بهینه‌سازی مسیرهای خروج اضطراری در معادن زیرزمینی بزرگ با کاربرد الگوریتم دیکسترا

سید محمد اسماعیل جلالی [i] ؛ مهدی نوروزی [ii]

چکیده

با وجود آنکه برای حفظ و افزایش سطح ایمنی در معادن زیرزمینی، تدابیری در راستای جلوگیری و کنترل حادثه اندیشیده می‌شود، اما گاهی دور شدن از منطقه خطر و گریز به سطح زمین یا محل‌های امن دیگر در داخل شبکه یک معدن زیرزمینی، تنها راه ممکن برای حفظ ایمنی است. بدیهی است که در این شرایط باید بهترین مسیر گریز از محل بروز حادثه تا یک مکان امن یافته و مورد توجه قرار گیرد. تنها روشی که تاکنون برای یافتن مسیرهای گریز بهینه ارائه گردیده، روش مبتنی بر استفاده از الگوریتم برنامه ریزی پویا است. مهمترین اشکال در استفاده از این الگوریتم پیچیدگی زمانی آن در شبکه‌های بزرگ معدنی است. در این مقاله از الگوریتم دیکسترا که دارای پیچیدگی زمانی بسیار کمتری است، برای یافتن کوتاه‌ترین فاصله زمانی بین هر نقطه فرضی در یک شبکه معدنی تا نقاط معلوم دیگر و نیز تعیین مسیر متناظر با کوتاه‌ترین فاصله زمانی بین نقاط یاد شده استفاده شده است. در پایان مقاله، الگوریتم‌های فلوید وارشال و دیکسترا بر روی شبکه زیرزمینی یک معدن زغال‌سنگ، اجرا شده و نتایج بدست آمده با یکدیگر مقایسه شده است.

کلمات کلیدی

خروج اضطراری، ایمنی،الگوریتم دیکسترا، الگوریتم فلوید وارشال، معادن زیرزمینی.

Improvement of Emergency Escape Routes in Large Underground Mines Using Dijkstra Algorithm

S. E. Jalali; M. Noroozi

ABSTRACT

In order to improve the safety of underground mines and also avoid and control possible accident, many solutions have been thought; though sometimes running away from the danger zone and getting to the ground surface or other possible safe place inside the underground mine's network is the only viable way to keep safe. Naturally, finding the best escape route from the accident zone to a safe place should be of high concern. The only feasible approach for finding the best escape route is through the Dynamic programming Algorithm, though its application is huge underground network is time-related complications. In this paper, the Dijkstra algorithm, which is less time-related complications, is applied in order to determine the shortest escape time from an assumed point to alternative points of an underground mine network and also their corresponding routes between assumed points. At the end of this article, the Dijkstra and Floyd-Warshall algorithms were performed on the coal mine underground network and results were compared.

KEYWORDS

Emergency Exit, Safety, Dijkstra Algorithm, Floyd-Warshall Algorithm, Underground Mines.

[i] استادیار دانشکده مهندسی معدن، نفت و ژئوفیزیک، دانشگاه صنعتی شاهرود، Email: jalalisme@shahroodut.ac.ir

[ii] دانشجوی دکترای دانشکده مهندسی معدن، نفت و ژئوفیزیک، دانشگاه صنعتی شاهرود، Email: mnoroozi.mine@gmail.com

۱- مقدمه

معدنکاری به ویژه نوع زیرزمینی آن یکی از حرفه‌های پرمخاطره به شمار می‌آید. معدنکاران معادن زیرزمینی در مقابل حوادث و خطراتی از جمله مسمومیت‌ها و انواع بیماری‌های شغلی حاد و مزمن، خفگی ناشی از انتشار گاز، آتش‌سوزی و انفجار، ریزش سقف، آب‌گرفتگی و ... قرار دارند.

حادثه هنگامی اتفاق می‌افتد که انسان در برابر یک منبع خطر (نظیر آتش، برق فشار قوی، اجسام دارای انرژی جنبشی، گازهای سمی و ...) قرار گرفته و انرژی به بدن انسان منتقل شود. بنابراین افزایش ایمنی به یکی از سه طریق ممکن است که عبارتند از:

الف- دور نگه داشتن منبع خطر از انسان: با این کار، تماس بین عامل خطرساز یا مکان پرخطر با انسان قطع شده و عامل حادثه به بدن انسان منتقل نمی‌شود. گریز از محل بروز حادثه یا استفاده از ابزار عایق برای کار با برق فشار قوی نمونه‌هایی از چگونگی دور نگهداشتن انسان از منبع خطر است.

ب- کاهش خطر یا عامل بروز حادثه: در این حالت عامل خطر یا حادثه ضعیف شده یا حتی از بین می‌رود و در صورت تماس با انسان باعث بروز حادثه نمی‌شود مانند رقیق نمودن یا کاهش عیار گازهای سمی با کاربرد سیستم تهویه در معدن.

ج- افزایش مقاومت بدن انسان: با انجام این کار می‌توان مقاومت بدن را در مقابل عوامل بروز حادثه افزایش داد، مانند واکسیناسیون در مقابل برخی بیماری‌های شغلی.

هر چند بهترین راه افزایش ایمنی، کاهش عامل بروز حادثه تا سطح ایمن و افزایش مقاومت انسان در مقابل حوادث است. اما گریز از محل حادثه به منظور دور شدن سریع از منبع انرژی موجود در محل بروز حادثه به ویژه در مواقع بروز آتش‌سوزی در معادن زیرزمینی یک موضوع طبیعی و منطقی‌ترین راه است. بنابراین برای فراهم کردن امکان گریز از حادثه در معادن زیرزمینی باید ضمن فرهنگ‌سازی و آموزش معدنکاران برای دور شدن از محل حادثه، مناسب‌ترین مسیرهای گریز تعیین و معرفی شوند.

با توجه به آنچه گفته شد، در این مقاله ابتدا الگوریتم دیکسترا[1] برای یافتن کوتاه‌ترین فاصله زمانی بین هر نقطه فرضی در یک شبکه معدنی تا نقاط معلوم دیگر ارائه شده و سپس نتایج آن با نتایج حاصل از الگوریتم فلوید وارشال[2] مقایسه شده است. در این شرایط، نقطه اول جایی است که یک معدنکار ممکن است در هنگام حادثه در آن نقطه حضور داشته باشد و نقاط دیگر، نقاط امنی هستند که برای گریز از خطر می‌تواند به آنها پناه ببرد.

۲- پیشینه مطالعات

دور شدن سریع از محل بروز حادثه و اهمیت کاهش فاصله زمانی بین دیدن خطر و تخلیه محیط همواره مورد توجه معدنکاران بوده است. کاستن از تاخیرها و بهبود آموزش معدنکاران دو عامل از مهمترین عوامل در حفظ جان معدنکاران در جریان آتش‌سوزی به شمار می‌رود. تاخیر، به عنوان مدت زمانی که در کشف آتش، اعلام خبر تخلیه محیط و بسیج افراد برای گریز تلف می‌شود، توصیف شده و آموزش معدنکاران به عنوان مهارت در استفاده از وسایل امداد شخصی و یافتن راه گریز مناسب و مطمئن از معدن تعریف شده است [۱].

اولین مورد از مهمترین کارهایی که در هنگام آتش‌سوزی اتفاقی باید انجام شود، عقب‌نشینی سریع معدنکاران حاضر در محل آتش‌سوزی، به ناحیه امن است [۲] و [۳]. مطالعات انجام شده نشان می‌دهد که احتمال بروز تلفات جانی در حین حوادث معدنی را می‌توان تنها با کاستن تأخیرها و بهبود شناخت راه‌های گریز تا اندازه‌ای نزدیک به نصف کاهش داد [۱].

با وجود اهمیت شناخت راه‌های گریز در هنگام حادثه، تاکنون تنها یک روش برای تعیین راه‌های گریز بهینه در معادن زیرزمینی در مواقع اضطراری ارائه شده است که مبتنی بر الگوریتم فلوید وارشال است [۴]. در این مقاله، با توجه به نقاط ضعف این الگوریتم، از الگوریتم دیکسترا برای یافتن راه‌های گریز در شبکه معدن زیرزمینی استفاده شده و مقایسه‌ای بین الگوریتم‌های فلوید وارشال و دیکسترا از جنبه‌های مختلف انجام شده است.

۳- مدلسازی شبکه معدنی

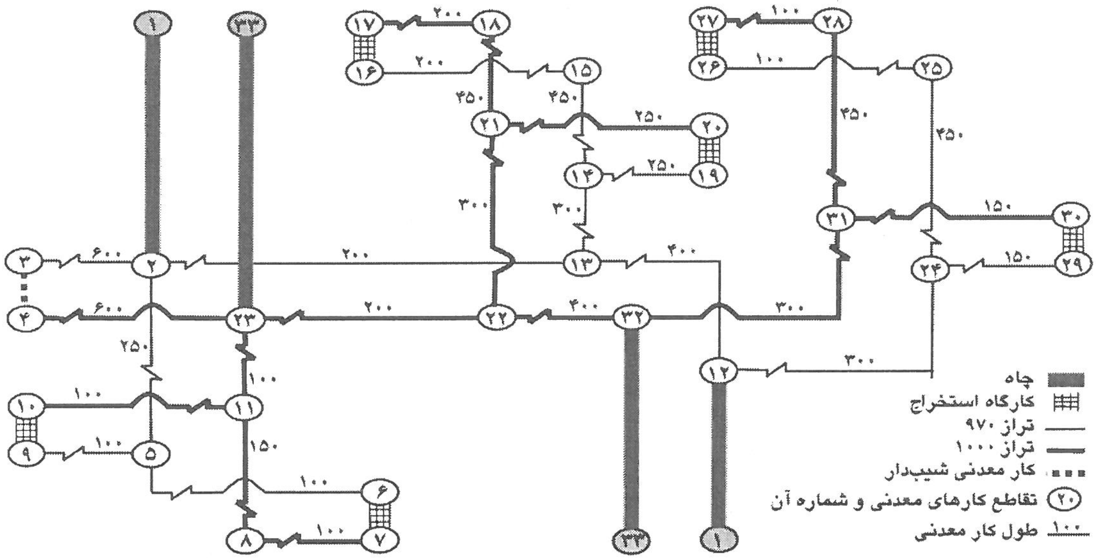
در این مقاله، شبکه معدنی به صورت یک گراف، مدلسازی شده است. برای مدلسازی یک شبکه معدنی با یک گراف، هر تقاطع از دو یا چند کار معدنی (تونل، چاه، رمپ و ...) با یک گره[3] و ارتباط دو گره مجاور، که یک کار معدنی خاص است با یک یال[4] نشان داده می‌شود. وزن هر یال بیانگر یک فاصله زمانی معینی از یک رأس به رأس بعدی است. ممکن است زمان رفت با زمان برگشت از یک گره به گره دیگر برابر نباشد. این موضوع در یال‌های متناظر با چاه‌های معدن، کارگاه‌های استخراج و یا در مسیرهای سربالایی و سرپایینی کارهای معدنی شیب‌دار امری طبیعی است.

فاصله زمانی، تابعی از طول کار معدنی متناظر با هر یال و ضریب سهولت عبور[5] (PSF) از آن کار معدنی است. ضریب سهولت عبور عبارتست از متوسط مقدار مسافتی که یک معدنکار در شرایط اضطراری در واحد زمان می‌تواند طی نماید. بهترین روش برای تعیین این ضریب در هر کار معدنی،

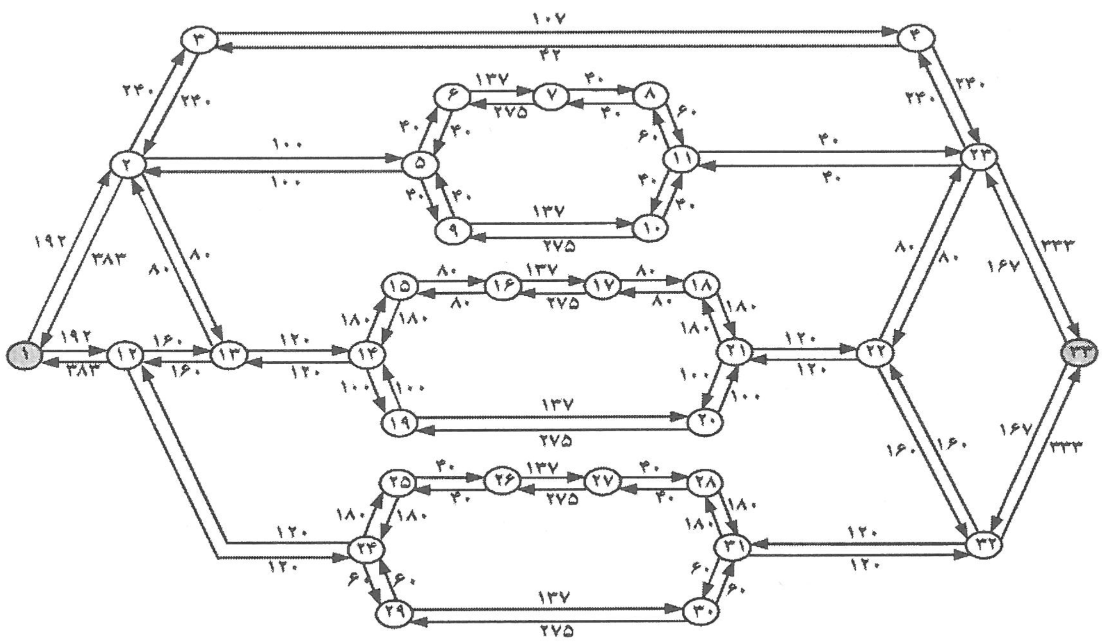
زمان‌سنجی آن است. مدل‌سازی شبکه معادن زیرزمینی بزرگ منجر به تشکیل یک گراف پیچیده می‌شود به طوری که کوتاه‌ترین مسیر بین دو گره خاص از شبکه مدل‌سازی شده به راحتی قابل تشخیص نیست. گراف یا شبکه پیچیده، شبکه‌ای است که نسبت تعداد یال‌ها به رئوس آن، زیاد باشد.

برای تشریح مسئله، یک شبکه معدنی ساده‌سازی شده از یک معدن زغال‌سنگ که به روش جبهه‌کار بلند ساده استخراج می‌شود در شکل (۱) نشان داده شده است. شبکه در دو طبقه و در ترازهای ۹۷۰ متری و ۱۰۰۰ متری احداث شده است و دارای دو چاه باربری و تهویه در سطح زمین (تراز ۱۲۰۰ متری) است [۵]. در این معدن ٦ کارگاه استخراج مطابق آنچه در شکل نشان داده شده جانمایی شده است. شبکه بدون مقیاس و فواصل نشان داده شده بر حسب متر می‌باشند.

مدل گرافی این شبکه در شکل (۲) نشان داده شده است. در این شکل، گره‌ها همانطور که در شکل (۱) نشان داده شده، محل تقاطع کارهای معدنی است. ارزش یال‌ها بیانگر فواصل زمانی بین دو گره بر حسب ثانیه است که از ضرب طول (بر حسب متر) و ضریب سهولت عبور از کار معدنی، PSF، (بر حسب متر بر ثانیه) محاسبه می‌شود.



شکل (۱): شبکه ساده‌سازی شده یک معدن زغال‌سنگ زیرزمینی [۵]

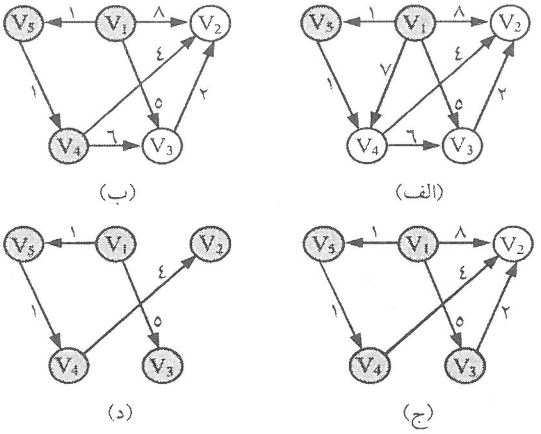


شکل (۲): گراف مدل‌سازی شده از شبکه معدن زیرزمینی زغال‌سنگ

٤- ارائه الگوریتم دیکسترا برای تحلیل شبکه معدنی

الگوریتم دیکسترا برای حل مسائل بهینه‌سازی و یافتن سریع‌ترین مسیر در یک گراف از یک رأس به سایر رئوس ارائه شده است. این الگوریتم با بهره‌مندی از منطق ریاضی تضمین می‌کند که جستجوی سریع‌ترین مسیر بر روی گراف، منجر به ارائه جواب بهینه شود [۶].

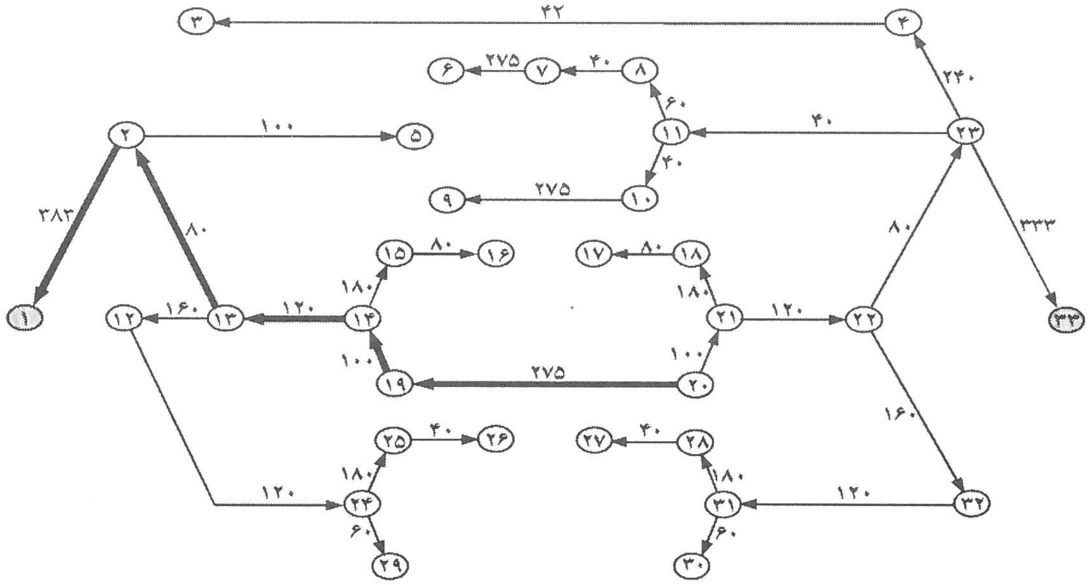
در روش مبتنی بر الگوریتم دیکسترا ابتدا مجموعه {V} مشتمل بر تمام رئوس گراف، انتخاب می‌شود. سپس دو مجموعه تهی، یکی به صورت زیرمجموعه‌ای تهی از یال‌ها، به نام {F} و دیگری زیرمجموعه‌ای تهی از رئوس، به نام {Y} تعریف می‌شوند. برای مقداردهی اولیه به مجموعه {Y} رأسی به صورت دلخواه انتخاب می‌شود. سپس کوتاهترین مسیر بین آن رأس و سایر رئوس مشخص می‌شود [۷]. برای مثال بر روی گراف نشان داده شده در شکل (۳-الف)، با فرض این که V_1 رأس دلخواه باشد، ابتدا رأس V_1 در مجموعه {Y} قرار می‌گیرد. سپس رأس V_5 عضو مجموعه {V} که از همه به V_1 نزدیک‌تر است، انتخاب و به مجموعه {Y} افزوده می‌شود و یال (V_1V_5) نیز به مجموعه {F} اضافه می‌شود. سپس مسیرهایی از V_1 به رئوس موجود در مجموعه {V-Y} (مجموعه V منهای Y خوانده می‌شود) مورد بررسی قرار می‌گیرند. بر این اساس، مسیر $(V_1V_5V_4)$ از بین مسیرهای محتمل به عنوان سریع‌ترین مسیر برگزیده می‌شود. این مسیر در شکل (۳-ب) نشان داده شده است. رأسی که در انتهای چنین مسیری باشد به مجموعه {Y} و یالی (بر روی مسیر) که آن رأس را در برگیرد، یعنی (V_5V_4)، به مجموعه {F} افزوده و سایر یال‌هایی که به آن رأس ختم می‌شوند، حذف می‌شوند (شکل (۳- ب)).



شکل (۳): مثالی از کاربرد روش دیکسترا

این فرآیند تا جایی ادامه می‌یابد که {Y} با {V} یعنی مجموعه همه رئوس برابر شود. در این حالت مجموعه {F} شامل یال‌های موجود بر روی مسیر متناظر با سریع‌ترین مسیر است. این مراحل در شکل‌های (۳-ج) و (۳-د) نشان داده شده است.

در شکل (۴) نتیجه اجرای الگوریتم دیکسترا بر روی گراف شکل (۲)، با انتخاب گره شماره ۲۰ به عنوان گره آغازین، نشان داده شده است. با توجه به این شکل، مسیرهای بهینه بین این گره و هر یک از گره‌های دیگر به همراه زمان متناظر با آن مسیرها قابل استخراج است. به عنوان مثال کوتاه‌ترین فاصله زمانی بین گره شماره ۲۰ تا سطح زمین (یعنی گره شماره ۱ در شکل (۲)) برابر ۹۴۲ ثانیه است. مسیر متناظر با آن نیز در شکل (۴) به صورت خطوط پر رنگ نشان داده شده است.



شکل (۴): مسیرهای بهینه بدست آمده از طریق اجرای الگوریتم دیکسترا بر روی گراف شکل (۲)

۵- تحلیل شبکه معدنی با الگوریتم فلوید وارشال

در این بخش برای یافتن مسیرهای گریز در شبکه معدن زیرزمینی از الگوریتم فلوید وارشال استفاده شده است. در این روش، یافتن مسیرهای گریز در دو مرحله انجام می‌شود. در مرحله اول کوتاه‌ترین فاصله زمانی بین تمام نقاط شبکه معدن محاسبه و در مرحله دوم مسیرهای متناظر با کوتاه‌ترین فاصله زمانی بین هر دو نقطه دلخواه با بهره‌گیری از روش ماتریس π مشخص می‌شود [۴].

الگوریتم فلوید- وارشال برای حل مسأله کوتاه‌ترین مسیرها بین هر جفت از رئوس گراف جهت‌دار بکار می‌رود. در این الگوریتم از رابطه بین مسیر p که در آن p مسیری منتهی به یکی از رئوس میانی در مسیرهای محتمل از رأس i به رأس j است، استفاده می‌شود.

اگر $d_{ij}^{(k)}$ وزن کوتاه‌ترین مسیر از رأس i به رأس j باشد که برای آن همه رأس‌های میانی در مجموعه $\{1,2,\ldots,k\}$ قرار دارند، وقتی که $k=0$ باشد، مسیر از رأس i به رأس j، هیچ رأس میانی ندارد. چنین مسیری حداکثر یک یال دارد و از این رو $d_{ij}^{(0)}=w_{ij}$ است. w_{ij} برابر وزن یال بین دو رأس است، اگر یالی بین دو رأس v_i و v_j وجود داشته باشد، در غیر این صورت، w_{ij}، برابر با بی‌نهایت و اگر $i=j$ باشد، w_{ij} برابر با صفر منظور می‌شود. ماتریسی که درایه‌های آن w_{ij} باشد، به نام ماتریس همجواری[۶] معروف است.

رابطه بازگشتی (۱) می‌تواند برای محاسبه مقادیر $d_{ij}^{(k)}$ به ترتیب افزایش مقادیر k استفاده شود [۸] و [۹].

$$d_{ij}^{(k)}=\begin{cases} w_{ij} & k=0 \quad \text{اگر} \\ \min(d_{ij}^{(k-1)},(d_{ik}^{(k-1)}+d_{kj}^{(k-1)})) & k\geq 1 \quad \text{اگر} \end{cases} \qquad (۱)$$

چون برای هر مسیر، همه رأس‌های میانی در مجموعه $\{1,2,\ldots,n\}$ واقع هستند، بنابراین:

$$D^{n}=(d_{ij}^{(n)}) \qquad (۲)$$

همزمان با محاسبه ماتریس‌های $D^{(k)}$ در الگوریتم فلوید- وارشال می‌توان ماتریسی که به نام ماتریس ماقبل[۷] یا π معروف است را نیز محاسبه کرد. $\pi_{ij}^{(k)}$ به عنوان ما قبل رأس j در کوتاه‌ترین مسیر از رأس i به رأس j با رأس‌های میانی واقع در مجموعه $\{1,2,\ldots,k\}$ تعریف می‌شود. هر یک از درایه‌های این ماتریس نشان‌دهنده شماره گره‌ای است که بر روی کوتاه‌ترین مسیر و درست قبل از گره j واقع است.

وقتی $k=0$ است، کوتاه‌ترین مسیر از i به j هیچ رأس میانی ندارد. بنابراین:

$$\pi_{ij}^{(0)}=\begin{cases} N & w_{ij}=\infty \;,\; i=j \quad \text{اگر} \\ i & w_{ij}<\infty \;,\; i\neq j \quad \text{اگر} \end{cases} \qquad (۳)$$

که در آن N نشان می‌دهد گره‌ای قبل از گره j وجود ندارد.

برای $k\geq 1$ مسیر $i\rightarrow k\rightarrow j$ در نظر گرفته می‌شود که در آن $k\neq j$، عنصر ما قبل j همان عنصری است که به عنوان ماقبل j در کوتاه‌ترین مسیر از k به j با همه رأس‌های میانی واقع در $\{1,2,\ldots,k\text{-}1\}$ انتخاب می‌شود. در غیر اینصورت همان ماقبل j انتخاب می‌شود که روی کوتاه‌ترین مسیر از i، با همه رأس‌های میانی واقع در $\{1,2,\ldots,k\text{-}1\}$ انتخاب شده بود.

به عبارت دیگر برای $k>1$:

$$\pi_{ij}^{(k)}=\begin{cases} \pi_{ij}^{(k-1)} & d_{ij}^{(k-1)}\leq d_{ik}^{(k-1)}+d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & otherwise \end{cases} \qquad (۴)$$

با استفاده از (۴) و با مقداردهی به k از ۱ تا n می‌توان ماتریس $\pi_{ij}^{(k)}$ را محاسبه نمود [۶] و [۱۰].

الگوریتم مورد نظر بر روی گراف مدل‌سازی شده از شبکه معدن که در شکل (۲) نشان داده شده است اجرا شده و ماتریس‌های D و π برای این گراف به ترتیب در شکل‌های (۵) و (۶) نشان داده شده است. با استفاده از این دو ماتریس می‌توان سریع‌ترین مسیر بین هر دو تقاطع از شبکه معدن را یافت.

درایه (i,j) از ماتریس $D^{(33)}$ نشان دهنده کمترین فاصله زمانی بین گره i و گره j در حرکت از گره i به گره j است. به عنوان مثال کوتاه‌ترین فاصله زمانی بین گره شماره ۲۰ تا گره شماره ۱ در شکل (۲) با استفاده از ماتریس $D^{(33)}$ برابر ۹۴۲ ثانیه است.

به منظور یافتن مسیر متناظر با این فاصله زمانی از ماتریس $\pi^{(33)}$ استفاده می‌شود. این ماتریس گره‌های پشت سر هم در مسیر را به ترتیب از گره آخر به گره اول مشخص می‌کند. برای این منظور، ابتدا باید درایه متناظر با مسیر مورد نظر از ماتریس $\pi^{(33)}$ که در اینجا $\pi_{20,1}$ است، مشخص شود. مقدار $\pi_{20,1}$ برابر ۲ است. مقدار این درایه بیانگر شماره گره ماقبل آخر است یعنی در مسیر از گره ۲۰ به گره ۱، قبل از گره ۱، گره ۲ قرار دارد. مسیر بین گره‌های ۲۰ و ۱ تا اینجا به صورت (۵) مشخص می‌شود:

(۵) ۲۰ → ؟...؟ → ۲ → ۱

اکنون، هدف یافتن مسیر بین گره‌های ۲۰ و ۲ است. با توجه به مطالب گفته شده باید درایه $\pi_{20,2}$ از ماتریس $\pi^{(33)}$ مشخص شود که برابر ۱۳ است و این بدان معنی است که قبل از گره ۲، گره ۱۳ قرار دارد. اینک با مشخص شدن گرهی دیگر، مسیر به شکل (۶) کامل‌تر می‌شود:

(۶) ۲۰ → ؟...؟ → ۱۳ → ۲ → ۱

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0	192	432	539	292	332	469	509	332	469	509	192	277	392	572	652	789	869	492	629	729	629	549	312	492	532	669	709	372	509	569	689	882
2	383	0	240	347	100	140	277	317	140	277	317	240	80	200	380	460	597	677	300	437	537	437	357	360	540	580	717	757	420	557	617	597	690
3	623	240	0	107	340	380	487	447	380	427	387	480	320	440	620	700	807	727	540	647	547	427	347	600	780	820	927	887	660	767	707	587	680
4	665	282	42	0	382	422	380	340	422	320	280	522	362	482	662	742	700	620	582	540	440	320	240	642	822	862	820	780	702	660	600	480	573
5	483	100	340	447	0	40	177	217	40	177	217	340	180	300	480	560	697	637	400	537	457	337	257	460	640	680	817	797	520	657	617	497	590
6	523	140	380	487	40	0	137	177	80	217	237	380	220	340	520	600	737	657	440	577	477	357	277	500	680	720	857	817	560	697	637	517	610
7	798	415	422	380	315	275	0	40	355	140	100	655	495	615	795	875	600	520	715	440	340	220	140	775	955	995	720	680	835	560	500	380	473
8	838	455	382	340	355	315	40	0	375	100	60	695	535	655	835	835	560	480	675	400	300	180	100	815	995	955	680	640	795	520	460	340	433
9	523	140	380	457	40	80	217	237	0	137	177	380	220	340	520	600	677	597	440	517	417	297	217	500	680	720	797	757	560	637	577	457	550
10	798	415	362	320	315	355	140	100	275	0	40	655	495	615	795	815	540	460	655	380	280	160	80	775	955	935	660	620	560	637	577	457	550
11	838	455	322	280	355	375	100	60	315	40	0	695	535	655	835	775	500	420	615	340	240	120	40	795	935	895	620	580	735	460	400	280	373
12	383	240	480	587	340	380	517	557	380	517	557	0	160	280	460	540	677	757	380	517	617	657	597	120	300	340	477	517	180	317	377	497	830
13	463	80	320	427	180	220	357	397	220	357	397	160	0	120	300	380	517	597	220	357	457	517	437	280	460	500	637	677	340	477	537	657	770
14	583	200	440	547	300	340	477	517	340	477	517	280	120	0	180	260	397	477	100	237	337	457	537	400	580	620	757	797	460	597	657	617	870
15	763	380	620	727	480	520	657	697	520	657	697	460	300	180	0	80	217	297	280	417	477	597	677	580	760	800	937	977	640	777	837	757	995
16	843	460	700	807	560	600	737	697	600	677	637	540	380	260	80	0	137	217	360	497	397	517	597	660	840	880	1017	977	720	857	797	677	930
17	1118	735	742	700	835	875	600	560	815	540	500	815	655	535	355	275	0	80	635	360	260	380	460	935	1115	1155	880	840	995	720	660	540	793
18	1198	815	662	620	775	795	520	480	735	460	420	895	735	615	435	355	80	0	555	280	180	300	380	975	1115	1075	800	760	915	640	580	460	713
19	683	300	540	647	400	440	577	537	440	517	477	380	220	100	280	360	497	417	0	137	237	357	437	500	680	720	857	817	560	697	637	517	770
20	958	575	582	540	675	715	440	400	655	380	340	655	495	375	555	635	360	280	275	0	100	220	300	775	955	995	720	680	835	560	500	380	633
21	1058	675	482	440	595	615	340	300	555	280	240	755	595	475	615	535	260	180	375	100	0	120	200	795	935	895	620	580	735	460	400	280	533
22	958	575	362	320	475	495	220	180	435	160	120	795	655	595	735	655	380	300	495	220	120	0	80	675	815	775	500	460	615	340	280	160	413
23	878	495	282	240	395	415	140	100	355	80	40	735	575	675	815	735	460	380	575	300	200	80	0	755	895	855	580	540	695	420	360	240	333
24	503	360	600	707	460	500	637	677	500	637	657	120	280	400	580	660	797	837	500	637	657	537	617	0	180	220	357	397	60	197	257	377	710
25	683	540	780	887	640	680	817	857	680	817	797	300	460	580	760	840	977	977	680	817	797	677	757	180	0	40	177	217	240	377	397	517	850
26	723	580	820	927	680	720	857	817	720	797	757	340	500	620	800	880	1117	937	720	857	757	637	717	220	40	0	137	177	280	417	357	477	810
27	998	855	862	820	955	995	720	680	935	660	620	615	775	895	1075	1155	880	800	995	720	620	500	580	495	315	275	0	40	555	280	220	340	673
28	1038	895	822	780	935	955	680	640	895	620	580	655	815	935	1115	1115	840	760	955	680	580	460	540	535	355	315	40	0	515	240	180	300	633
29	563	420	660	767	520	560	697	657	560	637	597	180	340	460	640	720	857	777	560	697	597	477	557	60	240	280	417	377	0	137	197	317	650
30	838	695	702	660	795	835	560	520	775	500	460	455	615	735	915	995	720	640	835	560	460	340	420	335	515	555	280	240	275	0	60	180	513
31	898	755	642	600	755	775	500	460	715	440	400	515	675	795	975	935	660	580	775	500	400	280	360	395	535	495	220	180	335	60	0	120	453
32	1018	735	522	480	635	655	380	340	595	320	280	635	795	755	895	815	540	460	655	380	280	160	240	515	655	615	340	300	455	180	120	0	333
33	1045	662	449	407	562	582	307	267	522	247	207	802	742	842	982	902	627	547	742	467	367	247	167	682	822	782	507	467	622	347	287	167	0

شکل (۵): ماتریس نهایی $\mathbf{D} = \mathbf{D}^{(33)}$ مربوط به گراف شکل (۲)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	*N*	1	2	3	2	5	6	7	5	9	10	1	2	13	14	15	16	17	14	19	20	23	11	12	24	25	26	27	24	29	30	31	23
2	2	*N*	2	3	2	5	6	7	5	9	10	13	2	13	14	15	16	17	14	19	20	23	11	12	24	25	26	27	24	29	30	22	23
3	2	3	*N*	3	2	5	8	11	5	11	23	13	2	13	14	15	18	21	14	21	22	23	4	12	24	25	28	31	24	31	32	22	23
4	2	3	4	*N*	2	5	8	11	5	11	23	13	2	13	14	15	18	21	14	21	22	23	4	12	24	25	28	31	24	31	32	22	23
5	2	5	2	3	*N*	5	6	7	5	9	10	13	2	13	14	15	16	21	14	19	22	23	11	12	24	25	26	31	24	29	32	22	23
6	2	5	2	3	6	*N*	6	7	5	9	8	13	2	13	14	15	16	21	14	19	22	23	11	12	24	25	26	31	24	29	32	22	23
7	2	5	4	23	6	7	*N*	7	5	11	8	13	2	13	14	15	18	21	14	21	22	23	11	12	24	25	28	31	24	31	32	22	23
8	2	5	4	23	6	7	8	*N*	10	11	8	13	2	13	14	17	18	21	20	21	22	23	11	12	24	27	28	31	30	31	32	22	23
9	2	5	2	23	9	5	6	11	*N*	9	10	13	2	13	14	15	18	21	14	21	22	23	11	12	24	25	28	31	24	31	32	22	23
10	2	5	4	23	9	5	8	11	10	*N*	10	13	2	13	14	17	18	21	20	21	22	23	11	12	24	27	28	31	30	31	32	22	23
11	2	5	4	23	9	7	8	11	10	11	*N*	13	2	13	14	17	18	21	20	21	22	23	11	29	26	27	28	31	30	31	32	22	23
12	12	13	2	3	2	5	6	7	5	9	10	*N*	12	13	14	15	16	17	14	19	22	32	11	12	24	25	26	27	24	29	30	31	32
13	2	13	2	3	2	5	6	7	5	9	10	13	*N*	13	14	15	16	17	14	19	22	23	11	12	24	25	26	27	24	29	30	31	23
14	2	13	2	3	2	5	6	7	5	9	10	13	14	*N*	14	15	16	17	14	19	22	21	22	12	24	25	26	27	24	29	30	22	23
15	2	13	2	3	2	5	6	7	5	9	10	13	14	15	*N*	15	16	17	14	19	18	21	22	12	24	25	26	27	24	29	30	22	23
16	2	13	2	3	2	5	6	11	5	11	23	13	14	15	16	*N*	16	17	14	19	18	21	22	12	24	25	26	31	24	29	32	22	23
17	2	13	4	23	2	5	8	11	10	11	23	13	14	15	16	17	*N*	17	14	21	18	21	22	12	24	25	28	31	24	31	32	22	23
18	2	13	4	23	9	7	8	11	10	11	23	13	14	15	16	17	18	*N*	20	21	18	21	22	29	26	27	28	31	30	31	32	22	23
19	2	13	2	3	2	5	6	11	5	11	23	13	14	19	14	15	16	21	*N*	19	20	21	22	12	24	25	26	31	24	29	32	22	23
20	2	13	4	23	2	5	8	11	10	11	23	13	14	19	14	15	18	21	20	*N*	20	21	22	12	24	25	28	31	24	31	32	22	23
21	2	13	4	23	9	7	8	11	10	11	23	13	14	19	16	17	18	21	20	21	*N*	21	22	29	26	27	28	31	30	31	32	22	23
22	2	5	4	23	9	7	8	11	10	11	23	24	2	19	16	17	18	21	20	21	22	*N*	22	29	26	27	28	31	30	31	32	22	23
23	2	5	4	23	9	7	8	11	10	11	23	13	2	19	16	17	18	21	20	21	22	23	*N*	29	26	27	28	31	30	31	32	22	23
24	12	13	2	3	2	5	6	7	5	9	23	24	12	13	14	15	16	21	14	19	22	32	22	*N*	24	25	26	27	24	29	30	31	32
25	12	13	2	3	2	5	6	7	5	9	23	24	12	13	14	15	16	21	14	19	22	32	22	25	*N*	25	26	27	24	29	28	31	32
26	12	13	2	3	2	5	6	11	5	11	23	24	12	13	14	15	16	21	14	19	22	32	22	25	26	*N*	26	27	24	29	28	31	32
27	12	13	4	23	2	5	8	11	10	11	23	24	12	13	14	15	18	21	14	21	22	32	22	25	26	27	*N*	27	24	31	28	31	32
28	12	13	4	23	9	7	8	11	10	11	23	24	12	13	14	17	18	21	20	21	22	32	22	25	26	27	28	*N*	30	31	28	31	32
29	12	13	2	3	2	5	6	11	5	11	23	24	12	13	14	15	16	21	14	19	22	32	22	29	24	25	26	31	*N*	29	30	31	32
30	12	13	4	23	2	5	8	11	10	11	23	24	12	13	14	15	18	21	14	21	22	32	22	29	24	25	28	31	30	*N*	30	31	32
31	12	13	4	23	9	7	8	11	10	11	23	24	12	13	14	17	18	21	20	21	22	32	22	29	26	27	28	31	30	31	*N*	31	32
32	12	5	4	23	9	7	8	11	10	11	23	24	12	19	16	17	18	21	20	21	22	32	22	29	26	27	28	31	30	31	32	*N*	32
33	2	5	4	23	9	7	8	11	10	11	23	24	2	19	16	17	18	21	20	21	22	23	33	29	26	27	28	31	30	31	32	33	*N*

شکل(۶): ماتریس نهایی $\pi = \pi^{(33)}$ مربوط به گراف شکل (۲)

به همین ترتیب از ماتریس $\pi^{(33)}$، مقدار درایه $\pi_{20,13}$ برابر ۱۴، $\pi_{20,14}$ برابر ۱۹ و با همین روال $\pi_{20,19}$ برابر ۲۰ است که در واقع نشان دهنده شماره گره شروع مسیر است. بدین ترتیب گره‌های واقع بر روی مسیر بهینه به صورت (۷) مشخص می‌شود:

$$۱ \rightarrow ۲ \rightarrow ۱۳ \rightarrow ۱۴ \rightarrow ۱۹ \rightarrow ۲۰ \qquad (۷)$$

۶- بحث

همانطور که در شکل‌های (۴)، (۵) و (۶) ملاحظه می‌شود برابری کاملی بین نتایج حاصل از الگوریتم دیکسترا و الگوریتم فلوید وارشال وجود دارد. به عنوان مثال، بر اساس الگوریتم دیکسترا، سریع‌ترین زمان برای رسیدن از گره ۲۰ به گره ۱ برابر با ۹۴۲ ثانیه است. فاصله زمان بین این دو گره و مسیر متناظر با آن با نتایج بدست آمده از الگوریتم فلوید وارشال یکسان است.

پیچیدگی زمانی[۸] الگوریتم فلوید وارشال $O(n^3)$ است، در حالی که پیچیدگی زمانی الگوریتم دیکسترا $O(n \log(n))$ است (n تعداد گره‌های موجود در شبکه است). باید توجه داشت که با هر بار اجرای الگوریتم دیکسترا، سریع‌ترین زمان دسترسی از یک گره مفروض به سایر گره‌ها قابل محاسبه است در حالی که با اجرای الگوریتم فلوید وارشال سریع‌ترین زمان دسترسی از هر گره به سایر گره‌ها بدست خواهد آمد. به عبارت دیگر با هر بار اجرای الگوریتم دیکسترا، تنها درایه‌های واقع بر یک سطر از ماتریس D که نشان دهنده نتایج سریع‌ترین زمان دسترسی از گره متناظر با سطر مورد نظر، با سایر گره‌ها می‌باشد، بدست خواهد آمد و برای کسب نتایج حاصل از اجرای الگوریتم فلوید وارشال، لازم است الگوریتم دیکسترا n بار یعنی به تعداد گره‌های موجود در شبکه، اجرا شود.

در زمان بروز حادثه، بررسی سریع‌ترین مسیر بین تمام نقاط واقع در شبکه معدنی غیر ضروری است و تنها مسیرهای بین نقاط حادثه دیده و نقاط امن و یا سطح زمین مورد نیاز است. برای تعیین چنین مسیرهایی، بررسی تمام مسیرها بین تمام نقاط در الگوریتم فلوید وارشال غیر ضروری و وقت گیر است. این در حالی است که الگوریتم دیکسترا می‌تواند برای یافتن سریع‌ترین مسیرها بین نقاط ضروری خاص مورد نیاز، اجرا شود. این موضوع در شبکه‌های پیچیده بسیار مهم است. در چنین شبکه‌هایی، الگوریتم دیکسترا با سرعت بالاتری اجرا می‌شود و زمان کمتری برای تعیین سریع‌ترین مسیرهای گریز، گرفته می‌شود.

برای مثال فرض شود یک معدن زیرزمینی دارای شبکه‌ای بزرگ با ۱۰۰۰ گره باشد. در این شرایط، پیچیدگی زمانی اجرای الگوریتم فلوید وارشال متناسب با عدد $O(n^3) = ۱۰^۹$ است. اما در همین شرایط، پیچیدگی زمانی اجرای الگوریتم دیکسترا بسیار کمتر و متناسب با عدد $O(n \log(n)) = ۳۰۰۰$ خواهد بود.

۷- نتیجه‌گیری

بسیاری از حوادث مرگبار معادن زیرزمینی زغال‌سنگ، جزء جدانشدنی این معادن به شمار می‌روند. با وجود تمام تدابیری که برای پیشگیری از بروز حادثه اندیشیده می‌شود گاهی تنها راه عملی برای در امان ماندن از حادثه، گریز از محل حادثه است. در این مقاله، پس از ارائه الگوریتم دیکسترا برای تعیین مسیرهای بهینه گریز، الگوریتم‌های فلوید وارشال و دیکسترا برای تعیین مسیرهای گریز و انتخاب کوتاه‌ترین مسیر در شبکه‌های پیچیده معادن زیرزمینی بزرگ به ویژه معادن زغال‌سنگ، مقایسه شده‌اند. کاربرد این الگوریتم‌ها، برابری کاملی را بین نتایج حاصل از الگوریتم دیکسترا و الگوریتم فلوید وارشال نشان می‌دهد. در مورد معادن زیرزمینی بزرگ با شبکه‌های پیچیده، الگوریتم دیکسترا به دلیل تعیین سریع‌ترین مسیرها بین نقاط مورد نیاز و ضروری (نقاط حادثه دیده و نقاط امن و یا سطح زمین) و در نتیجه صرف زمان کمتر برای اجرا، مناسب‌تر است.

با استفاده از این روش می‌توان در زمان کمتری نتایج حاصل از اجرای الگوریتم را به تابلوهای واقع در هر تقاطع از شبکه معدن که مسیر خروج اضطراری را مشخص می‌کند، ارسال نمود. بدین ترتیب می‌توان یکی از مهمترین علل آسیب دیدن معدنکاران در هنگام بروز حادثه، یعنی تأخیر در ترک محل حادثه دیده را به میزان بیشتری کاهش و در نتیجه ایمنی را افزایش داد.

۸- مراجع

[۱] Goodman, G., V., R.; Kissel, F., N.; "Important factors for escaping a mine fire", Operating Ideas, 1990.

[۲] Cwiek, B.; "Organization of a rescue operation in the polish mining industry and in particular a fire fighting operation", 21th International Conference of Safety in Mines, p.p. 229-236, 1985.

[۳] Barker-Read, G.R., Li, H.; "Automatic selection of safe egress routes away from underground fires", Mining Science and Technology 9, 289–308, 1989.

[۴] Jalali, S.E., Noroozi, M.; "Determination of the optimal escape routes of underground mine networks in emergency cases", Safety Science, 47, 1077–1082, 2009.

[۵] Wang, Y., J.; Mutmansky, J., M.; "Application of CPM procedures in mine ventilation", Proceeding of the 1st mine ventilation symposium, p.p. 159-168, 1982.

[۶] Cormen, T., H., et al.; "Intruduction to Algorithms", 2nd Edition, Mc Grow-Hill, 2001.

[۷] Dijkstra, E. W.; "A note on two problems in connexion with graphs", Numerische Mathematik 1, p.p. 269–271, 1959.

[۸] Floyd, R. W.; "Algorithm 97: Shortest Path", Communications of the ACM 5, no.6: 345, 1962.

[۹] Warshall, S.; "A theorem on Boolean matrices", J. ACM., 9, no. 1, 1962.

[۱۰] Neapolitan, R., E.; Naimipour, K.; "Foundation of Algorithms Using C++ Pseudocode", 3rd Edition, Jones and Bartleth Publisher, 2004.

۹- زیر نویس ها

---

1 Dijkstra
2 Floyd-Warshall
3 Vertex
4 Edge
5 Passage simplicity factor
6 Adjacency matrix
7 Predecessor matrix
8 Time-related complications