

Traffic Assignment Problem with Strict Capacity Constraints

By
Ghazaleh Azimi

Thesis:
Presented to the Faculty of the Civil Engineering
University of Tehran

in Fulfillment of the
Requirements for the Degree of
Master of Science

The University of Tehran
September 2017

Abstract

To improve the performance of the transportation network operation, traffic assignment problems require more reliable results. Disregard the solution method, one factor that leads to more accurate results in traffic assignment problems is considering the link capacity constraints. In traffic assignment problems, techniques such as the Frank-Wolfe (FW) algorithm has extensive applications, but not considering link capacity in these methods results in inaccurate outputs. In every iteration of FW, move direction is obtained by solving the minimum cost flow problem for the network without considering the capacity constraints, which is named the all or nothing assignment algorithm. On the contrary, there is another algorithm called the Out of kilter (OFK) algorithm that solves the traffic assignment problem by minimizing the cost flow problem in networks with link capacity constraints. In this dissertation, the movement direction of the FW algorithm is obtained by the OFK algorithm. Therefore, in every iteration, link flows are lower than or equal to their capacities. In this method, we assume that the network has a circulation form, meaning that all origins are pure origin, and all destinations are the pure destination. Moreover, the results of the proposed algorithm for the Hearn and Sioux Falls transportation networks are presented and compared to the condition that link capacities are considered infinite.

Keywords: Traffic Assignment, Capacity Constraint, Frank-Wolf Algorithm, Out of Kilter Algorithm

Abstract

To improve the performance of the transportation network operation, traffic assignment problems require more reliable results. Disregard the solution method, one factor that leads to more accurate results in traffic assignment problems is considering the link capacity constraints. In traffic assignment problems, techniques such as the Frank-Wolfe (FW) algorithm has extensive applications, but not considering link capacity in these methods results in inaccurate outputs. In every iteration of FW, move direction is obtained by solving the minimum cost flow problem for the network without considering the capacity constraints, which is named the all or nothing assignment algorithm. On the contrary, there is another algorithm named the Out of kilter (OFK) algorithm that solves the traffic assignment problem by minimizing the cost flow problem in networks with link capacity constraints. In this dissertation, the movement direction of the FW algorithm is obtained by the OFK algorithm. Therefore, in every iteration, link flows are lower than or equal to their capacities. In this method, we assume that the network has a circulation form, meaning that all origins are pure origin, and all destinations are the pure destination. Moreover, the results of the proposed algorithm for the Hearn and Sioux Falls transportation networks are presented and compared to the condition that link capacities are considered infinite.

Keywords: Traffic Assignment, Capacity Constraint, Frank-Wolf Algorithm, Out of Kilter Algorithm

۱	فصل اول مقدمه.....
۲	۱-۱- مقدمه.....
۳	۲-۱- اهمیت مسأله.....
۳	۳-۱- مروری بر فصول.....
۲	فصل دوم.....
۲	ادبیات مساله تخصیص.....
۳	۱-۲- مقدمه.....
۳	۲-۲- مروری بر مدل های تخصیص ترافیک.....
۹	۳-۲- مدل بهینه سازی تخصیص ترافیک.....
۱۰	۳-۲-۱- روش های نزول.....
۱۱	۳-۲-۲- الگوریتم فرانک-ولف.....
۱۶	۳-۲-۲- روش میانگین گیری متوالی.....
۱۷	۴-۲- مدل تخصیص ترافیک با ظرفیت محدود.....
۱۸	۵-۲- الگوریتم شهپر و همکاران.....
۲۳	فصل سوم مسئله جریان با هزینه مینیم در شبکه.....
۲۴	۱-۳- مقدمه.....
۲۵	۲-۳- مسئله جریان با هزینه مینیم.....
۲۶	۳-۳- الگوریتم بلمن-فورد.....
۲۷	۴-۳- الگوریتم دایجسترا.....
۳۲	۵-۳- الگوریتم درخت heap.....
۳۲	۳-۵-۱- هرم ماکزیمم (max-heap).....
۳۳	۳-۵-۲- هرم مینیمم (min-heap).....
۳۹	۳-۶- الگوریتم Out of Kilter.....
۵۴	۳-۷- مسئله تخصیص همه یا هیچ.....
۵۴	۳-۸- مقایسه الگوریتم های بل و دایجسترا.....
۵۶	فصل چهارم اثر اعمال محدودیت ظرفیت کمان بر مسئله تخصیص ترافیک.....
۵۷	۴-۱- مقدمه.....
۵۷	۴-۲- شبکه های حمل و نقل.....
۵۷	۴-۲-۱- شبکه هرن.....
۵۹	۴-۲-۲- شبکه سوفالز.....
۶۵	۴-۲-۳- شبکه آناهیم.....

۳-۴- مقایسه تحلیلی نتایج مسئله تخصیص ترافیک با محدودیت ظرفیت و بدون محدودیت ظرفیت.....	۶۵
۳-۴-۱- ظرفیت و عوامل تاثیر گذار بر آن.....	۶۶
۳-۴-۲- روش برچسب گذاری سه گانه.....	۶۷
۳-۴-۳- ساختار شبکه.....	۶۷
۳-۴-۴- استفاده از اطلاعات پیشین.....	۶۸
۳-۴-۵- روند پیمایش و برچسب گذاری.....	۶۹
۳-۴-۶- اعمال تغییرات جریان.....	۷۰
۳-۴-۷- اعمال تغییرات هزینه تقلیل یافته.....	۷۱
۳-۴-۸- شرط چرخشی بودن شبکه.....	۷۱
۳-۴-۹- نحوه انتخاب کمان های Out of Kilter.....	۷۲
۳-۴-۱۰- تحلیل و مقایسه نتایج.....	۷۲

۵ فصل پنجم خلاصه و پیشنهادات..... ۱۰۰

۵-۱- مقدمه.....	۱۰۱
۵-۲- خلاصه.....	۱۰۱
۵-۳- پیشنهادات.....	۱۰۱

پیوست..... ۱۰۲

منابع و مراجع..... ۱۳۰

فهرست اشکال

شکل ۳-۱- شبکه ۴ گره ای الگوریتم دایجسترا.....	۲۸
شکل ۳-۲- درخت هیپ ماکزیمم.....	۳۲
شکل ۳-۲- درخت هیپ مینیمم.....	۳۳
شکل ۳-۴- شبکه ۴ گره ای الگوریتم هیپ.....	۳۵
شکل ۳-۵- درخت مرحله اول الگوریتم هیپ مینیمم.....	۳۵
شکل ۳-۶- درخت مرحله دوم الگوریتم هیپ مینیمم.....	۳۶
شکل ۳-۷- درخت مرحله سوم الگوریتم هیپ مینیمم.....	۳۷
شکل ۳-۸- شبکه مطلوب الگوریتم OFK.....	۴۱
شکل ۳-۹- شبکه ۴ گره ای الگوریتم OFK.....	۴۲
شکل ۳-۱۰- درخت مرحله اول الگوریتم OFK.....	۴۳
شکل ۳-۱۱- درخت مرحله دوم الگوریتم OFK.....	۴۵
شکل ۳-۱۲- درخت مرحله سوم الگوریتم OFK.....	۴۵
شکل ۳-۱۳- درخت مرحله چهارم الگوریتم OFK.....	۴۶
شکل ۳-۱۴- مقایسه جریان در کمان ها برای الگوریتم های FW و OFK.....	۴۸
شکل ۳-۱۵- مقایسه جریان در کمان ها برای الگوریتم های FW و OFK.....	۵۱
شکل ۳-۱۶- مقایسه تعداد کمان های دارای جریان برای الگوریتم های FW و OFK.....	۵۲
شکل ۴-۱- نمایی از شبکه هرن.....	۵۷
شکل ۴-۲- نمایی از شبکه سوفالز.....	۵۹
شکل ۴-۳- نمودار رابطه بین جریان و زمان سفر.....	۶۵
شکل ۴-۴- مراحل الگوریتم OFK.....	۷۲
شکل ۴-۵- مقایسه جریان در کمان ها برای الگوریتم های FW و FW-OFK-INFINTECAP.....	۷۸
شکل ۴-۶- مقایسه هزینه کمان ها برای الگوریتم های FW و FW-OFK-INFINTECAP.....	۷۹
شکل ۴-۷- مقایسه جریان در کمان ها برای الگوریتم FW-OFK-CAPAITATED و FW-OFK-INFINTECAP.....	۸۰
شکل ۴-۸- مقایسه هزینه سفر کمان ها برای دو الگوریتم FW-OFK-INFINTECAP و FW-OFK-CAPACITATED.....	۸۰
شکل ۴-۹- مقایسه نسبت جریان به ظرفیت برای الگوریتم های FW و FW-OFK-INFINTECAP.....	۸۱
شکل ۴-۱۰- مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت.....	۸۲

شکل ۴-۱۱- مقایسه تغییرات هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت.	۸۲
شکل ۴-۱۲- مقایسه جریان بدست آمده از دو الگوریتم FW و FW-OFK-INFINITECAP	۸۶
شکل ۴-۱۳- مقایسه هزینه سفر بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP	۸۷
شکل ۴-۱۴- مقایسه جریان های بدست آمده از دو الگوریتم FW-OFK-CAPACITATED و FW-OFK- INFINITECAP	۸۸
شکل ۴-۱۵- مقایسه هزینه های بدست آمده از دو الگوریتم FW-OFK-CAPACITATED و FW-OFK- INFINITECAP	۸۸
شکل ۴-۱۶- مقایسه مقادیر جریان بر ظرفیت برای الگوریتم های FW ، FW-OFK-CAPACITATED و FW-OFK- INFINITECAP	۸۹
شکل ۴-۱۷- تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK-INFINITECAP	۹۰
شکل ۴-۱۸- تغییرات هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK-INFINITECAP	۹۱
شکل ۴-۱۹- مقایسه جریان های بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP	۹۴
شکل ۴-۲۰- مقایسه هزینه های بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP	۹۵
شکل ۴-۲۱- مقایسه بین جریان های بدست آمده از الگوریتم های FW-OFK-CAPACITATED و FW-OFK- INFINITECAP	۹۶
شکل ۴-۲۲- مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK-INFINITECAP	۹۶
شکل ۴-۲۳- مقایسه مقادیر جریان بر ظرفیت برای الگوریتم های FW،FW-OFK-CAPACITATED و FW-OFK- INFINITECAP	۹۷
شکل ۴-۲۴- مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK-INFINITECAP	۹۸
شکل ۴-۲۵- مقایسه تغییرات هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK-INFINITECAP	۹۸

فهرست جداول

صفحه

جدول ۱- الگوریتم های کوتاهترین مسیر.....	۲۴
جدول ۲- کوتاه ترین مسیر در مرحله ۱.....	۲۹
جدول ۳- کوتاه ترین مسیر.....	۲۹
جدول ۴- کوتاه ترین مسیر.....	۳۰
جدول ۵- کوتاه ترین مسیر.....	۳۰
جدول ۶- کوتاه ترین مسیر.....	۳۱
جدول ۷- کوتاه ترین مسیر.....	۳۱
جدول ۸- مینیمم مسیر گره های درخت هیپ ماکزیمم.....	۳۳
جدول ۹- اعداد کیلتر.....	۴۰
جدول ۱۰- جهت مطلوب تغییر جریان.....	۴۰
جدول ۱۱- جهت مطلوب تغییر هزینه تقلیل یافتن.....	۴۱
جدول ۱۲- اطلاعات کمان های شبکه.....	۴۳
جدول ۱۳- اطلاعات گره های شبکه.....	۴۴
جدول ۱۴- جریان های بروز شده.....	۴۵
جدول ۱۵- پتانسیل های بروز شده.....	۴۵
جدول ۱۶- پتانسیل های بروز شده.....	۴۶
جدول ۱۷- پتانسیل های بروز شده.....	۴۷
جدول ۱۸- نتایج بدست آمده از الگوریتم های FW و OFK.....	۴۸
جدول ۱۹- مقایسه هزینه کل بدست آمده از دو الگوریتم.....	۴۹
جدول ۲۰- اطلاعات شبکه ۱۲ گره ای.....	۵۰
جدول ۲۱- عرضه و تقاضا در مبدأ- مقصد شبکه.....	۵۱
جدول ۲۲- مقایسه نتایج بدست آمده از الگوریتم های FW و OFK.....	۵۱
جدول ۲۳- مقایسه هزینه کل بدست آمده از دو الگوریتم.....	۵۴
جدول ۲۴- مقایسه زمان حل الگوریتم های بل و دایجسترا برای شبکه شیکاگو.....	۵۵
جدول ۲۵- عرضه و تقاضا در مبدأ- مقصد شبکه هرن.....	۵۸
جدول ۲۶- ظرفیت بالا و پایین و زمان سفر آزاد کمان های شبکه هرن.....	۵۸
جدول ۲۷- عرضه و تقاضا در مبدأ- مقصد های شبکه سوفالز.....	۶۱
جدول ۲۸- ظرفیت و زمان سفر آزاد کمان های شبکه سوفالز.....	۶۲
جدول ۲۹- جریان های بدست آمده از سه الگوریتم FW،FW-OFK،FW-OFK-INFINITECAP.....	۷۴
CAPACITATED برای شبکه هرن.....	۷۴

جدول ۳۰- هزینه سفرهای بدست آمده از بدست آمده از سه الگوریتم FW-FW-OFK-INFINITECAP,FW	۷۵
OFK-CAPACITATED برای شبکه هرن.....	
جدول ۳۱- نتایج بدست آمده برای شبکه هرن.....	۷۷
جدول ۳۲- مقایسه نتایج حاصل از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED در تکرار های مختلف برای شبکه هرن.....	۷۸
جدول ۳۳- نتایج بدست آمده برای شبکه سوفالز تک مبدایی.....	۸۵
جدول ۳۴- مقایسه نتایج حاصل از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED در تکرار های مختلف برای شبکه سوفالز تک مبدایی.....	۸۶
جدول ۳۵- نتایج بدست آمده برای شبکه سوفالز دو مبدایی.....	۹۳
جدول ۳۶- مقایسه نتایج حاصل از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED در تکرار های مختلف برای شبکه سوفالز دو مبدایی.....	۹۴
جدول ۳۷- جریان های بدست آمده از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED برای شبکه سوفالز تک مبدایی.....	۱۰۳
جدول ۳۷- هزینه سفرهای بدست آمده از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED برای شبکه سوفالز تک مبدایی.....	۱۰۷
جدول ۳۹- جریان های بدست آمده از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED برای شبکه سوفالز دو مبدایی.....	۱۱۱
جدول ۴۰- هزینه سفر های بدست آمده از سه الگوریتم FW-OFK-FW-OFK-INFINITECAP,FW	
CAPACITATED برای شبکه سوفالز دو مبدایی.....	۱۱۵
جدول ۴۱- مشخصات کمان های شبکه سوفالز تک مبدایی.....	۱۱۹
جدول ۴۲- عرضه و تقاضا در مبدا- مقصدهای شبکه سوفالز تک مبدایی.....	۱۲۳
جدول ۴۳- مشخصات کمان های شبکه سوفالز دو مبدایی.....	۱۲۴
جدول ۴۴- عرضه و تقاضا در مبدا اول شبکه سوفالز دو مبدایی.....	۱۲۸
جدول ۴۵- عرضه و تقاضا در مبدا دوم شبکه سوفالز دو مبدایی.....	۱۲۹

چکیده

کاربرد وسیع مدل تخصیص ترافیک در ارزیابی و انتخاب راه کارهای مناسب برای بهبود عملکرد شبکه حمل و نقل نیازمند واقعی تر شدن هرچه بیشتر نتایج حاصل از آن است. یکی از عوامل واقعی تر شدن نتایج این مدل، مستقل از روش حل آن، در نظر گرفتن محدودیت ظرفیت کمان ها است. در مسائل تخصیص ترافیک، روش هایی مانند الگوریتم فرانک-ولف کارایی زیادی دارند ولی در نظر نگرفتن ظرفیت برای کمان ها در این روش ها باعث غیر دقیق شدن نتایج بدست آمده می شود. در این نوشتار الگوریتم فرانک-ولف با اعمال محدودیت ظرفیت اصلاح شده است با این تفاوت که جهت حرکت توسط الگوریتم Out Of Kilter به دست می آید. همچنین نتایج کاربرد این روش برای دو شبکه با زمان سفر ثابت و شبکه هرن و سوفالز با زمان سفر وابسته به جریان از نقطه نظر جریان در کمان و هزینه های سفر ارائه شده است

کلمات کلیدی: تخصیص ترافیک، محدودیت ظرفیت، الگوریتم فرانک-ولف

۱ فصل اول

مقدمه

۱-۱- مقدمه

سیستم حمل و نقل کالا و مسافر به عنوان یکی از زیرساخت های مهم شهری نقش بسزایی در کیفیت زندگی انسان ها دارد. از طرفی، افزایش روز افزون تقاضای سفر با وسایل نقلیه باعث افزایش مسایل ناشی از حمل و نقل از جمله آلودگی و تصادفات خواهد شد. از این رو انجام مطالعات حمل و نقل مستمر برای اصلاح یا جایگزینی سیستم های حمل و نقل شهری توصیه می شود. بخش عمده این مطالعات شامل مدلسازی عملکرد ترافیکی شبکه حمل و نقل در هر وضعیت از تقاضای سفر و تحت هر سیستم حمل و نقل پیشنهادی است. برای ارزیابی عملکرد یک شبکه باید بتوان میزان جریان عبوری در معابر آن را پیش بینی و متناسب با آن معیارهای عملکردی را محاسبه نمود. به طور کلی، برای هر وضعیت مشخص تقاضای سفر (مشاهده یا برآورد شده) ارزیابی هر وضعیت شبکه حمل و نقل (یا هر نوع سیستم حمل و نقل) شامل دو مرحله زیر است:

۱- پیش بینی الگوی جریان در شبکه

۲- استفاده از الگوی جریان برای تخمین معیارهای عملکردی شبکه

در مرحله اول با استفاده از یک سری فرضیات در مورد چگونگی انتخاب مسیر حرکت توسط استفاده کنندگان از شبکه، جریان در کمان های شبکه تعیین میشود. در مرحله دوم با استفاده از الگوی جریان، معیارهای عملکردی شبکه مثل شلوغی، سطح سرویس، نشر آلودگی محاسبه می شوند. سپس، با مقایسه معیارهای حاصل از سیستم های پیشنهادی و با استفاده از روش های اقتصاد مهندسی، راه حل هایی مناسب برای مشکلات سیستم تعیین میشوند. جریان در کمانهای شبکه با حل یک مدل ریاضی به نام "مسئله تخصیص ترافیک" حاصل می شوند. با دانستن مقدار تقاضای سفر بین زوج های مبدا- مقصد در شبکه، مسئله تخصیص ترافیک عبارت از چگونگی توزیع تقاضا بین مسیرهای موجود بین هر زوج مبدا- مقصد است. آنچه که در این پژوهش مورد نظر قرار گرفته شامل مقایسه روش های موجود تخصیص ترافیک و ارایه روشی کارا برای شبکه های بزرگ مقیاس است.

روش های مختلفی برای حل مسئله تخصیص ترافیک وجود دارد که به طور کلی می توان آنها را به سه دسته الگوریتم های بر پایه کمان، بر پایه مسیر و بر پایه مبدا تقسیم کرد. الگوریتم فرانک-ولف از معروف ترین الگوریتم های بر پایه کمان است که به خاطر مصرف حافظه بسیار کم برای بیش از پنج دهه کاربردی ترین الگوریتم تخصیص ترافیک بوده است. این الگوریتم، علی رغم مزایای زیادی که دارد، ظرفیت محدودی برای هرکمان در نظر نمی گیرد و آن را نامحدود فرض میکند. این فرض باعث می شود که جواب های به دست آمده از این الگوریتم دقیق و نزدیک به واقعیت نباشد. با پیشرفت رایانه ها در دو دهه اخیر، الگوریتم های بر پایه مسیر بیشتر مورد توجه قرار گرفته و توانسته اند با نرخ همگرایی بالا به دقتی در جواب مسئله دست پیدا کنند که الگوریتم فرانک-ولف هرگز توانایی رسیدن به آنها را نداشته است. در دهه اخیر نیز الگوریتم های بر پایه مبدا که به حافظه کمتری نسبت به الگوریتم های بر پایه

مسیر نیاز دارند و از نرخ همگرایی نسبتاً خوبی نیز برخوردار میباشند به ادبیات تخصیص ترافیک معرفی شده اند

حافظه قابل دسترس در رایانه های امروزی به حدی زیاد است که می توان با استفاده از آنها الگوریتم های با حافظه مصرفی زیاد را برای شبکه های بزرگ مقیاس نیز اجرا کرد. به همین دلیل کاهش حافظه مصرفی الگوریتم ها جزو اهداف این پژوهش نمی باشد هدف این پژوهش این است که با در نظر گرفتن ظرفیت محدود برای کمان ها در الگوریتم فرانک-ولف، تاثیر آن را در جواب های مسئله را بدست آورد.

۱-۲- اهمیت مسأله

دقت نتایج الگوریتم های تخصیص ترافیک تاثیر بسزایی در پیش بینی و کنترل ترافیک در شبکه های حمل و نقلی دارد. واضح است که کمان ها در یک شبکه حمل و نقلی دارای ظرفیت محدودی هستند و در نظر گرفتن این ظرفیت ها باعث افزایش دقت نتایج بدست آمده می شود و می توان با استفاده از این نتایج تخمین بهتری از شرایط بدست آورد

۱-۳- مروری بر فصول

این پایان نامه از پنج فصل تشکیل شده است که پس از این فصل، چهار فصل آینده بدین شرح می باشند. فصل دوم نگرش های گوناگونی که درباره مسئله تخصیص ترافیک وجود داشته و همچنین روش های مختلفی که محققین برای حل این مسئله بکار برده اند بیان می شود. در این فصل گام های مربوط به الگوریتم های فرانک-ولف و همچنین الگوریتم های تخصیص ترافیک با محدودیت ظرفیت تشریح می شوند.

در فصل سوم، الگوریتم های پیدا کردن کوتاه ترین مسیر بین دو گره با ذکر مثال تشریح و مقایسه شده و الگوریتم Out of Kilter نیز با ذکر مثالی تبیین، نتایج حاصل از اجرای این الگوریتم و الگوریتم فرانک-ولف بر روی دو شبکه با زمان سفر ثابت و همچنین مقایسه زمان حل اعمال الگوریتم های کوتاه ترین مسیر بل و دایجسترا نشان داده می شود. در فصل چهارم، ابتدا به توضیح روش های به کار برده شده برای کدنویسی الگوریتم OFK پرداخته و سپس الگوریتم های Out of Kilter و فرانک-ولف برای شبکه های هرن و سوفالز با زمان سفرهای وابسته به جریان اعمال و نتایج بدست آمده از آنها مقایسه می شود. فصل پنجم نیز به خلاصه ای از نتایج حاصل از این پژوهش و پیشنهاداتی برای مطالعات آینده اختصاص داده شده است

۲ فصل دوم

ادبیات مساله تخصیص ترافیک

۲-۱- مقدمه

در این فصل نخست به تشریح مسئله تخصیص ترافیک^۱ (TAP) و مفاهیم اولیه آن و مرور مقالات و تحقیقات انجام شده پیرامون مسئله تخصیص ترافیک می پردازیم. سپس مدل بهینه سازی تخصیص ترافیک در حالت ظرفیت نامحدود کمان ها ارائه و چند الگوریتم ساده ولی کاربردی برای حل آن بیان می شود. در انتها مدل تخصیص ترافیک با در نظر گرفتن محدودیت ظرفیت کمان ها بیان شده و یکی از الگوریتم ها حل آن ارائه می شود.

۲-۲- مروری بر مدل های تخصیص ترافیک

مطابق آشتیانی [۱] یکی از مناسب ترین معیارها به منظور ارزیابی عملکرد ترافیکی کمان ها در شبکه های حمل و نقل، زمان سفر کمان ها می باشد. زمان سفر هر کمان در حالت کلی تابعی از حجم جریان در آن کمان و نیز حجم دیگر کمان های شبکه در اطراف کمان است. در مسئله تخصیص ترافیک به منظور ساده سازی حل آن اغلب فرض میشود که زمان سفر هر کمان تنها تابعی از جریان در همان کمان بوده و به جریان در دیگر کمان ها وابسته نیست. همچنین زمان سفر هر مسیر از مجموع زمان سفر کمان های واقع در همان مسیر به دست می آید. واضح است که با افزایش جریان در یک کمان ابتدا سرعت عملکردی آن به طور ملایمی کاهش یافته و سپس با افزایش تراکم وسایل نقلیه و ایجاد صف، کاهش سرعت شدت یافته و زمان سفر به سرعت افزایش می یابد. در عمل فرض می شود که تابع زمان سفر-حجم تابعی پیوسته، مشتق پذیر، غیر منفی و صعودی از جریان در کمان است. توابع زمان سفر-حجم متفاوتی برای شرایط مختلف، توسط پژوهشگران ارائه شده است. توابع زمان سفر-حجم مورد استفاده در این پژوهش همان تابع BPR است.

در یک سیستم حمل و نقل مثل یک بازار رقابتی دو گروه وجود دارند. گروه اول استفاده کنندگان (رانندگان) که رفتارشان با تابع تقاضا معرفی می شود و گروه دوم عرضه کنندگان (تولیدکنندگان سیستم) که رفتارشان با تابع عرضه یا کارایی معرفی می شود.

سیستم حمل و نقل نوعی خدمت است که توسط عرضه کنندگان آن ارائه می گردد. این سیستم شامل انواع راه ها، طریقه ی سفر و غیره می شود که جزئی از شبکه محسوب می گردند. میزان کارایی این سیستم توسط سطح سرویس^۲ اندازه گیری می شود که می تواند شامل زمان و هزینه سفر، ایمنی سفر

¹ Traffic Assignment Problem

² Level of Service

و غیره باشد [۱]. تابع تقاضا نشان می دهد که با تغییرات سطح سرویس، تقاضا چگونه کم یا زیاد می شود. جریان توزیع شده در شبکه پس از رسیدن سیستم حمل و نقل به تعادل بین کارایی و تقاضا، جریان تعادلی نامیده می شود. مطابق آشتیانی [۱] مسئله تخصیص ترافیک تعادلی، نوعی مسئله تخصیص ترافیک برای تعیین جریان تعادلی در شبکه است. مسئله تخصیص ترافیک تعادلی بر پایه قانون تعادل استفاده کننده^۱ (UE) استوار می باشد و طبق این قانون در شرایط تعادل هیچ راننده ای نمی تواند با تغییر یک جانبی^۲ مسیر خود، زمان سفرش را کاهش دهد. براساس شفی [۲] در این قانون فرض می شود که اولاً تمام رانندگان اطلاع دقیقی از همه مسیرهای موجود و زمان سفرهای آنها دارند، دوماً الگوی جریان در شبکه در بازه زمانی مورد نظر ثابت است. فرض اول یعنی اطلاع دقیق رانندگان از مسیرها، می تواند دست کم برای همه رانندگان صادق نباشد. بر این اساس مدل های تعادلی به مدل های قطعی^۳ و تصادفی^۴ تقسیم می شوند. در مدل قطعی اطلاعات رانندگان از مسیرها یکسان، ولی در مدل تصادفی این اطلاعات ناقص و متفاوت هستند. به علاوه، هریک از این دو مدل، اگر تقاضا سفر با زمان تغییر کند مدل پویا^۵ و اگر تقاضا نسبت به زمان ثابت باشد مدل ایستا^۶ نامیده می شوند.

مدلهای تخصیص ترافیک تعادلی از نوع ایستا و قطعی، از کاربردی ترین مدل های تخصیص ترافیک به خصوص در برنامه ریزی های بلند مدت محسوب می شوند. در ادامه این پایان نامه، منظور از مدل تخصیص ترافیک همان مدل تخصیص ترافیک تعادلی ایستا و قطعی خواهد بود. فرضیاتی در این مدل وجود دارند. در فرض اول نرخ تقاضا در دوره زمانی مورد نظر ثابت است، در فرض دوم رانندگان اطلاعات دقیقی از مسیرها دارند و در فرض سوم هر راننده سعی میکند زمان سفر خود را کاهش دهد، که به معنی برقراری تعادل استفاده است.

واردراپ [۴] برای اولین بار به وضعیت تعادل در شبکه اشاره کرد که بعد ها به اصل اول واردراپ معروف شد. بر اساس این اصل، در شرایط تعادل، زمان سفر مسیرهای استفاده شده با هم برابر بوده و کمتر یا

¹ User Equilibrium

² Unilaterally

³ Deterministic

⁴ stochastic

⁵ Dynamic

⁶ Static

مساوی با زمان سفر مسیرهای استفاده نشده هستند. این اصل، قانون تعادل ترافیکی^۱ نیز نام دارد. در اینجا از دو نام استفاده کننده و ترافیکی برای مفهوم تعادل استفاده شده است تا بتوان تفاوت بین آنها را نشان داد. در ادبیات تخصیص ترافیک، تعادل استفاده کننده و ترافیکی با فرض آنکه جریان پیوسته و تابع زمان- سفر غیر نزولی و پیوسته باشد، یکسان در نظر گرفته می شوند [۱]. در این پژوهش نیز این دو مفهوم یکسان در نظر گرفته شده اند.

مسئله تخصیص ترافیک برای اولین بار در دهه چهل میلادی مطرح شده و اولین روش برای حل آن امروزه به عنوان روش همه یا هیچ^۲ شناخته می شود. این روش بر این فرض غیر واقعی استوار بوده است که زمان سفر هر کمان مستقل از جریان در شبکه می باشد. پس از پیشرفت های حاصله در الگوریتم های کوتاه ترین مسیر و معرفی مسئله تعادل استفاده کننده توسط واردراپ، تحول بزرگی در فرمول بندی مسئله تخصیص ترافیک به وجود آمد [۳]. امروزه الگوریتم های حل این مسئله را میتوان به سه گروه اصلی تقسیم کرد:

1 - الگوریتم هایبر پایه کمان^۳

2 - الگوریتم های بر پایه مسیر^۴

3 - الگوریتم های بر پایه مبدا^۵

علاوه بر طبقه بندی بالا، روش های ابتکاری که بر پایه فرمول بندی ریاضی نمی باشند را نیز میتوان به عنوان دسته ای جدا به حساب آورد.

مسئله تخصیص ترافیک با ظرفیت محدود^۶ (CTAP) همان مسئله تخصیص ترافیک استاندارد است با این تفاوت که محدودیت های ظرفیت کمان به آن اضافه شده است. به طور کلی، روش های حل مسئله تخصیص ترافیک با ظرفیت محدود به دو دسته کلی تقسیم می شود. دسته اول الگوریتم هایی که از

¹ Traffic Equilibrium

² All or Nothing

³ Link-Based Algorithms

⁴ Path-Based Algorithms

⁵ Origin-Based Algorithms

⁶ Capacitated Traffic Assignment Problem

توابع زمان سفر مجانبی^۱ استفاده می کنند. دسته دوم الگوریتم هایی که مسئله اصلی را به یک سلسله مسائل تخصیص ترافیک بدون محدودیت همراه با استراتژی جریمه/همزاد^۲ تبدیل می کنند.

این دو استراتژی میتواند به طور هم زمان با جداگانه اعمال شوند. اولین استراتژی رفتار ذاتی ترافیک است. یعنی هر استفاده کننده از راه سعی بر حداقل کردن زمان سفر خود دارد و بنابراین به عنوان قانون تعادل شناخته می شود. اگر بردار توابعی که که زمان سفر در کمان های شبکه را توضیح می دهند قابل انتگرال گیری و یکنواخت باشند، قانون بهینه سازی واردراپ موجب به یک مسئله ریاضی محدب می شود. توابع زمان سفر غیرقابل انتگرال گیری ماتریس ژاکوبین نامتقارن دارند و در نتیجه مسئله تخصیص ترافیک به دست آمده به بی نهایت میل می کند.

پاول [۲۶] و هستنز [۲۷] به طور جداگانه از روش ضرایب لاگرانژ تقویت شده^۳ (ALM) برای حل مسائل برنامه ریزی غیرخطی استفاده کردند. میلر و همکاران [۴۳] همچنین یک روش تولید ستون^۴ با در نظر گرفتن هزینه سفر (زمان سفر) ثابت ارائه کردند که در آن مسائل اصلی با محدودیت ظرفیت با استفاده از تکنیک تعمیم یافته کران بندی بالا^۵ حل شده اند. دگنزو [۲۸][۲۹] اولین کسی بود که محدودیت های ظرفیت کمان را با استفاده از توابع زمان سفر مجانبی در نظر گرفت و آن را برای شبکه ای با ۳ گره و ۵ کمان اجرا کرد. این روش همواره جواب های امکان پذیر دارد ولی مشکلاتی نیز دارد، که نتایج تحقیقات انجام شده توسط بویس و همکاران [۳۰] نشان می دهد که استفاده از این توابع ممکن است هزینه سفر را به طور غیر واقعی زیاد نشان دهد و باعث مشکلات عددی در نزدیکی ظرفیت کمان شود. وندراستراتین و تیلکوپین [۳۱] با استفاده از روش ضرایب لاگرانژ تقویت شده و همچنین استفاده از روش غیر خطی Out of Kilter برای زیرمسائل^۶ بدون محدودیت، مسائل تک-کالایی را حل کردند.

^۱ Asymptotical function

^۲ penalty/dual strategy

^۳ Augmented Lagrange Multipliers

^۴ Column Generation Approach

^۵ Generalized Upper Bounding Technique

^۶ Sub-problems

مینوکس [۴۷] از روش شاخه و کران^۱ با استفاده از آزادسازی لاگرانژی^۲ مسئله برنامه نویسی عدد صحیح و آزادسازی برنامه نویسی خطی^۳ برای حل شبکه با محدودیت و هزینه ثابت استفاده کرد و این الگوریتم را برای شبکه ای با ۵ گره و ۸ کمان اجرا کرد. خانگ و فوجیوارا [۴۸] از آزاد سازی لاگرانژی و بهینه سازی زیر گرادیان ها^۴ برای به دست آوردن کران پایین تر و مقیاس گذاری ابتکاری برای به دست آوردن یک کران بالاتر استفاده کردند. جایا کریشان و همکاران [۳۲] و همچنین چن و لی [۲۲] نشان دادند که مسائل با مقیاس بزرگ می توانند با استفاده از الگوریتم های بر مبنای مسیر حل شوند. برای اولین بار هرن [۳۳] مسئله تخصیص ترافیک با ظرفیت محدود را با استفاده از تبدیل آن به یک سلسله زیر مسائل تخصیص ترافیک بدون محدودیت ظرفیت حل کرد. او از یک تابع جریمه خارجی^۵ برای تبدیل مسائل با محدودیت ظرفیت به مسائل بدون محدودیت ظرفیت استفاده کرد و زیر مسائل را با استفاده از الگوریتم فرانک-ولف حل کرد. او این روش را برای شبکه ای با ۹ گره و ۱۸ کمان اجرا کرد که به شبکه هرن معروف شد. اینوی [۴۲] روش مشابه را با استفاده از توابع جریمه که در آن هزینه سفر به بی نهایت میل می کنند ارائه داد و این روش را توابع جریمه داخلی نامید^۶ و مانند هرن با استفاده از الگوریتم فرانک-ولف زیرمسائل را حل کرد. یانگ و یاگر [۴۴] نتایج اعمال این روش را با در نظر گرفتن محدودیت ظرفیت ارائه دادند و آن را بر روی شبکه ای با ۹ کمان و ۶ گره اعمال کردند. هرن و ریرا [۳۴] از روش ALM برای حل مسئله تخصیص ترافیک با محدودیت ظرفیت استفاده کردند. آن ها الگوریتم فرانک-ولف را برای حل زیر مسائل به کار بردن و نتایج را برای تعدادی شبکه کوچک ارائه کردند. آنها دو نوع از توابع لاگرانژ را در نظر گرفتند. لارسون و پاتریکسون [۳۵] روش ALM^۱ برای حل شبکه سوفالز پیشنهاد و ارزیابی کردند. آنها در این روش، روش توابع جریمه خارجی معمول با رویه همزاد لاگرانژ^۷ ترکیب شده کردند و نتایج به دست آمده از اعمال روش ALM را برای شبکه های کوچک تا متوسط

^۱ Branch-and-Bound

^۲ Lagrangean Relaxations

^۳ Linear Programming Relaxations

^۴ subgradients

^۵ Exterior Penalty Function

^۶ Inner Penalty Functions

^۷ Lagrangean Dual Schemes

ارائه کردند. در این روش زیرمسائل با استفاده از الگوریتم تجزیه غیرتجمعی ساده شده^۱ حل شده اند. آنها از روش تولید ستون برای حل مسائل با محدودیت های خطی استفاده کردند. برای بهبود عملکرد محاسباتی، آنها از رویه تثبیت همزاد^۲ استفاده کردند. این روش نرخ خطی همگرایی را نشان می دهد. بارگرا [۳۶] روش جدیدی برای حل زیرمسائل بدون محدودیت ظرفیت ارائه داد و آن را روش مبتنی بر مبدا^۳ (OB) نامید. در این روش تمام محاسبات تحت یک روند مکان نگر بر روی شبکه سوفالز اعمال شده است. همان طور که در [۳۶] نشان داده شده است این روش زمان محاسبات را به طور قابل ملاحظه ای کاهش می دهد یکی از مزایای اصلی روش OB این است که جواب های به دست آمده تفسیر جریان ساده ای دارند و مقدار حافظه مورد نیاز متوسط است [۳۷]. پرشکار و تولدو [۳۸] از روش IPF برای حل کردن زیر مسائل بدون محدودیت ظرفیت استفاده کردند و نتایج را با الگوریتم های یانگ و یاگر [۴۴] مقایسه کردند. جواب های سلسله زیرمسائل هنگامی به جواب بهینه مسئله اصلی همگرا می شوند که عبارت های جریمه به صفر میل کنند. این الگوریتم در مقایسه با الگوریتم یانگ و یاگر [۴۴] سریع تر و با رفتار پایدارتری همگرا می شود و بر روی شبکه ای با ۱۰ گره و ۴۰ کمان اعمال شده است. نی و همکاران [۳۹] روش IPF را برای شبکه های هرن، سوفالز و آناهیم اعمال کردند و از روش تصویر گرادیان^۴ (GP) برای حل زیرمسائل بدون محدودیت استفاده کردند. آنها روش IPF را با الگوریتم ALM مقایسه کردند و نتیجه ها نشان می دهد که IPF در به تعادل رساندن زیر مسائل کارایی کمتری از ALM دارد اما ALM نسبت به سرعت همگرایی، حساس تر از IPF است. هردو الگوریتم IPF و ALM نتایج مشابه و واقعی قابل قبولی را به دست آوردند. نگویان و همکاران [۴۰] فرض کردند که وضعیت سیستم وابسته به جریان است و رفتار استفاده کننده بیشتر به وسیله استراتژی های سفر به او تحمیل می شود تا انتخاب مسیر. تعادل استراتژیک زمانی به دست می آید که تمام استفاده کننده ها از استراتژی های حداقل کردن تاخیر مورد انتظار استفاده کنند. در این پژوهش برای حل کردن زیر مسائل از الگوریتم فرانک-ولف استفاده شده است و این الگوریتم بر روی شبکه ای با ۵ گره و ۸ کمان اجرا شده است. کورا و همکاران [۴۱] تعادل استفاده کننده در شبکه های با ظرفیت را با حالات کلی توابع تاخیر معرفی و آنالیز کردند. آن ها فرض کردند که توابع تاخیر توابعی پیوسته، غیر منفی و غیر کاهشی هستند و از ظرفیت کمان ها به عنوان محدودیت های اضافی استفاده کردند. نتایج آن ها یک روش استقرایی برای به

¹ Disaggregate Simplicial Decomposition Algorithm

² Dual Stabilization Scheme

³ Origin-Based Method

⁴ Gradient Projection

کار بردن تعادل استفاده کننده در مدل های تخصیص ترافیک فراهم کرد. شهپر و همکاران [۴۵] یک الگوریتم تکراری برای حل TAP با محدودیت خطی ارائه دادند. این الگوریتم بر مبنای اضافه کردن یک تابع جریمه دینامیک^۱ (DPF) برای زمان سفر کمان ها است و نتایج آن همگرایی مناسب این روش را نشان می دهد. DPF با کارایی بیشتری نسبت به روش های IPF و ALM به نتایج ممکن توابع بکمن دست می یابد چون تعداد تکرارهای روش DPF همواره از دو روش ذکر شده کمتر است. این الگوریتم بر روی شبکه های هرن، سوفالز و آناهیم اعمال شده است. شلیوند و همکاران [۴۶] الگوریتمی را ارائه دادند که مسئله اصلی را با یک سلسله TAP تقریب می زند. این زیر مسائل با الگوریتم های OB که در آن اصلاحاتی انجام گرفته است حل می شوند. این الگوریتم اصلاح شده، الگوریتم مبتنی بر مبدا توسعه یافته^۲ (EOB) نام دارد و بر روی شبکه های کوچک تا بزرگ اعمال شده است. روش پیشنهاد شده تفسیر جریان مسیر ساده ای دارد و برای شبکه های بزرگ کارا است و همچنین باعث کاهش زمان محاسبه و مقدار حافظه مورد نیاز می شود.

۲-۳- مدل بهینه سازی تخصیص ترافیک

همان طور که اشاره شد اولین بار واردراپ [۴] وضعیت تعادل در شبکه را بررسی کرد که بعدها با نام او با عنوان اصل واردراپ معروف شد. بر اساس آن، زمان سفر مسیرهای استفاده شده در شرایط تعادل با هم برابر بوده و کمتر یا مساوی زمان سفر مسیرهای استفاده نشده است. اصولی که عملکرد رانندگان را در انتخاب مسیر نشان می دهند غالباً بر مبنای موارد اشاره شده، بیان می شوند. در سال ۱۹۵۲ واردراپ این دو اصل اساسی را به نام های اصول اول و دوم به شرح زیر بیان نمود. اصل اول واردراپ به عنوان یک اصل منطقی و ساده برای تشریح نحوه توزیع تقاضا بین مسیرهای مختلف در شرایط تراکم پذیرفته شده است. مطابق این اصل، در شرایط تعادل استفاده کننده هیچ مسافری نمی تواند با تغییر مسیر، هزینه سفر خود (که می تواند زمان سفر او باشد) را کاهش دهد. جریان ترافیکی که این اصل را ارضا کند غالباً تحت عنوان جریان تعادل استفاده کننده یا جریان تعادلی نامیده می شود. زیرا هر راننده بهترین مسیر را انتخاب کرده است. از اصل اول واردراپ به سادگی می توان نتیجه گرفت که در شرایط تعادل استفاده کننده در شبکه های بدون محدودیت ظرفیت، زمان سفر مسیرهای استفاده شده بین هر زوج مبدا مقصد با هم برابر و کمتر یا مساوی با زمان سفر مسیرهای استفاده نشده هستند. این مساله بهینه سازی که مدل بکمن نامیده می شود به شرح رابطه ۲-۱ است.

¹ Dynamic Penalty Function

² Extended Origin-Based algorithm

$$\left\{ \begin{array}{l} \min z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega \\ s.t. \\ \sum_k f_k^{rs} = q_{rs} \quad \forall r,s \\ f_k^{rs} \geq 0 \quad \forall r,s,k \\ x_a = \sum_{(r,s)} \sum_k f_k^{rs} \delta_{ak}^{rs} \quad \forall a \in A \end{array} \right. \quad (1-2)$$

در این مدل x_a جریان در کمان a ، $t_a(x)$ تابع زمان سفر کمان a ، f_k^{rs} جریان در مسیر k برای زوج مبدا مقصد rs و q_{rs} تقاضای بین زوج مبدا مقصد rs می باشد. همچنین پارامتر δ_{ak}^{rs} برابر یک است اگر کمان a روی مسیر k مربوط به زوج مبدا مقصد rs واقع باشد و در غیر این صورت برابر صفر است. در مدل بکمن (۲-۲) تابع هدف عبارت است از مجموع انتگرال های توابع زمان سفر کمان ها. این تابع هدف همان گونه که شفای [۲] بیان می کند هیچ تعبیر اقتصادی یا رفتاری ندارد. در این مدل محدودیت اول بیان می دارد که مجموع جریان عبوری از مسیر های مربوط به یک زوج مبدا مقصد بایستی با تقاضای مربوط به آن زوج برابر باشد. محدودیت دوم به دلیل اطمینان از منطقی بودن جواب نهایی مساله بیان شده و محدودیت سوم نشان دهنده رابطه بین جریان در مسیر k مربوط به زوج مبدا مقصد rs و جریان در کمان a می باشد.

۲-۳-۱- روش های نزول

روش های نزول برای مینیمم سازی توابع محدب غیرخطی چندمتغیره به کار می روند. این الگوریتم ها از جریان در هر کمان به دست آمده در هر مرحله $(x^n = (x_1^n, \dots, x_I^n))$ جریان در مرحله بعد $(x^{n+1} = (x_1^{n+1}, \dots, x_I^{n+1}))$ را بدست می آورند.

مرحله اصلی روش های نزول محاسبه x^{n+1} از x^n است. این مرحله را می توان به صورت استاندارد مطابق رابطه ۲-۲ بیان کرد.

$$x^{n+1} = x^n + \alpha_n d^n \quad (2-2)$$

که در این رابطه d^n بردار نزول و α_n کمیتی غیرمنفی به نام جهت حرکت است. این رابطه بیان می کند که در نقطه x^n ، جهتی به نام d^n وجود دارد که تابع در آن نزولی است. در رابطه ۲-۲، هر عضو از بردار متغیر به روزرسانی می شود، که به صورت رابطه ۲-۳ بیان می شود:

$$x_i^{n+1} = x_i^n + \alpha_n d_i^n \quad \forall i=1,2, \dots, I \quad (3-2)$$

فرمول های بیان شده در روابط ۲-۲ و ۳-۲ برای تمام روش های نزول برای مینیمم سازی بدون محدودیت استفاده می شود. روش های نزول عمدتاً در قواعدی که برای یافتن جهت نزول است با هم تفاوت دارند. اندازه حرکت معمولاً طوری انتخاب می شود که تابع هدف را در جهت نزول مینیمم کند.

اگر تابع هدف دارای محدودیت باشد، الگوریتم های مینیمم سازی باید طوری اصلاح شوند که جهت مورد نظر باتوجه به قواعد موردنظر تغییر کند. به طور مشابه، اگر اندازه حرکت طوری باشد که تابع به خارج از محدوده امکان پذیر وارد شود، باید طوری اصلاح شود که امکان پذیری نقض شود.

۲-۳-۲ الگوریتم فرانک-ولف

فهم آسان، سادگی در اجرا، حافظه مصرفی کم و کارایی مناسب در تکرارهای اولیه باعث محبوبیت فراوان الگوریتم فرانک-ولف در میان محققان شده است، به طوری که در نرم افزارهای تخصصی تخصیص ترافیک از جمله EMME2 [۸] از این روش استفاده شده است. برای به دست آوردن جهت، ابتدا مسئله محدب را در نظر بگیریم.

$$\begin{cases} \min z(x) \\ s.t. \\ \sum_i h_{ij} x_i \geq b_j \quad \forall j \in J \end{cases} \quad (4-2)$$

در تکرار n ام، جواب را x^n در نظر میگیریم. بیشتر روش های یافتن جهات امکان پذیر از اطلاعات شیب تابع هدف در همسایگی x^n برای تعیین جهات نزول استفاده میکند. بنابراین، جهت نزول براساس عکس جهت گرادیان یا جهت تندترین شیب نزول امکان پذیر تعیین می شود. معیار انتخاب جهات براساس ضرب نرخ نزول در همسایگی x^n در جهت داده شده و طول منطقه امکان پذیر در آن جهت می باشد. این حاصلضرب که به عنوان افت شناخته می شود، حد بالای کاهش در تابع هدف می باشد. این الگوریتم از جهت حرکت استفاده میکند تا مقدار افت را ماکزیمم کند. برای بدست آوردن جهت نزول، الگوریتم از کل منطقه امکان پذیر برای بدست آوردن یک جواب کمکی ($y^n = (y_1^n, \dots, y_I^n)$) طوری استفاده میکند که جهت از x^n به y^n بیشترین مقدار افت را فراهم کند. جهت از x^n به هر جواب ممکن (y) برد ($y - x^n$) (یا بردار واحد $\frac{(y - x^n)}{\|y - x^n\|}$) است. شیب $Z(x^n)$ در جهت $(y - x^n)$ با استفاده از تصویرسازی عکس گرادیان $[-\nabla Z(x^n)]$ که به صورت رابطه ۵-۲ بیان میشود، به دست می آید.

$$-\nabla Z(x^n) \cdot \frac{(y - x^n)^T}{\|y - x^n\|} \quad (5-2)$$

مقدار افت در تابع هدف در جهت $(y-x^n)$ با استفاده از ضرب شیب در فاصله بین x^n و y ، $\|y-x^n\|$ ، به دست می آید و مطابق رابطه ۶-۲ محاسبه میشود.

$$-\nabla z(x^n) \cdot (y - x^n)^T \quad (6-2)$$

رابطه ۶-۲ باید با توجه به امکان پذیری y مینیمم شود. می توان این رابطه را در (-1) ضرب و آن را مینیمم کرد که به صورت رابطه ۷-۲ بدست می آید.

$$\begin{cases} \min \nabla z(x^n) \cdot (y - x^n) = \sum_i \frac{\partial z(x^n)}{\partial x_i} (y_i - x_i^n) \\ s.t. \\ \sum_i h_{ij} x_i \geq b_j \quad \forall j \in J \end{cases} \quad (7-2)$$

جواب این مسئله y^n و جهت نزول بردار متصل کننده x^n به y^n ، $d^n = y^n - x^n$ ، است. پیدا کردن جهت نزول شامل مینیمم سازی برنامه خطی ذکر شده در رابطه ۷-۲ می باشد. جهت نزول را می توان با در نظر گرفتن الگوریتم فرانک-ولف به عنوان یک روش تقریب خطی در نظر گرفت. این روش براساس یافتن جهت نزول با استفاده از مینیمم سازی تقریب خطی تابع هدف در نقطه جواب کنونی می باشد. مینیمم سازی این تابع خطی شده با محدودیت های خطی یک مسئله برنامه نویسی خطی می باشد که جواب های آن در گوشه های منطقه امکان پذیر است. خطی که نقاط جواب کنونی x^n را به جواب مسئله خطی شده y^n متصل می کند جهت جستجو می باشد. برای اثبات این مسئله، $Z_L^n(y)$ را تقریب خطی از مقدار تابع هدف در نقطه y در نظر میگیریم. تقریب خطی را می توان به صورت رابطه ۸-۲ بیان کرد.

$$Z_L^n(y) = z(x^n) + \nabla z(x^n) \cdot (y - x^n)^T \quad (8-2)$$

تابع خطی y باید با توجه به محدودیت های تابع اصلی مینیمم شود که به صورت رابطه ۹-۲ می توان آن را بیان کرد.

$$\begin{cases} \min Z_L^n(y) = z(x^n) + \nabla z(x^n) \cdot (y - x^n)^T \\ s.t. \\ \sum_i h_{ij} y_i \geq b_j \quad \forall j \in J \end{cases} \quad (9-2)$$

در نقطه $x = x^n$ مقدار تابع هدف ثابت است، پس می توان مقدار $z(x^n)$ را از رابطه ۹-۲ حذف کرد. رابطه باقیمانده همان رابطه ۷-۲ می باشد و جواب های آن می توانند در پیدا کردن جهت نزول برای الگوریتم فرانک-ولف استفاده شوند. تابع هدف مسئله خطی شده را می توان ساده سازی کرد. مقدار $\nabla z(x^n)$ در

نقطه x^n ثابت می باشد، پس می توان عبارت $\nabla Z(x^n) \cdot (x^n)^T$ را از رابطه ۹-۲ حذف کرد و رابطه ۱۰-۲ را بدست آورد.

$$\begin{cases} \min Z_L^n(y) = \nabla Z(x^n) \cdot y^T = \sum_i \left(\frac{\partial Z(x^n)}{\partial x_i} \right) y_i \\ s.t. \\ \sum_i h_{ij} y_i \geq b_j \quad \forall j \in J \end{cases} \quad (10-2)$$

جواب های این مسئله $(y^n = (y_1^n, \dots, y_I^n))$ برای بدست آوردن جهت نزول استفاده می شوند. اندازه حرکت در جهت d^n برابر با فاصله با نقطه ای است که $Z(x)$ را مینیمم می کند. الگوریتم فرانک-ولف به گام ویژه ای برای بدست آوردن اندازه گام بهینه نیاز ندارد. جواب جدید x^{n+1} باید بین x^n و y^n قرار گیرد. به عبارت دیگر، جستجو برای جهت نزول به طور اتوماتیک مرزی برای جستجوی خطی با در نظر گرفتن همه محدودیت ها ایجاد می کند. چون بازه جستجو مشخص است، تمام روش های کاهش بازه جستجو برای مینیمم سازی $Z(x)$ در جهت نزول، که برای حل رابطه ۱۱-۲ می باشد، مناسب می باشند.

$$\begin{cases} \min z[x^n + \alpha(y^n - x^n)] \\ 0 \leq \alpha \leq 1 \end{cases} \quad (11-2)$$

زمانی که جواب بهینه جستجوی خطی (α_n) بدست آمد، جواب بعدی را میتوان با استفاده از رابطه ۱۲-۲ بدست آورد.

$$x^{n+1} = x^n + \alpha_n (y^n - x^n) \quad (12-2)$$

حال برای آنکه رابطه اثبات شده را برای شبکه های حمل و نقلی به کار ببریم، تابع هدف قانون تعادل استفاده کننده را در نظر می گیریم.

$$\begin{cases} \min z(x) = \sum_a \int_0^{x_a} t_a(w) dw \\ s.t. \\ \sum_k f_k^{rs} = q_{rs} \quad \forall r, s \\ f_k^{rs} \geq 0 \quad \forall k, r, s \end{cases} \quad (13-2)$$

که x_a جریان در کمان a و f_k^{rs} جریان در مسیر k متصل کننده مبدا r و مقصد s است. حال با توجه به رابطه ۱۰-۲ داریم:

$$\min z^n(y) = \nabla Z(x^n) \cdot y^T = \sum_a \frac{\partial Z(x^n)}{\partial x_a} y_a \quad (14-2)$$

گرادیان $z(x)$ نسبت به جریان های کمان در تکرار n ام برابر با بردار زمان سفر کمان است زیرا که

$$\frac{\partial z(x^n)}{\partial x_a} = t_a^n$$
 بنابراین تابع هدف خطی شده برابر است با :

$$\begin{cases} \min z^n(y) = \sum_a t_a^n y_a \\ s.t. \\ \sum_k g_k^{rs} = q_{rs} \quad \forall r,s \\ g_k^{rs} \geq 0 \quad \forall r,s,k \end{cases} \quad (15-2)$$

که در آن $y_a = \sum_{r,s} \sum_k g_k^{rs} \delta_{ak}^{rs}$ ، $t_a^n = t_a^n(x^n)$ می باشد. در این مسئله خطی، y_a متغیر کمکی نشان دهنده جریان در کمان a و g_k^{rs} متغیر جریان کمکی برای مسیر k که مبدا-مقصد r - s را به هم متصل می کند، می باشد. این مسئله زمان سفر کلی را در یک شبکه با زمان سفر ثابت مینیمم می کند. زمان سفر کلی در شبکه با تخصیص تمام جریان به مسیر با کوتاه ترین زمان سفر مینیمم می شود. این تخصیص با روش تخصیص همه یا هیچ انجام می شود.

برای اثبات این موضوع که جواب رابطه ۱۵-۲ با استفاده از تخصیص همه یا هیچ بدست می آید، گام خطی سازی الگوریتم فرانک-ولف با گرفتن گرادیان تابع هدف با توجه به جریان های در مسیر بدست می آید. بنابراین مسئله خطی شده به صورت رابطه ۱۶-۲ نوشته می شود.

$$\begin{cases} \min z^n(g) = \nabla_f z[x(f^n)] \cdot g^T = \sum_{r,s} \sum_k c_k^{rs} g_k^{rs} \\ s.t. \\ \sum_k g_k^{rs} = q_{rs} \quad \forall r,s \\ g_k^{rs} \geq 0 \quad \forall r,s,k \end{cases} \quad (16-2)$$

که c_k^{rs} زمان سفر در مسیر k که مبدا-مقصد r - s را به هم متصل می کند در تکرار n ام می باشد. این مسئله علت ثابت بودن زمان سفر می تواند به تفکیک مبدا-مقصد جدا شود. زیرمسئله های حاصل برای هر زوج مبدا-مقصد r - s به صورت رابطه ۱۷-۲ می باشد.

$$\begin{cases} \min z^n(g^{rs}) = \sum_k c_k^{rs} g_k^{rs} \\ s.t. \\ \sum_k g_k^{rs} = q_{rs} \\ g_k^{rs} \geq 0 \quad \forall k \end{cases} \quad (17-2)$$

زمانی که جریان در مسیر (g_k^{rs}) پیدا شد، جریان در کمان ها با استفاده از رابطه ۱۸-۲ بدست می آید.

$$y_a = \sum_{r,s} \sum_k g_k^{rs} \delta_{ak}^{rs} \quad \forall a \quad (18-2)$$

و جواب این مسئله جهت نزول، $d^n = y^n - x^n$ ، را به ما میدهد.

جستجوی خطی برای یافتن اندازه گام بهینه را می توان با استفاده از روش دوبخشی^۱ انجام داد. علت آن این است که مشتق تابع هدف نسبت به α که معادل با رابطه ۲-۱۹ است، به آسانی برای هر مقدار α قابل محاسبه است.

$$\frac{\partial}{\partial \alpha} z [x^n + \alpha(y^n - x^n)] = \sum_a (y_a^n - x_a^n) t_a [x_a^n + \alpha(y_a^n - x_a^n)] \quad (19-2)$$

مراحل این الگوریتم به طور خلاصه در زیر آمده است:

گام (۱) - (مقداردهی/اولیه)

قرار دهید $n = 0$. با انجام یک تخصیص همه یا هیچ مقادیر اولیه جریان در کمان های x_a^n را بیابید..

گام (۲) - (آزمون توقف)

زمان سفر کمان های شبکه $t_a = t_a(x_a)$ را بهنگام کنید. اگر x جواب با دقت مورد نظر است، پایان.

گام (۳) - (تعیین جهت)

یک تخصیص همه یا هیچ را برای یافتن جوابهای کمکی y_a^n ها برای همه کمان های شبکه انجام دهید.

جهت نزول $(d_a = y_a - x_a)$ را محاسبه کرده

گام (۴) - (تعیین اندازه گام)

اندازه گام (α) را از طریق حل مسئله بهینه سازی تک متغیره زیر بدست آورید.

$$\begin{cases} \text{Min } Z(\alpha) = \sum \int_0^{x_a^{n+1}} t_a(v) dv \\ \text{s.t.} \quad 0 \leq \alpha \leq 1 \end{cases} \quad (20-2)$$

گام (۵) - (حرکت)

^۱bisection

جریان در کمان ها را به صورت $x_a^{n+1} = x_a + \alpha d$ محاسبه کنی. قرار دهید $n=n+1$ و به گام ۲ بروید.

لبلانک و همکاران [۶] روش فرانک- ولف [۷] را برای حل مسئله تخصیص ترافیک مورد استفاده قرار دادند. این الگوریتم، که معروف ترین روش برای حل مسئله تخصیص به شمار میرود، شبیه روش میانگین گیری متوالی است با این تفاوت که از یک نوع جستجوی خطی برای یافتن اندازه گام در هر تکرار استفاده میکند. طبق لبلانک [۹] و فلورین [۱۰] نرخ پایین همگرایی الگوریتم فرانک- ولف علیرغم مزیت های ذکر شده باعث شد تا تلاش های بسیاری برای بهبود آن از طریق بهبود جهت نزول و یا اندازه گام صورت گیرد. از جمله روش هایی که با تصحیح جهت نزول سعی در بهبود الگوریتم داشت روش PARTAN میباشد. در این روش، با ترکیب جواب حاصل از هر مرحله با جواب مرحله قبل و انجام یک جستجوی خطی اضافی در هر تکرار، جهت حرکت نسبت به الگوریتم فرانک- ولف تصحیح و با این کار سعی شده است تا از شدت حرکت زیگزاگی کاسته شود. فوکوشیما [۱۲] نیز با تصحیح جهت حرکت سعی در بهبود الگوریتم فرانک- ولف داشته است. او با ذخیره کردن جواب های حاصل از تخصیص همه یا هیچ مراحل قبل، جهت حرکت هر تکرار را از ترکیب محدب آنها بدست می آورد. وینتراب و همکاران [۱۳] سعی کردند تا با انتخاب اندازه گام های بزرگتر نسبت به اندازه گام حاصل از جستجوی خطی، حرکت زیگزاگی را حذف کنند. در صورتیکه اندازه گام انتخابی بیش از حد بزرگ می شد و بهبودی در تابع هدف حاصل نمی شد، همان اندازه گام معمول برای آن تکرار در نظر گرفته می شد. لی و نی [۱۴] با ایجاد تغییراتی در الگوریتم فوکوشیما، روش کاراتری را پیشنهاد کردند. در هر دو این روش ها با ذخیره کردن جواب های تخصیص همه یا هیچ تکرارهای قبل، یک جهت نزول جایگزین در هر تکرار محاسبه می شود. سپس این جهت با جهت حاصل از تخصیص همه یا هیچ همان تکرار مقایسه میگردد، بدین شرح که مشتق تابع هدف در این دو جهت محاسبه و جهتی که مقدار مشتق سویی کمتری داشته باشد به عنوان جهت حرکت انتخاب میگردد. فرق اساسی این الگوریتم با الگوریتم فوکوشیما در این است که الگوریتم دوم از تعداد جواب های کمتری برای ساختن جهت نزول جایگزین استفاده میکند. مقایسه الگوریتم پیشنهادی با دیگر روش های بهبود الگوریتم فرانک- ولف نشان داد که این الگوریتم دارای بهترین کارایی، و پس از آن به ترتیب کارایی الگوریتم های فوکوشیما، وینتراب، فرانک- ولف و PARTAN بهتر است. در ادامه روش میانگین گیری متوالی که از الگوریتم های تخصیص ترافیک است توضیح داده می شود.

۲-۳-۲- روش میانگین گیری متوالی

این روش که توسط شفی [۲] برای حل مسئله تخصیص ترافیک ارائه شده است مانند روش فرانک- ولف می باشد با این تفاوت که مقدار α را در هر گام ثابت در نظر می گیرد و آن را می توان با مراحل زیر بیان کرد:

گام (۱) - (مقدار دهی / اولیه)

برای ایجاد یک بردار مقدار جریان اولیه x^0 تخصیص ترافیک همه یا هیچ با زمان های $t_a^0 = t(a)$ انجام دهید. قرار دهید $n=1$.

گام (۲) - (به روز رسانی زمان سفر)

با مقدار جریان x^n زمان سفر کمان ها t_a^n را به روز رسانی کنید .

گام (۳) - (جهت یابی)

با مقادیر t_a^n تخصیص ترافیک همه یا هیچ را انجام دهید و مقادیر بردار جریان y_a^n را بدست آورید.

گام (۴) - (حرکت)

جواب بعدی x_a^{n+1} که بین x^n و y^n می باشد به صورت زیر بدست می آید:

$$x^n = x^{n-1} + \frac{1}{n}(y^n - x^{n-1}) \quad (21-22)$$

مقدار n را یک واحد افزایش دهید: $n=n+1$ و به گام ۲ بروید.

۲-۴- مدل تخصیص ترافیک با ظرفیت محدود

کاربرد وسیع مدل تخصیص ترافیک در ارزیابی و انتخاب راه کارهای مناسب برای بهبود عملکرد شبکه حمل و نقل نیازمند واقعی تر شدن هرچه بیشتر نتایج حاصل از آن است. یکی از عوامل واقعی تر شدن نتایج این مدل ، مستقل از روش حل آن، در نظر گرفتن محدودیت ظرفیت کمان ها است.

اضافه شدن محدودیت ظرفیت به مدل بهینه سازی تخصیص ترافیک سبب از بین رفتن کارایی روش های حل دیگر می شود. زیرا در این حالت زیر مسئله خطی شده مدل بهینه سازی شده - که معادل مسئله کوتاه ترین مسیر است - به مسئله جریان چند کالایی با هزینه کمینه در شبکه تبدیل می شود که حل آن نسبت به مسئله کوتاه ترین مسیر بسیار دشوارتر و زمان برتر است. به طور کلی در اغلب روش های مسئله تخصیص ترافیک با محدودیت ظرفیت تلاش می شود تا از حل مسئله چند کالایی پرهیز شود.

فرمول بندی مسئله فرانک-ولف با ظرفیت محدود برای هر کمان به شکل فرمول ۲-۲۲ می باشد.

$$\left\{ \begin{array}{ll} \min z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega & \\ s.t. & \\ \sum_k f_k^{rs} = q_{rs} & \forall r,s \\ f_k^{rs} \geq 0 & \forall r,s,k \\ x_a = \sum_{(r,s)} \sum_k f_k^{rs} \delta_{ak}^{rs} \leq c_a & \forall a \in A \end{array} \right. \quad (22-2)$$

تابع هدف و محدودیت های ۱ و ۲ در این فرمول بندی مشابه فرمول ۱-۲ است و محدودیت ۳ رابطه بین جریان در کمان x_a و ظرفیت هر کمان c_a را بیان می کنند.

۲-۵- الگوریتم شهپر و همکاران

در این پژوهش، شهپر و همکاران از یک تابع جریمه دینامیکی برای تبدیل مسئله تخصیص ترافیک با محدودیت ظرفیت برای هر کمان به زیرمسائل بدون ظرفیت استفاده کردند. آنها فرض کردند که تابع زمان سفر مثبت، پیوسته و صعودی می باشد و توابع محدودیت (که برای اجزایی از شبکه مانند گره ها و مسیرها) دارای مدل بهینه سازی محدب می باشند. در این الگوریتم زمان سفر کلی به صورت فرمول ۲-۲۳ تعریف میشود:

$$C_k^{rs}(x, \lambda) = \sum_{a \in A} (t_a(x_a) + \sum_{j \in J} \lambda_j \frac{\partial g_j(x)}{\partial x_a}) \delta_{ak}^{rs} \quad (23-2)$$

که در این مدل $C_k^{rs}(x, \lambda)$ برابر با کوتاه ترین زمان سفر از مسیر k بین زوج مبدا-مقصد (r, s) ، $g_j(x)$ برابر با j امین تابع محدودیت، x_a جریان در کمان a ، λ_j ضریب لاگرانژی مربوط به j امین تابع محدودیت و δ_{ak}^{rs} در صورتی که کمان a روی مسیر k باشد برابر با ۱ و در غیر این صورت صفر است، می باشند. در فرمول ۲-۲۳ عبارت $\sum_{j \in J} \lambda_j \frac{\partial g_j(x)}{\partial x_a}$ تاخیر ناشی از محدودیت جانبی در هر کمان می باشد.

تابع جریمه به صورت $P_j(x, \alpha)$ تعریف می شود که در آن α متغیری است که دامنه آن بین $(0, \infty)$ می باشد. فرض اول که برای تابع جریمه در این الگوریتم وجود دارد عبارتست از مشتق پذیر ، یکنواخت و غیرمنفی بودن این تابع. فرض دوم بیان میکند که اگر $g_j(x) > 1$ باشد، مقدار این تابع برابر است با α_j است. در فرض سوم اگر $g_j(x) > 1$ مقدار تابع جریمه به اندازه قابل توجهی بزرگ می باشد و اگر

$g_j(x) < 1$ مقدار تابع جریمه به طور قابل توجهی کوچک می باشد. در این پژوهش تابع جریمه به صورت فرمول ۲۴-۲ بیان می شود. در این مقاله تابع جریمه به توابع زمان سفر اضافه می شوند.

$$P_j(x, \alpha) = \alpha_j \Psi(g_j(x)) \quad (24-2)$$

که $\Psi(g_j(x))$ مانند فرمول ۲۵-۲ بیان میشود.

$$\Psi(g_j(x)) = \begin{cases} \frac{\rho}{2(1-g_j(x))} & y < 1-\rho \\ \frac{y-1}{2\rho} + 1 & y \geq 1-\rho \end{cases} \quad (25-2)$$

که ρ متغیری با دامنه $(0,1)$ می باشد. که α_j در هر مرحله به صورت زیر به روز رسانی می شود:

$$\alpha_j^n = P_j(x^n, \alpha_j^{n-1}) = \alpha_j^{n-1} \Psi(g_j(x^n)) \quad (26-2)$$

در این الگوریتم معیار توقف به صورت رابطه ۲۷-۲ بیان میشود.

$$\begin{aligned} g_j(x^n) &\leq 1 & \forall j \in J \\ (1-g_j(x^n))\alpha_j^n &\leq \alpha_j^0 \rho & \forall j \in J, g_j(x^n) < 1-\rho \end{aligned} \quad (27-2)$$

اگر $\alpha > 0$ زیر مسئله را می توان به زیر مسائل بدون محدودیت تبدیل کرد که در آنها زمان سفر کلی را می توان به صورت فرمول ۲۸-۲ بیان کرد:

$$\bar{t}_a(x_a) = t_a(x_a) + \sum_{j \in J} P_j(x, \alpha_j) \frac{\partial g_j(x)}{\partial x_a} \quad (28-2)$$

بدین صورت می توان λ_j را با $\sum_{j \in J} P_j(x, \alpha_j)$ تقریب زد و اگر جریان در مسیر را مطابق فرمول ۲۹-۲ بیان کرد، آنگاه با جایگذاری $\sum_{j \in J} P_j(x, \alpha_j)$ در فرمول ۲۳-۲، به رابطه ۳۰-۲ خواهیم رسید.

$$x_a(f) = \sum_r \sum_s \sum_k f_k^{rs} \delta_{ak}^{rs} \quad (29-2)$$

$$C_k^{rs}(x, \lambda) = \sum_{a \in A} (t_a(x_a) + \sum_{j \in J} \lambda_j \frac{\partial g_j(x)}{\partial x_a}) \delta_{ak}^{rs} \quad (30-2)$$

بنابراین مسئله ما به صورت رابطه ۳۱-۲ تعریف می شود.

$$\left\{ \begin{array}{l} (c_k^{rs}(f, \alpha) - u_{rs}) f_k^{rs} = 0 \\ c_k^{rs}(f, \alpha) - u_{rs} \geq 0 \\ (\sum_{k \in K_{rs}} f_k^{rs} - q_{rs}) u_{rs} = 0 \\ \sum_{k \in K_{rs}} f_k^{rs} - q_{rs} \geq 0 \\ u_{rs} \geq 0 \end{array} \right. \quad (31-2)$$

به این مسئله، مسئله تکمیلی غیرخطی^۱ می گویند. در هر تکرار، مسئله تکمیلی غیرخطی به تفکیک مبدا-مقصد به زیرمسائل بدون محدودیت تبدیل می شود. برای رفع مشکلات زیاد بودن تعداد زیرمسائل از روش تولید مسیر^۲ استفاده می شوند. در هر تکرار، تنها مسیرهایی که جواب مثبت دارند کنترل می شوند. آشتیانی [۱] این مسیرها را مسیرهای کاری^۳ نامیده و آن را با K_{rs}^w نشان داد. معیار توقف در روش تولید مسیر به صورت رابطه ۳۲-۲ بیان می شود.

$$\frac{\min_{k \in K_{rs}} c_k^{rs}(f, \alpha) - u_{rs}}{u_{rs}} \leq \varepsilon \quad (32-2)$$

که ε دقت موردنظر می باشد.

حتی در صورت کم بودن تعداد مسیرهای کاری، بدست آوردن مسیرهای کاری با استفاده از الگوریتم های تکمیلی غیرخطی بهینه نیست. آشتیانی [۱] با استفاده از الگوریتم خطی سازی تکراری سعی بر حل این مشکل کرد. با استفاده از این روش، در هر تکرار، مدل در جریان فعلی ($\bar{f}$) خطی می شود. مدل خطی شده شامل فرمول ۳۳-۲ می باشد.

$$\begin{aligned} (C_k^{rs}(\bar{f}, \alpha) + \sum_{k' \in K_{rs}} [(f_{k'} - \bar{f}_{k'}) \frac{\partial c_k^{rs}(\bar{f}, \alpha)}{\partial \bar{f}_{k'}}] - u_{rs}) f_k^{rs} &= 0 \quad \forall k \in K_{rs} \\ C_k^{rs}(\bar{f}, \alpha) + \sum_{k' \in K_{rs}} [(f_{k'} - \bar{f}_{k'}) \frac{\partial c_k^{rs}(\bar{f}, \alpha)}{\partial \bar{f}_{k'}}] - u_{rs} &\geq 0 \quad \forall k \in K_{rs} \end{aligned} \quad (33-2)$$

معیار توقف در این روش برابر با رابطه ۳۴-۲ میباشد.

¹non-linear complementarity problem

² Path generation

³ Working paths

$$\frac{\max_{k \in k_{rs}^w : \bar{f}_k > 0} c_k^{rs}(f, \alpha) - u_{rs}}{\max_{k \in k_{rs}^w : \bar{f}_k > 0} c_k^{rs}(f, \alpha)} \leq \varepsilon \quad (34-2)$$

الگوریتم زمانی تمام می شود که متوسط خطای جواب کنونی کمتر از ε باشد. یعنی:

$$\frac{\sum_{r \in R} \sum_{s \in S} q^{rs} \varepsilon^{rs}}{\sum_{r \in R} \sum_{s \in S} q^{rs}} \leq \varepsilon \quad (35-2)$$

که در آن ε^{rs} برای هر زوج مبدا-مقصد (r, s) به صورت رابطه ۲-۳۶ تعریف میشود.

$$\varepsilon^{rs} = \frac{\left| \max_{k \in k_{rs}^w : \bar{f}_k > 0} c_k^{rs}(f, \alpha) - u_{rs} \right|}{u_{rs}} \quad (36-2)$$

گام های این الگوریتم عبارتست از:

گام (۱) - (مقداردهی/اولیه)

مقادیر مثبتی برای ε و ρ و همچنین بردار λ^0 در نظر بگیرید. برای هر زوج مبدا-مقصد بردار $K_{rs}^w = \phi$ قرار دهید. و سایر پارامترها را به صورت زیر مقداردهی کنید:

$$\begin{aligned} f_k^{rs} &= 0 \\ x_a &= x(f) \\ n &= 0 \end{aligned}$$

گام (۲) - (تجزیه و خطی سازی)

$n = n + 1$ را قرار دهید. برای هر مقصد گام های زیر را انجام دهید.

۱-۲- کوتاه ترین مسیر بین هر زوج مبدا-مقصد را با استفاده از زمان سفر مرحله قبل بدست آورید. زمان سفر بین زوج مبدا-مقصد (I, S) را u_{rs} می نامیم.

۲-۲- برای هر مقصد با تقاضای مثبت گام های زیر را انجام دهید:

۱-۲-۲- اگر رابطه ۲-۳۲ با مقدار $\alpha = \lambda^{n-1}$ برقرار نبود، قرار دهید $k_{rs}^w = k_{rs}^w \cup \{k_{rs}\}$ و $f_{k_{rs}} = 0$

۲-۲-۲- برای $k = k_{rs}^w$ قرار دهید $\bar{f}_k = f_k$

۳-۲-۲- برای بدست آوردن (f^{rs}, u_{rs}) مسیرهای کاری را برای مسئله تکمیلی غیرخطی در نقطه $\bar{f}_k$ بدست آورید.

۲-۲-۴- اگر رابطه ۳۴-۲ برای $\alpha = \lambda^{n-1}$ برقرار نبود، آنگاه به گام ۲-۲-۲ بروید.

۲-۲-۵- برای $\alpha = \lambda^{n-1}$ مقدار ε^{rs} را بیابید.

گام (۳)- (به روز رسانی جریان کمان ها و ضرایب لاگرانژ)

قرار دهید $x^n = x(f)$ و $\lambda^n = P(x^n, \lambda^{n-1})$

گام (۴)- (کنترل توقف)

اگر رابطه ۳۵-۲ برقرار نبود به گام ۲ بروید. در غیر این صورت، اگر رابطه ۲۷-۲ با مقادیر $\alpha^0 = \lambda^0$ و $\alpha = \lambda^{n-1}$ برقرار بود، توقف کنید. در غیر این صورت به گام ۲ بروید.

۳ فصل سوم

مسئله جریان با هزینه منیم در شبکه

۳-۱- مقدمه

از بین الگوریتم های اشاره شده در فصل قبل، الگوریتم فرانک-ولف (که از الگوریتم های بر مبنای کمان است) به علت فهم آسان، سادگی در اجرا، حافظه مصرفی کم و کارایی مناسب در تکرارهای اولیه محبوبیت فراوانی در بین محققان دارد. با این وجود، نتایج این الگوریتم به علت در نظر نگرفتن ظرفیت های مربوط به کمان کمابیش نزدیک به واقعیت نیستند. به همین دلیل، یکی از اهداف اصلی این پژوهش در نظر گرفتن محدودیت ظرفیت کمان ها برای دقیق تر کردن هرچه بیشتر نتایج حاصل از این الگوریتم است.

در این فصل ابتدا مسئله جریان با هزینه مینیمم توضیح داده میشود و سپس به توضیح الگوریتم های بلمن-فورد، دایجسترا، درخت هیپ دوتایی که کوتاه ترین مسیر بین دو گره را با فرض ثابت بودن زمان سفر کمان ها در یک شبکه تعیین می کنند، پرداخته می شود. سپس به توضیح الگوریتم Out of Kilter و همه یا هیچ که جهت جریان در کمان های یک شبکه را مشخص می کنند پرداخته و نتایج حاصل از اجرای الگوریتم Out of Kilter را بر روی دو شبکه با زمان سفر ثابت نشان می دهیم. در انتها نیز به مقایسه زمان حل الگوریتم های بل و دایجسترا برای شبکه شیکاگو می پردازیم.

جدول ۱- الگوریتم های کوتاهترین مسیر

ردیف	نام الگوریتم	پژوهشگر	سال انتشار پژوهش
۱	بلمن-فورد	بلمن،فورد	۱۹۵۶
۲	دایجسترا	دایجسترا	۱۹۵۶
۳	درخت هیپ دوتایی	ویلیامز	۱۹۶۴
۴	Out of Kilter	فولکرسون	۱۹۶۱

۳-۲- مسئله جریان با هزینه مینیمم

مسئله جریان با هزینه مینیمم به شکل رابطه ۳-۱ زیر نوشته می شود:

$$\begin{cases} \min z(x) = \sum_{(i,j) \in N} t_{ij} x_{ij} \\ s.t. \\ \sum_i x_{ij} - \sum_j x_{ji} = b_i \\ x_{ij} \leq c_a \\ x_{ij} \geq 0 \end{cases} \quad (۳-۱)$$

در این مدل x_{ij} جریان در کمان بین گره های i و j ، t_{ij} زمان سفر در کمان بین گره های i و j ، b_i تقاضای جریان از گره i به j و c_a ظرفیت کمان a می باشد. در رابطه ۳-۱ تابع هدف عبارت است از مجموع حاصلضرب زمان سفر هر کمان در جریان همان کمان. در این مدل محدودیت اول بیانگر آن است که تفاضل مجموع جریان در کمان i و کمان j برابر با تقاضای جریان از i به j می باشد. محدودیت دوم برای اطمینان از بیشتر نشدن جریان در هر کمان از ظرفیت آن کمان و محدودیت سوم برای منطقی بودن جواب ها و مثبت بودن جریان در هر کمان می باشد. در صورت در نظر نگرفتن شرط محدودیت ظرفیت در فرمول ۳-۱، مسئله تبدیل به مسئله تخصیص همه یا هیچ تبدیل می شود که انتهای این بخش این مسئله توضیح داده می شود.

زمانی که تابع زمان سفر تابعی ثابت است، یعنی $t_a(x_a) = t_a^0$ ، در این صورت حل مسئله تخصیص تقاضا به مسئله کوتاه ترین مسیر بین زوج های مبدا و مقصد خواهد بود. برای اثبات این مسئله، رابطه ۳-۲ که همان مسئله بکمن است را در نظر بگیرید.

$$\begin{cases} \min z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega \\ s.t. \\ \sum_k f_k^{rs} = q_{rs} & \forall r, s \\ f_k^{rs} \geq 0 & \forall r, s, k \\ x_a = \sum_{(r,s)} \sum_k f_k^{rs} \delta_{ak}^{rs} & \forall a \in A \end{cases} \quad (۳-۲)$$

اگر زمان سفر ثابت باشد، خواهیم داشت:

$$\left\{ \begin{array}{l} \min Z(x) = \sum_a t_a x_a \\ s.t. \\ \sum_k f_k^{rs} = q_{rs} \quad \forall r, s \\ f_k^{rs} \geq 0 \quad \forall r, s, k \\ x_a = \sum_{(r,s)} \sum_k f_k^{rs} \delta_{ak}^{rs} \quad \forall a \in A \end{array} \right. \quad (3-3)$$

با جایگذاری محدودیت ۳ در تابع هدف، خواهیم داشت:

$$\min Z(x) = \sum_a t_a \sum_r \sum_s \sum_k f_k^{rs} \delta_k^{rs} = \sum_r \sum_s \sum_k f_k^{rs} \sum_a t_a \delta_k^{rs}$$

چون عبارت $\sum_a t_a \delta_k^{rs}$ برابر با زمان سفر کوتاه ترین (c_k^{rs}) است، پس عبارت بالا برابر است با:

$$\min Z(x) = \sum_r \sum_s \sum_k f_k^{rs} c_k^{rs}$$

پس مسئله بالا برابر با تخصیص تقاضای جریان به کوتاه ترین مسیر بین زوج های مبدأ و مقصد است.

۳-۳- الگوریتم بلمن-فورد

الگوریتم بلمن-فورد الگوریتم پیمایش گراف است که مسئله کوتاه ترین مسیر از مبدأ واحد را برای گراف های وزن داری که وزن یال ها ممکن است منفی باشد حل می کند. الگوریتم بلمن-فورد راه کاری برای پیدا کردن کم وزن ترین مسیر از رأس مشخص آغاز به بقیه رئوس در گراف جهت دار و وزن دار می دهد. وزن یک مسیر در گراف وزن دار برابر مجموع وزن یال های آن است. جهت دار نبودن یال ها هم مشکلی ایجاد نمی کند و می توان برای یال های غیر جهت دار دو یال فرض کرد.

یکی از مزیت های این الگوریتم نسبت توانایی اجرا شدن روی گراف ها با یال منفی است. اجرای الگوریتم $|V|-1$ دنباله d چنان تعریف می شود که برای هر رأس $|V|$ ، مقدار d_v در پایان مرحله i ام برابر وزن کوتاه ترین گذر از مبدأ به $|V|$ است با این شرط اضافه که تعداد یال های این گذر حداکثر i باشد. بنابراین در پایان مرحله $|V|-1$ ام d_v برابر وزن کوتاه ترین مسیر از مبدأ به $|V|$ خواهد بود. در واقع چون دور با مجموع وزن منفی نداریم، کوتاه ترین گذر با حداکثر $|V|-1$ یال از مبدأ به $|V|$ ، همان کوتاه ترین مسیر از مبدأ به $|V|$ در گراف خواهد بود.

اساس کار الگوریتم آزادسازی همه یال‌های گراف در هر مرحله است. آزادسازی یال (u,v) به این معناست که اگر $d_v < d_u + \text{weight}(u, v)$ آنگاه قرار می‌دهیم $d_v = d_u + \text{weight}(u, v)$ با این اوصاف اگر آزادسازی همه یال‌ها را برای بار $|V|$ ام هم تکرار کنیم و بعد از این مرحله هم دنباله d تغییر کند، آنگاه می‌توان نتیجه گرفت که گراف دور منفی‌ای دارد که از مبدأ قابل دستیابی است.

مشخص است که در $|V|-1$ مرحله، در هر مرحله $|E|$ عملیات بر روی یال‌ها انجام می‌شود. پس m پیچیدگی زمانی الگوریتم بلمن-فورد خواهد بود، که در آن $\bar{n}$ متوسط ورود هر گره و m تعداد کمان‌ها است. در این الگوریتم از دو مجموعه استفاده می‌شود. مجموعه اول مجموعه D است که اعضای آن به صورت d_i نمایش داده شده است و بیانگر کوتاه‌ترین مسیر از مبدأ به گره v_i در پایان هر تکرار الگوریتم است. این مقدار در ابتدا به ازای تمامی گره‌ها برابر ∞ است.

$$\text{For } v_1, v_2, \dots, v_n \Rightarrow d_i = \infty$$

مجموعه دوم مجموعه p است که اعضای آن به صورت p_i نمایش داده شده و در پایان هر تکرار گره پیشین گره v_i را که گره مبدأ از طریق آن به این گره در کوتاه‌ترین مسیر دسترسی دارد، مشخص می‌کند. این مقادیر در ابتدا برای هیچ کدام از گره‌ها تعریف نشده است و در تکرارهای آن مقداردهی می‌شوند.

۳-۴- الگوریتم دایجسترا

الگوریتم دایجسترا یک راهکار برای یافتن کوتاه‌ترین مسیر از مقصد ثابت به سایر گره‌های گراف وزن دار است. این گراف می‌تواند معرف مسیرهای یک شهر و تقاطع‌های آن و هدف این الگوریتم پیدا کردن کوتاه‌ترین مسیر بین گره‌های مختلف این شهر باشد. طبیعتاً این الگوریتم در یافتن کوتاه‌ترین مسیر بین دو گره مشخص نیز کاربرد دارد. تنها شرط لازم استفاده از این الگوریتم نامنفی بودن وزن یال‌هاست. الگوریتم دایجسترا به صورت حریصانه عمل کرده و در تکرارهای متوالی، کوتاه‌ترین مسیر از مبدأ به یکی از گره‌های گراف را بدست می‌آورد. در این الگوریتم از سه مجموعه استفاده می‌شود. مجموعه D که اعضای آن به صورت d_i نمایش داده شده است و بیانگر کوتاه‌ترین مسیر از مبدأ به گره v_i در پایان هر تکرار الگوریتم است. این مقدار در ابتدا به ازای تمامی گره‌ها برابر ∞ است.

$$\text{For } v_1, v_2, \dots, v_n \Rightarrow d_i = \infty$$

مجموعه دوم مجموعه p است که اعضای آن به صورت p_i نمایش داده شده و در پایان هر تکرار گره پیشین گره v_i را که گره مبدأ از طریق آن به این گره در کوتاه‌ترین مسیر دسترسی دارد، مشخص می‌کند. این مقادیر در ابتدا برای هیچ کدام از گره‌ها تعریف نشده است و در تکرارهای آن مقداردهی می‌شود.

شوند. مجموعه سوم مجموعه U است که اعضای آن گره های بررسی نشده در الگوریتم است. این مجموعه در ابتدا شامل تمامی گره های گراف است.

مراحل الگوریتم دایجسترا برای یافتن کوتاه ترین مسیر از گره های V_0 به سایر گره به این شرح است:

گام (۱) - (انتخاب گره مبدا)

گره V_0 را به عنوان گره جاری از U حذف کرده، مقدار d_0 را برابر صفر قرار داده و گام های ۲ تا ۴ را تکرار کنید.

گام (۲) - (محاسبه کوتاه ترین زمان سفر به هر گره)

به ازای هر یال خروجی از گره جاری (v_j) به یک گره از مجموعه U مانند v_k مقدار $d_j + e_{jk}$ را محاسبه کرده و اگر مقدار آن از d_k کوچکتر باشد، مقدار d_k را با $d_j + e_{jk}$ جایگزین کنید.

$$\text{If } d_j + e_{jk} < d_k \Rightarrow d_k = d_j + e_{jk}$$

گام (۳) - (انتخاب گره بعدی)

از مجموعه U گره های عضو گره با کوچکترین d را به عنوان گره جاری انتخاب کنید و از مجموعه U حذف کنید.

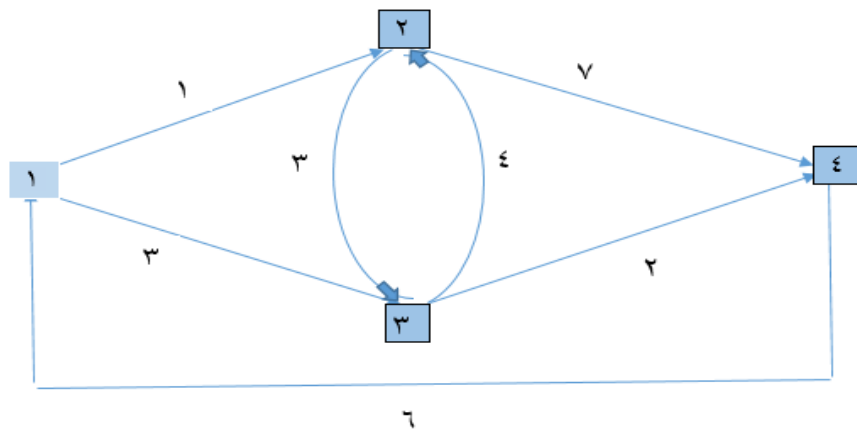
گام (۴) - (کنترل)

اگر U تهی است با در مرحله ۳ هیچ گره ای برای انتخاب به عنوان گره جدید وجود ندارد، به گام بعد بروید.

مقدار d_i رای گره v_i ، کوتاه ترین مسیر از مبدا به آن گره را مشخص کرده است که از طریق گره p_i برای گره مبدا در دسترس است.

مثال

برای توضیح نحوه عملکرد الگوریتم دایجسترا، در شکل ۳-۱ کوتاه ترین مسیر از گره شماره ۱ به سایر گره ها در گراف را بدست می آوریم. این شبکه ۴ گره و ۷ کمان دارد. اعداد روی هر کمان بیان گر زمان سفر آن کمان می باشند.



شکل ۳-۱- شبکه ۴ گره ای الگوریتم دایجسترا

مقادیر d برای تمامی گره ها مطابق جدول ۲ بینهایت قرار داده و همچنین مجموعه U شامل تمامی گره ها می باشد.

جدول ۲- کوتاه ترین مسیر در مرحله ۱

گره	1	2	3	4
کوتاه ترین مسیر از مبدا به گره (d)	∞	∞	∞	∞

$$U = \{1, 2, 3, 4\}$$

گام (۱)- (انتخاب گره مبدا)

گره ۱ را به عنوان گره جاری از U حذف کرده، مقدار d_1 را برابر صفر قرار داده و گام های ۲ تا ۴ را تکرار کنید.

جدول ۳- کوتاه ترین مسیر

گره	1	2	3	4
کوتاه ترین مسیر از مبدا به گره (d)	0	∞	∞	∞

$$U = \{2, 3, 4\}$$

گام (۲) - (محاسبه کوتاه ترین زمان سفر به هر گره)

گره های مجاور گره ۱ را بررسی میکنیم:

$$d_2 = d_1 + t_{12} = 1 + 0 = 1$$

$$d_3 = d_1 + t_{13} = 0 + 3 = 3$$

جدول ۴ - کوتاه ترین مسیر

گره	1	2	3	4
کوتاه ترین مسیر از مبدا به گره (d)	0	1	3	∞

گام (۳) - (انتخاب گره بعدی)

گره با کمترین d (گره ۲) انتخاب میشود:

$$U = \{3, 4\}$$

گام (۲) - (محاسبه کوتاه ترین زمان سفر به هر گره)

زمان گره های منتهی به کمان های خروجی از گره ۲ را محاسبه می کنیم:

$$d_4 = d_2 + t_{24} = 1 + 7 = 8$$

$$d_3 = d_2 + t_{23} = 1 + 3 = 4$$

جدول ۵ - کوتاه ترین مسیر

گره	1	2	3	4
کوتاه ترین مسیر از مبدا به گره (d)	0	1	3	8

گام (۳) - (انتخاب گره بعدی)

گره با کمترین d (گره ۳) انتخاب میشود.

$$U = \{4\}$$

گام (۲)- (محاسبه کوتاه ترین زمان سفر به هر گره)

زمان گره های منتهی به کمان های خروجی از گره ۳ را محاسبه میکنیم:

$$d_2 = d_3 + t_{32} = 3 + 4 = 7$$

$$d_4 = d_3 + t_{34} = 3 + 2 = 5$$

جدول ۶ - کوتاه ترین مسیر

گره	1	2	3	4
کوتاه ترین مسیر از مبدا به گره (d)	0	1	3	5

گام (۳)- (انتخاب گره بعدی)

گره با کمترین d (گره ۴) انتخاب میشود:

$$U = \emptyset$$

گام (۲)- (محاسبه کوتاه ترین زمان سفر به هر گره)

زمان گره های منتهی به کمانهای خروجی از گره ۴ را محاسبه میکنیم:

$$d_1 = d_4 + t_{41} = 5 + 6 = 11$$

جدول ۷ - کوتاه ترین مسیر

گره	1	2	3	4
کوتاه ترین مسیر از مبدا به گره (d)	0	1	3	5

۳-۵- الگوریتم درخت heap

یک درخت دودویی کامل است، هرگاه تمامی سطوح درخت به غیر از احتمالا آخرین سطح پر بوده و برگ‌های سطح آخر از سمت چپ قرار گرفته باشند.

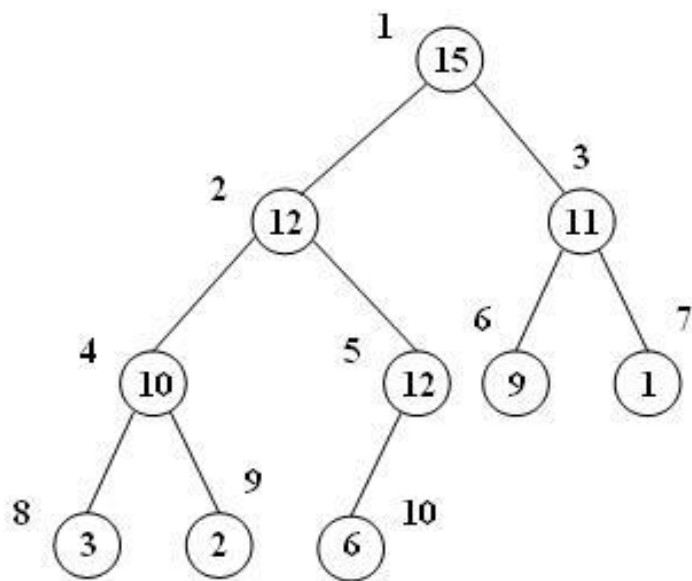
نمایش با استفاده از لیست پیوندی و آرایه دو شکل مشهور نمایش درخت دودویی در ساختمان داده‌ها است. در حالت عادی انتخاب یکی از این دو روش برای نمایش بهینه و با مصرف حافظه‌ی کمتر بسته به چیدمان عناصر درخت دارد. به عنوان مثال، در درخت‌های مورب روش نمایش با آرایه بدترین بازدهی و بیشترین مصرف حافظه را دارد. اما در درخت دودویی کامل این روش در مقایسه با روش لیست پیوندی بسیار بهینه‌تر است. در روش استفاده از آرایه برای نمایش درخت دودویی، گره‌های درخت با شروع از ریشه و در هر سطح از چپ به راست به ترتیب شماره‌گذاری شده و مقدار هر کدام از گره‌ها با توجه به شماره‌ی آن در یکی از خانه‌های آرایه قرار می‌گیرد. در آرایه‌ی متناظر درخت دودویی کامل، از همه‌ی عناصر به صورت کامل استفاده شده و هیچ حافظه‌ی هرزی وجود ندارد. به همین خاطر این روش نمایش برای درخت کامل مناسب است. فرض کنیم توابع $Left$ ، $Parent$ و $Right$ شماره‌ی یک گره را گرفته و به ترتیب شماره‌ی گره والد، فرزند چپ و فرزند راست را برگرداند. در این صورت :

$$\begin{cases} parent(i) = \lfloor \frac{i}{2} \rfloor \\ Left(i) = 2i \\ Right(i) = 2i + 1 \end{cases} \quad (3-4)$$

که منظور از $\lfloor \cdot \rfloor$ جزء صحیح (کف) عدد است.

۳-۵-۱- هرم ماکزیمم (max-heap)

درخت دودویی کاملی است که مقدار هر گره بیشتر یا مساوی فرزندان خود است. شکل ۳-۱ نمایشی از هرم ماکزیمم است.



شکل ۳-۲- درخت هیپ ماکزیمم

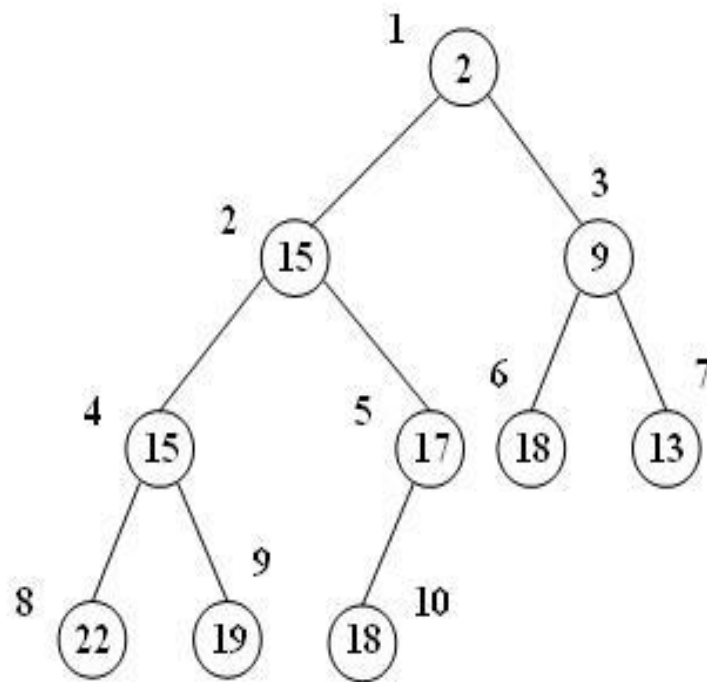
و نمایش آرایه‌ای در جدول ۸ نشان داده شده است.

جدول ۸- مینیمم مسیر گره های درخت هیپ ماکزیمم

Max Heap	1	2	3	4	5	6	7	8	9	10
	15	12	11	10	12	9	1	3	2	6

۳-۵-۲-هرم مینیمم (min-heap)

درخت دودویی کاملی است که مقدار هر گره کمتر یا مساوی فرزندان خود است. شکل ۳-۳ نمایشی از هرم مینیمم است.



شکل ۳-۳- درخت هیپ مینیمم

ساختن یک درخت heap در واقع وارد کردن متوالی گره‌ها در آن است. برای وارد کردن یک گره به درخت heap، طی دو مرحله به صورت زیر عمل می‌کنیم:

گام (۱) - (اضافه کردن گره)

گره مفروض را در محلی از درخت که شرط کامل بودن آن به هم نخورد (بدون در نظر گرفتن شرط بودن) درج می‌کنیم.

گام (۲) - (مرتب سازی گره‌ها)

اگر گره مذکور بر اساس موقعیت خود در درخت، شرط max-heap یا min-heap بودن را نقض نکند، نیاز به انجام کاری نیست و عملیات درج تمام شده است. در غیر اینصورت، با جابجا کردن گره با والد خود، درخت جدیدی حاصل می‌شود که باید مرحله‌ی ۲ در مورد آن تکرار شود.

حذف گره از درخت هیپ عموماً از ریشه‌ی آن صورت می‌گیرد. حذف گره‌ای غیر از گره ریشه، ممکن است هزینه‌ای معادل ساخت مجدد درخت تحمیل کند. چرا که با حذف یک گره غیر ریشه و جایگزین

کردن گرهی دیگر با آن، نه تنها شرط min-heap با max-heap بودن که شرط درخت کامل بودن هم ممکن است نقض شود. اکثر کاربردهای این نوع درخت نیز تنها با حذف گره از ریشه سر و کار دارند.

برای حذف گره ریشه‌ی درخت دو مرحله زیر را اجرا می‌کنیم:

گام (۱) - (حذف گره)

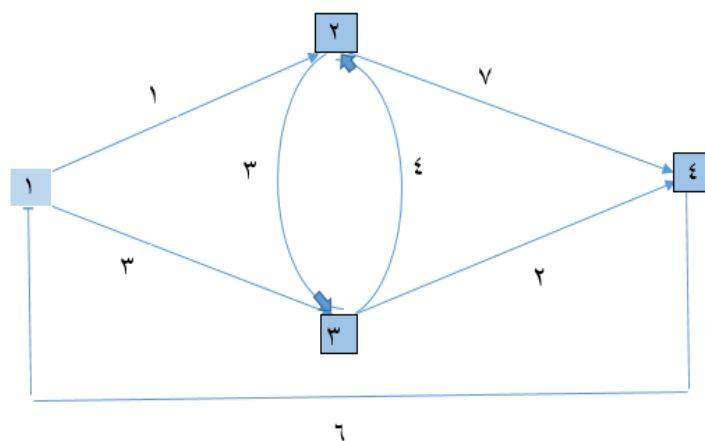
گره ریشه را حذف و سمت راست‌ترین برگ سطح آخر را جایگزین آن می‌کنیم.

گام (۲) - (مرتب سازی گره ها)

در صورتی که گره درج شده جدید شرط heap بودن را نقض نکند عملیات حذف تمام می‌شود. در غیر اینصورت این گره با فرزند مناسب جایگزین شده و این مرحله برای درخت جدید مجدداً اجرا می‌شود. با اجرای مرحله‌ی اول و جایگزین کردن آخرین گره آخرین سطح درخت، شرط کامل بودن درخت پایدار می‌ماند. اما عموماً شرط heap بودن نقض می‌شود. در مرحله‌ی دوم، گره تازه وارد را با یکی از فرزندان خود جایگزین می‌کنیم، تا به شرط heap بودن نزدیک شویم.

مثال

برای توضیح نحوه عملکرد الگوریتم هیپ، در شکل ۳-۴ کوتاه ترین مسیر از گره شماره ۱ به سایر گره ها در گراف را بدست می‌آوریم. این شکل، همان مثال ذکر شده در شکل ۱-۳ می‌باشد و ما درخت هیپ مینم را بدست می‌آوریم.



شکل ۳-۴ - شبکه ۴ گره ای الگوریتم هیپ

گره مبدا ما گره ۱ می باشد . مقدار d را برای گره مبدا برابر با صفر قرار می دهیم. گره های منتهی به کمان های خروجی از گره ۱ شامل گره های ۲ و ۳ می باشند و باید آن ها را به درخت اضافه کنیم.

گام (۱) - (اضافه کردن گره)

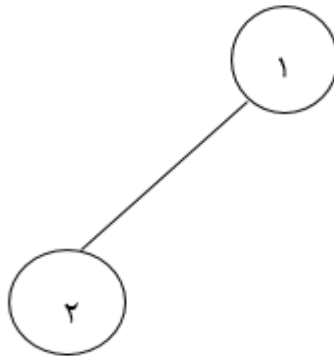
گره ۲ را به درخت اضافه کنیم و بنابراین درخت ما شامل گره های ۱ و ۲ می شود.

گام (۲) - (مرتب سازی گره ها)

مقدار زمان سفر گره ۲ برابر است با:

$$d_2 = d_1 + t_{12} = 1 + 0 = 1$$

چون زمان سفر گره ۲ بیشتر از زمان سفر گره ۱ می باشد، پس اضافه کردن گره ۲ شرط مینیمم بودن درخت را به هم نمیزند.



شکل ۳-۵- درخت مرحله اول الگوریتم هیپ مینیمم

گام (۱) - (اضافه کردن گره)

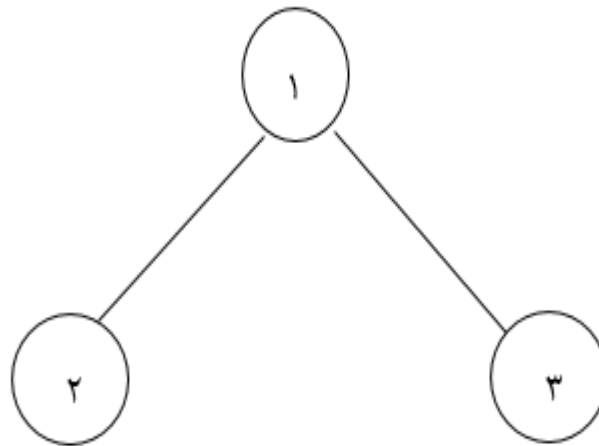
گره ۳ را به درخت اضافه کنیم و درخت ما شامل گره های ۱ و ۲ و ۳ می شود.

گام (۲) - (مرتب سازی گره ها)

مقدار زمان سفر گره ۳ برابر است با:

$$d_3 = d_1 + t_{13} = 0 + 3 = 3$$

زمان سفر گره ۳ از گره ۲ و ۱ بیشتر است . بنابراین شرط مینیمم بودن درخت را به هم نمی زند.



شکل ۳-۶- درخت مرحله دوم الگوریتم هیپ مینیمم

در گام بعد گره ای که مینیمم زمان سفر را دارد (گره ۲) را انتخاب می کنیم. گره ۲ به گره های ۳ و ۴ متصل است. پس باید زمان سفرهای آنها را در این مرحله حساب کنیم.

گام (۱) - (اضافه کردن گره)

زمان سفر گره ۳ برابر است با:

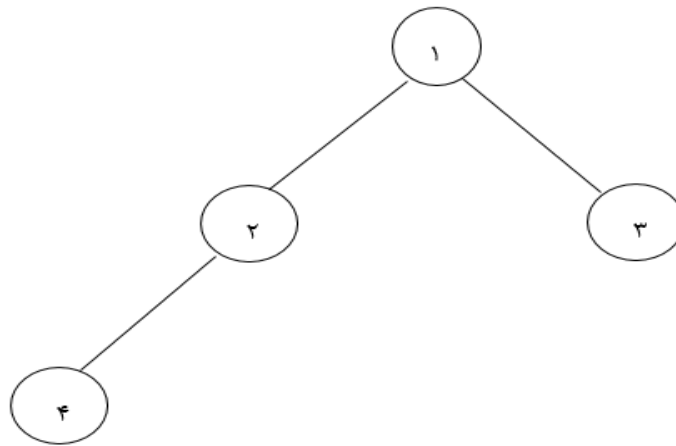
$$d_3 = d_2 + t_{23} = 1 + 3 = 4$$

چون زمان سفر در ای مرحله بیشتر از زمان سفر در مرحله قبل است، آن را تغییر نمی دهیم.

گام (۱) - (اضافه کردن گره)

زمان سفر گره ۴ برابر است با:

$$d_4 = d_2 + t_{24} = 1 + 7 = 8$$



شکل ۳-۷- درخت مرحله سوم الگوریتم هیپ مینیم

گره با مینیم زمان سفر (گره ۳) را انتخاب کرده و کمان های خروجی از آن را بررسی میکنیم.

گام (۱) - (اضافه کردن گره)

گره ۳ به گره های ۲ و ۴ متصل است. پس زمان سفر گره ۲ را بدست می آوریم.

$$d_2 = d_3 + t_{32} = 3 + 4 = 7$$

چون زمان سفر به دست آمده برای گره ۲ در این مرحله بیشتر از زمان سفر قبلی آن است، آن را تغییر نمی دهیم.

گام (۱) - (اضافه کردن گره)

زمان سفر گره ۴ برابر است با:

$$d_4 = d_3 + t_{34} = 3 + 2 = 5$$

چون زمان سفر این گره کاهش یافته است ، زمان سفر گره ۴ را کاهش می دهیم. چون زمان سفر گره ۴ بیشتر از گره ها ۲ و ۳ است، پس نیازی به مرتب سازی گره ها وجود ندارد.

گره ۴ را به عنوان گره بعدی انتخاب میکنیم.

گام (۱) - (اضافه کردن گره)

گره ۴ تنها به گره ۱ متصل است. زمان سفر گره ۱ در این مرحله برابر است با:

$$d_1 = d_4 + t_{41} = 5 + 6 = 11$$

چون زمان سفر به دست آمده برای گره ۱ در این مرحله بیشتر از زمان سفر قبلی آن است، آن را تغییر نمی دهیم

۳-۶- الگوریتم Out of Kilter

جریان در هر شبکه به فرم بسته را می توان به صورت فرمول ۳-۳ بیان کرد.

$$\begin{cases} z = \min \sum_{(i,j) \rightarrow A} C_{ij} X_{ij} \\ s.t. \\ \sum_j X_{ij} - \sum_k X_{kj} = 0 & \forall i = 1, \dots, m \\ l_{ij} \leq X_{ij} \leq q_{ij} \end{cases} \quad (5-3)$$

که در آن C_{ij} هزینه سفر از i به j ، X_{ij} جریان در کمان بین i و j ، l_{ij} حد پایین ظرفیت کمان بین i و j و q_{ij} حد بالای ظرفیت کمان بین i و j است.

فرض کنید $u \in R^m$ باشد. شرط بهینگی عبارتست از :

$$\sum_{(i,j) \in A} X_{ij} - \sum_{(k,i) \in A} X_{ki} = 0 \quad (6-3)$$

$$\begin{cases} \overline{C_{ij} - u_i - u_j} < 0 \rightarrow X_{ij} = q_{ij} \\ \overline{C_{ij} - u_i - u_j} = 0 \rightarrow l_{ij} \leq X_{ij} \leq q_{ij} \\ \overline{C_{ij} - u_i - u_j} > 0 \rightarrow l_{ij} = X_{ij} \end{cases} \quad (7-3)$$

که u_i برابر با کمترین هزینه گره i است.

برای مساله جریان در شبکه به فرم بسته، این الگوریتم در هر تکرار شرط رابطه ۳-۶ را حفظ می کند و به دنبال برقراری شرط رابطه ۳-۷ است. این الگوریتم با یک X که در شرط رابطه ۳-۶ صدق کند و یک u دلخواه شروع میکند. کمان هایی که در شرط رابطه ۳-۷ صدق میکنند in-kilter (در شرایط مناسب) و در غیر این صورت Out of Kilter نامیده می شوند.

جدول ۹ مقیاسی از میزان OFK بودن هر کمان را نشان میدهد، که مقدار Kilter نامیده می شود. اگر این عدد برای یک کمان صفر باشد کمان in-kilter و اگر مثبت باشد OFK است. M یک عدد بزرگ است.

جدول ۹ - اعداد کیلتر

	$X_{ij} < l_{ij}$	$X_{ij} = l_{ij}$	$l_{ij} < X_{ij} < q_{ij}$	$X_{ij} = q_{ij}$	$X_{ij} > q_{ij}$
$\bar{C}_{ij} < 0$	$-\bar{C}_{ij} (q_{ij} - X_{ij})$	$-\bar{C}_{ij} (q_{ij} - X_{ij})$	0	0	$-\bar{C}_{ij} (X_{ij} - q_{ij})$
$\bar{C}_{ij} = 0$	$M (l_{ij} - X_{ij})$	0	0	0	$M (X_{ij} - q_{ij})$
$\bar{C}_{ij} > 0$	$\bar{C}_{ij} (l_{ij} - X_{ij})$	0	$\bar{C}_{ij} (X_{ij} - l_{ij})$	$\bar{C}_{ij} (X_{ij} - l_{ij})$	$\bar{C}_{ij} (X_{ij} - l_{ij})$

به دو طریق میتوان مقدار Kilter یک کمان را به نحوی که این مقادیر برای سایر کمان ها اضافه نشود کاهش داد. روش اول تغییر جریان است که در آن X_{ij} به نحوی تغییر می کند که شرط رابطه ۳-۶ باقی بماند و مقادیر Kilter سایر کمان ها اضافه نشود. روش دوم تغییر هزینه های تقلیل یافته است که عبارت است از تغییر مقادیر $\bar{C}_{ij}$ (یا مقادیر u_i) به نحوی که مقادیر Kilter سایر کمان ها اضافه نشود. در جدول ۱۰ جهت جریان (X_{ij}) به نحوی که مقادیر Kilter کمان اضافه نشود، نشان داده شده است.

جدول ۱۰ - جهت مطلوب تغییر جریان

	$X_{ij} < l_{ij}$	$X_{ij} = l_{ij}$	$X_{ij} = l_{ij}$	$X_{ij} = q_{ij}$	$X_{ij} > q_{ij}$
$\bar{C}_{ij} < 0$	افزایش q_{ij}	افزایش q_{ij}	افزایش q_{ij}	ثابت	کاهش q_{ij}
$\bar{C}_{ij} = 0$	افزایش q_{ij}	ثابت q_{ij}	ثابت l_{ij} افزایش q_{ij}	ثابت l_{ij}	کاهش l_{ij}
$\bar{C}_{ij} > 0$	افزایش q_{ij}	ثابت	افزایش l_{ij}	کاهش l_{ij}	کاهش l_{ij}

در جدول ۱۱ جهت تغییر هزینه تقلیل به نحوی که مقدار Kilter کمان اضافه نشود نشان داده شده است:

جدول ۱۱- جهت مطلوب تغییر هزینه تقلیل یافتن

	$X_{ij} < l_{ij}$	$X_{ij} = l_{ij}$	$X_{ij} = l_{ij}$	$X_{ij} = q_{ij}$	$X_{ij} > q_{ij}$
$\bar{C}_{ij} < 0$	$+\infty$ افزایش	$+\infty$ افزایش	0 افزایش	$-\infty$ ثابت	$-\infty$ کاهش
$\bar{C}_{ij} = 0$	$+\infty$ افزایش	$+\infty$ ثابت	ثابت	$-\infty$ ثابت	$-\infty$ کاهش
$\bar{C}_{ij} > 0$	$+\infty$ افزایش	$+\infty$ ثابت	0 کاهش	$-\infty$ کاهش	$-\infty$ کاهش

گام های این الگوریتم به صورت زیر می باشد:

گام ۰- (مقدار دهی / اولیه)

یک جریان دلخواه که در شرط رابطه ۳-۶ صدق کند و یک شبه قیمت دلخواه انتخاب کنید.

گام (۱)- (کنترل)

اگر تمام کمانها in-kilter هستند، توقف کنید ، وگرنه یک کمان OFK مثل (k,l) انتخاب کنید و به گام (۲) بروید.

گام (۲)- (تغییر جریان)

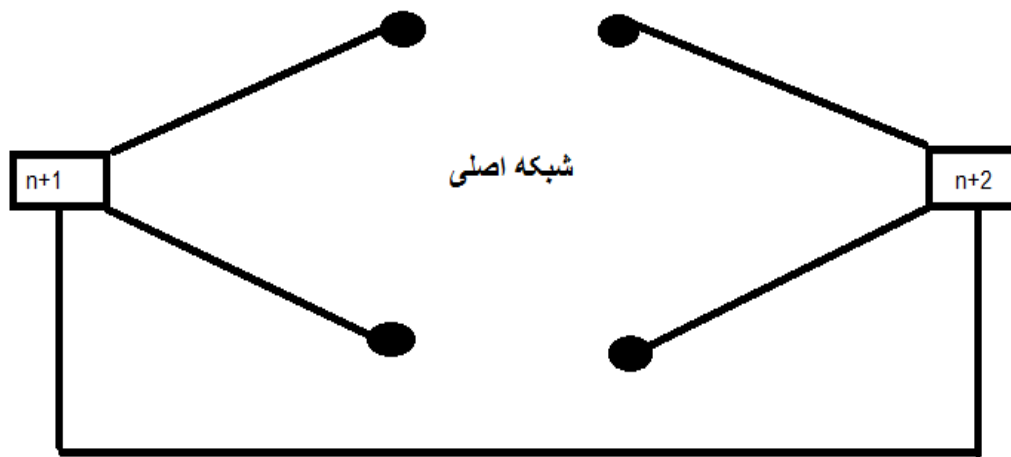
از گره l شروع به ساختن درختی کنید که در کمان های آن بتوان جریان را تغییر داد (در جهت مطلوب) به طوریکه مقدار Kilter هیچ کمانی اضافه نشود. اگر به گره k رسیدید، یک مسیر تغییر جریان پیدا شده که می توان در آن مسیر و کمان (k,l) را به میزان θ تغییر داد و شرط رابطه ۳-۶ را حفظ کرد. به گام (۱) بروید. اگر به گره k نرسیدید به گام ۳ بروید.

گام (۳)- (تغییر هزینه تقلیل یافته)

اگر مجموعه گره های درخت را با N_1 و بقیه را N_2 نشان دهید، کمان های بین N_1 و N_2 در یکی از حالات جدول ۹ هستند. زیرا در غیر ای صورت به درخت اضافه می شوند، در این صورت میتوان $\bar{C}_{ij}$ این کمان ها را کاهش (یا افزایش) داد ، که معادل افزایش (یا کاهش) u_i مربوطه به گره های در N_1 است، بدون اینکه مقادیر Kilter آنها اضافه شود. با تغییر u_i ها به اندازه کافی بزرگ (δ) یا کمان $k-l$ به In-kilter تبدیل می شود که در این صورت به گام (۱) بروید، یا یک کمان به درخت اضافه می شود که در این صورت به گام (۲) بروید. در غیر این صورت حل مساله امکان ناپذیر است، توقف کنید.

از آنجا که پس از هر تغییر جریان یا تغییر هزینه تقلیل یافته، مقادیر Kilter حداقل یک کمان کاهش می یابد، الگوریتم پس از تعداد محدودی تکرار متوقف می شود.

این الگوریتم زمانی از کارایی مناسب برخوردار خواهد بود که از اطلاعات مربوطه به درختی که در یک تکرار ساخته می شود، در تکرارهای بعدی هم استفاده شود. برای این منظور، میتوان یک جنگل گسترش که زیر شاخه های آن نشان دهنده مسیرهای تغییر جریان باشند ساخت و بعد از هر تغییر جریان یا انتخاب کمان OFK آن را بهنگام کرد. همچنین نحوه نمایش ساختار جنگل گسترش، جهت سهولت حرکت آن ، نقش مهمی در کارایی الگوریتم دارد. در شبکه هایی که به فرم بسته نیستند و بردار عرضه و تقاضا در آنها صفر نباشد، برای تبدیل مساله به فرم بسته میتوان دو گره مصنوعی و چند کمان مصنوعی مطابق شکل زیر به شبکه اضافه کرد.



شکل ۳-۸- شبکه مطلوب الگوریتم OFK

اطلاعات کمان های مصنوعی یعنی (q_{ij}, l_{ij}, c_{ij}) را میتوان به شکل های مختلفی تعریف کرد. تجربه نشان داده است که اگر این اطلاعات به نحوه رابطه ۳-۸ تعریف شوند، به طوریکه فقط کمان های مصنوعی مربوط به گره های عرضه OFK شوند، الگوریتم OFK به نحو کاراتری عمل خواهد کرد. در رابطه ۳-۸ عبارت اول برای کمان های متصل به گره های مبدا، رابطه دوم برای کمان های متصل به گره های مقصد و رابطه سوم برای کمان متصل کننده گره مجازی مبدا به گره مجازی مقصر می باشد.

$$\begin{cases} (q_{ij}, l_{ij}, c_{ij}) = (b_i, b_i, 0) \\ (q_{ij}, l_{ij}, c_{ij}) = (-b_i, 0, 0) \\ (q_{ij}, l_{ij}, c_{ij}) = (\sum_{b_i > 0} b_i, 0, 0) \end{cases} \quad (۳-۸)$$

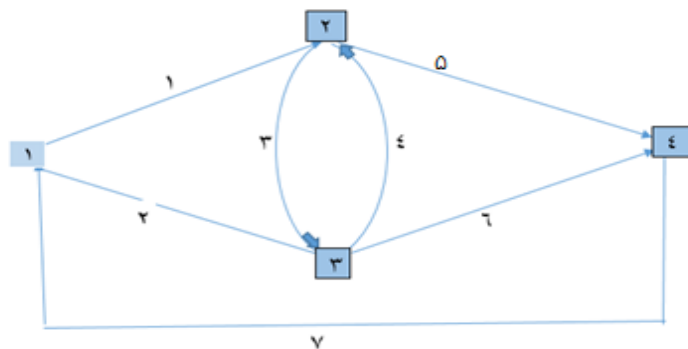
معیار همگرایی در این الگوریتم برای کمان های با زمان سفر متغیر به صورت رابطه ۳-۹ بیان می شود

$$RG = \frac{\sum_a t_a x_a - \sum_a t_a y_a}{\sum_a t_a x_a} \quad (9-3)$$

در این رابطه متغیرهای x_a و y_a همان متغیرهای الگوریتم فرانک-ولف که در رابطه ۲-۱۲ تعریف شده اند، می باشند.

مثال

برای درک بیشتر از نحوه به دست آوردن کوتاه ترین مسیر در یک شبکه مثالی از الگوریتم OFK را در شبکه شکل ۳-۹ زیر با ۴ گره و هفت کمان آورده شده است. اعداد روی هر کمان شماره کمان ها را مشخص میکند و همانطور که قبلاً توضیح داده شده است q حد بالای ظرفیت کمان، l حد پایین ظرفیت کمان، C ظرفیت کمان و x جریان در هر کمان است. اطلاعات مسئله شامل حد بالا و پایین ظرفیت، هزینه، جریان اولیه کمان ها و پتانسیل اولیه گره ها در جدول ۱۲ و ۱۳ آمده است.



شکل ۳-۹- شبکه ۴ گره ای الگوریتم OFK

جدول ۱۲- اطلاعات کمان های شبکه

کمان	1	2	3	4	5	6	7
حد بالای ظرفیت (q)	2	4	1	1	4	2	3
حد پایین ظرفیت (l)	0	0	0	0	0	1	3
هزینه (C)	2	5	1	1	3	6	1
جریان اولیه (x)	1	2	0	1	2	1	3

جدول ۱۳- اطلاعات گره های شبکه

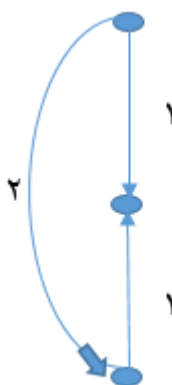
گره	1	2	3	4
کمترین هزینه گره (u)	2	0	0	0

گام (۱)-(کنترل)

با محاسبه عددهای Kilter و با استفاده از فرمول ۳-۷ کمان های ۲ و ۴ و ۵ کمان های OFK و بقیه کمان ها In-kilter هستند. کمان ۲ را انتخاب می کنیم که جریان در آن باید کاهش یابد. در نتیجه یک مسیر افزایش از ۱ به ۳ باید پیدا کنیم.

گام (۲)-(تغییر جریان)

از گره ۱ شروع به ساختن درخت افزایش می کنیم. (می توان از همان درخت مرحله قبل شروع کرد). درخت جدید به صورت زیر می باشد.



شکل ۳-۱۰- درخت مرحله اول الگوریتم OFK

در نتیجه یک مسیر افزایش پیدا شده ، $\theta = 1$ می شود و جریان های جدید عبارتند از:

جدول ۱۴- جریان های بروز شده

کمان	1	2	3	4	5	6	7
x	2	1	0	0	2	1	3

گام (۱) - (کنترل)

کمان ۲ هنوز OFK است و جریان در آن باید کاهش یابد.

گام (۲) - (تغییر جریان)

از گره ۱ شروع به ساختن درخت می کنیم. چون گره ای به درخت اضافه نمی شود به گام ۳ می رویم.

گام (۳) - (تغییر هزینه تقلیل یافته)

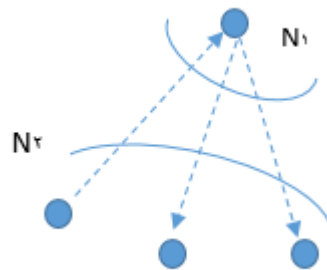
اگر مجموعه گره های درخت را با N_1 و بقیه را N_2 نشان دهید، کمان های بین N_1 و N_2 در یکی از حالات جدول ۱-۲ هستند. زیرا در غیر ای صورت به درخت اضافه می شوند، در این صورت میتوان $\bar{C}_{ij}$ این کمان ها را کاهش (یا افزایش) داد، که معادل افزایش (یا کاهش) u_i مربوطه به گره های در N_1 است، بدون اینکه مقادیر Kilter آنها اضافه شود. با تغییر u_i ها به اندازه کافی بزرگ (δ) یا کمان $k-1$ به In-kilter تبدیل می شود.

$N_1 = \{1\}$ و $N_2 = \{2,3,4\}$ است. کمان های بین دو مجموعه N_1 و N_2 روی شکل نشان داده شده است. در نتیجه:

$$\delta = \min \{3, \infty\} = 3 \Rightarrow u_1 = 5, u_2 = u_3 = u_4 = 0$$

جدول ۱۵- پتانسیل های بروز شده

گره	1	2	3	4
u	5	0	0	0



شکل ۳-۱۱- درخت مرحله دوم الگوریتم OFK

کمان ۲ In-kilter می شود. به گام (۱) بروید.

گام (۱)-(کنترل)

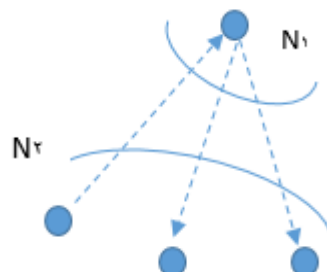
تنها کمان ۵ در حالت OFK است و جریان در آن باید کاهش یابد. در تکرارهای بعدی، نمی توان مسیرهای افزایش جریان پیدا کرد و تنها هزینه ها به صورت زیر تغییر می یابد.

گام (۳)-(تغییر هزینه تقلیل یافته)

$$\delta = \min \{1, \infty\} = 1 \Rightarrow u_1 = 5, u_2 = 1, u_3 = u_4 = 0$$

-پتانسیل های بروز شده ۱۶ جدول

گره	1	2	3	4
u	5	1	0	0



شکل ۳-۱۲- درخت مرحله سوم الگوریتم OFK

گام (۱)-(کنترل)

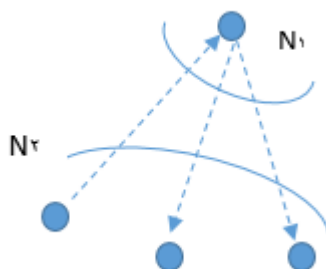
کمان ۵ در تکرار بعدی نیز OFK است.

گام (۳)-(تغییر هزینه تقلیل یافته)

$$\delta = \min \{2, \infty\} = 2 \Rightarrow u_1 = 7, u_2 = 3, u_3 = 2, u_4 = 0$$

جدول ۱۷-پتانسیل های بروز شده

گره	1	2	3	4
u	7	3	2	0



شکل ۳-۱۳-درخت مرحله چهارم الگوریتم OFK

گام (۱)-(کنترل)

تمامی کمان ها In-kilter هستند. توقف کنید.

مثال

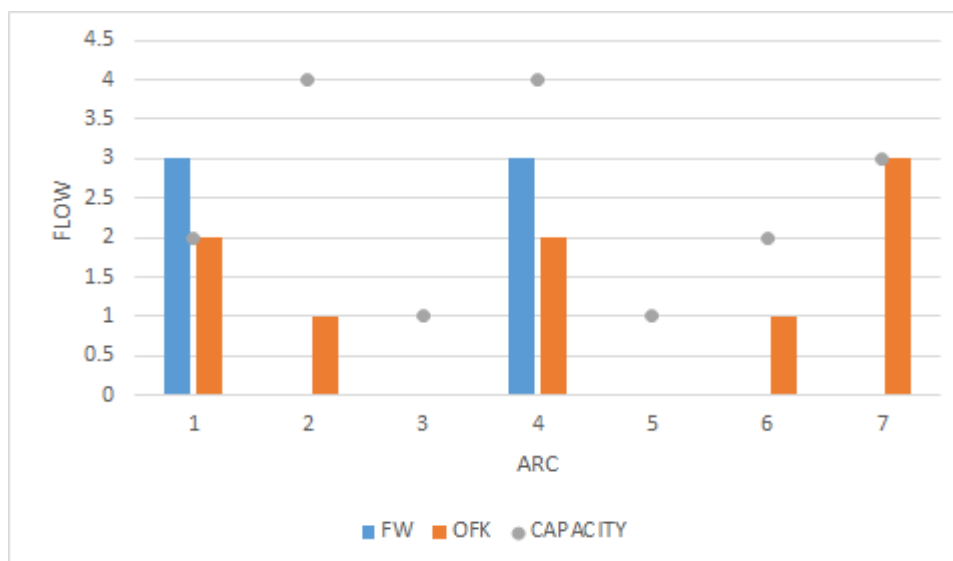
برای مقایسه نتایج الگوریتم های Out of Kilter و فرانک-ولف ، شبکه شکل ۳-۹ را در نظر بگیرید.

نتایج حاصل از اجرای الگوریتم های Out of Kilter و فرانک-ولف در جدول ۱۸ و شکل ۳-۱۵ نشان داده شده است. مقایسه شکل ۳-۱۴ به وضوح نشان می دهد که الگوریتم فرانک-ولف با در نظر نگرفتن ظرفیت نتایج غیر منطقی از جریان در کمانها را به ما می دهد. کمان ۱ ظرفیتی برابر با ۲ واحد دارد . در حالی که در الگوریتم فرانک-ولف جریان اختصاص داده شده به این کمان ۳ واحد است که در این صورت جریان از ظرفیت بیشتر است و هزینه سفر به بی نهایت میل می کند. اما الگوریتم فرانک-ولف هزینه سفر

در این کمان را ۱ واحد به دست داده است. این نتایج باعث غیر صحیح و غیر منطقی شدن تحلیل های انجام شده می شود. الگوریتم Out of kilter با در نظر گرفتن ظرفیت هر کمان، جریان در هر کمان را با توجه به ظرفیت آن تنظیم می کند و اگر جریان اولیه ورودی به یک کمان از ظرفیت آن بیشتر شود آن را بین کمان های دیگر پخش می کند. یکی از دلایلی که معمولاً تعداد کمان های بیشتری در الگوریتم Out of Kilter دارای جریان می شوند نیز همین تنظیم جریان کمان ها می باشد. لازم به ذکر است که در الگوریتم OFK با هزینه سفر ثابت چون میزان جریان از ظرفیت بیشتر نمی شود هزینه سفر برابر با هزینه ثابت کمان می باشد. جدول ۱۹ نیز به مقایسه هزینه کل دو الگوریتم که برابر با مجموع حاصلضرب هزینه در جریان می باشد، می پردازد. همانطور که مشاهده می شود هزینه کل برای الگوریتم OFK بیشتر از الگوریتم فرانک-ولف است که به علت اضافه شدن یک محدودیت به تابع هدف می باشد. مشاهده می شود که تعداد کمان های دارای جریان در الگوریتم OFK برابر با ۵ کمان و در الگوریتم FW برابر با ۲ کمان است که این نشان از توزیع جریان در الگوریتم OFK می باشد.

جدول ۱۸- نتایج بدست آمده از الگوریتم های FW و OFK

کمان	FW	جریان OFK	محدودیت ظرفیت (q)
1	3	2	2
2	0	1	4
3	0	0	1
4	3	2	4
5	0	0	1
6	0	1	2
7	0	3	3



شکل ۳-۱۴- مقایسه جریان کمان هابرای الگوریتم های FW و OFK

جدول ۱۹-مقایسه هزینه کل بدست آمده از دو الگوریتم

	Frank-wolfe	OFK
هزینه کل	9	20

مثال

برای مقایسه بهتر دو الگوریتم، دو الگوریتم را بر شبکه ای با ۱۲ گره و ۲۱ کمان نیزمقایسه میکنیم. در جدول ۲۰ اطلاعات مورد نیاز درباره این شبکه را مشاهده میکنید. همچنین جدول ۲۱ اطلاعات عرضه و تقاضا را در گره های این شبکه نشان می دهد.

جدول ۲۰- اطلاعات شبکه ۱۲ گره ای

کمان a	هزینه (C_a)	حد پایین ظرفیت (q_a)	حد پایین ظرفیت (l_a)	جریان اولیه (X_a)
1	0	8	8	0
2	0	7	7	0
3	0	5	5	0
4	2	10	0	0
5	3	5	0	0
6	2	5	0	0
7	1	5	0	0
8	2	4	0	0
9	4	5	0	0
10	2	4	0	0
11	5	10	0	0
12	2	4	0	0
13	2	4	0	0
14	3	5	0	0
15	3	10	0	0
16	1	3	0	0
17	1	3	0	0
18	2	5	0	0
19	0	7	7	0
20	0	6	6	0
21	0	7	7	0

جدول ۲۱- عرضه و تقاضا در مبدأ-مقصد شبکه

مبدأ-مقصد	12
1	20

در این مثال به این دلیل که شبکه به فرم چرخشی نیست یک کمان مجازی خروجی از گره ۲۲ و ورودی به گره ۱ را به شبکه اضافه کرده ایم که اطلاعات خروجی آن نشان نداده شده است.

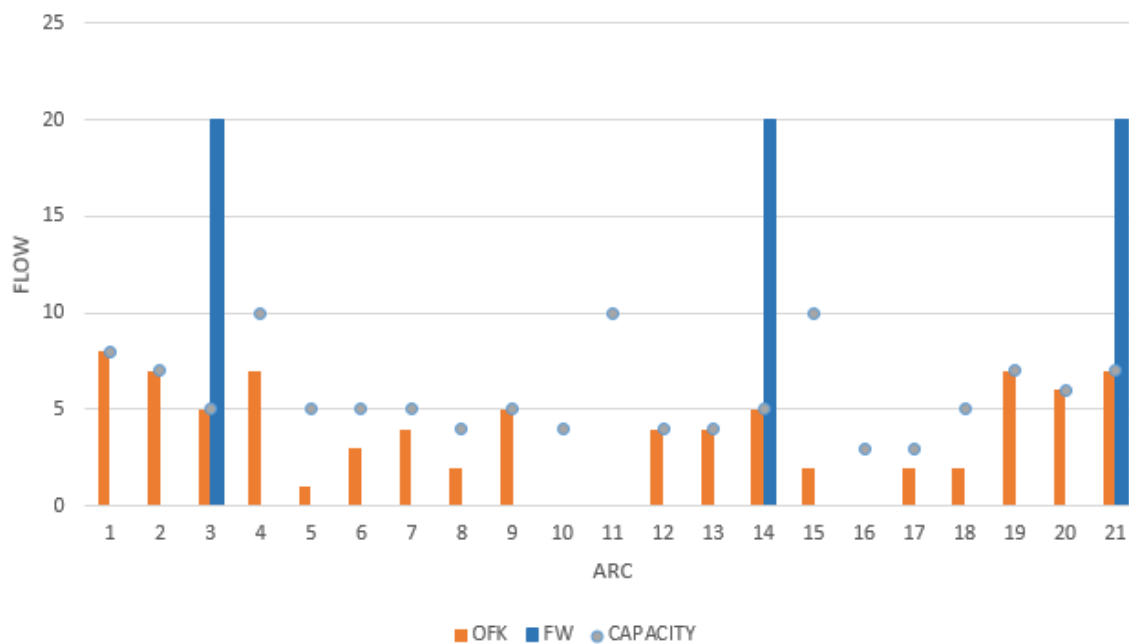
شکل ۳-۱۵ مقایسه بین جریان های کمان و ظرفیت هر کمان را به صورت نمودار نشان می دهد و جدول ۲۲ نتایج حاصل از اجرای الگوریتم های Out of Kilter و فرانک-ولف بر روی شبکه را نشان می دهند. مانند مثال قبل در الگوریتم فرانک-ولف جریان در کمان های ۳، ۱۴ و ۲۱ بیشتر از ظرفیت کمان ها می باشد. همچنین مقایسه تعداد کمان های دارای جریان در شکل ۳-۱۶ نیز نشان می دهد که تعداد کمان های دارای جریان در الگوریتم Out of Kilter بسیار بیشتر از تعداد کمان های دارای جریان در الگوریتم فرانک-ولف می باشد. هر چه مقیاس شبکه بزرگتر باشد، تفاوت در تعداد کمان های دارای جریان بیشتر می شود.

جدول ۲۳ نیز به مقایسه هزینه کل دو الگوریتم می پردازد که مانند مثال قبل هزینه کل برای الگوریتم فرانک-ولف کمتر از الگوریتم OFK می باشد.

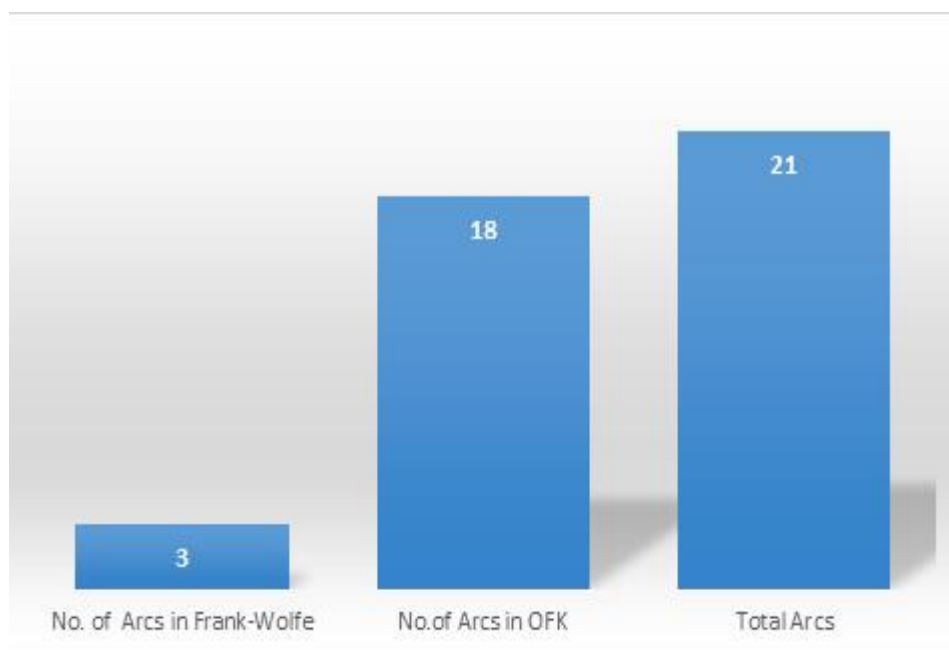
جدول ۲۲-مقایسه نتایج بدست آمده از الگوریتم های FW و OFK

کمان	جریان فرانک-ولف	جریان OFK	محدودیت ظرفیت (q)
1	0	8	8
2	0	7	7
3	20	5	5
4	0	7	10
5	0	1	5
6	0	3	5
7	0	4	5

8	0	2	4
9	0	5	5
10	0	0	4
11	0	0	10
12	0	4	4
13	0	4	4
14	20	5	5
15	0	2	10
16	0	0	3
17	0	2	3
18	0	2	5
19	0	7	7
20	0	6	6
21	20	7	7



شکل ۳-۱۵- مقایسه جریان در کمان ها برای الگوریتم های FW و OFK



شکل ۳-۱۶- مقایسه تعداد کمان های دارای جریان برای الگوریتم های FW و OFK

-مقایسه هزینه کل بدست آمده از دو الگوریتم ۲۳ جدول

	Frank-wolfe	OFK
هزینه کل	60	94

۳-۷- مسئله تخصیص همه یا هیچ

در این مسئله فرض بر این است که زمان سفر کمان ها ثابت و مستقل از جریان در آنها است و همچنین تمام استفاده کنندگان اطلاع دقیقی از زمان سفر در کمان ها دارند. بر اساس این فرضیات و این واقعیت که استفاده کنندگان مسیری را انتخاب می کنند که زمان سفر آنها را مینیمم می کند، این مسئله تمام جریان بین یک مبدا و مقصد مشخص را به مسیری که کمترین هزینه سفر را دارد اختصاص میدهد.

در این روش هر بار یک زوج مبدا-مقصد مثل (I, S) انتخاب می شود و سپس همه تقاضای جریان به کوتاه ترین مسیر از I به S تخصیص داده می شود. (یعنی جریان در کمان های روی کوتاه ترین مسیر به اندازه جریان تخصیص داده شده) افزایش می یابد. این روش وقتی کاربرد دارد که میزان تراکم در شبکه بسیار کم باشد زیرا زمان سفر در کمان ها را ثابت و مستقل از جریان در نظر گرفته ایم.

حال اگر بخواهیم مسئله مطرح شده در شکل ۳-۱۰ را با الگوریتم همه یا هیچ حل کنیم، فرض میکنیم تقاضا از گره ۱ به گره ۴ برابر با ۱۰ واحد است. همانطور که توسط الگوریتم های بلمن-فورد، دایجسترا و هیپ نشان دادیم کوتاه ترین مسیر از گره ۱ به ۴ به ترتیب از گره های ۱، ۳ و ۴ میگذرد. بنابراین تمام ۱۰ واحد جریان به این مسیر اختصاص می یابد.

۳-۸- مقایسه الگوریتم های بل و دایجسترا

الگوریتم های کوتاه ترین مسیر دارای مرتبه های مختلفی برای حل مسئله دارند. همان طور که در فصل سوم اشاره شد الگوریتم دایجسترا از مرتبه $O(n^2)$ و الگوریتم بل از مرتبه $O(\bar{n} * m)$ است که $\bar{n}$ برابر با متوسط ورود هر گره به الگوریتم می باشد. گلدن [۱] در سال ۱۹۷۶ نشان داده است که برای شبکه های دارای کمتر از ۱۰۰۰ گره مقدار $\bar{n}$ به طور متوسط برابر ۲ می باشد. او در این پژوهش به مقایسه زمان حل الگوریتم های بل و دایجسترا درختی که از مرتبه $O(n * \log n)$ پرداخته است و نتیجه گیری کرده است که زمان حل الگوریتم بل سریع تر از الگوریتم دایجسترا درختی میباشد. نتایج حاصل از اجرای الگوریتم بل بر روی شبکه با شیکاگو ۱۲۹۸۲ گره نشان داده است که مقدار متوسط $\bar{n}$ برای این شبکه برابر با ۴,۳

می باشد. جدول ۲۴ زمان حل مسئله کوتاه ترین مسیر را برای ۱۴ تکرار در الگوریتم های بل و دایجسترا مقایسه می کند. لازم به ذکر است که ساختار الگوریتم دایجسترا به گونه ای است که در هر تکرار ، هر گره فقط یک بار کنترل می شود.

جدول ۲۴-مقایسه زمان حل الگوریتم های بل و دایجسترا برای شبکه شیکاگو

تکرار	الگوریتم بل			الگوریتم دایجسترا
	متوسط ورود هر گره برای الگوریتم	زمان حل هر تکرار	تعداد ورود هر گره	زمان حل هر تکرار
0	4.360036974	0.024000	56602	23.024700
1	5.712139886	0.031000	74155	22.046100
2	3.416807888	0.019000	44357	25.018900
3	5.41341858	0.029000	70277	22.013900
4	4.589431521	0.024000	59580	23.058700
5	3.911955015	0.021000	50785	24.014900
6	4.186951163	0.023000	54355	25.072700
7	4.440224927	0.024000	57643	23.034600
8	3.343013403	0.018000	43399	22.096500
9	4.274302881	0.023000	55489	23.023700
10	3.604914497	0.020000	46799	24.029400
11	5.317516561	0.029000	69032	23.055600
12	4.4450778	0.024000	57706	22.087700
13	3.271914959	0.018000	42476	23.076200

همان طور که مشاهده میشود مقدار $\bar{n}$ برای شبکه بزرگ مقیاس شیکاگو بزرگ تر از عدد ۲ که در مقاله گلدن تخمین زده شده است، می باشد. بنابراین برای شبکه های واقعی مقایسه الگوریتم بل و دایجسترا درختی در مقاله گلدن دقیق نمی باشد. برای واقعی و دقیق شدن نتایج الگوریتم کوتاه ترین مسیر، استفاده از الگوریتم دایجسترا درختی برای شبکه های بزرگ مقیاس مانند شبکه شیکاگو در پژوهش های آینده پیشنهاد می شود.

۴ فصل چهارم

اثر اعمال محدودیت ظرفیت کمان بر مسئله تخصیص ترافیک

۴-۱- مقدمه

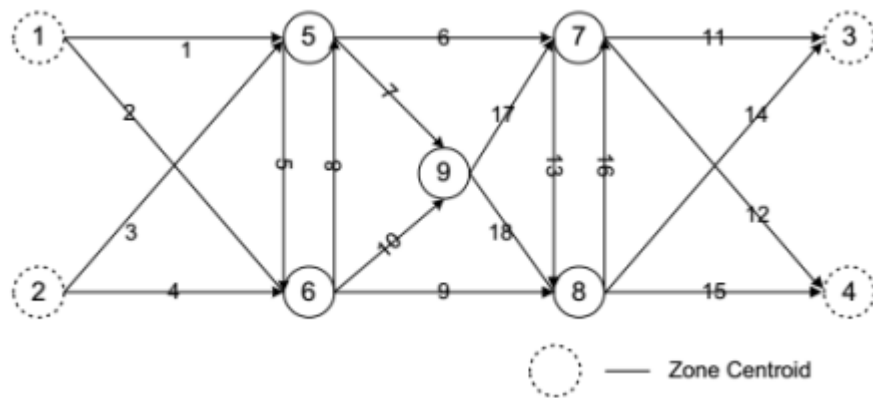
همان طور که اشاره شد در نظر گرفتن ظرفیت برای کمان های یک شبکه حمل و نقلی باعث دقیق و واقعی تر شدن نتایج به دست آمده از تحلیل شبکه می شود. به دلیل پیچیدگی های ناشی از در نظر گرفتن ظرفیت برای کمان ها، تاکنون پژوهش های نسبتاً کمی در این زمینه انجام گرفته است. در این فصل ابتدا به معرفی شبکه های حمل و نقلی که تاکنون مسئله تخصیص ترافیک با محدودیت ظرفیت بر آنها اعمال شده است می پردازیم. سپس به توضیح ساختارهای استفاده شده در کد برنامه نویسی الگوریتم OFK میپردازیم. در نهایت نتایج حاصل از الگوریتم Out of Kilter بر روی شبکه های هرن و سوفالز با هزینه سفر وابسته به جریان مورد بررسی و تحلیل قرار می گیرند. لازم بذکر است در تعریف ساختارهای مورد استفاده در کد برنامه نویسی الگوریتم OFK از پایان نامه آشتیانی [۵۱] استفاده شده است.

۴-۲- شبکه های حمل و نقل

شبکه های متعددی در مسئله تخصیص ترافیک در نظر گرفته شده و مورد بررسی و تحلیل قرار گرفته اند. در ادامه به معرفی ویژگی های سه شبکه که بیشتر از سایر شبکه در مسئله تخصیص ترافیک شناخته شده اند می پردازیم. همانطور که گفته شد مسئله تخصیص ترافیک با محدودیت ظرفیت دارای پیچیدگی های ریاضی و برنامه نویسی می باشد به همین دلیل شبکه هایی که در این مسئله به کار گرفته شده اند معمولاً کوچک مقیاس می باشند.

۴-۲-۱- شبکه هرن

یکی از محبوب ترین شبکه های به کار گرفته شده در مسئله تخصیص ترافیک با محدودیت ظرفیت، شبکه هرن و ریبلا [۳۳] می باشد. این شبکه از ۹ گره، ۱۸ کمان و ۴ زوج مبدا-مقصد تشکیل شده است. نمایی از این شبکه را در شکل ۴-۱ مشاهده می کنید. تقاضا بین زوج های مبدا-مقصد [۴۵] در این شبکه مطابق جدول ۲۵ و اطلاعات ظرفیت و زمان سفر آزاد [۴۵] هر کمان مطابق با جدول ۲۶ می باشد.



شکل ۴-۱-نمایی از شبکه هرن

جدول ۲۵- عرضه و تقاضا در مبدأ-مقصد شبکه هرن

مبدأ/ مقصد	3	4
1	10	20
2	30	40

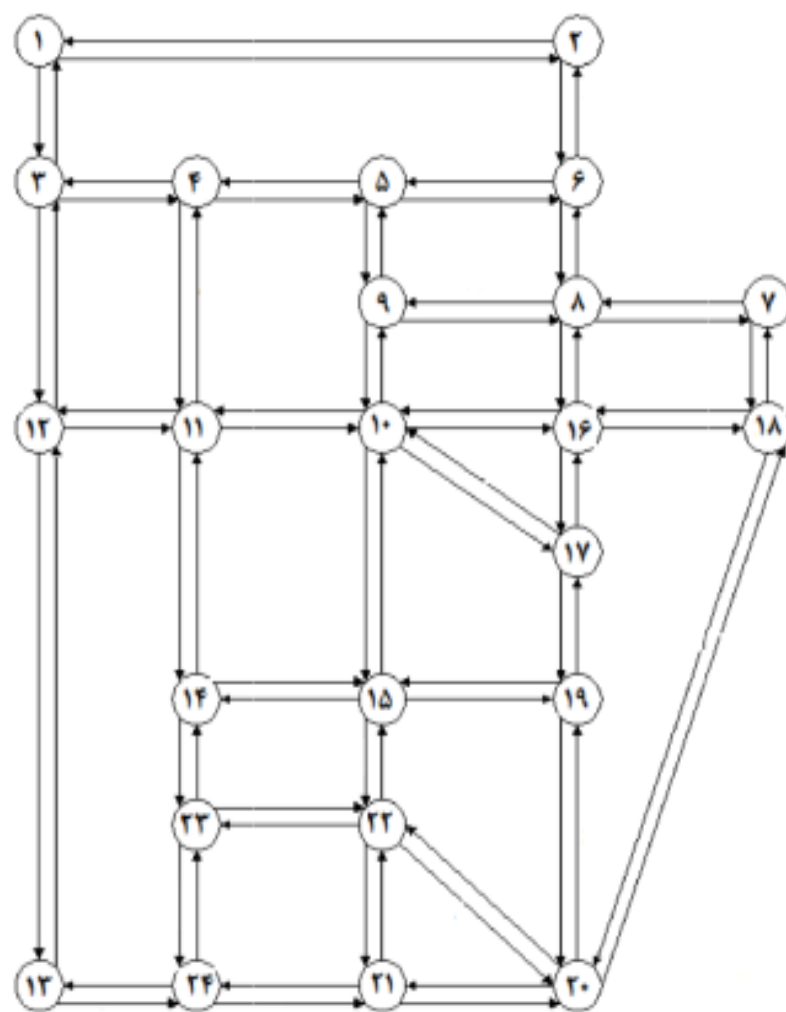
جدول ۲۶- ظرفیت بالا و پایین و زمان سفر آزاد کمان های شبکه هرن

کمان a	هزینه (C_a)	حد بالای ظرفیت (q_a)	حد پایین ظرفیت (l_a)
1	5	12.02	0
2	6	18.02	0
3	3	43.59	0
4	9	26.59	0
5	1	50.00	0

6	5	25.00	0
7	2	35.00	0
8	1	50.00	0
9	5	25.00	0
10	2	35.00	0
11	3	25.00	0
12	6	24.00	0
13	1	50.00	0
14	8	39.00	0
15	6	43.00	0
16	1	50.00	0
17	2	35.00	0
18	2	25.00	0

۴-۲-۲- شبکه سوفالز

شبکه سوفالز به عنوان یکی از شناخته شده ترین و محبوب ترین شبکه های حمل و نقلی، همواره مورد توجه پژوهشگران بوده است. این شبکه، همانطور که در شکل ۴-۲ نشان داده شده است، از ۲۴ گره، ۷۶ کمان و ۵۵۲ زوج مبدا-مقصد تشکیل شده است. تقاضا در این شبکه نیز مطابق جدول ۲۷ [۴۹] می باشد. اطلاعات زمان سفر آزاد و ظرفیت کمان ها نیز در جدول ۲۸ آورده شده است.



شکل ۴-۲- نمایشی از شبکه سופالز

جدول ۲۷ - عرضه و تقاضا در مبدا-مقصدهای شبکه سوفالز

مبدا/مقصد	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	1	1	5	2	3	5	8	5	13	5	2	5	3	5	5	4	1	3	3	1	4	3	1
2	1	0	1	2	1	4	2	4	2	6	2	1	3	1	1	4	2	0	1	1	0	1	0	0
3	1	1	0	2	1	3	1	2	1	3	3	2	1	1	1	2	1	0	0	0	0	1	1	0
4	5	2	2	0	5	4	4	7	7	12	14	6	6	5	5	8	5	1	2	3	2	4	5	2
5	2	1	1	5	0	2	2	5	8	10	5	2	2	1	2	5	2	0	1	1	1	2	1	0
6	3	4	3	4	2	0	4	8	4	8	4	2	2	1	2	9	5	1	2	3	1	2	1	1
7	5	2	1	4	2	4	0	10	6	19	5	7	4	2	5	14	10	2	4	5	2	5	2	1
8	8	4	2	7	5	8	10	0	8	16	8	6	6	4	6	22	14	3	7	9	4	5	3	2
9	5	2	1	7	8	4	6	8	0	28	14	6	6	6	9	14	9	2	4	6	3	7	5	2
10	13	6	3	12	10	8	19	16	28	0	40	20	19	21	40	44	39	7	18	25	12	26	18	8
11	5	2	3	15	5	4	5	8	14	39	0	14	10	16	14	14	10	1	4	6	4	11	13	6
12	2	1	2	6	2	2	7	6	6	20	14	0	13	7	7	7	6	2	3	4	3	7	7	5
13	5	3	1	6	2	2	4	6	6	19	10	13	0	6	7	6	5	1	3	6	6	13	8	8
14	3	1	1	5	1	1	2	4	6	21	16	7	6	0	13	7	7	1	3	5	4	12	11	4
15	5	1	1	5	2	2	5	6	10	40	14	7	7	13	0	12	15	2	8	11	8	26	10	4
16	5	4	2	8	5	9	14	22	14	44	14	7	6	7	12	0	28	5	13	16	6	12	5	3
17	4	2	1	5	2	5	10	14	9	39	10	6	5	7	15	28	0	6	17	17	6	17	6	3
18	1	0	0	1	0	1	2	3	2	7	2	2	1	1	2	5	6	0	3	4	1	3	1	0
19	3	1	0	2	1	2	4	7	4	18	4	3	3	3	8	13	17	3	0	12	4	12	3	1
20	3	1	0	3	1	3	5	9	6	25	6	5	6	5	11	16	17	4	12	0	12	24	7	4
21	1	0	0	2	1	1	2	4	3	12	4	3	6	4	8	6	6	1	4	12	0	18	7	5
22	4	1	1	4	2	2	5	5	7	26	11	7	13	12	26	12	17	3	12	24	18	0	21	11
23	3	0	1	5	1	1	2	3	5	18	13	7	8	11	10	5	6	1	3	7	7	21	0	7
24	1	0	0	2	0	1	1	2	2	8	6	5	7	4	4	3	3	0	1	4	5	11	7	0

جدول ۲۸- ظرفیت و زمان سفر آزاد کمان های شبکه سופالز

ظرفیت	زمان سفر آزاد	کمان
25.90	0.06	1
23.40	0.04	2
25.90	0.06	3
4.96	0.05	4
23.40	0.04	5
17.11	0.04	6
23.40	0.04	7
17.11	0.04	8
17.78	0.02	9
4.91	0.06	10
17.78	0.02	11
4.95	0.04	12
10	0.05	13
4.96	0.05	14
4.95	0.04	15
4.96	0.02	16
7.84	0.03	17
23.40	0.02	18
4.90	0.02	19
7.84	0.03	20

21	0.1	5.05
22	0.05	5.04
23	0.05	10
24	0.1	5.05
25	0.03	13.91
26	0.03	13.91
27	0.05	10
28	0.06	13.51
29	0.04	4.85
30	0.08	4.99
31	0.06	4.91
32	0.05	10
33	0.06	4.91
34	0.04	4.88
35	0.04	23.40
36	0.06	4.91
37	0.03	25.90
38	0.03	25.90
39	0.04	5.09
40	0.04	4.88
41	0.05	5.12
42	0.04	4.92

43	0.06	13.51
44	0.05	5.12
45	0.03	14.56
46	0.03	9.59
47	0.05	5.04
48	0.04	4.85
49	0.02	5.23
50	0.03	19.68
51	0.08	4.99
52	0.02	5.23
53	0.02	4.82
54	0.02	23.40
55	0.03	19.68
56	0.04	23.40
57	0.03	14.56
58	0.02	4.82
59	0.04	5.00
60	0.04	23.40
61	0.04	5.00
62	0.06	5.06
63	0.05	5.07
64	0.06	5.05

65	0.02	5.22
66	0.03	4.89
67	0.03	9.59
68	0.05	5.07
69	0.02	5.22
70	0.04	5
71	0.04	4.92
72	0.04	5
73	0.02	5.07
74	0.04	5.09
75	0.03	4.88
76	0.02	5.08

۳-۲-۴- شبکه آناهایم

این شبکه از ۴۱۶ گره، ۹۱۵ کمان و ۱۴۰۶ مبدا-مقصد تشکیل شده است. این شبکه به دلیل بزرگ-مقیاس بودن و دشواری های تحلیل آن برای مسئله تخصیص ترافیک با محدودیت ظرفیت ، بسیار کم مورد استفاده قرار گرفته است.

۳-۴- مقایسه تحلیلی نتایج مسئله تخصیص ترافیک با محدودیت

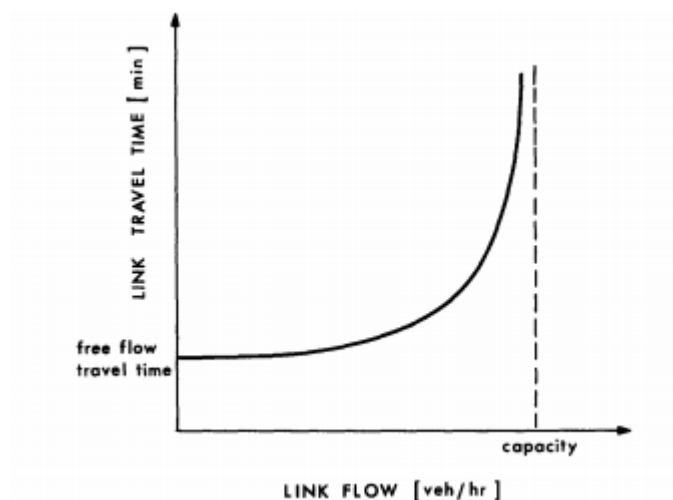
ظرفیت و بدون محدودیت ظرفیت

هدف این بخش بررسی اثرات در نظر گرفتن محدودیت ظرفیت برای کمان های یک شبکه بر جریان در آن شبکه می باشد. برای این منظور ابتدا به تعریف ظرفیت و عوامل تاثیر گذار بر آن و سپس به نحوه

تعریف متغیرهای الگوریتم Out of Kilter و ساختار داده ای آن می پردازیم. در این پژوهش، از روش برچسب گذاری سه گانه^۱ برای پیمایش درخت، جداکردن یک شاخه از درخت و متصل کردن یک شاخه به درخت استفاده شده است.

۴-۳-۱- ظرفیت و عوامل تاثیر گذار بر آن

ظرفیت هر کمان به مقدار ماکزیمم جریانی است که آن کمان می تواند از خود عبور دهد. واضح است که هر مسیر ظرفیت محدودی دارد که به ویژگی هایی مانند قدرت مانور وسایل نقلیه، تعداد تقاطعات مسیر، عرض مسیر و مشخصات سطح راه بستگی دارد. یک کمان نمی تواند ظرفیت بی نهایت داشته باشد و به در نظر گرفتن این ظرفیت بی نهایت برای کمان ها غیر منطقی باشد. هرچقدر جریانی که از یک کمان عبور می کند به ظرفیت آن کمان نزدیک تر باشد، زمان سفر در آن کمان افزایش می یابد. همانطور که در شکل ۴-۳ نشان داده شده است اگر جریان برابر با ظرفیت کمان شود، زمان سفر آن کمان به بی نهایت میل می کند



شکل ۴-۳- نمودار رابطه بین جریان و زمان سفر

^۱ Triple-label Method

۴-۳-۲-روش برچسب گذاری سه گانه

در روش برچسب گذاری سه گانه به هر گره سه برچسب تعلق می گیرد. یک جد^۱، یک فرزند^۲ و یک برادر^۳ برای هر گره تعریف می شود. در هر درخت، دقیقاً یک گره ریشه^۴ نامیده می شود که برای آن پدر تعریف نمی شود. از آنجا که ممکن است به غیر از گره ریشه، برای گره های دیگری نیز در درخت پدر تعریف نشده باشد، ما این گره ها را ریشه های یتیم^۵ می نامیم. بنابراین هر درخت از یک ریشه و تعدادی ریشه یتیم تشکیل شده است. در هنگام اعمال الگوریتم Out of Kilter، این درخت را با اضافه کردن درخت های یتیم به درخت موجود گسترش داده می شود.

۴-۳-۳- ساختار شبکه

در این بخش ابتدا به تعریف دو مفهوم نام ظرفیت بهینه^۶ و کمان تقویتی می پردازیم. ظرفیت بهینه هر کمان برابر با ماکزیمم میزان جریانی است که می توان به یک کمان اضافه کرد و یا از آن کم کرد بدون آن که آن کمان وارد حالت جدیدی از Out of Kilter شود. کمان تقویتی نیز به کمانی گفته می شود که در آن بتوان جریان را کم یا زیاد کرد.

در ساختار شبکه این الگوریتم، ما به طور مداوم اطلاعات هر کمان را با توجه به نقطه ابتدا و انتهای گره را بروز رسانی می کنیم. چون جستجوی یک کمان یا گروهی از کمان ها در تمام کمان های یک شبکه زمان بر است، ما به ساختاری که این عملیات را تسریع کند نیاز داریم. برای این منظور از دو ساختار برای تعریف گره ها و کمان ها استفاده می کنیم. در ساختار اول، برای هر گره دو آرایه تعریف می کنیم. آرایه اول به شماره اولین کمانی که از گره خارج می شود و آرایه دوم به شماره اولین کمانی که به گره

^۱ Predecessor

^۲ Successor

^۳ Brother

^۴ Root

^۵ Orphan roots

^۶ Effective Capacity

وارد می شود اشاره می کند. در ساختار دوم، برای هر کمان دو آرایه تعریف می کنیم. اولین آرایه به کمان بعدی در لیست که گره ابتدایی یکسانی با این کمان دارد و دومین آرایه به کمان بعدی در لیست که گره انتهایی یکسانی با این کمان دارد اشاره می کند. این دو ساختار تمام کمان ها با گره ابتدایی مشترک و همچنین تمام کمان ها با گره انتهایی مشترک را به هم متصل میکنند.

۴-۳-۴- استفاده از اطلاعات پیشین

در ضمن پیدا کردن یک مسیر تغییر جریان از گره i مانند l به گره k ، ممکن است تعداد بسیار زیادی مسیر تغییر جریان یافت شود که به گره k ختم نمی شوند. علاوه بر این، بعد از هر تغییر جریان، تنها عدد $kilter$ تعداد کمی از کمان های مسیر تغییر جریان از گره l به k کاهش می یابد. به عبارت دیگر، یک مسیر تغییر جریان از تعدادی مسیر تغییر جریان کوچکتر تشکیل شده است. نگهداری اطلاعات این مسیرها و استفاده از آن برای پیدا کردن یک مسیر تقویتی جدید، الگوریتم را بهینه می کند. به همین دلیل، استفاده از جنگل برای استفاده از این اطلاعات مناسب به نظر می رسد.

یک جنگل از یک درخت و تعدادی درخت یتیم تشکیل شده است. ریشه درخت زمانی که در پی یافتن مسیر تغییر جریان برای کمان $k-l$ از l به k هستیم برابر با l می باشد. با پیمایش در طول درخت، سعی بر اضافه کردن کمان های بیشتری به درخت برای یافتن گره k می باشیم. بنابراین، برای بهبود عملکرد، زمانی که در یک درخت یتیم یک کمان تقویتی از گره l به گره j یافت می شود، تمام فرزندان گره j را نیز می توان به درخت اضافه کرد. اما برای اطمینان از اینکه هر گره در جنگل تنها یک جد دارد (هیچ حلقه ای در جنگل وجود ندارد) گره j ابتدا باید از جد خود جدا شود تا تبدیل به یک گره یتیم شود. زمانی که یک مسیر تغییر جریان از گره l به k یافت شد، برای تغییر جریان، جریان را در همه کمان های تقویتی مستقیم مسیر افزایش و در کمان های تقویتی معکوس کاهش می دهیم (میران این تغییر برابر با ماکزیمم ظرفیت بهینه یک چرخه است). بعد از تغییرات جریان، گره فرزند هر کمان موجود در مسیر که به ظرفیت بهینه خود می رسد باید از جد خود جدا شود و تبدیل به یک ریشه یتیم جدید شود.

۴-۳-۵- روند پیمایش و برچسب گذاری

زمان حل الگوریتم Out of Kilter به چگونگی و نحوه پیمایش درخت و برچسب گذاری گره ها بسیار حساس است. دو روند کلی برای پیمایش و برچسب گذاری در این پژوهش استفاده شده است که در ادامه توضیح داده می شود.

یک گره زمانی برچسب گذاری می شود که برای اولین بار در پیمایش درخت به آن رسیده باشیم. یک گره برچسب گذاری شده زمانی اسکن می شود که هیچ کمانی از آن به یک گره برچسب گذاری نشده را بتوان به درخت اضافه کرد. در روند اول، ما با توجه به سطح پیمایش می کنیم و به این روند "برچسب گذاری شده در اول، اسکن شده در اول"^۱ می گوییم. در روند دوم، ما به سمت راست پیمایش می کنیم و به این روند "برچسب گذاری شده در آخر، اسکن شده در اول"^۲ می گوییم. نتایج نشان داده است که هرچند طول مسیر تغییر جریان برای روند اول نسبت به روند دوم کوتاه تر است ولی این روند بسیار کندتر از روند دوم است.

در ادامه گام های روند دوم، که به آن پیمایش به راست نیز گفته می شود، توضیح داده می شود. منظور از گره فعلی گره است که در حال اسکن آن برای اضافه کردن کمان های ممکن هستیم.

گام (۱)- (برچسب گذاری به راست)

تا حد ممکن برچسب گذاری را به سمت پایین و راست با حرکت از یک گره به تمام فرزندان انجام دهید. این کار را تا زمانی که گره z با هیچ فرزند برچسب گذاری نشده پیدا شود ادامه دهید.

گام (۲)- (جستجو کمان تقویتی)

اگر کمان تقویتی جدیدی در شبکه از گره فعلی به یک گره برچسب گذاری نشده وجود دارد، به جستجو برای پیدا کردن کمان تقویتی جدید ادامه دهید. در غیر این صورت به گام ۳ بروید.

گام (۳)- (اسکن کردن گره فعلی)

^۱ First labeled-First Scanned

^۲ Last labeled- First Scanned

گره فعلی هیچ فرزند برجسب گذاری نشده ای ندارد و در نتیجه هیچ مسیر تقویتی جدید ایجاد نمی کند پس اسکن شده است. اگر گره فعلی برادر داشته باشد، از گام یک برای برادر این گره عملیات را انجام دهید. در غیر این صورت به گام ۴ بروید.

گام (۴) - (کنترل)

اگر گره فعلی جد دارد از گام یک برای جد این گره عملیات را انجام دهید. در غیر این صورت هیچ مسیرتغییر جریان جدیدی وجود ندارد و u کمان ها باید تغییر کند.

چندین گزینه برای بهبود روند جستجوی مسیر تغییر جریان وجود دارد. اول آنکه زمانی که یک کمان تقویتی مانند $j-m$ به درخت اضافه می شود، برای تشخیص آنکه گره موردنظر (k) در زیردرخت j وجود دارد لازم است که از گره k به سمت جدش پیمایش کنیم. یک راه برای خودداری از این عمل استفاده از آرایه منطقی طول گره^۱ برای پیگیری گره هایی در مسیر بین گره k و ریشه یتیم درخت است. دوم آنکه این روش می تواند همزمان روی چندین کمان Out of Kilter به طور همزمان اعمال شود. ما می توانیم به دنبال تمام مسیرهای تغییر جریان که از گره l شروع می شوند و به گره j در کمان Out of Kilter مربوطه $j-l$ ختم می شوند باشیم. این روش نیاز به کار بیشتری برای هر تکرار دارد ولی مسیرهای تغییر جریان سریع تر یافت می شوند.

۴-۳-۶- اعمال تغییرات جریان

بعد از پیدا کردن مسیر تغییر جریان از گره l به گره k ، یک روند دو مرحله ای برای تغییر جریان مورد نیاز است. در مرحله اول ماکزیمم تغییرات مسیر (θ) محاسبه می شود. در مرحله دوم تغییرات جریان اعمال می شود. کمانی که در مسیر به تغییرات ماکزیمم خود می رسد از درخت تغییر جریان جدا می شود و فرزند آن یک ریشه یتیم می شود.

^۱ Node-length logical array

۴-۳-۷- اعمال تغییرات هزینه تقلیل یافته

مانند تغییرات جریان، تغییرات هزینه می تواند با یک روند دو مرحله ای اعمال شود. در مرحله اول ماکزیمم تغییرات هزینه مجاز (δ) محاسبه می شود و در مرحله دوم تغییرات هزینه اعمال می شود. در هر مرحله همان روند پیمایش که برای پیدا کردن مسیر تغییر جریان استفاده شد را می توان برای دست یافتن به تمام کمان های میانی بین گره های برچسب گذاری شده و برچسب گذاری نشده به کار برد.

۴-۳-۸- شرط چرخشی^۱ بودن شبکه

این کد فرض می کند که شبکه در فرم چرخشی است. چرخشی بودن شبکه به این معنا است که شبکه متصل است و حداقل یک کمان به هر گره وارد می شود و یک کمان از آن خارج می شود. به علاوه، جریان خروجی از هر گره با جریان ورودی به آن برابر است.

اگر یک شبکه متصل و غیر چرخشی با تعدادی گره منبع^۲ و چاه^۳ باشد، می توان آن را با اضافه کردن دو گره مصنوعی به عنوان منبع و چاه مجازی و تعدادی کمان مصنوعی آن را به یک شبکه چرخشی تبدیل کرد. هر گره منبع به گره منبع مجازی با یک کمان مصنوعی متصل می شود. مشخصات این کمان برابر است با:

۱. هزینه دلخواه کم (معمولاً یک عدد منفی بزرگ)

۲. حد بالای ظرفیت معادل با مقدار عرضه گره منبع

۳. جریان اولیه و حد پایین ظرفیت صفر

به طور مشابه کمان مصنوعی از گره چاه به گره چاه مجازی با پارامترهای بالا ایجاد می شود. تنها تفاوت آن این است که حد بالای ظرفیت معادل با مقدار تقاضا در گره چاه است. در نهایت، یک کمان مصنوعی

^۱ Circulation

^۲ Source

^۳ Sink

گره چاه مجازی و منبع مجازی را با ظرفیت بالای معادل با کل عرضه و سایر پارامترهای مشابه بالا متصل می کند.

۴-۳-۹- نحوه انتخاب کمان های Out of Kilter

چندین روش برای چگونگی و ترتیب انتخاب الگوریتم های Out of Kilter، به خصوص زمانی که کمان های مصنوعی برای تبدیل مسئله به فرم گردشی اضافه می شوند، وجود دارد. ما می توانیم هزینه و ظرفیت کمان های مصنوعی را طوری اضافه کنیم که

۱. تنها یک کمان Out of Kilter بین گره چاه مجازی و گره منبع مجازی وجود داشته باشد.

۲. تنها کمان های Out of Kilter بین گره چاه مجازی و گره منبه مجازی وجود داشته باشد

۳. تنها کمان های مصنوعی بین گره های منبع و چاه وجود داشته باشد.

نتایج نشان داده است که گزینه دوم بهینه تر است. دلیل آن شاید این است که درخت بین گره های بیشتری گسترش می یابد و سریع تر عمل می کند.

۴-۳-۱۰- تحلیل و مقایسه نتایج

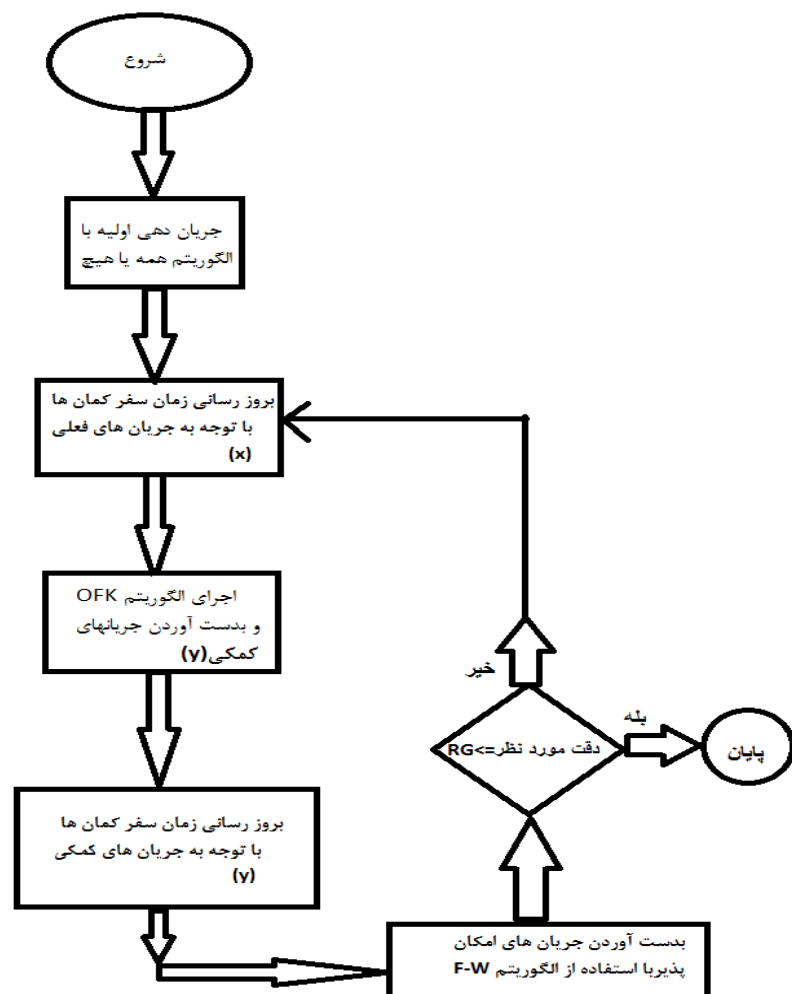
همانطور که در فصل دوم توضیح داده شد، در الگوریتم فرانک-ولف ابتدا با الگوریتم A/ϕ به کوتاه ترین مسیر بین زوج های مبدا و مقصد تقاضای اولیه را اختصاص داده، و سپس بر اساس اندازه گام و جهت حرکت که برای مینیمم کردن تابع هدف به دست می آیند، جریان بعدی برای هر کمان را به دست می آوریم. اما در الگوریتم Out of Kilter ابتدا به هر کمان یک جریان (x_a) و هزینه (C_a) اختصاص داده و سپس در هر تکرار جریان یا هزینه تقلیل یافته کمان ها طوری تغییر می کند که آنها را به کمان های In-Kilter تبدیل کند.

در فصل سوم نتایج حاصل از اجرای الگوریتم های OFK و فرانک-ولف را بر روی دو شبکه با زمان سفر ثابت نشان دادیم. اما در واقعیت زمان سفر در کمان ها وابسته به جریان می باشند و با افزایش جریان، افزایش می یابند. بنابراین در ادامه نتایج حاصل از اجرای این الگوریتم را بر روی شبکه هایی که در آن ها زمان سفر وابسته به جریان می باشند نشان می دهیم.

همانطور که در فصل ۳ اشاره شد معیار همگرایی در این الگوریتم برای کمان های با زمان سفر متغیر به صورت رابطه ۴-۱ بیان می شود

$$RG = \frac{\sum_a t_a x_a - \sum_a t_a y_a}{\sum_a t_a x_a} \quad (4-1)$$

در شکل ۴-۴ نمایشی کلی از گام های الگوریتم OFK برای شبکه های با زمان سفر متغیر آمده است.



شکل ۴-۴- مراحل الگوریتم OFK

همانطور که مشاهده می کنید جریان های اولیه الگوریتم با الگوریتم همه یا هیچ بدست می آید و الگوریتم OFK این قابلیت را دارد که جریان هایی را که بیشتر از ظرفیت یک کمان باشد را به حالت امکان پذیر تبدیل کند و سپس الگوریتم فرانک-ولف این جریان ها را به عنوان ورودی گرفته و جریان های امکان پذیر از لحاظ جهت حرکت را بدست می آورد.

$$t_a = t_0(1 + \alpha(\frac{f}{Q})^\beta) \quad (۱-۴)$$

در این رابطه t_0 برابر با زمان سفر آزاد، f جریان در کمان، Q ظرفیت کمان و α و β ضرایب ثابت با توجه به ویژگی های کمان میباشند. در این پژوهش $\alpha = 0.15$ و $\beta = 4$ می باشد.

شبکه اول که مورد بررسی قرار می گیرد شبکه هرن می باشد که مشخصات آن در جدول ۲۵ و ۲۶ و ساختار آن در شکل ۴-۱ آمده است.

جریان ها و هزینه سفرهای کمان های این شبکه بدست آمده برای الگوریتم های مختلف در جدول ۲۹ و ۳۰ ارائه شده است. دو الگوریتم FW و FW-OFK-INFINITECAP جریان ها و هزینه سفرهای نزدیک بهم برای کمان ها را بدست آورده اند زیرا که ظرفیت در هر دو بی نهایت در نظر گرفته شده است. همچنین مقایسه بین هزینه کل و هزینه مصرف کننده برای هفت الگوریتم در جدول ۳۱ آمده است. این الگوریتم ها برابر با الگوریتم های فرانک-ولف، فرانک-ولف با ظرفیت بی نهایت، تابع جریمه دینامیک برای ظرفیت بی نهایت (۵۰)، الگوریتم فرانک-ولف با در نظر گرفتن ظرفیت محدود برای هر کمان، تابع جریمه دینامیک با در نظر گرفتن ظرفیت محدود برای هر کمان (۵۰)، ALM (۳۹) و IPF (۳۹) می باشند.

جدول ۲۹-جریان های بدست آمده از سه الگوریتم FW-OFK،FW-OFK-INFINITECAP،FW

CAPACITATED برای شبکه هرن

FW		FW-OFK-INFINITECAP		FW-OFK-CAPACITATED	
جریان	ظرفیت	جریان	ظرفیت	جریان	ظرفیت
12.3350	400	12.3517	400	12.0200	12.02
17.6641	400	17.6482	400	17.9800	18.02
70.00000	400	70.0000	400	43.5900	43.59

0.00000	400	0.00000	400	26.4100	26.59
10.5914	400	10.6669	400	0.0000	50.00
23.57257	400	23.5546	400	20.6100	25.00
48.1710	400	48.1302	400	35.0000	35.00
0.00000	400	0.0000	400	0.0000	50.00
3.2899	400	3.2565	400	20.4055	25.00
24.9665	400	25.0586	400	23.9844	35.00
40.0000	400	40.000	400	25.0000	25.00
23.7115	400	23.6905	400	24.0000	24.00
0.00000	400	0.0000	400	5.5944	50.00
0.00000	400	0.0000	400	15.0000	39.00
36.28848	400	36.3094	400	36.0000	43.00
0.10783	400	0.0000	400	0.0000	50.00
40.03112	400	40.1359	400	33.9844	35.00
33.10643	400	33.0529	400	25.0000	25.00

جدول ۳۰- هزینه سفرهای بدست آمده از بدست آمده از سه الگوریتم FW-OFK-FW

FW-OFK-CAPACITATED.INFINITECAP برای شبکه هرن

FW	FW-OFK- INFINITECAP	FW-OFK- CAPACITATED
5.8318	5.8363	5.7500
6.8315	6.8280	6.89204
5.8812	5.9927	3.4500
9.00000	9.0000	10.313
1.0002	1.0003	1.0000

5.5928	5.5910	5.3438
3.0769	3.0728	2.2953
1.00000	1.0000	1.0000
5.0002	5.0002	5.4994
2.0777	2.0788	2.0466
5.9440	5.9491	3.4500
6.8566	6.8545	6.9000
1.00000	1.0000	1.000
8.00000	8.0000	8.0262
6.4509	6.4575	6.4421
1.0000	1.0000	1.0000
2.5135	2.5188	2.2055
2.9250	2.9166	2.3000

همان طور که در فصل سوم عنوان شد اضافه کردن محدودیت ظرفیت به کمان ها باعث افزایش هزینه کل و هزینه مصرف کننده می شود. این موضوع را می توان در جدول ۳۱ نیز مشاهده کرد. مقدار هزینه کل و هزینه مصرف کننده برای الگوریتم هایی که محدودیت ظرفیت کمان ها را در نظر گرفته اند بیشتر از الگوریتم هایی است که ظرفیت را بی نهایت در نظر گرفته اند. همچنین چون در الگوریتم FW-OFK با ظرفیت بی نهایت، ظرفیت مانند الگوریتم FW در نظر گرفته نمیشود، هزینه کل و هزینه مصرف کننده در این دو الگوریتم تقریباً مشابه یکدیگر می باشد. مقایسه هزینه کل، هزینه مصرف کننده، جریان ها و هزینه های سفر برای اطمینان از درستی الگوریتم پیشنهادی می باشد.

همانطور که در جدول ۳۱ مشاهده می کنید الگوریتم FW-OFK در تعداد تکرار بسیار کمتری نسبت به الگوریتم FW در حالت بدون ظرفیت به شکاف نسبی 10^{-4} میرسد. در حالت با محدودیت ظرفیت الگوریتم FW-OFK تابع هدف و هزینه کل کوچکتری نسبت به الگوریتم های ALM و IPF دارد. این مقادیر با مقادیر بدست آمده از الگوریتم DPF برابر است.

جدول ۳۱- نتایج بدست آمده برای شبکه هرن

	Algorithm	TC	UC	RG	Iteration
بدون ظرفیت	FW	1803.59	1396.83	10^{-4}	2424
	FW-OFK	1803.03	1396.71	10^{-4}	133
	DPF	NA	1396.71	10^{-6}	56
با ظرفیت	FW-OFK	1709.16	1572.27	10^{-4}	3
	DPF	NA	1572.27	10^{-6}	1682
	ALM	1716.68	1572.36	10^{-6}	65
	IPF	1709.53	1572.31	10^{-6}	19

در جدول ۳۲ مقادیر هزینه کل، هزینه مصرف کننده، شکاف نسبی، α برای الگوریتم فرانک-ولف و ماکزیمم جریان بر ظرفیت را برای چند تکرار الگوریتم های FW-OFK،FW-OFK-INFINITECAP،FW CAPACITATED برای شبکه هرن مشاهده می کنید. همانطور که مشاهده میکنید در دو الگوریتم اول

مقادیر جریان بر ظرفیت بیشتر از یک می باشد ولی در الگوریتم سوم حداکثر این مقدار برابر یک می باشد که این نشان از منطقی بودن جواب های این الگوریتم می باشد. همچنین نرخ همگرایی و کاهش شکاف نسبی در الگوریتم اول بسیار کمتر از الگوریتم دوم و سوم می باشد.

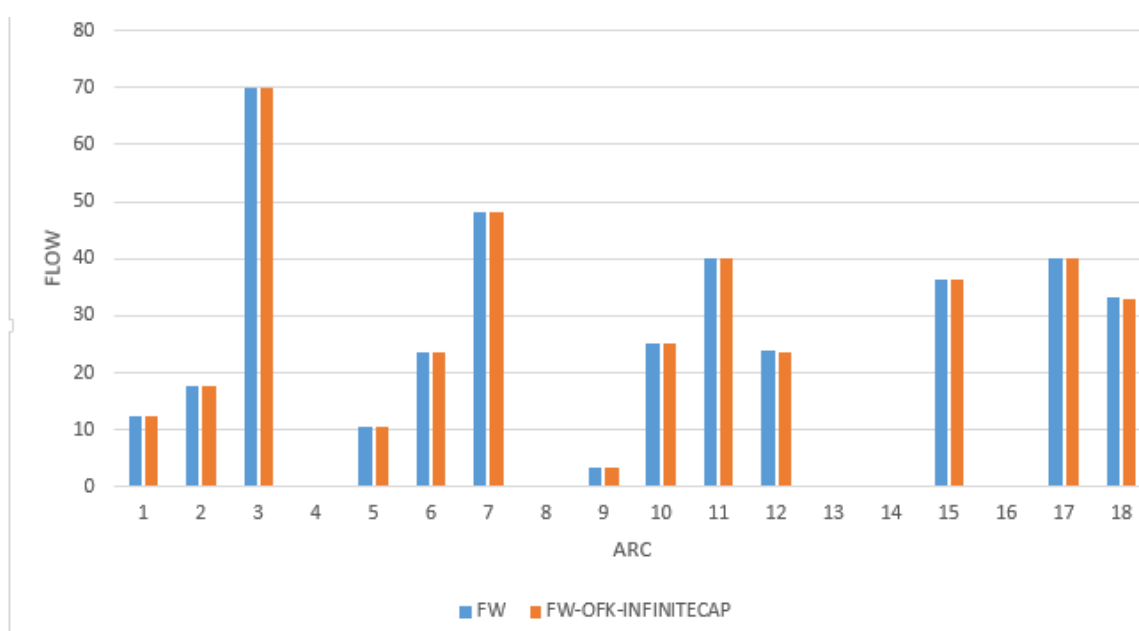
جدول ۳۲- مقایسه نتایج حاصل از سه الگوریتم FW-OFK-INFINITECAP, FW-OFK, FW

CAPACITATED در تکرار های مختلف برای شبکه هرن

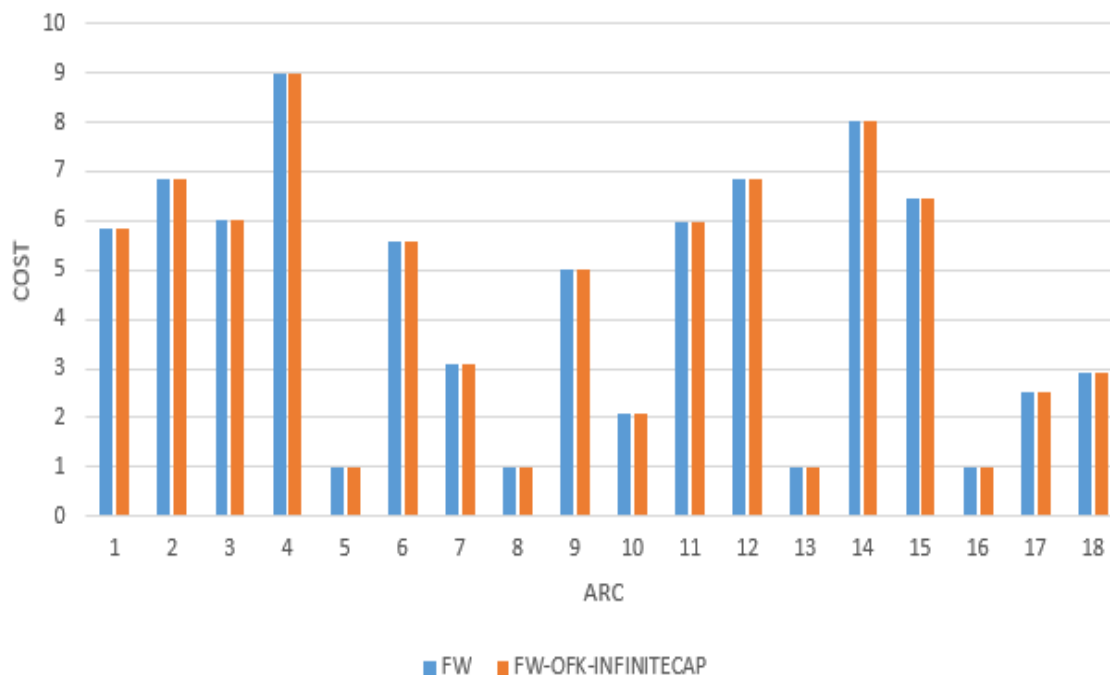
الگوریتم	شماره تکرار	هزینه کل	هزینه مصرف کننده	شکاف نسبی	α	Max(v/C)
FW	1	5375.85	2067.17	0.686	0.250	2.857
	2	2532.74	1520.55	0.308	0.375	2.142
	3	1936.61	1426.07	0.103	0.156	1.862
	4	1831.23	1410.19	0.051	0.078	1.606
	5	1833.30	1406.54	0.028	0.117	1.606
	2424	1803.59	1396.83	0.00008	0.003	1.606
FW-OFK-INFINITECAP	1	3029.41	1677.88	0.482	0.078	2.400
	2	1811.60	1407.32	0.060	0.055	1.606
	3	1807.02	1399.50	0.018	0.039	1.606
	4	1815.70	1398.71	0.016	0.058	1.606
	5	1801.08	1398.15	0.012	0.035	1.606
	133	1803.03	1396.71	0.00008	0.0003	1.606

FW-OK-CAPACITATED	1	1756.27	1603.30	0.0235	1.000	1.000
	2	1709.43	1572.33	0.0003	0.003	1.000
	3	1709.16	1572.27	0.00004	0.004	1.000

در شکل ۴-۵ مقایسه بین جریان های و شکل ۴-۶ مقایسه بین هزینه سفرهای بدست آمده از الگوریتم های مختلف را در حالت بدون ظرفیت داده شده است. همانطور که مشاهده می کنید جریان در این حالت برای دو الگوریتم بسیار نزدیک به هم و در چند مورد منطبق بر یکدیگر است. به علت اینکه جریان در کمان ها نزدیک به هم می باشد هزینه سفر کمان ها نیز که در شکل ۴-۶ آمده است نیز به هم نزدیک می باشند.

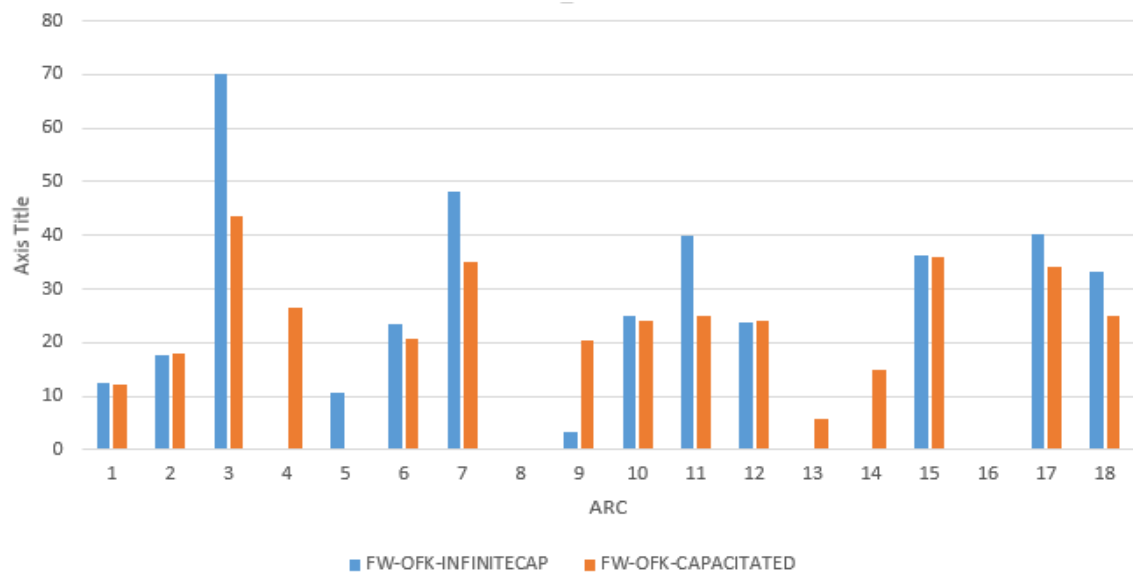


شکل ۴-۵-مقایسه جریان در کمان ها برای الگوریتم های FW و FW-OK-INFINTECAP



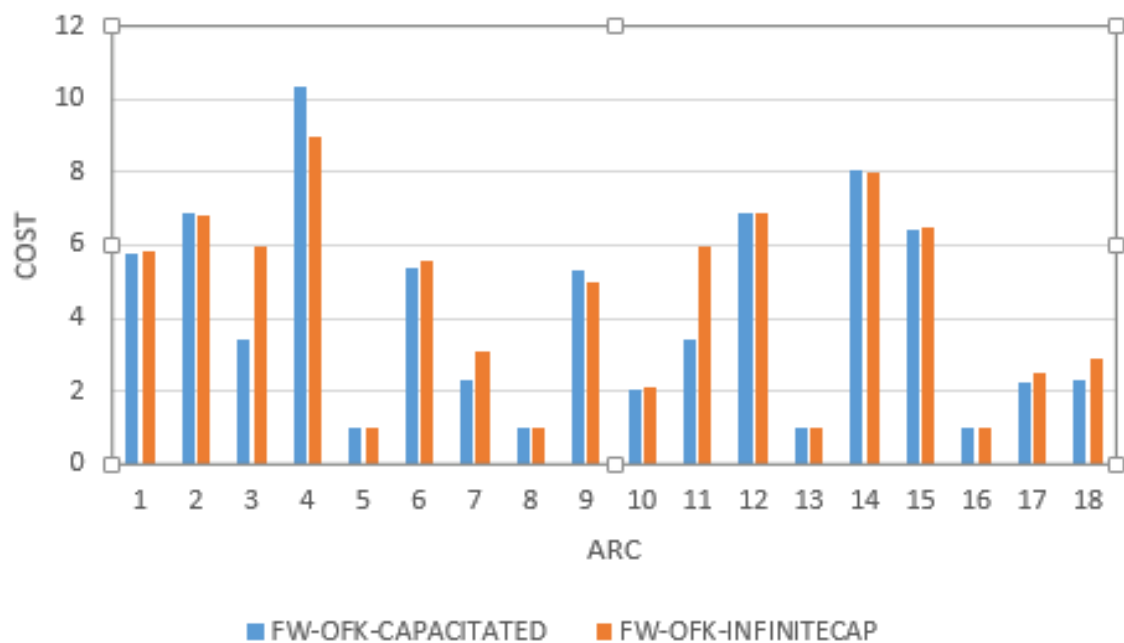
شکل ۴-۶- مقایسه هزینه کمان ها برای الگوریتم های FW و FW-OFK-INFINITECAP

شکل ۴-۷ و ۴-۸ نیز به مقایسه جریان و هزینه سفر کمان ها در حالت محدودیت ظرفیت و با محدودیت ظرفیت پرداخته است. در شکل ۴-۹ نسبت جریان به ظرفیت برای الگوریتم های فرانک-ولف، FW-OFK بدون ظرفیت و با ظرفیت نشان داده شده است. همانطور که مشاهده می کنید جریان های بدست آمده از الگوریتم فرانک-ولف در بعضی کمان ها از ظرفیت آن ها بیشتر است که این نشان از غیر دقیق بودن جواب های فرانک-ولف می باشد. نمی توان از یک کمان با ظرفیت محدود انتظار داشت که جریان بیشتری از ظرفیت خود عبور دهد بدون آنکه هزینه سفر آن به صورت تصاعدی بالا نرود. در شکل ۴-۸ نیز هزینه سفر برای دو حالت بدون ظرفیت و با ظرفیت آورده شده است. همانطور که در ابتدای این فصل گفته شد اگر جریان از ظرفیت بیشتر شود، هزینه سفر به سمت بی نهایت میل می کند. اما در الگوریتم FW با وجود اینکه جریان از ظرفیت بیشتر شده است، هزینه سفر مقدار محدودی است که این نیز دلیلی بر غیردقیق بودن نتایج حاصل از الگوریتم FW دارد.



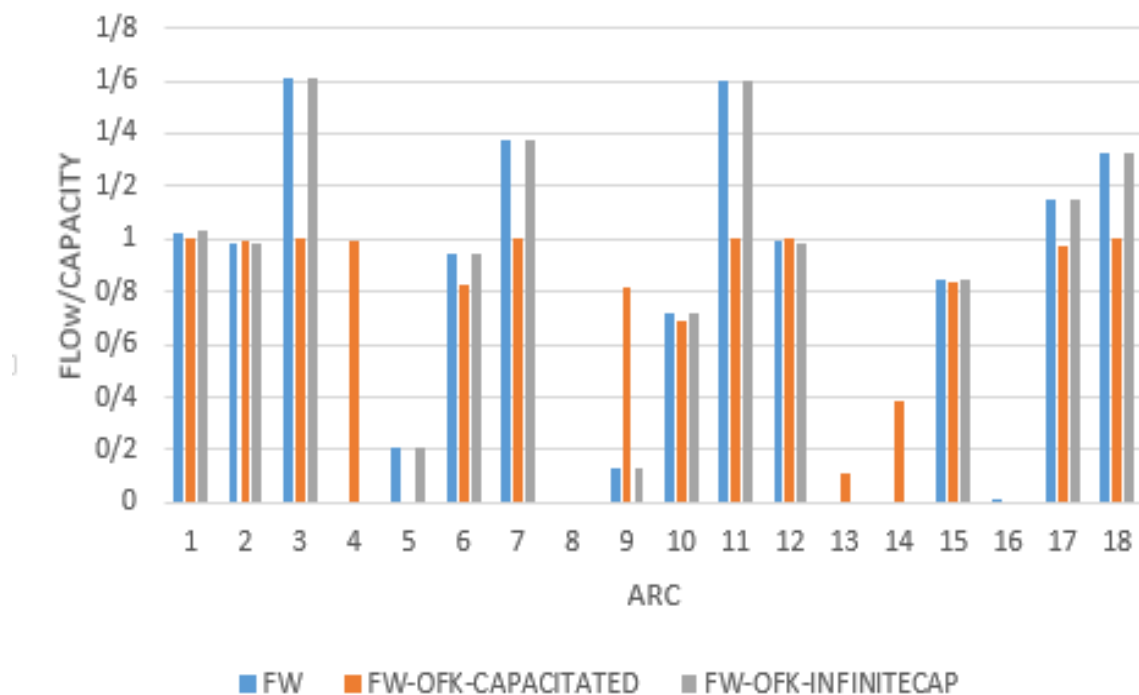
شکل ۴-۷- مقایسه جریان در کمان ها برای دو الگوریتم FW-OFK-CAPACITATED و

FWOFK-INFINITECAP



شکل ۴-۸- مقایسه هزینه سفر کمان ها برای دو الگوریتم FW-OFK-CAPACITATED و

FWOFK-INFINITECAP

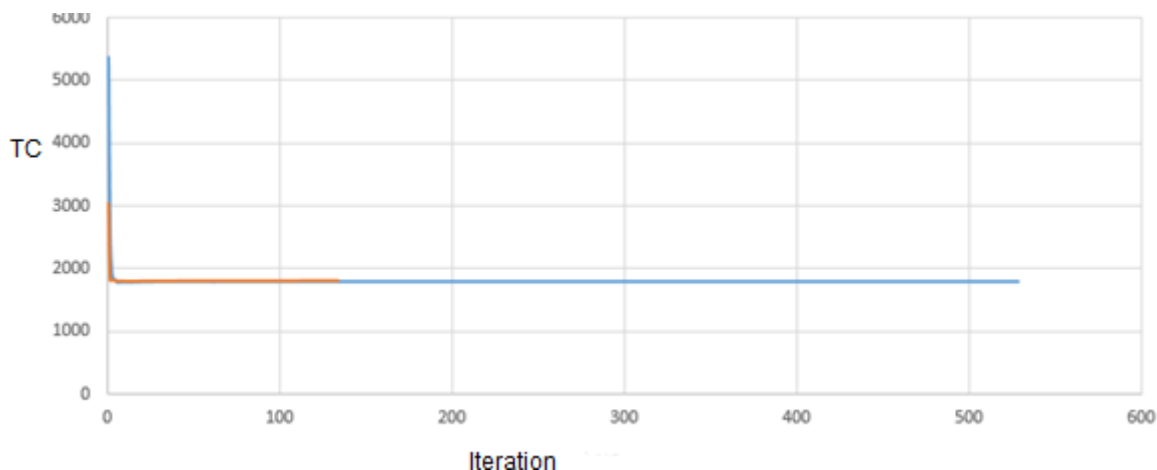


شکل ۴-۹- مقایسه نسبت جریان به ظرفیت برای الگوریتم های FW،FW-OFK-CAPACITATED و

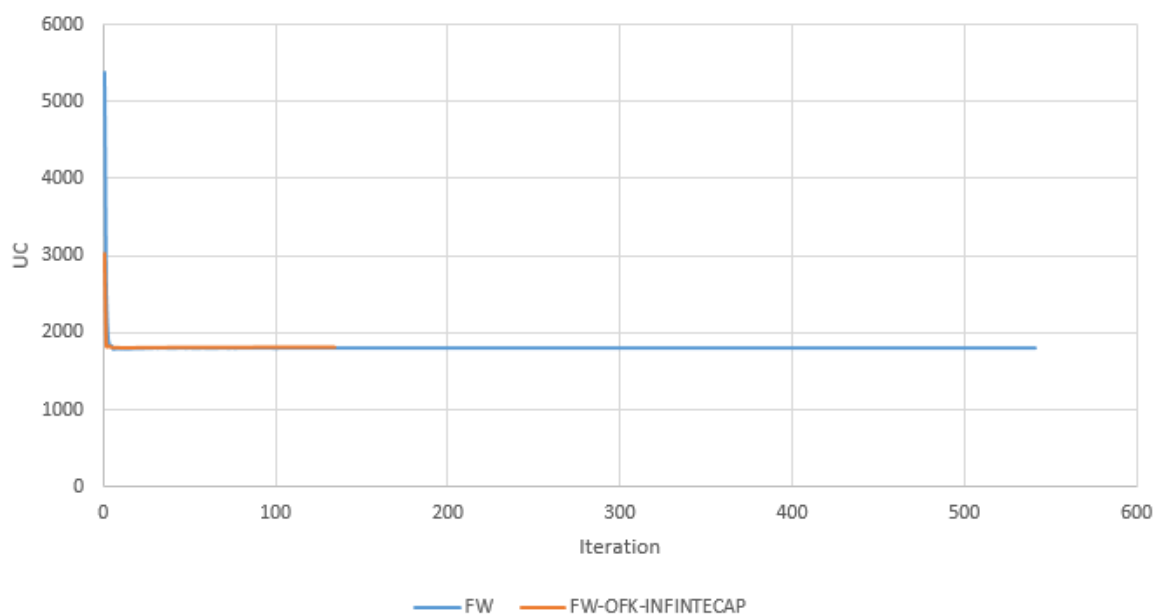
FW-OFK-INFINITECAP

در شکل ۴-۱۰ میزان تغییرات هزینه کل نسبت به تکرار الگوریتم در حالت بدون ظرفیت نشان داده شده است. الگوریتم FW با هزینه کل اولیه بزرگتری نسبت به الگوریتم FW-OFK شروع شده و همچنین در تعداد تکرارهای بیشتری نسبت به الگوریتم FW-OFK به جواب نهایی همگرا شده است که دلیلی بر کارایی کمتر این الگوریتم نسبت به الگوریتم FW-OFK دارد..

شکل ۴-۱۱ نیز تغییرات هزینه مصرف کننده را برای دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت نشان میدهد. مشاهده می شود که الگوریتم FW-OFK در تعداد تکرار کمتری و با هزینه مصرف کننده اولیه کمتری نسبت به الگوریتم FW به جواب نهایی همگرا می شود



شکل ۴-۱۰- مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت



شکل ۴-۱۱- مقایسه تغییرات هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت

در ادامه به بررسی نتایج الگوریتم OFK بر روی شبکه سوفالز می پردازیم. همانطور که در این فصل ذکر شد، شبکه مورد استفاده در الگوریتم OFK باید چرخشی باشد بدین معنا که به هر گره حداقل یک کمان وارد شود و حداقل یک کمان خارج شود. بنابراین برای شبکه سوفالز که یک سری از گره ها مبدا و یک سری از گره ها مقصد می باشند، باید کمان هایی مجازی اضافه کرد که به مشخصات کمان های اضافه شده به صورت زیر می باشد:

- برای کمان های مجازی ورودی به گره مبدا و خروجی از گره مبدا اصلی، حد بالا، حد پایین ظرفیت و جریان کمان برابر با عرضه گره مبدا و هزینه کمان صفر تعریف می شود.
- برای کمان های مجازی خروجی از گره مقصد و ورودی به گره مقصد اصلی، حد بالا، حد پایین ظرفیت و جریان کمان برابر با تقاضا گره مقصد و هزینه کمان صفر تعریف می شود.
- برای کمان های مجازی خروجی از گره مقصد اصلی و ورودی به گره مبدا اصلی، حد بالا، حد پایین ظرفیت و جریان کمان برابر با عرضه یا تقاضای کل شبکه و هزینه کمان صفر تعریف می شود.

یکی از مشکلاتی که در حل شبکه سوفالز با الگوریتم OFK وجود داشت را می توان عدم توانایی این الگوریتم در تشخیص تقاضا به تفکیک مبدا نام برد. جریان های اولیه کمان های این الگوریتم همان جریان های اولیه ای است که از الگوریتم تخصیص همه یا هیچ در الگوریتم فرانک-ولف بدست می آید، می باشد. بنابراین تقاضا در این الگوریتم به تفکیک مبدا-مقصد اعمال نمی شود و این عامل باعث می شود که نتایج نهایی صحیح نباشد یکی دیگر از مشکلات این الگوریتم آن است که در شبکه باید به گره های مبدا هیچ تقاضایی وجود نداشته باشد و همچنین گره های مقصد هیچ عرضه ای نداشته باشند. برای حل این مشکل شبکه سوفالز کوچک شده، ابتدا نتایج را برای یک مبدا و سپس برای بررسی بیشتر شبکه را برای دو مبدا که تقاضای آنها به طور جداگانه به شبکه اعمال می شود، بدست آورده شده است در این حالت جواب اولیه جریان کمان های اصلی همان جریان های بدست آمده از الگوریتم همه یا هیچ در فرانک-ولف می باشد که به صورت اطلاعات ورودی به الگوریتم OFK داده می شود. برای بدست آوردن ظرفیت کمان ها در شبکه با یک مبدا چون ظرفیت ها بسیار کوچک و غیر منطقی بدست می آمد، عرضه و تقاضا پنج برابر شده و نتایج را برای این تقاضا بدست آمده است. مشخصات این شبکه در جدول ۴۱ و ۴۲ پیوست آمده است. در این شبکه مبدا فقط گره ۱ و بقیه گره ها مقصد می باشند.

در جدول ۴۱ تنها ظرفیت کمان های ۳۹،۳۶،۳۴،۲۹،۱۶،۱۳،۱۰،۶،۴،۲ افزایش یافته است و ظرفیت باقی کمان ها همان ظرفیت قبلی است. در جدول ۳۳ به مقایسه نتایج بدست آمده از الگوریتم های FW و OFK-FW پرداخته شده است. مانند مثال شبکه هرن، در این حالت نیز هزینه کل و هزینه مصرف کننده برای حالتی که ظرفیت کمان ها محدود در نظر گرفته شده است بیشتر از حالتی است که ظرفیت بی نهایت است. همچنین در حالت بدون ظرفیت تعداد تکرارهای الگوریتم FW-OFK برای رسیدن به دقت مورد نظر کمتر از الگوریتم های مشابه می باشد که نشان از کارا بودن این الگوریتم است. در حالت

بدون ظرفیت تعداد تکرارهای لازم برای رسیدن به دقت موردنظر تقریباً یکی می باشد. در این جدول همان طور که قبلاً گفته شد مقدار هزینه کل و هزینه مصرف کننده برای الگوریتم های FW و FW-OFK بدون محدودیت ظرفیت باید یکسان باشند. در حالت با محدودیت ظرفیت نیز مقدار هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK برابر با هم می باشند. علت بیشتر بودن هزینه کل در الگوریتم DPF در نظر رفتن اثر عامل تاخیر در هزینه کل در این الگوریتم می باشد.

جدول ۳۳- نتایج بدست آمده برای شبکه سوفالز تک مبدایی

	Algorithm	TC	UC	RG	Iteration
بدون ظرفیت	FW	9.605	7.695	10^{-4}	557
	FW-OFK	9.605	7.695	10^{-4}	571
با ظرفیت	FW-OFK	9.856	7.931	10^{-4}	2
	DPF	11.071	7.931	10^{-6}	100

در جدول ۳۴ مقادیر هزینه کل، هزینه مصرف کننده، شکاف نسبی، α برای الگوریتم فرانک-ولف و ماکزیمم جریان بر ظرفیت را برای چند تکرار الگوریتم های FW، FW-OFK-INFINITECAP، FW-OFK-CAPACITATED برای شبکه سوفالز مشاهده می کنید. همانطور که مشاهده میکنید در دو الگوریتم اول

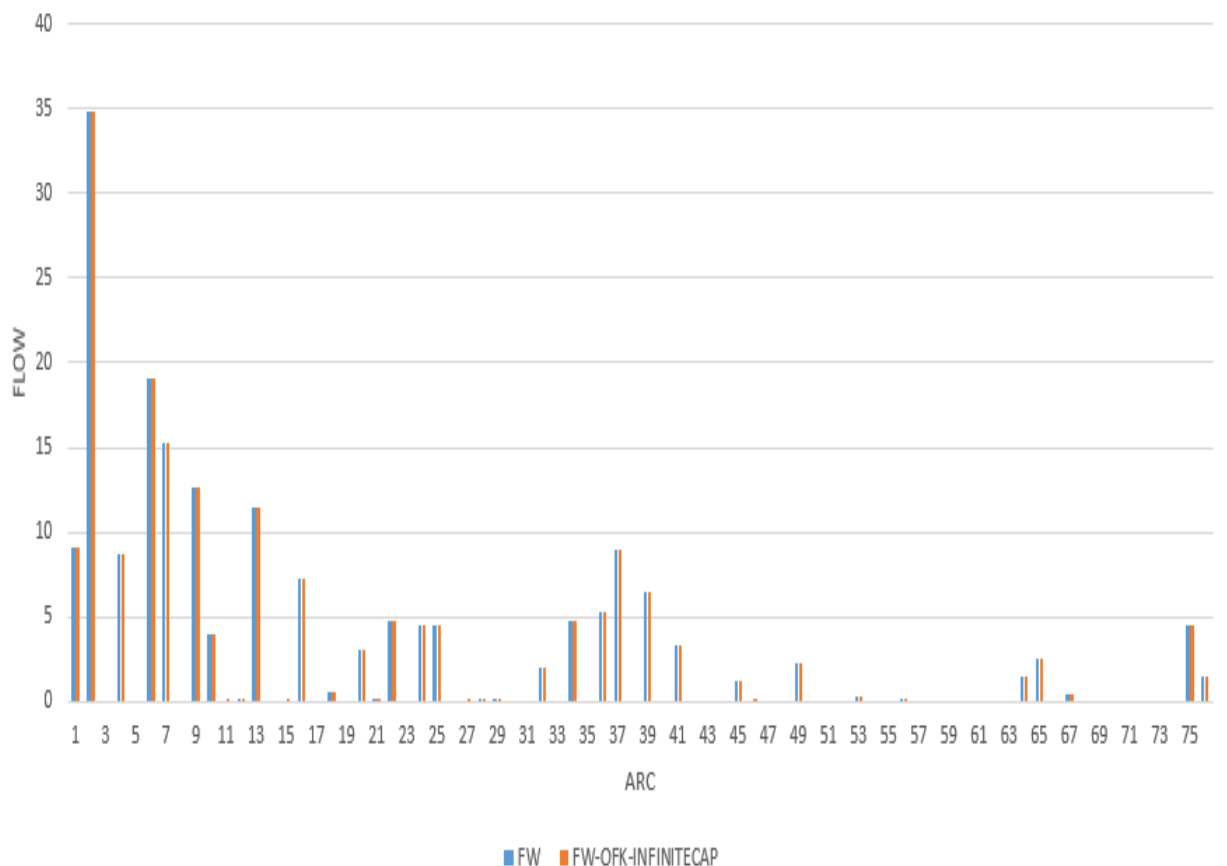
مقادیر جریان برظرفیت بیشتر از یک می باشد ولی در الگوریتم سوم حداکثر این مقدار برابر یک می باشد که این نشان از منطق بودن جواب های این الگوریتم می باشد.

جدول ۳۴- مقایسه نتایج حاصل از سه الگوریتم FW-OFK-INFINITECAP، FW-OFK-INFINITECAP، FW-OFK-INFINITECAP

CAPACITATED در تکرار های مختلف برای شبکه سوفالز تک مبدایی

الگوریتم	شماره تکرار	هزینه کل	هزینه مصرف کننده	شکاف نسبی	α	Max(v/C)
FW	1	24.07	10.37	0.628	0.250	3.227
	2	12.13	8.23	0.244	0.312	2.420
	3	9.87	7.84	0.037	0.141	1.940
	4	9.74	7.81	0.022	0.351	1.869
	5	9.71	7.78	0.030	0.117	1.714
	557	9.61	7.69	0.00009	0.0005	1.746
FW-OFK-INFINITECAP	1	92.66	25.09	0.906	0.25	4.481
	2	29.02	12.24	0.695	0.25	3.361
	3	14.71	9.16	0.390	0.375	2.521
	4	10.27	8.17	0.128	0.312	2.039
	5	10.15	7.98	0.059	0.0005	1.781
	571	9.61	7.69	0.00009	0.0005	1.746
FW-OK-CAPACITATED	1	10.1	8.11	0.018	1.000	1.000
	2	9.86	7.93	0.000	1.000	1.000

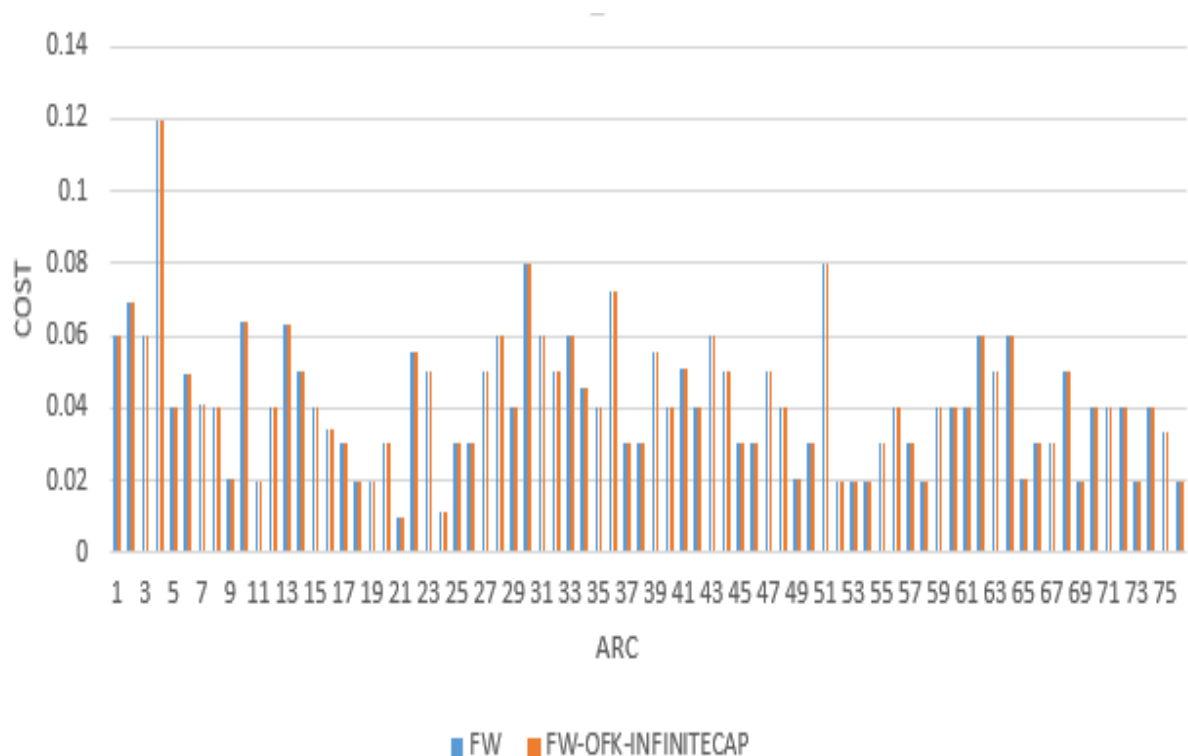
شکل ۴-۱۲ به مقایسه جریان های بدست آمده از دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت می پردازد. همانطور که در شکل مشخص است، چون ظرفیت بی نهایت در نظر گرفته شده است، جریان کمان ها در دو الگوریتم بسیار نزدیک به هم و در بسیاری از کمان ها برابر هم می باشد. نتایج عددی جریان ها در جدول ۳۷ پیوست آمده است.



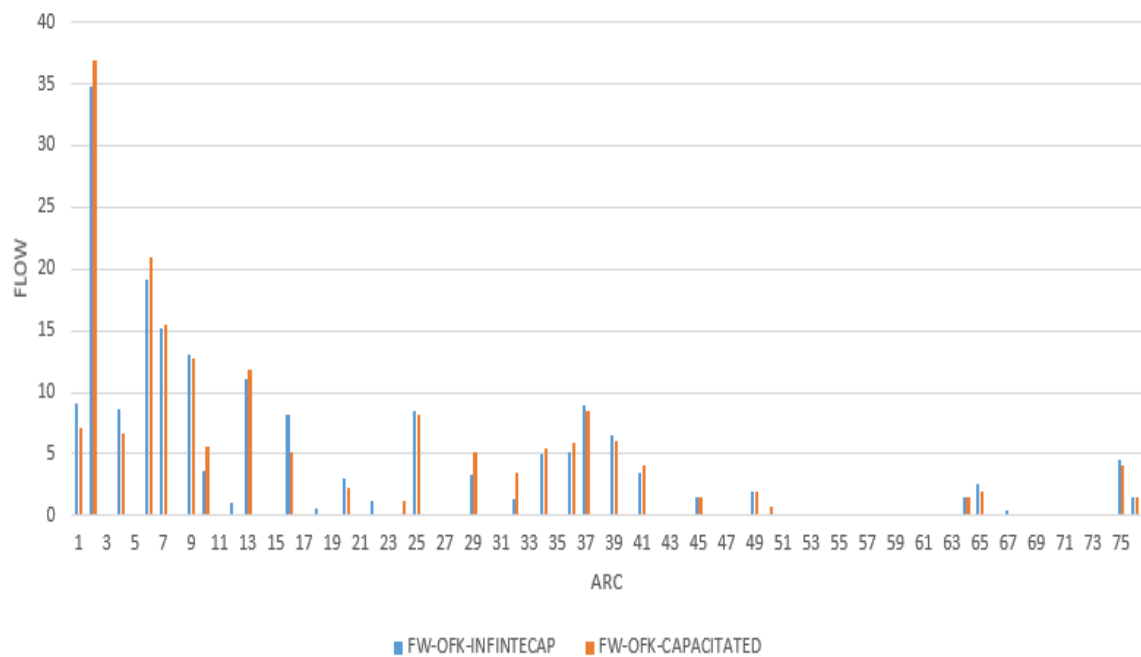
شکل ۴-۱۲-مقایسه جریان بدست آمده از دو الگوریتم FW و FW-OFK-INFINITECAP

شکل ۴-۱۳ نیز به مقایسه هزینه های بدست آمده از دو الگوریتم FW و FW-OFK در حالت بدون ظرفیت می پردازد. به علت نزدیک بودن جریان کمان ها، هزینه های بدست آمده نیز بهم نزدیک هستند. نزدیک شدن نتایج عددی جریان و هزینه کمان ها در حالت ظرفیت بی نهایت نشان می دهد که الگوریتم FW-OFK به درستی عمل می کند زیرا که در این حالت در الگوریتم FW-OFK ظرفیت را مانند الگوریتم FW معمولی بی نهایت در نظر می گیریم و بنابراین جواب ها باید یکسان باشد. برای مقایسه دقیق تر، نتایج عددی هزینه کمان ها در جدول ۳۸ پیوست آمده است.

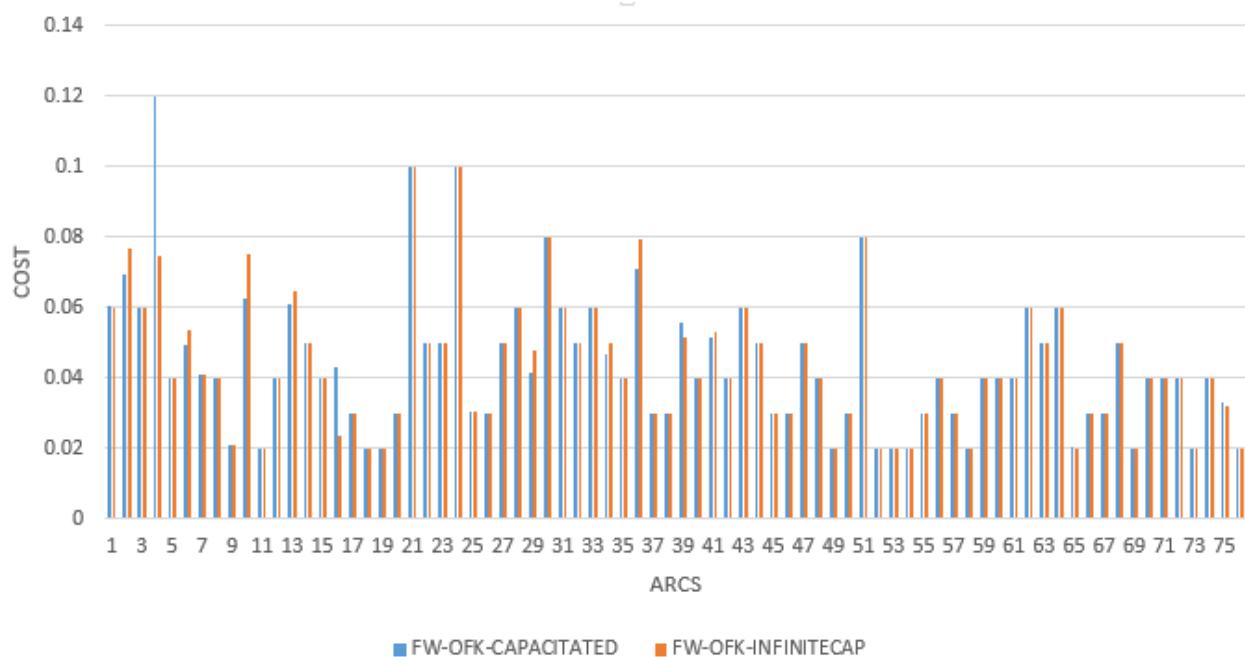
شکل ۴-۱۳- مقایسه هزینه سفر بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP در



شکل ۴-۱۴ مقایسه بین جریان کمان های بدست آمده از الگوریتم های FW-OFK- و CAPACITATED و FW-OFK-INFINITECAP آمده است. در شکل ۴-۱۵ نیز مقایسه بین هزینه های بدست آمده از این دو الگوریتم آمده است. مانند مثال های قبل در الگوریتم دوم در کمان هایی که جریان از ظرفیت بیشتر می باشد، هزینه سفر کمان ها باید به بی نهایت میل کند ولی از آنجایی که ظرفیت بی نهایت در نظر گرفته شده است هزینه سفر این کمان ها مقدار محدودی می باشد. این موضوع اهمیت در نظر گرفتن ظرفیت را به خوبی نشان می دهد چرا که در صورت در نظر نگرفتن ظرفیت تخمین شرایط برای طراحی شبکه، بهبود معابر و سایر موارد دقیق نمی باشد. نتایج عددی جریان کمان های بدست آمده از الگوریتم FW-OFK-CAPACITATED در جدول ۳۷ و هزینه های سفر در جدول ۳۸ پیوست آمده است.

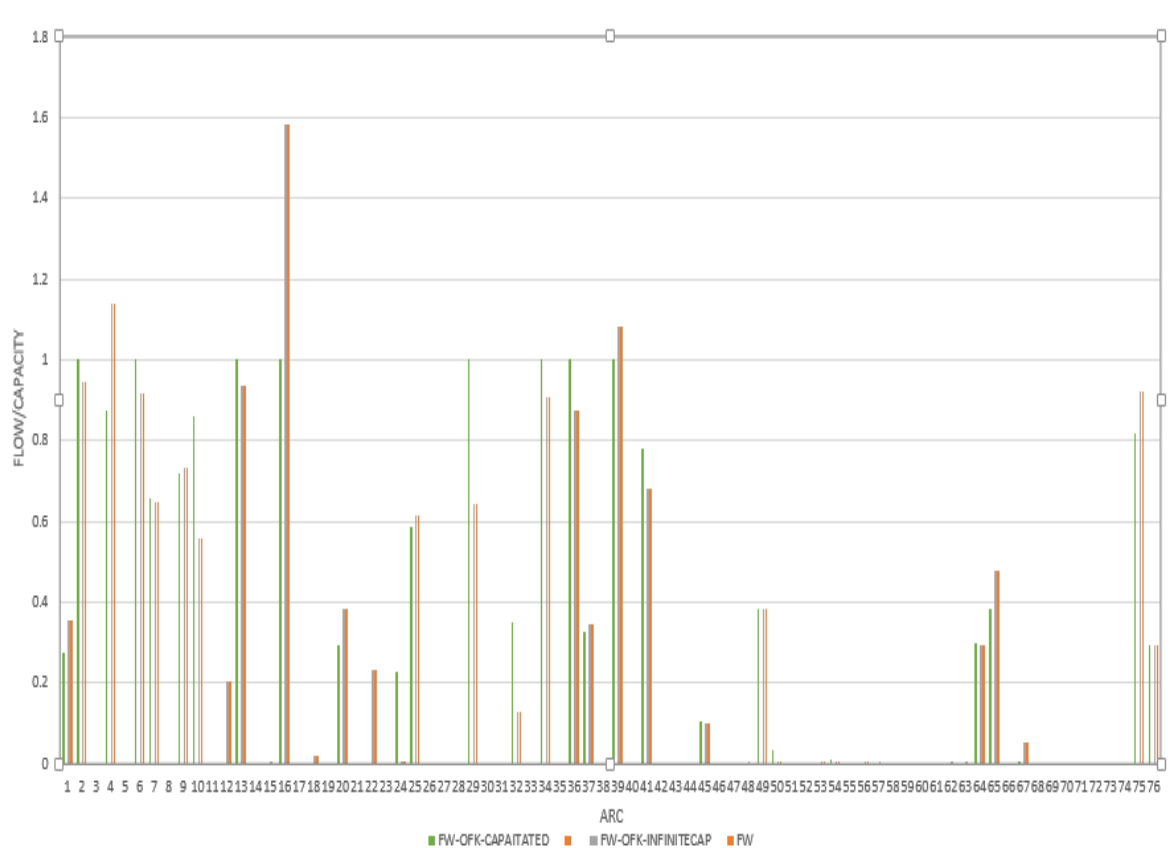


شکل ۴-۱۴- مقایسه جریان های بدست آمده از دو الگوریتم FW-OFK-CAPACITATED و FW-OFK-INFINITECAP



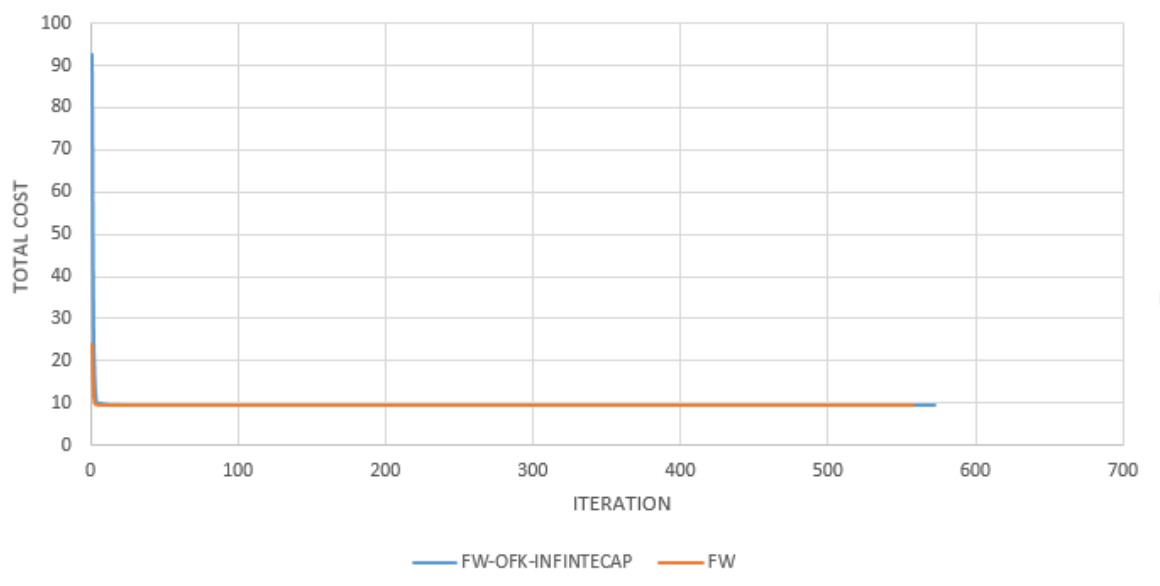
شکل ۴-۱۵- مقایسه هزینه های بدست آمده از دو الگوریتم FW-OFK-CAPACITATED و FW-OFK-INFINITECAP

در شکل ۴-۱۶ نسبت جریان به ظرفیت کمان ها برای سه الگوریتم FW-OFK، FW-INFINITECAP و FW-OFK-CAPACITATED به نمایش گذاشته شده است. همان طور که مشاهده می شود، این نسبت در دو الگوریتم اول، برای بعضی از کمان ها بیشتر از یک می باشد که در نتیجه باید زمان سفر آن ها را عدد بسیار بزرگی بدست بیاورد ولی از آنجا که به دلیل سهولت ظرفیت بی نهایت در نظر گرفته شده است زمان سفر این کمان ها مقدار محدودی بدست آمده است. علاوه بر زمان سفر کمان ها ، بزرگتر بودن جریان از ظرفیت روی هزینه کل و هزینه مصرف کننده نیز تاثیر دارد و مقادیر آنها را نیز باید بسیار بزرگ بدست بیاید که در نتیجه های بدست آمده از الگوریتم های بدون محدودیت ظرفیت مشاهده نمی شود.

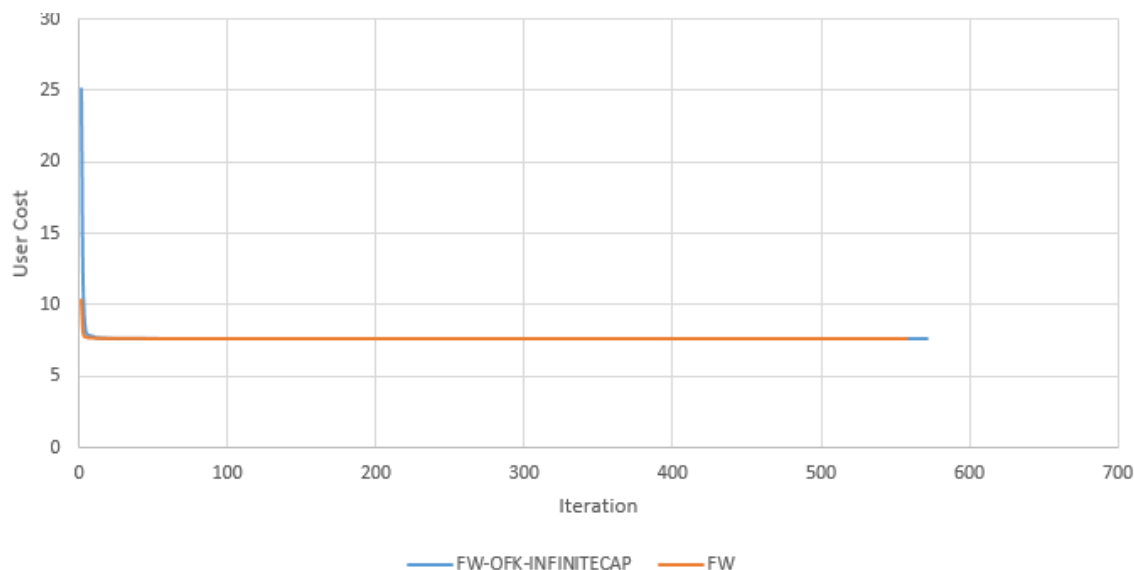


شکل ۴-۱۶-مقایسه مقادیر جریان بر ظرفیت برای الگوریتم های FW-OFK، FW-CAPACITATED و FW-OFK-INFINITECAP

شکل ۴-۱۷ به مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK-INFINITECAP می پردازد. الگوریتم FW-OFK-INFINITECAP در شبکه سופالز هزینه کل اولیه بیشتری نسبت به الگوریتم FW دارد و در تعداد تکرارهای بیشتری به دقت مورد نظر می رسد. بنابراین در مورد شبکه سופالز با یک مبدا می توان گفت که الگوریتم FW کارا تر از الگوریتم FW-OFK-INFINITECAP می باشد. در ادامه برای اجرای الگوریتم بر روی شبکه با ابعاد واقعی تر، به تحلیل و بررسی نتایج حاصل از اعمال الگوریتم FW-OFK بر روی شبکه سופالز با دو مبدا می پردازیم. شکل ۴-۱۸ نیز تغییرات هزینه مصرف کننده را برای دو الگوریتم FW و FW-OFK-INFINITECAP نشان می دهد. در شکل ۴-۱۷ نیز مشاهده می شود که الگوریتم FW با سرعت بیشتر و هزینه مصرف کننده اولیه کمتری نسبت به الگوریتم FW-OFK همگرا می شود.



شکل ۴-۱۷-مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK-INFINITECAP



شکل ۴-۱۸- تغییرات هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK-INFINITECAP

همانطور که پیش از این گفته شد الگوریتم OFK قادر به تشخیص تقاضا به تفکیک مبدا-مقصد نمی باشد. برای حل این مشکل ما تقاضا را در دو مرحله به الگوریتم اعمال می کنیم. در مرحله اول جواب های اولیه الگوریتم همه با هیچ ناشی از تقاضا مبدا اول به عنوان ورودی الگوریتم OFK داده می شود و جریان ها به روز رسانی می شود و در آرایه ای جدا ذخیره می شود. در مرحله دوم همین اقدامات برای مبدا دوم اجرا می شود. جریان های بدست آمده از مرحله اول و دوم با هم جمع شده و به عنوان ورودی به الگوریتم فرانک-ولف داده می شود. مشخصات کمان های شبکه سوفالز در حالت دومبدایی در جدول ۵۲، ۵۳ و ۵۴ پیوست آمده است. در این حالت نیز عرضه و تقاضا در مبدا و مقصدها پنج برابر شده است.

جدول ۳۵ نتایج اعمال الگوریتم های FW، FW-OFK-INFINITECAP، و FW-OFK-CAPACITATED را نشان می دهد. در شبکه سوفالز دو مبدايي الگوریتم FW و FW-OFK-INFINITECAP با تعداد تکرارهای نزدیک به هم به دقت 10^{-4} می رسند.

در مورد الگوریتم FW-OFK-CAPACITATED باید اشاره کرد که برای حل شبکه سوفالز دومبدایی همانطور که اشاره شد ابتدا جریان های اولیه ناشی از تقاضا مبدا اول را به الگوریتم داده و جریان ها را بروز رسانی شده سپس همین موارد برای مبدا دوم انجام می شود. ولی بعد از اعمال الگوریتم برای مبدا اول باید حد بالای ظرفیت اصلاح شود چرا که بعد از اعمال الگوریتم OFK برای مبدا اول جریان هایی از کمان عبور می کنند باعث می شوند که حد بالا کاهش یابد. این روش برای دو مبدا قابل اجرا نیست چرا که نمی توان حد بالای ظرفیت را تجزیه کرد. حد بالای ظرفیت را نمی توان به صورت دو عدد جدا برای دو مبدا تعریف کرد. بنابراین در این پژوهش حد بالای ظرفیت به صورت یک عدد ثابت و بدون تغییر برای دو مبدا در نظر گرفته می شود. در نتیجه اعداد بدست آمده از الگوریتم FW-OFK-CAPACITATED با الگوریتم DPF متفاوت می باشد. هزینه مصرف کننده در این حالت بین هزینه

مصرف کننده بدست آمده از الگوریتم FW و DPF می باشد. در نتیجه جواب های بدست آمده از جواب های FW به واقعیت نزدیک تر می باشند چرا که در این حالت ظرفیت ما بی نهایت نمی باشد بلکه ظرفیت ما یک عدد محدود می باشد که در بعضی کمان ها از جریان کوچکتر می باشد.

جدول ۳۵- نتایج بدست آمده برای شبکه سופالز دو مبدایی

بدون ظرفیت	Algorithm	TC	UC	RG	Iteration
	FW	24.681	13.366	10^{-4}	1318
	FW-OFK	24.680	13.366	10^{-4}	1326
با ظرفیت	FW-OFK	24.511	13.402	10^{-4}	637
	DPF	26.964	13.688	10^{-6}	500

در جدول ۳۴ مقادیر هزینه کل، هزینه مصرف کننده، شکاف نسبی، α برای الگوریتم فرانک-ولف و ماکزیمم جریان بر ظرفیت را برای چند تکرار الگوریتم های FW-OFK، FW-OFK-INFINITECAP، FW-CAPACITATED برای شبکه سופالز مشاهده می کنید. در جدول مشاهده می کنید که مقدار جریان بر ظرفیت نهایی در دو الگوریتم FW و FW-OFK-INFINITECAP برابر با ۲،۴ می باشد در حالیکه همین مقدار در الگوریتم FW-OFK-CAPACITATED برابر با ۱،۴۶ می باشد که نشان دهنده آن است که جواب های بدست آمده از این الگوریتم بهتر و نزدیکتر به واقعیت هستند چرا که مقدار آن ها نزدیکتر به ظرفیت می باشند.

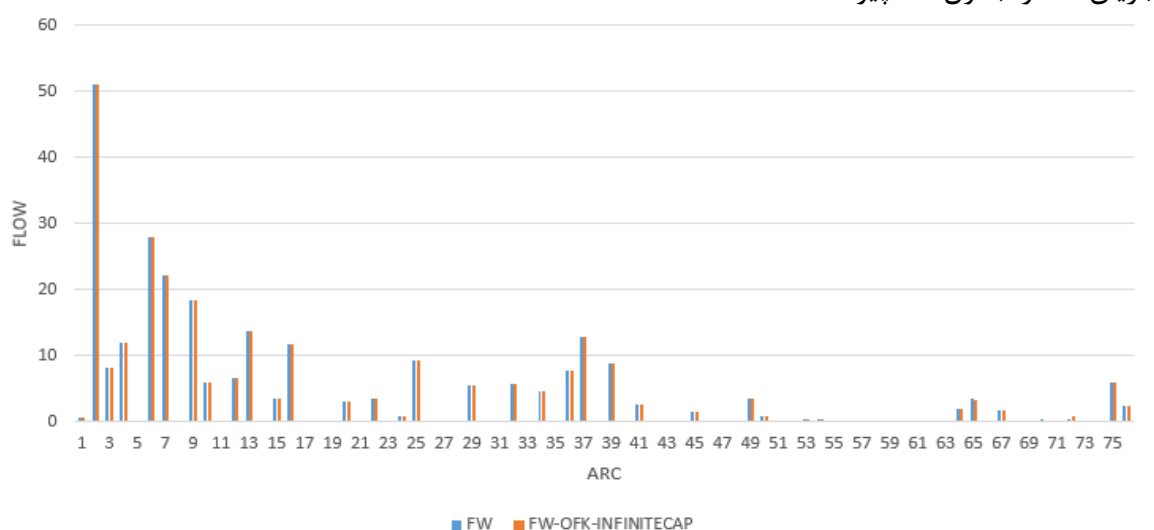
جدول ۳۶- مقایسه نتایج حاصل از سه الگوریتم FW،FW-OFK-INFINITECAP،FW-OFK

CAPACITATED در تکرار های مختلف برای شبکه سوفالز دو مبدایی

الگوریتم	شماره تکرار	هزینه کل	هزینه مصرف کننده	شکاف نسبی	α	Max(v/C)
FW	1	35.22	16.40	0.407	0.125	3.22
	2	28.55	15.19	0.197	0.187	2.82
	3	25.80	14.78	0.087	0.094	2.29
	4	26.08	14.68	0.087	0.094	2.53
	5	25.63	14.55	0.063	0.719	2.38
	1318	24.68	13.37	0.00003	0.0008	2.40
FW-OFK-INFINITECAP	1	148.53	39.80	0.918	0.25	4.79
	2	58.24	21.58	0.647	0.25	3.59
	3	31.40	16.30	0.275	0.375	2.69
	4	26.36	14.70	0.088	0.156	2.65
	5	26.56	14.55	0.093	0.061	2.41
	1326	24.68	13.37	0.000009	0.0003	2.40
FW-OK-CAP	1	47.15	19.45	0.399	0.625	1.63

	2	27.43	14.58	0.157	0.187	1.59
	3	25.19	14.27	0.071	0.156	1.56
	4	25.37	14.12	0.047	0.172	1.49
	5	24.82	14.03	0.045	0.109	1.49
	673	24.51	13.40	0.00009	0.001	1.46

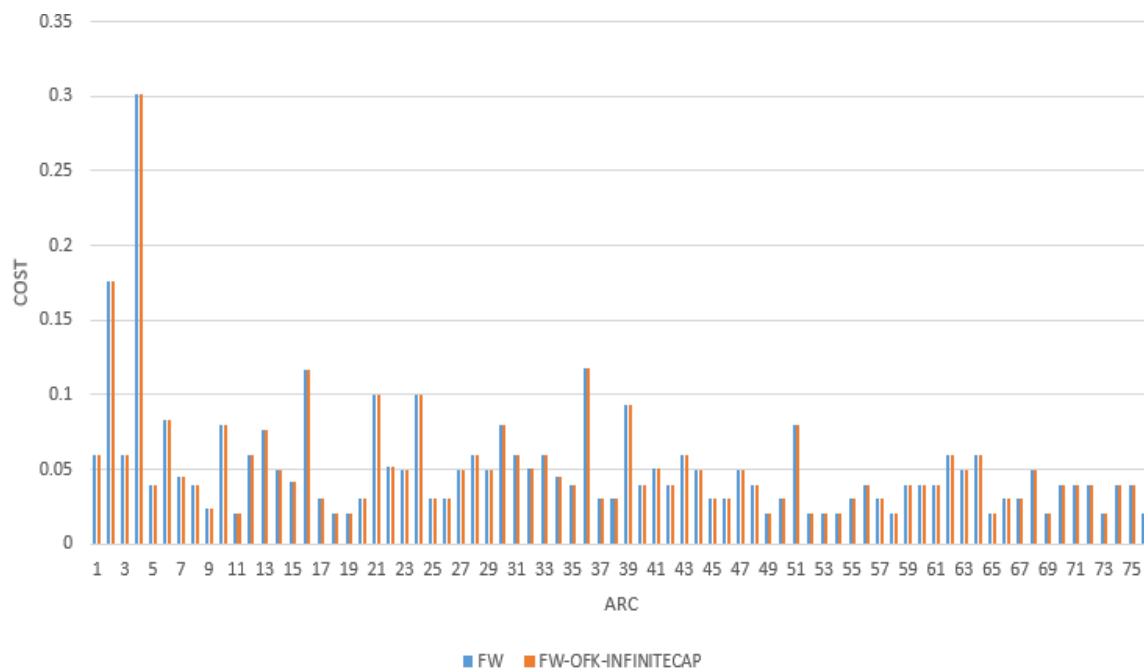
در شکل ۴-۱۹ به مقایسه جریان های بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP پرداخته شده است. در این حالت مانند شبکه های قبل چون ظرفیت را برای شبکه بی نهایت در نظر گرفته ایم، جریان های به دست آمده از الگوریتم FW-OFK-INFINITECAP بسیار نزدیک و در چندین مورد مطابق جریان های بدست آمده از الگوریتم FW می باشد. نتایج عددی جریان ها در جدول ۳۹ پیوست آمده است.



شکل ۴-۱۹-مقایسه جریان های بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP

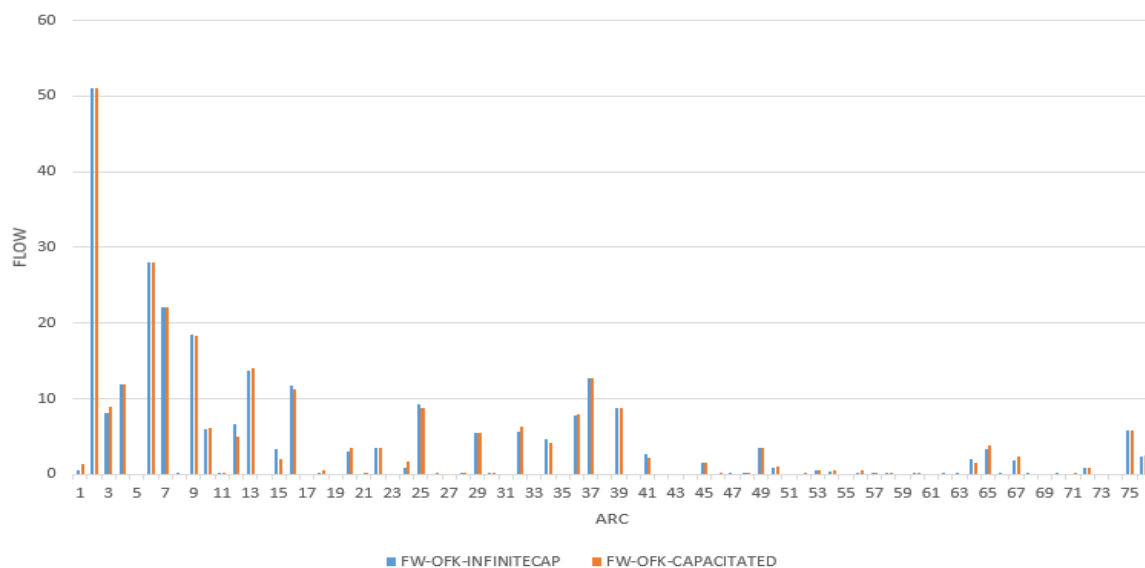
شکل ۴-۲۰ مقایسه هزینه های بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP را نشان میدهد. در این حالت مانند شبکه های قبل چون جریان های بدست آمده بسیار نزدیک بهم می باشد، هزینه ها نیز تفاوت کمی باهم دارند. نتایج عددی هزینه ها در جدول ۴۰ پیوست آمده است. بدین ترتیب می توان گفت در الگوریتم FW-OFK-INFINITECAP در صورتی که تقاضا را در شبکه هایی که چند مبدا دارند

یا یک یا چند مبدا در آن ها جریان ورودی و یک یا چند مقصد در آن ها جریان خروجی داشته باشند به صورت مرحله ای اعمال کرد، نتایج بدست آمده را می توان قابل استناد یافت همان طور که در جدول ۳۵ نیز همین نتایج را نشان می دهد .

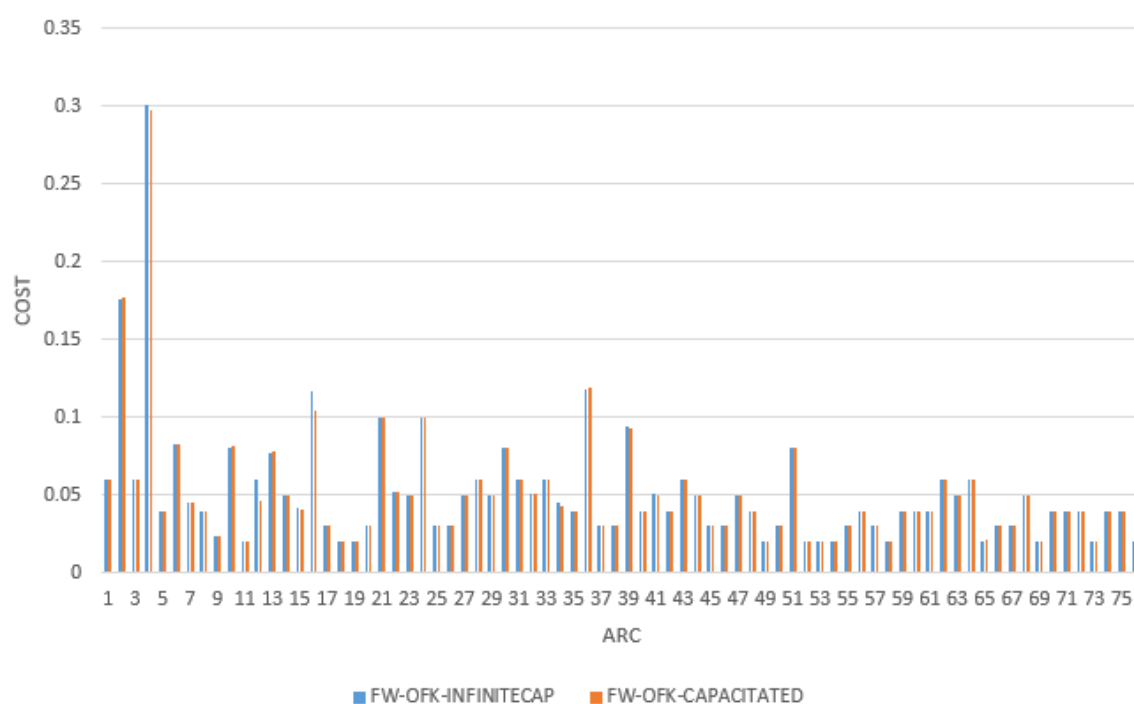


شکل ۴-۲۰- مقایسه هزینه های بدست آمده از الگوریتم های FW و FW-OFK-INFINITECAP

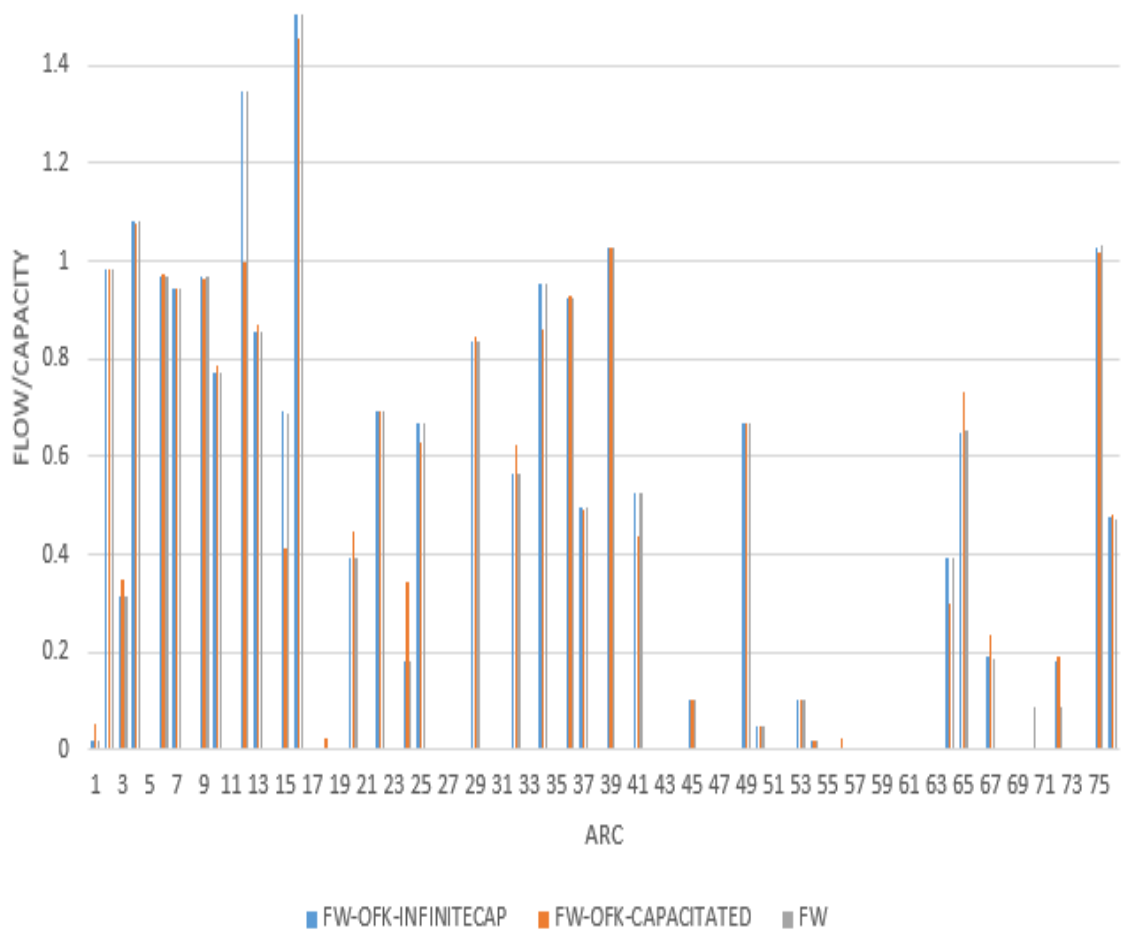
در شکل ۴-۲۱ مقایسه بین جریان های بدست آمده از الگوریتم های FW-OFK- و CAPACITATED و FW-OFK-INFINITECAP انجام شده است. در شکل ۴-۲۲ نیز مقایسه بین هزینه سفر های بدست آمده از این دو الگوریتم انجام شده است. در شکل ۴-۲۳ نیز مقادیر جریان بر ظرفیت برای کمان های شبکه به نمایش گذاشته شده است . در این حالت نیز می توان گفت با وجود آنکه جریان از ظرفیت بیشتر شده است هزینه ها مقدار محدودی می باشد. باید توجه داشت که برای کمان ها مقدار جریان بر ظرفیت بدست آمده از الگوریتم FW-OFK-CAPACITATED کمتر از الگوریتم FW می باشد. نتایج عددی جریان ها و هزینه سفرهای بدست آمده از الگوریتم FW-OFK- CAPACITATED به ترتیب در جدول ۳۹ و ۴۰ پیوست آمده است.



شکل ۴-۲۱- مقایسه بین جریان های بدست آمده از الگوریتم های FW-OFK-CAPACITATED و FW-OFK-INFINITECAP

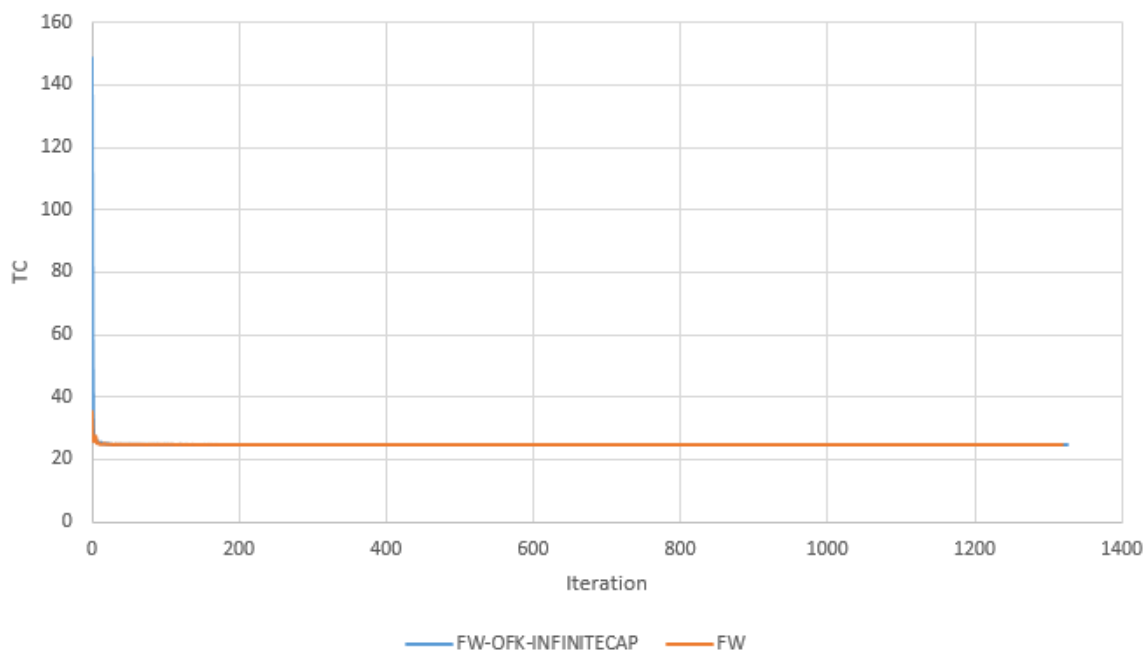


شکل ۴-۲۲- مقایسه بین هزینه سفر های بدست آمده از الگوریتم های FW-OFK-CAPACITATED و FW-OFK-INFINITECAP



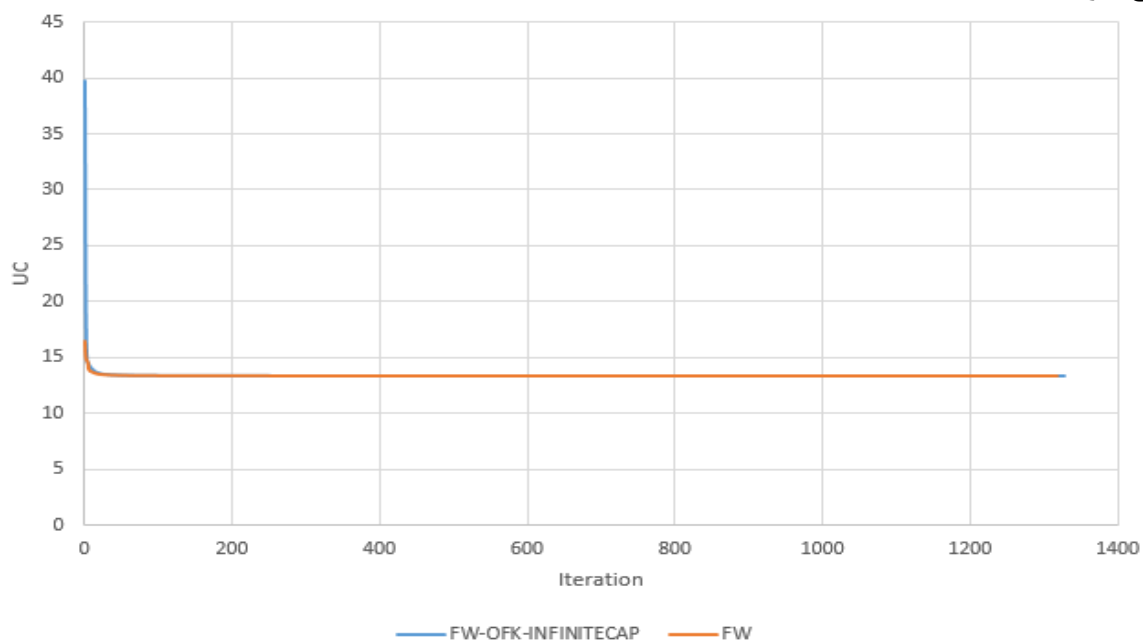
شکل ۴-۲۳- مقایسه مقادیر جریان بر ظرفیت برای الگوریتم های FW، FW-OFK، FW-OFK-INFINITECAP و CAPACITATED

در شکل ۴-۲۴ مقایسه بین تغییرات هزینه کل برای دو الگوریتم FW و FW-OFKINFINITECAP را نشان می دهد. همان طور که در شکل مشاهده می شود الگوریتم FW با هزینه اولیه کمتر و تعداد تکرار کمتر نسبت به الگوریتم دیگر به جواب نهایی همگرا می شود. با توجه به این شبکه و شبکه قبل می توان چنین استنباط کرد که الگوریتم FW-OFK در شبکه هایی که ظرفیت کمان ها بینهایت باشد کارایی چندانی نسبت به الگوریتم FW ندارد چراکه هزینه اولیه بیشتری دارد و در تعداد تکرارهای بیشتری نیز همگرا می شود. از الگوریتم FW-OFK-INFINITECAP می توان برای درستی اطلاعات ورودی و ... استفاده کرد.



شکل ۴-۲۴- مقایسه تغییرات هزینه کل برای دو الگوریتم FW و FW-OFK-INFINITECAP

در شکل ۴-۲۵ تغییرات هزینه مصرف کننده مقایسه شده است. هزینه مصرف کننده نیز مانند هزینه کل در الگوریتم FW با تعداد تکرار کمتر و مقدار اولیه کمتری نسبت به الگوریتم FW-OFK همگرا می شود.



شکل ۴-۲۵- مقایسه تغییرات هزینه مصرف کننده برای دو الگوریتم FW و FW-OFK-INFINITECAP

۵ فصل پنجم

خلاصه و پیشنهادات

۵-۱- مقدمه

در این فصل ابتدا خلاصه ای از کارهای انجام شده در این پایان نامه ارائه شده و سپس پیشنهاداتی برای کارهای آتی مطرح می شوند.

۵-۲- خلاصه

هدف این پژوهش بررسی اهمیت در نظر گرفتن ظرفیت برای کمان های شبکه حمل و نقلی بر مسئله تخصیص ترافیک می باشد. ، تا کنون پژوهش های بسیار کمی به این مسئله توجه کردند که شاید دلیل آن مشکلاتی که برنامه نویسی پیچیده و زمان حل زیاد این مسئله برای شبکه های بزرگ مقیاس دارد، باشد . در این پایان نامه ابتدا مسئله تخصیص ترافیک با ظرفیت محدودیت به صورت ریاضی تعریف شده و نتایج حل مسئله طراحی شبکه برای دو شبکه کوچک-مقیاس با هزینه سفر کمان ثابت ارائه شد . سپس نحوه تعریف ساختار شبکه مورد استفاده در الگوریتم Out of Kilter بیان شده و مسئله برای شبکه های معروف هرن و سوفالز با هزینه سفر کمان وابسته به جریان اعمال شده و نتایج مورد بحث و بررسی قرار گرفته اند. نتایج نشان داد در نظر گرفتن ظرفیت برای کمان های شبکه می تواند بر جواب مسئله بسیار موثر باشد.

۵-۳- پیشنهادات

انجام پژوهش های زیر در مطالعات آتی پیشنهاد می شوند.

- حل مشکل چند مبدایی الگوریتم OFK و مقایسه نتایج آن با الگوریتم FW
- اعمال الگوریتم OFK برای شبکه بزرگ و مقایسه زمان حل با الگوریتم های DPF و FW
- استفاده از جواب های اولیه بدست آمده از الگوریتم OFK برای الگوریتم DPF

پیوست

جدول ۳۷- جریان های بدست آمده از سه الگوریتم FW-OFK-،FW-OFK-INFINITECAP،FW

CAPACITATED برای شبکه سופالز تک مبدایی

FW		FW-OFK-INFINITECAP		FW-OFK-CAPACITATED	
جریان	ظرفیت	جریان	ظرفیت	جریان	ظرفیت
9.156	400	9.154	400	7.16	25.90
34.844	400	34.846	400	36.84	36.84
0.00	400	0	400	0.00	25.90
8.660	400	8.653	400	6.66	7.60
0.00	400	0	400	0.00	23.40
19.173	400	19.20	400	20.91	20.91
15.171	400	15.167	400	15.42	23.40
0	400	0	400	0.00	17.11
13.053	400	13.054	400	12.81	17.78
3.619	400	3.625	400	5.60	6.51
0.00	400	0.00	400	0.00	17.78
1.01	400	1.017	400	0.00	4.95
11.043	400	11.039	400	11.81	11.81
0.00	400	0.00	400	0.00	4.96
0.00	400	0.001	400	0.00	4.95
8.166	400	8.169	400	5.16	5.16
0.00	400	0.00	400	0.00	7.84
0.502	400	0.502	400	0.00	23.40
0.00	400	0.00	400	0.00	4.90
3.00	400	3.00	400	2.30	7.84
0.00	400	0.00	400	0.00	5.05

1.168	400	1.172	400	0.00	5.04
0.00	400	0.003	400	0.00	10
0.002	400	8.536	400	1.14	5.05
8.541	400	0.00	400	8.17	13.91
0.00	400	0.00	400	0.00	13.91
0.00	400	0.00	400	0.00	10
0.00	400	3.336	400	0.00	13.51
3.338	400	0.00	400	5.20	5.20
0.00	400	0.00	400	0.00	4.99
0.00	400	0.00	400	0.00	4.91
1.297	400	1.300	400	3.53	10
0.00	400	0.00	400	0.00	4.91
4.993	400	5.000	400	5.50	5.50
0.00	400	0.00	400	0.00	23.40
5.171	400	5.172	400	5.92	5.92
9.00	400	8.994	400	8.50	25.90
0.00	400	0.00	400	0.00	25.90
6.500	400	6.494	400	6.00	6.00
0.00	400	0.00	400	0.00	4.88
3.493	400	3.50	400	4.00	5.12
0.00	400	0.00	400	0.00	4.92
0.00	400	0.00	400	0.00	13.51
0.00	400	0.00	400	0.00	5.12
1.496	400	1.496	400	1.50	14.56
0.00	400	0.00	400	0.00	9.59

0.00	400	0.00	400	0.00	5.04
0.00	400	0.001	400	0.00	4.85
2.004	400	2.004	400	2.00	5.23
0.002	400	0.003	400	0.70	19.68
0.00	400	0.00	400	0.00	4.99
0.00	400	0.00	400	0.00	5.23
0.004	400	0.004	400	0.00	4.82
0.001	400	0.001	400	0.20	23.40
0.00	400	0.00	400	0.00	19.68
0.003	400	0.003	400	0.00	23.40
0.00	400	0.001	400	0.00	14.56
0.00	400	0.00	400	0.00	4.82
0.00	400	0.00	400	0.00	5.00
0.00	400	0.00	400	0.00	23.40
0.00	400	0.00	400	0.00	5.00
0.00	400	0.00	400	0.00	5.06
0.00	400	0.00	400	0.00	5.07
1.497	400	1.50	400	1.50	5.05
2.503	400	2.50	400	2.00	5.22
0.00	400	0.00	400	0.00	4.89
0.501	400	0.480	400	0.00	9.59
0.00	400	0.00	400	0.00	5.07
0.00	400	0.00	400	0.00	5.22
0.00	400	0.00	400	0.00	5
0.00	400	0.00	400	0.00	4.92

0.00	400	0.00	400	0.00	5
0.00	400	0.00	400	0.00	5.07
0.00	400	0.00	400	0.00	5.09
4.50	400	4.494	400	4.00	4.88
1.50	400	1.50	400	1.50	5.08

جدول ۳۸- هزینه سفرهای بدست آمده از سه الگوریتم FW،FW-OFK-INFINITECAP،FW-OFK-

CAPACITATED برای شبکه سופالز تک مبدایی

FW	FW-OFK-INFINITECAP	FW-OFK-CAPACITATED
0.0601	0.0601	0.0600
0.0695	0.0694	0.0768
0.0600	0.0600	0.0600
0.1200	0.1190	0.0744
0.0400	0.0400	0.0400
0.0495	0.0495	0.0533
0.0411	0.0412	0.0411
0.0400	0.0400	0.0400
0.0209	0.0209	0.0208
0.0627	0.0627	0.0753
0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0612	0.0611	0.0646
0.0500	0.0500	0.0500
0.0400	0.0400	0.0400
0.0432	0.0432	0.0237
0.030	0.0300	0.0300
0.020	0.0200	0.0200
0.020	0.0200	0.0200
0.0301	0.0301	0.0300
0.1000	0.1000	0.1000
0.0500	0.0500	0.0500

0.0500	0.0500	0.0500
0.1000	0.1000	0.1000
0.0306	0.0306	0.0305
0.0300	0.0300	0.0300
0.0500	0.0500	0.0500
0.0600	0.0600	0.0600
0.0413	0.0413	0.0479
0.0800	0.0800	0.0800
0.0600	0.0600	0.0600
0.0500	0.0500	0.0501
0.0600	0.0600	0.0600
0.0466	0.0466	0.0497
0.0400	0.0400	0.0400
0.0711	0.0711	0.0791
0.0300	0.0300	0.0300
0.0300	0.0300	0.0300
0.0559	0.0559	0.0516
0.0400	0.0400	0.0400
0.0516	0.0516	0.0528
0.0400	0.0400	0.0400
0.0600	0.0600	0.0600
0.0500	0.0500	0.0500
0.0300	0.0300	0.0300
0.0300	0.0300	0.0300
0.0500	0.0500	0.0500

0.0400	0.0400	0.0400
0.0200	0.0201	0.0200
0.0300	0.0300	0.0300
0.0800	0.0800	0.0800
0.0200	0.0200	0.0200
0.0200	0.0200	0.0200
0.0200	0.0200	0.0200
0.0300	0.0300	0.0300
0.0400	0.0400	0.0400
0.0300	0.0300	0.0300
0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400
0.0600	0.0600	0.0600
0.0500	0.0500	0.0500
0.0601	0.0601	0.0600
0.0202	0.0202	0.0200
0.0300	0.0300	0.0300
0.0300	0.0300	0.0300
0.0500	0.0500	0.0500
0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400

0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0332	0.0332	0.0320
0.0200	0.0200	0.0200

جدول ۳۹- جریان های بدست آمده از سه الگوریتم FW-OFK-،FW-OFK-INFINITECAP،FW

CAPACITATED برای شبکه سوفالز دو مبدایی

FW		FW-OFK-INFINITECAP		FW-OFK-CAPACITATED	
جریان	ظرفیت	جریان	ظرفیت	جریان	ظرفیت
0.5160	400	0.5131	400	1.3625	25.90
51.0746	400	51.075	400	51.1200	51.99
8.0906	400	8.0879	400	8.9824	25.90
11.9254	400	11.9252	400	11.8801	11.04
0.0000	400	0.0000	400	0.0000	23.40
27.9559	400	27.9530	400	27.9747	28.80
22.1186	400	22.1226	400	22.1452	23.40
0.0000	400	0.0009	400	0.0000	17.11
18.4542	400	18.4546	400	18.3704	19.03
6.0022	400	6.0005	400	6.1048	7.76
6.0005	400	0.0030	400	0.0005	17.78
6.6627	400	6.6631	400	4.9413	4.95
13.7066	400	13.7053	400	13.9588	16.06
0.0000	400	0.0000	400	0.0000	4.96
3.4157	400	3.4168	400	2.0302	4.95
11.6725	400	11.6715	400	11.2911	7.75
0.0000	400	0.0000	400	0.0000	7.84
0.0114	400	0.0112	400	0.4972	23.40
0.0000	400	0.0000	400	0.000	4.90
3.0924	400	3.0892	400	3.5218	7.84
0.0000	400	0.0000	400	0.0000	5.05

3.4945	400	3.4939	400	3.4966	5.04
0.0000	400	0.0000	400	0.0000	10
0.9123	400	0.9107	400	1.7279	5.05
0.2943	400	9.2946	400	7.8320	13.91
0.0000	400	0.0000	400	0.0004	13.91
0.0000	400	0.0000	400	0.0000	10
0.0005	400	0.0020	400	0.0004	13.51
5.4222	400	5.4249	400	5.4741	6.49
0.0004	400	0.0005	400	0.0021	4.99
0.0000	400	0.00	400	0.0000	4.91
5.6288	400	5.6316	400	6.2445	10
0.0000	400	0.0000	400	0.0000	4.91
4.6978	400	4.6914	400	4.2301	4.92
0.0000	400	0.0000	400	0.0000	23.40
7.8244	400	7.8224	400	7.8698	8.46
12.7943	400	12.8002	400	12.7754	25.90
0.0000	400	0.0000	400	0.0000	25.90
8.7943	400	8.8002	400	8.7754	8.55
0.00	400	0.00	400	0.0000	4.88
2.6953	400	2.6914	400	2.2304	5.12
0.0028	400	0.0000	400	0.0000	4.92
0.0000	400	0.0000	400	0.0000	13.51
0.0000	400	0.0000	400	0.0000	5.12
1.5033	400	1.5044	400	1.5020	14.56
0.0000	400	0.0000	400	0.0001	9.59

0.0032	400	0.0008	400	0.0000	5.04
0.00	400	0.0013	400	0.0005	4.85
3.4977	400	3.4955	400	3.4971	5.23
0.9158	400	0.9212	400	0.9739	19.68
0.00	400	0.00	400	0.0000	4.99
0.00	400	0.00	400	0.0009	5.23
0.4981	400	0.4980	400	0.4985	4.82
0.4179	400	0.4219	400	0.4754	23.40
0.00	400	0.00	400	0.0000	19.68
0.0096	400	0.0108	400	0.4961	23.40
0.00	400	0.0004	400	0.0001	14.56
0.00	400	0.0020	400	0.0002	4.82
0.0013	400	0.00	400	0.0000	5.00
0.0003	400	0.0004	400	0.0003	23.40
0.00	400	0.00	400	0.0000	5.00
0.00	400	0.0001	400	0.0000	5.06
0.00	400	0.0005	400	0.0000	5.07
1.9894	400	1.2000	400	1.5042	5.05
3.4140	400	3.3935	400	3.8239	5.22
0.00	400	0.0001	400	0.0000	4.89
1.8074	400	1.8107	400	2.2709	9.59
0.00	400	0.0003	400	0.0000	5.07
0.0001	400	0.00	400	0.0000	5.22
0.4460	400	0.0002	400	0.0000	5
0.0003	400	0.00	400	0.0003	4.92

0.4474	400	0.9171	400	0.9469	5
0.00	400	0.00	400	0.0000	5.07
0.00	400	0.00	400	0.0000	5.09
5.9033	400	5.8833	400	5.8281	5.71
2.3909	400	2.4169	400	2.4473	5.08

جدول ۴۰- هزینه سفر های بدست آمده از سه الگوریتم FW،FW-OFK-INFINITECAP،FW-OFK-

CAPACITATED برای شبکه سوفالز دو مبدایی

FW	FW-OFK-INFINITECAP	FW-OFK-CAPACITATED
0.0600	0.0600	0.0600
0.176	0.176	0.1766
0.0601	0.0601	0.0601
0.301	0.301	0.2972
0.0400	0.0400	0.0400
0.0828	0.0827	0.0828
0.0448	0.0448	0.0448
0.0400	0.0400	0.0400
0.0235	0.0235	0.0234
0.0801	0.0801	0.0815
0.0200	0.0200	0.0200
0.0597	0.0597	0.0459
0.0764	0.0765	0.0785
0.0500	0.0500	0.0500
0.0414	0.0414	0.0402
0.1167	0.1167	0.1047
0.030	0.0300	0.0300
0.020	0.0200	0.0200
0.020	0.0200	0.0200
0.0301	0.0301	0.0302
0.1000	0.1000	0.1000
0.0517	0.0517	0.0517

0.0500	0.0500	0.0500
0.1000	0.1000	0.1000
0.0309	0.0309	0.0307
0.0300	0.0300	0.0300
0.0500	0.0500	0.0500
0.0600	0.0600	0.0600
0.0493	0.0493	0.0497
0.0800	0.0800	0.0800
0.0600	0.0600	0.0600
0.0507	0.0507	0.0511
0.0600	0.0600	0.0600
0.0451	0.0451	0.0434
0.0400	0.0400	0.0400
0.1181	0.1180	0.1190
0.0303	0.0303	0.0303
0.0300	0.0300	0.0300
0.0934	0.0936	0.0930
0.0400	0.0400	0.0400
0.0506	0.0506	0.0503
0.0400	0.0400	0.0400
0.0600	0.0600	0.0600
0.0500	0.0500	0.0500
0.0300	0.0300	0.0300
0.0300	0.0300	0.0300
0.0500	0.0500	0.0500

0.0400	0.0400	0.0400
0.0206	0.0206	0.0206
0.0300	0.0300	0.0300
0.0800	0.0800	0.0800
0.0200	0.0200	0.0200
0.0200	0.0200	0.0200
0.0200	0.0200	0.0200
0.0300	0.0300	0.0300
0.0400	0.0400	0.0400
0.0300	0.0300	0.0300
0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400
0.0600	0.0600	0.0600
0.0500	0.0500	0.0500
0.0602	0.0602	0.0607
0.0205	0.0205	0.0209
0.0300	0.0300	0.0300
0.0300	0.0300	0.0300
0.0500	0.0500	0.0500
0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400
0.0400	0.0400	0.0400

0.0200	0.0200	0.0200
0.0400	0.0400	0.0400
0.0396	0.0395	0.0391
0.0201	0.0200	0.0200

جدول ۴۱- مشخصات کمان های شبکه سופالز تک مبدایی

ظرفیت	زمان سفر آزاد	کمان
25.90	0.06	1
36.84	0.04	2
25.90	0.06	3
7.60	0.05	4
23.40	0.04	5
20.91	0.04	6
23.40	0.04	7
17.11	0.04	8
17.78	0.02	9
6.51	0.06	10
17.78	0.02	11
4.95	0.04	12
11.81	0.05	13
4.96	0.05	14
4.95	0.04	15
5.15	0.02	16
7.84	0.03	17
23.40	0.02	18
7.84	0.02	19
4.90	0.03	20

21	0.1	5.05
22	0.05	5.04
23	0.05	10
24	0.1	5.05
25	0.03	13.91
26	0.03	13.91
27	0.05	10
28	0.06	13.51
29	0.04	5.19
30	0.08	4.99
31	0.06	4.91
32	0.05	10
33	0.06	4.91
34	0.04	5.49
35	0.04	23.40
36	0.06	5.92
37	0.03	25.90
38	0.03	25.90
39	0.04	6.00
40	0.04	4.88
41	0.05	5.12
42	0.04	4.92

43	0.06	13.51
44	0.05	5.12
45	0.03	14.56
46	0.03	9.59
47	0.05	5.04
48	0.04	4.85
49	0.02	5.23
50	0.03	19.68
51	0.08	4.99
52	0.02	5.23
53	0.02	4.82
54	0.02	23.40
55	0.03	19.68
56	0.04	23.40
57	0.03	14.56
58	0.02	4.82
59	0.04	5.00
60	0.04	23.40
61	0.04	5.00
62	0.06	5.06
63	0.05	5.07
64	0.06	5.06

65	0.02	5.22
66	0.03	4.89
67	0.03	9.59
68	0.05	5.07
69	0.02	5.22
70	0.04	5
71	0.04	4.92
72	0.04	5
73	0.02	5.07
74	0.04	5.09
75	0.03	4.88
76	0.02	5.08

جدول ۴۲- عرضه و تقاضا در مبدا- مقصدهای شبکه سופالز تک مبدایی

مبدا/مقصد	1
2	0.5
3	0.5
4	2.5
5	1.0
6	1.5
7	2.5
8	4.0
9	2.5
10	6.5
11	2.5
12	1.0
13	2.5
14	1.5
15	2.5
16	2.5
17	2.0
18	0.5
19	1.5
20	1.5
21	0.5
22	2.0
23	1.5
24	0.5

جدول ۴۳- مشخصات کمان های شبکه سופالز دو مبدایی

کمان	زمان سفر آزاد	ظرفیت
1	0.06	25.90
2	0.04	51.99
3	0.06	25.90
4	0.05	11.04
5	0.04	23.40
6	0.04	28.80
7	0.04	23.40
8	0.04	17.11
9	0.02	19.03
10	0.06	7.76
11	0.02	17.78
12	0.04	4.95
13	0.05	16.06
14	0.05	4.96
15	0.04	4.95
16	0.02	7.75
17	0.03	7.84
18	0.02	23.40
19	0.02	4.89
20	0.03	7.84

21	0.1	5.05
22	0.05	5.04
23	0.05	10
24	0.1	5.05
25	0.03	13.91
26	0.03	13.91
27	0.05	10
28	0.06	13.51
29	0.04	6.49
30	0.08	4.99
31	0.06	4.91
32	0.05	10
33	0.06	4.91
34	0.04	4.91
35	0.04	23.40
36	0.06	8.64
37	0.03	25.90
38	0.03	25.90
39	0.04	8.55
40	0.04	4.88
41	0.05	5.12
42	0.04	4.92

43	0.06	13.51
44	0.05	5.12
45	0.03	14.56
46	0.03	9.59
47	0.05	5.04
48	0.04	4.85
49	0.02	5.23
50	0.03	19.68
51	0.08	4.99
52	0.02	5.23
53	0.02	4.82
54	0.02	23.40
55	0.03	19.68
56	0.04	23.40
57	0.03	14.56
58	0.02	4.82
59	0.04	5.00
60	0.04	23.40
61	0.04	5.00
62	0.06	5.06
63	0.05	5.07
64	0.06	5.06

65	0.02	5.22
66	0.03	4.89
67	0.03	9.59
68	0.05	5.07
69	0.02	5.22
70	0.04	5
71	0.04	4.92
72	0.04	5
73	0.02	5.07
74	0.04	5.09
75	0.03	4.88
76	0.02	5.08

جدول ۴۴- عرضه و تقاضا در مبدا اول شبکه سופالز دو مبدایی

مبدا/مقصد	1
2	0.5
3	0.5
4	2.5
5	1.0
6	1.5
7	2.5
8	4.0
9	2.5
10	6.5
11	2.5
12	1.0
13	2.5
14	1.5
15	2.5
16	2.5
17	2.0
18	0.5
19	1.5
20	1.5
21	0.5
22	2.0
23	1.5
24	0.5

جدول ۴۵- عرضه و تقاضا در مبدا دوم شبکه سوفالز دو مبدایی

مبدا/مقصد	2
1	0.5
3	0.5
4	1.0
5	0.5
6	2.0
7	1.0
8	2.0
9	1.0
10	3.0
11	1.0
12	0.5
13	1.5
14	0.5
15	0.5
16	2.0
17	1.0
18	0.0
19	0.5
20	0.5
21	0.0
22	0.5
23	0.0
24	0.0

References

1. Aashtiani, H.Z. (1976), Solving large-scale network optimization problems by the Out-of-kilter method, Massachusetts Institute of Technology.
2. Aashtiani, H.Z. The Multi-Modal Traffic Assignment Problem. Ph.D. Dissertation Thesis, MIT, 1976.
3. Arzeki, Y. and Van Vliet, D. A Full Analytical Implementation of the PARTAN/Frank-Wolfe Algorithm for Equilibrium Assignment. *Transportation Science*, 24, 58-62, 1990 131
4. Azimi, G., Asgari, H., Rahimi, A., & Jin, X. (2019). *Investigation of heterogeneity in severity analysis for large truck crashes* (No. 19-02748).
5. Azimi, G., Rahimi, A., Asgari, H., & Jin, X. (2020). Severity analysis for large truck rollover crashes using a random parameter ordered logit model. *Accident Analysis & Prevention*, 135, 105355.
6. Bar-Gera H. Origin-Based Algorithms for Transportation Network Modeling, Technical Report #103, National Institute of Statistical Science. 1999
7. Bar-Gera H., Origin-Based Algorithm for the traffic assignment problem, *Transportation Science* 36 (4), 2002, pp.398-417.
8. Bar-Gera, H. Origin-based algorithms for transportation network modeling. National Institute of Statistical Sciences, Research Triangle Park, NC, 1999.
9. Beckman, M.J. and McGuire, C.B. and Winsten, C.B. (1956), *Studies in the Economics of Transportation*, Yale University Press, U.S.A.
10. Boyce D. and Jonson B.N. and Eash R.W., The effect on equilibrium trip assignment of different link congestion function, *Transportation Research* 15A, 1981, pp.223-232.
11. Chen A. and Lee D. H. and Jayakrishnan R., Path-based algorithms for large-scale traffic equilibrium problems: a comparison between DSD and GP, *Mathematics and Computers in Simulation*. 2001.
12. Chen, A. Effects of Flow Update Strategies on the Implementation of the FrankWolfe Algorithm for the Traffic Assignment Problem. *Transportation Research Board Annual Meeting*, Washington, DC, 2001 132
13. Chen, A., Jayakrishnan, R. A path based gradient projection algorithm: Effects of equilibration with a restricted path set under two flow policies, *Proceedings of the Transportation Research Board Annual Meeting*,
14. Chen, A., Lee, D. H., Jayakrishnan, R. and Nie, Y. Path and link based traffic assignment algorithms: a computational study. Working paper, 2000
15. Chen, A., R. Jayakrishnan, and W.K. Tsai. Faster Frank-Wolfe Traffic Assignment with New Flow Update Scheme. *Journal of Transportation Engineering*, Vol. 128, TECHNICAL PAPERS, 2002
16. Correa J.R. and Schulz A.S and Stier-Moses N.E., Selfish Routing in Capacitated Networks, *Mathematics of Operation research* 29, No 4, 2004, pp.961-976. 134
17. Deganzo C.F. On the traffic assignment problem with flow dependent cost-I, *Transportation research* 11, 1977, pp 433-437.

18. Deganzo C.F. On the traffic assignment problem with flow dependent cost-II, Transportation research 11, 1977, pp 439-441.
19. Dial, R. B. A Path-Based User-Equilibrium Traffic Assignment Algorithm
20. Emme. <http://www.inro.ca/en/products/emme/index.php>.
21. Florian, M. and He, Sh. Changing assignment algorithm: the price of better convergence. TRB Annual Meeting 2010, Washington DC
22. Florian, M. Guelat, J. and Spiess, H. An Efficient Implementation of the PARTAN Variant of Linear Approximation Method for Network Equilibrium Problem. Networks, 17, 1987, p.p. 319-339.
23. Frank, M. and P. Wolf. An Algorithm for Quadratic Programming, Naval Research Logistics Quarterly 3(1-2), pp. 95-110, 1956.
24. Fukushima, M. A Modified Frank-Wolfe Algorithm for Solving the Traffic Assignment Problem. Transportation Research, Vol. 18B, 1985, pp.169-177
25. Hearn D.W. and Ribera J., Bounded flow equilibrium problems by penalty methods, In Preceeding of IEEE International Conference on Circuits and Computers, pp.162-166. 1980.
26. Hearn D.W., Bounding flows in traffic assignment problems, Research Report 80-4 Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL. 1980.
27. Hestenes M.R. Multipliers and gradient methods, Journal Of Optimization theories and applications 4 , (1969), pp.303-320.
28. Inouye H., Traffic equilibria and its solution in congested road networks, In: R. Genser, (Ed.), Proceeding of IFAC Conference on Control in Transportation Systems, 1987, pp.267-272.
29. Khang D.B. and Fujiwara O. (1991), Approximate solutions of capacitated fixed-charge minimum cost network flow problems, Networks 21,689-704.
30. Larsson T. and Patriksson M., An Augmented Lagrangean Dual Algorithm for Link Capacity Side Constrained Traffic Assignment Problems, Transportation Research Part B, 1995, 29, pp. 433-455.
31. LeBlanc, L. J., E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation Research Part E: Logistics and Transportation Review, Vol.9, No. 3, 1975, pp. 308-318.
32. LeBlanc, L. J., R. V. Helgason, and D. E. Boyce. Improved Efficiency of the Frank-Wolfe Algorithm for Convex Network Problems. Transportation Science, Vol. 19, 1985, pp. 445-462.
33. Lee, D. H. and Nie, Y. Accelerating Strategies and Computational Studies of the Frank-Wolfe Algorithm for the Traffic Assignment Problem. TRB Annual Meeting 2001, Washington DC
34. Lemke, C. E. Bimatrix equilibrium points and mathematical programming, Management Science, Vol. 11, No. 7, Series A, pp. 681-689, 1965
35. Marcotte P. and Nguyen S. and Schoeb A., A Strategic Flow Model of Traffic Assignment in Static Capacitated Networks, Operation Research, 52, 2004, pp. 191-212.

36. Miller S.D. and Payne H.J. and Thompson W.A., An algorithm for traffic assignment on capacity constrained transportation networks with queues. Paper presented at the Johns Hopkins Conference on Information Sciences and Systems, The Johns Hopkins University, Baltimore, MD, 24 April. 1975.
37. Minoux, M. (1989). Network synthesis and optimum network design problems: Models, solution methods, and applications. *Networks* 19, 313–360.
38. Morowati-Shalilvand S. and Mehri-Tekmeh J., An extended originbased method for solving capacitated traffic assignment problem, *Acta Universitatis Apulensis*, No 35, pp.169-186, 2013.
39. Nie Y. and Zhang H.M. and Lee D.H., Models and Algorithms for the Traffic Assignment Problem with Link Capacity Constraints, *Transportation Research* 38B, 2004, pp.285-312.
40. Nie, Yu (Marco). A class of bush-based algorithms for the traffic assignment problem. *Transportation Research Part B: Methodological*, Volume 44, Issue 1. January 2010, Pages 73-89
41. Patriksson, M. *The Traffic Assignment Problem: Models and Methods*. Utrecht The Netherlands, 1994.
42. Powell M.J.D. A method for nonlinear constraints in minimization problems, In: Fletcher R., Editor, *Optimization*, Academic Press, New York, 1969, PP. 283-298.
43. Prashkar J.N. and Toledo T., A Gradient Projection Algorithm for Side constrained traffic assignment, *EJTIR*, 4, 2004, pp.177-193.
44. Rahimi, A. (2017). Incorporating Variable Demand Assignment Into Discrete Network Design Problem.
45. Rahimi, A., Azimi, G., Asgari, H., & Jin, X. (2019). Clustering approach toward large truck crash analysis. *Transportation research record*, 2673(8), 73-85.
46. Sheffi, Y. *Urban transportation networks: equilibrium analysis and mathematical programming methods*. Prentice Hall, Englewood Cliffs, N.J., 1985
47. Toobaie, S., Aashtiani H. Z., M. Hamedi, and A. Haghani. Application of a Complementarity Approach to Large Scale Traffic Equilibrium Problems. *TRB Annual Meeting 2010*, Washington DC
48. Vanderstraeten-Tilquin G., *Problemes de circulation avec tous convexes*. Publication 268, Departement d'biformatique et de recherché operationelle, Universite de Montreal, Montreal. 1977. 133
49. Wardrop, J. G. Some Theoretical Aspects of Road Traffic Research. *Proc., Institution of Civil Engineers*, Part II, 1, pp. 325–378, 1952 Washington, DC, 1998
50. Weintraub, A., C. Ortiz, and J. Gonzales. Accelerating Convergence of the Frank-Wolfe Algorithm. *Transportation Research*, Vol. 19B, 1985, pp.113-122
51. X. Jin, H. Asgari, M. Lee, A. Rahimi, G. Azimi. Large Truck Crash Analysis for Freight Mobility and Safety Enhancement in Florida. Florida Department of Transportation (2019). <https://fdotwww.blob.core.windows.net/sitefinity/docs/default-source/research/reports/fdot-bdv29-977-31-rpt.pdf> [Google Scholar][TRID]
52. Yang H. and Yagar S., Traffic assignment and traffic control in general freeway arterial corridor system, *Transportation Research* 28B, pp.463-486, 1987.