مجله تحقیق در عملیات در کاربردهای آن، سال چهاردهم، شماره چهارم (پیاپی ۵۵)، زمستان ۹۶، ص ص ۸۷-۶۷
شاپا ۷۲۸۶-۲۲۵۱

الگوریتم مستطیل آبشاری و ماتریس انتقال در شبکه‌های کوتاه‌ترین مسیر با دور

اصغر عینی*[1]، کورش عشقی[2]
۱ - دانشجوی دکتری، دانشگاه صنعتی شریف، دانشکده مهندسی صنایع، تهران، ایران
۲- استاد، دانشگاه صنعتی شریف، دانشکده مهندسی صنایع، تهران، ایران

رسید مقاله: ۱۸ خرداد ۱۳۹۶
پذیرش مقاله: ۱۹ آبان ۱۳۹۶

چکیده
مساله کوتاه‌ترین مسیر یکی از مسایل مشهور، بنیادی و پرطرفدار در نظریه گراف و شبکه است. در ادبیات این مساله، الگوریتم‌های کارای زیادی برای تعیین کوتاه‌ترین مسیر و مسافت بین هرزوج گره بر پایه جبر ماتریسی وجود دارد. در این مقاله، یک الگوریتم دقیق جدید با نام الگوریتم مستطیل آبشاری به‌کمک ساختار اصلی الگوریتم‌های قبلی و با بهبود روش‌های جدید، ارایه شده است. در الگوریتم ارایه شده سعی می‌شود تمام محاسبات و عملیات ریاضی الگوریتم در قالب تعدادی مستطیل پیاده‌سازی شود. الگوریتم مستطیل آبشاری، یک الگوریتم کاراست به‌طوری که روش اجرای ساده و زمان اجرای سریع دارد. بعلاوه، یک ماتریس جدید براساس ماتریس مسیر، با نام ماتریس انتقال تعریف شده که در تحلیل حساسیت و بهینه‌سازی مجدد شبکه‌های کوتاه‌ترین مسیر کاربرد دارد. در پایان، برای نشان دادن جزییات اجرای الگوریتم‌های مستطیل آبشاری، فلوید- وارشال، ماتریس تجدید نظرشده هو و ماتریس انتقال، یک مثال به‌صورت گام به‌گام حل شده است.

کلمات کلیدی: الگوریتم‌های شبکه کوتاه‌ترین مسیر با دور، الگوریتم مستطیل آبشاری، روش ماتریس تجدید نظر شده، الگوریتم فلوید - وارشال، ماتریس انتقال.

۱ مقدمه و مرور ادبیات

۱-۱ مقدمه
شبکه‌های کوتاه‌ترین مسیر از بنیادی‌ترین شبکه‌ها در جریان‌های شبکه بوده و کاربردهای فراوانی در حوزه‌های مختلف دارد. برای حل اینگونه شبکه‌ها، روش‌ها و الگوریتم‌های زیادی با رویکردهای متفاوت وجود دارد از

* عهده‌دار مکاتبات
آدرس الکترونیکی: ainiasghar@ie.sharif.edu

[Downloaded from jorlu.ir on 2026-11-29]
[DOR: 20.1001.1.22517286.2017.14.4.5.6]

جمله می‌توان به الگوریتم‌های جذاب فلوید – وارشال [۱و۲]، آبشاری [۳] ، ماتریس تجدید نظر شده [۴]، مستطیلی [۵]، دایجسترا [۶] و غیره اشاره کرد.

مساله کوتاه‌ترین مسیر از طرفی ساده‌ترین و از طرفی دیگر یکی از بنیادی‌ترین مسایل جریان‌های شبکه است [۷]. این مساله از رده‌ی مسایل P بوده، اما برای بعضی از نسخ آن از قبیل مساله کوتاه‌ترین مسیر بامحدودیت و مساله کوتاه‌ترین مسیر با سیکل منفی NP-hard بودن آن‌ها ثابت شده است [۸-۱۰]. این مساله مورد علاقه پژوهشگران و مهندسان حوزه‌های مختلف است و در موارد کاربردی متعددی نظیر شبکه‌های حمل و نقل، شبکه‌های ارتباطی و مخابراتی، شبکه‌های لجستیک و پشتیبانی استفاده می‌شود. روش‌ها و الگوریتم‌های آسان و کارا برای حل مساله کوتاه‌ترین مسیر وجود دارد که جزو ساده‌ترین مدل‌های شبکه محسوب شده که در تبدیل شبکه‌های بزرگ و پیچیده به شبکه‌های ساده‌تر و در حل بسیاری از مسایل بهینه‌سازی ترکیبی نیز مورد استفاده قرار می‌گیرد. مطالعه مساله کوتاه‌ترین مسیر نقطه عطف شروع برای تولید مفاهیم مهم و اساسی جریان‌های شبکه شامل ساختار داده هوشمند و توسعه کارائی الگوریتم‌هاست [۷].

۱-۲ مرور ادبیات

پژوهشگران مسایل کوتاه‌ترین مسیر را به‌صورت زیر تقسیم‌بندی نموده‌اند:

۱- مساله کوتاه‌ترین مسیرو مسافت از یک گره به هرگره[1]

۲- مساله کوتاه‌ترین مسیرو مسافت از چند زوج گره[2]

۳- مساله کوتاه‌ترین مسیرو مسافت از هرزوج گره[3]

۴- مسایل کوتاه‌ترین مسیر تعمیم یافته[4]

طی چند دهه اخیر الگوریتم‌های زیادی برای مساله کوتاه‌ترین مسیر توسعه داده شده است. عموما این الگوریتم‌ها به سه دسته زیر تقسیم می‌شوند [۱۱]:

۱- الگوریتم‌هایی براساس پیمایش شبکه[5]

۲- الگوریتم‌هایی براساس برنامه‌ریزی خطی[6]

۳- الگوریتم‌هایی برپایه جبر ماتریسی[7]

ابتدا در سال ۱۹۶۲ الگوریتم جذاب فلوید – وارشال ارایه شد. وارشال [۱] مبانی نظری الگوریتم را ارایه کرد و فلوید [۲] آن‌را پیاده‌سازی نمود. سپس الگوریتم معروف آبشاری معرفی شد [۳]. آنگاه هو [۴] روش ماتریس تجدیدنظرشده را با بهبود در الگوریتم آبشاری ارایه کرد. چند سال بعد، با ارایه ساختار جبری جدید به مدل‌سازی و حل مسایل شبکه‌های جریان از قبیل شبکه‌های کوتاه‌ترین مسیر پرداخته شد [۱۲].

[1] Single Source Shortest Path Problem – SSSP Problem
[2] Multiple Pairs Shortest Path Problem - MPSP Problem
[3] All Pairs Shortest Path Problem – APSP Problem
[4] Generalized Shortest Path Problem
[5] Network Traversal Algorithms
[6] LP – Based Algorithms
[7] Matrix Algebra Algorithms

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

همچنین با استفاده از تکنیک‌های ماتریس متراکم، دو روش تولید کد و تولید آدرس- جدول برای اجرای بهتر برنامه‌های کامپیوتری جهت حل مساله کوتاه‌ترین مسیر هر زوج گره ارایه شد [۱۳]. تاکائوکا [۱۴] به‌کمک ضرب ماتریس مسافت تحت مدل RAM الگوریتم سریع‌تری برای این مساله معرفی نمود. سپس دوئین [۱۵] دو الگوریتم جدید بر اساس بهبود معادلات بلمن در شبکه‌های بزرگ و متراکم ارایه کرد. در همین سال، دمترسو و ایتالیانو [۱۶] الگوریتم جدیدی برای شبکه‌های کوتاه‌ترین مسیر پویا براساس خواص جبری و ساختار داده ارایه کردند. چان [۱۷] یک الگوریتم جدید و ساده با بهبود الگوریتم‌های زیرمکعبی از قبیل فریدمن ارایه نمود. موحمد و همکاران [۱۸]، به‌کمک الگوریتم فراابتکاری بهینه‌سازی توده ذرات[1] مساله کوتاه‌ترین مسیر را حل کردند. دو سال بعد، هوگاردی [۸] با بهبود الگوریتم جذاب فلوید- وارشال توانست سیکل منفی را در شبکه‌های کوتاه‌ترین مسیر بادور و باوزن منفی شناسائی کند. در همین‌سال، موحامد و همکاران [۱۹]، به‌کمک الگوریتم فراابتکاری ژنتیک[2] مساله کوتاه‌ترین مسیر دوهدفه را حل کردند. در اینگونه مسایل علاوه بر وزن مسافت، وزن زمان هم درنظرگرفته می‌شود. یاستر [۲۰] نیز یک الگوریتم تقریبی برای مساله کوتاه‌ترین مسیر با وزن‌های حقیقی ارایه کرد. پنگ و همکاران[۲۱] با بهبود الگوریتم دایجسترا به‌کمک دو رویکرد بهینه‌سازی آن‌را در حل شبکه‌های بزرگ کوتاه‌ترین مسیر نظیر شبکه‌های اینترنت، الکترونیک، حمل و نقل و اجتماعی استفاده کردند. همچنین دوئورو همکاران [۲۲] دو الگوریتم مختلف برای این مساله ارایه کردند. اولین الگوریتم با رویکرد مکانیزم‌های تعمیر و دومین الگوریتم بر اساس انتخاب والد است. در همین سال، هاگت و همکاران [۲۳]، یک الگوریتم کارا برای شبکه‌های کوتاه‌ترین مسیر چند ماژولی[3] و وانگ و کمبکه [۲۴]، الگوریتمی جدید با زمان چندجمله‌ای برای محاسبه کوتاه‌ترین مسیر در سیستم‌های توزیع‌شده ارایه کردند. آمروسینی و سوماچن [۲۵]، یک الگوریتم جدید برای شبکه‌های کوتاه‌ترین مسیر چند ماژول شهری معرفی کردند.

۱-۳ مزایای الگوریتم مستطیل آبشاری

اساس الگوریتم مستطیل آبشاری برگرفته از الگوریتم‌های ماتریس تجدیدنظر شده هو و آبشاری فاربی و همکاران است. صحت اجرای الگوریتم‌های هو و آبشاری فاربی و همکاران برای محاسبه ماتریس‌های مسافت و مسیر قبلا اثبات شده است [۳ و۴]. در نگارش الگوریتم مستطیل آبشاری، از ساختار گام‌ها، به‌ویژه نمایش ماتریس‌های مسافت و مسیر الگوریتم‌های جذاب فلوید- وارشال و مستطیلی استفاده شده است [۱، ۲ و ۵]. به عبارت‌بهتر، الگوریتم مستطیل آبشاری ترکیبی از الگوریتم‌های هو، فاربی و همکاران، الگوریتم جذاب فلوید- وارشال و الگوریتم مستطیلی است که با تغییرات زیر پیاده‌سازی شده است.

- الگوریتمی کردن روش انجام کار در سه گام و زیرگام در قالب مراحل B و F و با ساختار ماتریسی (بهبود روش ماتریس تجدیدنظرشده هو از نظر ساختار الگوریتم و نمایش ماتریسی بجای جدولی داده‌های مسیر و مسافت).

[1] Particle Swarm Optimization (PSO)
[2] Genetic Algorithm
[3] Multi Modal

[Downloaded from jorlu.ir on 2026-11-29]
[DOR: 20.1001.1.22517286.2017.14.4.5.6]

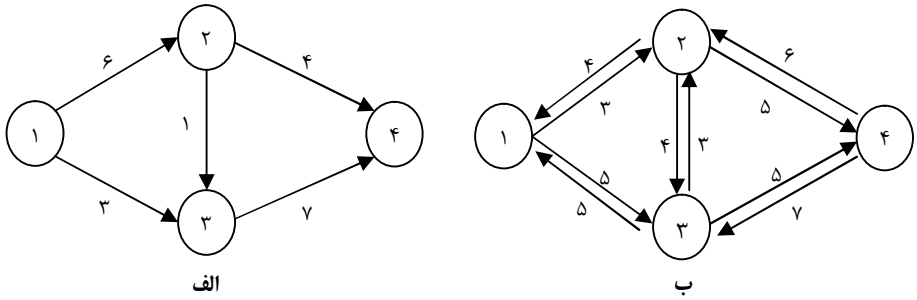
- وابسته کردن کلیه محاسبات عناصر ماتریس مسیر R به عناصر ماتریس مسافت D (بهبود کامل در الگوریتم‌های هو، فاربی و فلوید – وارشال با حذف محاسبات ماتریس مسیر R) این موضوع در بند دوم گام دوم و بند د گام سوم الگوریتم مستطیل آبشاری آمده است. این بهبود با کاهش تعداد محاسبات و درنتیجه افزایش سرعت اجرای الگوریتم همراه خواهد بود.
- انجام کلیه محاسبات و مقایسات ریاضی موردنیاز الگوریتم مستطیل آبشاری در قالب $(n-2)$ مستطیل (این مستطیل‌ها که معرف محاسبات الگوریتم هستند به‌صورت ذهنی قابل رسم بوده و به‌سادگی محاسبه می‌شوند).
- کاهش حجم محاسبات از n محاسبه و مقایسه درالگوریتم‌های هو و فاربی به $(n-2)$محاسبه و مقایسه در الگوریتم مستطیل آبشاری برای محاسبه هر عنصر ماتریس مسافت D. به‌عبارت‌بهتر، در الگوریتم هو و فاربی برای محاسبه هر عنصرماتریس مسافت D، به تعداد $2n(n^2-n)$ محاسبه و مقایسه نیاز است به‌نحوی که در مستطیل آبشاری، این تعداد به $2(n-2)(n^2-n)$ کاهش می‌یابد. در الگوریتم‌های هو، فاربی و فلوید- وارشال برای محاسبه ماتریس مسیر به تعداد محاسبات ماتریس مسافت موردنیاز است بطوری‌که، در الگوریتم مستطیل آبشاری این محاسبات حذف شده ومحاسبات عناصرماتریس مسیر کاملا به نتیجه محاسبات عناصر ماتریس مسافت وابسته شده است.
- مقایسه ذهنی عنصر d_{ik} باعناصر d_{ij} و d_{jk}. به عبارت‌بهتر، اگر $d_{ik} \le d_{ij}$ یا $d_{ik} \le d_{jk}$ باشد، آنگاه $d_{ik} \le d_{ij} + d_{jk}$ خواهد بود. در نتیجه، این محاسبه و مقایسه قابل حذف خواهد بود. این مقایسه ذهنی باعث کاهش حجم محاسبات و در نتیجه افزایش سرعت الگوریتم مستطیل آبشاری خواهد شد.
- روش تصحیح مسیر در الگوریتم‌های جبر ماتریسی به‌ویژه الگوریتم فلوید- وارشال. الگوریتم‌های جبر ماتریسی از قبیل الگوریتم جذاب فلوید- وارشال در محاسبه صحیح عناصر ماتریس مسیر در برخی از شبکه‌ها با ساختارهای خاص دچار محدودیت محاسباتی می‌شود به‌نحوی که این موضوع، با ارایه روشی در مثال اول ۵-۲ برطرف شده است.
- رویکرد انتخاب صحیح مسیر در الگوریتم مستطیل آبشاری. با روش ارایه شده در آخر پاراگراف بند د از گام سوم، عناصر ماتریس مسافت در الگوریتم مستطیل آبشاری بدرستی محاسبه می‌شوند.
- ارایه ماتریس انتقال براساس ماتریس مسیر. ماتریس انتقال از مباحث جدید بوده که توسط نویسندگان ارایه شده به‌نحوی که در تحلیل حساسیت و بهینه‌سازی مجدد شبکه‌های کوتاه‌ترین مسیر کاربرد اساسی دارد. این ماتریس در هر رویکرد جبر ماتریسی که در آن ماتریس مسیر تولید می‌شود، قابل پیاده‌سازی است.

۲ شبکه‌های کوتاه‌ترین مسیر با دور و بدون دور

شبکه‌های کوتاه‌ترین مسیر از نظر ساختار شبکه‌ای به دو دسته‌ی شبکه‌های کوتاه‌ترین مسیرجهت‌دار بدون دور و شبکه‌های کوتاه‌ترین مسیر جهت‌دار با دور تقسیم می‌شوند. در شکل ۱، دو شبکه ساده نمونه از شبکه‌های کوتاه‌ترین مسیر آورده شده که درآن شکل ۱- الف ساختار یک شبکه کوتاه‌ترین مسیر جهت‌دار بدون دور و شکل ۱-ب ساختار یک شبکه کوتاه‌ترین مسیر جهت‌دار با دور را نشان می‌دهد.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]



شکل ۱. دو شبکه کوتاه‌ترین مسیر جهت‌دار ساده، الف. بدون دور و ب. با دور

شبکه‌های کوتاه‌ترین مسیر جهت‌دار با دور، نوع جامع و کامل‌تری از شبکه‌های کوتاه‌ترین مسیرهستند. در این شبکه‌ها حداقل یک دور وجود دارد. این نوع شبکه‌ها فاقد گره شروع و پایان مشخصی هستند، لذا هر گره می‌تواند معرف گره شروع یا ختم باشد[۵]. الگوریتم مستطیل آبشاری ارایه‌شده در این مقاله برای حل شبکه‌های کوتاه‌ترین مسیر با دور برای هرزوج گره است.

۳ الگوریتم مستطیل آبشاری

شبکه جهت‌دار با دور $G=(N,A)$ با مجموعه گره‌های $N=\{1,2,\ldots,n\}$، ($|N|=n$) و مجموعه کمان‌های $A=\{(i,k)\,|\,i,k\in N, i\neq k\}$ و با وزن (مسافت) مثبت d_{ik} برای هر کمان $(i,k)\in A$را در نظر بگیرید. هدف پیداکردن کوتاه‌ترین مسافت و مسیر بین هر دو گره دلخواه i و kاست. برای این منظور از دو ماتریس مسافت D_s و مسیر R_s با ابعاد $n\times n$ استفاده می‌شود که طی دو مرحله $s=B,F$به جواب بهینه می‌رسند. مرحله B (مرحله پایه)[1] و مرحله F (مرحله نهایی[2]) نامیده می‌شود. به‌عبارت‌بهتر، در شروع اجرای الگوریتم، دو ماتریس D_B و R_B به‌ترتیب مقادیر اولیه مسافت و مسیر را که برگرفته از داده‌های شبکه است در خود ذخیره کرده و در پایان اجرای الگوریتم، دو ماتریس D_F و R_F به‌ترتیب مقادیر کوتاه‌ترین مسافت و مسیر را در خود ذخیره می‌کنند. در این الگوریتم محاسبه ماتریس‌های R_s به ماتریس D_s وابسته هستند که این موضوع، باعث افزایش سرعت و کاهش زمان اجرای الگوریتم مستطیل آبشاری می‌شود. دلیل نام‌گذاری الگوریتم به مستطیل آبشاری این است که تمام محاسبات و عملیات ریاضی الگوریتم در قالب $n-2$ مستطیل پیاده‌سازی می‌گردد که دارای راس مشترک d_{ik} هستند. به‌عبارت‌بهتر، در الگوریتم مستطیل آبشاری برای محاسبه هر عنصر d_{ik}، از $n-2$ مقایسه و در قالب مستطیل که معرف روابط ریاضی هستند، استفاده شده درصورتی‌که در الگوریتم‌های فاربی یا هو از n مقایسه در قالب روابط ریاضی برای محاسبه هر عنصر d_{ik} استفاده می‌شود درنتیجه، در الگوریتم پیشنهادی علاوه بر رویکرد مستطیلی که انجام محاسبات را ساده‌تر نموده تعداد مقایسات برای محاسبه هر عنصر نیز کاهش یافته است که این مزیت، باعث افزایش سرعت و کاهش زمان اجرای الگوریتم شده است. چون اساس الگوریتم مستطیل آبشاری برگرفته از الگوریتم‌های اثبات شده آبشاری فاربی به‌ویژه ماتریس تجدید نظر شده هو است نیازی به

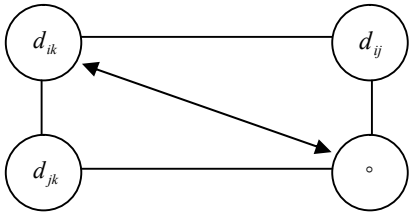
[1] Basic Stage
[2] Final Stage

[Downloaded from jorlu.ir on 2026-11-29]
[DOR: 20.1001.1.22517286.2017.14.4.5.6]

اثبات صحت الگوریتم نخواهد بود. جهت مطالعه بیشتر به‌مراجع [۳ و۴] رجوع گردد.

۳-۱ روش تشکیل مستطیل

در مرحله F، برای محاسبه هر عنصر d_{ik} (عنصر سطر i و ستون k)، $n-2$ مستطیل با استفاده از صفرهای قطر اصلی طوری رسم می‌شوند که عنصر d_{ik} روبروی صفرهای قطر اصلی قرار گیرد. به‌عبارت‌بهتر، یک مستطیل از چهار راس تشکیل می‌شود به‌نحوی که یک راس آن را عنصر d_{ik}، راس مقابل را صفر قطر اصلی و دو راس دیگر را عناصر d_{ij} و d_{jk} تشکیل می‌دهند. براساس مکان هر عنصر d_{ik} جایگاه صفرهای قطر اصلی می‌تواند متغیر باشد، اما همواره صفرها باید روبروی عنصر d_{ik} باشد. یکی از انواع این مستطیل‌ها در شکل ۲ نشان داده شده است.



شکل ۲. نحوه ایجاد مستطیل در الگوریتم مستطیل آبشاری. در این شکل، صفر همان صفرهای قطر اصلی ماتریس‌های D_B و D_F هستند.

به‌عبارت‌بهتر، به‌کمک این مستطیل‌ها، مقدار هر عنصر d_{ik} با $i \neq k$ ماتریس D_F به‌ترتیب رابطه (۱) محاسبه خواهد شد.

$$d_{ik} = Min\left(d_{ik}, d_{ij} + d_{jk}\right), \forall j \ , j \neq i, j \neq k \qquad (۱)$$

در رابطه (۱)، با تغییر مقدار هر عنصر d_{ik}، مقدار جدید جایگزین مقدار قبلی در ماتریس‌های مسافت D_B و D_F شده، سپس محاسبات رفت یا برگشت ادامه می‌یابد.

یکی از روش‌های کاهش حجم محاسبات و در نتیجه افزایش سرعت الگوریتم مستطیل آبشاری مقایسه ذهنی عنصر d_{ik} با عناصر d_{ij} و d_{jk} است. به‌عبارت‌بهتر، اگر $d_{ik} \leq d_{ij}$ یا $d_{ik} \leq d_{jk}$ باشد، آنگاه $d_{ik} \leq d_{ij} + d_{jk}$ خواهد بود. در نتیجه، این محاسبه یا مقایسه قابل حذف خواهد بود. عناصر d_{ik}، d_{ij} و d_{jk} رئوس مستطیل در شکل ۲ هستند.

۳-۲ گام‌های الگوریتم

گام اول: اگر n برابر تعداد گره‌های شبکه باشد، دو ماتریس مربع $n \times n$، D_s و R_s به‌ازای $s = B, F$ که در آن s را مرحله نامند، تعریف کنید.

گام دوم: در مرحله $s = B$ عناصر ماتریس‌های مسافت و مسیر D_B و R_B را به‌ترتیب زیر محاسبه کنید:

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

محاسبه عناصر ماتریس D_B

$$d_{ik} = \begin{cases} d_{ik} & \text{اگر کمان مستقیم از گره } i \text{ به گره } k \text{ وجود داشته باشد} \\ \infty & \text{اگر کمان مستقیم از گره } i \text{ به گره } k \text{ وجود نداشته باشد} \\ ۰ & \text{اگر } i = k \end{cases}$$

محاسبه عناصر ماتریس R_B

$$r_{ik} = \begin{cases} k & \text{ماتریس در آن متناظر عنصر اگر } D_B \text{، } ۰ \text{ یا } \infty \text{ نباشد} \\ - & \text{در غیر این‌صورت} \end{cases}$$

گام سوم: در مرحله $s = F$ ماتریس‌های D_F و R_F را براساس الگوی زیر پیاده‌سازی کنید.

الف. عناصر قطر اصلی ماتریس‌های D_F و R_F را از ماتریس‌های D_B و R_B تکرار کنید.

ب. برای محاسبه سایر عناصر ماتریس D_F و به‌طور مشخص برای هر عنصر d_{ik} با $i \neq k$ تعداد $n-۲$ مستطیل در ماتریس D_B رسم کنید. این مستطیل‌ها طوری رسم می‌شوند که عنصر d_{ik} روبروی صفرهای قطر اصلی باشد. چون برای محاسبه عنصر d_{ik} از صفرهای موجود در سطر i و ستون k نمی‌توان استفاده نمود یا به‌عبارت‌بهتر، با اینگونه صفرها نمی‌توان مستطیل ساخت، به‌همین‌دلیل تعداد مستطیل‌ها برابر $n-۲$ خواهد بود. این محاسبات که از عنصر $d_{۱۲}$ شروع شده و به عنصر $d_{n,n-۱}$ ادامه می‌یابد را محاسبات رفت بنامید. به‌محض تغییر عنصر d_{ik}، آن را در ماتریس مسافت جایگزین کرده، سپس محاسبات را ادامه دهید.

ج. محاسبات برگشت (همانند محاسبات رفت) را از عنصر $d_{n,n-۱}$ ماتریس D_F شروع کرده تا به عنصر $d_{۱۲}$ ادامه دهید.

د. بر خلاف الگوریتم‌های هو، فاربی و فلوید- وارشال، برای محاسبه عناصر ماتریس R_F، محاسبات جدیدی انجام نمی‌شود، بلکه محاسبات عناصر ماتریس‌های مسیر R_B و R_F به‌ترتیب به محاسبات عناصر ماتریس‌های D_B و D_F وابسته شده است. به‌عبارت‌بهتر، اگر عنصری از ماتریس D_F تغییر نکند در آن صورت، عنصر متناظر آن در R_F نیز تغییر نخواهد کرد. درغیراین صورت، عنصر متناظر در ماتریس R_F نیز تغییر خواهد یافت و این تغییرات بستگی به مختصات صفرهای قطر اصلی در مستطیل‌های مربوطه خواهد داشت. در صورت برابری مقادیر مستطیل‌ها، مقدار r_{ik} را از سطر i و ستون‌های مرتبط با عناصر قطر اصلی مستطیل‌های برابر انتخاب کنید.

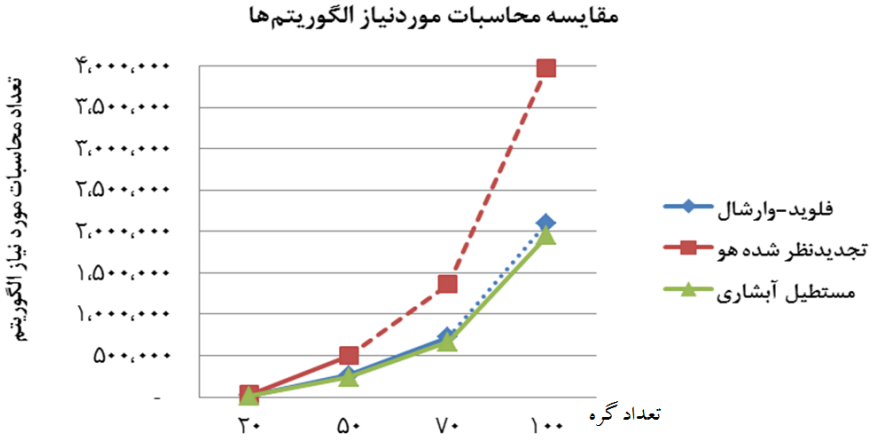
دلیل وابسته‌کردن محاسبات ماتریس مسیر به محاسبات ماتریس مسافت، تشابه محاسباتی ماتریس‌های مسیر و مسافت در الگوریتم‌های هو، فاربی و فلوید- وارشال است. به عبارت‌بهتر، در این الگوریتم‌ها، از دو بیان برای یک منظور استفاده شده است. در نتیجه، صحت اجرای الگوریتم مستطیل آبشاری همانند الگوریتم‌های اثبات شده هو، فاربی و فلوید- وارشال خواهد بود.

قضیه: پیچیدگی زمانی بدترین حالت الگوریتم مستطیل آبشاری، $O\left(n^{۳}\right)$ است.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

اثبات: برای محاسبه عناصر ماتریس D_B، تعداد محاسبات برابر n^2-n و برای محاسبه عناصر ماتریس D_F تعداد محاسبات رفت و برگشت یکسان خواهد بود [۳ و۴]. به‌عبارت‌بهتر، برای اجرای الگوریتم مستطیل آبشاری به تعداد $2(n^2-n)$ محاسبه با $(n-2)$ مستطیل برای هر عنصر d_{ik} ماتریس D_F نیاز است به‌طوری‌که محاسبات رفت در ماتریس D_B و محاسبات برگشت در ماتریس D_F انجام می‌شود؛ بنابراین کل محاسبات موردنیاز، برای محاسبه عناصر ماتریس‌های مسافت D_B و D_F برابر $2(n^2-n)(n-2)+(n^2-n)$ خواهد بود. به‌جهت وابسته کردن ماتریس‌های مسیر به ماتریس‌های مسافت، برای تعیین عناصر ماتریس‌های مسیر R_B و R_F محاسبات جدیدی نیاز نخواهد بود. درنتیجه، تعداد کل محاسبات مورد نیاز الگوریتم مستطیل آبشاری برابر $2(n^2-n)(n-2)+(n^2-n)$ خواهد شد. به‌عبارت‌بهتر، می‌توان نتیجه گرفت که پیچیدگی زمانی الگوریتم مستطیل آبشاری در بدترین حالت، همانند الگوریتم‌های فلوید- وارشال و ماتریس تجدیدنظرشده هو $O(n^3)$ خواهد بود. قابل ذکر است که در اکثر شبکه‌ها که دارای ساختار خاص نیستند محاسبات رفت کفایت خواهد کرد، دراین صورت حجم محاسبات الگوریتم مستطیل آبشاری به نصف کاهش خواهد یافت. درشکل ۳، تعداد محاسبات موردنیاز سه الگوریتم برای شبکه‌ها با گره‌های مختلف نشان داده شده است.



شکل ۳. عملکرد محاسباتی الگوریتم‌های مستطیل آبشاری، فلوید-وارشال و ماتریس تجدیدنظرشده هو برای شبکه‌های مختلف

با توجه به شکل ۳، تعداد محاسبات موردنیاز الگوریتم مستطیل آبشاری کم‌تر از الگوریتم فلوید- وارشال و بسیار کم‌تر از الگوریتم ماتریس تجدیدنظرشده هو به‌ویژه در مقیاس بزرگ خواهد بود.

٤ ماتریس انتقال

پس از حل شبکه‌های کوتاه‌ترین مسیر هرزوج گره، به‌کمک الگوریتم‌های جبر ماتریسی از قبیل الگوریتم‌های فلوید – وارشال، آبشاری، ماتریس تجدیدنظر شده هو، مستطیلی [۵]، مستطیل آبشاری و غیره دو ماتریس مسافت D و مسیر R محاسبه شده به‌طوری‌که کوتاه‌ترین مسیر و مسافت هرزوج گره در آن‌ها ذخیره می‌شوند. در این قسمت

[Downloaded from jorlu.ir on 2026-11-29]
[DOR: 20.1001.1.22517286.2017.14.4.5.6]

ماتریس انتقال یا جابجائی[1] ارایه شده که عناصر آن براساس عناصر ماتریس مسیر محاسبه می‌شود. عناصر ماتریس انتقال نشان‌دهنده تعداد جابه‌جایی کمان‌های شبکه در جواب بهینه است. به‌کمک این ماتریس، میزان ارزش هر کمان در جابه‌جایی بهینه شبکه قابل محاسبه است. این ماتریس در تحلیل حساسیت و بهینه‌سازی مجدد شبکه‌های کوتاه‌ترین مسیر هر زوج گره کاربرد مهمی دارد. ماتریس انتقال با T نشان داده شده است. تعداد جابه‌جایی یا انتقال کمان را " عدد انتقال کمان[2] " نامیده و آن‌را با t_{ik} نشان می‌دهیم که همان عناصر ماتریس انتقال هستند. به عبارت‌بهتر، t_{ik} بیانگر تعداد جابه‌جایی بین گره‌های i و k در جواب بهینه است؛ یعنی با حذف کمان $i-k$، t_{ik} جابه‌جایی یا انتقال از جابه‌جایی‌های بهینه شبکه دچار اختلال خواهد شد. هرچه مقدار عددی آن بزرگ‌تر باشد اهمیت آن کمان در تعداد جابه‌جایی بهینه شبکه بیشتر خواهد بود و برعکس، مقدار عددی کوچک‌تر آن، معرف اهمیت کم‌تر آن کمان در جابه‌جایی‌های بهینه شبکه است. نقش یا ارزش هر کمان در جابه‌جایی شبکه را "ارزش انتقال کمان"[3] نام‌گذاری کرده که w_{ik} نشان‌دهنده آن است. ارزش انتقال هر کمان به صورت نسبت عدد انتقال یا جابه‌جایی هرکمان به کل جابه‌جایی در جواب بهینه شبکه تعریف شده و بر حسب درصد نشان داده می‌شود. به عبارت‌بهتر، با حذف هر کمان دلخواه $i-k$، w_{ik} درصد از جابه‌جایی‌های بهینه شبکه دچار اختلال خواهد شد. عدد و ارزش انتقال هر کمان کاربرد اساسی در تحلیل حساسیت و بهینه‌سازی مجدد شبکه‌های کوتاه‌ترین مسیردارد، زیرا به‌کمک آن‌ها می‌توان نقش کمان‌ها را در جابه‌جایی شبکه‌ها بررسی نمود و درنتیجه، کمان‌های زاید را شناسائی و حذف کرد. ماتریس انتقال T، همانند دو ماتریس مسافت D و مسیر R یک ماتریس مربع $n\times n$ است. عناصراین‌ماتریس براساس‌عناصرماتریس R محاسبه می‌شود. روش محاسبه عناصر ماتریس T به‌ترتیب رابطه (۲) است.

$$t_{ik}=\begin{cases}- & if\ \ i=k\\ \sum_{k=1,i\neq k,r_{ik}^{F}=i}^{k=n} r_{ik}^{F}+\sum_{i=1,i\neq k,r_{ik}^{F}=k}^{i=n} r_{ik}^{F} & if\ \ r_{ik}^{F}=k\\ 0 & Otherwis\end{cases} \qquad (۲)$$

اگر t معرف جمع عناصر ماتریس انتقال باشد در آن‌صورت به‌کمک رابطه (۳) ارزش انتقال هر کمان درشبکه قابل محاسبه است.

$$t=\sum_{i=1}^{i=n}\sum_{k=1,i\neq k}^{k=n} t_{ik}$$

$$w_{ik}=\frac{t_{ik}}{t}\times 100 \qquad (۳)$$

عدد و ارزش انتقال برای گره‌ها هم قابل تعریف است. به‌عبارت‌بهتر، می‌توان ارزش جابه‌جایی هر گره i در کل جابه‌جایی شبکه را محاسبه نمود. هر گره i به $n-1$ گره دیگر جابه‌جایی دارد و همه‌ی $n-1$ گره به گره i هم

[1] Transposition Matrix
[2] Arc Transposition Number
[3] Arc Transposition Worth

[Downloaded from jorlu.ir on 2026-11-29]
[DOR: 20.1001.1.22517286.2017.14.4.5.6]

جابه‌جایی دارند، بنابراین کل جابه‌جایی از گره i به سایر گره‌ها و برعکس معادل $۲(n-۱)$ خواهد بود. اگر به مقدار $۲(n-۱)$ تعداد دفعاتی که گره i واسطه گره‌ها قرار می‌گیرد اضافه شود تعداد کل جابه‌جایی گره i محاسبه خواهد شد و این به شرطی است که در ماتریس R عنصری با مقدار ∞ وجود نداشته باشد، درغیراینصورت، تعداد ∞ ها از مقدار $۲(n-۱)$ کم خواهد شد. اگر in_i و t_i به‌ترتیب نشان‌دهنده تعداد واسطه‌ها و عدد جابه‌جایی گره i باشد، مقادیر آن ها به‌کمک روابط (۴) و (۵) قابل محاسبه است. برای محاسبه تعداد واسطه‌های هرگره i، لازم است در ماتریس R_F به جز ستون i −ام، عدد i آن شمارش گردد.

$$in_i = \sum_{k=۱,k\neq i}^{k=n} \sum_{i=۱,r_{ik}^F=i,i\neq k}^{i=n} r_{ik}^F \qquad (۴)$$

$$t_i = ۲(n-۱) + in_i \, , r_{ik} \neq \infty \qquad (۵)$$

اگر it و tt به ترتیب معرف کل واسطه‌ها و کل جابه‌جایی باشند، مقدار ارزش جابه‌جایی گره i یا w_i با استفاده از رابطه (۶) محاسبه می‌شود.

$$it = \sum_{i=۱}^{i=n} in_i$$

$$tt = it + ۲n(n-۱)$$

$$w_i = \frac{t_i}{tt} \, , r_{ik} \neq \infty \times ۱۰۰ \qquad (۶)$$

به‌عبارت‌بهتر، w_i نشان‌دهنده درصد ارزش یا اهمیت گره i در کل جابجایی‌های بهینه شبکه است یعنی با حذف گره i، مقدار w_i درصد از جابه‌جایی‌های شبکه دچار اختلال خواهد شد. بزرگی عددی آن معرف نقش مهمتر گره در جابه‌جایی بهینه خواهد بود. عدد و ارزش انتقال هر گره، همانند عدد و ارزش انتقال هر کمان، کاربرد مهمی در تحلیل حساسیت و بهینه‌سازی مجدد شبکه‌های کوتاه‌ترین مسیر دارد، زیرا به‌کمک آن‌ها می‌توان نقش گره‌ها را در جابه‌جایی بهینه شبکه تعیین نمود و درنتیجه، گره‌ها با ارزش جابه‌جایی کم‌تر را شناسائی کرد. برای محاسبه تعداد واسطه‌های هر گره i، لازم است در ماتریس R_F به جز ستون i−ام، عدد i آن شمارش گردد.

۵ مثال نمونه

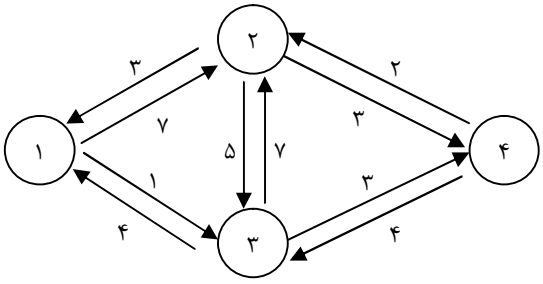
درادامه، با ارایه یک مثال، روش اجرا، حجم محاسبات و صحت اجرای سه الگوریتم مستطیل آبشاری، فلوید-وارشال و ماتریس تجدید نظر شده هو بررسی می‌شود. ابتدا، مثال به‌کمک الگوریتم مستطیل آبشاری حل می‌شود.

۵-۱ حل مثال نمونه به‌کمک الگوریتم مستطیل آبشاری

شکل ۴ را در نظر بگیرید. هدف پیداکردن کوتاه‌ترین مسیر و مسافت بین هرزوج گره به‌کمک الگوریتم مستطیل آبشاری است.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]



شکل ٤. مثال نمونه

گام اول: چون n برابر ۴ است، دو ماتریس مربع ۴×۴مسافت و مسیر، D_s و R_s به‌ازای $s = B, F$ که در آن s را مرحله نامند، تعریف کنید.

گام دوم: در مرحله $s = B$، ماتریس‌های D_B و R_B را محاسبه کنید.

$$D_B = \begin{bmatrix} ۰ & ۷ & ۱ & \infty \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix},\ R_B = \begin{bmatrix} - & ۲ & ۳ & - \\ ۱ & - & ۳ & ۴ \\ ۱ & ۲ & - & ۴ \\ - & ۲ & ۳ & - \end{bmatrix}$$

گام سوم: در مرحله $s = F$ ، بااستفاده از بند الف گام سوم، قطر اصلی ماتریس‌های D_B و R_B را تکرار کنید.

$$D_F = \begin{bmatrix} ۰ & & & \\ & ۰ & & \\ & & ۰ & \\ & & & ۰ \end{bmatrix},\ R_F = \begin{bmatrix} - & & & \\ & - & & \\ & & - & \\ & & & - \end{bmatrix}$$

گام سوم: بااستفاده از بند ب گام سوم، برای محاسبه هر عنصر d_{ik}، $i \neq k$ از ماتریس D_F، به تعداد ۲-۴ مستطیل درماتریس D_B رسم می‌شود. این مستطیل‌ها معرف محاسبات و مقایسات الگوریتم هستند. برای محاسبه عنصر $d_{۱۲}$، تعداد ۲ مستطیل به‌نحوی رسم کنید که یک راس آن عنصر $d_{۱۲}$ یا ۷ و رئوس روبروی آن صفرهای قطر اصلی یا عناصر $d_{۳۳}$ و $d_{۴۴}$ باشند دو راس دیگر بسادگی قابل استخراج بوده که همان مقادیر عناصر d_{ij} و d_{jk} هستند. برای محاسبه هر عنصر d_{ik}، $i \neq k$، با صفرهای dii و d_{kk} نمی‌توان مستطیل ایجاد کرد، به‌همین‌دلیل تعداد مستطیل‌ها به اندازه $n-۲$ خواهد بود.

$$D_B = \begin{bmatrix} ۰ & ۷ & ۱ & \infty \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix} \mathrm{d}_{۱۲} = Min(۷, ۷+۱, ۲+\infty) = ۷ \rightarrow r_{۱۲} = r_{۱۲}$$

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

با مقایسه عنصر d_{ik} یا $d_{۱۲}=۷$ با عنصر d_{ik} یا $d_{۳۲}=۷$ از مستطیل رسم شده کوچک، چون $d_{۱۲}\le d_{۳۲}$ است و با مقایسه عنصر d_{ik} یا $d_{۱۲}=۷$ با عناصر d_{ij} یا $d_{۱۴}=\infty$ از مستطیل رسم شده بزرگ، چون $d_{۱۲}\le d_{۱۴}$ است در نتیجه، مقدار عنصر d_{ik} یا $d_{۱۲}$ نیاز به محاسبه نداشته و ثابت خواهد ماند. به‌جهت وابسته‌بودن عناصر ماتریس مسیر R به عناصرماتریس مسافت D، چون مقدار عنصر $d_{۱۲}$ تغییر نکرد، آنگاه مقدار عنصر $r_{۱۲}$ هم تغییر نخواهد کرد. در اجرای عملی الگوریتم، برای محاسبه هر عنصر دلخواه d_{ik} با $i\neq k$، رسم مستطیل‌ها و در نتیجه، انجام محاسبات و مقایسات به‌صورت ساده انجام شده و نیازی به رسم مستطیل به‌صورت فیزیکی و محاسبات مربوطه نخواهد بود.

به‌طورمشابه می‌توان سایر عناصر ماتریس مسافت را محاسبه نمود. در ادامه چند عنصر به‌صورت تصادفی انتخاب و محاسبه شده است.

$$D_B=\begin{bmatrix} ۰ & ۷ & ۱ & \infty \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix} \mathrm{d}_{۱۴}=Min\left(\infty,۷+۳,۱+۳\right)=۴$$

چون مقدار عنصر $d_{۱۴}$ تغییرکرد و این تغییر مربوط به مستطیل پررنگ، یعنی ۴ و وابسته با صفر عنصر $d_{۳۳}$ است به‌همین‌دلیل مقدار عنصر $r_{۱۴}=۳$ خواهد شد. در ادامه محاسبات رفت، مقدار عنصر $d_{۱۴}$ را با ۴ جایگزین کنید.

$$D_B=\begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix} \mathrm{d}_{۳۲}=Min\left(۷,۲+۳,۴+۷\right)=۵$$

چون مقدار عنصر $d_{۳۲}$، تغییرکرد و این تغییر مربوط به مستطیل پررنگ، یعنی ۵ و وابسته با صفر عنصر $d_{۴۴}$ است به‌همین دلیل مقدار عنصر $r_{۳۲}=۴$ خواهد شد. در ادامه محاسبات رفت، مقدار عنصر $d_{۳۲}$ را با ۵ جایگزین کنید.

$$D_B=\begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix} \mathrm{d}_{۴۱}=Min\left(\infty,۳+۲,۴+۴\right)=۵\rightarrow r_{۴۱}=۲$$

با ادامه‌ی اجرای الگوریتم برای محاسبات رفت به‌سادگی می‌توان سایر عناصر ماتریس D_F را محاسبه کرد و به طور مشابه ماتریس R_F نیز محاسبه خواهد شد.

$$D_F=\begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix}, R_F=\begin{bmatrix} - & ۲ & ۳ & ۳ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۴ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix}$$

برای انجام محاسبات برگشت، از عنصر $d_{۴۳}$ ماتریس D_F شروع کرده و تعداد ۲-۴ مستطیل برای محاسبه هر عنصر d_{ik} رسم کنید. این محاسبات همانند محاسبات رفت است.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

$$D_F = \begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix} \quad \mathrm{d}_{۴۳} = Min\left(۴, ۲+۴, ۵+۱\right) = ۴ \rightarrow r_{۴۳} = r_{۴۳}$$

محاسبات برگشت را تا عنصر $d_{۱۲}$ ادامه دهید.

$$D_F = \begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix} \quad \mathrm{d}_{۱۲} = Min\left(۷, ۵+۱, ۲+۴\right) = ۶$$

چون مقدار عنصر $d_{۱۲}$، تغییر کرد در نتیجه مقدار عنصر $r_{۱۲}$ هم تغییر خواهد کرد. این تغییر هم مربوط به مستطیل اول، یعنی ۶ و متناسب با صفر $d_{۳۳}$ بوده و همینطور این تغییرمی‌تواند مربوط به مستطیل دوم با مقدار ۶ و متناسب با صفر $d_{۴۴}$ باشد. معیار انتخاب مقدار عنصر $r_{۱۲}$، عناصر $r_{۱۳}$ و $r_{۱۴}$ ماتریس R_F است چون مقادیر این عناصر برابر ۳ است؛ لذا مقدار عنصر $r_{۱۲} = ۳$ خواهد بود. در ادامه محاسبات برگشت، مقدار عنصر $d_{۱۲}$ را با ۶ جایگزین کنید.

در پایان اجرای الگوریتم، ماتریس‌های مسیر و مسافت نهایی به‌ترتیب زیرمحاسبه خواهند شد:

$$D_F = \begin{bmatrix} ۰ & ۶ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix}, R_F = \begin{bmatrix} - & ۳ & ۳ & ۳ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۴ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix}$$

با وجود ماتریس‌های D_F و R_F می‌توان کوتاه‌ترین مسافت و مسیر را از هر گره i تا هر گره i را به دست آورد. به‌عنوان‌مثال، کوتاه‌ترین مسافت بین دو گره ۱ و ۲ عبارت است از ۶ یا ($d_{۱۲} = ۶$) و کوتاه‌ترین مسیر به ترتیب زیر قابل محاسبه است:

$$r_{۱۲} = ۳, r_{۳۲} = ۴, r_{۴۲} = ۲ \rightarrow ۱-۳-۴-۲$$

در این قسمت، مثال نمونه را به کمک الگوریتم فلوید – وارشال حل کرده تا روش اجرای این الگوریتم، حجم محاسبات و صحت اجرای آن بررسی گردد. در این الگوریتم، برای محاسبه عناصر ماتریس مسیر، محاسبات مشابه ماتریس مسافت انجام می‌شود.

۵-۲ حل مثال نمونه به‌کمک الگوریتم فلوید - وارشال

شکل ۴ را مجددا در نظر بگیرید. هدف پیداکردن کوتاه‌ترین مسیر و مسافت بین هرزوج گره در شبکه به‌کمک الگوریتم فلوید- وارشال است [۱و۲].

ابتدا ماتریس‌های $D_۰$ و $R_۰$ را محاسبه کنید. با پیاده‌سازی گام ۲ الگوریتم فلوید- وارشال به‌راحتی می‌توان دو ماتریس $D_۰$ و $R_۰$ را به‌صورت زیر به دست آورد.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

$$D_0=\begin{bmatrix} 0 & 7 & 1 & \infty \\ 3 & 0 & 5 & 3 \\ 4 & 7 & 0 & 3 \\ \infty & 2 & 4 & 0 \end{bmatrix},\quad R_0=\begin{bmatrix} - & 2 & 3 & - \\ 1 & - & 3 & 4 \\ 1 & 2 & - & 4 \\ - & 2 & 3 & - \end{bmatrix}$$

برای محاسبه ماتریس‌های D_1 و R_1، باتوجه به اینکه $j=1$ است، لذا مقادیر سطر و ستون ۱ تکرار خواهد شد. قطر اصلی درکلیه مراحل ثابت خواهد ماند. سایر عناصر با پیمودن گام ۳ الگوریتم فلوید- وارشال قابل محاسبه است. در ادامه چند عنصر این ماتریس‌ها محاسبه شده‌اند.

$$d_{23}=Min\left(d_{23},d_{21}+d_{13}\right)=\text{Min}\left(5,3+1\right)=4$$
$$d_{34}=Min\left(d_{34},d_{31}+d_{14}\right)=Min\left(3,4+\infty\right)=3$$

با ادامه محاسبات، ماتریس D_1 مطابق زیر به‌دست خواهد آمد:

$$D_1=\begin{bmatrix} 0 & 7 & 1 & \infty \\ 3 & 0 & 4 & 3 \\ 4 & 7 & 0 & 3 \\ \infty & 2 & 4 & 0 \end{bmatrix}$$

با انجام محاسبات زیر، ماتریس مسیر R_1، قابل محاسبه خواهد بود:

$$d_{23}\geq d_{21}+d_{13}\equiv 5\geq 3+1\rightarrow r_{23}=j=1$$
$$d_{34}\leq d_{31}+d_{14}\equiv 3\leq 4+\infty\rightarrow r_{34}=r_{34}=4$$

با ادامه محاسبات، ماتریس R_1 مطابق زیر به‌دست خواهد آمد:

$$R_1=\begin{bmatrix} - & 2 & 3 & - \\ 1 & - & 1 & 4 \\ 1 & 2 & - & 4 \\ - & 2 & 3 & - \end{bmatrix}$$

برای محاسبه ماتریس‌های D_2 و R_2، باتوجه به اینکه $j=2$ است؛ لذا مقادیر سطر و ستون ۲ تکرار خواهد شد. با پیمودن گام ۳ الگوریتم فلوید- وارشال، محاسبات چند عنصر به‌صورت زیر خواهد بود:

$$d_{13}=Min\left(d_{13},d_{12}+d_{23}\right)=Min\left(1,7+4\right)=1$$
$$d_{34}=Min\left(d_{34},d_{32}+d_{24}\right)=Min\left(3,7+3\right)=3$$

بر اساس محاسبات انجام شده، ماتریس D_2 مطابق زیر به‌دست خواهد آمد:

$$D_2=\begin{bmatrix} 0 & 7 & 1 & 10 \\ 3 & 0 & 4 & 3 \\ 4 & 7 & 0 & 3 \\ 5 & 2 & 4 & 0 \end{bmatrix}$$

$$d_{13}\leq d_{12}+d_{23}\equiv 1\leq 7+4\rightarrow r_{13}=r_{13}=3$$
$$d_{34}\leq d_{32}+d_{24}\equiv 3\leq 7+3\rightarrow r_{34}=r_{34}=4$$

در نتیجه، ماتریس R_2 مطابق زیر به‌دست خواهد آمد:

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

$$R_۲=\begin{bmatrix} - & ۲ & ۳ & ۲ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۲ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix}$$

با ادامه اجرای الگوریتم فلوید- وارشال ماتریس‌های $D_۴$ و $R_۴$ به‌ترتیب زیر محاسبه می‌شوند:

$$D_۴=\begin{bmatrix} ۰ & ۶ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix}, R_۴=\begin{bmatrix} - & ۴ & ۳ & ۳ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۴ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix}$$

با در دست داشتن دو ماتریس $D_۴$ و $R_۴$ می توان مسافت و مسیر را از هر گره دلخواه i به هر گره دلخواه k به دست آورد. به‌عنوان‌مثال، کوتاه‌ترین مسافت بین دو گره ۱ و ۲ عبارت است از ۶ یا ($d_{۱۲}=۶$) و کوتاه‌ترین مسیر به ترتیب زیر قابل محاسبه است:

$r_{۱۲}=۴, r_{۴۲}=۲ \rightarrow ۱-۴-۲$

الگوریتم فلوید- وارشال مسیر ۲-۴-۱ را کوتاه‌ترین مسیر محاسبه کرده است درصورتی‌که مسیر صحیح، مسیر ۲-۴-۳-۱ است. برای تصحیح مسیر باید از وجود کمان مستقیم بین هردو گره i و k استفاده نمود. به‌عبارت‌بهتر، اگر $r_{ik}=k$ باشد بین دو گره i و k کمان مستقیم وجود دارد درغیراینصورت، اگر $r_{ik}=l$ باشد حداقل یک گره واسط l بین گره‌های i و k وجود دارد. مسیر استخراج شده از الگوریتم فلوید- وارشال ۲-۴-۱ بوده که از سه گره تشکیل شده است چون $r_{۱۴}=۳$ است، لذا گره ۳ واسط دو گره ۱ و ۴ خواهد بود. چون $r_{۳۴}=۴$ است بنابراین گره واسط بین دو گره ۳ و ۴ وجود ندارد، بنابراین زیرمسیر ۴-۱ به زیرمسیر ۴-۳-۱ تبدیل خواهد شد. چون $r_{۴۲}=۲$ است لذا گرهی واسط بین گره‌های ۴ و ۲ وجود ندارد. بنابراین، کوتاه‌ترین مسیر از گره ۱ به ۲ برابر مسیر ۲-۴-۳-۱ خواهد بود. درنتیجه، الگوریتم جذاب فلوید- وارشال در محاسبه ماتریس مسیر برخی از شبکه‌ها با ساختار خاص، دارای محدودیت محاسباتی است.

در ادامه، مثال نمونه را با الگوریتم ماتریس تجدیدنظرشده هو حل کرده تا روش اجرای این الگوریتم، حجم محاسبات و صحت اجرای آن بررسی شود. همانندالگوریتم فلوید –وارشال، در این الگوریتم نیز، برای محاسبه عناصر ماتریس مسیر محاسبات مشابه ماتریس مسافت انجام می‌شود.

۵-۳ حل مثال نمونه به‌کمک روش ماتریس تجدیدنظر شده هو

شکل ۴ را مجددا در نظر بگیرید. هدف پیداکردن کوتاه‌ترین مسیرو مسافت بین هر زوج گره به‌کمک روش ماتریس تجدیدنظر شده هو است[۴].

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

ابتدا ماتریس مسافت را محاسبه کنید. چون n برابر ۴ است، این ماتریس مربع و ۴×۴ خواهد بود. در الگوریتم ماتریس تجدیدنظر شده هو از جدول برای نمایش عناصر مسافت و مسیر استفاده شده است. دراینجا، برای سهولت نگارش، بجای جدول از ماتریس استفاده می‌شود.

	۱	۲	۳	۴
۱	۰	۷	۱	∞
۲	۳	۰	۵	۳
۳	۴	۷	۰	۳
۴	∞	۲	۴	۰

$$\rightarrow D = \begin{bmatrix} ۰ & ۷ & ۱ & \infty \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix}$$

محاسبات رفت را به ترتیب زیر انجام دهید.

برای انجام محاسبات رفت، جهت محاسبه هر عنصر d_{ik}^{F}، به تعداد $n = ۴$ مقایسه نیاز است. در ادامه تعدادی از عناصر ماتریس مسافت محاسبه شده‌اند.

$$d_{۱۲}^{F} = Min\left(d_{۱۱}+d_{۱۲}, d_{۲۲}+d_{۱۲}, d_{۱۳}+d_{۳۲}, d_{۱۴}+d_{۴۲}\right) = Min\left(۰+۷, ۷+۰, ۱+۷, \infty+۲\right) = ۷$$

$$d_{۱۴}^{F} = Min\left(d_{۱۱}+d_{۱۴}, d_{۱۲}+d_{۲۴}, d_{۱۳}+d_{۳۴}, d_{۱۴}+d_{۴۴}\right) = Min\left(۰+\infty, ۷+۳, ۱+۳, \infty+۰\right) = ۴$$

چون مقدار عنصر $d_{۱۴}$، تغییر کرد، لذا مقدار جدید عنصر را جایگزین مقدار قبلی کرده، سپس محاسبات رفت را ادامه دهید. به‌عبارت‌بهتر، ماتریس مسافت جدید به‌ترتیب زیرخواهد شد:

$$D = \begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix}$$

$$d_{۲۳}^{F} = Min\left(d_{۲۱}+d_{۱۳}, d_{۲۲}+d_{۲۳}, d_{۲۳}+d_{۳۳}, d_{۲۴}+d_{۴۳}\right) = Min\left(۳+۱, ۰+۵, ۵+۰, ۳+۴\right) = ۴$$

مقدار عنصر $d_{۲۳}$ را به ۴ تغییر داده و محاسبات رفت را تا عنصر $d_{۳۴}$ ادامه دهید.

$$d_{۳۴}^{F} = Min\left(d_{۳۱}+d_{۱۴}, d_{۲۴}+d_{۳۲}, d_{۳۳}+d_{۳۴}, d_{۳۴}+d_{۴۴} = Min\left(۴+۴, ۳+۵, ۰+۳, ۳+۰\right) = ۳\right)$$

در پایان محاسبات رفت، ماتریس مسافت به‌ترتیب زیر خواهد شد:

$$D^{F} = \begin{bmatrix} ۰ & ۷ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix}$$

محاسبات برگشت را از عنصر $d_{۴۳}$ ماتریس مسافت D^{F} شروع کرده و تا عنصر $d_{۱۲}$ ادامه دهید. در ادامه، محاسبه برگشت تعدادی از عناصر آمده است. برای انجام محاسبات برگشت، به‌تعداد $n = ۴$ مقایسه برای محاسبه هر عنصر d_{ik}^{F} نیاز خواهد بود.

$$d_{۴۳}^{F} = Min\left(d_{۴۱}+d_{۱۳}, d_{۴۲}+d_{۲۳}, d_{۴۳}+d_{۳۳}, d_{۴۳}+d_{۴۴}\right) = Min\left(۵+۱, ۲+۴, ۰+۴, ۴+۰\right) = ۴$$

$$d_{۴۱}^{F} = Min\left(d_{۱۱}+d_{۴۱}, d_{۴۲}+d_{۲۱}, d_{۴۳}+d_{۳۱}, d_{۴۱}+d_{۴۴}\right) = Min\left(۰+۵, ۲+۳, ۴+۴, ۵+۰\right) = ۵$$

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

$$d_{۱۲}^{F} = Min\left(d_{۱۱}+d_{۱۲}, d_{۱۲}+d_{۲۲}, d_{۱۳}+d_{۳۲}, d_{۱۴}+d_{۴۲}\right) = Min\left(۰+۷, ۷+۰, ۱+۵, ۴+۲\right) = ۶$$

در پایان محاسبات برگشت، ماتریس مسافت به‌ترتیب زیرخواهد شد.

$$D^{B} = \begin{bmatrix} ۰ & ۶ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix}$$

برای محاسبه ماتریس مسیر، چون n برابر ۴ است، ماتریس مسیر مربع ۴×۴، R را به‌ترتیب زیر محاسبه کنید.

$$R = \begin{bmatrix} - & ۲ & ۳ & - \\ ۱ & - & ۳ & ۴ \\ ۱ & ۲ & - & ۴ \\ - & ۲ & ۳ & - \end{bmatrix}$$

چون $d_{۱۲} = d_{۱۳} + d_{۳۲} \equiv ۶ = ۱+۵ \rightarrow r_{۱۲} = j = ۳$ خواهد بود. از طرفی دیگر، چون $d_{۱۲} = d_{۱۴} + d_{۴۲} \equiv ۶ = ۴+۲ \rightarrow r_{۱۲} = j = ۴$ خواهد شد. فرض کنید مقدار $r_{۱۲} = j = ۳$ در نظرگرفته شود. محاسبه تعدادی از عناصر ماتریس مسیر به‌ترتیب زیر خواهد بود.

$$d_{۱۳} = d_{۱۳} + d_{۳۳} \equiv ۱ = ۱+۰ \rightarrow r_{۱۳} = ۳$$
$$d_{۱۴} = d_{۱۳} + d_{۳۴} \equiv ۴ = ۱+۳ \rightarrow r_{۱۴} = j = ۳$$
$$d_{۴۱} = d_{۴۲} + d_{۲۱} \equiv ۵ = ۲+۳ \rightarrow r_{۴۱} = j = ۲$$

در پایان محاسبات، ماتریس مسیر R به‌ترتیب زیر به دست خواهد آمد:

$$R = \begin{bmatrix} - & ۳ & ۳ & ۳ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۴ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix}$$

با وجود ماتریس‌های D و R می‌توان کوتاه‌ترین مسافت و مسیر را از هر گره i تا هر گره k را به‌دست آورد. به‌عنوان مثال، کوتاه‌ترین مسافت بین دو گره ۱ و ۲ عبارت است از ۶ یا ($d_{۱۲} = ۶$) و کوتاه‌ترین مسیر به‌ترتیب زیر قابل محاسبه است:

$$r_{۱۲} = ۳, r_{۳۲} = ۴, r_{۴۲} = ۲ \rightarrow ۱-۳-۴-۲$$

اگر $r_{۱۲} = j = ۴$ فرض شود. الگوریتم هو، کوتاه‌ترین مسافت را به‌ترتیب زیر محاسبه خواهد کرد:

$$r_{۱۲} = ۴, r_{۴۲} = ۲ \rightarrow ۱-۴-۲$$

به‌عبارت‌بهتر، الگوریتم هو همانند الگوریتم فلوید- وارشال دارای محدویت محاسباتی خواهد بود. تصحیح مسیر همانند روش ارایه شده درالگوریتم فلوید- وارشال است.

در پایان، از مثال نمونه، برای محاسبه ماتریس انتقال و تحلیل آن استفاده می‌شود. یکی از کاربردهای این ماتریس، شناخت کمان‌های کم‌اهمیت و زاید در شبکه بهینه است.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

٤-٥ محاسبه ماتریس انتقال در مثال نمونه

شکل ۴ را مجددا در نظر بگیرید. هدف، محاسبه ماتریس انتقال این شبکه است.

فرض کنید ماتریس‌های D_B، R_B، D_F و R_F بر اساس یکی از الگوریتم‌های جبر ماتریسی جهت محاسبه ماتریس‌های مسافت و مسیر هر زوج گره از قبیل مستطیل آبشاری محاسبه شده باشند.

$$D_B=\begin{bmatrix} ۰ & ۷ & ۱ & \infty \\ ۳ & ۰ & ۵ & ۳ \\ ۴ & ۷ & ۰ & ۳ \\ \infty & ۲ & ۴ & ۰ \end{bmatrix} \rightarrow D_F=\begin{bmatrix} ۰ & ۶ & ۱ & ۴ \\ ۳ & ۰ & ۴ & ۳ \\ ۴ & ۵ & ۰ & ۳ \\ ۵ & ۲ & ۴ & ۰ \end{bmatrix}, R_B=\begin{bmatrix} - & ۲ & ۳ & - \\ ۱ & - & ۳ & ۴ \\ ۱ & ۲ & - & ۴ \\ - & ۲ & ۳ & - \end{bmatrix} \rightarrow$$

$$R_F=\begin{bmatrix} - & ۳ & ۳ & ۳ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۴ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix}$$

برای محاسبه عناصرماتریس انتقال که براساس فرمت نام‌گذاری ماتریس‌های مسافت و مسیر الگوریتم مستطیل آبشاری با T_F نشان داده شده از ماتریس R_F استفاده کنید. قطر اصلی ماتریس T_F همانند ماتریس R_F، برابر " $-$ " خواهد بود. برای محاسبه عناصر $r_{ik}=k$، یا برای محاسبه عنصر $t_{۱۲}$، در سطر ۱ ماتریس R_F، تعداد عناصری که برابر ۲ هستند + تعداد عناصر برابر ۱ ستون ۲ شمارش گردد، برای محاسبه عنصر $t_{۱۳}$، تعداد عناصری که برابر ۳ هستند + تعداد عناصر برابر ۱ ستون ۳ شمارش‌شده و غیره. برای عناصری که مقدار $r_{ik}\neq k$ است مقدار ۰ را جایگزین کنید. این محاسبات در ادامه آمده است.

$$R_F=\begin{bmatrix} - & ۳ & ۳ & ۳ \\ ۱ & - & ۱ & ۴ \\ ۱ & ۴ & - & ۴ \\ ۲ & ۲ & ۳ & - \end{bmatrix} \rightarrow T_F=\begin{bmatrix} - & ۰ & ۳+۱ & ۰ \\ ۲+۱ & - & ۰ & ۱+۰ \\ ۱+۰ & ۰ & - & ۲+۱ \\ ۰ & ۲+۱ & ۱+۰ & - \end{bmatrix}=\begin{bmatrix} - & ۰ & ۴ & ۰ \\ ۳ & - & ۰ & ۱ \\ ۱ & ۰ & - & ۳ \\ ۰ & ۳ & ۱ & - \end{bmatrix}$$

عناصر ماتریس T_F نشان‌دهنده عدد انتقال کمان است. برای مثال، $t_{۱۲}=۳$ به این معنی است که کمان ۲ $\rightarrow$ ۱، ۳ مرتبه در کل جابه‌جایی شبکه مورد استفاده قرار گرفته است. به‌عبارت بهتر، با حذف این کمان، ۳ مسیر بهینه دچار اختلال خواهد شد. در ادامه عدد انتقال و ارزش انتقال هر کمان آمده است.

$t_{۱۲}=۰,\quad t_{۱۳}=۴,\quad t_{۱۴}=۰\quad t_{۲۱}=۳,\quad t_{۲۳}=۰,\quad t_{۲۴}=۱$

$t_{۳۱}=۱,\quad t_{۳۲}=۰,\quad t_{۳۴}=۳\quad t_{۴۱}=۰,\quad t_{۴۲}=۳,\quad t_{۴۳}=۱$

$t=۰+۴+۰+۳+۰+۱+۱+۰+۳+۰+۳+۱=۱۶$

$w_{۱۲}=\frac{۰}{۱۶}=۰,\quad w_{۱۳}=\frac{۴}{۱۶}=۲۵\%,\quad w_{۱۴}=\frac{۰}{۱۶}=۰$

$w_{۲۱}=\frac{۳}{۱۶}=۱۸/۷۵\%,\quad w_{۲۳}=\frac{۰}{۱۶}=۰,\quad w_{۲۴}=\frac{۱}{۱۶}=۶/۲۵\%$

$w_{۳۱}=\frac{۱}{۱۶}=۶/۲۵\%,\quad w_{۳۲}=\frac{۰}{۱۶}=۰,\quad w_{۳۴}=\frac{۳}{۱۶}=۱۸/۷۵\%$

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

$$w_{۴۱}=\frac{۰}{۱۶}=۰, \qquad w_{۴۲}=\frac{۳}{۱۶}=۱۸/۷۵\%, \qquad w_{۴۳}=\frac{۱}{۱۶}=۶/۲۵\%$$

در نتیجه، ارزش انتقال کمان‌های $۳\to۱$ برابر $\frac{۴}{۱۶}$ یا ۲۵%، ارزش انتقال کمان‌های $۱\to۲$، $۴\to۳$ و $۲\to۴$ برابر $\frac{۳}{۱۶}$ یا $۱۸/۷۵\%$ ، ارزش انتقال کمان‌های $۴\to۲$، $۱\to۳$ و $۳\to۴$ برابر $\frac{۱}{۱۶}$ یا $۶/۲۵\%$ و ارزش انتقال کمان‌های $۲\to۱$، $۴\to۱$، $۳\to۲$ و $۲\to۳$، $۱\to۴$ برابر ۰ است. به‌عبارت‌بهتر، اگر کمان $۳\to۱$ حذف شوند، ۲۵ درصد و با حذف هر یک از کمان‌های $۱\to۲$، $۴\to۳$ و $۲\to۴$، $۱۸/۷۵\%$، و با حذف هریک از کمان‌های $۴\to۲$، $۱\to۳$ و $۳\to۴$، $۶/۲۵\%$ از جابه‌جایی‌های بهینه شبکه دچار آسیب خواهد شد و با حذف کمان‌های زاید $۲\to۱$، $۳\to۲$ و $۲\to۳$ خللی در جابه‌جایی‌ها ایجاد نخواهد شد. ارزش انتقال ۰ برای کمان‌ها به معنی عدم وجود کمان یا حذف آن کمان بدون ایجاد خلل در جابه‌جایی‌های بهینه شبکه است.

برای محاسبه عدد و ارزش انتقال هرگره، تعداد واسطه‌های گره را محاسبه کنید. برای محاسبه واسطه‌های گره ۱، ۲، ۳ و ۴ لازم است شماره گره i در غیر ستون i ماتریس R_F شمارش گردد. برای مثال، جهت محاسبه تعداد واسطه‌های گره ۱، ستون ۱ را حذف کرده و عدد ۱ را در عناصر باقیمانده ماتریس مسیر شمارش کنید. به‌کمک ماتریس‌های زیر واسطه‌های هر گره بسادگی قابل استخراج خواهد بود.

$$R_F=\begin{bmatrix}\square & ۳ & ۳ & ۳\\ - & ۱ & ۴\\ ۴ & - & ۴\\ ۲ & ۳ & -\end{bmatrix}, R_F=\begin{bmatrix}- & \square & ۳ & ۳\\ ۱ & & ۱ & ۴\\ ۱ & & - & ۴\\ ۲ & & ۳ & -\end{bmatrix},$$

$$R_F=\begin{bmatrix}- & ۳ & \square & ۳\\ ۱ & - & & ۴\\ ۱ & ۴ & & ۴\\ ۲ & ۲ & & -\end{bmatrix}, R_F=\begin{bmatrix}- & ۳ & ۳ & \square\\ ۱ & - & ۱ & \\ ۱ & ۴ & - & \\ ۲ & ۲ & ۳ & \end{bmatrix}$$

بنابراین، تعداد واسطه‌ها برای گره ۱ برابر ۱، تعداد واسطه‌ها برای گره ۲ برابر ۱، تعداد واسطه‌ها برای گره ۳ برابر ۲ و تعداد واسطه‌ها برای گره ۴ برابر ۱ خواهد بود. درنتیجه کل واسطه ها برابر ۵ است. چون در ماتریس R_F عنصر ∞ وجود ندارد درآنصورت، هرگره به تعداد $۶=(۱-۴)۲$ جابه‌جایی دارد. عدد و ارزش انتقال هر گره به‌ترتیب زیر قابل محاسبه خواهد بود:

$$t_۱=۱+۶=۷, \qquad t_۲=۱+۶=۷, \qquad t_۳=۲+۶=۸, \qquad t_۴=۱+۶=۷$$

$$w_۱=\frac{۱+۶}{۵+۲\times۴\times۳}=\frac{۷}{۲۹}=۲۴/۱۴\ \%\ ,\ w_۲=\frac{۱+۶}{۵+۲\times۴\times۳}=\frac{۷}{۲۹}=۲۴/۱۴\ \%$$

$$w_۳=\frac{۲+۶}{۵+۲\times۴\times۳}=\frac{۸}{۲۹}=۲۷/۵۸\ \%\ ,\ w_۴=\frac{۱+۶}{۵+۲\times۴\times۳}=\frac{۷}{۲۹}=۲۴/۱۴\ \%$$

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

در نتیجه، ارزش انتقال گره‌های ۱، ۲ و ۴ برابر ۲۴/۱۴ درصد و گره ۳ برابر ۲۷/۵۸ درصد خواهد بود. به‌عبارت بهتر، با حذف گره ۱، ۲ یا ۴، ۲۴/۱۴ درصد و با حذف گره ۳، ۲۷/۵۸ درصد از جابجایی‌های بهینه شبکه دچار اختلال خواهند شد.

٦ نتیجه و جمع‌بندی

در این مقاله الگوریتم جدیدی بانام الگوریتم مستطیل آبشاری براساس بهبود و بازتعریف چند الگوریتم دقیق برای محاسبه کوتاه‌ترین مسافت و مسیر بین هرزوج گره درشبکه‌های کوتاه‌ترین مسیر به‌همراه ماتریس انتقال که کاربرد مهمی در تحلیل حساسیت و بهینه‌سازی مجدد اینگونه شبکه‌ها دارد، ارایه شده است. مزیت اساسی الگوریتم مستطیل آبشاری نسبت به الگوریتم‌های مشابه این است که تمام محاسبات و عملیات ریاضی این الگوریتم، در قالب $n-۲$ مستطیل که معرف روابط ریاضی هستند، پیاده‌سازی شده است. الگوریتم مستطیل آبشاری، از طرفی منجر به کاهش قابل توجه حجم محاسبات و درنتیجه کاهش زمان انجام محاسبات در الگوریتم‌های مشابه از قبیل الگوریتم‌های هو و فلوید- وارشال شده است (به‌جهت وابسته کردن محاسبات ماتریس مسیر به ماتریس مسافت) و از طرفی دیگر، به‌دلیل اجرای مستطیلی این الگوریتم، در مسایل بزرگ بسیار جذاب و ساده خواهد بود. علاوه بر این، فهم الگوریتم برای اهداف آموزشی بسیار مناسب و آسان‌تر خواهد بود.

منابع

[1] Warshall, S., (1962). A theorem on boolean matrices. Journal of the ACM, 9, 11-12.
[2] Floyd, R. W., (1962). Algorithm 97: Shortest path. Communications of the ACM, 5, 345.
[3] Farbey, B. A., Land, A. H., Murchland, J. D., (1967). The cascade algorithm for finding all shortest distances in a directed graph. Management Science, 14(1), 19-28.
[4] Hu, T., (1967). Revised matrix algorithms for shortest paths. SIAM Journal of Applied Mathematics, 15, 207-218.
[5] Aini, A., Salehipour, A., (2012). Speeding up the floyd-warshall algorithm for the cycled shortest path problem. Applied Mathematics Letter, 25, 1-5.
[6] Dijkstra, E. W., (1959). A note on two problems in connection with Graphs. Numerische Mathematik, 1, 269–271.
[7] Ahuja, R. K., Magnanti, T. L., Orlin, J. B., (1993). Network flows: theory, algorithms, and applications. John Wiley & Sons.
[8] Hougardy, S., (2010). The floyd–warshall algorithm on graphs with negative cycles. Information Processing Letters, 110, 279–281.
[9] Lozano, L., Medaglia, A. L., (2013). On an exact method for the constrained shortest path problem. Computers & Operations Research, 40, 378–384.
[10] Santos L., Coutinho-Rodrigues, J., Current J. R., (2007). An improved solution algorithm for the constrained shortest path problem. Transportation Research, 41, 756–71.
[11] Wang, L., Johnson, E. L., Sokol, J. S., (2005). A multiple pairs shortest path algorithm. Transportation Science, 39, 465-476.
[12] Carre, B. A., (1971). An algebra for network routing problems. Journal of the Institute of Mathematics and its Applications, 7, 273-294.
[13] Goto, S., T., Ohtsuki, Yoshimura, T., (1976). Sparse matrix techniques for the shortest path problem, IEEE Transaction on CAS, 23, 752-758.
[14] Takaoka, T., (2004). A faster algorithm for the all-pairs shortest path problem and its application. Lecture Notes in Computer Science, 3106, 278–289.
[15] Duin, C. W., (2007). Two fast algorithms for all-pairs shortest paths. Computer & Operations Research, 34, 2827-2839.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]

[16] Demetrescu, C., Italiano, G. F., (2007). Algorithmic techniques for maintaining shortest routes in dynamic networks. Electronic notes in theoretical computer science, 171, 3-15.
[17] Chan, T. M., (2008). All-pairs shortest paths with real weights in $O(n^3 / \log n)$ time. Algorithmica, 50, 236-243.
[18] Mohemmed, A. W., Sahoo, N. C., Geok, T. K., (2008). Solving shortest path problem using particle swarm optimization. Applied soft computing, 8, 1643–1653.
[19] Mohamed, C., Bassem, J., Taicir, L., (2010). A genetic algorithms to solve the bicriteria shortest path problem. Electronic notes in discrete mathematics, 36, 851-858.
[20] Yuster, R., (2012). Approximate Shortest Paths in Weighted Graphs. Journals of Computer and System Science, 78, 632-637.
[21] Peng, W., Hu, X., Zhao, F., Su, J., (2012). A fast algorithm to find all-pairs shortest paths in complex networks. Procedia Computer Science, 9, 557 – 566.
[22] Doerr, B., Johannsen, D., Kotzing, T., Neumannc, F., Thile, M., (2013). More effective crossover operators for the all-pairs shortest path problem. Theoretical Computer Science, 471, 12-26.
[23] Huguet, M., Kirchler, D., Parent, P., Calvo, R. W., (2013). Efficient algorithms for the 2- way multi modal shortest path problem. Electronic notes in discrete mathematics, 41, 431–437.
[24] Wang, J., Kaempke, T., (2004). Shortest route computation in distributed systems. Computers & operations research, 31, 1621–1633.
[25] Ambrosino, D., Sciomachen, A., (2014). An algorithmic framework for computing shortest routes in urban multimodal networks with different criteria. Procedia - Social and Behavioral Sciences, 108, 139-152.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.4.5.6]