولف به سمت جواب بهینه کاسته می‌شود. اگر چه این الگوریتم نرخ همگرایی فرانک - ولف را بهبود می‌بخشد، اما به علت پیچیدگی ناشی از جست‌وجوی اضافی و محاسبه گام ماکزیمم نتوانست جایگزین مناسبی برای آن شود.

فوکوشیما [۱۰] با ذخیره کردن جواب‌های تخصیص‌های همه یا هیچ (نقاط حدی^{۱۵}) چندین تکرار قبلی، دو جهت جست‌وجو در هر تکرار محاسبه می‌کند. اولین جهت از ترکیب محدب نقاط حدی ذخیره شده و نقطه حدی به دست آمده در تکرار فعلی (۲ تا ۱۰ تکرار متوالی، در مجموع) با استفاده از ضرایب وزنی یکنواخت به دست می‌آید، و دومین جهت همان فرانک - ولف حاصل از تخصیص همه یا هیچ در تکرار فعلی است. سپس، مشتقات جهتی تابع هدف در جواب فعلی در این دو جهت (پس از نرمال‌سازی جهات) محاسبه و جهتی که مشتق جهتی کمتری داشته باشد، به عنوان جهت جست‌وجو بهتر انتخاب می‌شود. بنابراین می‌توان انتظار داشت که الگوریتم فرانک - ولف اصلاح شده فوکوشیما حداقل به اندازه الگوریتم اصلی فرانک - ولف کارایی داشته باشد. در این الگوریتم، جست‌وجوی خطی^{۱۶} (تعیین

ترافیک^۱ [۱] نامیده می‌شود. این مسئله بر اساس قانون تعادل استفاده کننده^۲ به صورت یک مدل بهینه‌سازی به نام مدل بکمن^۳ [۲] فرمول‌بندی شده است. در شرایط تعادل استفاده کننده، هیچ استفاده کننده‌ای نمی‌تواند با تغییر یک جانبه مسیر خود زمان سفر مبدا - مقصد خود را کاهش دهد. در این مقاله، منظور از حل مسئله تخصیص ترافیک، همان حل مدل بکمن است. با توجه به این که ارزیابی گزینه‌های شبکه معابر در مطالعات حمل‌ونقل نیازمند حل مکرر این مسئله است، ارائه و شناخت کاربردی‌ترین و در عین حال سریع‌ترین روش حل برای این مسئله ضروری است.

برای حل مسئله تخصیص ترافیک روش‌های گوناگونی مطرح شده‌اند که به طور کلی به سه نوع الگوریتم‌های بر پایه کمان^۴، بر پایه مسیر^۵ و بر پایه بوش^۶ تقسیم‌بندی می‌شوند. الگوریتم‌های بر پایه کمان نسبت به دو نوع دیگر از حافظه کمتری استفاده می‌کنند، ولی در عوض نرخ همگرایی ضعیف‌تری دارند. نام آشناترین این الگوریتم‌ها، الگوریتم فرانک - ولف^۷ است. این الگوریتم در ابتدا توسط فرانک و ولف [۳] برای حل مسائل برنامه‌ریزی درجه دوم با محدودیت‌های خطی ارائه شده بود، و سپس لبلانک و همکاران [۴] از آن برای حل مدل بهینه‌سازی بکمن استفاده کردند. در هر تکرار این الگوریتم، جهت جست‌وجو^۸ توسط یک بار تخصیص همه یا هیچ^۹، و اندازه گام^{۱۰} توسط روش تقسیم دودویی^{۱۱} [۱] (الگوریتم بولزانو^{۱۲}) محاسبه می‌شوند. از دلایل محبوبیت این الگوریتم می‌توان به سادگی محاسبه جهات جست‌وجو و نیز سرعت همگرایی بالای آن در تکرارهای اولیه اشاره کرد. ولی، نرخ همگرایی این الگوریتم در نزدیکی جواب بهینه کاهش می‌یابد، که علت آن متعامد شدن جهت جست‌وجو با گرادیان تابع هدف (در جریان فعلی) و در نتیجه حرکت زیگزاگی^{۱۳} به سمت جواب بهینه است [۵]. به همین دلیل تلاش‌هایی برای اصلاح جهت جست‌وجو [۷-۱۳] یا اندازه گام در الگوریتم فرانک - ولف انجام شده است [۱۵]. نتیجه این تلاش‌ها ارائه الگوریتم‌های بر پایه کمان جدیدتر بوده است که در ادامه مرور می‌شوند.

- 1 Traffic Assignment Problem (TAP)
- 2 User Equilibrium (UE) Law
- 3 Beckmann's Model
- 4 Link-Based
- 5 Path-Based
- 6 Bush-Based
- 7 Frank-Wolfe (FW) Algorithm

Search Direction 8

- 9 All or Nothing (AON)
- 10 Step Size
- 11 Bisection Method
- 12 Bolzano Search
- 13 Zigzagging

- 14 Parallel Tangents (PARTAN)
- 15 Extreme Points
- 16 Line Search

تکرار قبلی، و در الگوریتم فرانک - ولف دو مزدوج با جهات دو تکرار قبلی مزدوج می‌شود. همچنین، هولمگرن و لیندبرگ [۱۴] با مطالعه روی نحوه بهنگام‌سازی زیر مسائل، عملکرد الگوریتم فرانک - ولف دو مزدوج را ارتقاء بخشیدند.

انجام جست‌وجوی خطی برای بهینه‌سازی اندازه گام تضمین کننده همگرایی الگوریتم فرانک - ولف به جواب تعادلی است [۴]. همچنین، پاول و شفلی [۱۵] ثابت کردند که این الگوریتم با دنباله‌ای از اندازه گام‌های از پیش تعیین شده نیز تحت شرایطی خاص همگرا می‌شود. وینتراوب و همکاران [۱۶] با اصلاح اندازه گام‌های حاصل از جست‌وجوی خطی توانستند حرکت زیگزاگی الگوریتم فرانک - ولف به سمت جواب بهینه را تا حدی کاهش دهند، و این روش را نوعی جبران برای عدم کارایی جهت فرانک - ولف نامیدند. الگوریتم پیشنهادی آن‌ها مشابه الگوریتم فرانک - ولف است، با این فرق که اندازه گام بهینه در هر تکرار، در عددی بزرگ‌تر از یک ضرب شده و در عین حال برای حفظ امکان‌پذیری به ۱ محدود می‌شود. در ضمن، اگر اندازه گام اصلاح شده کاهش در تابع هدف ایجاد نکند، از همان اندازه گام عادی فرانک - ولف در آن تکرار استفاده می‌شود. این پژوهشگران با آزمایش الگوریتم پیشنهادی خود روی شبکه‌های مختلف کارایی قابل توجه آن را نشان دادند.

الگوریتم‌های بر پایه مسیر، برخلاف الگوریتم‌های بر پایه کمان، مسئله تخصیص ترافیک را بر حسب جریان در مسیرها حل می‌کنند. سرعت همگرایی این الگوریتم‌ها به جواب‌های دقیق بسیار بیشتر از الگوریتم‌های بر پایه کمان است، ولی در عوض حافظه‌ی مصرفی بالاتری نیز دارند، زیرا مجبورند اطلاعات مسیرهای فعال (با جریان مثبت) بین زوج‌های مبدا - مقصد را در طول اجرای الگوریتم ذخیره‌سازی کنند. در هر تکرار این الگوریتم‌ها، مسئله روی زوج‌های مبدا - مقصد تجزیه شده و زیر مسئله هر زوج مبدا - مقصد به صورت مجزا برای جریان در مسیرهای فعال بین آن زوج حل می‌شود. برخی از مهم‌ترین این الگوریتم‌ها در ادامه مرور می‌شوند. چن و همکاران [۱۷] با استفاده از تکنیک بهنگام‌سازی "هر بار یک مبدا - مقصد" الگوریتمی به نام فرانک - ولف بر پایه مبدا - مقصد^۹ را معرفی کردند، که تحول عمده‌ای در نرخ همگرایی الگوریتم فرانک - ولف ایجاد کرد. در هر تکرار این الگوریتم، مدل بکمن محدود شده به مسیرهای فعال (زیر مجموعه‌ای از تمام مسیرهای موجود که تاکنون تولید و ذخیره شده است) روی زوج‌های مبدا - مقصد تجزیه شده، و هر زیر مسئله با استفاده از

اندازه گام) توسط روش تقسیم طلایی^۱ انجام می‌شود. الگوریتم تجزیه سیمپلیکال محدود شده^۲ هرن و همکاران [۱۱] نیز جهت حرکت در هر تکرار را با استفاده از ترکیب محدب مجموعه‌ای از تعدادی محدود نقطه حدی ذخیره شده به دست می‌آورد، ولی از چند جنبه با الگوریتم فوکوشیما متفاوت است: نخست، مجموعه نقاط حدی ذخیره شده لزوماً مربوط به تکرارهای متوالی نیستند (هر چند نقطه حدی تکرار فعلی را همیشه شامل می‌شود)؛ و دومین تفاوت، ضرایب وزنی نقاط حدی ذخیره شده از حل یک مسئله مادر^۳ توسط روش نیوتون تصویر شده^۴ به دست می‌آیند. این پژوهشگران ثابت کردند که اگر تعداد نقاط حدی ذخیره شده به قدر کافی بزرگ باشند، جواب بهینه مسئله در تعداد محدودی تکرار به دست می‌آید. در ضمن، الگوریتم فرانک - ولف حالت خاصی از الگوریتم تجزیه سیمپلیکال محدود شده است، اگر فقط یک نقطه حدی ذخیره شود. الگوریتم ارائه شده توسط لی و نی [۱۲] نیز مشابه الگوریتم فوکوشیما است، با این تفاوت که ترکیب محدب نقاط حدی به دست آمده در تکرارهای قبلی، به جای همه این نقاط حدی، در یک بردار کمکی ذخیره می‌شود. این برداری کمکی در هر تکرار با ترکیب خطی خودش و نقطه حدی فعلی (حاصل از تخصیص همه یا هیچ) بهنگام می‌شود، در حالی که ضریب وزنی این ترکیب با افزایش شماره تکرار افزایش می‌یابد. میتراجیوا و لینبرگ [۱۳] الگوریتم‌های فرانک - ولف مزدوج^۵ و دو مزدوج^۶ را با هدف اصلاح جهت جست‌وجوی فرانک - ولف ارائه کردند. در روش فرانک - ولف مزدوج جهت جست‌وجو با ترکیب جهت فعلی و جهت تکرار قبل و همچنین در نظر گرفتن هسیان^۷ تابع هدف (در جواب فعلی) به دست می‌آید. در الگوریتم فرانک - ولف دو مزدوج جهت جست‌وجو با ترکیب جهت فعلی و جهات دو تکرار قبلی و در نظر گرفتن هسیان تابع هدف به دست می‌آید. ایده این دو الگوریتم برگرفته از روش گرادیان مزدوج^۸ [۶] است، که در آن جهت جست‌وجو در هر تکرار از ترکیب محدب گرادیان تابع هدف در نقطه فعلی و جهت جست‌وجوی تکرار قبلی به گونه‌ای به دست می‌آید که این دو جهت با هم مزدوج شوند (در حالت درجه دوم، جهت جست‌وجو در هر تکرار با تمام جهات قبلی خود نیز مزدوج می‌شود [۶]). جهت جست‌وجو در هر تکرار الگوریتم فرانک - ولف مزدوج با جهت

- 1 Golden Section Method
- 2 Restricted Simplicial Decomposition (RSD)
- 3 Master Problem
- 4 Projected Newton
- 5 Conjugate FW (CFW)
- 6 Bi-conjugate FW (BFW)
- 7 Hessian
- 8 Conjugate Gradient

9 ² Origin-destination Based FW (ODBFW)

مبدأ^۶ توسط بارگرا [۲۴] پیشنهاد شد. در این الگوریتم، سهم هر کمان از کل جریان ورودی به گره انتهایی آن کمان به عنوان متغیر اصلی مسئله در نظر گرفته می‌شود. همچنین، برای بهنگام‌سازی جریان در مسیرهای روی هر بوش با مبدأ مفروض، زیرمسئله تجزیه شده آن مبدأ توسط روش شبه - نیوتون تصویر شده^۷ حل می‌شود. الگوریتم ارائه شده توسط دایال [۲۵] یکی دیگر از الگوریتم‌های بر پایه بوش است. در این الگوریتم، برای به تعادل رساندن هر بوش، جریان از مسیر دارای بیشترین زمان سفر بین گره مبدأ بوش و هر گره مقصد، به مسیر با کمترین زمان سفر بین آن زوج مبدأ - مقصد منتقل می‌شود، و میزان این انتقال با حل یک معادله غیرخطی توسط روش نیوتون به دست می‌آید. این فرایند تا وقتی که بوش فعلی به تعادل برسد، تکرار شده و سپس نوبت بوش بعدی می‌رسد. بارگرا [۲۶] الگوریتم تاپاس^۸ که یکی از پیچیده‌ترین الگوریتم‌های تخصیص ترافیک بر پایه بوش است را ارائه کرد. یک پاس بنا به تعریف یک زوج قطعه (دنباله‌ای از کمان‌ها) موازی بین دو گره مفروض از شبکه است. ایده اصلی الگوریتم تاپاس انتقال جریان بین قطعه‌های هر پاس به منظور متعادل ساختن زمان سفر آن‌ها است. در صورتی که زمان سفر داخل تمام پاس‌ها در تعادل باشند، آن‌گاه شرایط تعادل استفاده کننده نیز برقرار خواهد بود. شی و شی [۲۷] با ایجاد تغییراتی در الگوریتم تاپاس، الگوریتم تاپاس بهبود یافته را معرفی کردند. این الگوریتم داری سرعت همگرایی بیشتری نسبت به تاپاس است. الگوریتم‌های جهات مزدوج شامل پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج جزو مهم‌ترین اصلاحات فرانک - ولف محسوب می‌شوند و نیز پیاده‌سازی آن‌ها بسیار ساده است. میتراجیوا و لینبرگ [۱۳] بر اساس شواهدی از مراجع مختلف و نیز ارائه نتایج عددی برای شبکه بزرگ مقیاس شبکه‌گو نشان دادند که الگوریتم‌های فرانک - ولف مزدوج و فرانک - ولف دو مزدوج نسبت به سایر اصلاحات الگوریتم فرانک - ولف (به جز پارتان) در تعداد تکرارهای به مراتب کمتری به جوابی با خطای 10^{-4} می‌رسند، ولی مقایسه‌ای برای دقت‌های کم‌تر و بیشتر از دقت 10^{-4} ارائه نکرده‌اند (فقط مقایسه‌ای بین الگوریتم‌های فرانک - ولف، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج بر حسب تعداد تکرار ارائه شده است، که ممکن است نشانگر سرعت همگرایی آن‌ها نباشد). از طرفی، طبق مطالعه بویس و همکاران [۲۸]، جریان‌هایی کمانی در شبکه‌های بزرگ مقیاس پس از رسیدن به

روش فرانک - ولف بر حسب جریان‌های مسیری حل می‌شود. مسیرهای فعال بین زوج‌های مبدأ - مقصد با استفاده از یک روش تولید ستون^۱ در هر تکرار الگوریتم بهنگام می‌شوند. دو تکنیک تجزیه مسئله روی زوج‌های مبدأ - مقصد و تولید مسیرهای فعال در سایر الگوریتم‌های بر پایه مسیر نیز انجام می‌شوند. فلورین و همکاران [۱۸] با بهره بردن از روش گرادیان تصویر شده^۲، الگوریتم بر پایه مسیری معرفی کردند که در آن جریان بین هر مبدأ - مقصد از مسیرهای با زمان سفر بیشتر از میانگین به مسیرهای با زمان سفر کمتر از آن جابه‌جا می‌شود. کومار و پیتا [۱۹] با الهام از فلورین و همکاران [۱۸] الگوریتم بر پایه مسیر دیگری ارائه دادند که در آن جریان در مسیرهای بین هر مبدأ - مقصد بر اساس حساسیت زمان سفر هر مسیر به جریان آن جابه‌جا می‌شود. دی لورنزو و همکاران [۲۰] الگوریتم کمینه‌سازی متوالی نادقیق را بسط دادند که جریان مسیرهای هر زوج مبدأ - مقصد را از مسیر با بیشترین زمان سفر به مسیرهای با زمان سفر کمتر انتقال می‌دهد. شی و همکاران [۲۱] الگوریتم بر پایه مسیر دیگری به نام گریدی^۳ ارائه داده‌اند که در آن از روشی حریصانه برای حل زیر مسائل تجزیه شده زوج‌های مبدأ - مقصد استفاده می‌شود. این روش اولین بار توسط جنتیله [۲۲] در الگوریتم لوچه^۴ معرفی شد. بابازاده و همکاران [۲۳] یک الگوریتم تخصیص ترافیک بر پایه مسیر توسعه داده‌اند که در آن هر زیر مسئله با استفاده از روش گرادیان تقلیل یافته^۵ حل می‌شود. در این روش، ابعاد هر زیر مسئله با تجزیه مسیرهایی فعال آن به دو زیر مجموعه پایه و غیرپایه، و بازنویسی زیرمسئله بر حسب مسیرهای غیرپایه کاهش داده می‌شود. نتایج اجرای الگوریتم گرادیان تقلیل یافته روی شبکه‌های بزرگ مقیاس شبکه‌گو و فیلادفیا نشان داد که این الگوریتم سریع‌تر از الگوریتم گریدی [۲۱] به جواب‌های بسیار دقیق می‌رسد.

الگوریتم‌های بر پایه بوش، گروه دیگری از الگوریتم‌های تخصیص ترافیک هستند که در آن‌ها مسئله تخصیص ترافیک به جای زوج‌های مبدأ - مقصد، روی مبداها (یا مقصدها) تجزیه می‌شود. این الگوریتم‌ها با تشکیل یک زیرشبکه بدون حلقه (بوش) برای هر مبدأ آغاز شده، و در هر تکرار ابتدا بوش‌ها بهنگام‌سازی شده و سپس جریان در مسیرهای روی هر بوش به تعادل رسانده می‌شوند. اولین الگوریتم بر پایه بوش به نام الگوریتم بر پایه

6 ² Origin-Based Algorithm (OBA)

7 ³ Projected Quasi-Newton

8 ⁴ Traffic Assignment by Paired Alternative Segments (TAPAS)

1 ³ Column Generation

2 ⁴ Projected Gradient (PG)

3 ⁵ Greedy

4 Linear User Cost Equilibrium (LUCE)

5 Reduced Gradient (RG)

تعریفی بین جریان‌های مسیری و کمانی است. تابع زمان سفر $t_a(x_a)$ تابعی مشتق‌پذیر پیوسته^۱، مثبت و اکیداً صعودی فرض می‌شود. در این مقاله از تابع زمان سفر BPR^2 که در این شرایط صادق است، استفاده می‌شود:

$$t_a(x_a) = t_a^0 \left[1 + 0.15 \left(\frac{x_a}{q_a} \right)^4 \right] \quad (2)$$

که در آن t_a^0 زمان سفر آزاد، و q_a ظرفیت عملی^۲ کمان a است. همچنین، جهت سهولت در بیان روابط، $t(x) = (t_a(x_a))$ به عنوان تابع برداری زمان سفر کمان‌ها تعریف می‌شود.

فرض کنید $X \subset \mathbb{R}^{|A|}$ مجموعه جواب‌های امکان‌پذیر مسئله (۱) است. با توجه خطی بودن محدودیت‌های مسئله (۱) و نیز مشتق‌پذیر پیوسته بودن توابع $t_a(x_a)$ ، واضح است که $Z: X \rightarrow \mathbb{R}$ تابعی دو بار مشتق‌پذیر پیوسته روی چند وجهی محدب X می‌باشد. همچنین، چون $t_a(x_a)$ اکیداً صعودی است، تابع هدف مسئله (۱) اکیداً محدب بوده و در نتیجه مسئله دارای جواب یگانه $x \in X$ است [۲].

الگوریتم اصلی فرانک - ولف توسط فرانک و ولف [۳] برای حل مسائل برنامه‌ریزی درجه دوم با محدودیت‌های خطی ارائه شده است. در هر تکرار n ، این الگوریتم تابع Z را با خطی‌سازی در نقطه x_n به صورت زیر تقریب می‌زند:

$$z_n(x) \approx z(x_n) + \nabla z(x_n)^T (x - x_n) \quad (3)$$

(رونویس T به معنای ترانهادگی است). گام نخست در این الگوریتم تعیین جهت جست‌وجو است، که با مینیمم‌سازی تابع $z_n(x)$ روی X انجام می‌شود:

$$\min_{x \in X} z_n(x) \quad (4)$$

جوابی با خطای 10^{-5} پایدار می‌شوند. بنابراین، در این مقاله مقایسه‌هایی مستقیم بین الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج از نظر سرعت همگرایی به جوابی با خطای کمتر از 10^{-5} روی یک شبکه بزرگ مقیاس و یک شبکه کوچک مقیاس انجام می‌شود. این مقایسه‌ها در شرایط نرم افزاری و سخت افزاری یکسانی برای تمام الگوریتم‌ها انجام می‌شود و بنابراین سرعت الگوریتم‌های جهات مزدوج را نسبت به الگوریتم پایه فرانک - ولف روشن می‌سازد.

۲- فرمول‌بندی مسئله تخصیص ترافیک و الگوریتم فرانک - ولف

مدل بکمن [۲] برای مسئله تخصیص ترافیک یک مسئله بهینه‌سازی با تابع هدف غیرخطی و محدودیت‌های خطی است. برای یک شبکه حمل‌ونقل با مجموعه کمان‌های A و مجموعه زوج‌های مبدا-مقصد I این مسئله به صورت زیر بیان می‌شود:

$$\begin{cases} \min z(x) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw \\ s.t. \\ \sum_{k \in K_i} h_k = D_i & \forall i \in I \\ h_k \geq 0 & \forall k \in K_i, i \in I \\ x_a = \sum_{i \in I} \sum_{k \in K_i} h_k \delta_{ak} & \forall a \in A \end{cases} \quad (1)$$

که در آن x_a جریان در کمان a ، $x = (x_a)$ بردار جریان‌های کمانی، $t_a(x_a)$ تابع زمان سفر کمان a ، K_i مجموعه مسیرهای بین زوج مبدا - مقصد i ، D_i تقاضای ثابت بین زوج مبدا - مقصد i ، h_k جریان در مسیر k ، و $z(x)$ تابع هدف است. همچنین δ_{ak} متغیری دودویی است، که برابر با یک است اگر کمان a روی مسیر k واقع باشد، و در غیر این صورت برابر صفر است.

تابع هدف مسئله (۱) برابر با مجموع انتگرال‌های توابع زمان سفر کمان‌ها در جریان تعادلی x است. این تابع هیچ تعبیر اقتصادی و یا رفتاری ندارد، بلکه صرفاً یک تابع ساختگی جهت معادل ساختن مسئله بهینه‌سازی (۱) با شرایط تعادل استفاده کننده است [۲]. محدودیت اول این مسئله شرط بقای جریان، محدودیت دوم شرط غیرمنفی بودن جریان، و محدودیت سوم رابطه

1 ¹ Practical Capacity

2 ² Bureau of Public Roads (BPR)

3 ³ Continuously Differentiable

گام ۲ (شرط توقف): بردار زمان سفر $t_n = t_n(x_n)$ بهنگام کنید. جواب کمکی $y_n^{FW} \in X$ با انجام یک تخصیص همه یا هیچ بر اساس t_n ثابت بیابید. $RGap$ را طبق رابطه (۶) محاسبه کنید، و در صورت $RGap < \varepsilon$ توقف کنید.

گام ۳ (تعیین جهت فرانک - ولف): قرار دهید $d_n^{FW} = y_n^{FW} - x_n$ گام ۴ (جستجوی خطی): اندازه گام α_n را از حل مسئله (۵) به دست آورید.

گام ۵ (بهنگامسازی): قرار دهید $x_{n+1} = x_n + \alpha_n d_n^{FW}$ و $n = n + 1$ به گام ۲ بروید.

۳- الگوریتم‌های جهات مزدوج

در این بخش اصلاحات الگوریتم فرانک - ولف که در آن‌ها جهات جستجوی متوالی با یکدیگر مزدوج یا تقریباً مزدوج شده‌اند، ارائه می‌شوند. این الگوریتم‌ها شامل پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج هستند.

۳-۱- الگوریتم پارتان

روش پارتان یکی از اصلاحات معروف برای کاهش حرکت زیگزاگی الگوریتم فرانک - ولف است. این روش یک گام اضافی نسبت به فرانک - ولف انجام می‌دهد، که به وسیله آن جواب فرانک - ولف در هر تکرار (جواب میانی) با جواب به دست آمده در دو تکرار قبلی طوری ترکیب می‌شود که تابع هدف روی خط واصل این دو جواب (جهت پارتان) مینیمم شود. چالش ذاتی این روش تعیین اندازه گام ماکزیمم در جهت پارتان است تا جواب به دست آمده در هر تکرار، امکان‌پذیر باقی بماند. لبلانک و همکاران [۷] برای اولین بار الگوریتم پارتان را برای مسئله تخصیص ترافیک معرفی کردند و نشان دادند این الگوریتم موجب افزایش سرعت همگرایی در شبکه‌های واقعی می‌شود.

فرض کنید v_n و x_n به ترتیب جواب فرانک - ولف و جواب پارتان در تکرار n باشند. جواب بعدی x_{n+1} با انجام یک جستجوی خطی اضافی از x_{n-1} در پارتان، $d_n^P = v_n - x_{n-1}$ ، به دست می‌آید:

$$\min_{0 \leq \rho \leq \rho_n^{\max}} z(x_{n-1} + \rho d_n^P) \quad (7)$$

جواب مسئله خطی فوق با y_n^{FW} نشان داده شده و جواب کمکی فرانک - ولف نامیده می‌شود. با استفاده از این جواب کمکی، جهت جستجوی فرانک - ولف برابر با $d_n^{FW} = y_n^{FW} - x_n$ قرار داده می‌شود. توجه شود که حل مسئله خطی (۴) معادل با یک بار تخصیص همه یا هیچ به ازای بردار زمان سفرهای ثابت $t_n = (t_n(x_n))$ است. گام بعدی در الگوریتم فرانک - ولف، انجام یک جستجوی خطی در جهت فرانک - ولف برای یافتن جواب بعدی x_{n+1} است. به عبارت دیگر، مسئله یک متغیره مینیمم‌سازی $z(x)$ روی خط واصل x_n و نقطه کمکی y_n^{FW} حل می‌شود:

$$\min_{0 \leq \alpha \leq 1} z(x_n + \alpha d_n^{FW}) \quad (5)$$

که در آن α اندازه گام حرکت است. جواب α_n مسئله (۴) به عنوان اندازه گام بهینه برای انتخاب جواب بعدی $x_{n+1} = x_n + \alpha_n d_n^{FW}$ استفاده می‌شود. الگوریتم تکرار می‌شود تا وقتی که خطای جواب به دست آمده در محدوده‌ای قابل قبول باشد. بنابراین، این الگوریتم نیازمند ذخیره‌سازی دو بردار با طول $|A|$ در حافظه برای محاسبه جهت حرکت است.

شروط توقف مختلفی برای الگوریتم فرانک - ولف قابل استفاده است. شرط توقفی که در این مقاله استفاده می‌شود، شکاف نسبی^۱ بین کل زمان سفر در جواب فعلی x_n و کل زمان سفر در جواب کمکی y_n^{FW} است:

$$RGap = \frac{(x_n - y_n^{FW})^T t_n}{x_n^T t_n} \quad (6)$$

واضح است $0 \leq RGap < 1$ و هر چه به صفر نزدیک شود خطای جواب کاهش یافته و به جواب تعادل استفاده کننده نزدیک‌تر می‌شود.

گام‌های الگوریتم فرانک - ولف به طور زیر خلاصه می‌شوند:
گام ۱ (تعیین جواب اولیه): عدد اعشاری کوچک $\varepsilon > 0$ را انتخاب کنید. جواب اولیه $x_1 \in X$ را با انجام یک تخصیص همه یا هیچ بر اساس بردار زمان سفرهای $t_n = t_n(0)$ بیابید. شماره تکرار را برابر $n = 1$ قرار دهید.

¹ Relative Gap (RGap)

درجه دوم است. میتراجیوا و لینبرگ [۱۳] الگوریتم فرانک - ولف مزدوج را به شرح زیر برای حل مسئله تخصیص ترافیک معرفی کردند.

فرض کنید x_n جواب الگوریتم فرانک - ولف مزدوج در تکرار n است (شکل ۱)، که توسط یک جست‌وجوی خطی با اندازه گام α_{n-1} در جهت $d_{n-1}^{CFW} = s_{n-1}^{CFW} - x_{n-1}$ (جهت فرانک - ولف مزدوج) با استفاده از جواب دیگر $s_{n-1}^{CFW} \in X$ (جواب فرانک - ولف مزدوج) به دست آمده است. در این الگوریتم، جهت جدید $d_n^{CFW} = s_n^{CFW} - x_n$ طوری تعیین می‌شود که با جهت d_{n-1}^{CFW} نسبت به هسیان H_n تابع هدف Z در نقطه x_n ، $H_n = \nabla^2 Z(x_n)$ ، مزدوج باشد. برای این منظور، جواب کمکی $y_n^{FW} \in X$ و $s_n^{CFW} \in X$ به صورت زیر برابر ترکیب محدب s_{n-1}^{CFW} و y_n^{FW} (جواب کمکی فرانک - ولف در نقطه x_n) انتخاب می‌شود:

$$s_n^{CFW} = \beta_n s_{n-1}^{CFW} + (1 - \beta_n) y_n^{FW}, \quad 0 \leq \beta_n \leq 1 \quad (9)$$

که در آن

$$\beta_n = \frac{\bar{d}_{n-1}^T H_n d_n^{FW}}{\bar{d}_{n-1}^T H_n (d_n^{FW} - \bar{d}_{n-1})} \quad (10)$$

و $\bar{d}_{n-1} = s_{n-1}^{CFW} - x_n$ است. با این انتخاب، ماکزیمم اندازه گام α_n در جهت s_n^{CFW} همیشه برابر ۱ است. چند مشکل در مورد رابطه (۱۰) وجود دارد. نخست، باید داشته باشیم $\beta_n \in [0, 1]$. سپس، لازم است از انتخاب $\beta_n = 1$ اجتناب شود، زیرا در این صورت $s_n^{CFW} = s_{n-1}^{CFW}$ و جهت جدیدی به وجود نمی‌آید. در نهایت، اگر مخرج کسر برابر صفر شود (وقتی رخ می‌دهد که مثلاً $\alpha_{n-1} = 1$ باشد) انتخاب محتاطانه $d_n^{CFW} = d_n^{FW}$ (یا به عبارتی $\beta_n = 0$) انجام می‌شود. با تعریف N_n و D_n به عنوان صورت و مخرج رابطه (۱۰)، β_n به صورت زیر تعیین می‌شود:

$$\beta_n = \begin{cases} \frac{N_n}{D_n} & \text{if } D_n \neq 0 \text{ and } \frac{N_n}{D_n} \in [0, 1 - \delta] \\ 1 - \delta & \text{if } D_n \neq 0 \text{ and } \frac{N_n}{D_n} > 1 - \delta \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

که در آن ρ_n^{\max} اندازه گام ماکزیمم برای حفظ امکان‌پذیری جواب چهار بردار با طول $|A|$ در حافظه بین تکرارها است. بنابراین، این الگوریتم نیازمند نگهداری [۷] فرمول‌هایی مجزا برای تعیین ρ_n^{\max} در ۵ تکرار اول ارائه دادند. فلورین و همکاران [۸] با در نظرگیری شرط محدود کننده $\rho \geq 1$ رابطه‌ای تحلیلی برای محاسبه اندازه گام در هر تعداد تکرار ارائه دادند، و از زکی و ون ولیت [۹] این رابطه تحلیلی را برای پوشش دادن تمام حالات ممکن به صورت رابطه زیر تعمیم دادند:

$$\rho_n^{\max} = \begin{cases} \frac{1}{1 - (1 - \alpha_{n-1})(1 - \alpha_n) \left(1 - \frac{\rho_{n-1} - 1}{\rho_{n-1}^{\max} - 1}\right)} & \rho_{n-1} > 1 \\ \frac{1}{1 - (1 - \alpha_{n-1})(1 - \alpha_n) \rho_{n-1}} & \rho_{n-1} \leq 1 \end{cases} \quad (8)$$

گام‌های الگوریتم پارتان به شرح زیر هستند:

گام ۱ (تعیین جواب اولیه): گام ۱ الگوریتم فرانک - ولف را انجام دهید.

گام ۲ (شرط توقف): گام ۲ الگوریتم فرانک - ولف را انجام دهید.

گام ۳ (تعیین جهت فرانک - ولف): گام ۳ الگوریتم فرانک - ولف را

انجام دهید.

گام ۴ (تعیین جهت پارتان): اندازه گام α_n را از حل مسئله (۵) به

دست آورید و قرار دهید $v_n = x_n + \alpha_n d_n^{FW}$. اگر $n < 2$ قرار دهید $x_{n+1} = v_n$ ، $n = n + 1$ و به گام ۲ برگردید وگرنه قرار دهید $d_n^P = v_n - x_{n-1}$.

گام ۵ (جست‌وجوی خطی): ρ_n^{\max} را با استفاده از رابطه (۸)، و اندازه

گام ρ_n از حل مسئله (۷) به دست آورید.

گام ۶ (بهنگام‌سازی): قرار دهید $x_{n+1} = x_{n-1} + \rho_n d_n^P$.

$n = n + 1$ و به گام ۲ بروید.

۳-۲ الگوریتم فرانک - ولف مزدوج

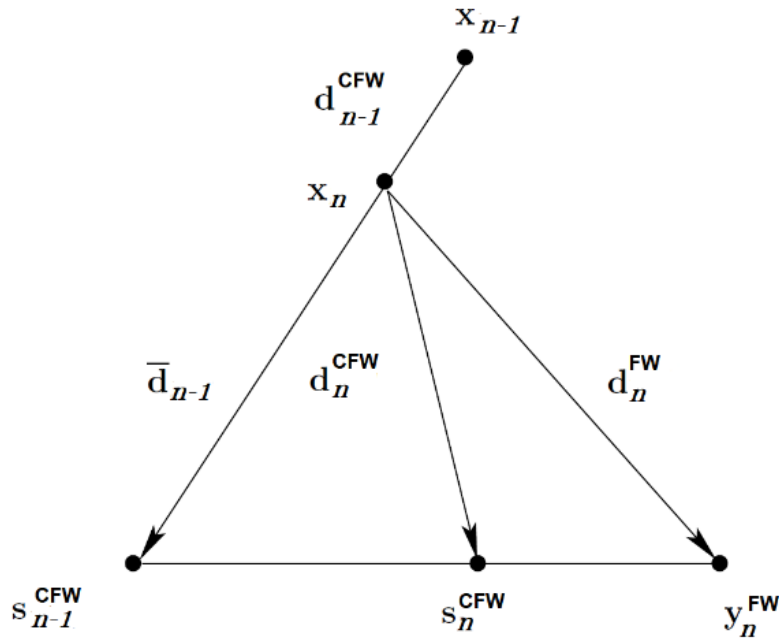
روش فرانک - ولف مزدوج تعمیمی از روش گرادیان مزدوج [۶] برای

حل مسائل بهینه‌سازی بدون محدودیت است. گرادیان مزدوج را می‌توان حد

وسطی بین روش‌های مرتبه اول (مثل فرانک - ولف) و روش‌های مرتبه

دوم بهینه‌سازی تصور کرد. انگیزه این روش سرعت بخشیدن به همگرایی

روش فرانک - ولف و در عین حال اجتناب از ملزومات محاسباتی مشتقات



شکل ۱. تعیین جهت جست و جو در الگوریتم فرانک - ولف مزدوج

Fig. 1. Determining the search direction in the CFW algorithm

گام ۶ (بهنگام‌سازی): گام ۵ الگوریتم فرانک - ولف را با استفاده از d_n^{CFW} به جای d_n^{FW} انجام دهید.

۳-۳ الگوریتم فرانک - ولف دو مزدوج

روش فرانک - ولف مزدوج را می‌توان یک مرحله توسعه داد تا از نظر حافظه مصرفی در شرایطی یکسان با روش پارتان قرار گیرد، بدین صورت که مزدوج‌سازی به جای جهت قبلی نسبت به دو جهت قبلی اعمال شود. الگوریتم فرانک - ولف دو مزدوج بر اساس همین ایده به عنوان تعمیمی از الگوریتم فرانک - ولف مزدوج برای حل مسئله تخصیص ترافیک ارائه شده است [۱۳].

برای توصیف الگوریتم فرانک - ولف دو مزدوج، فرض کنید x_n جواب فعلی است (شکل ۲)، که با یک جست‌وجوی خطی با اندازه گام α_{n-1} در جهت $d_{n-1}^{BFW} = s_{n-1}^{BFW} - x_{n-1}$ (جهت فرانک - ولف دو مزدوج) به کمک جواب دیگر $s_{n-1}^{BFW} \in X$ به دست آمده است (x_{n-1} نیز به نوبه خود با یک جست‌وجوی خطی با اندازه گام α_{n-2} در جهت $d_{n-2}^{BFW} = s_{n-2}^{BFW} - x_{n-2}$ به دست آمده است). همچنین، فرض کنید

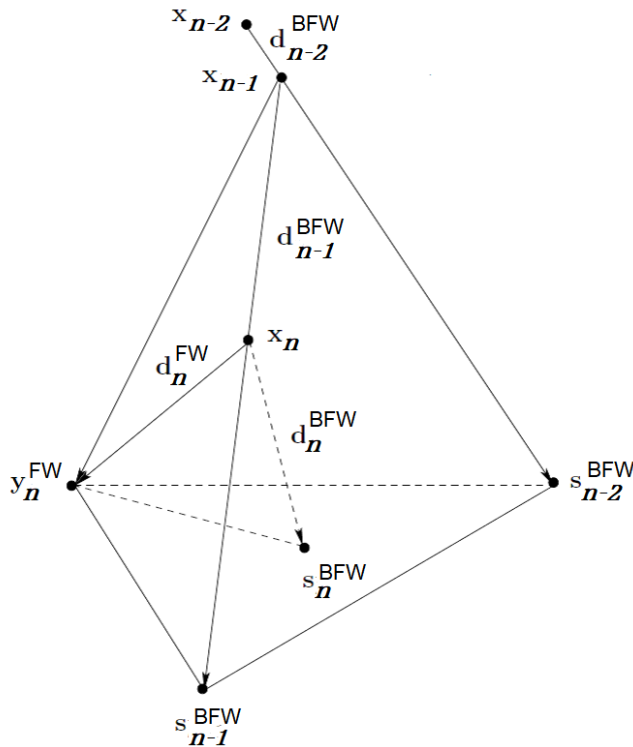
که در آن $\delta > 0$ عددی کوچک است. جاذبه این روش به خاطر فرمول‌های ساده (۹) و (۱۰) برای بهنگام‌سازی جهت حرکت است. پیاده‌سازی این روش تنها اندکی پیچیده‌تر از فرانک - ولف است، ولی در تعداد کمتری تکرار به جواب می‌رسد. این الگوریتم نیازمند نگهداری سه بردار با طول $|A|$ در حافظه برای محاسبه جهت حرکت است.

گام‌های الگوریتم فرانک - ولف مزدوج به شرح زیر قابل بیان هستند: گام ۱ (تعیین جواب اولیه): گام ۱ الگوریتم فرانک - ولف را انجام دهید. عدد کوچک $\delta > 0$ را انتخاب کنید.

گام ۲ (شرط توقف): گام ۲ الگوریتم فرانک - ولف را انجام دهید. گام ۳ (تعیین جهت فرانک - ولف): گام ۳ الگوریتم فرانک - ولف را انجام دهید.

گام ۴ (تعیین جهت فرانک - ولف مزدوج): اگر $n = 1$ یا $\alpha_{n-1} = 1$ قرار دهید $s_n^{CFW} = y_n^{FW}$. وگرنه β_n را طبق رابطه (۱۱) و s_n^{CFW} را طبق رابطه (۹) محاسبه کنید. قرار دهید $d_n^{CFW} = s_n^{CFW} - x_n$.

گام ۵ (جست‌وجوی خطی): گام ۴ الگوریتم فرانک - ولف را با استفاده از d_n^{CFW} به جای d_n^{FW} انجام دهید.



شکل ۲. تعیین جهت جست و جو در الگوریتم فرانک - ولف دو مزدوج

Fig. 2. Determining the search direction in the BFW algorithm

ضرایب β_n^i طوری تعیین می‌شوند که جهت d_n^{BFW} با دو جهت قبلی d_{n-1}^{BFW} و d_{n-2}^{BFW} نسبت به هسیان H_n تابع هدف Z در نقطه x_n مزدوج باشند. به عبارت دیگر:

$$\begin{cases} (d_n^{BFW})^T H_n d_{n-1}^{BFW} = 0 \\ (d_n^{BFW})^T H_n d_{n-2}^{BFW} = 0 \end{cases} \quad (14)$$

میتراجیوا و لینبرگ [۱۳] با استفاده از روابط (۱۳) و (۱۴) و با فرض $(d_{n-1}^{BFW})^T H_n d_{n-2}^{BFW} = 0$ ضرایب $\beta_n^0, \beta_n^1, \beta_n^2$ را به صورت زیر تعیین کردند (فرض اخیر فقط به صورت تقریبی درست است، زیرا d_{n-1}^{BFW} با d_{n-2}^{BFW} نسبت به H_{n-1} مزدوج شده است):

$$\beta_n^0 = \frac{1}{1 + \mu_n + \nu_n}, \quad \beta_n^1 = \nu_n \beta_n^0, \quad \beta_n^2 = \mu_n \beta_n^0 \quad (15)$$

$y_n^{FW} \in X$ جواب مسئله خطی شده (۴) در نقطه x_n است. جهت جست‌وجوی جدید به صورت $d_n^{BFW} = s_n^{BFW} - x_n$ است، که در آن نقطه کمکی s_n^{BFW} ترکیب محدبی از y_n^{FW} و دو نقطه کمکی قبلی s_{n-1}^{BFW} و s_{n-2}^{BFW} می‌باشد. بنابراین، به ازای وزن‌های مناسب $\beta_n^i \geq 0$ ، با $\sum_{i=0}^2 \beta_n^i = 1$ داریم:

$$s_n^{BFW} = \beta_n^0 y_n^{FW} + \beta_n^1 s_{n-1}^{BFW} + \beta_n^2 s_{n-2}^{BFW} \quad (12)$$

که با استفاده از $d_n^{FW} = y_n^{FW} - x_n$ و $d_n^{BFW} = s_n^{BFW} - x_n$ نتیجه می‌دهد:

$$d_n^{BFW} = \beta_n^0 d_n^{FW} + \beta_n^1 (s_{n-1}^{BFW} - x_n) + \beta_n^2 (s_{n-2}^{BFW} - x_n) \quad (13)$$

از d_n^{BFW} به جای d_n^{FW} انجام دهید.

گام ۶ (بهنگام‌سازی): گام ۵ الگوریتم فرانک - ولف را با استفاده از

d_n^{BFW} به جای d_n^{FW} انجام دهید.

برای مقایسه روند حل مسئله توسط الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج، شکل ۳ فلوچارت این الگوریتم‌ها را در کنار یکدیگر نشان می‌دهد.

۴- نتایج عددی

بدیهی است که الگوریتم‌ها را از جنبه‌های مختلفی می‌توان مورد ارزیابی قرار داد، اما بدون شک نرخ همگرایی جزء مهم‌ترین شاخص‌های ارزیابی محسوب می‌شود. در این مقاله، برای مقایسه نرخ همگرایی الگوریتم فرانک - ولف با الگوریتم‌های جهات مزدوج پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج از زمان مورد نیاز برای رسیدن به جوابی با خطای مشخص استفاده می‌شود. برای انجام مقایسه‌ای صحیح بین الگوریتم‌ها باید از ساختاری مشترک برای کدنویسی آن‌ها استفاده شود تا نتایج با هم قابل مقایسه باشند. از این رو، کدهای کامپیوتری مورد استفاده در این مقاله با زبان برنامه‌نویسی یکسان نوشته شده‌اند، و زیربرنامه‌های مورد استفاده در آن‌ها (مثل زیربرنامه‌های کوتاه‌ترین مسیر و تقسیم دودویی) مشترک است. کلیه کدها در محیط برنامه‌نویسی ++C نوشته شده و توسط نرم‌افزار ویژوال استودیو ۲۰۱۳ روی کامپیوتری دارای پردازشگر چهار هسته‌ای ۲/۸ گیگاهرتز و ۸ گیگابایت حافظه RAM اجرا می‌شوند.

به منظور یکسان کردن شرط توقف الگوریتم‌ها از معیار شکاف نسبی طبق تعریف ۶ به عنوان معیار همگرایی استفاده می‌شود، که یکی از رایج‌ترین معیارهای همگرایی استفاده شده در الگوریتم‌های تخصیص ترافیک در سال‌های اخیر است. در ادامه، منظور از خطای جواب همان شکاف نسبی جواب است. با توجه به اهداف این مقاله (پاراگراف آخر بخش ۱ ملاحظه شود) از شکاف نسبی 10^{-5} به عنوان معیار همگرایی الگوریتم‌ها استفاده شده است.

شبکه شیکاگو با داشتن بیش از ۱۲۰۰۰ گره، ۳۹۰۰۰ کمان و سه میلیون زوج مبدا - مقصد جزو شبکه‌های بزرگ مقیاس محسوب می‌شود. این شبکه، در سال‌های اخیر، در مقالات متعددی به عنوان یک شبکه معیار برای ارزیابی الگوریتم‌های تخصیص ترافیک و مقایسه سرعت همگرایی آن‌ها استفاده شده است (برای نمونه، در [۱۲-۱۴]، [۱۸] و [۲۰-۲۸]). اطلاعات مربوط به کمان‌های شبکه‌ها و ماتریس تقاضا در مرجع [۲۹] موجود است. مشخصات

که در آن‌ها μ_n و ν_n برابرند با

$$\mu_n = -\frac{\bar{d}_{n-2}^T H_n d_n^{FW}}{\bar{d}_{n-2}^T H_n (s_{n-2}^{BFW} - s_{n-1}^{BFW})} \quad (16)$$

$$\nu_n = -\frac{\bar{d}_{n-1}^T H_n d_n^{FW}}{\bar{d}_{n-1}^T H_n \bar{d}_{n-1}} + \frac{\mu_n \alpha_{n-1}}{1 - \alpha_{n-1}} \quad (17)$$

و $\bar{d}_{n-1}^T = s_{n-1}^{BFW} - x_n$ و $\bar{d}_{n-1}^T = \alpha_{n-1} s_{n-1}^{BFW} + (1 - \alpha_{n-1}) s_{n-2}^{BFW} - x_{n-1}$ تعریف می‌شوند. بنابراین نیازی به ذخیره‌سازی جهات d_{n-1}^{BFW} و d_{n-2}^{BFW} وجود ندارد. در حالتی که مخرج کسرها (۱۶) و (۱۷) برابر صفر شود (وقتی رخ می‌دهد که $\alpha_{n-1} = 1$ یا $\alpha_{n-2} = 1$ باشد)، مشابه الگوریتم فرانک - ولف مزدوج، جهت جدید d_n^{BFW} برابر با d_n^{FW} قرار داده می‌شود (یا به عبارتی $\beta_n^0 = 1$ و $\beta_n^1 = \beta_n^2 = 0$). در ضمن، μ_n و ν_n نباید منفی باشند، زیرا در این صورت حداقل یکی از وزن‌های β_n^i ، $i = 1, 2, 3$ ، منفی می‌شود. هر چند در [۱۳] به این موضوع اشاره نشده است، در این مقاله μ_n و ν_n پس از محاسبه از پایین به صفر محدود می‌شوند. محاسبات الگوریتم فرانک - ولف دو مزدوج نسبت به فرانک - ولف مزدوج اندکی پیچیده‌تر است، اما تفاوت بسیار کمی در پیاده‌سازی آن‌ها وجود دارد. این الگوریتم نیز همچون پارتان به ذخیره‌سازی ۴ بردار با طول $|A|$ در حافظه نیاز دارد.

گام‌های الگوریتم فرانک - ولف دو مزدوج به شرح زیر است:

گام ۱ (تعیین جواب اولیه): گام ۱ الگوریتم فرانک - ولف را انجام دهید.

گام ۲ (شرط توقف): گام ۲ الگوریتم فرانک - ولف را انجام دهید.

گام ۳ (تعیین جهت فرانک - ولف): گام ۳ الگوریتم فرانک - ولف را

انجام دهید.

گام ۴ (تعیین جهت فرانک - ولف دو مزدوج): اگر $n \leq 2$ ، $\alpha_{n-1} = 1$ یا

$\alpha_{n-2} = 1$ قرار دهید $s_n^{BFW} = y_n^{FW}$ و گرنه μ_n و ν_n را طبق

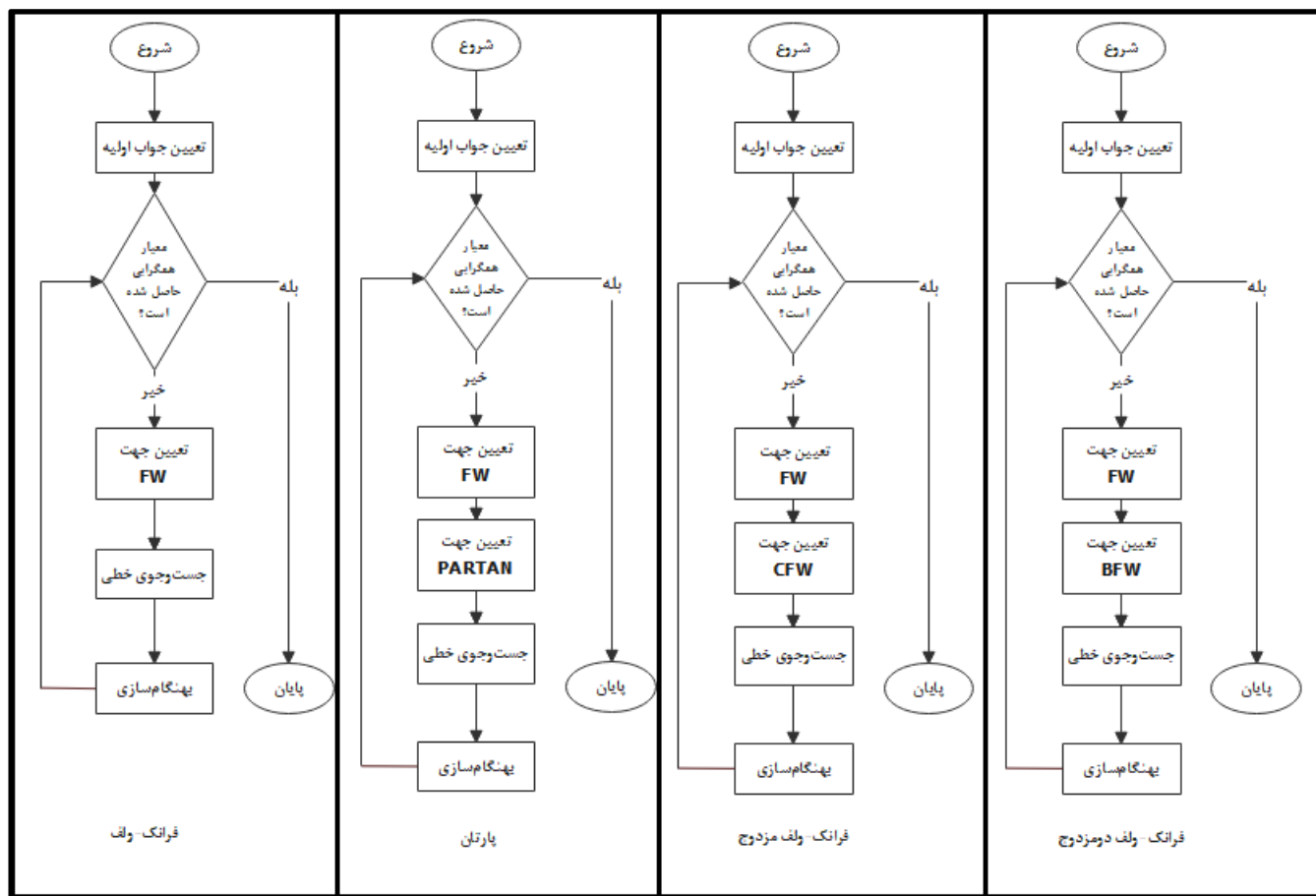
روابط (۱۶) و (۱۷) محاسبه کنید؛ اگر $\mu_n < 0$ قرار دهید $\mu_n = 0$ ؛

اگر $\nu_n < 0$ قرار دهید $\nu_n = 0$ ؛ β_n^0 ، β_n^1 و β_n^2 را طبق روابط

(۱۵) محاسبه کنید؛ و s_n^{BFW} را طبق رابطه (۱۲) محاسبه کنید. قرار دهید

$$d_n^{BFW} = s_n^{BFW} - x_n$$

گام ۵ (جست‌وجوی خطی): گام ۴ الگوریتم فرانک - ولف را با استفاده



شکل ۳. فلوجارت الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج

Fig. 3. Flowchart of FW, PARTAN, CFW and BFW Algorithms

و $disf = 0.25$ در نظر گرفته شده است. شکل ۴ تصویری از شبکه شیکاگو را نشان می‌دهد. شبکه سوفالز^۱ شبکه‌ای کوچک - مقیاس است که در این مقاله برای راستی‌آزمایی نتایج شبکه شیکاگو مورد استفاده قرار می‌گیرد. این شبکه نیز در ادبیات حمل‌ونقل کاملاً شناخته شده بوده، و البته نسخه‌های مختلفی از آن وجود دارد. در این مقاله، ماتریس تقاضا و توابع زمان سفر مورد استفاده توسط لبلانک و همکاران [۳۰] برای تعریف اطلاعات شبکه در نظر گرفته شده است. مشخصات کلی این شبکه در جدول ۱ ارائه شده است. توابع زمان سفر کمان‌های شبکه به صورت تابع BPR ارائه شده در رابطه (۱) است. لازم به ذکر است که الگوریتم فرانک - ولف مزدوج نیاز به انتخاب یک عدد کوچک $\delta > 0$ نیز دارد. در این مقاله از مقدار $\delta = 0.01$ برای هر دو شبکه شیکاگو و سوفالز استفاده می‌شود.

کلی این شبکه شامل تعداد گره‌ها، کمان‌ها، ناحیه‌ها، و جمع عناصر ماتریس تقاضا در جدول ۱ ارائه شده است. تابع زمان سفر کمان‌های این شبکه به صورت زیر تعریف شده است:

$$t_a(x_a) = tof \cdot to_a + disf \cdot dis_a + t_a^0 \left[1 + 0.15 \left(\frac{x_a}{q_a} \right)^4 \right] \quad (18)$$

که در آن to_a هزینه پرداختی بر حسب سنت برای عبور از کمان a ، ضریب تبدیل هزینه به زمان، dis_a طول کمان a بر حسب مایل و $disf$ نیز ضریب تبدیل طول به زمان است. تعاریف سایر پارامترها در تابع (۱۸) همانند قبل است. مقادیر ضرایب تبدیل در این تابع برابر $tof = 0.1$

جدول ۱. مشخصات شبکه‌های آزمایشی

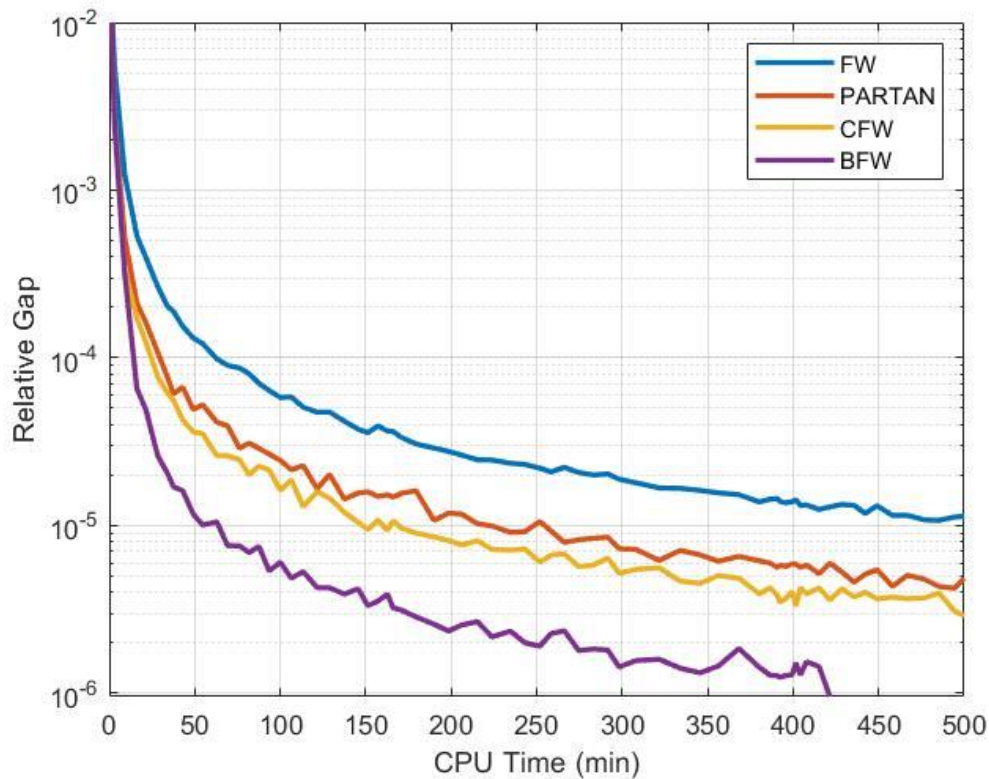
Table 1. Specifications of the test networks

شبکه	تعداد گره	تعداد کمان	تعداد ناحیه	تعداد مبدا - مقصد	جمع تقاضا
شیکاگو	۱۲,۹۸۲	۳۹,۰۱۸	۱,۷۹۰	۳,۱۳۶,۴۴۱	۱,۳۶۰,۴۲۸
سوفالز	۲۴	۷۶	۲۴	۵۵۲	۳۶۰/۶



شکل ۴. شبکه آزمایشی شیکاگو [۳۱]

Fig. 4. Chicago test network



شکل ۵. نرخ همگرایی الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج برای شبکه شیکاگو

Fig. 5. Convergence rate of FW, PARTAN, CFW and BFW algorithms for Chicago network

زمانی در حدود ۴۱۷ دقیقه به جوابی به خطای کمتر از 10^{-6} می‌رسد، در حالی که سه الگوریتم دیگر حتی پس از ۵۰۰ دقیقه نیز نمی‌توانند به چنین جوابی نزدیک شوند.

جزئیات محاسباتی چهار الگوریتم فوق برای شبکه شیکاگو در جدول ۲ گزارش شده است. همان‌طور که ملاحظه می‌شود، عملکرد این چهار الگوریتم برای یافتن جوابی با خطای 10^{-2} تقریباً یکسان است، در حالی که اختلاف عملکرد آن‌ها با کاهش خطای جواب مسئله از 10^{-3} تا 10^{-5} به تدریج زیاد می‌شود. الگوریتم فرانک - ولف دو مزدوج با انجام ۶۴۲ تکرار و در مدت زمان ۵۳/۲۷ دقیقه به دقت 10^{-5} می‌رسد، که از این نظر بهترین عملکرد را دارد، و پس از آن الگوریتم‌های فرانک - ولف مزدوج، پارتان و فرانک - ولف به ترتیب پس از ۱۶۵۹، ۲۲۴۹ و ۶۰۵۰ تکرار و در مدت زمان ۱۴۰/۲۴، ۱۸۷/۸۵ و ۵۰۴/۴۷ دقیقه به دقت مورد نظر رسیده‌اند. به عبارت دیگر، در مقایسه با الگوریتم فرانک - ولف، الگوریتم‌های فرانک - ولف دو

شکل ۵ نرخ همگرایی الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج را نسبت به زمان اجرای آن‌ها برای شبکه شیکاگو نشان می‌دهد. همان‌طور که مشاهده می‌شود، همگرایی الگوریتم فرانک - ولف نسبت به سایر الگوریتم‌های ذکر شده بسیار کندتر است، به طوری که اجرای آن برای شکاف نسبی کمتر از 10^{-5} بیش از ۵۰۰ دقیقه زمان می‌برد. در ضمن، الگوریتم پیشنهادی فرانک - ولف دو مزدوج بهترین همگرایی را در بین این الگوریتم‌های رقیب دارد، و الگوریتم‌های فرانک - ولف مزدوج و پارتان با عملکردهایی نزدیک به یکدیگر به ترتیب در رده‌های دوم و سوم قرار می‌گیرند. همان‌طور که در این شکل ملاحظه می‌شود، با کاهش خطای حل مسئله از 10^{-2} (دقت به دست آمده تا حدود ۲۰ تکرار اول) تا 10^{-5} (دقت لازم برای رسیدن به جواب‌های پایدار)، اختلاف زمان حل الگوریتم فرانک - ولف دو مزدوج نسبت به سایر الگوریتم‌ها افزایش می‌یابد. در ضمن، الگوریتم فرانک - ولف دو مزدوج پس از صرف

جدول ۲. نتایج الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج تا دقت‌های مختلف برای شبکه شیکاگو

Table 2. Results of FW, PARTAN, CFW and BFW algorithms to different precisions for Chicago network

الگوریتم	خطای جواب	تعداد تکرار	کل زمان سفر	تابع هدف بکمن	شکاف نسبی	زمان
فرانک - ولف	۱۰ ^{-۲}	۲۱	۳۸۹۱۲۵۱۸/۲۹۶۰۸	۳۱۰۰۰۰۲۲/۴۸۸۱۷۰۲۱۴۰	۰/۰۰۹۴۲۸۱	۱/۹۹
	۱۰ ^{-۳}	۱۱۷	۳۸۶۲۷۸۸۶/۶۱۲۵۴	۳۰۸۱۳۱۱۸/۴۰۶۶۱۷۱۸۷۰	۰/۰۰۰۹۲۱۹	۱۰/۱۴
	۱۰ ^{-۴}	۷۰۲	۳۸۶۱۰۹۱۱/۳۸۶۵۰	۳۰۷۹۴۶۶۴/۶۵۴۷۲۰۸۹۹	۰/۰۰۰۰۹۹۰۹	۵۹/۷۳
	۱۰ ^{-۵}	۶۰۵۰	۳۸۶۰۹۹۳۲/۹۲۴۱۸	۳۰۷۹۲۸۲۲/۴۳۹۶۶۴۸۸۲۰	۰/۰۰۰۰۰۹۷۱	۵۰۴/۴۷
پارتان	۱۰ ^{-۲}	۱۵	۳۸۸۲۶۹۹۷/۹۷۷۳۱	۳۰۹۶۰۲۶۵/۵۸۹۶۸۶۵۳۵۰	۰/۰۰۹۴۹۷۹	۱/۴۹
	۱۰ ^{-۳}	۵۹	۳۸۶۱۲۹۳۶/۶۴۱۶۴	۳۰۸۰۸۸۱۷/۸۹۲۳۵۵۰۷۳۰	۰/۰۰۰۹۷۱۶	۵/۱۷
	۱۰ ^{-۴}	۳۱۱	۳۸۶۰۹۱۴۷/۶۷۷۸۸	۳۰۷۹۴۰۸۰/۰۲۹۳۲۲۹۸۲۰	۰/۰۰۰۰۹۴۸	۲۶/۱۸
	۱۰ ^{-۵}	۲۲۴۹	۳۸۶۰۹۷۲۰/۷۵۹۲۶	۳۰۷۹۲۷۸۲/۷۹۱۱۸۶۰۵۳۰	۰/۰۰۰۰۰۹۸۴	۱۸۷/۸۵
فرانک - ولف مزدوج	۱۰ ^{-۲}	۱۶	۳۸۷۸۱۰۹۳/۲۹۲۱۷	۳۰۹۲۲۰۱۴/۲۸۶۲۶۳۴۵۵۰	۰/۰۰۹۵۲۹۳۹	۱/۶۰
	۱۰ ^{-۳}	۶۲	۳۸۶۰۱۸۲۵/۷۵۷۰۸	۳۰۸۰۰۸۳۴/۸۰۹۸۱۰۰۳۵۰	۰/۰۰۰۹۱۸۸۶	۵/۵۲
	۱۰ ^{-۴}	۲۵۵	۳۸۶۰۸۰۲۶/۰۵۸۱۷	۳۰۷۹۳۴۵۲/۹۸۹۴۲۱۲۰۷۰	۰/۰۰۰۰۹۸۲۵	۲۱/۷۳
	۱۰ ^{-۵}	۱۶۵۹	۳۸۶۰۹۵۳۹/۱۲۶۰۵	۳۰۷۹۲۷۰۹/۰۳۰۷۴۸۰۷۳۰	۰/۰۰۰۰۰۹۷۸	۱۴۰/۲۴
فرانک - ولف دو مزدوج	۱۰ ^{-۲}	۱۵	۳۸۷۷۷۱۶۵/۱۹۴۵۷	۳۰۹۱۷۱۹۸/۲۲۲۷۵۹۹۵۸۰	۰/۰۰۹۶۲۵۱	۱/۴۸
	۱۰ ^{-۳}	۴۸	۳۸۵۹۷۱۲۴/۹۰۷۴۰	۳۰۸۰۱۶۸۸/۳۱۳۷۸۲۸۷۱۰	۰/۰۰۰۹۳۶۸	۴/۲۴
	۱۰ ^{-۴}	۱۶۷	۳۸۶۰۵۶۴۶/۳۴۳۴۱	۳۰۷۹۲۹۰۹/۲۸۵۷۶۴۳۸۹۰	۰/۰۰۰۰۹۶۱	۱۴/۱۱
	۱۰ ^{-۵}	۶۴۲	۳۸۶۰۹۲۵۰/۸۱۵۸۲	۳۰۷۹۲۶۴۲/۱۰۳۰۳۲۵۵۲۰	۰/۰۰۰۰۰۹۵۰	۵۳/۲۷
	۱۰ ^{-۶}	۵۰۵۲	۳۸۶۰۹۷۶۶/۸۶۰۰۳	۳۰۷۹۲۶۱۵/۰۲۵۴۹۹۳۵۱۰	۰/۰۰۰۰۰۰۹۵	۴۱۶/۹۴

زیرا فرانک - ولف دو مزدوج جهت جست‌وجو را به جای جهت قبلی نسبت به دو جهت قبلی مزدوج می‌کند، بدون آنکه محاسبات چندان بیشتری انجام داده باشد. در مجموع، کارایی الگوریتم فرانک - ولف دو مزدوج در مقایسه با الگوریتم فرانک - ولف (با توجه به کاهش ۸۹ درصدی زمان حل) بسیار چشمگیر است. جدول ۲ نتایج الگوریتم فرانک - ولف دو مزدوج را برای شکاف نسبی کمتر از ۱۰^{-۶} نیز ارائه داده است، دقتی که سه الگوریتم دیگر نتوانسته‌اند حتی به نزدیک آن برسند. شایان ذکر است که تعداد تکرار ارائه شده در جدول ۳ برای هر الگوریتم معادل با تعداد تخصیص‌های همه یا هیچ انجام شده برای شبکه شیکاگو توسط آن الگوریتم است.

مزدوج، فرانک - ولف مزدوج و پارتان سرعت همگرایی به شکاف نسبی کمتر از ۱۰^{-۵} را به ترتیب در حدود ۸۹، ۷۲ و ۶۳ درصد افزایش داده‌اند، که دلیل آن استفاده از تکنیک‌های مزدوج‌سازی جهت جست‌وجو در این سه الگوریتم است. الگوریتم فرانک - ولف مزدوج در مقایسه با الگوریتم پارتان سرعت همگرایی به شکاف نسبی ۱۰^{-۵} در حدود ۲۵ درصد افزایش داده است، زیرا در اولی مزدوج‌سازی جهت جست‌وجو نسبت به تکرار قبل با چند فرمول ساده انجام می‌شود، ولی در دومی مزدوج‌سازی جهت جست‌وجو به صورت تقریبی انجام شده و به علاوه یک جست‌وجوی خطی اضافی نیز در هر تکرار انجام می‌شود. همچنین، الگوریتم فرانک - ولف دو مزدوج حدود ۶۲ درصد سریع‌تر از الگوریتم فرانک - ولف مزدوج به همان دقت ۱۰^{-۵} رسیده است،

جدول ۳. حافظه مورد نیاز الگوریتم‌های مورد مطالعه برای شبکه شیکاگو (مگابایت) *

Table 3. Required memory of the studied algorithms for Chicago network (MB)

فرانک - ولف	پارتان	فرانک - ولف مزدوج	فرانک - ولف دو مزدوج
۰/۳۱۲	۰/۶۲۴	۰/۴۶۸	۰/۶۲۴

* شامل حافظه مصرفی برای ذخیره سازی اطلاعات شبکه و تقاضا نمی‌باشد.

جدول ۴. تعداد تکرارهای نسبی شده الگوریتم‌های مورد مطالعه تا دقت 10^{-5} برای شبکه‌های شیکاگو و سوفالز *

Table 4. Relative number of iterations of the studied algorithms to the precision of 10^{-5} for Chicago and Sioux Falls networks

شبکه	فرانک - ولف	پارتان	فرانک - ولف مزدوج	فرانک - ولف دو مزدوج
شیکاگو	۱	۰/۳۷	۰/۲۷	۰/۱۱
سوفالز	۱	۰/۳۵	۰/۱۸	۰/۰۲

* برای شبکه سوفالز، الگوریتم فرانک - ولف در ۱۰۲۱۹ تکرار به دقت 10^{-5} و تابع هدف بکمن ۴۲/۳۱۳۸۱۸۶۴ می‌رسد.

یافته‌اند. با توجه به نتایج ارائه شده در ۲، ملاحظه می‌شود که این درصدهای کاهش، معادل با درصدهای کاهش زمان حل این الگوریتم‌ها نسبت به الگوریتم فرانک - ولف است. به عبارت دیگر، می‌توان گفت که زمان حل الگوریتم‌های جهات مزدوج متناسب با تعداد تکرار آن‌ها است.

نتایج ارائه شده در جدول ۴ برای شبکه سوفالز نشان می‌دهد که تعداد تکرارهای الگوریتم‌های فرانک - ولف دو مزدوج، فرانک - ولف مزدوج و پارتان، در مقایسه با الگوریتم فرانک - ولف، به ترتیب ۹۲، ۸۲ و ۶۵ درصد کاهش یافته‌اند. ملاحظه می‌شود که این درصدهای کاهش تکرار، نسبت به درصدهای نظیر برای شبکه شیکاگو، افزایش یافته‌اند (خصوصاً، تکرارهای الگوریتم فرانک - ولف مزدوج نزدیک به ۱۰ درصد کاهش یافته است). بر اساس این نتایج محدود، می‌توان حدس زد که سرعت همگرایی الگوریتم‌های جهات مزدوج نسبت به الگوریتم فرانک - ولف به مراتب بیشتر است، و نیز این افزایش سرعت با کاهش ابعاد شبکه افزایش می‌یابد.

جدول ۳ حافظه مورد نیاز الگوریتم‌های رقیب را برای شبکه شیکاگو نشان می‌دهد (حافظه مصرفی برای ذخیره‌سازی اطلاعات شبکه و تقاضا در محاسبات در نظر گرفته نشده‌اند). همان‌طور که ملاحظه می‌شود، حافظه مورد نیاز این الگوریتم‌ها در مجموع (در مقایسه با ۲ گیگابایت حافظه قابل دسترس RAM) بسیار ناچیز و کمتر از ۰/۰۰۱ گیگابایت است. شایان ذکر است که یک الگوریتم بر پایه مسیر، در شرایط مشابه، فقط نزدیک به ۱ گیگابایت حافظه برای ذخیره‌سازی مسیرهای فعال نیاز دارد.

جدول ۴ نتایج یک مقایسه مستقیم بین الگوریتم‌های رقیب را برحسب تعداد تکرار تا رسیدن به دقت 10^{-5} برای شبکه‌های شیکاگو و سوفالز نشان می‌دهد. در این جدول، تعداد تکرارهای الگوریتم‌های پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج نسبت به تعداد تکرار الگوریتم فرانک - ولف نرمال شده‌اند. نتایج شبکه شیکاگو نشان می‌دهند که تعداد تکرارهای الگوریتم‌های فرانک - ولف دو مزدوج، فرانک - ولف مزدوج و پارتان، در مقایسه با الگوریتم فرانک - ولف، به ترتیب ۸۹، ۷۳ و ۶۳ درصد کاهش

۵- نتیجه‌گیری

مسئله تخصیص ترافیک در شبکه‌های حمل و نقل طبق قانون تعادل استفاده کننده به صورت یک مسئله بهینه‌سازی محدب به نام مدل بکمن فرمول‌بندی می‌شود. الگوریتم‌های موجود برای حل مسئله تخصیص ترافیک به سه نوع الگوریتم‌های بر پایه کمان، بر پایه مسیر و بر پایه مبدا تقسیم می‌شوند، که در این میان الگوریتم‌های بر پایه کمان به دلیل حافظه مصرفی کم‌تر نسبت به دو نوع دیگر کاربرد بیشتری دارند. الگوریتم فرانک - ولف به دلیل سادگی و نیز سرعت همگرایی آن در تکرارهای اولیه (شکاف نسبی حدود 10^{-2} تا 10^{-3}) هنوز جزو محبوب‌ترین الگوریتم‌های تخصیص ترافیک محسوب می‌شود. ولی، همگرایی این الگوریتم برای رسیدن به جواب‌های پایدار (شکاف نسبی کمتر از 10^{-5}) بسیار کند می‌شود، که علت آن متعامد شدن جهات جست‌وجو با گرادیان تابع هدف و در نتیجه حرکت زیگزآگی به سمت جواب بهینه است. به همین دلیل تلاش‌های زیادی برای اصلاح الگوریتم فرانک - ولف انجام شده است، که مؤثرترین آن‌ها الگوریتم جهات مزدوج شامل پارتان، فرانک- ولف مزدوج و فرانک - ولف مزدوج دو مزدوج می‌باشند. در این مقاله مقایسه‌ای مستقیم بین الگوریتم‌های فرانک - ولف، پارتان، فرانک - ولف مزدوج و فرانک - ولف دو مزدوج از نظر سرعت همگرایی به جواب‌های پایدار روی شبکه بزرگ مقیاس شیکاگو در شرایط نرم افزاری و سخت افزاری یکسان انجام شد. نتایج نشان دادند که سه الگوریتم فرانک - ولف دو مزدوج، فرانک - ولف مزدوج و پارتان سرعت همگرایی به شکاف نسبی کمتر از 10^{-5} را به ترتیب در حدود ۸۹، ۷۲ و ۶۳ درصد افزایش می‌دهند. به عبارت دیگر، مزدوج‌سازی جهات جست‌وجو (حتی به صورت تقریبی) توانسته است کارایی الگوریتم پایه فرانک - ولف را بیش از ۶۰ درصد افزایش دهد. کارایی الگوریتم فرانک - ولف دو مزدوج در بین الگوریتم‌های مورد بررسی به مراتب بیشتر است، که دلیل آن مزدوج‌سازی جهات جست‌وجو در هر تکرار نسبت به دو جهت قبلی است. الگوریتم فرانک - ولف مزدوج کندتر از فرانک - ولف دو مزدوج به جوابی پایدار رسید، زیرا در آن مزدوج‌سازی فقط نسبت به جهت قبلی انجام می‌شود. الگوریتم پارتان از نظر کارایی بین دو الگوریتم فرانک - ولف مزدوج و فرانک - ولف قرار گرفت، زیرا در آن مزدوج‌سازی به طور تقریبی انجام شده و به علاوه نیازمند یک جست‌وجوی خطی اضافی در هر تکرار است. در ضمن، در بین سه الگوریتم جهات مزدوج، فقط الگوریتم فرانک - ولف دو مزدوج توانست جوابی با شکاف نسبی کمتر از 10^{-6} را در زمانی کمتر از ۵۰۰ دقیقه بیابد. تفاوت این سه الگوریتم با الگوریتم پایه فرانک - ولف از نظر حافظه مصرفی

تنها در حد $0/3$ مگابایت بود. در ضمن، مقایسه نتایج به دست آمده برای شبکه کوچک مقیاس سوفالز با نتایج شبکه شیکاگو نشان داد که کارایی الگوریتم‌های جهات مزدوج نسبت به الگوریتم فرانک - ولف با کاهش اندازه شبکه افزایش می‌یابد.

منابع

- [1] Y. Sheffi, Urban transportation networks, Prentice-Hall, Englewood Cliffs, NJ (1985).
- [2] M.J. Beckmann, C.B. McGuire, C.B. Winsten, Studies in the economics of transportation, (1955).
- [3] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval research logistics quarterly, 3(1-2) (1956) 95-110.
- [4] L.J. LeBlanc, E.K. Morlok, W.P. Pierskalla, An efficient approach to solving the road network equilibrium traffic assignment problem, Transportation research, 9(5) (1975) 309-318.
- [5] M. Patriksson, The traffic assignment problem—models and methods. VSP, Utrecht, Definition: where with the projection operator given by, (1994).
- [6] D.G. Luenberger, Y. Ye, Linear and nonlinear programming, 3rd edition, Springer (2008).
- [7] L.J. LeBlanc, R.V. Helgason, D.E. Boyce, Improved efficiency of the Frank-Wolfe algorithm for convex network programs, Transportation Science, 19(4) (1985) 445-462.
- [8] M. Florian, J. Guálat, H. Spiess, An efficient implementation of the “Partan” variant of the linear approximation method for the network equilibrium problem, Networks, 17(3) (1987) 319-339.
- [9] Y. Arezki, D. Van Vliet, A full analytical implementation of the PARTAN/Frank-Wolfe algorithm for equilibrium assignment, Transportation Science, 24(1) (1990) 58-62.
- [10] M. Fukushima, A modified Frank-Wolfe algorithm for solving the traffic assignment problem, Transportation Research Part B: Methodological, 18(2) (1984) 169-177.
- [11] D. Hearn, S. Lawphongpanich, J.A. Ventura, Finiteness

- Algorithm for Traffic Assignment, *Transportation Research Record*, 2672(48) (2018) 36-44.
- [22] G. Gentile, Linear User Cost Equilibrium: a new algorithm for traffic assignment, *Transportmetrica A*, (2012) 15-54.
- [23] A. Babazadeh, B. Javani, Comparison Between Path-based Algorithms for Traffic Assignment Problem, in: *Transportation Research Board 91st Annual Meeting*, (2012)
- [24] H. Bar-Gera, Origin-based algorithm for the traffic assignment problem, *Transportation Science*, 36(4) (2002) 398-417.
- [25] R.B. Dial, A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration, *Transportation Research Part B: Methodological*, 40(10) (2006) 917-936.
- [26] H. Bar-Gera, Traffic assignment by paired alternative segments, *Transportation Research Part B: Methodological*, 44(8-9) (2010) 1022-1046.
- [27] J. Xie, and C. Xie, New insights and improvements of using paired alternative segments for traffic assignment. *Transportation Research Part B: Methodological*, 93 (2016) 406-424.
- [28] D. Boyce, B. Ralevic-Dekic, H. Bar-Gera, Convergence of traffic assignments: how much is enough?, *Journal of Transportation Engineering*, 130(1) (2004) 49-55.
- [29] *Transportation networks for research*. <https://github.com/bstabler/TransportationNetworks>. Accessed July, 20, 2019.
- [30] L.J. LeBlanc, E.K. Morlok, W.P. Pierskalla, An efficient approach to solving the road network equilibrium traffic assignment problem, *Transportation Research*, 9(5) (1975) 309-318.
- [31] M. Florian, C. D. Morosan, on uniqueness and proportionality in multi-class equilibrium assignment, *Transportation Research Part B: Methodological*, 70 (2014) 173-185.
- in restricted simplicial decomposition, *Operations Research Letters*, 4(3) (1985) 125-130.
- [12] D.-H. Lee, Y. Nie, Accelerating strategies and computational studies of the Frank-Wolfe algorithm for the traffic assignment problem, *Transportation Research Record*, 1771(1) (2001) 97-105.
- [13] M. Mitradjieva, P.O. Lindberg, The stiff is moving - conjugate direction Frank-Wolfe Methods with applications to traffic assignment, *Transportation Science*, 47(2) (2013) 280-293.
- [14] J. Holmgren, P.O. Lindberg, Upright stiff: subproblem updating in the FW method for traffic assignment, *EURO Journal on Transportation and Logistics*, 3(3-4) (2014) 205-225.
- [15] W.B. Powell, Y. Sheffi, The convergence of equilibrium algorithms with predetermined step sizes, *Transportation Science*, 16(1) (1982) 45-55.
- [16] A. Weintraub, C. Ortiz, J. González, Accelerating convergence of the Frank-Wolfe algorithm, *Transportation Research Part B: Methodological*, 19(2) (1985) 113-122.
- [17] A. Chen, Effects of Flow Update Strategies on Implementation of the Frank-Wolfe Algorithm for the Traffic Assignment Problem, *Transportation research record*, 1771(1) (2001) 132-139.
- [18] M. Florian, I. Constantin, D. Florian, A new look at projected gradient method for equilibrium assignment, *Transportation Research Record*, 2090(1) (2009) 10-16.
- [19] A. Kumar and S. Peeta, Slope-based path shift propensity algorithm for the static traffic assignment problem, *International Journal for Traffic and Transport Engineering*, 4(3) (2014) 297-319.
- [20] D. Di Lorenzo, A. Galligari, and M. Sciandrone, A convergent and efficient decomposition method for the traffic assignment problem, *Computational optimization and applications*, 60(1) (2015) 151-170.
- [21] J. Xie, Y. Nie, and X. Liu, A Greedy Path-Based

چگونه به این مقاله ارجاع دهیم

V. Karimi, A. Babazadeh, *Comparison of Convergence Speed among Conjugate Directions Traffic Assignment Algorithms*, *Amirkabir J. Civil Eng.*, 54(6) (2022) 2219-2236.

DOI: [10.22060/ceej.2021.19708.7244](https://doi.org/10.22060/ceej.2021.19708.7244)

